

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



“Diseño e implementación de un medidor trifásico
multifunción utilizando el IC ADE7754”

PRESENTADO POR:
FERNANDO ALBERTO ARÉVALO NAVAS
DANIEL ANTONIO CORTEZ FRANCO
DOUGLAS ALBERTO LÓPEZ HERNÁNDEZ

PARA OPTAR AL TÍTULO DE:
INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, FEBRERO DE 2005

UNIVERSIDAD DE EL SALVADOR

RECTORA :
Dra. María Isabel Rodríguez

SECRETARIA GENERAL :
Lic. Alicia Margarita Rivas de Recinos

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO :
Ing. Mario Roberto Nieto Lovo

SECRETARIO :
Ing. Oscar Eduardo Marroquín Hernández

ESCUELA DE INGENIERÍA ELECTRICA

DIRECTOR :
Ing. Luis Roberto Chévez Paz

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:
INGENIERO ELECTRICISTA

Título : "Diseño e implementación de un medidor trifásico multifunción utilizando el IC ADE7754"

Presentado por :
Fernando Alberto Arévalo Navas
Daniel Antonio Cortez Franco
Douglas Alberto López Hernández

Trabajo de Graduación aprobado por:

Docente Director :
Ing. José Roberto Ramos López

San Salvador, Febrero de 2005

Trabajo de Graduación Aprobado por:

Docente Director:

Ing. José Roberto Ramos López

AGRADECIMIENTOS

A Dios por ser tan bondadoso y estar conmigo en mi vida.

A mis padres por su apoyo incondicional y estar siempre conmigo a lo largo de toda mi vida.

A mis hermanos y hermanas que me brindan su apoyo y me dan ánimos a lo largo de toda mi vida.

A mis profesores que me proporcionaron los fundamentos teóricos y prácticos para poder finalizar con éxito mi presente trabajo de graduación.

A todos mis compañeros por estar siempre conmigo cuando yo los necesite.

Daniel Cortez.

AGRADECIMIENTOS

En primer lugar a Dios por su infinita gracia, misericordia, sabiduría y por sus constantes y abundantes bendiciones.

A mi madre Elvira por brindarme su amor y dedicación, por su apoyo en aquellos momentos difíciles y buenos consejos.

A mi amigo Andrés quien siempre me apoyo y me dio animo para seguir adelante con mis estudios.

A mis compañeros por su ayuda y brindarme apoyo en todas las situaciones difíciles, y haber hecho de mis momentos en la Universidad muy especiales.

Y a todos aquellos que contribuyeron en algún momento o de alguna forma para que yo alcanzara esta meta.

Douglas López.

Para Sandra Arévalo, mi hermana.

AGRADECIMIENTOS

A Jehová Dios y su hijo Jesucristo, por la vida y la salud, por acompañarme en todo momento, por darme fortaleza en todos los momentos que necesite.

A Ana, que me aconsejó, me regañó, me apoyó, lloró y se rió conmigo, por ser todo lo que es, por ser mi mamá.

A Fernando, por los consejos, por el apoyo, por el cariño a su manera, mi viejo, mi papá.

A Raúl, por el apoyo incondicional, por todo el cariño, mi hermano del alma.

A mi familia por estar pendiente, por el apoyo, por el cariño.

A mis amigos que me apoyaron, que compartieron sus alegrías, sus tristezas y su confianza conmigo.

A los amigos con quienes me desvele estudiando, con quienes estuvimos en la cancha, con quienes compartí los buenos y los malos momentos en mi vida universitaria.

A mis asesores por creer en nosotros, y por la ayuda que nos proporcionaron durante todo este proceso.

A mis profesores y los laboratoristas que compartieron su conocimiento conmigo.

A todos los que de una manera u otra me ayudaron a concluir esta etapa de mi vida...GRACIAS.

Fernando Arévalo.

CONTENIDO

CAPÍTULO	TÍTULO	PÁGINA
	INTRODUCCION	X
1. GENERALIDADES		
1.1.	Descripción del tema	1
1.2.	Objetivos	2
1.2.1.	Objetivo general	2
1.2.2.	Objetivos específicos	2
1.3.	Justificación	3
1.4.	Alcances	3
1.5.	Limitaciones	4
2. TEORIA BÁSICA		
2.1.	Conceptos básicos	5
2.2.	Instrumentos de medición	6
2.3.	Medidores de facturación	8
2.4.	Sistemas monofásicos	9
2.4.1.	Cálculo de potencia	9
2.4.2.	Factor de potencia	10
2.4.3.	Potencia activa	11
2.4.4.	Potencia reactiva	11
2.4.5.	Cuadrantes para la potencia reactiva y aparente	13
2.5.	Sistemas trifásicos	14
2.5.1.	Cálculo de potencia	15
2.5.2.	El factor de potencia	16

2.5.3. Componentes armónicas y Tasa de Distorsión Armónica (THD)	17
2.6. Transformadores de medida	18
2.6.1. Transformadores de intensidad	19
2.6.2. Transformadores de tensión	20
2.7. Medición de potencia trifásica y análisis vectorial	21
2.7.1. Medición Estrella de 3 conductores	21
2.7.2. Medición Delta de 3 conductores	25
2.7.3. Medición Estrella de 4 conductores	28
2.7.4. Medición Delta de 4 conductores	31

3. COMPONENTES ELECTRONICOS DEL MEDIDOR

3.1. El circuito integrado ADE7754	35
3.1.1. Características generales	35
3.1.2. Características de funcionamiento	36
3.1.3. Diagrama de los pines	37
3.1.4. Descripción de las funciones de los pines	37
3.1.5. Diagrama básico de conexión	39
3.2. El microcontrolador PIC16F877	40
3.2.1. Características generales	40
3.2.2. Características de funcionamiento	41
3.2.3. Diagrama de los pines	42
3.2.4. Descripción de las funciones de los pines	42
3.2.5. Comunicación asíncrona SCI	43
3.2.6. Comunicación síncrona SPI	44
3.3. La memoria RAM no volátil DS1245Y	46
3.3.1. Características generales	46
3.3.2. Características de funcionamiento	47
3.3.3. Diagrama de los pines	48
3.3.4. Descripción de las funciones de los pines	48

4. HARDWARE DEL MEDIDOR

4.1. Diagrama de bloques	51
4.2. Etapas del medidor de Energía	52
4.2.1. Tarjeta del Medidor de Energía	52
4.2.2. Tarjeta de la etapa de memoria	56
4.2.3. Tarjeta de Temporización	57

5. PROTOCOLO DE COMUNICACIÓN

5.1. Diagrama de bloques del sistema	59
5.2. Configuración de los protocolos implementados en el microcontrolador maestro	60
5.3. Interfaz serial SPI del ADE7754	61
5.4. Interrupciones internas con el ADE7754	64
5.5. Reloj de tiempo real	65
5.6. Memoria de 128Kx8	67

6. INTERFASE GRAFICA DEL USUARIO

6.1. El lenguaje de programación LabVIEW	68
6.2. Flujogramas de la interfase gráfica del usuario	69
6.2.1. Principal	69
6.2.2. Calibración	75
6.3. La interfase gráfica del usuario	76
6.3.1. VI Principal	76
6.3.2. VI de Calibración	81

7. CALIBRACIÓN DEL MEDIDOR

7.1. Fundamentos teóricos del IC ADE7754	82
7.2. Fórmulas utilizadas para la calibración del medidor	83
7.2.1. Calibración de ganancia de energía activa usando el registro LAENERGY	83
7.2.2. Calibración de la fase usando el registro LAENERGY	84
7.2.3. Como obtener el valor de potencia activa promedio	85
7.2.4. Como obtener el valor de potencia aparente promedio	85
7.2.5. Como obtener el valor de voltaje rms	86
7.2.6. Como obtener el valor de corriente rms	86
7.3. Tipos de conexiones disponibles para el IC ADE7754	87
7.4. Obtención de la constante del medidor	88
7.5. Procedimiento de calibración	89
7.5.1. Calibración del pulso de frecuencia CF utilizando el registro de energía activa LAENERGY	90
7.5.2. Calibración de la fase usando el registro LAENERGY	93
7.5.3. Ajuste fino del CFDEN	95
7.5.4. Guardando las referencias de magnitudes y de registros para medición	97
7.5.5. Resumen de la calibración	98
7.6. Resultados obtenidos	99
RECOMENDACIONES	XII
CONCLUSIONES	XIII
BIBLIOGRAFÍA Y FUENTES DE CONSULTA	XIV

ANEXOS

ANEXO	TÍTULO	PAG
A.1	Manual del usuario	1
A.2	Tarjeta programadora de microcontroladores	47
A.3	Lista de componentes y precios	50
A.4	Circuitos impresos de las etapas	53
A.5	Proceso de fabricación de los circuitos impresos sobre tarjetas de cobre	56
A.6	Glosario	58
A.7	Interfaz del puerto serial asíncrono SCI	60
A.8	Interfaz del puerto serial síncrono SPI	63
A.9	Flujogramas de los programas del microcontrolador	66
A.10	Código del programa del microcontrolador	94

LISTADO DE TABLAS

TABLA	TÍTULO	PAG
3.1	Descripción de las funciones de los pines del IC ADE7754	37-39
3.2	Descripción de las funciones de los pines del microcontrolador PIC16F877	42-43
3.3	Descripción de las funciones de los pines de la memoria RAM no volátil DS1245Y	48
7.1	Tipo de conexión para potencia activa	87
7.2	Tipo de conexión para potencia aparente	87
7.3	Parámetros para obtención de constante del medidor	88
7.4	Valores nominales para calibración	89
7.5	Cálculo del error del medidor por el patrón de calibración WECO PORTATIL PORTÁTIL	92
7.6	Barrido sobre el registro CFDEN, prueba 1	95
7.7	Barrido sobre el registro CFDEN, prueba 2	95
7.8	Cálculo del error del medidor por el patrón de calibración WECO PORTATIL	96
7.9	Comparación del medidor con el patrón WECO PORTÁTIL prueba 1 y 2	99
7.10	Comparación del medidor con el patrón WECO PORTÁTIL prueba 3 y 4	99
7.11	Comparación del medidor con el patrón WECO PORTÁTIL prueba 5 y 6	100
7.12	Comparación del medidor con el patrón WECO PORTÁTIL prueba 7 y 8	100

LISTADO DE FIGURAS

TABLA	TÍTULO	PAG
2.1	Diagrama fasorial para el voltaje y la corriente. El voltaje es el fasor referencia	10
2.2	Triangulo de potencia	12
2.3	Cuadrantes de la potencia reactiva y aparente	13
2.4	Sistemas trifásicos (1) Delta. (2) Estrella	15
2.5	Diagrama fasorial de un sistema trifásico en el que cada fase tiene diferente ángulo de fase. El factor de potencia en este caso no puede ser definido como el coseno del ángulo	16
2.6	Medidor de un elemento para conexión estrella 3 hilos	22
2.7	Diagrama fasorial de un medidor de 1 elemento para conexión estrella 3 hilos	22
2.8	Medidor de dos elemento para conexión estrella 3 hilos	23
2.9	Diagrama fasorial de un medidor de 2 elementos para conexión estrella 3 hilos	24
2.10	Medidor de tres elemento para conexión estrella 3 hilos	24
2.11	Diagrama fasorial de un medidor de 3 elementos para conexión estrella 3 hilos	25
2.12	Medidor de dos elemento para conexión delta 3 hilos	26
2.13	Diagrama fasorial de un medidor de 2 elementos para conexión delta 3 hilos	26
2.14	Medidor de tres elemento para conexión delta 3 hilos	27
2.15	Diagrama fasorial de un medidor de 3 elementos para conexión delta 3 hilos	27
2.16	Medidor de dos y medio elementos para conexión estrella 4 hilos	28
2.17	Diagrama fasorial de un medidor de 2 ½ elementos para conexión	29

	estrella 4 hilos	
2.18	Medidor de tres elementos para conexión estrella 4 hilos	30
2.19	Diagrama fasorial de un medidor de 3 elementos para conexión estrella 4 hilos	30
2.20	Medidor de 2 ½ elementos para conexión delta 4 hilos	31
2.21	Diagrama fasorial de un medidor de 2 ½ elementos para conexión delta 4 hilos	32
2.22	Medidor de 3 elementos para conexión delta 4 hilos	33
2.23	Medidor de 3 elementos para conexión delta 4 hilos	33
3.1	Diagrama de pines del IC ADE7754	37
3.2	Diagrama básico de conexión del ADE7754	39
3.3	Diagrama de pines del microcontrolador PIC16F877	42
3.4	Diagrama de pines de la memoria RAM no volátil DS1245Y	48
4.1	Diagrama de bloques del medidor trifásico multifunción	51
4.2	Etapa de acondicionamiento de señal de canales de voltaje y corriente	53
4.3	Tarjeta del IC ADE7754	55
4.4	Diagrama de implementación de una memoria con interfase SPI	57
4.5	Tarjeta de Temporización donde se incluyen también el microcontrolador maestro y el convertidor TTL-RS232	58
5.1	Diagrama de bloques del sistema	59
5.2	Diagrama de tiempos para la configuración de la comunicación SPI	60
5.3	Direccionamiento de los registros del ADE7754 vía la interfaz serial	63
5.4	Lectura de datos del ADE7754 vía la interfaz serial	63
5.5	Escritura de datos del ADE7754 vía la interfaz serial	64
5.6	Diagrama de tiempos de la interrupción	64
5.7	Diagrama de tiempos del módulo SPI del Reloj	65

6.1	El lenguaje de programación gráfico LabVIEW	68
6.2	Flujograma del programa principal	69
6.3	Flujograma de subrutina de lectura de valores actuales rms y de potencia	70
6.4	Flujograma de subrutina de lectura de valores rms y de potencia de archivo de una fecha específica	71
6.5	Flujograma de subrutina de lectura de valores rms y de potencia de archivo entre dos fechas	72
6.6	Flujograma de subrutina de descarga de valores rms y de potencia	73
6.7	Flujograma de subrutina de generación de reportes HTML	74
6.8	Flujograma de subrutina de calibración	75
6.9	VI Principal.	77
6.10	Lectura actual del medidor de valores rms y de potencia.	78
6.11	Lectura de archivo de valores rms y de potencia de una fecha específica.	79
6.12	Lectura de archivo de valores rms y de potencia entre dos fechas	79
6.13	Descarga del medidor de valores rms y de tiempo.	80
6.14	Generación de archivos de reporte.	80
6.15	VI de calibración.	81

INTRODUCCION

Hoy en día, los medidores de estado sólido son sinónimo de precisión, exactitud y automatización en el campo de las mediciones eléctricas; debido a que están diseñados con componentes electrónicos que no son susceptibles a desgastes mecánicos, lo que permite mayor autonomía, datos más confiables y capacidad de comunicarse con una computadora.

Estos sistemas electrónicos de medición permiten obtener los mismos parámetros eléctricos que los medidores electromecánicos, y poseen nuevas características como memoria de almacenamiento, ajustes de calibración, así como más posibilidades de configuración.

Estos dispositivos electrónicos son capaces de medir la potencia reactiva que no se convierte en trabajo útil permitiendo a la compañía distribuidora facturar dichas pérdidas; y al usuario corregir el consumo de potencia reactiva para evitar penalizaciones.

El sistema de medición propuesto posee 6 canales analógicos, 3 de corriente y 3 de voltaje. El sistema está capacitado para operar en un sistema trifásico a 240 V línea a línea, y 5 A rms, para el canal de voltaje y el canal de corriente respectivamente; el medidor es capaz de realizar mediciones para diferentes tipos de servicio; mediante la configuración del hardware del medidor.

Este sistema está conformado por un IC ADE7754 que es un medidor de alta exactitud de energía trifásico, que proporciona información adicional acerca de los parámetros eléctricos como voltaje y corriente rms por fase, con una interfase serial síncrona SPI.

El sistema incluye un microcontrolador capaz de comunicarse con el ADE7754, mediante su interfase serial síncrona SPI; este microcontrolador PIC16F877 se encarga además de realizar el protocolo de comunicación con la computadora, esto se realiza mediante la interfase serial asíncrona SCI; ambas interfases SPI y SCI del microcontrolador operan a niveles TTL. El medidor posee un reloj de tiempo real, el cual es usado para proporcionar la fecha y la hora para las mediciones periódicas, que a su vez se almacenan en una memoria RAM estática, que forma parte del medidor.

El medidor viene acompañado de una interfase gráfica del usuario desarrollada en el lenguaje de programación de alto nivel LabVIEW, que permite realizar el procedimiento de calibración, así como obtener mediciones actuales y descargas de la memoria del medidor, entre otras funciones.

Se incluye el procedimiento de calibración que se siguió para ajustar el medidor conforme a un patrón de calibración; y a partir de los resultados obtenidos, determinar si el medidor cumple con un estándar de medición.

1. GENERALIDADES

1.1 DESCRIPCIÓN DEL TEMA

En la actualidad muchos procesos como la facturación y la generación de energía eléctrica requieren el monitoreo de la calidad de la energía, para lo cual es necesario el muestreo periódico de ciertos parámetros eléctricos como voltaje, corriente y potencia.

Hoy en día muchas de las compañías están interesadas en el ahorro de energía para lo cual es necesario que utilicen medidores de potencia para comparar el consumo de energía facturado por la distribuidora, y el obtenido por el medidor.

Para satisfacer esta necesidad se plantea este proyecto, que consiste en el diseño e implementación de un sistema de medición trifásico de parámetros eléctricos como voltaje, corriente y potencia.

El sistema de medición propuesto estará conformado por un ADE7754 que es un IC medidor de energía eléctrica trifásico de alta exactitud con una interfase serial, el cual será interrogado por un microcontrolador a través de una comunicación serial.

Este microcontrolador servirá para el almacenamiento de datos, y la comunicación con una computadora implementando un protocolo estándar para la descarga de los datos.

Se desarrollará una aplicación en un lenguaje de programación multiplataforma de alto nivel para el procesamiento de datos y la elaboración de graficas, historial

de lecturas anteriores para fines de facturación, cálculos de factor de potencia y THD.

El dispositivo podrá ser utilizado en los laboratorios de la EIE con fines académicos en las materias de instrumentación electrónica y mediciones eléctricas.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Diseñar e implementar un medidor de potencia trifásico de bajo costo utilizando como base el circuito integrado ADE7754; e implementando un protocolo estándar para la comunicación de datos con una computadora.

1.2.2 OBJETIVOS ESPECÍFICOS

Implementar un medidor trifásico de voltaje, corriente y potencia; con el IC ADE7754 con su respectivo modulo de acondicionamiento de señal.

Interrogar el ADE7754 para extraer los datos almacenados y enviarlos a una base de datos, por medio de un protocolo de comunicación estándar.

Realizar una aplicación en un lenguaje de programación multiplataforma de alto nivel para el procesamiento de datos y la presentación de resultados como gráficos e historial de lecturas.

1.3 JUSTIFICACIÓN

Muchas de las compañías en el país tienen altos niveles de consumo energético y cada vez más se interesan por sistemas o dispositivos que realicen mediciones de forma confiable y además que su implementación sea de bajo costo; además es interesante realizar mediciones para comparar el consumo de energía facturado por la distribuidora, y el obtenido por el medidor propio.

En la actualidad existen muchos medidores y compañías que proporcionan este servicio cuyo costo es elevado; es por eso que se pretende la implementación de un sistema de medición trifásico de energía confiable y de bajo costo.

1.4 ALCANCES

Al final de la implementación de este sistema de medición se espera contar con un módulo que realice las mediciones de voltaje, corriente y potencia y energía través del IC ADE7754 con su respectiva etapa de acondicionamiento de señal para niveles de 240 Voltios y 5 Amperios en el secundario; y que sea capaz de almacenar las mediciones obtenidas, y a la vez implemente un protocolo estándar para la comunicación de datos con una computadora.

El procesamiento y la presentación de datos se realizarán por medio de un lenguaje de programación multiplataforma de alto nivel; esta aplicación contara con graficas de voltaje, corriente, potencia y energía, así como un historial de lecturas, cálculo y presentación de factor de potencia y THD.

1.5 LIMITACIONES

Debido a la tolerancia de los dispositivos electrónicos que se utilizarán para la implementación de este sistema, se puede tener un margen de error mínimo en la obtención de las mediciones, pero dentro del estándar requerido en el mercado nacional.

2. TEORIA BÁSICA

2.1 CONCEPTOS BÁSICOS

Medición

Se entiende por medición la determinación del valor de una magnitud por comparación (directa o indirecta) con un patrón del sistema de unidades empleado. Toda medición requiere de un principio de medición, el cual es el fundamento científico del método de medición, siendo este último el conjunto de operaciones teóricas y prácticas, en términos generales, involucradas en la realización de mediciones de acuerdo con el principio establecido.

Metrología

La metrología es la ciencia de la medición. Todo lo que tiene que ver con la medición, se encuentra en diseño, conducción o análisis de resultados de una prueba, existen en la metrología.

Calibración

La calibración es la comparación de un aparato de medición contra un estándar igual o mejor. Un estándar de medición es considerado como la referencia; este es el [instrumento] tomado como el más correcto en la comparación de los dos. Uno calibra para averiguar cuán lejana esta la incógnita del patrón.

Energía

En electricidad, el trabajo o la energía eléctrica que se utiliza para realizar un trabajo se cuantifica a partir de la cantidad de potencia activa empleada multiplicada por el tiempo en que se aplica.

Teorema de Blondel

Establece que si una red es alimentada a través de N conductores, la potencia total es medida por medio de sumar las lecturas de N conductores, la potencia total es medida por medio de sumar las lecturas de N vatímetros conectados de tal modo, que el elemento de corriente de cada vatímetro esté conectado en cada línea; y el correspondiente elemento de tensión esté conectado entre esa línea y un punto común. Si el punto común está localizado en una de las líneas, entonces la potencia puede ser medida por medio de $N-1$ vatímetros, sin restricciones de números de fases o conductores o balance de carga entre las fases.

2.2 INSTRUMENTOS DE MEDICIÓN.

Los instrumentos y medidores modernos cuentan con capacidad de registro electrónico, o con memoria que puede ser leída por otras computadoras. Estos instrumentos permiten almacenar más información y por ello calculo y análisis adicionales algunos de ellos son los siguientes.

Amperímetros.

Son aparatos que se utilizan para medir la intensidad de la corriente eléctrica, que circula por las líneas, bancos de transformadores, alimentadores, etc. Si la corriente es menor de 5 amperios, es conectado un amperímetro directamente en el circuito para ser medida. Si la corriente es mayor, el amperímetro es conectado a un transformador de intensidad.

En conexiones trifásicas debe de conectarse un amperímetro por cada fase. En caso de existir la seguridad de que las cargas son balanceadas, se puede usar un solo aparato en cualquiera de las fases. Cuando las instalaciones son grandes, se

acostumbra utilizar un solo amperímetro por circuito trifásico, efectuándose las lecturas de cada fase a través de un conmutador de amperímetro de tres vías.

Voltímetros.

Son aparatos que se utilizan para medir la tensión o diferencia de potencial en voltios, de los diferentes circuitos de una instalación, terminales o conductores. El voltímetro es conectado directamente entre los puntos que se necesita medir la diferencia de potencial.

En los circuitos trifásicos se acostumbra usar un solo voltímetro, que por medio un conmutador de tres vías permite leer las tensiones entre cada par de fases de la instalación.

Vatímetros.

Un vatímetro mide la magnitud de la potencia eléctrica que es suministrada a la carga. La aplicación correcta de este instrumento requiere polaridad y faseo correcto de corriente y voltaje. Los factores de escala para los vatímetros típicamente e indican kilovatios o megavatios. Esta medición se efectúa por medio de dos bobinas, una de intensidad conectada en serie y otra de tensión conectada en paralelo.

Vármetros.

Son aparatos semejantes a los vatímetros, con la diferencia de que miden la potencia reactiva de una instalación, la cual se expresa en volt-amperios reactivos (VAR). Los Vármetros tienen un punto cero en el centro de la escala, porque la potencia reactiva puede ser en adelanto o en atraso.

2.3 MEDIDORES DE FACTURACIÓN.

Son los dispositivos que distinguen registran la integral de una cantidad respecto al tiempo entre estos podemos mencionar a los Vatihorímetro y los Varhorímetros.

Vatihorímetro.

Es un instrumento eléctrico que mide y registra la integral con respecto al tiempo, de la potencia activa del circuito en que se conecta. Esta integral de potencia es la energía consumida por el circuito durante al intervalo de tiempo en que se realiza. La energía eléctrica es especificada en vatios-hora debido a que es de uso común en la industria, aunque en el sistema internacional de unidades la unidad de energía eléctrica es el Joule (J).

$$1 \text{ vatio-hora} = 3\,600 \text{ J}$$

Varhorímetro.

Son aparatos que integran la energía de reactiva que circula de por una instalación eléctrica. Son análogos a los vatihorímetros.

Los mecanismos internos de los Varhorímetros son idénticos a los vatihorímetros. Sin embargo, la tensión aplicada a estos medidores es desfasada 90 grados eléctricos. Un vatihorímetro estándar y un transformador desfasador pueden ser conectados para funcionar como un varhorímetro.

2.4 SISTEMAS MONOFÁSICOS

2.4.1 CÁLCULO DE POTENCIA EN SISTEMAS MONOFÁSICOS

Para los circuitos monofásicos, la potencia suministrada para realizar trabajo es fácilmente calculada. Dado un voltaje senoidal de magnitud rms V (rms: valor medio cuadrático) y una corriente senoidal de magnitud rms I , desplazada por un ángulo θ , en un tiempo t :

$$\text{Voltaje instantáneo} \quad v = \sqrt{2}V\text{sen}(2\pi ft)$$

$$\text{Corriente instantánea} \quad i = \sqrt{2}I\text{sen}(2\pi ft - \theta)$$

Donde θ puede tener un valor positivo o negativo.

$$\text{Potencia instantánea} \quad p = vi$$

$$p = 2VI\text{sen}(2\pi ft)\text{sen}(2\pi ft - \theta)$$

$$p = VI \cos(\theta) - \cos(4\pi ft - \theta)$$

La potencia promedio sobre un número entero de ciclos:

$$p = VI \cos(\theta) \quad \text{Ecuación 2.1}$$

Por lo tanto el factor de potencia es:

$$FP = \frac{p}{U} = \frac{VI \cos(\theta)}{VI} = \cos(\theta) \quad \text{Ecuación 2.2}$$

Las ecuaciones 2.1 y 2.2 son las ecuaciones fundamentales que definen la potencia para los sistemas en los cuales la corriente y el voltaje son senoidales.

2.4.2 FACTOR DE POTENCIA

Expresar la corriente en componentes ortogonales en un diagrama fasorial ilustra como la potencia suministrada puede variar desde un valor máximo a cero, dependiendo del ángulo de fase entre las ondas senoidales de voltaje y de corriente.

La figura 2.1 muestra el vector de voltaje junto con la corriente descompuesta en sus componentes ortogonales.

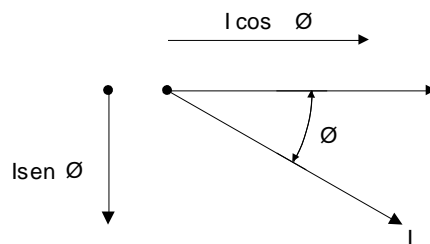


Figura 2.1 Diagrama fasorial para el voltaje y la corriente. El voltaje es el fasor referencia.

La corriente I en un ángulo θ relativo al voltaje puede ser descompuesta en dos vectores: $I \cos(\theta)$ y $I \sin(\theta)$. La componente en fase $I \cos(\theta)$ multiplicada por el voltaje proporciona la potencia promedio (activa) en vatios. La componente de la corriente que está 90° desfasada del voltaje, $I \sin(\theta)$, no está asociada con la potencia suministrada (activa) y no contribuye al trabajo.

Puesto que esta corriente podría estar asociada con campos magnéticos, algunas veces es llamada corriente magnetizante debido a que, a pesar de que no hace trabajo, esta corriente, interactúa a través de la reactancia inductiva de los devanados de los motores de CA, para proveer el campo magnético necesario y que tales motores funcionen.

Han sido definidos tres tipos de potencia para los sistemas en los que el voltaje y la corriente son senoidales. A través de los años, se les ha dado diferentes nombres a estos tipos de potencia. Sin embargo, serán enfatizados, los nombres que se usan en la actualidad.

2.4.3 POTENCIA ACTIVA.

La potencia activa esta dada por el símbolo **P** y es definida por la ecuación 2.1:

$$P = VI \cos(\theta)$$

Otros nombres para la potencia activa incluyen: (1) potencia real; (2) potencia suministrada. La potencia activa es la potencia que hace trabajo. Note que mientras todas las magnitudes de potencia son el producto de voltio-amperio, solo la potencia activa es expresada en vatios.

2.4.4 POTENCIA REACTIVA.

La potencia reactiva esta dada por el símbolo **Q** y es definida por la ecuación 2.3:

$$Q = VI \sin(\theta) \quad \text{Ecuación 2.3}$$

Otros nombres para la potencia reactiva incluyen: (1) potencia imaginaria; (2) potencia magnetizante. La potencia reactiva es expresada en voltamperios reactivos o vars. Si la carga es predominantemente inductiva, la corriente esta en atraso respecto al voltaje y la potencia reactiva es de signo positivo. Si la carga es predominantemente capacitiva, la corriente esta en adelante respecto al voltaje y la potencia reactiva es de signo negativo.

El fasor de potencia esta dado por el símbolo **S** y es definido por la ecuación 2.4:

$$S = \sqrt{P^2 + Q^2} \quad \text{Ecuación 2.4}$$

La figura 2.2 muestra un diagrama fasorial, frecuentemente llamado triangulo de potencia, el cual muestra las relaciones entre los tres tipos de potencia mencionados anteriormente. La potencia reactiva es ortogonal ala potencia activa, y es positiva para las corrientes de atraso. Es claro que la definición del fasor de potencia, es derivada geoméricamente de la potencia activa y reactiva.

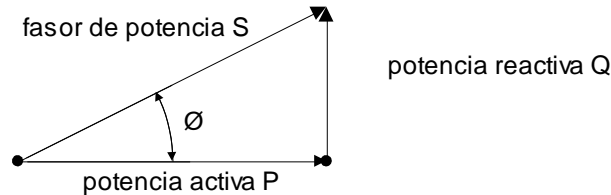


Figura 2.2 Triangulo de potencia.

El factor de potencia esta dado por el símbolo FP y para magnitudes senoidales es definido por la ecuación 2.5:

$$FP = \frac{P}{S} = \cos \theta \quad \text{Ecuación 2.5}$$

Puesto que el factor de potencia puede ser expresado en referencia al ángulo de desplazamiento entre el voltaje y la corriente, el factor de potencia así definido debe de ser llamado factor de potencia de desplazamiento, el símbolo frecuentemente es escrito como FP. Los valores para el factor de potencia de desplazamiento son desde cero hasta uno a medida que el ángulo de desplazamiento varia desde cero (corriente y voltaje en fase) a 90°.

Un bajo factor de potencia es el resultado del consumo de potencia reactiva. La transferencia de potencia reactiva sobre una red de distribución causa pérdidas de energía. Para forzar a los consumidores a corregir este factor de potencia se utilizan monitores de potencia reactiva para la detección y así penalizar al usuario por un pobre factor de potencia.

2.4.5 CUADRANTES PARA LA POTENCIA REACTIVA Y APARENTE

La potencia reactiva y la potencia aparente se ubican mediante los cuadrantes, en donde el eje vertical es la potencia reactiva y en el eje horizontal es la potencia activa; en cada cuadrante se forma el triángulo de potencia. Por lo cual si tenemos que la potencia activa y la potencia reactiva son positivas, luego la potencia reactiva esta ubicada en el primer cuadrante, y por lo tanto la potencia aparente esta ubicada en este cuadrante. En la Fig. 2.3 se puede observar a que cuadrante corresponde cada condición de potencia activa y reactiva.

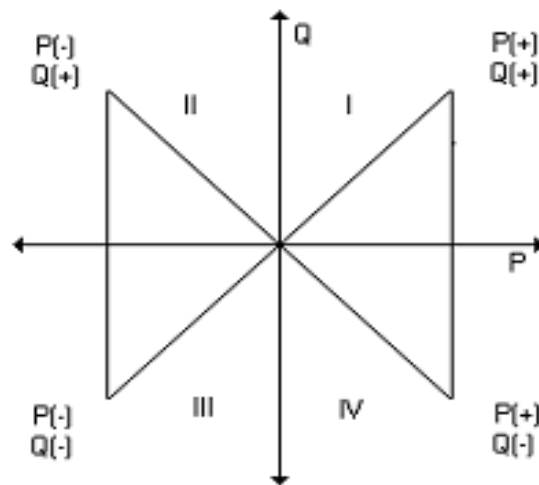


Fig. 2.3 Cuadrantes de la potencia reactiva y aparente.

En donde: P(+) Potencia activa consumida o entregada
P(-) Potencia activa generada o recibida

Q(+) Potencia reactiva consumida o entregada

Q(-) Potencia reactiva generada o recibida

Es de hacer notar que tanto la potencia reactiva como la aparente se ubican por el cuadrante que ocupan, y a esto se añade si la potencia activa es entregada o recibida.

2.5 SISTEMAS TRIFÁSICOS

Un sistema trifásico desarrolla un alto nivel de potencia para la industria y aplicaciones comerciales; las tres fases corresponden a tres líneas de potencial desfasadas 120 grados eléctricos entre ellas. Las conexiones típicas son la conexión en estrella y la conexión delta.

En un sistema trifásico, los niveles de voltaje entre las fases y el neutro son uniformes y definidas por:

$$V_{an} = V_{bn} = V_{cn} = \frac{V_{ab}}{\sqrt{3}} = \frac{V_{bc}}{\sqrt{3}} = \frac{V_{ac}}{\sqrt{3}}$$

Los voltajes entre las fases varían dependiendo de los factores de la carga y de la calidad de la distribución de los transformadores. Los sistemas trifásicos son distribuidos en diferentes niveles de voltaje: 240V, 480V, 2400V, 4160V, 6900V, 13800V, etc.

Configuraciones típicas de los sistemas trifásicos se muestran en la figura 2.4.

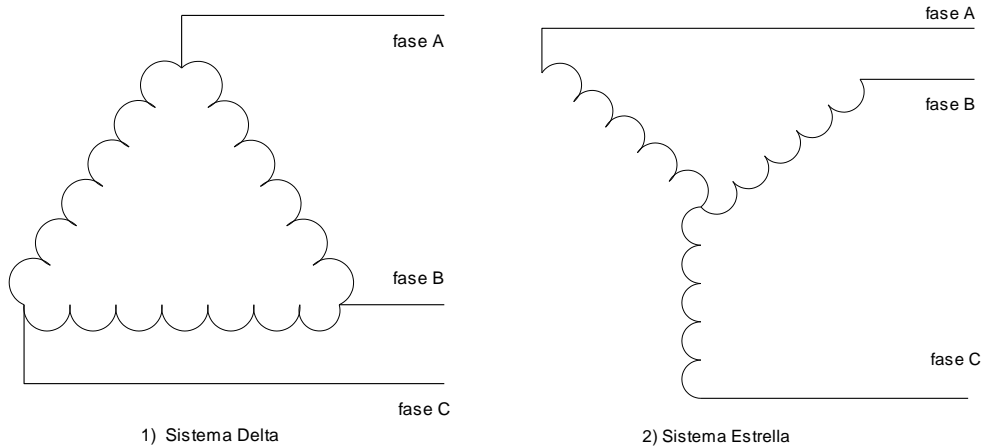


Figura 2.4 Sistemas trifásicos (1) Delta. (2) Estrella.

2.5.1 CÁLCULO DE POTENCIA EN SISTEMAS TRIFÁSICOS.

Los conceptos de potencia desarrollados para los circuitos monofásicos con voltajes y corrientes senoidales pueden ser extendidos a los circuitos polifásicos. Tales circuitos pueden dividirse en juegos bifilares, con el conductor neutro asociado con cada uno de los otros conductores.

Las ecuaciones 2.1 y 2.3 pueden ser rescritas para definir los términos de potencia equivalente a los términos monofásicos. En estas ecuaciones, k representa el número de fase, m es el número total de fases, α y β son, respectivamente, los ángulos de voltaje y de corriente con respecto a un marco de referencia común.

$$P = \sum_{k=1}^m V_k I_k \cos(\alpha - \beta) \quad \text{Ecuación 2.6}$$

$$Q = \sum_{k=1}^m V_k I_k \text{sen}(\alpha - \beta) \quad \text{Ecuación 2.7}$$

Para un servicio de potencia de distribución senoidal de tres fases, con fases A, B y C las ecuaciones para las potencias activa, reactiva y aparente son:

$$P = V_a I_a \cos(\alpha_a - \beta_a) + V_b I_b \cos(\alpha_b - \beta_b) + V_c I_c \cos(\alpha_c + \beta_c)$$

$$Q = V_a I_a \text{sen}(\alpha_a - \beta_a) + V_b I_b \text{sen}(\alpha_b - \beta_b) + V_c I_c \text{sen}(\alpha_c + \beta_c)$$

$$S = \sqrt{P^2 + Q^2}$$

2.5.2 EL FACTOR DE POTENCIA EN SISTEMAS TRIFÁSICOS.

El factor de potencia esta definido por la ecuación 2.8 note que no siempre es verdadero decir que el factor de potencia es igual al coseno del ángulo de fase.

En muchos sistemas trifásicos balanceados, los ángulos de fase de todas las fases son iguales y se mantiene la relación del coseno. En sistemas desbalanceados, como el que se muestra en la figura 2.5, cada fase tiene un ángulo de fase diferente, las fases de los voltajes y corrientes no son iguales, y la relación del coseno no es aplicable.

$$P = \frac{\text{Potencia activa total}}{\text{factor de potencia}} = \frac{P_{\text{Equitotal}}}{S} \text{ frecuentemente } \neq \cos \theta \quad \text{Ecuación 2.8}$$

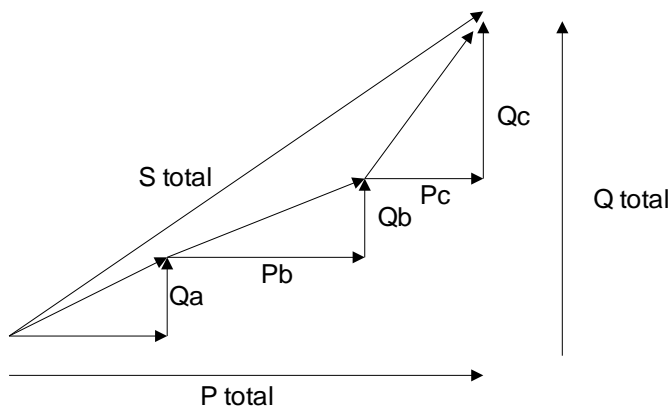


Figura 2.5 Diagrama fasorial de un sistema trifásico en el que cada fase tiene diferente ángulo de fase. El factor de potencia en este caso no puede ser definido como el coseno del ángulo.

2.5.3 COMPONENTES ARMÓNICAS Y TASA DE DISTORSIÓN ARMÓNICA (THD).

Las componentes armónicas son magnitudes senoidales en las que puede descomponerse una magnitud periódica, de las cuales la 1º o fundamental, tiene la misma frecuencia de la magnitud considerada, y las siguientes tienen frecuencias múltiplos de ella. Las armónicas son definidas como múltiplos enteros continuos de la forma de onda fundamental. Cada onda de armónica es definida por su número de armónica, su amplitud y su relación de fase con la fundamental.

Calcular la potencia de suministrada para hacer trabajo en una carga no lineal es mucho mas complicado que para una corriente senoidal, debido a los armónicos que contiene la corriente de dicha carga, si h es igual al número de armónica, y V_h e I_h son amplitudes máximas del voltaje armónico y corriente armónica respectivamente, el voltaje y la corriente pueden ser representados por series de Fourier de la siguiente forma:

$$v(t) = \sum_{h=1}^{\infty} V_h \text{sen}(2\pi f_h t + \alpha_h)$$
$$i(t) = \sum_{h=1}^{\infty} I_h \text{sen}(2\pi f_h t + \beta_h)$$

Las expresiones generales de voltaje y corriente para cualquier armónica serán:

$$v_{h(t)} = V_h \text{sen}(2\pi f_h t + \alpha_h)$$
$$i_{h(t)} = I_h \text{sen}(2\pi f_h t + \beta_h)$$

Los valores rms para la corriente y el voltaje, considerando el efecto de todas las armónicas, son las siguientes:

$$V_{rms} = \sqrt{\sum_{h=1}^{\infty} \frac{V_h^2}{2}} = \sqrt{\sum_{h=1}^{\infty} V_{hrms}^2}$$

$$I_{rms} = \sqrt{\sum_{h=1}^{\infty} \frac{I_h^2}{2}} = \sqrt{\sum_{h=1}^{\infty} I_{hrms}^2}$$

Una medida de los niveles de armónicas, usada frecuentemente, es la tasa de distorsión armónica THD (por su significado en inglés: Total Harmonic Distortion), la cual es razón del valor rms de las armónicas (arriba de la componente fundamental) al valor rms de la componente fundamental, y es expresada en porcentaje.

$$THD_V = \frac{\sqrt{\sum_{h=2}^{\infty} V_{hrms}^2}}{V_{1rms}} \cdot 100\%$$

$$THD_I = \frac{\sqrt{\sum_{h=2}^{\infty} I_{hrms}^2}}{V_{1rms}} \cdot 100\%$$

Obviamente si no hay armónicas, es decir, la carga es lineal, entonces ambas tasas de distorsión armónicas son cero.

2.6 TRANSFORMADORES DE MEDIDA

Son dispositivos electromecánicos cuya función principal es reducir a escala, las magnitudes de tensión y corriente que se utilizan para protección y medición de los diferentes circuitos de una subestación o un sistema eléctrico en general; con el fin de disminuir el costo y los peligros que representan las altas tensiones.

Los transformadores de medida se clasifican en transformadores de tensión y transformadores de intensidad.

Los objetivos principales de los transformadores de medida son:

- Aislar o separar los circuitos y aparatos de medición y de protección; de la alta tensión y cualquier nivel de tensión.
- Evitar perturbaciones electromagnéticas de corrientes grandes, y reducir las corrientes de cortocircuitos a valores admisibles en los equipos de medición y protección.
- Obtener intensidades de corrientes o tensiones proporcionales a las que se desea medir, para transmitir las a los equipos apropiados.

2.6.1 TRANSFORMADORES DE INTENSIDAD

Son dispositivos en los que la corriente secundaria es prácticamente proporcional a la corriente primaria, dentro de las condiciones normales de operación, aunque ligeramente desfasadas.

Desarrollan dos tipos de función: transformar la corriente y aislar los instrumentos de protección y medición conectados a los circuitos de alta tensión. El primario del transformador de intensidad (TI) se conecta en serie con el circuito a controlar y el secundario se conecta en serie con los aparatos de medición y de protección.

Por su construcción los tipos de TI son:

- Tipo devanado: Consta de dos devanados, primario y secundario, totalmente aislados y montados permanentemente sobre un circuito magnético.
- Tipo barra: Es similar al tipo devanado, excepto en que el primario es un solo conductor recto de tipo barra.

- Tipo ventana: Tiene un devanado secundario totalmente aislado y montado permanentemente sobre el circuito magnético y una ventana a través de la cual puede hacerse pasar un conductor que proporciona el devanado primario.
- Tipo borne: Es un tipo especial de ventana proyectado para colocarse en los bornes aislados de los aparatos, actuando el conductor del borne como devanado primario.

Un TI puede tener uno o varios secundarios, embobinados a su vez sobre uno o varios circuitos magnéticos. Estos pueden ser fabricados para servicio interior o exterior.

Se fabrican con aislamiento de resinas sintéticas para tensiones bajas o medias, mientras que para altas tensiones se utilizan aislamientos de papel, aceite y porcelana. La tensión del aislamiento de un TI debe ser cuando menos igual a la tensión más elevada del sistema al que va a estar conectado.

El devanado secundario siempre debe estar cortocircuitado antes de desconectar la carga. Si se abre el circuito secundario con circulación de corriente por el primario, todos los amperio-vueltas primario son magnetizantes y normalmente producirán una tensión secundaria excesivamente elevada en los bornes del circuito abierto.

2.6.2 TRANSFORMADORES DE TENSIÓN

Son dispositivos en que la tensión secundaria, dentro de las condiciones normales de operación, es prácticamente proporcional a la tensión primaria, aunque ligeramente desfasada. Desarrollan dos funciones: transformar la tensión y aislar los instrumentos de protección y medición conectados a los circuitos de alta o cualquier otro nivel de tensión.

El devanado primario se conecta en paralelo con el circuito, cuyo voltaje será medido o controlado. El devanado secundario entrega un voltaje proporcional al voltaje de línea, y a este se conectan en paralelo las bobinas de tensión de los diferentes aparatos de medición o de protección.

Estos transformadores se fabrican para servicio interior o exterior, y al igual que los de intensidad, se fabrican con aislamiento de resinas sintéticas para tensiones bajas o medias, mientras que para altas tensiones se utilizan aislamientos de papel, aceite y porcelana.

2.7 MEDICIÓN DE POTENCIA TRIFÁSICA Y ANÁLISIS VECTORIAL

Un medidor de energía trifásico puede ser conectado a diferentes tipos de servicio de energía como trifásico de tres líneas o de cuatro líneas, en una carga en arreglo delta o estrella.

Para efectos de medición de potencia se proporcionan las fórmulas para cada tipo de configuración, así como el diagrama de conexión y su respectivo diagrama fasorial visto por el medidor.

2.7.1 MEDICIÓN ESTRELLA DE 3 CONDUCTORES

Según el teorema de Blondel, un circuito trifásico de tres conductores requiere de un medidor de dos elementos (2 TI y 2 TT). Este tipo de medición se pudiera efectuar con un medidor de tres elementos (3 TI y 3 TT) pero incurriría en más gastos, debido a que la exactitud en la medición es razonablemente la misma.

MEDIDOR DE UN ELEMENTO

Este método de medir potencia trifásica con un medidor de un elemento (1 TI y 1 TT) requiere que la carga sea balanceada. Esta configuración se le llama medición en cuadratura, debido a la referencia de 90° utilizada en el argumento de las funciones trigonométricas en las definiciones de potencia. Ver en Fig. 2.6 el tipo de conexión.

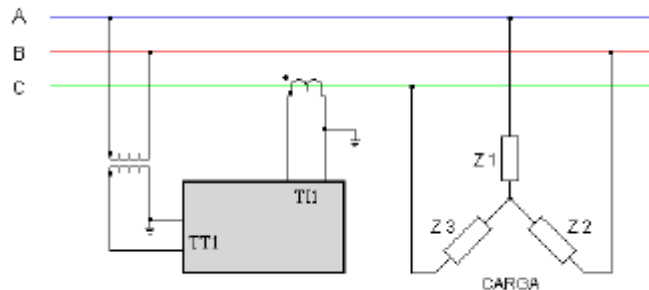


Figura 2.6 Medidor de un elemento para conexión estrella 3 hilos.

En donde:

TT1 mide V_{AB} .

TI1 mide I_C .

Para una carga resistiva, el análisis vectorial por la carga es representado por:

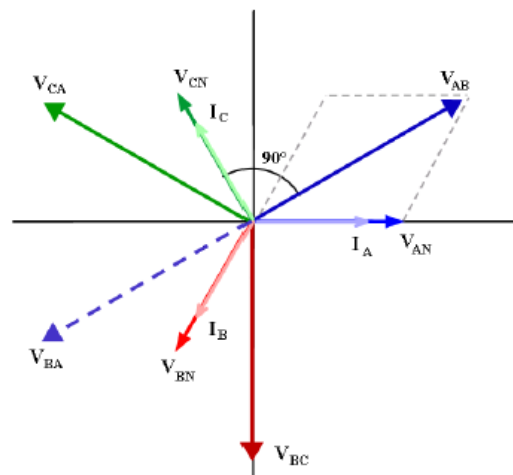


Figura 2.7 Diagrama fasorial de un medidor de 1 elemento para conexión estrella 3 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = \sqrt{3}V_{AB}I_C \cos[90^\circ - (\phi_{V_{AB}} - \phi_{I_{CN}})]$$

$$P = \sqrt{3}V_{AB}I_C \text{sen}[90^\circ - (\phi_{V_{AB}} - \phi_{I_{CN}})]$$

MEDIDOR DE DOS ELEMENTOS

Este medidor cumple el teorema de Blondel para un circuito trifásico de tres conductores, debido a que tiene dos elementos (2 TI y 2 TT). En la Fig. 2.8 se puede apreciar la conexión.

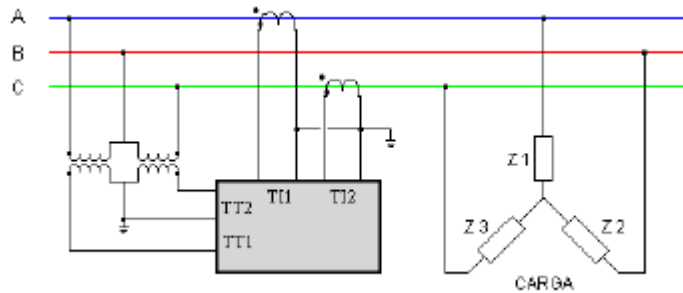


Figura 2.8 Medidor de dos elemento para conexión estrella 3 hilos

En donde:

TT1 mide V_{AB} y TT2 mide V_{CB} .

TI1 mide I_A y TI2 mide I_C .

El siguiente es el análisis vectorial visto por el medidor:

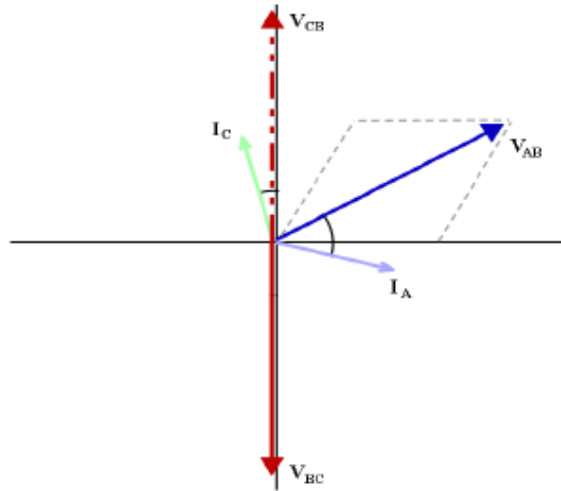


Figura 2.9 Diagrama fasorial de un medidor de 2 elementos para conexión estrella 3 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = V_{AB} I_A (\cos 30^\circ \cos \phi_1 - \text{sen} 30^\circ \text{sen} \phi_1) + V_{CB} I_C (\cos 30^\circ \cos \phi_3 + \text{sen} 30^\circ \text{sen} \phi_3)$$

$$Q = V_{AB} I_A (\text{sen} 30^\circ \cos \phi_1 + \cos 30^\circ \text{sen} \phi_1) - V_{CB} I_C (\text{sen} 30^\circ \cos \phi_3 - \cos 30^\circ \text{sen} \phi_3)$$

MEDIDOR DE TRES ELEMENTOS

Con esta configuración se logra una medición más exacta. En la Fig. 2.10 se muestra la conexión.

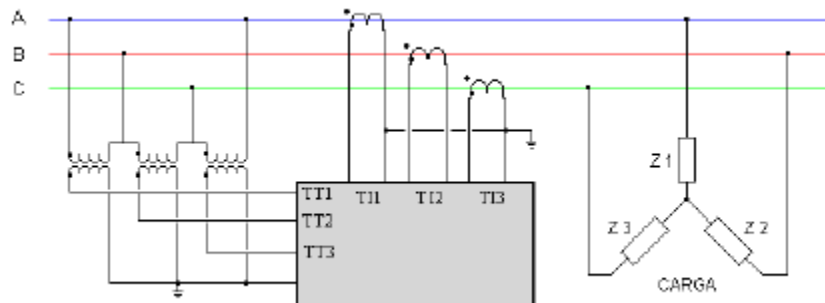


Figura 2.10 Medidor de tres elemento para conexión estrella 3 hilos

En donde:

TT1 mide V_{AB} , TT2 mide V_{BC} y TT3 mide V_{CA} .

TI1 mide I_A , TI2 mide I_B y TI3 mide I_C .

El siguiente es el análisis vectorial visto por el medidor:

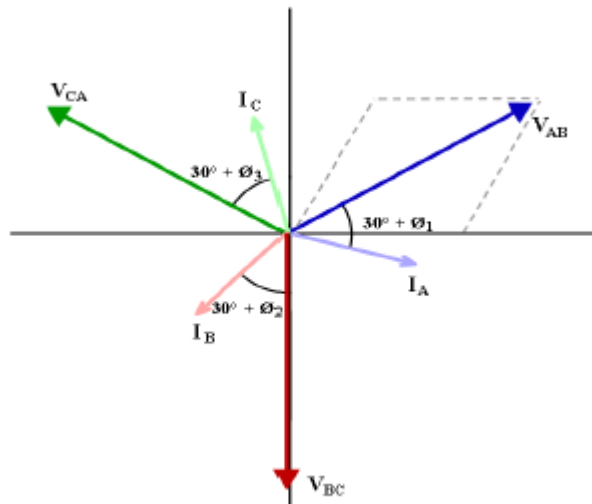


Figura 2.11 Diagrama fasorial de un medidor de 3 elementos para conexión estrella 3 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = \frac{V_{AB}}{\sqrt{3}} I_A \cos \phi_1 + \frac{V_{BC}}{\sqrt{3}} I_B \cos \phi_2 + \frac{V_{CA}}{\sqrt{3}} I_C \cos \phi_3$$

$$Q = \frac{V_{AB}}{\sqrt{3}} I_A \text{sen} \phi_1 + \frac{V_{BC}}{\sqrt{3}} I_B \text{sen} \phi_2 + \frac{V_{CA}}{\sqrt{3}} I_C \text{sen} \phi_3$$

2.7.2 MEDICIÓN DELTA DE 3 CONDUCTORES

Al igual que la conexión estrella trifásica de tres hilos, se necesita un medidor de dos elementos, según el teorema de Blondel. Esta medición se puede realizar así mismo con un medidor de tres elementos.

MEDIDOR DE 2 ELEMENTOS

El diagrama de conexión se muestra a continuación, en la Fig.2.12:

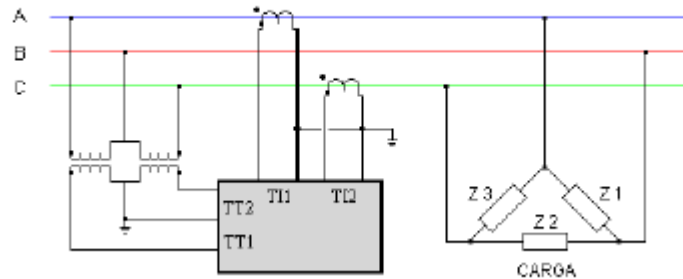


Figura 2.12 Medidor de dos elemento para conexión delta 3 hilos

En donde:

TT1 mide V_{AB} y TT2 mide V_{CB} .

TI1 mide I_A y TI2 mide I_C .

El análisis vectorial visto por el medidor es el siguiente:

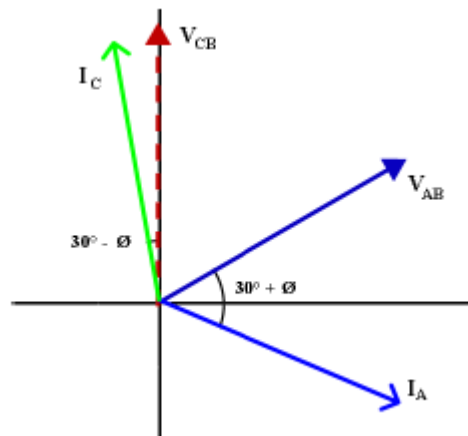


Figura 2.13 Diagrama fasorial de un medidor de 2 elementos para conexión delta 3 hilos.

Las potencias trifásicas, activa y reactiva, se calculan con las siguientes ecuaciones:

$$P = V_{AB} I_A (\cos 30^\circ + \phi_1) + V_{CB} I_C (\cos 30^\circ - \phi_3)$$

$$Q = V_{AB} I_A (\text{sen} 30^\circ + \phi_1) - V_{CB} I_C (\text{sen} 30^\circ - \phi_3)$$

MEDIDOR DE 3 ELEMENTOS

Así como en la carga conectada en estrella, con el medidor de tres elementos se consigue una medición más exacta. El esquema de conexión es el mostrado en la Fig. 2.14.

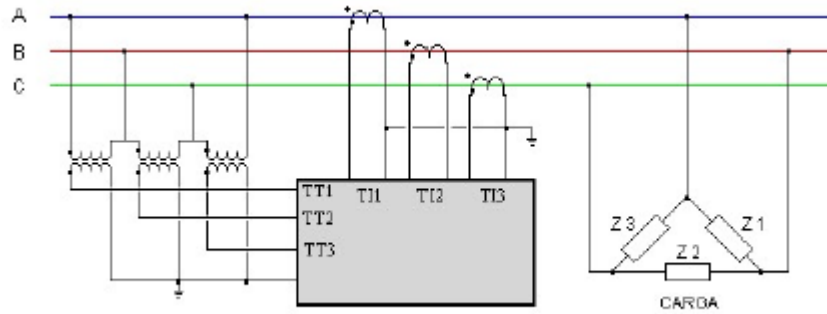


Figura 2.14 Medidor de tres elemento para conexión delta 3 hilos

En donde:

TT1 mide V_{AB} , TT2 mide V_{BC} y TT3 mide V_{CA} .

TI1 mide I_A , TI2 mide I_B y TI3 mide I_C .

El siguiente es el análisis vectorial visto por el medidor:

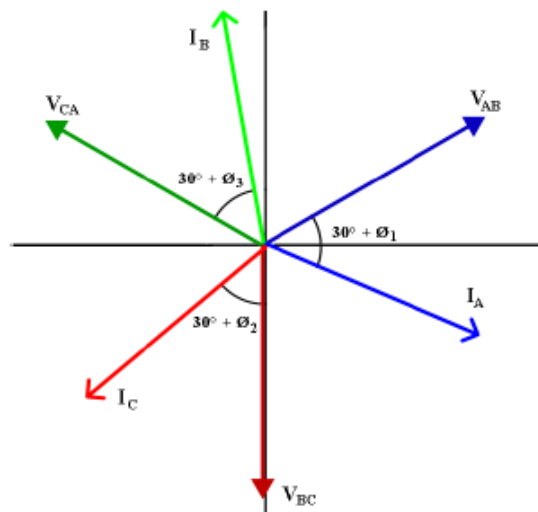


Figura 2.15 Diagrama fasorial de un medidor de 3 elementos para conexión delta 3 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = \frac{V_{AB}}{\sqrt{3}} I_A \cos \phi_1 + \frac{V_{BC}}{\sqrt{3}} I_B \cos \phi_2 + \frac{V_{CA}}{\sqrt{3}} I_C \cos \phi_3$$

$$Q = \frac{V_{AB}}{\sqrt{3}} I_A \text{sen} \phi_1 + \frac{V_{BC}}{\sqrt{3}} I_B \text{sen} \phi_2 + \frac{V_{CA}}{\sqrt{3}} I_C \text{sen} \phi_3$$

2.7.3 MEDICIÓN ESTRELLA DE 4 CONDUCTORES

Según el teorema de Blondel se requiere de un medidor de tres elementos para efectuar la medición de potencia trifásica para una conexión estrella de 4 conductores. Si los voltajes son balanceados, se puede realizar la conexión "Z", que consiste en un medidor de dos y medio elementos (3 TI y 2 TT), con igual exactitud que la anterior.

EL MEDIDOR DE 2 1/2 ELEMENTOS

Este medidor esta comprendido de dos sensores de voltaje y tres sensores de corriente. El punto común de los sensores de voltaje deberá ser conectado al conductor neutro. Si los voltajes entre cada línea y el neutro son balanceados entre los límites aceptables, la precisión es generalmente considerada satisfactoria. Ver el diagrama de conexión en la Fig. 2.16.

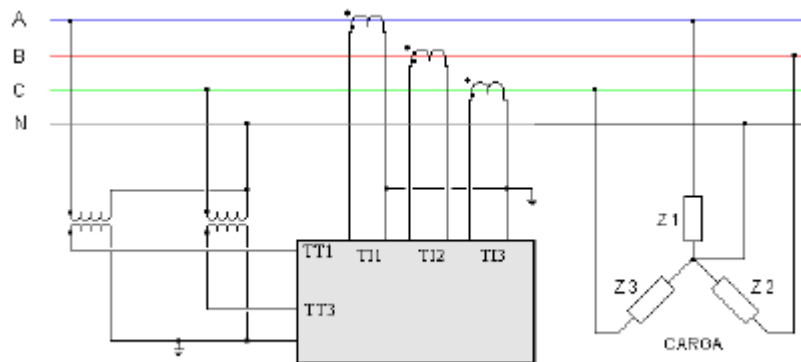


Figura 2.16 Medidor de dos y medio elementos para conexión estrella 4 hilos

En donde:

TT1 mide V_{AN} y TT3 mide V_{CN} .

TI1 mide I_A , TI2 mide I_B y TI3 mide I_C .

El diagrama vectorial es el siguiente:

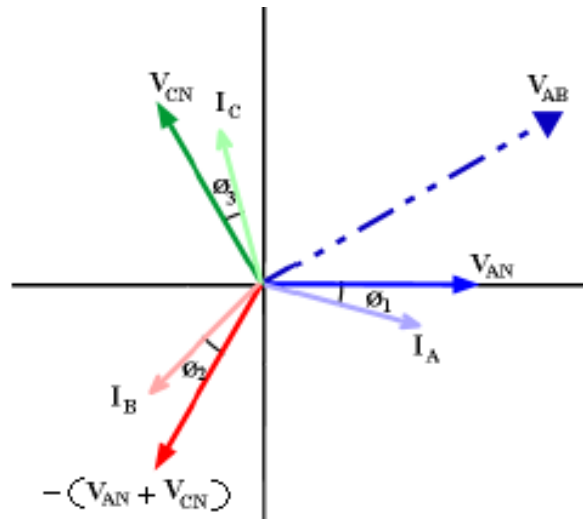


Figura 2.17 Diagrama fasorial de un medidor de 2 1/2 elementos para conexión estrella 4 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = V_{AN} I_A \cos \phi_1 - (V_{AN} + V_{CN}) I_B \cos \phi_2 + V_{CN} I_C \cos \phi_3$$

$$Q = V_{AN} I_A \sin \phi_1 - (V_{AN} + V_{CN}) I_B \sin \phi_2 + V_{CN} I_C \sin \phi_3$$

EL MEDIDOR DE TRES ELEMENTOS

Este medidor trifásico está comprendido de tres sensores de voltaje y tres sensores de corriente. El punto común de los sensores de voltaje deberá ser conectado al conductor neutro. Este tipo de conexión cumple conexión exacta bajo condiciones de carga balanceada o desbalanceada. Ver conexión en la Fig. 2.18.

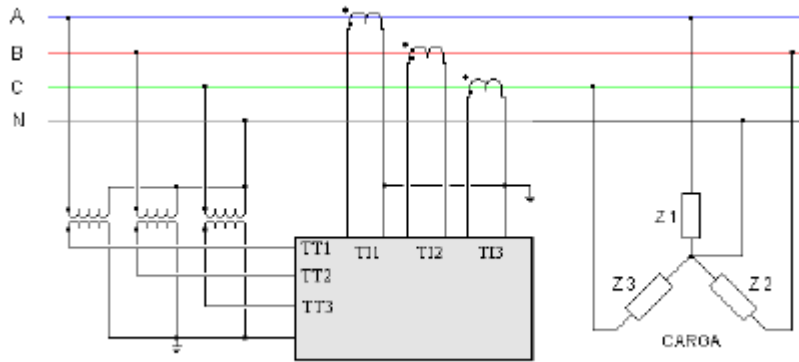


Figura 2.18 Medidor de tres elementos para conexión estrella 4 hilos

En donde:

TT1 mide V_{AN} , TT2 mide V_{BN} y TT3 mide V_{CN} .

TI1 mide I_A , TI2 mide I_B y TI3 mide I_C .

El siguiente es el análisis vectorial visto por el medidor:

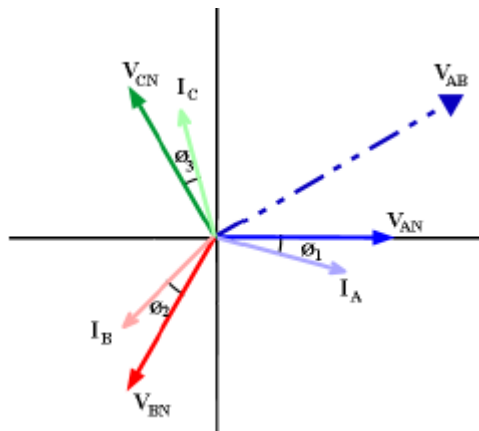


Figura 2.19 Diagrama fasorial de un medidor de 3 elementos para conexión estrella 4 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = V_{AN} I_A \cos \phi_1 + V_{BN} I_B \cos \phi_2 + V_{CN} I_C \cos \phi_3$$

$$Q = V_{AN} I_A \sin \phi_1 + V_{BN} I_B \sin \phi_2 + V_{CN} I_C \sin \phi_3$$

2.7.4 MEDICIÓN DELTA DE 4 CONDUCTORES

La potencia eléctrica en un circuito trifásico conectado en delta cerrada o abierta de cuatro conductores, con el neutro formado por un punto de conexión al punto medio de uno de los devanados de fase. Según el teorema de Blondel, si se tienen 4 conductores se requiere de un medidor de tres elementos, o de un medidor de dos y medio elementos con la condición de tener los voltajes balanceados.

EL MEDIDOR DELTA DE 2 ½ ELEMENTOS

Este medidor trifásico está compuesto por dos sensores de voltaje y tres sensores de corriente. El punto común de los sensores de voltaje deberá estar conectado al conductor neutro. Si el neutro está verdaderamente conectado en el borne central (voltajes usados para definir el neutro son iguales entre los límites aceptables). Entonces solamente dos sensores de voltajes necesitan ser usados. Ver Fig. 2.20 para el tipo de conexión.

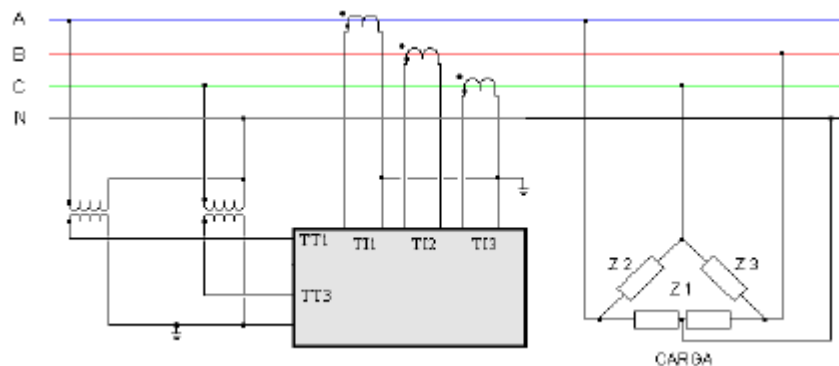


Figura 2.20 Medidor de 2 ½ elementos para conexión delta 4 hilos.

En donde:

TT1 mide V_{AN} y TT3 mide V_{CN} .

TI1 mide I_A , TI2 mide I_B y TI3 mide I_C .

El diagrama fasorial es el siguiente:

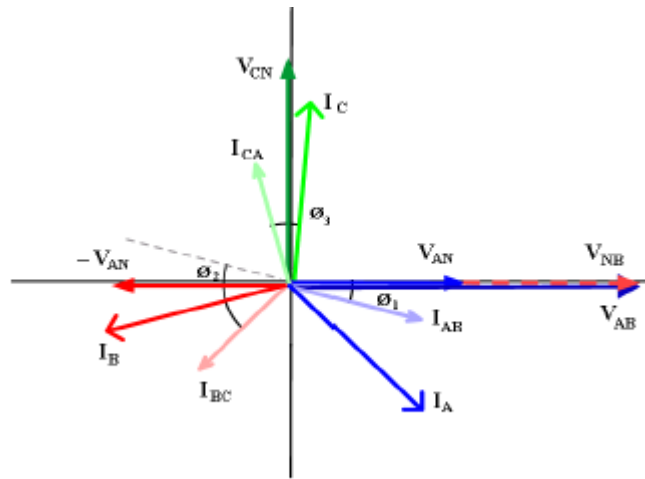


Figura 2.21 Diagrama fasorial de un medidor de 2 ½ elementos para conexión delta 4 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = V_{AN} I_A \cos(30^\circ + \phi_1) - V_{AN} I_B \cos(30^\circ - \phi_2) + V_{CN} I_C \cos \phi_3$$

$$Q = V_{AN} I_A \text{sen}(30^\circ + \phi_1) - V_{AN} I_B \text{sen}(30^\circ - \phi_2) + V_{CN} I_C \text{sen} \phi_3$$

EL MEDIDOR DE TRES ELEMENTOS

Este medidor de tres elementos cumple con el teorema de Blondel para mediciones de potencia trifásica de 4 conductores. Uno de los elementos puede tener un medio de la capacidad nominal de corriente y dos veces la capacidad nominal de voltaje de los otros dos elementos, aunque los tres elementos pueden ser de igual capacidad nominal previendo que en el circuito serán conectados transformadores de intensidad y transformadores de tensión de diferentes relaciones. Ver Fig. 2.22 para conexión.

Las dos bobinas de intensidad de igual capacidad son conectadas una en cada uno de los conductores de fase, entre los que está el devanado con conexión en su punto medio y las bobinas de tensión asociadas conectadas entre estos conductores de fase y el conductor del punto medio del devanado o neutro. La

bobina de intensidad de un medio de la capacidad nominal de corriente es conectada en el conductor de fase restante, y su bobina de tensión del doble de capacidad nominal es conectada entre el conductor de fase y el conductor del punto medio del devanado(neutro). Este método es preciso para todas las condiciones de carga y factor de potencia, con o sin desbalance de voltaje.

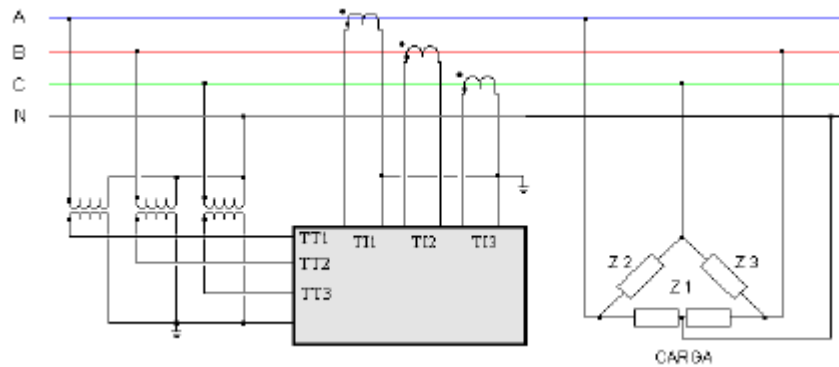


Figura 2.22 Medidor de 3 elementos para conexión delta 4 hilos.

En donde:

TT1 mide V_{AN} , TT2 mide V_{BN} y TT3 mide V_{CN} .

TI1 mide I_A , TI2 mide I_B y TI3 mide I_C .

El siguiente es el análisis vectorial visto por el medidor:

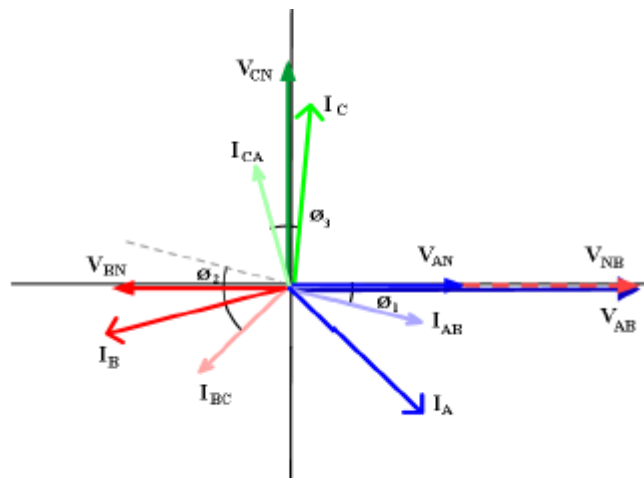


Figura 2.23 Diagrama fasorial de un medidor de 3 elementos para conexión delta 4 hilos.

Las potencias trifásicas, activa y reactiva, se calculan a partir de las siguientes ecuaciones:

$$P = V_{AN} I_A \cos(30^\circ + \phi_1) + V_{BN} I_B \cos(30^\circ - \phi_2) + V_{CN} I_C \cos \phi_3$$

$$Q = V_{AN} I_A \operatorname{sen}(30^\circ + \phi_1) + V_{BN} I_B \operatorname{sen}(30^\circ - \phi_2) + V_{CN} I_C \operatorname{sen} \phi_3$$

3. COMPONENTES ELECTRÓNICOS DEL MEDIDOR

3.1 EL CIRCUITO INTEGRADO ADE7754

Este dispositivo electrónico de última generación distribuido por la compañía Analog Devices salió al mercado recientemente. Su arquitectura versátil y de bajo costo hace de este integrado, una herramienta útil porque a parte de medir parámetros eléctricos como voltaje, corriente y energía activa, permite transmitir esta información a un ordenador mediante una interfase serial.

Este IC además posee un servicio de interrupción, lo cual unido a la interfase serial, permite conectarlo a un microcontrolador; que lo convierte en un medidor autónomo capaz de almacenar información, con posibilidades de descarga en un computador.

3.1.1 CARACTERÍSTICAS GENERALES

Este es un dispositivo de alta exactitud de medición de energía eléctrica con una interfase serial y un pulso de salida para propósitos de calibración. Posee seis canales analógicos: 3 de voltaje y 3 de corriente; para la medición de energía activa y aparente. Además permite la conexión de varios tipos de servicio como trifásico de 4 hilos y trifásico de 3 hilos.

EL IC ADE7754 permite realizar ajustes por canal, mediante la escritura de los registros internos, como: Corrección de offset, Calibración de fase, Calibración de ganancia.

Los datos del ADE7754 pueden ser obtenidos con la interfase serial SPI, para una posterior descarga sobre una computadora o algún otro dispositivo que tenga esta interfase. Y provee menos del 0.1% de error en medición de potencia activa.

El IC ADE7754 proporciona la siguiente información:

- Energía activa
- Energía aparente
- Cálculo RMS simultáneo en las seis entradas análogas.
- Muestras de la forma de onda de los 6 canales analógicos con diferentes frecuencias de muestreo 3.3 KSPS, 6.5 KSPS, 13 KSPS Y 26 KSPS.
- Detección de picos de voltaje y corriente

3.1.2 CARACTERÍSTICAS DE FUNCIONAMIENTO

- Voltaje de alimentación de 5V \pm 5% para AV_{DD} y DV_{DD}.
- Rango de temperatura de operación de [-40 °C; +85 °C].
- Rango de entrada en los pines analógicos máximo pico de 500mV con PGA (Registro de ganancia programable) de 1, 2 o 4 en todos los canales.
- Frecuencia de operación de 10 MHz.
- Referencia interna y externa de 2.4V \pm 8%
- Entradas lógicas /RESET, DIN, SCLK, CLKIN Y /CS, nivel alto de 2.4V mín. y nivel bajo de 0.8V máx., con corriente máx. de entrada de \pm 3 μ A.
- Salidas lógicas CF, IRQ, DOUT y CLKOUT, con nivel alto de 4 V mín. y nivel bajo de 1 V máx.

3.1.3 DIAGRAMA DE PINES

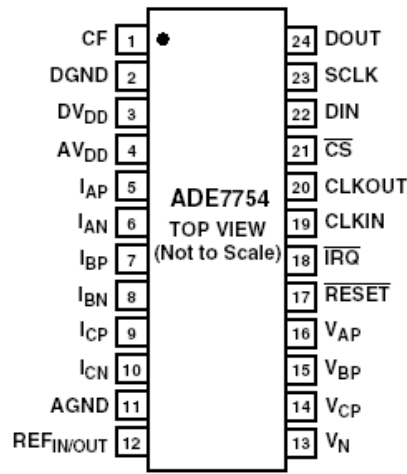


Figura 3.1 Diagrama de pines del IC ADE7754

3.1.4 DESCRIPCIÓN DE LAS FUNCIONES DE LOS PINES

Pin	Nemónico	Descripción
1	CF	Salida lógica de frecuencia para calibración. Esta salida es utilizada para propósitos operativos y de calibración; provee información de potencia activa.
2	DGND	Esta es la referencia a tierra para los circuitos digitales (por ejemplo multiplicadores, filtros y convertidores digital-a-frecuencia).
3	DV _{DD}	Fuente de alimentación digital. La alimentación debe ser 5V ± 5%. Este pin deberá ser desacoplado de DGND con un capacitor de 10 μF en paralelo con un capacitor cerámico de 100 nF.
4	AV _{DD}	Fuente de alimentación análoga. La alimentación debe ser 5V ± 5%. Este pin deberá ser desacoplado de AGND de la misma manera que DVDD.

Pin	Nemónico	Descripción
5,6 7,8 9,10	$I_{AP}, I_{AN};$ $I_{BP}, I_{BN};$ I_{CP}, I_{CN}	Entradas análogas para canales de corriente. Este canal se utiliza en conjunto con su transductor de corriente. Estas entradas son diferenciales con un máximo de $\pm 0.5V$, $\pm 0.25V$, y $\pm 0.125V$, dependiendo de la selección del registro de ganancia PGA.
11	AGND	Esta es la referencia a tierra análoga. Usada para ADC's, sensor de temperatura, y referencia; y debería ser utilizado para todos los circuitos análogos como filtros anti-aliasing y transductores de voltaje y corriente.
12	REF _{IN/OUT}	Este pin provee el acceso a la referencia de voltaje del IC, que tiene un valor de $2.4V \pm 8\%$. Una referencia externa puede ser conectada en este pin. Este pin debería ser desacoplado de AGND con un capacitor cerámico de $1 \mu F$.
13,14 15,16	$V_N, V_{CP};$ V_{BP}, V_{AP}	Entradas análogas para el canal de voltaje. Este canal se utiliza en conjunto con su etapa de atenuación. Estas entradas son tienen un nivel máximo de $\pm 0.5V$, $\pm 0.25V$, y $\pm 0.125V$, dependiendo de la selección de ganancia PGA.
17	/RESET	Un nivel bajo lógico en este pin mantiene los ADCs y la circuitería digital en una condición de reset.
18	/IRQ	Salida de petición de interrupción. Este es un pin activo bajo, de salida lógica de drenaje abierto.
19	CLKIN	Master Clock para ADCs y procesamiento digital de señales. Se puede utilizar un reloj externo con sus respectivos capacitores, o un resonador. La frecuencia debe ser de 10 MHz. Los capacitores deben ser de 22pF a 33pF según los requerimientos del cristal.
20	CLKOUT	Esta salida puede ser utilizada por otros dispositivos, y contiene la señal del oscilador.

Pin	Nemónico	Descripción
21	/CS	Chip Select. Es parte de la interfase serial. Este pin activo bajo permite al ADE7754 compartir el bus serial con otros dispositivos.
22	DIN	Data Input para la interfase serial. Es la entrada de datos.
23	SCLK	Serial Clock Input para la interfase serial síncrona. Toda la transferencia de datos serie son sincronizadas a este reloj.
24	DOUT	Data Output para la interfase serial. Esta es la salida de datos.

Tabla 3.1 Descripción de las funciones de los pines del IC ADE7754

3.1.5 DIAGRAMA BÁSICO DE CONEXIÓN

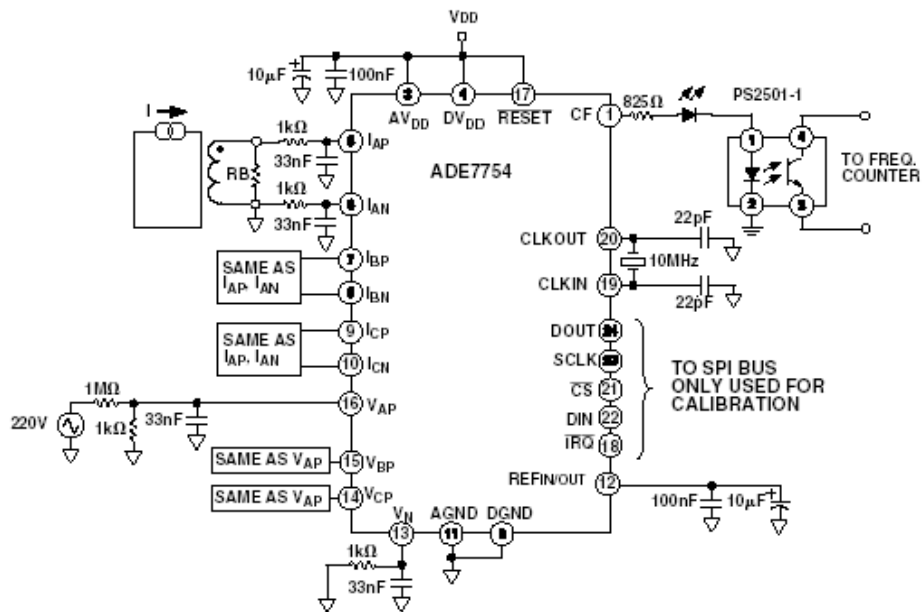


Figura 3.2 Diagrama básico de conexión del ADE7754

3.2 EL MICROCONTROLADOR PIC16F877

Un microcontrolador es un circuito integrado independiente, que no necesita memoria ni puertos externos pues los lleva en su interior, que facilita la tarea de diseño y reduce el espacio, traduciéndose todo a una aplicación final más económica y fiable.

El microcontrolador PIC16F877 es un circuito integrado fabricado por la empresa Microchip Technologies y es actualmente uno de los más utilizados a nivel mundial, la razón de esto es por que son rápidos, modernos, se pueden escribir programas y borrarlos muchas veces, poseen gran cantidad de documentación técnica y experimental a nivel mundial en la Red de Internet, de bajo costo, etc.

3.2.1 CARACTERISTICAS GENERALES

Un microcontrolador (MCU) es un circuito integrado que incorpora una unidad central de proceso (CPU) y una serie de recursos internos. La CPU permite que el microcontrolador pueda ejecutar instrucciones almacenadas en una memoria. Los recursos internos son memoria RAM, memoria ROM, memoria EEPROM, posee puerto serie, 5 puertos de entrada / salida, temporizadores, etc.

Se puede decir que es una evolución del microprocesador, al añadirle a este último las funciones que antes era necesario situar externamente con otros circuitos. El ejemplo típico está en los puertos de entrada / salida y en la memoria RAM, en los sistemas con microprocesadores es necesario desarrollar una lógica de control y unos circuitos para implementar las funciones anteriores, con un microcontrolador no hace falta porque lo lleva todo incorporado, además en el caso de tener que ampliar el sistema ya ofrece recursos que facilitan esto.

3.2.2 CARACTERÍSTICAS DE FUNCIONAMIENTO

- El microcontrolador PIC16F877 es un circuito integrado de 40 pines fabricado con tecnología CMOS, existen en diferentes tipos de encapsulado como por ejemplo DIP, SOIC y PLCC.
- La memoria programable es de 8Kx 14 palabras tipo FLASH, la memoria de datos (RAM) es de 368x 8 bytes y la memoria de datos no volátil (EEPROM) es de 256x 8 bytes.
- Capacidad de atender interrupciones por hardware y software
- Rango de voltaje de operación de 2.0 a 5.5 V y 25mA.
- La velocidad de operación máxima de reloj es de 20MHz, cada instrucción se ejecuta en dos ciclos de reloj por tanto, cada instrucción se ejecuta en 200ns.
- Posee Comunicación Síncrona SPI (Serial Peripheral Interfase) y Comunicación Asíncrona SCI (Serial Communication Interfase).

3.2.3 DIAGRAMA DE PINES

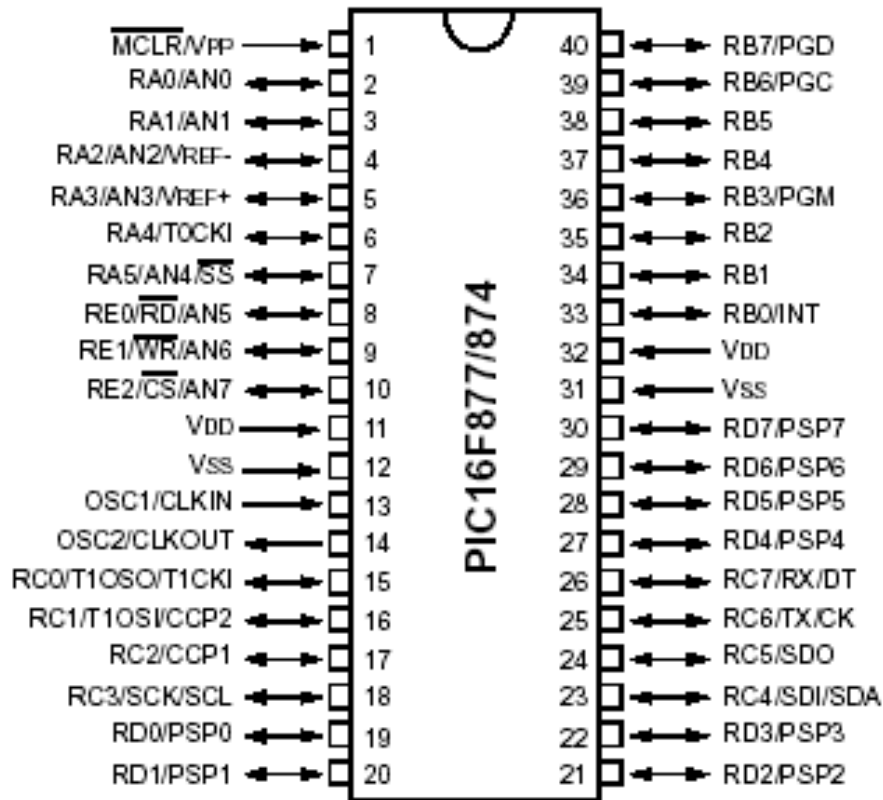


Figura 3.3 Diagrama de pines del microcontrolador PIC16F877

3.2.4 DESCRIPCION DE LOS PINES

Pin	Nemónico	Descripción
25, 26	RX, TX	Interfase de comunicación serial asíncrona SCI
23, 24, 18	SDI, SDO, SCK	Interfase de comunicación serial síncrona SPI
19-22, 27-30, 8-10	PSP0-PSP7, /RD, /WR, /CS	Interfase de comunicación paralela
1, 39, 40	/MCLR, PGD, PGC	Interfase de programación
1-5, 7-10	AN0 – AN7	Convertidor Analógico Digital
2-7	RA0 – RA5	Puerto A de datos bidireccional
33-40	RB0 – RB7	Puerto B de datos bidireccional

Pin	Nemónico	Descripción
15-18, 23-26	RC0 – RC7	Puerto C de datos bidireccional
19-22, 27-30	RD0 – RD7	Puerto D de datos bidireccional
8-10	RE0 – RE2	Puerto E de datos bidireccional
33	INT	Petición de interrupción
13,14	CLKIN, CLKOUT	Entrada y salida de reloj respectivamente
11	VDD	Alimentación 5 V
12	VSS	Tierra digital.

Tabla 3.2 Descripción de las funciones de los pines del microcontrolador PIC16F877

3.2.5 COMUNICACIÓN ASINCRONA SCI

El SCI (Serial Communications Interfase) esta formado por una unidad de transmisión y una unidad de recepción que son totalmente independientes, lo que permite que las comunicaciones sean bidireccionales, es decir, se permite la comunicación Full-duplex.

La unidad de información a transmitir está formada por un registro de un byte; al introducir un valor en este registro, comienza a desplazarse hacia la derecha el contenido del registro, enviando los bits por la línea de transmisión, a una velocidad que es configurada por el usuario. De la misma manera la unidad de recepción dispone de otro registro, que recibe los bits en serie y los va desplazando hasta obtener un dato en paralelo que puede ser leído.

Tanto el registro de transmisión como el de recepción están mapeados en la misma dirección de memoria, al escribir en esa dirección de memoria, el dato se cargara en el registro de transmisión. Al efectuar una lectura el dato se leerá del registro de recepción. Ambos registros comparten la misma dirección física de la memoria pero se trata de dos registros diferentes.

La interfase SCI reduce el número de líneas usadas para establecer una comunicación de datos, comparado con el uso del puerto paralelo. Solamente utiliza tres pines: el Transmisor de datos (TXD), el receptor de datos (RXD) y una línea de referencia denominada DGND (Tierra Digital).

3.2.6 COMUNICACIÓN SINCRONA SPI

La comunicación SPI (Serial Peripheral Interfase) es un sistema de comunicación serie síncrona de alta velocidad. El SPI puede ser utilizado para comunicar varios dispositivos entre sí, ya sean simples periféricos o varios microcontroladores.

Para realizar la comunicación, el Microcontrolador permite seleccionar entre dos modos de funcionamiento: **el modo maestro** y **el modo esclavo**. Cuando se realizan redes de comunicación (entre dos o más dispositivos) solamente esta permitido la existencia de un solo maestro, mientras que la de esclavos esta indefinida. Realmente su número viene dado por las necesidades del sistema en desarrollo.

La potencia de la interfase SCI llega al límite al permitir transmisiones full duplex (en ambos sentidos simultáneamente) entre un maestro y un esclavo. A partir de aquí, es posible realizar desde una simple comunicación unidireccional entre un microcontrolador y un periférico, hasta construir enlaces jerárquicos complejos entre un microcontrolador y/o periféricos.

Protocolo

Cuando el maestro tiene que mandar un mensaje a uno o varios esclavos deben proceder a realizar una selección de los mismos, trabajando al igual que si se tratara de un chip-enable. De esta forma, al ser activado el esclavo, recibe el dato manteniendo el sincronismo gracias a una señal de reloj conjunta. Es posible que cuando un esclavo sea activado con el fin de recibir un dato, este

desea enviar una trama de respuesta al maestro, esto será posible mientras su línea de activación la mantenga el maestro, de modo que si es necesario la transmisión se efectuara simultáneamente en los dos sentidos.

Existen cuatro líneas asociados a la SPI mediante las cuales es posible montar los diferentes enlaces:

SDO (Serial Data Out): esta es la línea por donde circulan los datos que el maestro quiere enviar a los esclavos, por lo tanto será la señal de salida de datos de la unidad que funciona como maestro y la señal de entrada de datos para los esclavos.

SDI (Serial Data In): por esta línea viajarán los datos que sean enviados desde algún esclavo hacia el maestro, de esta forma será una señal de entrada para el maestro y las respectivas salidas para los esclavos.

SCK (Serial Clock): representa la señal de reloj con la que se producen las comunicaciones, si bien es posible que cada unidad configure mediante software la velocidad deseada, es previsible pensar que en una comunicación solo podrá prevalecer una, siendo esta la del maestro. Por lo tanto, para los esclavos representará una señal de entrada mientras que para el maestro será una salida.

/SS (Slave Select): esta línea tiene una funcionalidad muy concreta en las unidades esclavas ya que representa sus respectivas entradas de habilitación. Es notable el ahorro de líneas de conexión que se genera en comparación con el puerto paralelo, donde es necesario como mínimo el cableado del bus de datos.

3.3 MEMORIA RAM NO VOLATIL DS1245Y

La memoria no volátil SRAM DS1245 1024k está organizada en 131,072 palabras de 8 bits. Cada memoria tiene en su interior una fuente de energía de litio y una circuitería de control que monitorea constantemente el Vcc en un rango de tolerancia. Cuando Vcc está fuera de rango, la fuente de energía de litio es automáticamente encendida y se habilita la protección de escritura para evitar datos inválidos.

No hay límite en el número de ciclos de escritura que pueden ser ejecutados, y no es necesario ningún tipo de circuitería extra para comunicar la memoria con un microprocesador.

3.3.1 CARACTERISTICAS GENERALES

Modo de lectura

La memoria DS1245 ejecuta un ciclo de lectura cuando WE está inactivo (alto) y CE y OE están activos (bajo). La única dirección especificada por las 17 entradas (A0-A16) define cual de los 131,072 bytes de datos será accesado.

Modo de escritura

La memoria DS1245 ejecuta un ciclo de escritura cuando el WE y CE están activos (bajo) después de que las entradas de dirección estén estables. Todas las entradas de dirección deben mantenerse válidas durante el ciclo de escritura. El OE debe mantenerse inactivo (alto) durante el ciclo de escritura.

Modo de retención de datos

La memoria DS1245Y provee capacidad plena para funcionar en valores de alimentación más grandes de 4.5 V y protección de escritura para valores de voltaje menores a 4.5 V.

Los datos son mantenidos en ausencia de Vcc sin ningún tipo de circuitería adicional. La memoria monitorea constantemente el Vcc. Cuando la fuente de alimentación decae, la memoria protege los datos escritos automáticamente, todas las entradas llegan a ser "no importa", y todas las salidas llegan a estar en alta impedancia. Cuando el Vcc cae aproximadamente a 3 V, se activa la fuente de energía de litio para retener los datos mediante un circuito de transferencia de alimentación.

3.3.2 CARACTERÍSTICAS DE FUNCIONAMIENTO

- Tamaño de la memoria de 128K Bytes.
- Tipo de bus de datos paralelo.
- 10 años mínimos de retención de datos en ausencia de fuente externa.
- Los datos son automáticamente protegidos durante la pérdida de potencia.
- Reemplaza las memorias RAM estáticas, EEPROM o memoria FLASH de 128k x 8.
- Ilimitados ciclos de escritura.
- CMOS de bajo consumo de potencia.
- Tiempos de acceso de lectura y escritura tan rápida como 70 ns.
- Fuente de energía de Litio desconectada eléctricamente hasta que es alimentada por primera vez.
- Rango de alimentación de $\pm 10\%$ Vcc, de 4.5 - 5.5 V.
- Rango de temperatura de 0 °C a 70 °C.
- Encapsulado tipo DIP de 32 pines.

3.3.2 DIAGRAMA DE PINES

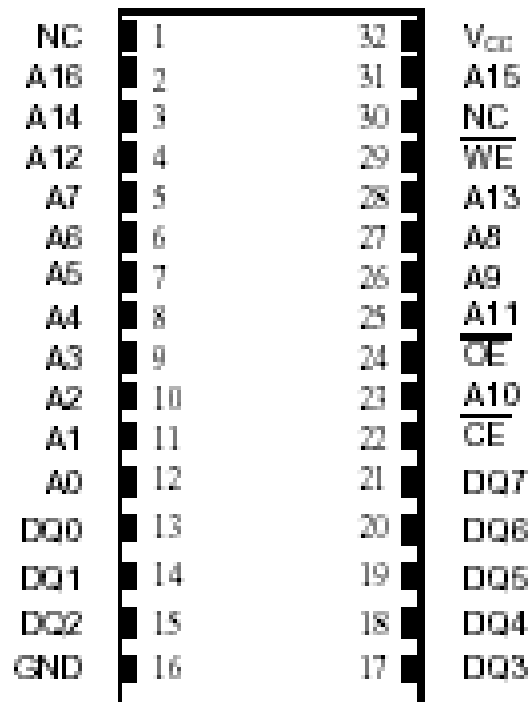


Figura 3.4 Diagrama de pines de la memoria RAM no volátil DS1245Y

3.3.3 DESCRIPCIÓN DE LOS PINES

Pin	Nemónico	Descripción
2-12, 23, 25-28, 31	A0-A16	Entradas para direccionamiento
13,14, 17-21	DQ0-DQ7	Bus de datos
22	/CE	Habilitador del integrado
29	/WE	Habilitador de escritura de datos
24	/OE	Habilitador de lectura de datos
32	VCC	Alimentación (+5 V)
16	GND	Tierra
1,30	NC	Sin conexión

Tabla 3.3 Descripción de las funciones de los pines de la memoria RAM no volátil DS1245Y

4.0 HARDWARE DEL MEDIDOR

En la actualidad, los medidores de estado sólido están ganando más terreno sobre los medidores electromecánicos, debidos a su alta precisión y no son sujetos a posibles errores debido a desgastes y deformaciones en las partes móviles como es el caso en los medidores electromecánicos.

El sistema de medición propuesto está conformado por un IC ADE7754 que es un medidor de alta exactitud de energía trifásico y que proporciona información adicional acerca de los parámetros eléctricos como voltaje y corriente rms por cada fase, potencia activa, potencia aparente, potencia Reactiva y factor de potencia. Todo esto a través de una interfase de comunicación serial síncrona SPI. Una salida de petición de interrupción /IRQ de tipo drenaje abierto con salida lógica baja. Esta petición llega a ser activa baja cuando una o mas interrupciones han ocurrido en el ADE7754.

El medidor posee 6 canales analógicos, 3 de voltaje y 3 de corriente donde cada una de estas entradas permite un nivel de voltaje máximo de 500 mV pico por lo que se cuenta con su respectiva red de atenuación para cada fase. El sistema está capacitado para operar en un sistema trifásico a 240 V y 5 A rms, para el canal de voltaje y el canal de corriente respectivamente; el medidor es capaz de realizar mediciones sobre cualquier tipo de servicio ya sea delta o estrella, con o sin aterrizaje del sistema; mediante la combinación de los jumper que se encuentran dentro de medidor, en nuestro caso el sistema esta configurado para un sistema 9S o servicio estrella de 4 hilos.

Para que el IC ADE7754 pueda operar necesita 355 mV rms en sus entradas analógicas, para ello cada canal está provisto de una etapa de acondicionamiento de señal conformada por la red de atenuación y por filtros anti-alias en los 6

canales y filtros de compensación de fase para los canales de corriente ya que el uso de transformadores de corriente (TC) introduce un desfase que debe de ser corregido.

El sistema además incluye un microcontrolador PIC16F877 capaz de comunicarse con el ADE7754, mediante su interfase serial síncrona SPI conformada por las líneas de reloj (SCLK), datos de salida (DOUT), datos de entrada (DIN) y habilitación (/CS). En este sistema se utiliza la arquitectura maestro esclavo donde el microcontrolador es el maestro y el ADE7754 y otros microcontroladores funcionan como esclavos; este microcontrolador PIC16F877 se encarga además de realizar el protocolo de comunicación con la computadora, esto se realiza mediante la interfase serial asíncrona SCI; ambas interfases SPI y SCI operan a niveles TTL en el microcontrolador por lo que se requiere de un convertidor de nivel TTL a RS232 para la comunicación SCI con la computadora.

La función principal del microcontrolador será la de ejecutar un protocolo que le permitirá al usuario, obtener información del ADE7754, al acceder la base de datos de la memoria.

Para la programación del microcontrolador PIC16F877 se necesita de una tarjeta programadora, para descargar el programa que residirá en la memoria EEPROM del PIC16F877; lo que permitirá al microcontrolador ser autónomo. Cabe destacar que el programador si bien es una herramienta fundamental para el desarrollo del medidor, no forma parte del sistema.

4.1 DIAGRAMA DE BLOQUES

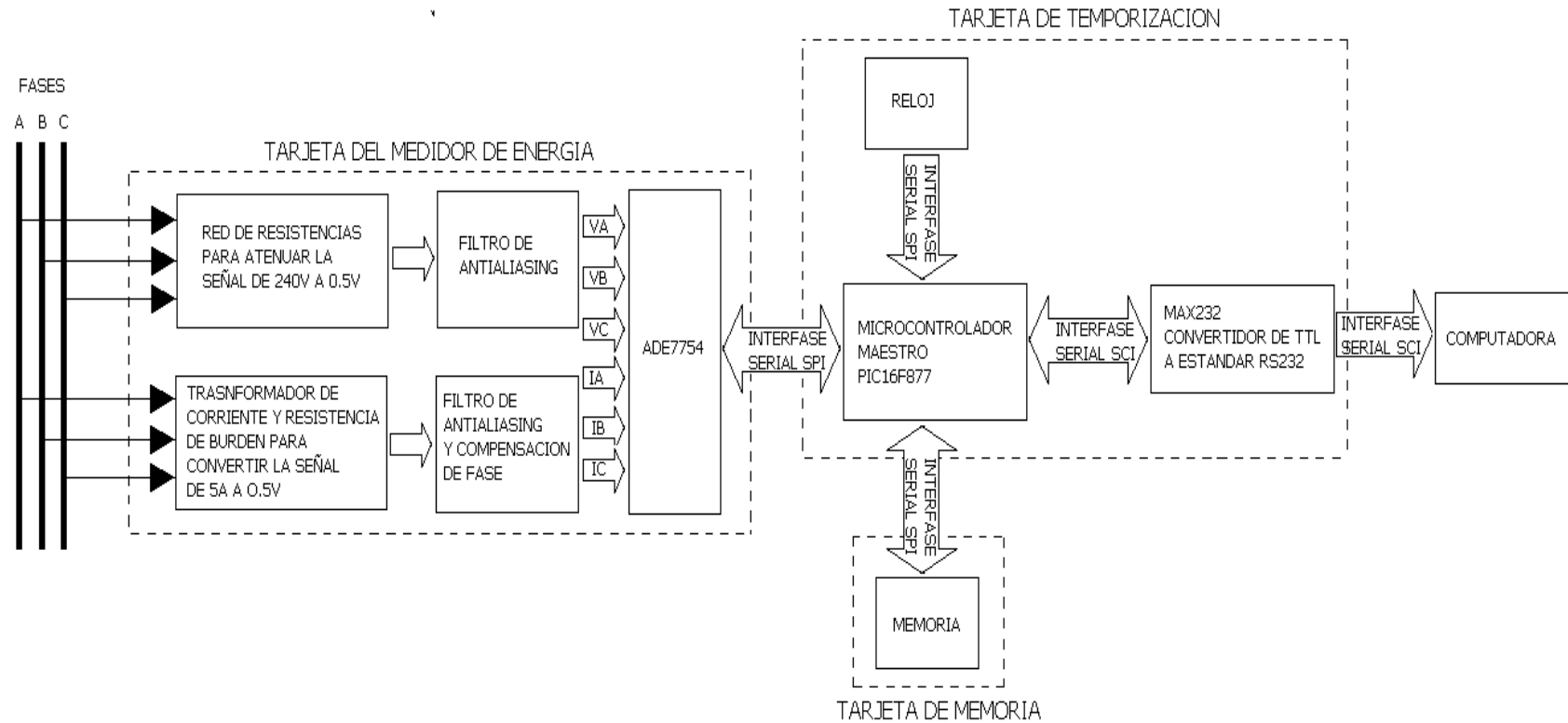


Figura 4.1 Diagrama de bloques del medidor trifásico multifunción

4.2 ETAPA DEL MEDIDOR DE ENERGIA.

4.2.1 TARJETA DEL MEDIDOR ENERGIA.

ACONDICIONAMIENTO DE SEÑAL

Canal de corriente

Para utilizar estos canales es necesario utilizar transductores de corriente, y luego un filtro anti-alias los cuales sirven para evitar el traslape de las muestras de la señal de interés. Cabe destacar que todos los filtros en las tarjetas de medidor pueden ser habilitados o deshabilitados por medio de los Jumper correspondientes identificados en la tarjeta. Los transformadores de corriente (TC) pueden ser utilizados como transductores en conjunto con una resistencia de Burden, para realizar la transformación de corriente a voltaje que para este caso en particular es de 355mVrms que necesita el ADE7754 en sus entradas analógicas a full escala.

Los transformadores de corriente provocan un desfase en la señal, para lo cual el sistema está provisto de una red de filtros de compensación conformado por resistencias y capacitores para la compensación de fase. Además la compensación de fase de cada canal de corriente, pueden realizarse mediante la escritura de registros internos del ADE7754 (registros APHCAL, BPHCAL Y CPHCAL para las fases A, B y C respectivamente). Para propósitos de calibración.

Canales de voltaje

Para utilizar estos canales es necesario utilizar un circuito de atenuación de voltaje, que consiste de una red de resistencias o divisor de voltaje, y luego un filtro anti-alias al igual que los canales de corriente

A continuación se muestra el acondicionamiento de señal para los canales analógicos de voltaje y corriente del ADE7754, para una fase. Ver Fig. 4.2.

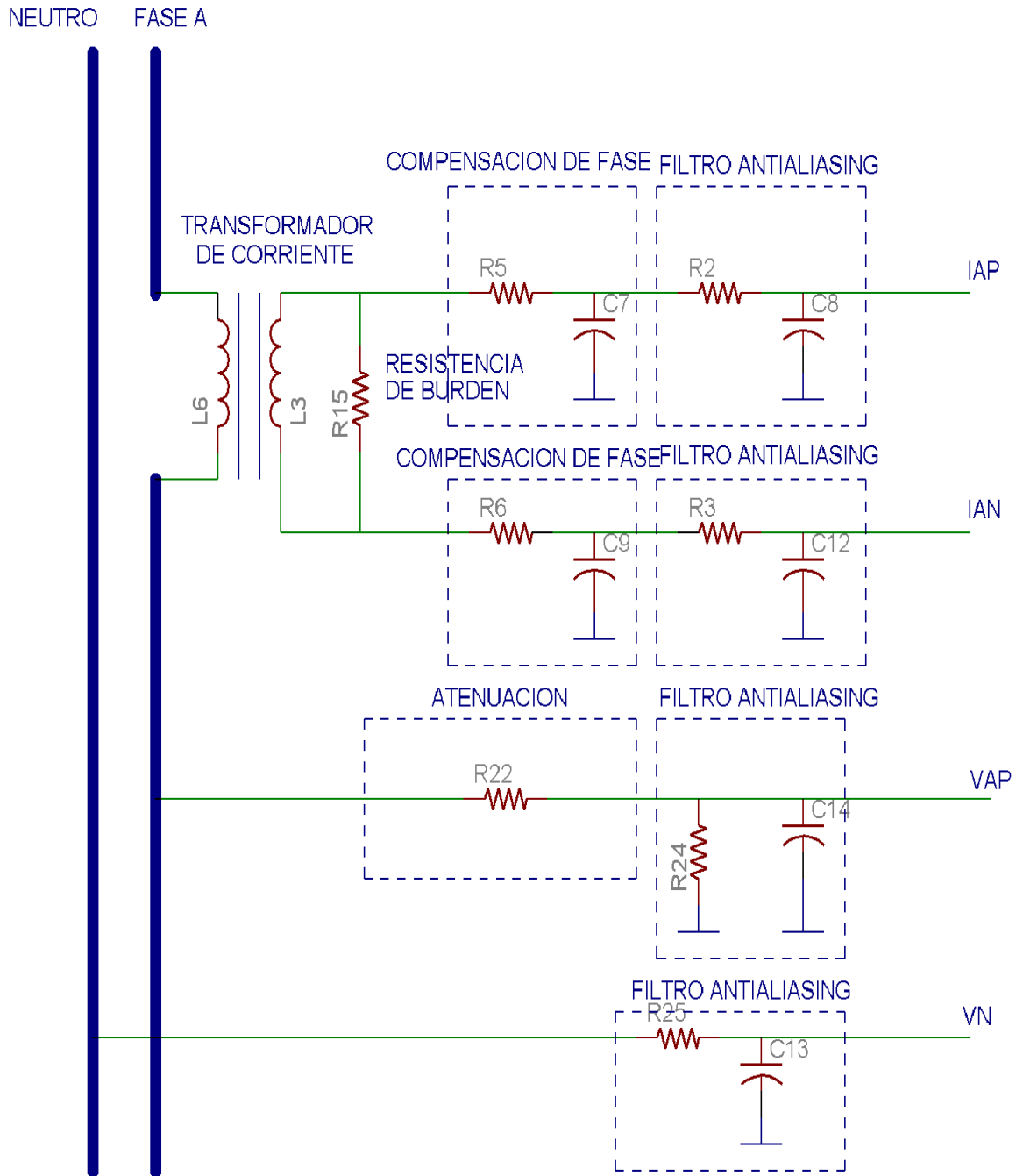


Figura 4.2 Etapa de acondicionamiento de señal de canales de voltaje y corriente

TARJETA DEL MEDIDOR TRIFÁSICO.

Esta tarjeta contiene la etapa de acondicionamiento de señal para los canales analógicos de voltaje y corriente del ADE7754 de las tres fases de alimentación; además de las líneas de conexión que corresponden al bus de comunicación SPI de microcontrolador maestro PIC16F877 y los indicadores LED del contador de frecuencia (CF) para la potencia activa y la salida /IRQ que indica la petición de un servicio de interrupción. Cabe destacar que el ADE posee un cristal de operación de 10Mhz que es el valor recomendado por fabricante para una operación optima del IC ADE7754. Ver Fig. 4.3

TARJETA DEL MEDIDOR DE ENERGIA

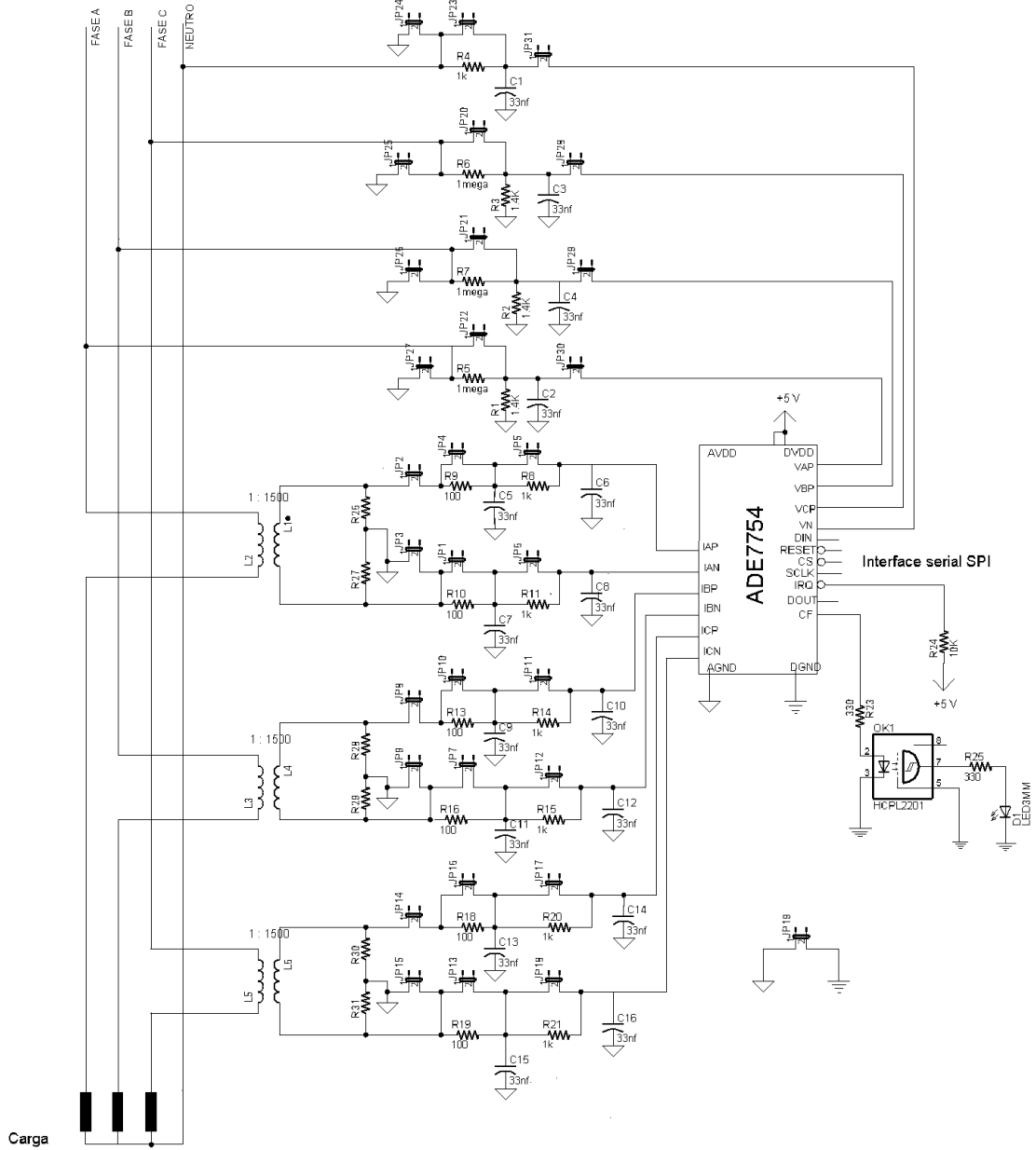


Figura 4.3 Tarjeta del IC ADE7754

4.2.2 TARJETA DE LA ETAPA DE MEMORIA

Esta tarjeta esta provista de una memoria del tipo RAM estática de una densidad de 128Kx8 del tipo DS1245Y-120 de la compañía DALLAS semiconductor que se ha implementado con una interfase serial periférica SPI para adecuarse a la interfase del microcontrolador maestro y a nuestras necesidades de almacenamiento de datos, por lo que se utilizo un microcontrolador PIC16F877 para su implementación ya que esta memoria DS1245Y-120 posee un bus de direcciones de 17 líneas en paralelo para poder accesar a las 128K localidades de memoria, el puerto E del microcontrolador habilita la memoria en el pin CE y las operaciones de escritura y lectura con los pines WE y OE respectivamente, los pines del bus de datos corresponden al puerto B del microcontrolador así como los pines del bus de direcciones corresponden a los puertos A, C y D. Cabe destacar que este microcontrolador funciona en la modalidad de esclavo y estará siendo controlado por el PIC principal del medidor cuando este así lo solicite. Ver Fig. 4.4.

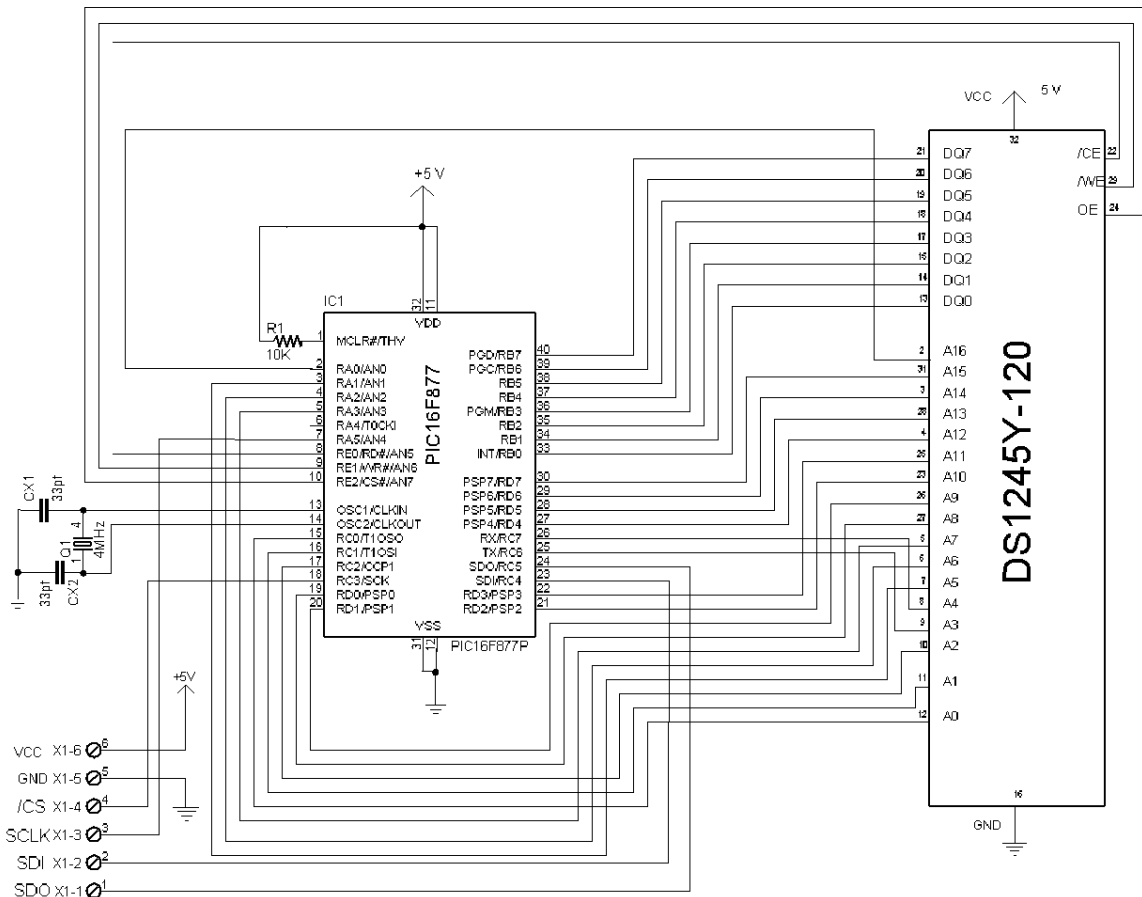


Figura 4.4 diagrama de implementación de una memoria con interfase SPI.

4.2.3 TARJETA DE TEMPORIZACIÓN

Esta tarjeta ha sido implementada con dos microcontroladores uno de ellos es el PIC16F877 maestro que es el encargado de coordinar todos los eventos del medidor ya que controla a la memoria con interfase SPI, al IC ADE7754 y al reloj. Este último también fue implementado en un PIC16F877 para tener compatibilidad con la interfase serial SPI y es el encargado de llevar el tiempo de cada uno de los eventos que se llevan a cabo dentro de las mediciones que se realizan. Posee un formato de salida de 4 bytes que corresponden al tiempo en segundos que posteriormente se tratara desde la computadora para llevar bitácora de las actividades en segundos, minutos, horas, días y meses para un completo historial de las mediciones realizadas. Ver figura 4.5.

TARJETA DE TEMPORIZACION DEL MEDIDOR Y MICROCONTROLADOR MAESTRO

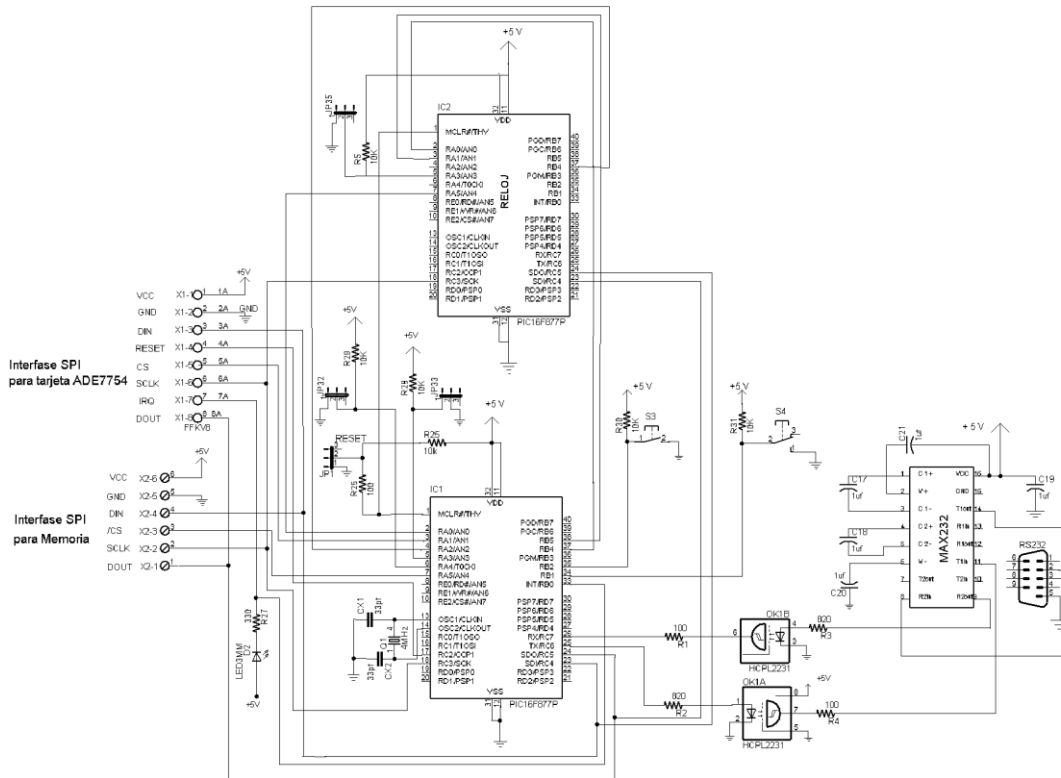


Figura 4.5 tarjeta de Temporización donde se incluyen también el microcontrolador maestro y el convertidor TTL-RS232

Esta tarjeta contiene todos los componentes necesarios para que el microcontrolador opere de manera autónoma, es decir, su resonador, su circuito de reset, y su convertidor de TTL a RS232. Incluye conectores que permite tener acceso a los pines más importantes del microcontrolador, como son los puertos, la interfase serial SPI y SCI.

También contiene un IC MAX232 que se trata de un convertidor de nivel TTL a RS232 para que el microcontrolador PIC16F877 pueda establecer comunicación con una computadora mediante el puerto serial por lo que cuenta con un conector DB9 para poder realizar la conexión, y así establecer una comunicación mediante la interfase del usuario desarrollada con el programa LabVIEW.

5. PROTOCOLO DE COMUNICACIÓN

5.1 DIAGRAMA DE BLOQUES DEL SISTEMA

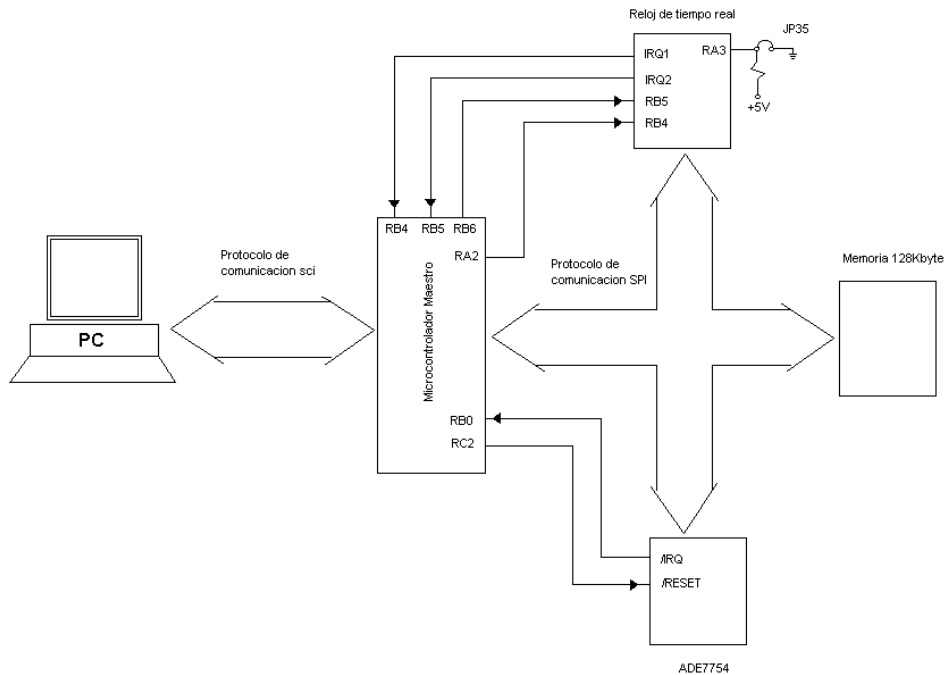


Figura 5.1 Diagrama de bloques del sistema

El medidor trifásico multifunciones está formado por 4 módulos: un microcontrolador maestro (PIC16F887), un reloj de tiempo real con interfaz SPI (implementado con un PIC16F887), una memoria con interfaz SPI (implementada con un PIC16F887A y una memoria direccionable DS1245Y/AB) y el IC medidor de energía ADE7754. El microcontrolador maestro se puede comunicar con una computadora personal a través de la interfaz SCI.

5.2 CONFIGURACIÓN DE LOS PROTOCOLOS IMPLEMENTADOS EN EL MICROCONTROLADOR MAESTRO

PROTOCOLO SCI (COMUNICACIÓN ASÍNCRONA)

- 9600 baudios.
- un bit de stop.
- un bit de start.
- 8 bit de datos.

PROTOCOLO SPI (COMUNICACIÓN SÍNCRONA)

- Se configura el PIC16F877 como maestro.
- Frecuencia de reloj 1 MHz.
- Cuando el canal de comunicación está ocioso, el reloj esta en nivel bajo.
- Los datos se transmiten en flancos de bajadas del reloj.
- Los datos de entrada se muestrean en el final del tiempo del dato de salida.
- El #CS del ADE7754 es controlado con RA1.
- El #CS del Reloj de tiempo real es controlado con RA0.
- El #CS de la memoria de 128kbytes x 8 es controlado con RA5.

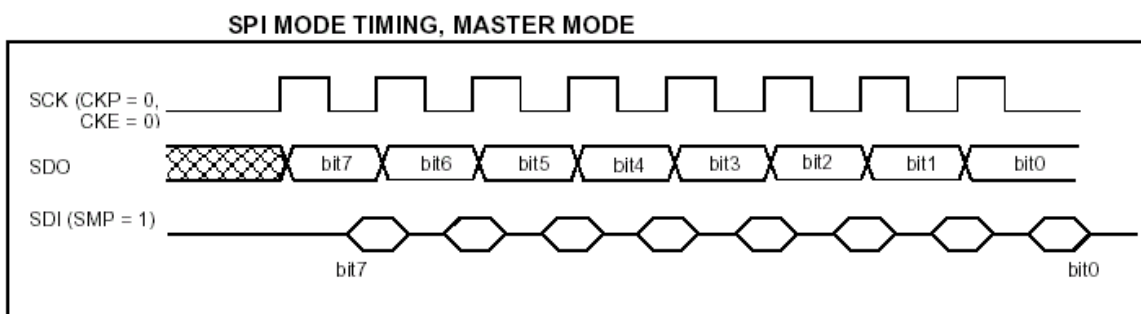


Figura 5.2 Diagrama de tiempos para la configuración de la comunicación SPI

Adicional al modulo SPI, el microcontrolador maestro utiliza las interrupciones de cambio de estado en los pines RB5, RB4 y la interrupción de petición externa del pin RB0.

Cuando el cambio de estado ocurre en el pin RB5 (esto ocurre cada 3 ó 5 minutos según la configuración del reloj) se mide la energía y se acumula. Cuando el cambio de estado ocurre en el pin RB4 (cada 12 ó 20 minutos) se mide la energía y luego es promediada con las otras tres anteriores, Además se miden los valores RMS de voltaje y corriente, luego estos datos son guardados en la memoria DS1245Y/AB.

La interrupción de petición externa es utilizada para saber cuando hay un dato valido de energía en el ADE7754.

5.3 INTERFAS SERIAL SPI DEL ADE7754

La interfaz serial del ADE7754 es una serial asíncrona (SPI), por tanto esta desarrollada con cuatro señales: SCLK, DIN, DOUT y #CS. Toda operación de transferencia de datos esta sincronizada con la señal del reloj SCLK (señal de entrada). Los datos enviadas son desplazados por la salida lógica DOUT en el flanco de subida de SCLK. Los datos de entrada son desplazados a través de la entrada lógica DIN en el flanco de caída de SCLK.

La entrada lógica #CS es el chip-select. Esta entrada es usada cuando múltiples dispositivos forman el bus de serial. En el flanco de caída de #CS la interfaz serial se resetea y el ADE7754 entra en el modo comunicación (espera un comando de lectura o escritura). #CS debe estar en nivel bajo para la transferencia de datos, poner en nivel alto #CS durante una transferencia de datos aborta la transferencia y pone el bus serial en alta impedancia. #CS puede estar siempre en nivel bajo si el ADE7754 es el único dispositivo en el bus serial sin embargo con #CS siempre

en nivel bajo toda inicialización de transferencia debe de ser terminada para que el ADE7754 regrese al modo de comunicación.

Todas las funcionalidades de ADE7754 son accesadas por medio de varios registros. Vea la figura 5.3. El contenido de estos pueden ser escritos o leídos usando la interfaz serial. Cuando el ADE7754 es energizado o reinicializado a través del pin #RESET o #CS cambia de un estado alto a bajo, este entra en modo comunicación. En modo comunicación el ADE7754 espera que un MCU (microcontrolador) envíe un comando o petición, este comando es guardado en un registro de un byte llamado "communication register".

El dato escrito en este registro contiene la dirección de un registro interno del ADE7754 y especifica si la siguiente transferencia de datos es lectura o escritura del registro direccionado por el comando. Por tanto toda operación de transferencia con el ADE7754 ya sea lectura o escritura a los registros internos, debe iniciar con la escritura al registro "communication register".

Las figuras 5.4 y 5.5 muestran la secuencia de transferencia de datos para una operación de lectura y escritura, respectivamente.

Una transferencia de datos es completada cuando el LSB del dato contenida en el registro diseccionado es transmitido desde el ADE7754.

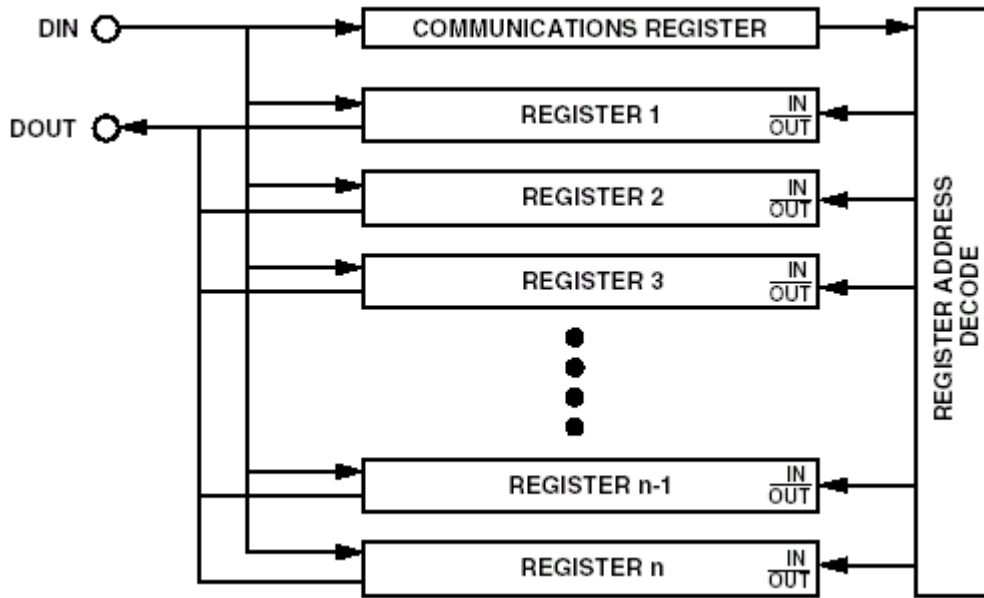


Figura 5.3 Direccionamiento de los registros del ADE7754 vía la interfaz serial

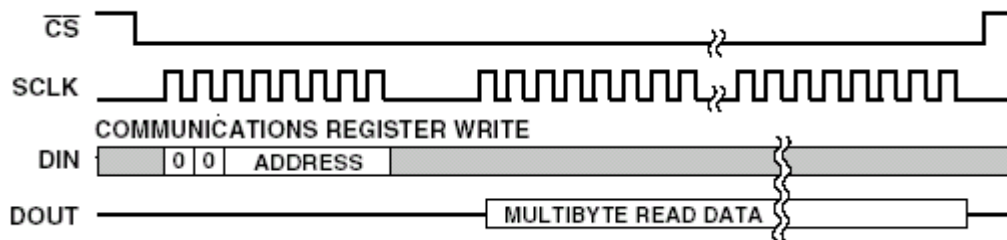


Figura 5.4 Lectura de datos del ADE7754 vía la interfaz serial

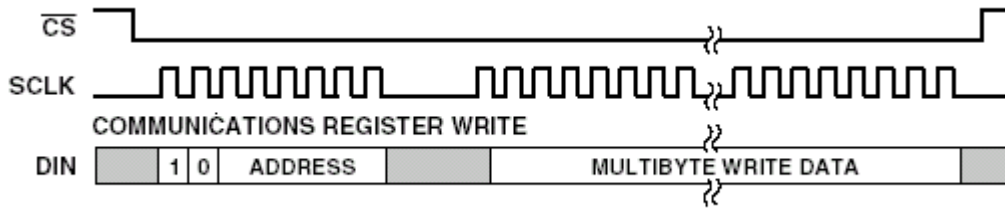


Figura 5.5 Escritura de datos del ADE7754 vía la interfaz serial

5.4 INTERRUPCIONES INTERNAS CON EL ADE7754.

Las interrupciones del ADE7754 son manejadas a través del registro de estado de las interrupciones (STATUS[15:0]) y el registro de habilitación de interrupciones (IRQEN[15:0]). Cuando un evento de interrupción ocurre en el ADE7754, la correspondiente bandera en el registro de estado de interrupción es puesta en 1 y el pin #IRQ es puesto a cero logico. para determinar la fuente de la interrupción, el sistema maestro (MCU) debe realizar una lectura del registro de estados de interrupciones con reset en la dirección 11h. La salida lógica #IRQ es puesta a nivel alto cuando el comando de lectura es recibido por el ADE7754 vea la figura 5.6.

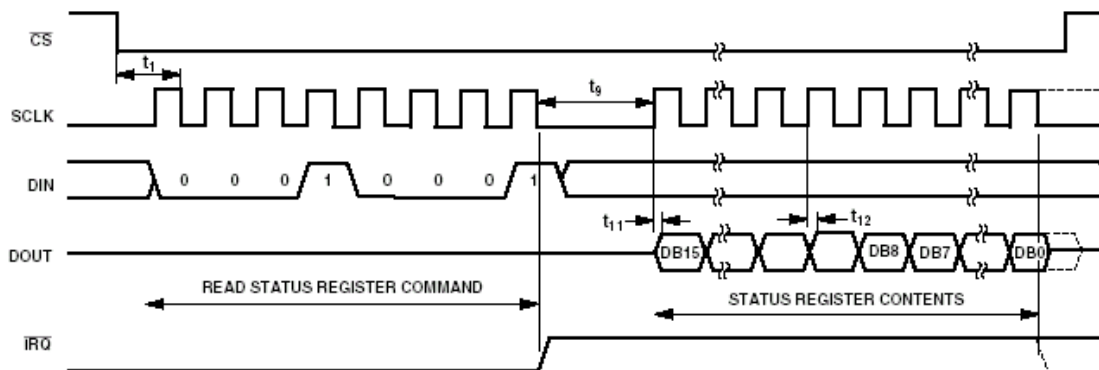


Figura 5.6 Diagrama de tiempos de la interrupción

Cuando se realiza una lectura con reset al registro de estados de interrupciones, no se pierden eventos de interrupciones, debido a que el ADE7754 esta diseñado para que capture un evento de interrupción durante la esta lectura y cuando la lectura de este registro termina la #IRQ es puesta en nivel bajo indicando que se a dado un evento de interrupción nuevamente y el registro de estado de la banderas de interrupción es actualizado.

5.5 RELOJ DE TIEMPO REAL

El reloj de tiempo real esta implementado con un PIC16F877; usando el TMR1, el modulo SPI Y cinco pines de los puertos I/O. El modulo SPI esta configurado como se muestra en la figura 5.7, de forma tal que interactué correctamente con la configuración del modulo SPI microcontrolador maestro (vea la figura 5.2)

Configuración del modula SPI del reloj:

- Se configura el PIC16F877 como esclavo
- Cuando el canal de comunicación está ocioso, el reloj esta en nivel bajo
- Los datos se transmiten en flancos de bajadas del reloj

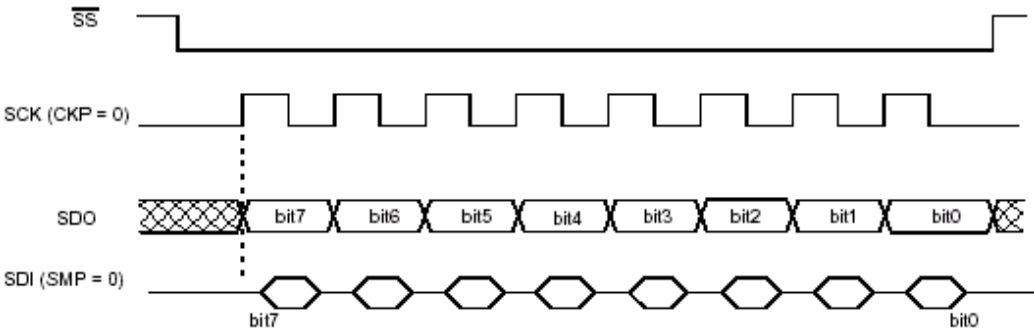


Figura 5.7 Diagrama de tiempos de la del modulo SPI del Reloj

Adicional a la comunicación SPI, se han implementado dos pines de salida los cuales realizan un cambio de estado cuando han pasado 3 ó 5 minutos (RA1) y cuando han pasado 12 ó 20 minutos (RA0), para que el microcontrolador maestro ejecute una interrupción determinada, también sean implementado dos pines configurados como entradas (RB4 y RB5); cuando RB4 cambia el estado ejecuta una interrupción, la cual envía el tiempo actual en 4 bytes, enviando primero el MSB y por ultimo el LSB. Y cuando RB5 cambia de estado se ejecutan una interrupción, la cual espera cuatro bytes de PIC16F877 configurado como maestro, para fijar la hora del reloj.

El reloj esta diseñado para que cada 3 ó 5 y 12 ó 20 segundos realizan un cambio de estado en los pines RA1 y RA0 respectivamente. y luego envíe el tiempo actual, a través del modulo SPI.

La selección de los tiempos de acumulación de energía se realizan a través de el jumper JP35 situado en el pin RA3.

El tiempo es transmitido en 4 bytes, primero el MSB y por ultimo el LSB. Estos 4 bytes contienen el tiempo en segundos.

5.6 MEMORIA DE 128Kbytes X 8

Esta memoria esta conformado por un PIC16F877 y una DS1245Y/AB. La cual es direccionada a través del microcontrolador para guardar o leer bytes. El microcontrolador es configurado para que su modulo SPI este como esclavo y este sea controlado por el microcontrolador maestro.

El modulo SPI se configura igual que el modulo del Reloj de tiempo real vea la figura 5.7

6. INTERFASE GRAFICA DEL MEDIDOR

6.1 EL LENGUAJE DE PROGRAMACIÓN LABVIEW

LabVIEW es un lenguaje de programación gráfico altamente productivo para construir sistemas de adquisición de datos e instrumentación. Con LabVIEW es posible crear interfaces de usuario. Ver figura 6.1.



Figura 6.1 El lenguaje de programación gráfica LabVIEW

Una de las ventajas más grandes de LabVIEW es la ganancia de productividad que ofrece; porque se tiene la posibilidad de cambiar de manera fácil el código, en un tiempo corto. LabVIEW ha llegado a ser una herramienta estándar industrial para: pruebas y medición, control de procesos, automatización de fábricas, monitoreo de máquinas, investigación y análisis, etc.

Los Instrumentos Virtuales (VI) de LabVIEW son modulares en su diseño, así cualquier VI puede ejecutarse individualmente o ser usado como parte de otro VI. Es posible crear iconos para los VIs propios.

6.2 FLUJOGRAMAS DE LA INTERFASE GRAFICA DEL USUARIO

6.2.1 PRINCIPAL

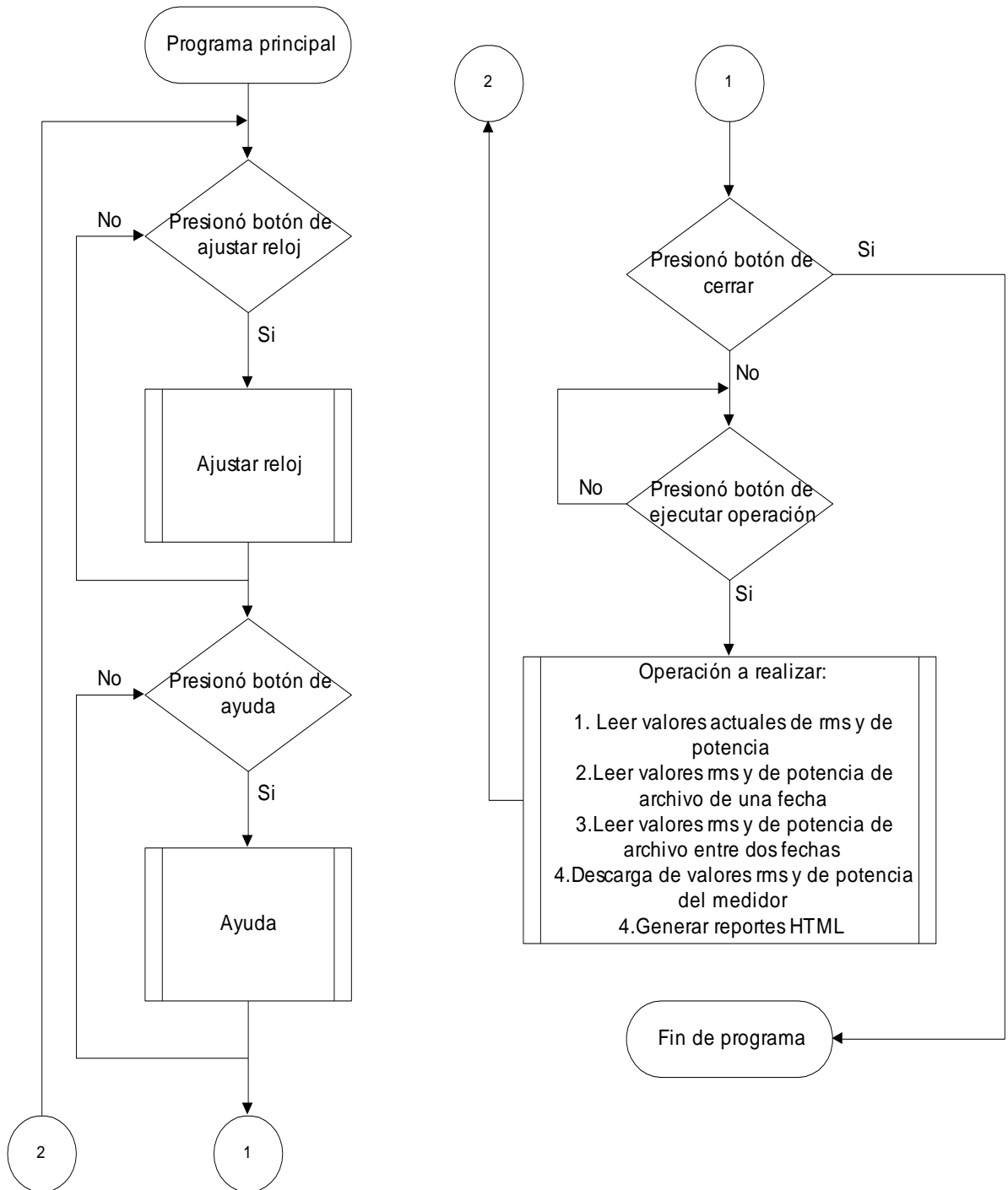


Figura 6.2. Flujoograma del programa principal

Subrutina de leer valores actuales rms y de potencia

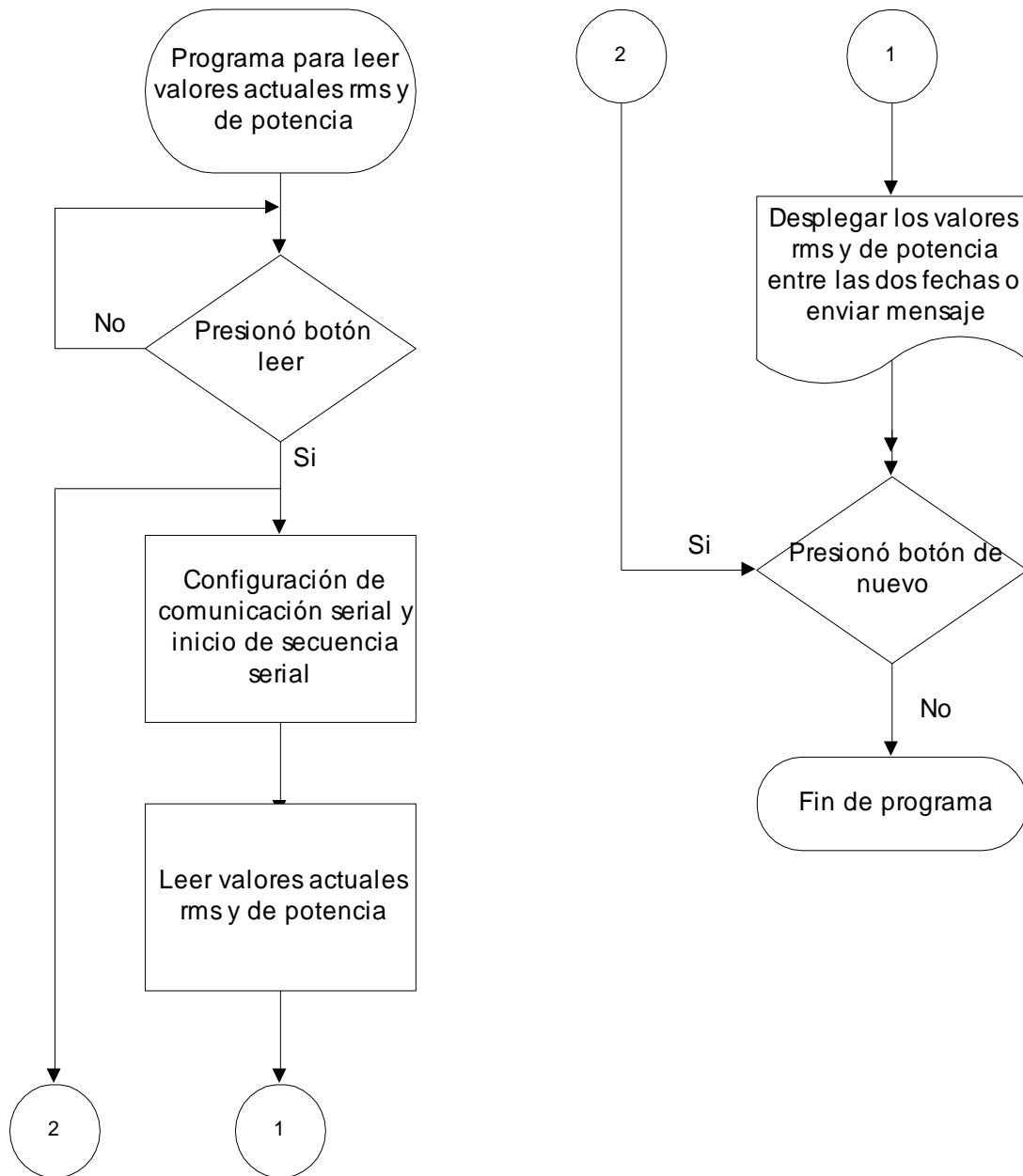


Figura 6.3. Flujograma de subrutina de lectura de valores actuales rms y de potencia

Subrutina de leer valores rms y de potencia de archivo de una fecha específica

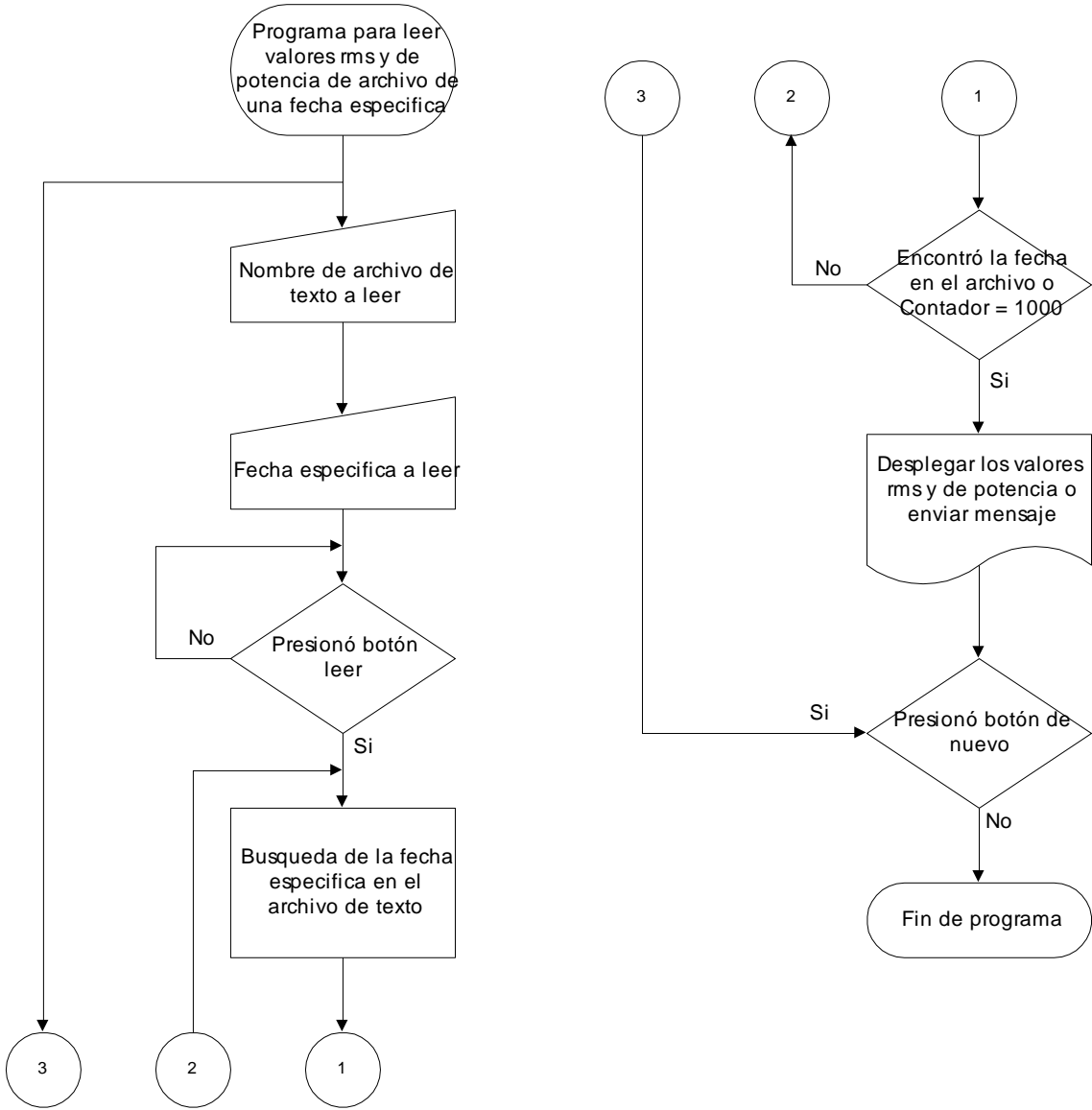


Figura 6.4. Flujograma de subrutina de lectura de valores rms y de potencia de archivo de una fecha específica

Subrutina de leer valores rms y de potencia de archivo entre dos fechas

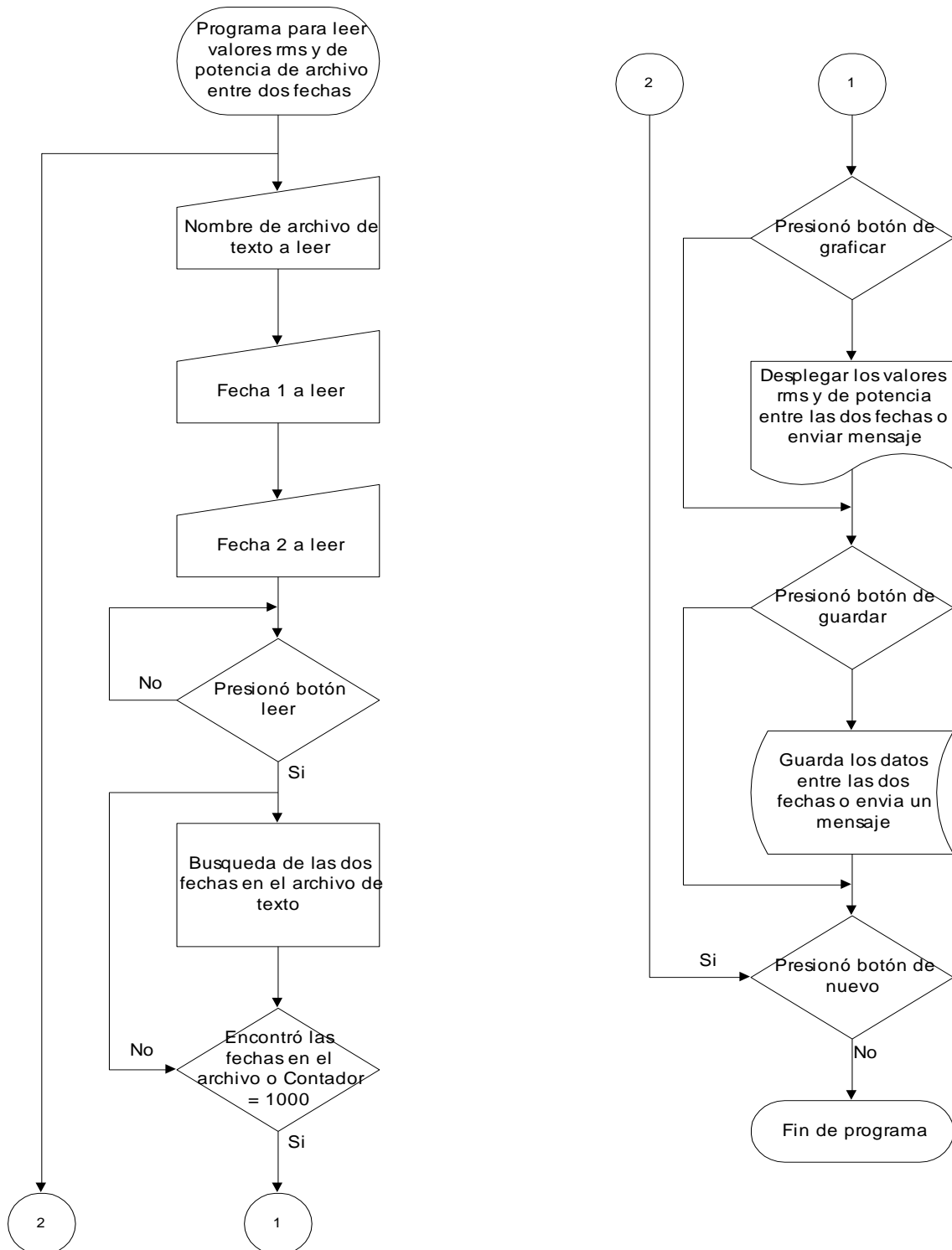


Figura 6.5. Flujograma de subrutina de lectura de valores rms y de potencia de archivo entre dos fechas

Subrutina de descarga del medidor de valores rms y de potencia

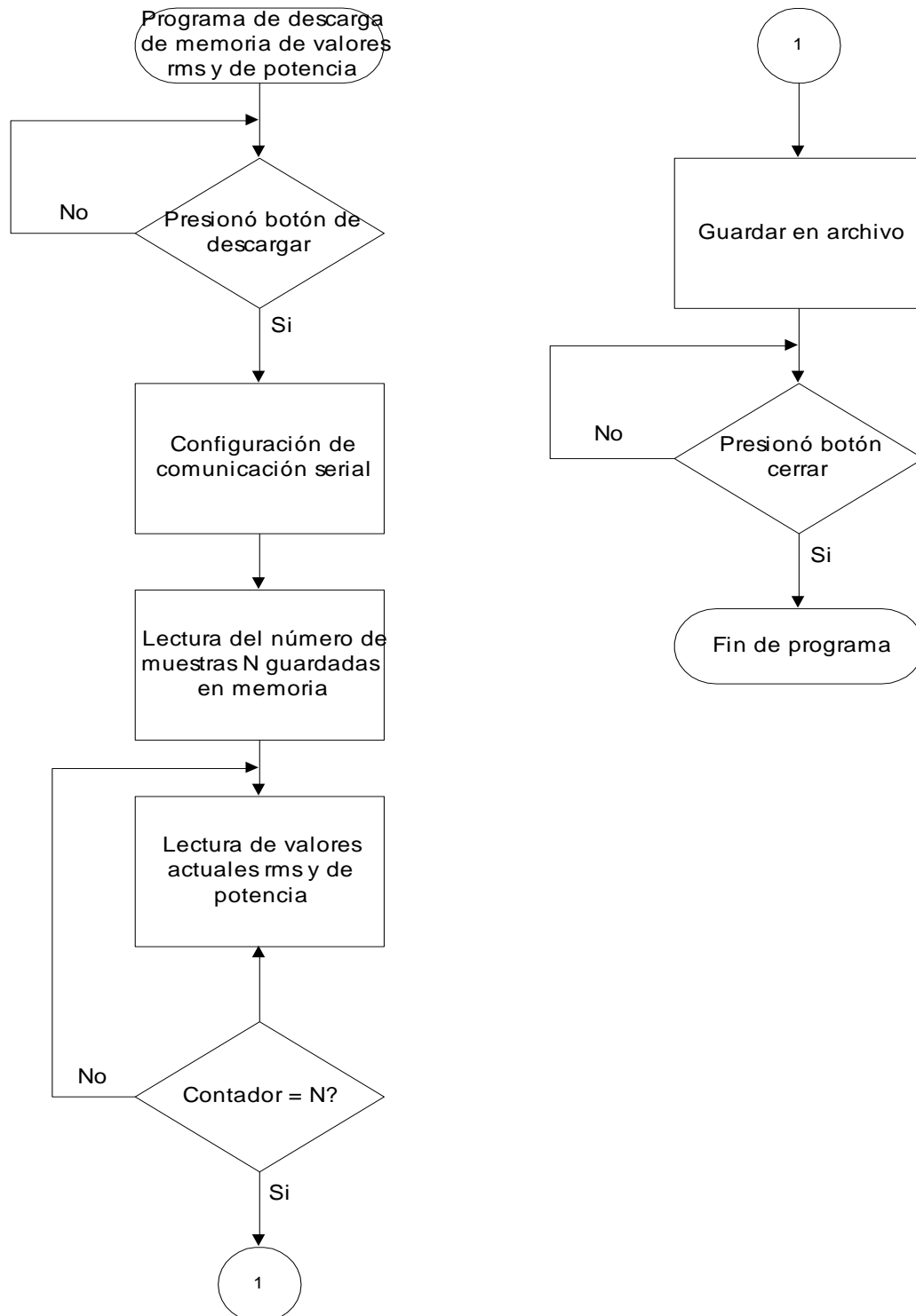


Figura 6.6. Flujograma de subrutina de descarga de valores rms y de potencia

Subrutina de generación de archivos de reportes HTML

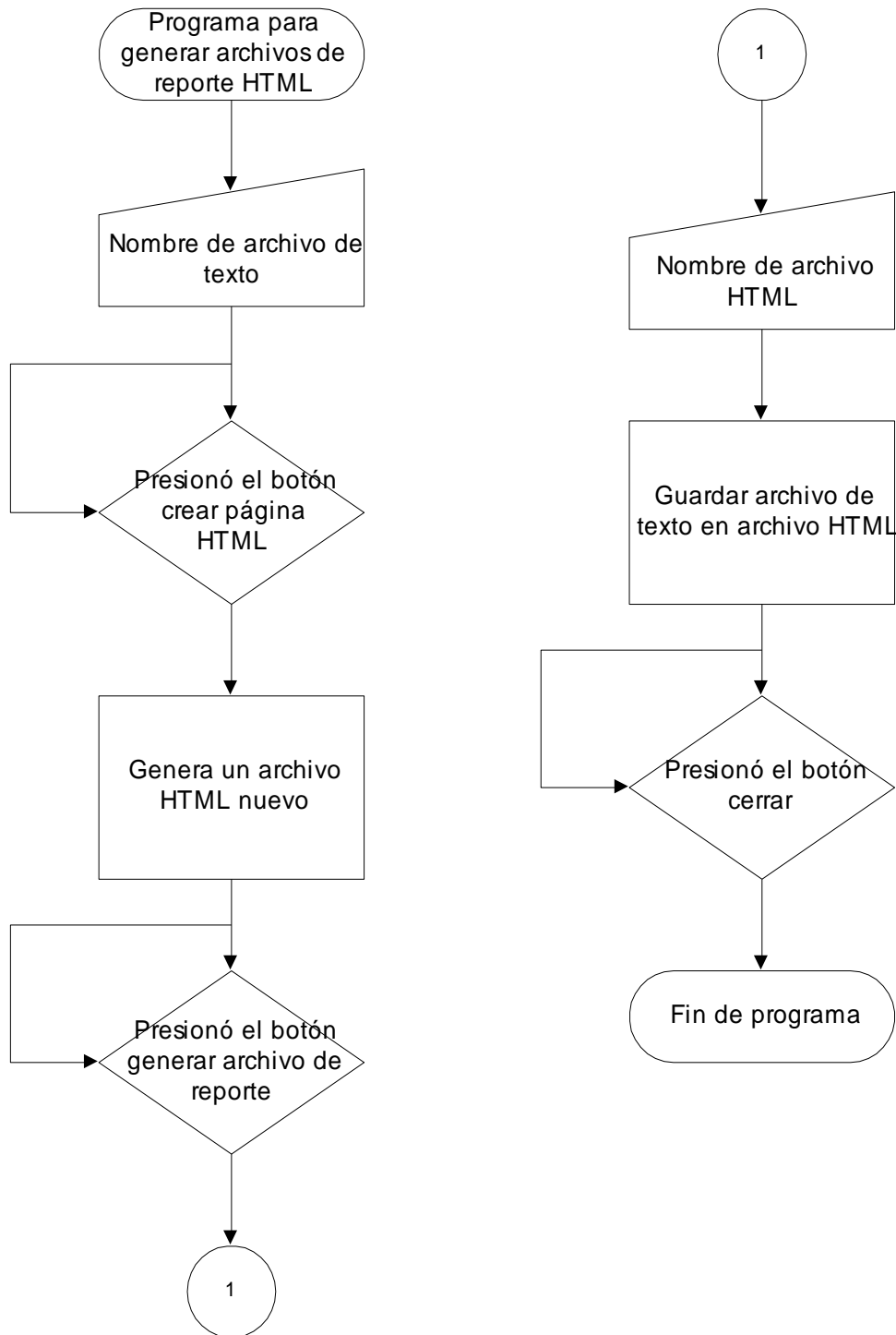


Figura 6.7. Flujograma de subrutina de generación de reportes HTML

6.2.2 CALIBRACION

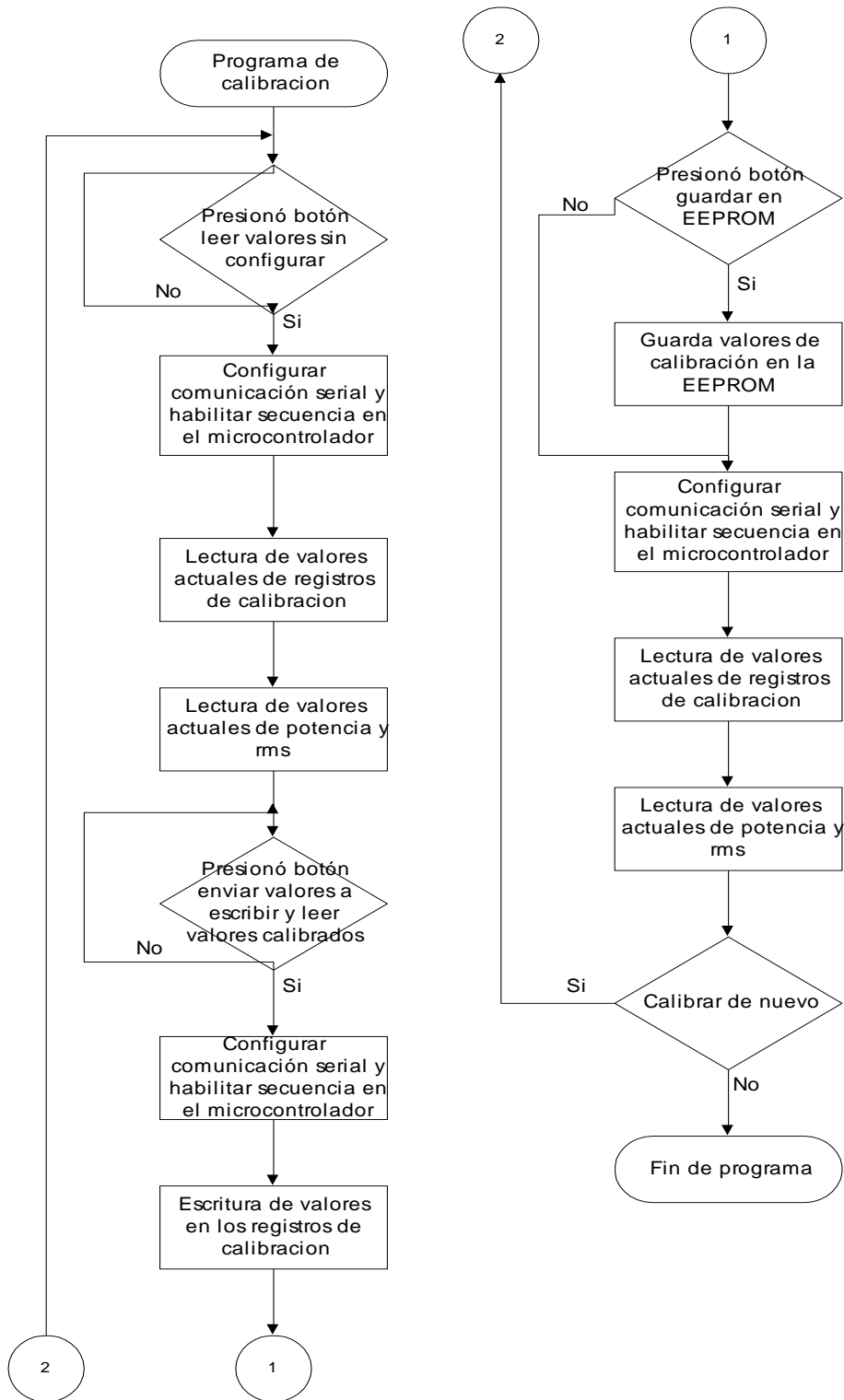


Figura 6.8. Flujograma de subrutina de calibración

6.3 LA INTERFASE GRÁFICA DEL USUARIO

La interfase gráfica del usuario (GUI) es desarrollada bajo el lenguaje de programación de alto nivel LabVIEW.

Un VI consta de dos partes principales: un panel frontal y un diagrama de bloques. El panel frontal sirve como la interfase para el usuario, aquí se colocan los controles y las pantallas de salida de datos del sistema. El diagrama de bloques es la estructura del programa, para construir el VI se seleccionan los objetos y se utilizan los conectores para unirlos entre sí; en un lenguaje de programación convencional se escriben muchas líneas de código, en LabVIEW cada icono representa una función.

A continuación se mostrará vistas parciales de los VI que conforman la interfase gráfica del usuario, como:

- ❖ Programa principal
- ❖ Programa de calibración

La guía de operación de los VIs principales se encuentra en el Manual del Usuario, en el Anexo 1.

6.3.1 VI PRINCIPAL

Este VI es el principal, ya que en este se puede escoger la operación a realizar, introducir la relación de los transformadores de corriente (TI) y de los transformadores de tensión (TT), ajustar el reloj del medidor, acceder a la ayuda o cerrar la interfase gráfica.

Entre las operaciones a realizar están:

- ❖ Lectura actual del medidor de valores rms y de potencia
- ❖ Lectura de archivo de valores rms y de potencia entre dos fechas
- ❖ Lectura de archivo de valores rms y de potencia de una fecha especifica
- ❖ Descarga del medidor de valores rms y de tiempo
- ❖ Generación de archivos de reporte

MEDIDOR TRIFASICO MULTIFUNCION

PUERTO: COM2

RELACION DE TI: Sin TI

RELACION DE TT: Sin TT

OPERACION A REALIZAR: Lectura actual del medidor de valores rms y de potencia

AYUDA

AJUSTAR RELOJ

CARGAR REFERENCIAS

EJECUTAR

CERRAR

Nº PROCESOS: 0

Figura 6.9 VI Principal.

A continuación se muestra el panel frontal de los VIs más importantes que son llamados por el VI principal.

VI DE LECTURA ACTUAL DEL MEDIDOR DE VALORES RMS Y DE POTENCIA

Con este VI es posible obtener los valores actuales rms de cada uno de los seis canales analógicos, los tres de voltaje y los tres de corriente; además de los valores actuales de potencia activa, aparente y reactiva del medidor.

LECTURA ACTUAL DEL MEDIDOR DE VALORES RMS Y DE POTENCIA

LEER DE NUEVO CERRAR

Hora y fecha

P3F S3F Q3F FP 3F

VA VB VC

IA IB IC

Figura 6.10 Lectura actual del medidor de valores rms y de potencia.

VI DE LECTURA DE ARCHIVO DE VALORES RMS Y DE POTENCIA DE UNA FECHA ESPECIFICA

Con este VI es posible observar el comportamiento de la carga medida en una fecha específica. Se introduce el nombre del archivo, obtenido previamente en una descarga de memoria, se introduce la fecha, e inmediatamente se despliegan los valores rms y de potencia de la fecha introducida.

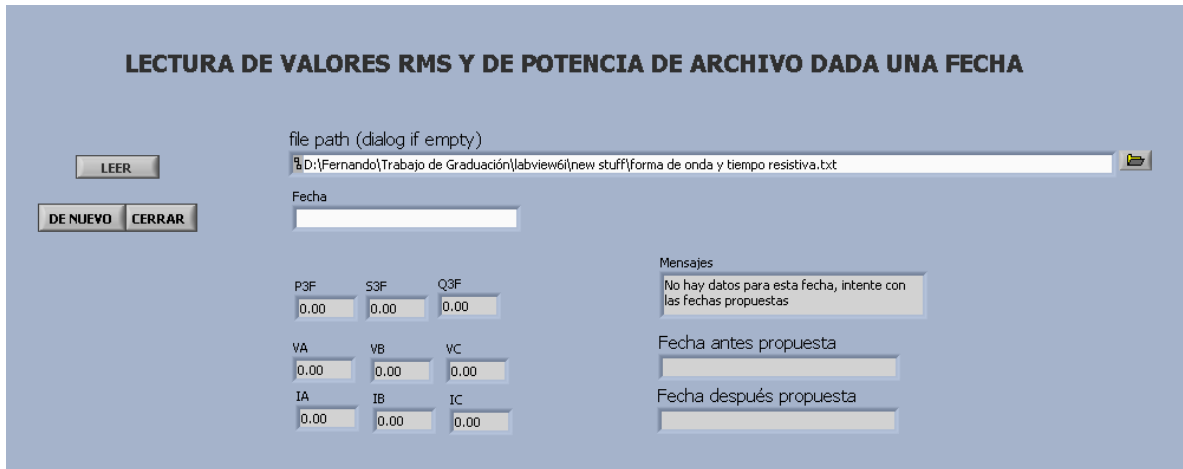


Figura 6.11 Lectura de archivo de valores rms y de potencia de una fecha específica.

VI DE LECTURA DE ARCHIVO DE VALORES RMS Y DE POTENCIA ENTRE DOS FECHAS

Este VI es similar al anterior, porque sirve para observar el comportamiento de la carga, pero a diferencia del anterior, este VI permite observar el comportamiento de la carga entre dos fechas; desplegando en una gráfica valores rms y de potencia por separado, así como la opción de guardar los datos obtenidos entre estas dos fechas.

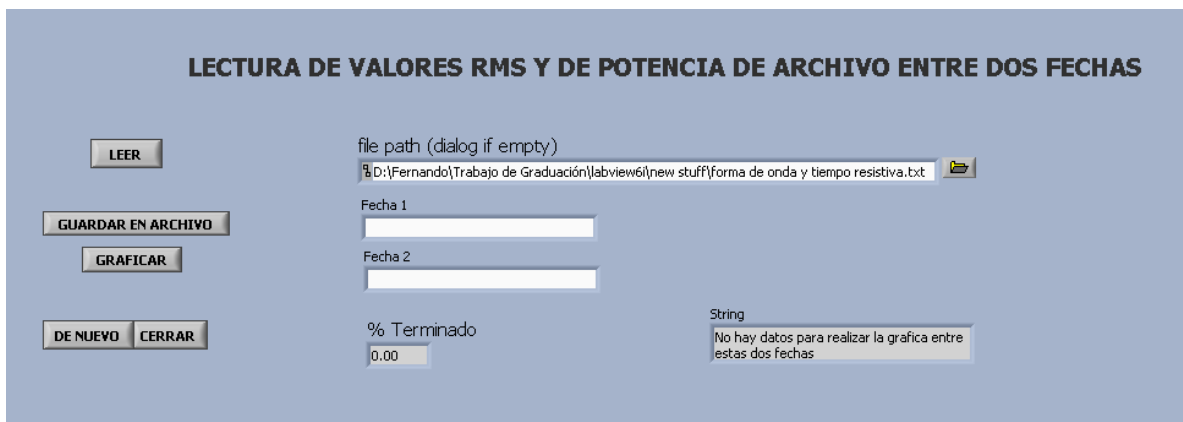


Figura 6.12 Lectura de archivo de valores rms y de potencia entre dos fechas.

VI DE DESCARGA DEL MEDIDOR DE VALORES RMS Y DE TIEMPO

Con este VI es posible descargar el contenido de la memoria utilizada, de 128Kx8 bits disponibles, en un archivo de texto. Posee una pantalla de salida que muestra el %Terminado.

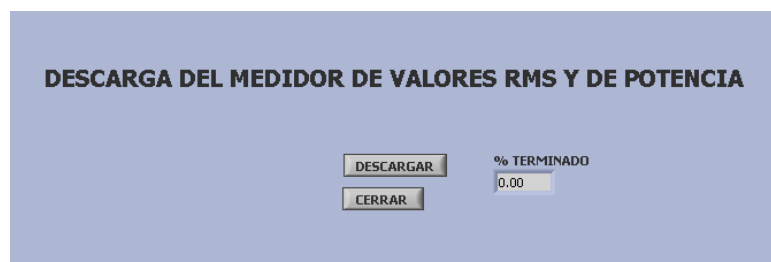


Figura 6.13 Descarga del medidor de valores rms y de tiempo.

VI DE GENERACIÓN DE ARCHIVOS DE REPORTE HTML

Con este VI es posible crear archivos de reporte HTML, a partir de un archivo de texto previamente descargado de memoria.

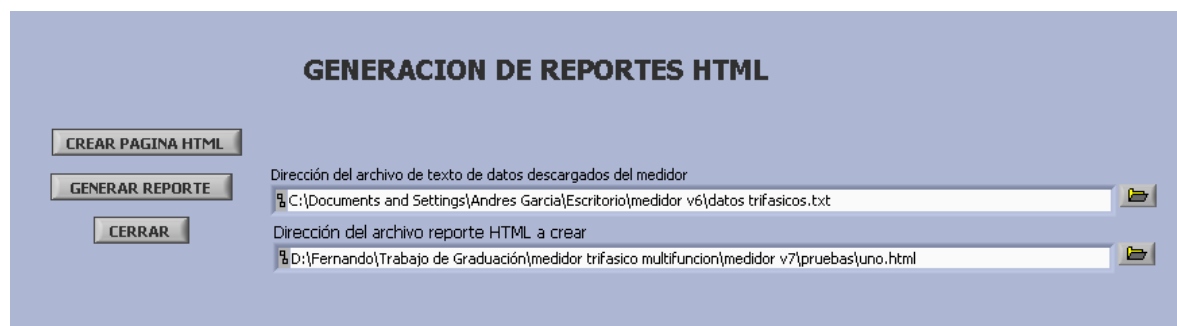


Figura 6.14 Generación de archivos de reporte.

6.3.2 VI DE CALIBRACION

Este VI es muy importante, porque es posible calibrar el medidor mediante la escritura de sus registros a través de comunicación serial.

Es posible regular los registros de ganancia de la potencia activa, la ganancia de voltaje y corriente, el offset del voltaje y corriente, y además ajustar la constante del medidor. Se tiene la opción de registrar estos cambios en la memoria EEPROM del medidor, a fin de mantener los cambios de manera permanente.

CALIBRACION DEL MEDIDOR TRIFASICO MULTIFUN

REGISTROS DE CALIBRACION				LEER VALORES SIN CALIBRAR	ENVIAR VALORES A ESCRIBIR Y LEER VALORES CALIBRADOS	ESCRIBIR A EEPROM	CALIBRAR DE NUEVO	CERRAR
GANANCIA DE POTENCIA ACTIVA	VALOR SIN CALIBRAR	VALOR A ESCRIBIR	VALOR CALIBRADO					
	AWG 0.00	AWG 0.00	AWG 0.00					
	BWG 0.00	BWG 0.00	BWG 0.00					
GANANCIA DE POTENCIA APARENTE	AVAG 0.00	AVAG 0.00	AVAG 0.00					
	BVAG 0.00	BVAG 0.00	BVAG 0.00					
	CVAG 0.00	CVAG 0.00	CVAG 0.00					
ATRIESTE DE FASE	APHCAL 0.00	APHCAL 0.00	APHCAL 0.00					
	BPHCAL 0.00	BPHCAL 0.00	BPHCAL 0.00					
	CPHCAL 0.00	CPHCAL 0.00	CPHCAL 0.00					
OFFSET DE POTENCIA ACTIVA	AAPOS 0.00	AAPOS 0.00	AAPOS 0.00					
	BAPOS 0.00	BAPOS 0.00	BAPOS 0.00					
	CAPOS 0.00	CAPOS 0.00	CAPOS 0.00					
NUMERO DE MEDIOS CICLOS DE LINEA	LINCYC 0.00	LINCYC 0.00	LINCYC 0.00					
	CFNUM 0.00	CFNUM 0.00	CFNUM 0.00					
	CFDEN 0.00	CFDEN 0.00	CFDEN 0.00					
				REGISTROS DE CALIBRACION				
OFFSET DE CORRIENTE RMS	VALOR SIN CALIBRAR	VALOR A ESCRIBIR						
	AIRMSOS 0.00	AIRMSOS 0.00						
	BIRMSOS 0.00	BIRMSOS 0.00						
OFFSET DE VOLTAJE RMS	AVRMSOS 0.00	AVRMSOS 0.00						
	BVRMSOS 0.00	BVRMSOS 0.00						
	CVRMSOS 0.00	CVRMSOS 0.00						
GANANCIA DE CORRIENTE RMS	AAPGAIN 0.00	AAPGAIN 0.00						
	DAPGAIN 0.00	DAPGAIN 0.00						
	CAPGAIN 0.00	CAPGAIN 0.00						
GANANCIA DE VOLTAJE RMS	AVGAIN 0.00	AVGAIN 0.00						
	BVGAIN 0.00	BVGAIN 0.00						
	CVGAIN 0.00	CVGAIN 0.00						

Figura 6.15 VI de calibración.

7.0 CALIBRACIÓN DEL MEDIDOR

7.1 FUNDAMENTOS TEÓRICOS DEL IC ADE7754

FILTROS ANTIALIASING

Este filtro análogo paso-bajo RC, es usado para prevenir ALIASING. Este filtro es usado para atenuar las frecuencias altas a 900KHz, y así evitar distorsión en la banda de interés de medición de 40Hz a 2KHz. Para ello se utiliza un filtro RC simple de un polo, con una frecuencia de corte de 10 KHz produce una atenuación de aproximadamente 40dB a 833KHz. Cada canal de voltaje y de corriente posee un filtro ANTIALIASING.

COMPENSACIÓN DE FASE

Los transductores de corriente pueden introducir errores de fase de 0.1° a 0.3° , estos errores son particularmente notables en bajos factores de potencia. El ADE7754 permite un pequeño retardo o adelanto de tiempo, utilizando los registros de compensación de fase, para compensar pequeños errores de fase en el rango de 0.1° a 0.5° , ó de $-19.2\mu\text{S}$ a $19.2\mu\text{S}$. Cuando se utilizan los registros de compensación de fase, la respuesta de fase es casi cero de 45Hz a 1KHz, esto es lo necesario requerido en aplicaciones típicas de medición de energía.

7.2 FORMULAS UTILIZADAS PARA LA CALIBRACIÓN DEL MEDIDOR

El IC ADE7754 en su hoja técnica, así como en su nota de aplicación AN-624 proporciona fórmulas y procedimientos para obtener los valores del pulso de frecuencia CF, de magnitudes de voltaje, corriente como de potencia.

Es importante destacar que los valores obtenidos en los registros son números en complemento a 2, de diferentes tamaños. Que en su mayoría son números que pueden alcanzar un máximo de 24 bits con o sin signo, según sea la configuración de los registros de calibración. Para energía activa y reactiva que corresponde al registro LAENERGY son números de 24 bits con signo; para la energía aparente que corresponde al registro LVAENERGY son números de 24 bits sin signo; para los valores de voltaje y corriente que son los registros VRMS y IRMS por fase, son números de 24 bits.

7.2.1 CALIBRACIÓN DE GANANCIA DE ENERGÍA ACTIVA USANDO EL REGISTRO LAENERGY

Se tienen las siguientes fórmulas:

$$CF(Hz) = \frac{LAENERGY}{4 * TiempodeAcumulacion} * \frac{CFNUM}{CFDEN} * \left(1 + \frac{WG}{2^{12}}\right) \quad \text{Ecuación 7.1}$$

$$TiempodeAcumulacion(s) = \frac{LINCYC}{2 * Frecuenciadelinea * Numerodefases} \quad \text{Ecuación 7.2}$$

$$Frecuenciadelinea(Hz) = \frac{1}{PERIOD * 2.4\mu s} \quad \text{Ecuación 7.3}$$

Donde:

CF, es el pulso de frecuencia proporcional a la potencia
 LAENERGY, es el registro de energía activa
 CFNUM, es el numerador del registro de CF
 CFDEN, es el denominador del registro de CF
 WG, es el registro de ganancia de potencia activa para la fase dada
 LINCYC, es el número de medios ciclos que acumula LAENERGY
 PERIOD, es el período de la línea

7.2.2 CALIBRACIÓN DE LA FASE USANDO EL REGISTRO LAENERGY

El acoplamiento de fase entre las entradas de corriente y voltaje de una fase es un tema crítico. Un error de fase tan pequeño como 0.5° causa un error de 1.5% en la medición de potencia activa. Una medición de dos puntos es necesaria.

$$Error = \frac{LAENERGY(FP = 0.5) - \frac{LAENERGY(FP = 1)}{2}}{\frac{LAENERGY(FP = 1)}{2}} \quad \text{Ecuación 7.4}$$

FP = 0.5 inductivo FP = 0.5 capacitivo la ecuación cambia de signo

$$PhaseError(^{\circ}) = -\arcsin\left(\frac{Error}{\sqrt{3}}\right) \quad \text{Ecuación 7.5}$$

$$PHCAL Register = \frac{PhaseError(^{\circ}) * PERIOD Register * 2.4 \mu s}{360^{\circ} * 1.2 \mu s} \quad \text{Ecuación 7.6}$$

El ajuste del registro PHCAL es de ±0.34 a 50 Hz, y de ±0.41 a 60 Hz.

Si se requieren correcciones muy grandes, se debe utilizar componentes pasivos externos. El atraso generado por el filtro antialiasing conformado por la resistencia de 1KΩ y el capacitor de 0.33μF es de 0.56° para 50 Hz.

7.2.3 COMO OBTENER EL VALOR DE POTENCIA ACTIVA PROMEDIO

Se utiliza la siguiente fórmula para obtener el valor en Watt de la potencia activa promedio:

$$PotenciaActivaPromedio(W) = LAENERGYactual * \frac{Wcalibración}{LAENERGYcalibración} \text{ Ecuación 7.7}$$

Donde:

$LAENERGYactual$, es el registro de energía activa obtenido por el medidor en una medición.

$Wcalibración$, es la potencia en Watt conocida inyectada durante el procedimiento de calibración.

$LAENERGYcalibración$, es el registro de energía activa obtenido después de calibrar el medidor.

7.2.4 COMO OBTENER EL VALOR DE POTENCIA APARENTE PROMEDIO

Se utiliza la siguiente fórmula para obtener el valor en VA de la potencia aparente promedio:

$$PotenciaAparentePromedio(VA) = LVAENERGYactual * \frac{VAcalibración}{LVAENERGYcalibración}$$

Ecuación 7.8

Donde:

$LVAENERGYactual$, es el registro de energía aparente obtenido por el medidor en una medición.

$VAcalibración$, es la potencia en VA conocida inyectada durante el procedimiento de calibración.

$LVAENERGYcalibración$, es el registro de energía aparente obtenido después de calibrar el medidor.

7.2.5 COMO OBTENER EL VALOR DE VOLTAJE RMS

Se utiliza la siguiente fórmula para obtener el valor en V del voltaje rms:

$$\text{Voltaje}_{rms}(V) = VRMS_{actual} * \frac{V_{calibración}}{VRMS_{calibración}} \quad \text{Ecuación 7.9}$$

Donde:

$VRMS_{actual}$, es el registro de voltaje rms obtenido por el medidor en una medición.

$V_{calibración}$, es el voltaje en V conocido inyectada durante el procedimiento de calibración.

$VRMS_{calibración}$, es el registro de voltaje rms obtenido después de calibrar el medidor.

7.2.6 COMO OBTENER EL VALOR DE CORRIENTE RMS

Se utiliza la siguiente fórmula para obtener el valor en A de la corriente rms:

$$\text{Corriente}(A) = IRMS_{actual} * \frac{I_{calibración}}{IRMS_{calibración}} \quad \text{Ecuación 7.10}$$

Donde:

$IRMS_{actual}$, es el registro de corriente rms obtenido por el medidor en una medición.

$I_{calibración}$, es la corriente en A conocida inyectada durante el procedimiento de calibración.

$IRMS_{calibración}$, es el registro de corriente rms obtenido después de calibrar el medidor.

7.3 TIPOS DE CONEXIONES DISPONIBLES PARA EL IC ADE7754

Los tipos de conexiones disponible según la ANSI C12.10 para el ADE7754 son:

Numero de hilos	Tipo de conexión	Tipo de base	Número de elementos	Fórmula
3 hilos	Delta	5S/13S	3 elementos	$P = V_A * I_A + V_B * I_B + V_C * I_C$
4 hilos	Estrella	6S/14S	2 ½ elementos	$P = V_A * (I_A - I_B) + V_C * (I_C - I_B)$
4 hilos	Delta	8S/15S	2 ½ elementos	$P = V_A * (I_A - I_B) + V_C * I_C$
4 hilos	Estrella	9S/16S	3 elementos	$P = V_A * I_A + V_B * I_B + V_C * I_C$

Tabla 7.1. Tipo de conexión para potencia activa.

Numero de hilos	Tipo de conexión	Tipo de base	Número de elementos	Fórmula
3 hilos	Delta	5S/13S	3 elementos	$S = V_{Arms} * I_{Arms} + V_{Brms} * I_{Brms} + V_{Crms} * I_{Crms}$
4 hilos	Estrella	6S/14S	2 ½ elementos	$S = V_{Arms} * I_{Arms} + (V_{Arms} + V_{Crms}) / 2 * I_{Brms} + V_{Crms} * I_{Crms}$
4 hilos	Delta	8S/15S	2 ½ elementos	$S = V_{Arms} * I_{Arms} + V_{Brms} * I_{Brms} + V_{Crms} * I_{Crms}$
4 hilos	Estrella	9S/16S	3 elementos	$S = V_{Arms} * I_{Arms} + V_{Brms} * I_{Brms} + V_{Crms} * I_{Crms}$

Tabla 7.2. Tipo de conexión para potencia aparente.

7.4 OBTENCIÓN DE LA CONSTANTE DEL MEDIDOR

La constante del medidor está directamente relacionada con la frecuencia del pin CF, es decir, es necesario tomar una referencia para el valor de pulsos por segundo que tendrá este pin a una determinada potencia aplicada.

Para este caso, se aplicarán los siguientes valores:

Parámetro	Valor
Voltaje aplicado	120 V
Corriente máxima	5 A
Número de fases	3
Tiempo de acumulación	20 minutos
Numero de pulsos	3000
Frecuencia de la línea	60 Hz

Tabla 7.3. Parámetros para obtención de constante del medidor

Se aplicarán las siguientes fórmulas:

$$Wh_{MAXIMO}(W) = Voltaje * Corriente * NumerodeFases$$

$$PW \left(\frac{Wh}{pulso} \right) = \frac{WMAXIMO}{\frac{NumerodePulsos}{int\ intervalo} * \frac{\#int\ ervalos}{hora}}$$

$$TiempoporPulso \left(\frac{pulso}{s} \right) = \frac{NumerodePulsos}{TiempodeAcumulacionenSegundos}$$

Tomando los valores dados, tenemos:

$$Wh_{MAXIMO}(W) = Voltaje * Corriente * NumerodeFases = 120V * 5A * 3fases = 1800W$$

$$PW \left(\frac{Wh}{pulso} \right) = \frac{WMAXIMO}{\frac{NumerodePulsos}{int\ intervalo} * \frac{\#int\ ervalos}{hora}} = \frac{1800}{3000 * 3} = 0.2 \frac{Wh}{pulso}$$

$$Frecuencia\ de\ CF\left(\frac{pulso}{s}\right) = \frac{NumerodePulsos}{TiempodeAcumulacionenSegundos} = \frac{3000}{20\ min\ utos * \frac{60segundos}{1\ min\ uto}}$$

$$Frecuencia\ de\ CF\left(\frac{pulso}{s}\right) = 2.5 \frac{pulso}{s} = 2.5\ Hz$$

Es decir, realiza 2.5 pulsos en un segundo, ó cada pulso toma 0.4 segundos. Esta frecuencia será conocida como la frecuencia de diseño, es decir, cuando el medidor sea calibrado se esperará que el pulso de salida de frecuencia sea de 2.5 Hz a un FP = 1, a V =120 y I =5 A en sus tres fases.

7.5 PROCEDIMIENTO DE CALIBRACIÓN

Para ilustrar el procedimiento de calibración, se utilizará parte del procedimiento efectuado según la hoja de aplicación AN-624 del ADE7754. Las magnitudes de voltaje y corriente fueron inyectadas al medidor, por el patrón de calibración, la maleta WECO PORTÁTIL de la marca WATTHOUR ENGINEERING CO., INC, número serie del patrón 9001007. El procedimiento de calibración fue efectuado en el laboratorio de metrología de la Empresa Transmisora de El Salvador ETESAL.

VALORES NOMINALES PARA CALIBRACIÓN

Parámetro	Valor
Corriente máxima	5 A
Corriente mínima	0.25 A
Voltaje de prueba	120 V
Frecuencia	60 Hz
Constante del medidor	0.2 Wh / pulso
Número de medios ciclos de línea	20
Frecuencia esperada en CF	2.5 Hz

Tabla 7.4. Valores nominales para calibración

7.5.1 AJUSTE DEL PULSO DE FRECUENCIA CF UTILIZANDO EL REGISTRO DE ENERGÍA ACTIVA LAENERGY

Se procede a ajustar el pulso de frecuencia CF al valor de 2.5 Hz, que es la frecuencia de diseño, dado que la constante del medidor es de 0.2 Wh / pulso.

El procedimiento de calibración se realiza utilizando el VI de calibración, para ajustar los valores del medidor.

Para ello se utilizarán las siguientes fórmulas:

$$CF_{Trifásico}(Hz) = \frac{LAENERGY_{Trifásico}}{4 * TiempodeAcumulacion} * \frac{CFNUM}{CFDEN}$$

$$TiempodeAcumulacion(s) = \frac{LINCYC}{2 * Frecuencia delinea * Numerodefases}$$

$$Frecuencia delinea(Hz) = \frac{1}{PERIOD * 2.4\mu s}$$

Se obtuvo un valor de PERIOD = 6944, con un LINCYC = 20 y número de fases = 1. A los registros de ganancia de energía activa, y los de ganancia de corriente, AWG, BWG, CWG, AAPGAIN, BAPGAIN y CAPGAIN, respectivamente; se les asigna el valor de -2047. Se obtiene un valor de LAENERGY = 6064. El registro CFNUM = 0 y CFDEN = 0, esto hace que en la fórmula se conviertan en 1.

$$Frecuencia delinea(Hz) = \frac{1}{6944 * 2.4\mu s} = 60.0038Hz$$

$$TiempodeAcumulacion(s) = \frac{20}{2 * 60 * 1} = 0.167segundos$$

$$CF_{Trifásico}(Hz) = \frac{6064}{4 * 0.167} * \frac{1}{1} = 9096Hz$$

Como se tiene que la frecuencia deseada es de 2.5 Hz, se calcula un CFDEN = 3638 y CFNUM = 0.

$$CF_{Trifásico}(Hz) = CF_{inicial} * \frac{CFNUM}{CFDEN} = 9096 * \frac{1}{3638} = 2.5003Hz$$

Se procedió a utilizar un Multímetro FLUKE 867 para medir la frecuencia, y se obtuvo $CF_{Frecuencímetro}(Hz) = 2.531$.

$$Error = \frac{CF_{Deseada}(Hz) - CF_{Frecuencímetro}(Hz)}{CF_{Deseada}(Hz)} * 100\% = \frac{2.5 - 2.531}{2.5} * 100\% = 1.24\%$$

Se procede a hacerle una prueba con el patrón de calibración WECO PORTATIL, para obtener el error del medidor.

TIPO DE PRUEBA	SEÑALES DE ENTRADA			NUMERO DE PULSOS	RESULTADO OBTENIDO				REGISTROS	
	VOLTAJE (voltios)	CORRIENTE (amperios)	FACTOR DE POTENCIA		PRUEBA 1 (%)	PRUEBA 2 (%)	PRUEBA 3 (%)	PROMEDIO PRUEBAS	REQUERIDO (%)	DESVIACION (%)
CARGA ALTA	120	5.00	0 °	10	102.29	102.27	102.33	102.298	100.000	2.298
FACTOR DE POTENCIA	120	5.00	60 °	10	105.72	105.68	105.77	105.724	100.000	5.724
FACTOR DE POTENCIA	120	5.00	- 36 °	10	100.92	100.86	100.83	100.869	100.000	0.869
FACTOR DE POTENCIA	120	2.50	+ 30 °	10	103.51	103.55	103.43	103.497	100.000	3.497
CARGA ALTA	120	2.50	0 °	10	102.13	102.25	102.15	102.174	100.000	2.174
FACTOR DE POTENCIA	120	2.50	60 °	10	106.16	101.36	106.28	104.595	100.000	4.595
CARGA BAJA	120	0.50	0 °	5	101.27	106.28	101.38	102.973	100.000	2.973
CARGA BAJA	120	0.25	0 °	2	100.76	101.14	100.39	100.762	100.000	0.762
PROMEDIOS								102.862	100.000	2.862

Tabla 7.5. Cálculo del error del medidor por el patrón de calibración WECO PORTATIL

Se puede observar que para factores de potencia bajos, el error del medidor aumenta; por lo que se ajustará el registro de calibración fase para cada una de las fases.

7.5.2 CALIBRACIÓN DE LA FASE USANDO EL REGISTRO LAENERGY

Primero se calibra la fase A:

$$Error = \frac{LAENERGY(FP = 0.5) - \frac{LAENERGY(FP = 1)}{2}}{\frac{LAENERGY(FP = 1)}{2}} = \frac{1054 - \frac{2051}{2}}{\frac{2051}{2}} = 0.0278$$

$$PhaseError(^{\circ}) = -\sin^{-1}\left(\frac{Error}{\sqrt{3}}\right) = -\sin^{-1}\left(\frac{0.0278}{\sqrt{3}}\right) = -0.0161^{\circ}$$

$$PHCAL\ Register = \frac{PhaseError(^{\circ}) * PERIOD\ Register * 2.4\ \mu s}{360^{\circ} * 1.2\ \mu s} = \frac{-0.0161^{\circ} * 6944 * 2.4\ \mu s}{360^{\circ} * 1.2\ \mu s}$$

$$PHCAL\ Register = -0.6211 \cong -1$$

Luego se calibra la fase B:

$$Error = \frac{LAENERGY(FP = 0.5) - \frac{LAENERGY(FP = 1)}{2}}{\frac{LAENERGY(FP = 1)}{2}} = \frac{1055 - \frac{2071}{2}}{\frac{2071}{2}} = 0.0188$$

$$PhaseError(^{\circ}) = -\sin^{-1}\left(\frac{Error}{\sqrt{3}}\right) = -\sin^{-1}\left(\frac{0.0188}{\sqrt{3}}\right) = -0.0109^{\circ}$$

$$PHCAL\ Register = \frac{PhaseError(^{\circ}) * PERIOD\ Register * 2.4\ \mu s}{360^{\circ} * 1.2\ \mu s} = \frac{-0.0109^{\circ} * 6944 * 2.4\ \mu s}{360^{\circ} * 1.2\ \mu s}$$

$$PHCAL\ Register = -0.4205 \cong 0$$

Luego se calibra la fase C:

$$Error = \frac{LAENERGY(FP = 0.5) - \frac{LAENERGY(FP = 1)}{2}}{\frac{LAENERGY(FP = 1)}{2}} = \frac{1045 - \frac{2056}{2}}{\frac{2056}{2}} = 0.0165$$

$$PhaseError(^{\circ}) = -\sin^{-1}\left(\frac{Error}{\sqrt{3}}\right) = -\sin^{-1}\left(\frac{0.0165}{\sqrt{3}}\right) = -0.0095^{\circ}$$

$$PHCAL Register = \frac{PhaseError(^{\circ}) * PERIOD Register * 2.4 \mu s}{360^{\circ} * 1.2 \mu s} = \frac{-0.0095^{\circ} * 6944 * 2.4 \mu s}{360^{\circ} * 1.2 \mu s}$$

$$PHCAL Register = -0.3665 \cong 0$$

Después de obtener valores preliminares para los registros de calibración de fase, se obtuvieron los mejores resultados con el valor de -15 en el registro de calibración de fase para cada una de las fases.

7.5.3 AJUSTE FINO DEL CFDEN

Se desea obtener un error de al menos 2%, por lo que se procede a hacer un barrido sobre el registro CFDEN sobre el valor de 3638.

CF	2.46	2.45	2.48	2.48	2.48	2.48
CFDEN	3750	3770	3710	3720	3715	3725
SFL	99.29	98.79	100.34	100.09	100.26	99.97
SLL	98.3	98.1	99.37	98.74	99.2	98.74
SPF	101.51	100.96	102.59	102.33	102.44	102.17
SPF	98.37	97.89	99.48	99.21	99.31	99.03
PROM	-0.6325	-1.065	0.445	0.0925	0.3025	-0.0225

Tabla 7.6. Barrido sobre el registro CFDEN, prueba 1

CF	2.49	2.49	2.49	2.48	2.49	2.49
CFDEN	3700	3705	3708	3711	3710	3710
SFL	100.65	100.49	100.45	100.33	100.38	100.39
SLL	99.08	99.09	99.19	99.56	99.48	99.49
SPF	102.79	102.73	102.58	102.46	102.46	102.59
SPF	99.72	99.6	99.5	99.43	99.43	99.43
PROM	0.56	0.4775	0.43	0.445	0.4375	0.475

Tabla 7.7. Barrido sobre el registro CFDEN, prueba 2

Al observar el error obtenido de las lecturas respecto al barrido del CFDEN, se tiene que los mejores resultados se obtienen para CFDEN = 3710.

Después de ajustar la fase y del ajuste fino del CFDEN, se realiza otra prueba del error de medición utilizando el patrón de calibración WECO PORTATIL.

TIPO DE PRUEBA	SEÑALES DE ENTRADA			NUMERO DE PULSOS	RESULTADO OBTENIDO				REGISTROS	
	VOLTAJE (voltios)	CORRIENTE (amperios)	FACTOR DE POTENCIA		PRUEBA 1 (%)	PRUEBA 2 (%)	PRUEBA 3 (%)	PROMEDIO PRUEBAS	REQUERIDO (%)	DESVIACION (%)
CARGA ALTA	120	5.00	0 °	10	100.47	100.47	100.51	100.483	100.000	0.483
FACTOR DE POTENCIA	120	5.00	60 °	10	102.69	102.66	102.61	102.653	100.000	2.653
FACTOR DE POTENCIA	120	5.00	- 36 °	10	99.60	99.63	99.59	99.607	100.000	(0.393)
FACTOR DE POTENCIA	120	2.50	+ 30 °	10	101.17	101.16	101.30	101.210	100.000	1.210
CARGA ALTA	120	2.50	0 °	10	100.30	100.22	100.30	100.273	100.000	0.273
FACTOR DE POTENCIA	120	2.50	60 °	10	103.21	103.21	103.18	103.200	100.000	3.200
CARGA BAJA	120	0.50	0 °	5	99.12	99.28	98.98	99.127	100.000	(0.873)
CARGA BAJA	120	0.25	0 °	2	99.76	99.43	98.26	99.150	100.000	(0.850)
PROMEDIOS								100.713	100.000	0.713

Tabla 7.8 Cálculo del error del medidor por el patrón de calibración WECO PORTATIL

7.5.4 GUARDANDO LAS REFERENCIAS DE MAGNITUDES Y DE REGISTROS PARA MEDIDICION

Después de calibrar el medidor se procede a guardar los valores de calibración para las magnitudes y los registros, que servirán por ejemplo, para obtener el valor de la potencia activa promedio a partir del registro LAENERGY en complemento a 2.

El valor guardado en el archivo de texto "referencias 3 febrero 12-59 etesal lincyc 200" contiene la siguiente información:

50718.000000	"registro LAENERGY P"
231.000000	"registro LVAENERGY S"
794634.000000	"registro VRMS A"
793015.000000	"registro VRMS B"
795522.000000	"registro VRMS C"
1076512.000000	"registro IRMS A"
1073617.000000	"registro IRMS B"
1069148.000000	"registro IRMS C"
1789.2	"potencia activa en Watt"
1789.2	"potencia aparente en VA"
120	"voltaje A en V"
120	"voltaje B en V"
120	"voltaje C en V"
4.97	"corriente A en A"
4.97	"corriente B en A"
4.97	"corriente C en A"

7.5.5 RESUMEN DE LA CALIBRACIÓN

Para los valores inyectados por el patrón de calibración WECO PORTATIL, de $V = 120$ V, $I = 5$ A para cada una de las fases, a un $FP = 1$. Con una constante del medidor $K_h = 0.2$ Wh / pulso, con una frecuencia de diseño de 2.5 Hz.

Con los valores guardados en la EEPROM del medidor, para los siguientes registros de calibración:

AWG = -2047

BWG = -2047

CWG = -2047

APHCAL=-15

BPHCAL=-15

CPHCAL=-15

AAPGAIN = -2047

BAPGAIN = -2047

CAPGAIN = -2047

LINCYC = 20

CFDEN = 3710

Y con el valor de 0 en los demás registros de calibración.

Se obtuvo un error del medidor de 0.713%, y una frecuencia de 2.49 Hz en el pulso de frecuencia CF.

7.6 RESULTADOS OBTENIDOS

El medidor se comparó respecto al patrón de calibración WECO PORTATIL, y se obtuvo los siguientes resultados.

	PRUEBA 1(Angulo 0°)			PRUEBA 2(Angulo 0°)		
	Valor verdadero	Valor medido	Error (%)	Valor verdadero	Valor medido	Error (%)
Va (volt.)	119.95	120.15	0.17	119.95	120.2	0.21
Vb (volt.)	119.95	120.46	0.43	119.95	120.35	0.33
Vc (volt.)	119.95	120.43	0.40	119.95	120.35	0.33
Ia (Ampt.)	4.98	4.96	-0.40	2.5	2.45	-2.00
Ib (Ampt.)	4.98	4.96	-0.40	2.5	2.49	-0.40
Ic (Ampt.)	4.98	4.96	-0.40	2.5	2.47	-1.20
Potencia activa (Watts)	1792.1	1791	-0.06	899.63	894	-0.63
Potencia aparente (VA)	1792.1	1787	-0.28	899.63	888	-1.29
Potencia reactiva (Var)	0	125		0	103	
factor de potencia	1	1	0	1	1.01	1

Tabla 7.9 Comparación del medidor con el patrón WECO PORTATIL prueba 1 y 2

	PRUEBA 3(Angulo 0°)			PRUEBA 4(Angulo 60°)		
	Valor verdadero	Valor medido	Error (%)	Valor verdadero	Valor medido	Error (%)
Va (volt.)	119.9	119.93	0.03	119.90	120	0.08
Vb (volt.)	119.9	119.97	0.06	119.90	120.02	0.10
Vc (volt.)	119.9	119.98	0.07	119.90	120.01	0.09
Ia (Ampt.)	0.5	0.48	-4.00	4.95	4.92	-0.61
Ib (Ampt.)	0.5	0.49	-2.00	4.95	4.99	0.81
Ic (Ampt.)	0.5	0.48	-4.00	4.95	4.96	0.20
Potencia activa (Watts)	179.85	176	-2.14	890.26	907	1.88
Potencia aparente (VA)	179.85	174	-3.25	1780.52	1783	0.14
Potencia reactiva (Var)	0	26		1541.97	1535	-0.45
factor de potencia	1	1.01	1.00	0.50	0.51	2.00

Tabla 7.10 Comparación del medidor con el patrón WECO PORTATIL prueba 3 y 4

	PRUEBA 5(Angulo -36°)			PRUEBA 6(Angulo 0°)		
	Valor verdadero	Valor medido	Error (%)	Valor verdadero	Valor medido	Error (%)
Va (volt.)	119.9	120	0.08	119.95	120.02	0.06
Vb (volt.)	119.9	120.03	0.11	119.95	120.36	0.34
Vc (volt.)	119.9	120.01	0.09	119.95	120.37	0.35
Ia (Ampt.)	4.95	4.93	-0.40	0.24	0.24	0.00
Ib (Ampt.)	4.95	4.98	0.61	0.24	0.24	0.00
Ic (Ampt.)	4.95	4.96	0.20	0.24	0.24	0.00
Potencia activa (Watts)	1440.47	1445	0.31	86.36	87	0.74
Potencia aparente (VA)	1780.51	1786	0.31	86.36	86	-0.42
Potencia reactiva (Var)	-1046.56	-1049	0.23	0	13	
factor de potencia	0.81	0.81	0	1	1.01	1.00

Tabla 7.11 Comparación del medidor con el patrón WECO PORTATIL prueba 5 y 6

	PRUEBA 7(Angulo 60°)			PRUEBA 8(Angulo 30°)		
	Valor verdadero	Valor medido	Error (%)	Valor verdadero	Valor medido	Error (%)
Va (volt.)	119.5	120.02	0.44	120	120.02	0.02
Vb (volt.)	119.5	120.06	0.47	120	120.09	0.08
Vc (volt.)	119.9	120.04	0.12	120	120.03	0.03
Ia (Ampt.)	2.48	2.57	3.63	2.45	2.46	0.41
Ib (Ampt.)	2.48	2.48	0.00	2.45	2.48	1.22
Ic (Ampt.)	2.48	2.46	-0.81	2.45	2.46	0.41
Potencia activa (Watts)	444.54	456	2.58	763.83	785	2.77
Potencia aparente (VA)	889.08	885	-0.46	882	901	2.15
Potencia reactiva (Var)	769.97	758.00	-1.55	441.00	444	0.68
factor de potencia	0.5	0.51	2	0.87	0.87	0

Tabla 7.12 Comparación del medidor con el patrón WECO PORTATIL prueba 7 y 8

RECOMENDACIONES

- ❖ Implementar un circuito de aislamiento desde los canales analógicos de voltaje y de corriente, para proteger de una falla en la red eléctrica, todos los componentes electrónicos que conforman el medidor.
- ❖ En la programación de los microcontroladores es conveniente utilizar macros con directivas que permitan decidir si el código fuente de un macro se agrega al código principal del microcontrolador o no.
- ❖ Utilizar archivos independientes para definir variables, constantes y macros; para depurar de manera más fácil el código fuente del programa principal.
- ❖ En la programación de la interfase gráfica del usuario en el lenguaje de alto nivel LabVIEW, es conveniente el uso de subVIs, debido a que por ser bloques autónomos que pueden formar parte de una aplicación; hace que el proceso de depuración de un programa principal sea más fácil, porque el código gráfico del programa principal se hace más pequeño, y para hacer modificaciones se realizan en el subVI de interés.

CONCLUSIONES

- ❖ Cuando se transmite información a través de una interfaz serial SPI, no basta con habilitar el módulo de comunicación del periférico con el que se desea establecer la comunicación; hay que tomar en cuenta además, que el periférico debe de estar listo para recibir, cada vez que transmite el microcontrolador maestro, ya que si esto no se cumple los bytes enviados por el maestro se perderán o no serán leídos por el periférico. Si el periférico desea enviar uno o mas bytes al microcontrolador maestro, el periférico debe de esperar a que este inicie una transmisión de datos.
- ❖ Cuando se desea ajustar el pulso de frecuencia CF al valor de frecuencia de diseño, se hace uso de la ecuación 7.1 para obtener el valor calculado de CFDEN, responsable de la exactitud del pulso de frecuencia CF; el valor calculado para CFDEN sirve como referencia para efectuar un barrido cercano, con el fin de obtener un ajuste más fino del pulso de frecuencia CF.
- ❖ Debido a la necesidad de mantener energizado todo el sistema del medidor, se vio la necesidad de implementar un sistema de respaldo que mantuviese alimentado al medidor durante un tiempo determinado, cuando existan fallas en la red; para asegurar el buen funcionamiento del medidor, y principalmente los datos almacenados en memoria.
- ❖ Para establecer una sincronía de transmisión y recepción de datos, entre la interfase gráfica del usuario y el programa del microcontrolador, se implementó un conjunto de comandos que conforman parte del protocolo, para evitar pérdidas de datos o traslapes entre datos.

BIBLIOGRAFÍA Y FUENTES DE CONSULTA

BIBLIOGRAFÍA

- ❖ **MICROCONTROLADORES <<PIC>>. DISEÑO PRACTICO DE APLICACIONES.** Jose M^a. Angulo Usategui, Ignacio Angulo Martínez. McGraw-Hill interamericana de España. S.A.U. Primera edición. 1997 ISBN: 84-481-1238-5

- ❖ **DISEÑO Y ELABORACIÓN DE INSTRUMENTOS VIRTUALES DE MEDICION PARA EL LABORATORIO DE MAQUINAS ELECTRICAS.** Mauricio Amílcar Ayala Arévalo, Walter Omar Servano Alvarez. Trabajo de graduación de la Escuela de Ingeniería Eléctrica, de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador. 2002.

- ❖ **ANÁLISIS DE SISTEMAS DE POTENCIA.** John J. Grainger, William D. Stevenson Jr. . Capítulo: Conceptos Básicos. Editorial MacGraw-Hill. 1996.

- ❖ **QUANTUM MULTIFUNCTION METER HARDWARE INSTRUCTION MANUAL 1610.** Schlumberger Industries, Electricity Division. Effective November 1993.

OTRAS FUENTES DE CONSULTA

❖ <http://www.analog.com>

- Manual de datos del IC ADE7754. Polyphase Multifunction Energy Metering IC with Serial Port.
- Evaluation Board Documentation ADE7754 Energy metering IC EVAL-ADE7754EB. Preliminary Technical Data.
- AN-624 APPLICATION NOTE. Calibration of a 3-Phase Energy Meter Board on the ADE7754. Etienne Moulin.
- Calibrating the ADE7754 for Watt, RMS and VA measurements.
- Multi-functional Polyphase Energy Metering IC with Serial Port Interfase ADE7754. Energy Measurement Group Precision Converters (PRC) Division.

❖ <http://www.microchip.com>

- Manual de datos del PIC16F877. PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers. © 2001 Microchip Technology Inc.

❖ <http://www.beyondlogic.org>

- Página web muy completa sobre especificaciones y programación de diversos dispositivos.

❖ <http://www.hp.com>

- En esta página se pueden encontrar optoaisladores de alta velocidad HCPL2232.

❖ <http://www.crmagnetics.com>

- En esta página se puede encontrar información sobre transformadores de corriente.

A.1 MANUAL DEL USUARIO

MEDIDOR TRIFÁSICO MULTIFUNCIÓN

MANUAL DE INSTRUCCIONES DE HARDWARE Y SOFTWARE

FEBRERO 2005

TABLA DE CONTENIDO	PAG.
SECCIÓN 1: INFORMACIÓN GENERAL	4
1.1 Introducción	4
1.2 Descripción general	4
1.3 Especificaciones	5
1.3.1 Exactitud	5
1.3.2 Constante de pulso KH	5
1.3.3 Reloj de tiempo real	6
1.3.4 Memoria	6
1.3.5 Comunicación	6
1.3.6 Fuente de alimentación	6
1.3.7 Dimensiones	6
SECCIÓN 2: HARDWARE DEL MEDIDOR	7
2.1 Presentación del medidor	7
2.2 Tarjetas del medidor	10
2.2.1 Tarjeta del medidor ADE7754	10
2.2.2 Tarjeta del microcontrolador maestro y reloj	12
2.2.3 Tarjeta de la memoria SRAM	13
2.2.4 Tarjeta de la fuente de voltaje con batería de respaldo	14
2.3 Tipos de servicio de conexión	15
2.4 Configuración de los jumper	17
SECCIÓN 3: INSTRUCCIONES DE OPERACIÓN	19
SECCIÓN 4: SOFTWARE DEL MEDIDOR	22
4.1 VI de calibración	22
4.2 VI Principal	28
4.2.1 VI de lectura actual de potencia y de valores rms	31
4.2.2 VI de lectura de archivo de potencia y de valores rms para una fecha específica	32
4.2.3 VI de lectura de archivo de potencia y de valores rms entre dos fechas	33
4.2.4 VI de descarga de memoria	37
4.2.5 VI de generación de reportes HTML	38

4.2.6 VI de ayuda 39

SECCIÓN 5: RESOLUCIÓN DE PROBLEMAS 41

SECCIÓN 6: CIRCUITOS ESQUEMÁTICOS 43

SECCIÓN 1: INFORMACIÓN GENERAL

1.1 INTRODUCCIÓN

Este manual de instrucciones contiene instrucciones de instalación y operación para el medidor trifásico multifunción. Este manual contiene las especificaciones técnicas del medidor, así como sus características generales.

Contiene una descripción detallada del hardware del medidor, los tipos de servicio de conexión que se pueden realizar, así como la configuración de los jumper.

Se provee una guía de instrucciones de operación necesarias para poner en funcionamiento el medidor; así como, los pasos detallados para utilizar el software de aplicación.

Se pone a la disposición una sección de resolución de problemas, y una sección de circuitos esquemáticos.

1.2 DESCRIPCIÓN GENERAL

El medidor trifásico multifunción es un medidor electrónico de estado sólido, que mide parámetros eléctricos como potencia activa, potencia aparente, potencia reactiva, así como el valor rms de voltaje y corriente por fase. Ver Figura A.1.1. y Figura A.1.2.

El medidor contiene una unidad de control implementada con un microcontrolador, la cual es responsable de efectuar el protocolo de comunicación de datos con la computadora, y de llevar a cabo operaciones como almacenar en memoria los datos obtenidos, enviar lecturas actuales de los valores de potencia y valores rms a la computadora, entre otras funciones.

El medidor posee un software de aplicación, con el cual es posible comunicarse con el medidor mediante el puerto serial de la computadora; para obtener lecturas actuales de potencia y de valores rms, así como descargar la memoria del medidor.

Posee un pulso de salida de frecuencia con el que es posible calibrar el medidor, utilizando un software de aplicación de calibración asociado al medidor, mediante la conexión RS232.



Figura A.1.1 Cara frontal del medidor trifásico multifunción



Figura A.1.2 Cara trasera del medidor trifásico multifunción

1.3 ESPECIFICACIONES

1.3.1 EXACTITUD

El medidor posee una exactitud del 0.713%, y fue calibrado con el patrón de calibración WECO PORTÁTIL de la marca WATTHOUR ENGINEERING CO., INC., en las instalaciones del laboratorio de metrología, que es parte del Área de Protección y Mediciones, del Departamento de Mantenimiento, de la Empresa Transmisora de El Salvador SA de CV, ETESAL.

1.3.2 CONSTANTE DE PULSO KH

La constante de pulso del medidor es de 0.2 Wh/pulso; y se obtiene una frecuencia máxima en el pulso de salida de frecuencia de 2.5 Hz, para magnitudes de voltaje y de corriente, de 120 V y 5 A, a un FP = 1.

1.3.3 RELOJ DE TIEMPO REAL

Este reloj es implementado por un microcontrolador, y puede ser configurado por el jumper de intervalo de acumulación para que emita una salida cada 12 o cada 20 minutos; esto es de utilidad, debido a que con esta señal el microcontrolador maestro guarda las lecturas de los parámetros eléctricos cada 12 o 20 minutos en la memoria.

1.3.4 MEMORIA

La capacidad de la memoria RAM estática no volátil es de 128KBx8. Y el máximo de días que se pueden almacenar se especifica a continuación:

Tiempo del intervalo de acumulación del medidor	12 minutos	20 minutos
Número de días	35 días	58 días

1.3.5 COMUNICACIÓN

La comunicación de datos se efectúa mediante el puerto serial asíncrono con la interfase RS232, con el conector DB9. Las características de la comunicación son las siguientes: 9600 baudios, 1 bit de inicio, 1 bit de parad y 8 bits de datos.

1.3.6 FUENTE DE ALIMENTACION

La fuente de alimentación es de 5V, 0.5 A; y posee un circuito de respaldo con una batería de 12 V, 1.2 Ah, la cual se atenúa a 5V y es capaz de suplir por un máximo de dos días.

1.3.7 DIMENSIONES

Las dimensiones del medidor son las siguientes: Largo = 30 cm, Ancho = 30 cm y Altura = 8 cm. Ver Figura A.1.3

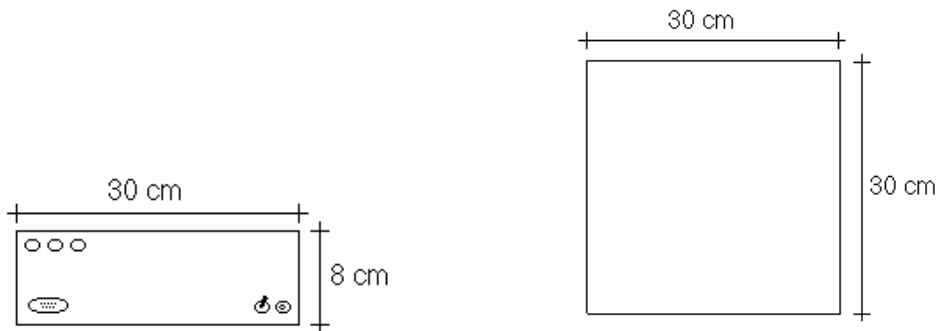


Figura A.1.3 Dimensiones físicas del medidor.

SECCIÓN 2: HARDWARE DEL MEDIDOR

2.1 PRESENTACION DEL MEDIDOR.

La figura A.1.4 muestra la vista frontal del medidor multifunción donde se pueden apreciar los indicadores del status del medidor así como los controles que intervienen para poner en funcionamiento al medidor, a continuación se detalla cada uno de ellos.

Señal IRQ: este indicador se apagara cuando en el microcontrolador ADE7754 se lleva a cabo una interrupción.

Pulso de reloj: este indicador permanece intermitente desde el momento que se energiza el medidor y corresponde a cada cambio de estado del reloj medidos en segundos.

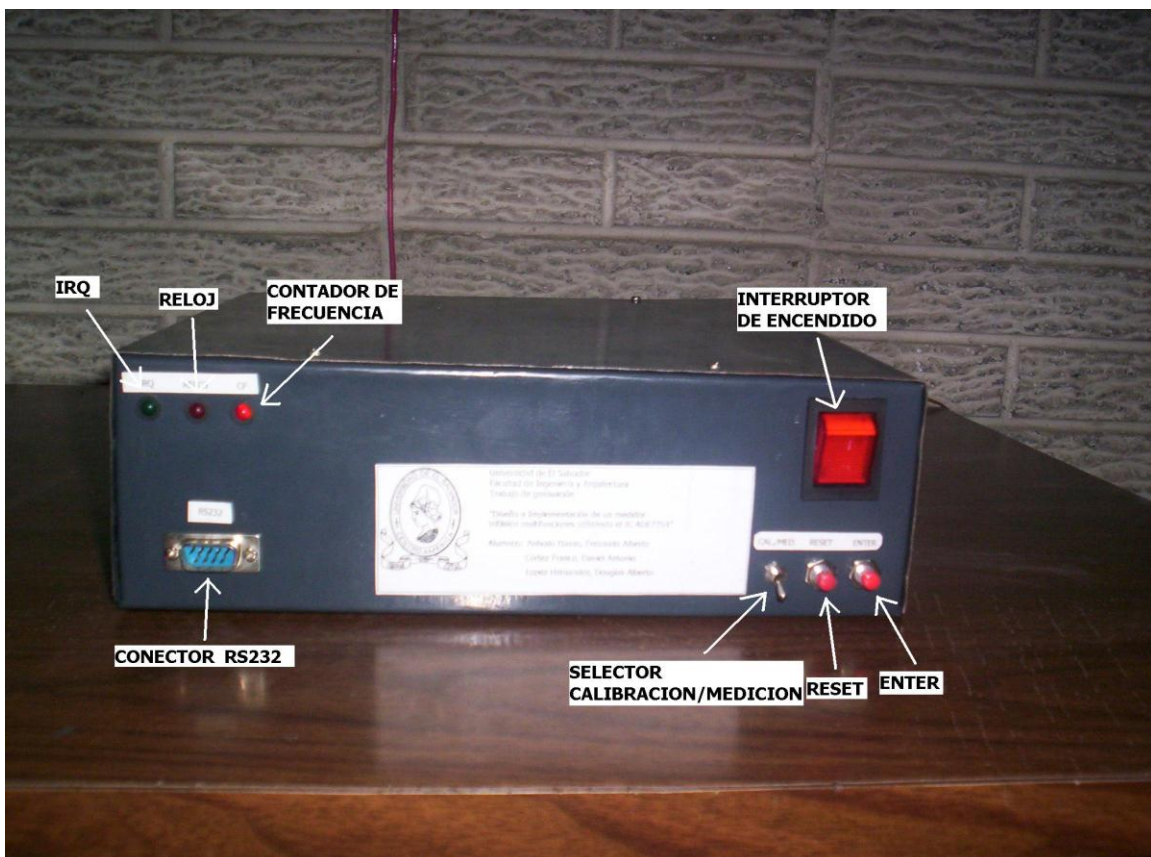


Figura A.1.4 Vista frontal del medidor

Contador de frecuencia: este pulso de salida sirve para propósito de calibración y es proporcional a la cantidad de potencia medida por del equipo.

Interfase RS232: este puerto de comunicación es el medio para conectarse a una computadora para poder obtener los datos que el medidor tiene almacenados en memoria o los datos actuales del medidor y posteriormente ser procesados por el computador.

EL selector de modo calibración y medición es un switch de dos posiciones con el cual el usuario puede definir si quiere calibrar el medidor en cualquier momento.

Switch de Enter es un pulsador que tiene como función inicializar al medidor cuando este se ha instalado y energizado y debe de pulsarse cada vez que el medidor ha sido puesto en reset.

Reset: con este boton se da un reset a todo el sistema del medidor.

Swicht de encendido: es el boton que enciende y apaga al medidor.

En la figura A.1.5 se muestra la parte posterior del medidor donde pueden apreciarse las entradas de los canales de voltaje y corriente para las diferentes configuraciones del medidor(para una mayor información refiérase a la sección tipos de servicio de conexión). Además se muestra la entrada de alimentación AC para el medidor con su respectivo elemento de protección.



Figura A.1.5 Vista trasera del medidor

2.2 TARJETAS DEL MEDIDOR

El medidor trifásico multifunción consta de 4 tarjetas electrónicas las cuales se detallan a continuación.

2.2.1 TARJETA DEL MEDIDOR ADE7754.

Esta tarjeta contiene la etapa de atenuación de voltaje y corriente que necesita el ADE7754 en sus entradas analógicas para atenuar las señales a niveles de 0.5 Voltios pico a full escala, donde cada una de las 6 entradas analógicas posee un filtro antialias para evitar el traslape de las señales que se muestrean y garantizar que el ADE7754 no recibe muestras erróneas, véase la figura A.1.6.

También pueden encontrarse los filtros de compensación de fase que son necesarios para corregir el desfase que introducen los transformadores de corriente.

Los resistores Bourden son los encargados de la transformar la corriente a voltaje para poder ser medido por el ADE7754 en su entrada analógica y los resistores de atenuación de las señales de voltaje de la red, para obtener un valor de voltaje pequeño necesario para la medición.

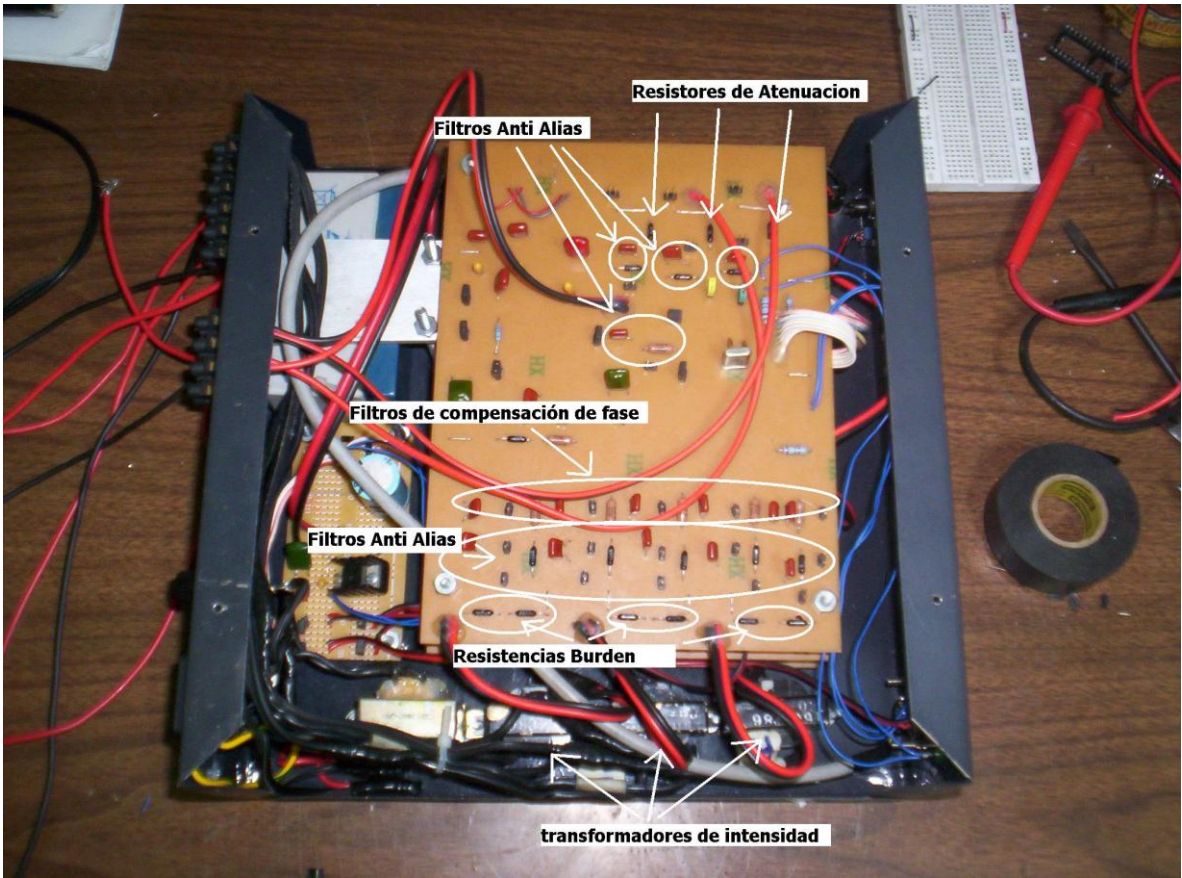


Figura A.1.6 Tarjeta del medidor ADE7754

2.2.2 TARJETA DEL MICROCONTROLADOR MAESTRO Y RELOJ.

Esta constituida por dos microcontroladores PIC16F877 uno de ellos es el maestro encargado de vigilar todos los procesos que se llevan a cabo en el medidor y quien habilita cada uno de los microcontroladores esclavos.

El reloj es el encargado de llevar el tiempo de cada uno de los eventos que se registran en el medidor, llevando así un historial que esta siendo almacenado en la memoria. Ver figura A.1.7

Además se muestra los circuitos integrados utilizados como aislamiento entre la interfaz del medidor y la computadora para proteger así contra una falla en el medidor a un periférico conectado al puerto.

El circuito integrado MAX232 es el encargado de la conversión de niveles TTL a RS232 ya que el microcontrolador maestro trabaja con niveles TTL en la interfaz SCI y la interfaz con la computadora necesita de niveles utilizados por el estándar RS232.

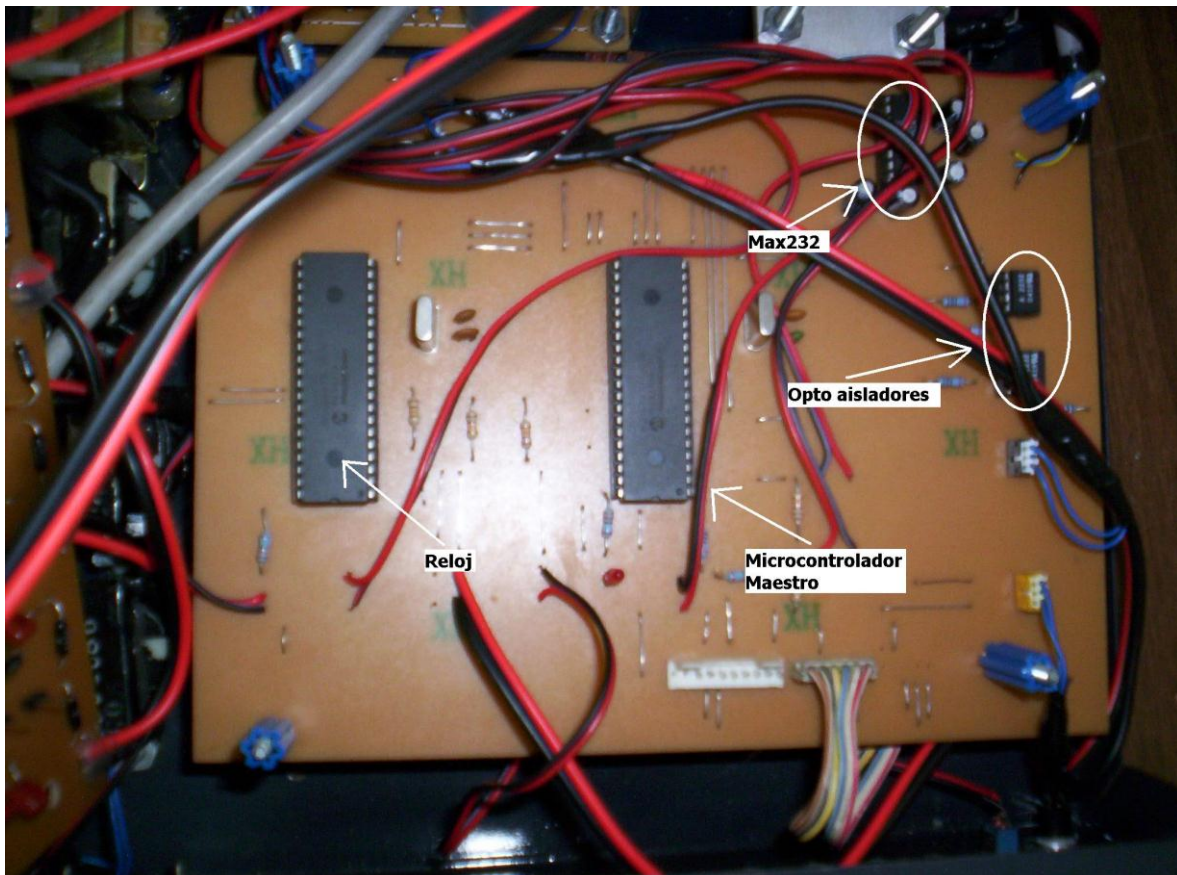


Figura A.1.7 Tarjeta del microcontrolador maestro

2.2.3 TARJETA DE LA MEMORIA SRAM.

Esta tarjeta esta implementada con una memoria RAM estática de 128Kx8 para el almacenamiento de los datos que el medidor esta obteniendo y fue desarrollada con la ayuda de un microcontrolador PIC16F877 por la facilidad de poseer una interfaz SPI que se adecua a las necesidades del medidor. Ver figura A.1.8.

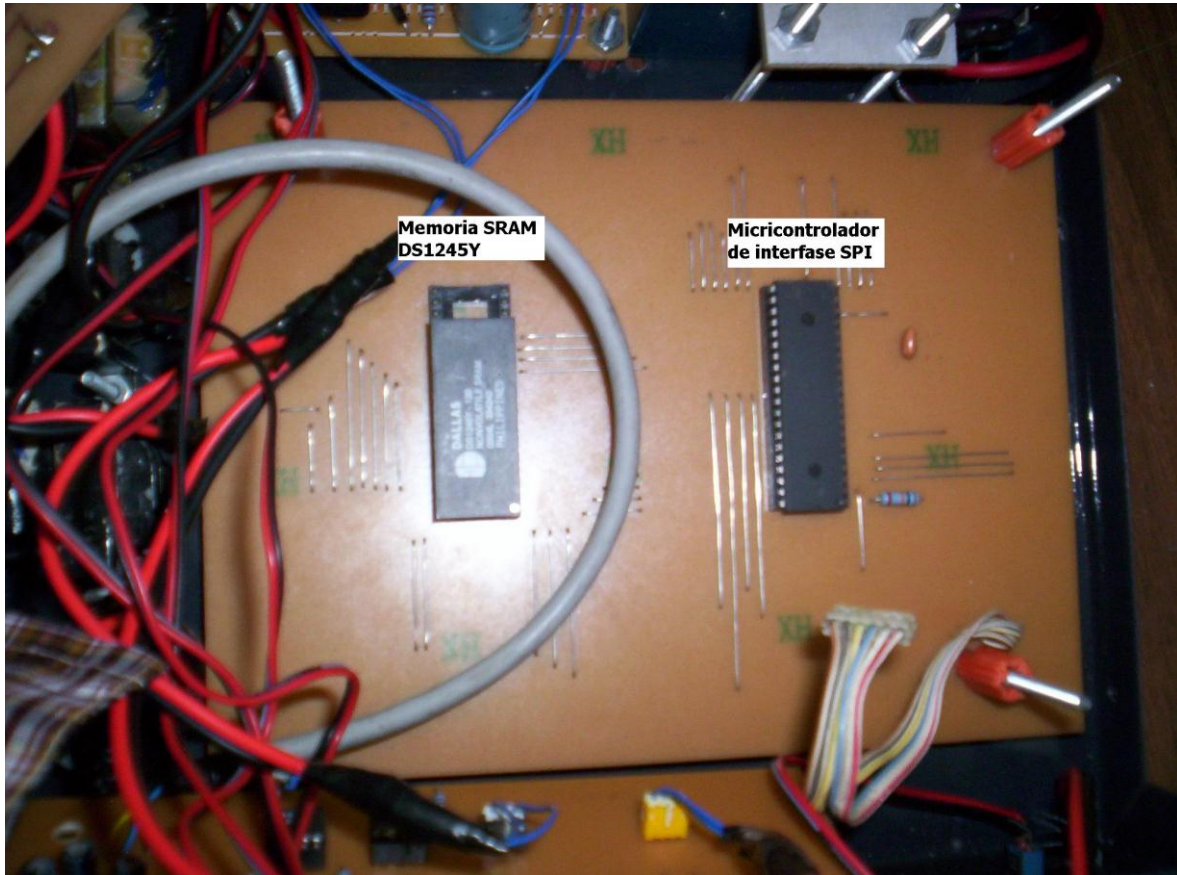


Figura A.1.8 Tarjeta de memoria

2.2.4 TARJETA DE LA FUENTE DE VOLTAJE CON BATERIA DE RESPALDO.

Esta fuente se implemento con e propósito de mantener en cualquier momento el servicio de energía para el medidor multifunción ya que si existiera una falla en al red eléctrica el sistema tendrá la posibilidad de continuar operando sin ninguna alteración de los datos por el uso de la batearía que se incluye en el medidor teniendo un respaldo de aproximadamente dos días mientras se reestablece el suministro de la red. La figura A.1.9 muestra la fuente de voltaje y la batería que se encuentran dentro del medidor.

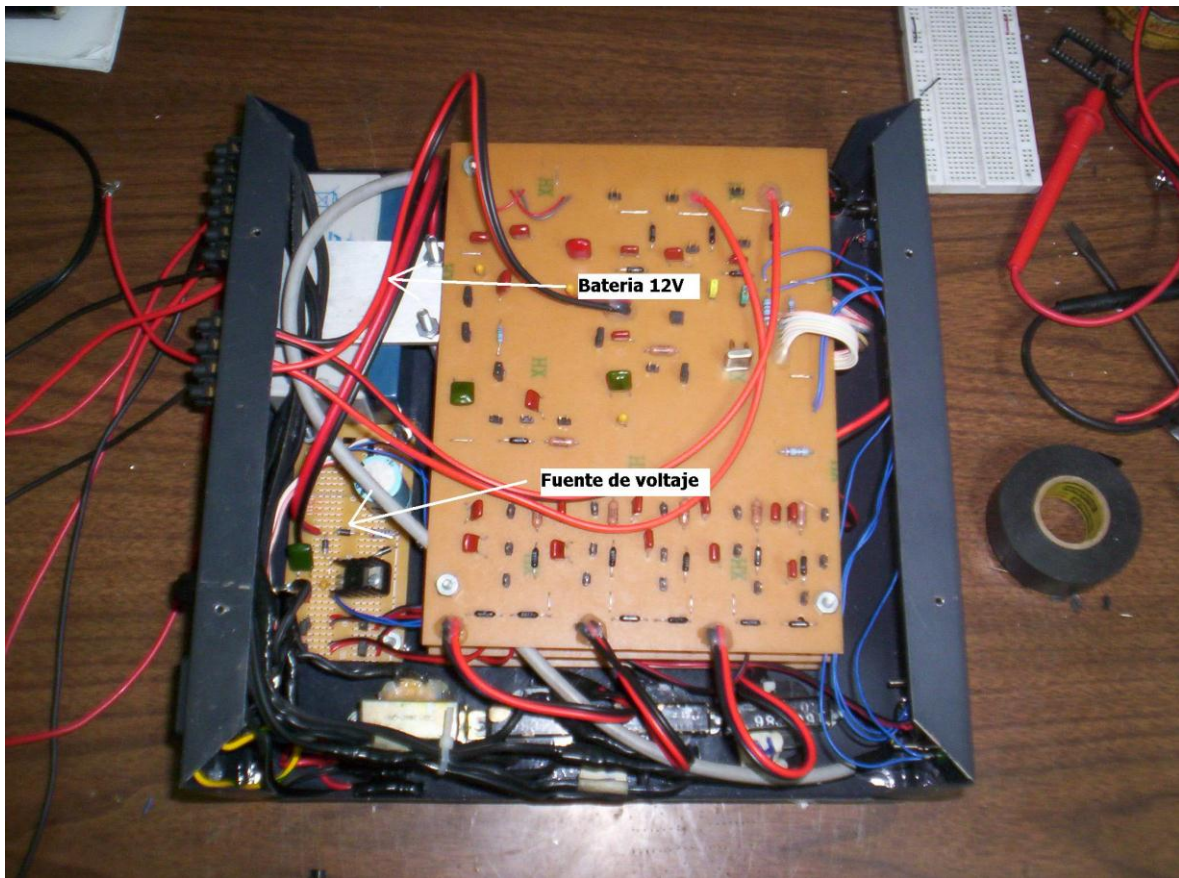


Figura A.1.9 Tarjeta de la fuente con batería

2.2 TIPOS DE SERVICIO DE CONEXIÓN

ESQUEMA DE CONEXION PARA UN SERVICIO ESTRELLA 4 HILOS/ 9S SEGUN LA DISPOSICION DE LAS BORNERAS DEL MEDIDOR

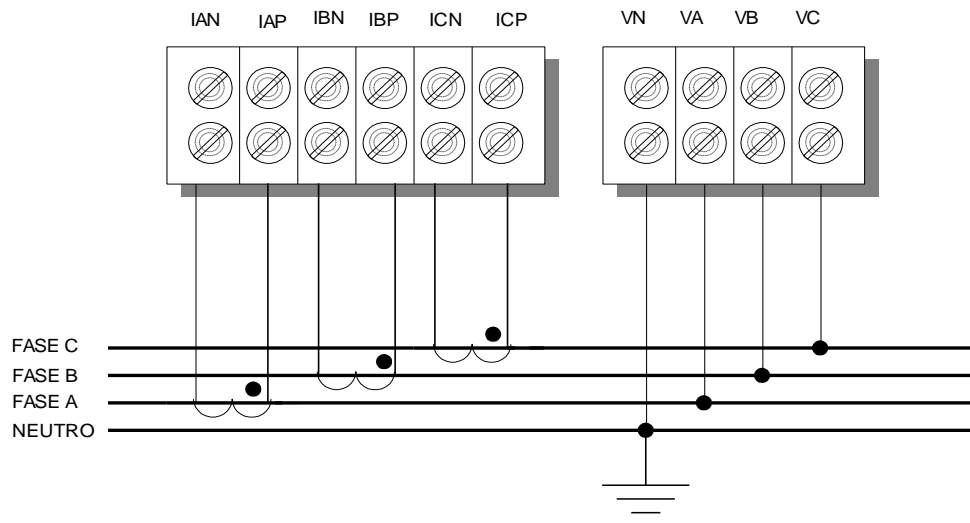


Figura A.1.10 Esquema de conexión para estrella 4 hilos 9S

ESQUEMA DE CONEXION PARA UN SERVICIO DELTA 4 HILOS/ 8S SEGUN LA DISPOSICION DE LAS BORNERAS DEL MEDIDOR

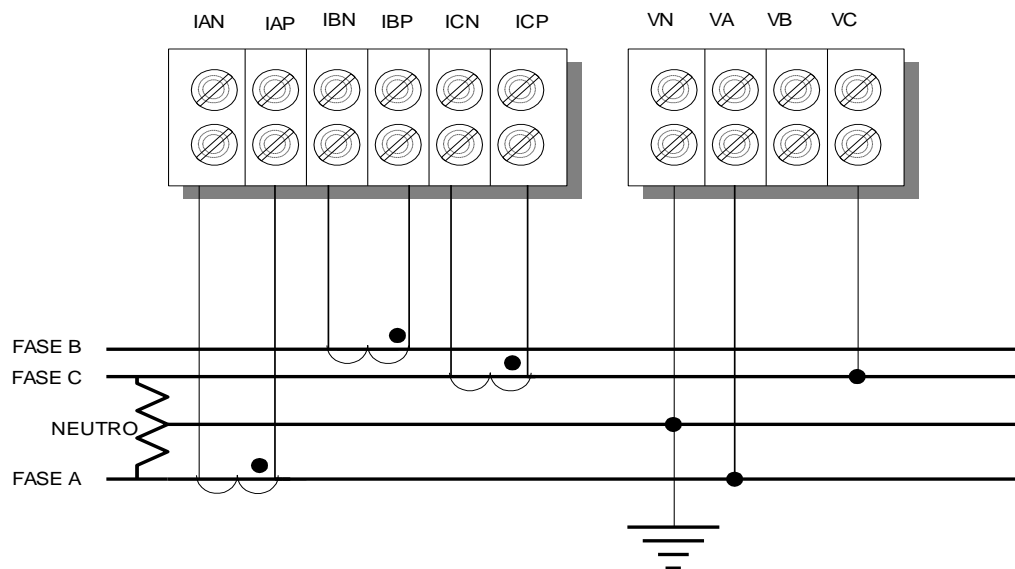


Figura A.1.11 Esquema de conexión para delta 4 hilos 8S

ESQUEMA DE CONEXION PARA UN SERVICIO ESTRELLA 4 HILOS/ 6S SEGUN LA DISPOSICION DE LAS BORNERAS DEL MEDIDOR

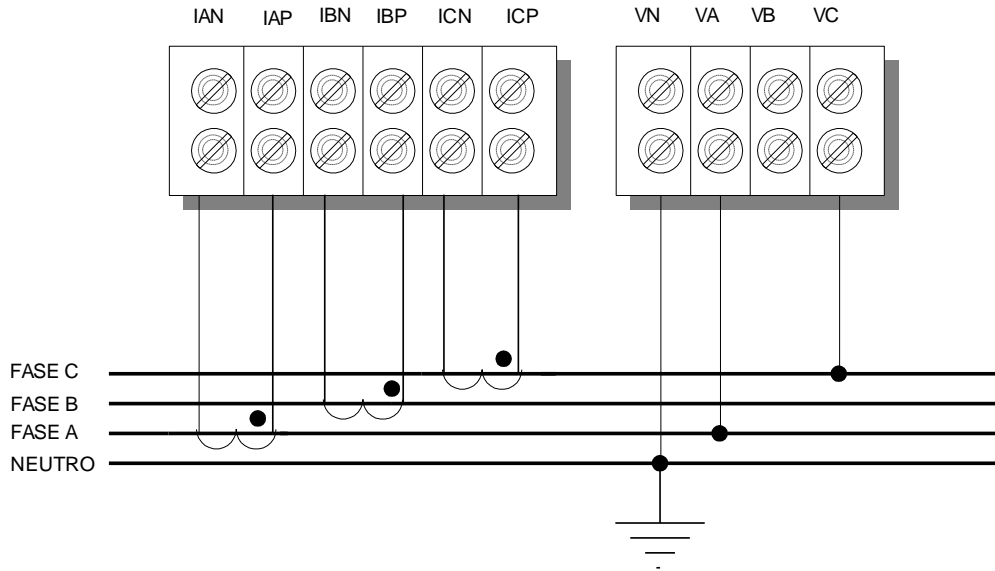


Figura A.1.12 Esquema de conexión para estrella 4 hilos 6S

ESQUEMA DE CONEXION PARA UN SERVICIO DELTA DE 3 HILOS/5S SEGUN LA DISPOSICION DE LAS BORNERAS DEL MEDIDOR

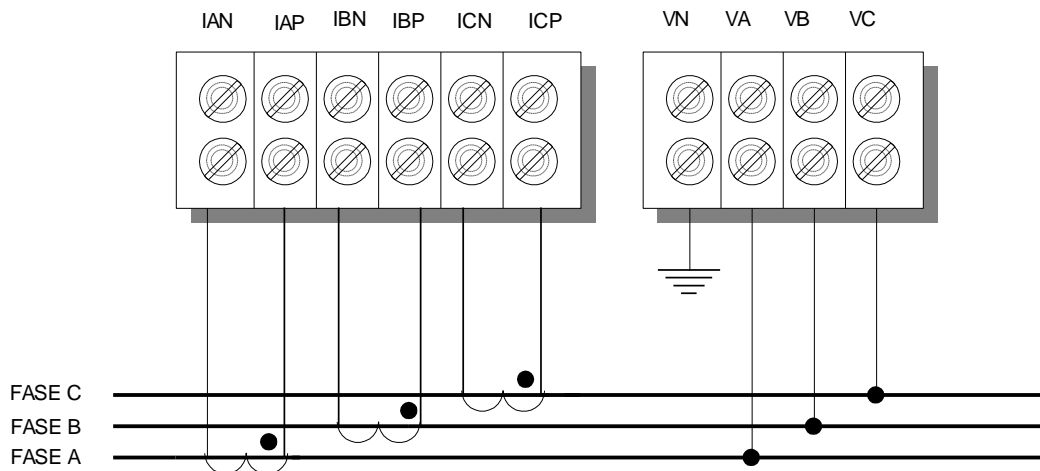


Figura A.1.13 Esquema de conexión para delta 3 hilos 5S

2.5 CONFIGURACIÓN DE LOS JUMPER

Jumper	Opción	Descripción
JP1-JP7-JP1	cerrado	el efecto es deshabilitar los filtros de compensación de fase para las entradas analógicas IAN, IBN, ICN respectivamente. Por default abiertos
JP2-JP8-JP14	cerrado	esto conectara la entradas analógicas IAP, IBP y ICP respectivamente a tierra. Por default abiertos.
JP3-JP9-JP15	cerrado	esto conectara las entradas analógicas IAN, IBN, ICN respectivamente a tierra. Por default abiertos
JP4-JP10-JP16	cerrado	el efecto es deshabilitar los filtro de compensación de fase para los transformadores de corriente de las entradas analógicas IAN, IBN y ICN respectivamente. Por default abiertos.
JP5-JP11-JP17	cerrado	el efecto es deshabilitar los filtros anti-alias de las entradas analógicas IAP, IBP y ICP respectivamente. Por default abiertos
JP6-JP12-JP18	cerrado	el efecto es deshabilitar los filtros anti-alias de las entradas analógicas IAN, IBN y ICN respectivamente. Por default abiertos.
JP19	cerrado	esto cortocircuitara las tierras analógica y digital para una misma referencia. Por default cerrado
JP20-JP21-JP22	cerrado	esto cortocircuitara las redes de atenuación de los canales de voltaje. Por default abiertos.
JP23	cerrado	el efecto es deshabilitar los filtros anti-alias de la entrada analógica VN. Por default abierto
JP24	cerrado	esto conectara a tierra la entrada analógica VN. Por default cerrado
JP25-JP26-JP27	cerrado	esto conectara a tierra las entradas analógicas VCP, VBP y VAP respectivamente. Por default abiertos
JP28-JP29-JP30	cerrado	esto desconectara totalmente las entradas analógicas VCP, VBP y VAP del ADE7754. por default cerrados

Jumper	Opción	Descripción															
JP31	cerrado	esto desconectara totalmente la entrada analógica VN del ADE7754. por default cerrado															
JP32-JP33	-----	<p>la combinación de estos jumper seleccionara el tipo de servicio del medidor.</p> <table> <tr> <td>JP 33</td> <td>JP32</td> <td>TIPO SERVICIO</td> </tr> <tr> <td>Cerrado</td> <td>Cerrado</td> <td>9S/estrella 4 hilos</td> </tr> <tr> <td>Cerrado</td> <td>Abierto</td> <td>8S/delta 4 hilos</td> </tr> <tr> <td>Abierto</td> <td>Cerrado</td> <td>6S/estrella 4 hilos</td> </tr> <tr> <td>Abierto</td> <td>Abierto</td> <td>5S/delta 3 hilos.</td> </tr> </table>	JP 33	JP32	TIPO SERVICIO	Cerrado	Cerrado	9S/estrella 4 hilos	Cerrado	Abierto	8S/delta 4 hilos	Abierto	Cerrado	6S/estrella 4 hilos	Abierto	Abierto	5S/delta 3 hilos.
JP 33	JP32	TIPO SERVICIO															
Cerrado	Cerrado	9S/estrella 4 hilos															
Cerrado	Abierto	8S/delta 4 hilos															
Abierto	Cerrado	6S/estrella 4 hilos															
Abierto	Abierto	5S/delta 3 hilos.															
JP35	-----	<p>este jumper sirve para seleccionar el tiempo de acumulación de datos en la memoria RAM del medidor.</p> <table> <tr> <td>Cerrado</td> <td>acumula cada 3 minutos</td> </tr> <tr> <td>Abierto</td> <td>acumula cada 5 minutos</td> </tr> </table>	Cerrado	acumula cada 3 minutos	Abierto	acumula cada 5 minutos											
Cerrado	acumula cada 3 minutos																
Abierto	acumula cada 5 minutos																

Tabla A.1.1 Configuración de los jumper

SECCIÓN 3: INSTRUCCIONES DE OPERACIÓN

Para el buen funcionamiento del medidor multifunción se requiere seguir una serie de pasos que nos ayudaran a corregir inconvenientes y posibles daños en el medidor, por lo que recomendamos leer detenidamente esta sección.

En primer lugar asegúrese que el medidor se encuentra configurado internamente para el tipo de servicio al que estará conectado para no obtener lecturas errores de la medición. Para ello debe de quitar la cubierta superior del medidor para poder accesar a los jumper internos de configuración.

Advertencia riesgo de choque eléctrico.

Asegúrese que la alimentación de la red esta desconectada cada vez que va a manipular estos jumper, ya que puede recibir una descarga eléctrica.

Una vez abierto el medidor la configuración puede realizarse en la tarjeta de la fuente de voltaje donde se encuentran los jumper JP32 y JP33 de configuración, según se muestran en la figura A.1.14. para mayor referencia puede consultar la **sección configuración de los jumper** para determinar la conexión de los mismos para cada uno de los diferentes tipos servicio que puede ser conectado el medidor.

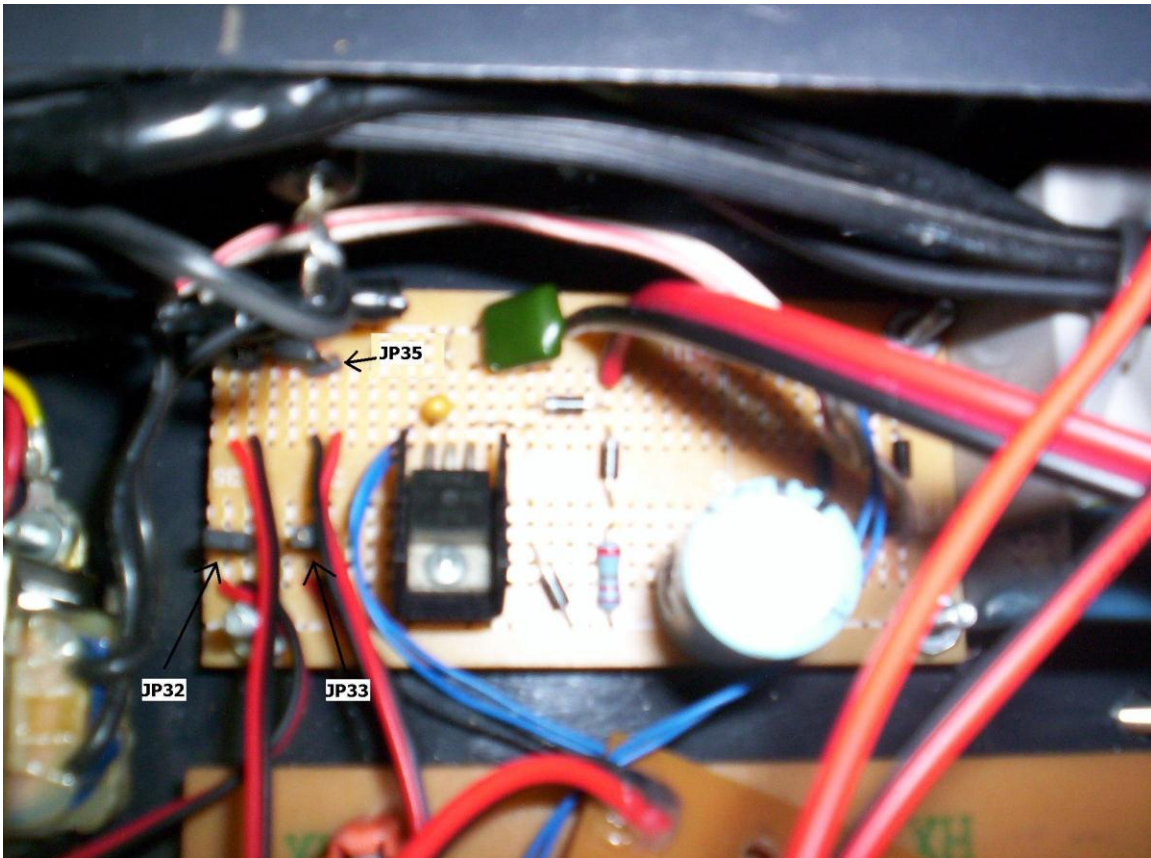


Figura A.1.14 Detalle de jumper de configuración

El JP35 corresponde al tiempo de acumulación que tendrán los datos en la memoria para mayor referencia véase la sección de configuración de jumper para determinar que tiempo de acumulación se seleccionara en el microcontrolador maestro.

Una vez hecho lo anterior puede colocar la cubierta superior del medidor y se procede a realiza las conexiones en las borneras de entradas de voltaje y corriente de acuerdo al tipo de servicio que se halla seleccionado para un completo detalle vea la sección tipos de servicio de conexión.

Ahora el medidor esta listo para ser energizado de la red eléctrica colocando el cable de poder que se proporciona y colocar el swicht principal en la posicion de encendido. El sistema comenzara a utilizar la energía de la red automáticamente y solo será necesario colocar el switch de modo de calibración / medición ubicado en el panel frontal en la posición calibración y pulsar el botón de Enter para inicializar las rutinas de calibración en el PIC maestro.

desde este momento el medidor esta operando y listo para comenzar a medir. Cuando la energía Eléctrica fluya por las borneras de las entradas de corriente y

estén presentes los voltajes en el medidor comenzara a observarse que se enciende y se apaga el led del pulso contador de frecuencia CF indicando que se esta midiendo la potencia activa de la carga, cuando todo lo anterior se ha hecho el swicht calibración/medicion se dejara en la posicion medicion.

Ahora el usuario podrá hacer uso del software proporcionado como interfase grafica para poder realizar la descarga de los datos de la memoria en cualquier momento durante el medidor esta en funcionamiento.

SECCIÓN 4: SOFTWARE DEL MEDIDOR

El software del medidor fue hecho en el lenguaje de programación de alto nivel LabVIEW. Las aplicaciones desarrollados son dos: el VI de Calibración y el VI Principal.

4.1 VI DE CALIBRACIÓN

Con esta aplicación es posible ajustar el valor del pulso de frecuencia de salida CF, mediante la configuración de los registros del ADE7754.

Como se puede apreciar en la Figura A.1.15, este VI contiene entradas, salidas y botones. Las salidas de texto son los cuadros de fondo gris que tienen números en su interior; así como las entradas de texto se pueden distinguir por tener un fondo blanco. Como se puede observar en el VI, existen tres columnas para cada registro de calibración, de izquierda a derecha, la primera columna son los valores leídos conforme a los datos guardados de los registros en la EEPROM por default, la segunda columna es para escribir los nuevos para los registros; y finalmente la tercer columna muestra los valores escritos en los registros del ADE7754.

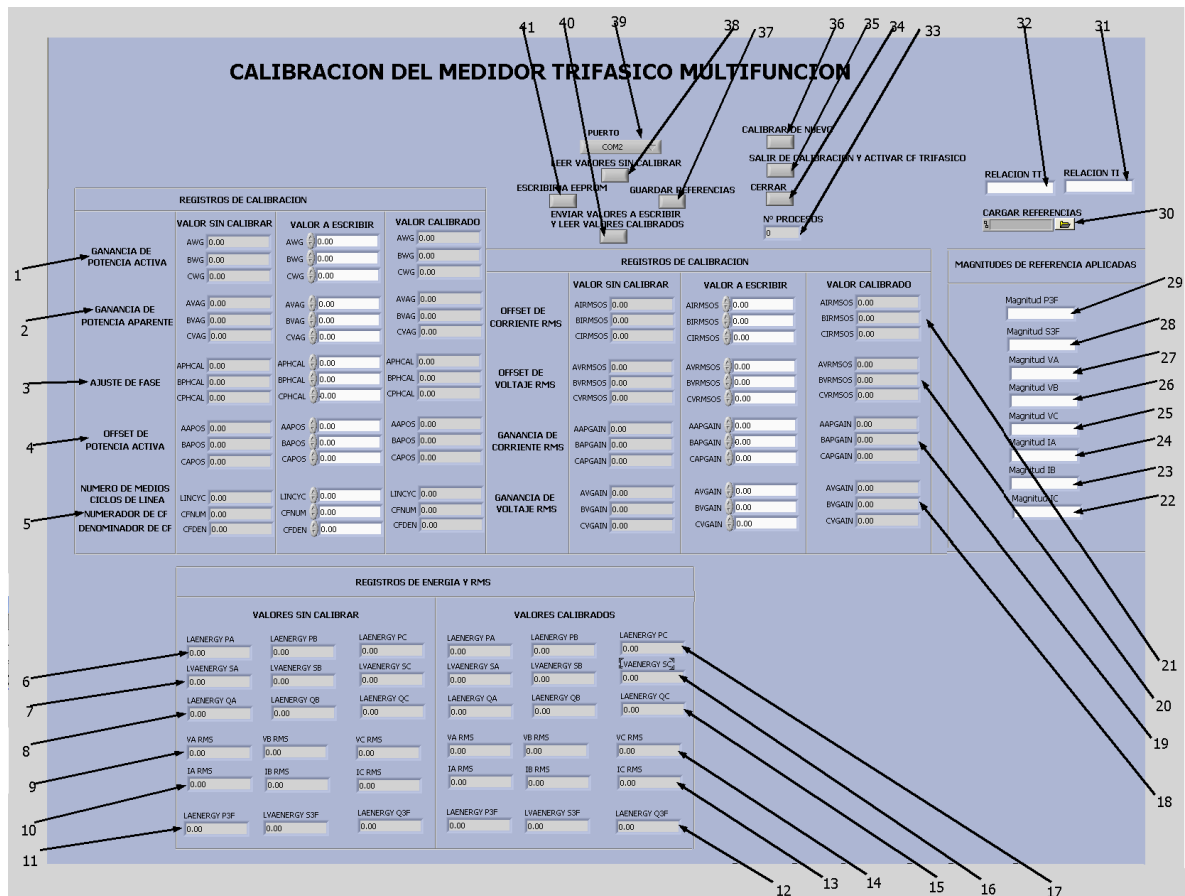


Figura A.1.15 VI de calibración

Para una mejor comprensión, se muestran secciones de este VI para poder apreciar más claramente los detalles.

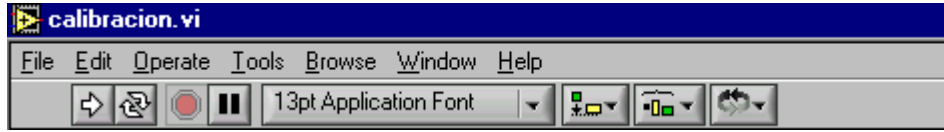


Figura A.1.16 Barra de ejecución de LabVIEW

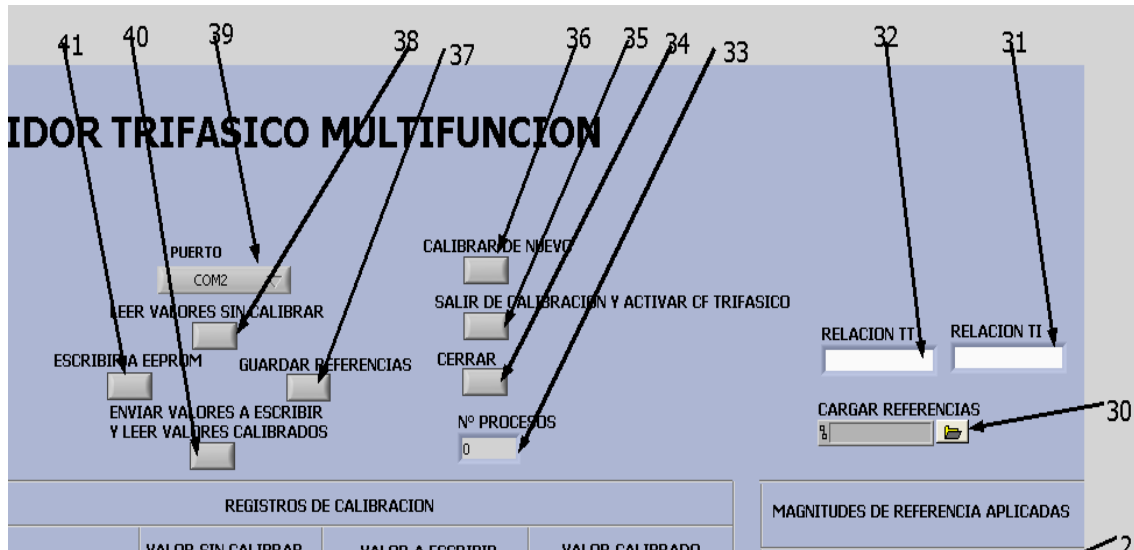


Figura A.1.17 VI de calibración sección 1



Figura A.1.18 VI de calibración sección 2

ENVIAR VALORES A ESCRIBIR Y LEER VALORES CALIBRADOS Nº PROCESOS: 0 CARGAR REFERENCIAS

REGISTROS DE CALIBRACION			MAGNITUDES DE REFERENCIA APLICADAS	
	VALOR SIN CALIBRAR	VALOR A ESCRIBIR	VALOR CALIBRADO	
OFFSET DE CORRIENTE RMS	AIRMSOS 0.00	AIRMSOS 0.00	AIRMSOS 0.00	Magnitud P3F Magnitud S3F Magnitud VA Magnitud VB Magnitud VC Magnitud IA Magnitud IB Magnitud IC
	BIRMSOS 0.00	BIRMSOS 0.00	BIRMSOS 0.00	
	CIRMSOS 0.00	CIRMSOS 0.00	CIRMSOS 0.00	
OFFSET DE VOLTAJE RMS	AVRMSOS 0.00	AVRMSOS 0.00	AVRMSOS 0.00	
	BVRMSOS 0.00	BVRMSOS 0.00	BVRMSOS 0.00	
	CVRMSOS 0.00	CVRMSOS 0.00	CVRMSOS 0.00	
GANANCIA DE CORRIENTE RMS	AAPGAIN 0.00	AAPGAIN 0.00	AAPGAIN 0.00	
	BAPGAIN 0.00	BAPGAIN 0.00	BAPGAIN 0.00	
	CAPGAIN 0.00	CAPGAIN 0.00	CAPGAIN 0.00	
GANANCIA DE VOLTAJE RMS	AVGAIN 0.00	AVGAIN 0.00	AVGAIN 0.00	
	BVGAIN 0.00	BVGAIN 0.00	BVGAIN 0.00	
	CVGAIN 0.00	CVGAIN 0.00	CVGAIN 0.00	

VALORES CALIBRADOS	
A LAENERGY PB 0.00	LAENERGY PC 0.00
SA LVAENERGY SB 0.00	LVAENERGY SC 0.00
QA LAENERGY QB 0.00	LAENERGY QC 0.00
VB RMS 0.00	VC RMS 0.00
IB RMS 0.00	IC RMS 0.00
S3F LVAENERGY S3F 0.00	LAENERGY Q3F 0.00

Figura A.1.19 VI de calibración sección 3

REGISTROS DE ENERGIA Y RMS						
VALORES SIN CALIBRAR			VALORES CALIBRADOS			
6	LAENERGY PA 0.00	LAENERGY PB 0.00	LAENERGY PC 0.00	LAENERGY PA 0.00	LAENERGY PB 0.00	LAENERGY PC 0.00
7	LVAENERGY SA 0.00	LVAENERGY SB 0.00	LVAENERGY SC 0.00	LVAENERGY SA 0.00	LVAENERGY SB 0.00	LVAENERGY SC 0.00
8	LAENERGY QA 0.00	LAENERGY QB 0.00	LAENERGY QC 0.00	LAENERGY QA 0.00	LAENERGY QB 0.00	LAENERGY QC 0.00
9	VA RMS 0.00	VB RMS 0.00	VC RMS 0.00	VA RMS 0.00	VB RMS 0.00	VC RMS 0.00
10	IA RMS 0.00	IB RMS 0.00	IC RMS 0.00	IA RMS 0.00	IB RMS 0.00	IC RMS 0.00
11	LAENERGY P3F 0.00	LVAENERGY S3F 0.00	LAENERGY Q3F 0.00	LAENERGY P3F 0.00	LVAENERGY S3F 0.00	LAENERGY Q3F 0.00

Figura A.1.20 VI de calibración sección 4

PROCEDIMIENTO DE CALIBRACION

1. Para activar este VI se presiona el botón con la flecha en la barra de ejecución de LabVIEW. Si se desea detener antes de terminar el proceso de calibración presione el botón con el círculo rojo. Ver Figura A.1.16.
2. Se selecciona el tipo de puerto serial que se utilizará, el COM1 o el COM2; presionando el botón de PUERTO correspondiente al literal 39 de la Figura A.1.17.
3. Se digita la relación de transformación de los TI en el campo de texto del literal 31 Figura A.1.17, así como la relación de transformación del los TT, en el literal 32 Figura A.1.17. Esto se hace de la siguiente manera:

600:5 significa que cuando se tenga una lectura de 600 A en el primario se tendrá una lectura de 5 A en el secundario, del transformador de corriente.
4. Se presiona el botón de cargar referencias del literal 30 de la Figura A.1.17. Cuando se presiona el botón aparece un menú emergente, ver Figura A.1.21, el cual solicita la ubicación del archivo de referencias. Al cargar las referencias, se recomienda utilizar las referencias dadas con el software de aplicación.

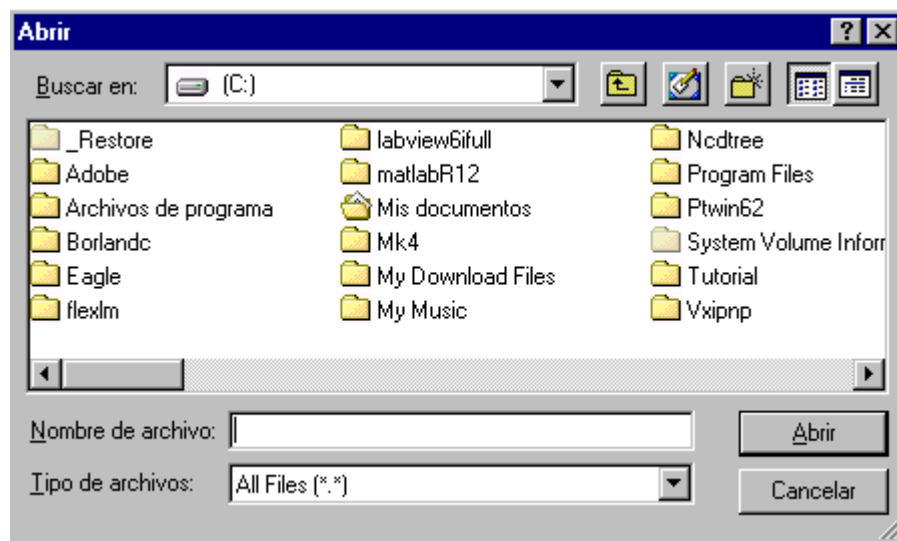


Figura A.1.21 Menú emergente para abrir un archivo

5. Se presiona el botón LEER VALORES SIN CALIBRAR correspondiente al literal 38 de la Figura A.1.17; el medidor mostrará los valores de los registros de calibración que se encuentran en la memoria EEPROM, así como los registros de potencia y de valores rms actuales.
6. Observe las salidas numéricas de la primera columna correspondiente a los valores sin calibrar, para los registros de potencia, de voltaje y de corriente:

Registros de potencia activa sin calibrar	literal 6 Figura A.1.20
Registros de potencia aparente sin calibrar	literal 7 Figura A.1.20
Registros de potencia reactiva sin calibrar	literal 8 Figura A.1.20
Registros de voltaje rms sin calibrar	literal 9 Figura A.1.20
Registros de corriente rms sin calibrar	literal 10 Figura A.1.20
Registros de potencia trifásica sin calibrar	literal 11 Figura A.1.20

Así como para los registros de calibración:

Registros de ganancia de potencia activa	literal 1 Figura A.1.18
Registros de ganancia de potencia aparente	literal 2 Figura A.1.18
Registros de ganancia de fase	literal 3 Figura A.1.18
Registros de offset de potencia activa	literal 4 Figura A.1.18
Números de medios ciclos de línea	literal 5 Figura A.1.18
Multiplicador o numerador de CF	literal 5 Figura A.1.18
Divisor o denominador de CF	literal 5 Figura A.1.18

Registros de ganancia de voltaje rms	literal 21 Figura A.1.19
Registros de ganancia de corriente rms	literal 20 Figura A.1.19
Registros de offset de voltaje rms	literal 19 Figura A.1.19
Registros de offset de corriente rms	literal 18 Figura A.1.19

7. Para ajustar el pulso de salida de frecuencia, se tiene acceso a los registros del ADE7754. Los registros de ganancia y de offset son por fase. Cabe destacar que los valores para los registros de ganancia deben estar en el rango de -2047 a 2047 . Los registros de ajuste de fase entre -15 a 15 , el registro de CFNUM entre 0 a 4095 , el registro de CFDEN entre 1 a 4095 , y el número de ciclos de línea entre 20 a 2000 .
8. Se escribe en los campos de entrada de texto correspondientes a cada uno de los registros de configuración, los valores deseados. Si se desea guardar estos valores de manera permanente en la memoria EEPROM, se presiona el botón de ESCRIBIR A EEPROM literal 41 Figura A.1.17.

9. Si se desea guardar las referencias de calibración actuales, se digitan las magnitudes de referencia aplicadas de potencia, voltaje y corriente; conforme a los siguientes campos de texto:

Magnitud de potencia activa trifásica	literal 29 Figura A.1.19
Magnitud de potencia aparente trifásica	literal 28 Figura A.1.19
Magnitud de voltaje rms fase A	literal 27 Figura A.1.19
Magnitud de voltaje rms fase B	literal 26 Figura A.1.19
Magnitud de voltaje rms fase C	literal 25 Figura A.1.19
Magnitud de corriente rms fase A	literal 24 Figura A.1.19
Magnitud de corriente rms fase B	literal 23 Figura A.1.19
Magnitud de corriente rms fase C	literal 22 Figura A.1.19

Finalmente se presiona el botón GUARDAR REFERENCIAS literal 37 Figura A.1.17.

10. Después de presionado el botón, aparecerá un menú emergente solicitando la ubicación donde se almacenará el archivo con las referencias de calibración que contiene el valor de los registros en complemento a 2 así como las magnitudes digitadas, de potencia trifásica, voltaje y corriente por fase.
11. Se presiona el botón ENVIAR VALORES A ESCRIBIR Y LEER VALORES CALIBRADOS literal 40 Figura A.1.17.
12. Observe las salidas numéricas en la tercera columna correspondiente a los valores calibrados, para los registros de calibración. Así se puede observar también los valores de la energía, los valores de voltaje y corriente.

Registros de potencia activa calibrados	literal 6 Figura A.1.20
Registros de potencia aparente calibrados	literal 7 Figura A.1.20
Registros de potencia reactiva calibrados	literal 8 Figura A.1.20
Registros de voltaje rms calibrados	literal 9 Figura A.1.20
Registros de corriente rms calibrados	literal 10 Figura A.1.20
Registros de potencia trifásica calibrados	literal 11 Figura A.1.20

13. Si se desea modificar los valores de calibración nuevamente, se presiona el botón CALIBRAR DE NUEVO literal 36 Figura A.1.17, y el campo de salida N° PROCESOS literal 33 Figura A.1.17 se incrementará en 1, y se repiten todos los pasos anteriores; en caso contrario, se debe presionar el botón SALIR DE CALIBRACIÓN Y ACTIVAR CF TRIFÁSICO literal 35 Figura A.1.17, en este modo el medidor se queda con los valores calibrados guardados en la memoria EEPROM, y el pulso de salida de frecuencia trifásico se activa; este pulso es de una

gran utilidad, debido a que con este se logra determinar el % de error en la exactitud del medidor. Cuando no se utiliza más el modo de calibración, se puede presionar el botón CERRAR literal 34 Figura A.1.17, y se cierra la aplicación, y el medidor comienza en modo medición.

4.2 VI PRINCIPAL

Con esta aplicación es posible comunicarse con el medidor para: ajustar el reloj de tiempo real, hacer lecturas actuales de potencia y valores rms, y hacer descargas de potencia y valores rms. También se tienen funciones para procesar los archivos de textos creados con la descarga de datos del medidor.

Cuando se selecciona una operación aparece un VI emergente correspondiente, que al cerrarlo, retorna a esta aplicación.

Las funciones de lectura actual de potencia y de valores rms, y la de descarga de memoria, necesitan que el medidor este en funcionamiento, y que el cable de comunicación este conectado al DB9 del medidor, y al DB9 de la PC si se utiliza el COM1 o el DB25 de la PC si se utiliza el COM2.

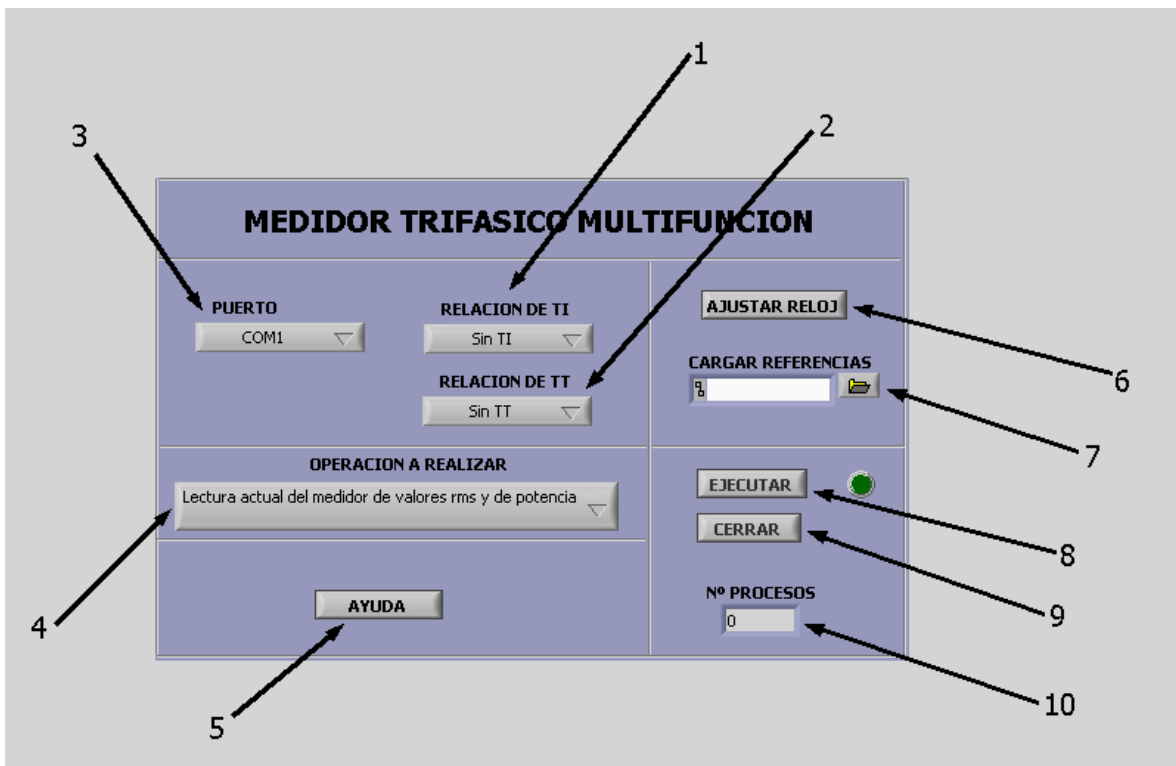


Figura A.1.22 VI Principal

PROCEDIMIENTO PARA UTILIZAR EL PROGRAMA PRINCIPAL

1. Se debe conocer cual puerto será el que se utilizará si el COM1 o el COM2, luego se procede a seleccionar el puerto en el VI presionando el botón literal 3 Figura A.1.22.
2. Se debe conocer la relación de transformación de los TI y de los TT, y para ello se tienen los botones literal 1 y literal 2 Figura A.1.22.
3. Se debe ajustar el reloj del medidor si hace falta. Para ello se presiona el botón AJUSTAR RELOJ literal 6 Figura A.1.22, y aparecerá un VI emergente de MODIFICAR HORA Y FECHA DEL MEDIDOR, ver Figura A.1.23.

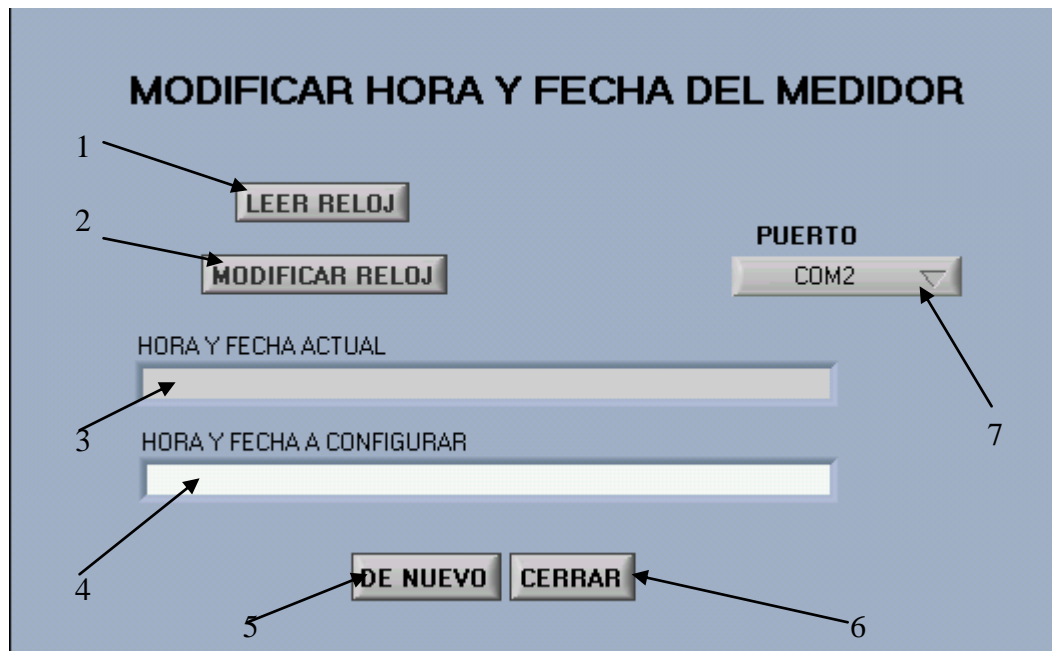


Figura A.1.23 VI de modificar hora y fecha del medidor

Se selecciona el tipo de puerto con el botón PUERTO literal 7 Figura A.1.23, luego se presiona el botón LEER RELOJ literal 1 Figura A.1.23, y aparece la hora y la fecha actual en el campo de salida literal 3 Figura A.1.23, si se desea leer el reloj o modificarlo, se debe presionar el botón DE NUEVO literal 5 Figura A.1.23; en caso contrario, se presiona el botón CERRAR literal 6 Figura A.1.23.

Si se presionó el botón DE NUEVO, se tiene la opción de leer el reloj nuevamente presionando el botón LEER RELOJ, o de modificar el reloj presionando el botón MODIFICAR RELOJ literal 2 Figura A.1.23. Al introducir la fecha, se hace en el siguiente formato:

Día/Mes/Año/Hora:Minuto:Segundo/AM(ó PM)
1/Enero/2000/12: 1: 0/AM

4. Se deben cargar las referencias de calibración presionando el botón CARGAR REFERENCIAS literal 7 Figura A.1.22, y aparecerá un menú emergente solicitando el archivo de referencias.
5. Se escoge la función que se desea, presionando el botón OPERACIÓN A REALIZAR literal 4 Figura A.1.22, con el cual se observarán cinco opciones al presionarlo. Ver Figura A.1.24.

Estas opciones son:

- ❖ Lectura actual de potencia y de valores rms
- ❖ Lectura de archivo de potencia y de valores rms dada una fecha
- ❖ Lectura de archivo de potencia y de valores rms entre dos fechas
- ❖ Descarga de memoria
- ❖ Generación de archivos de reporte HTML

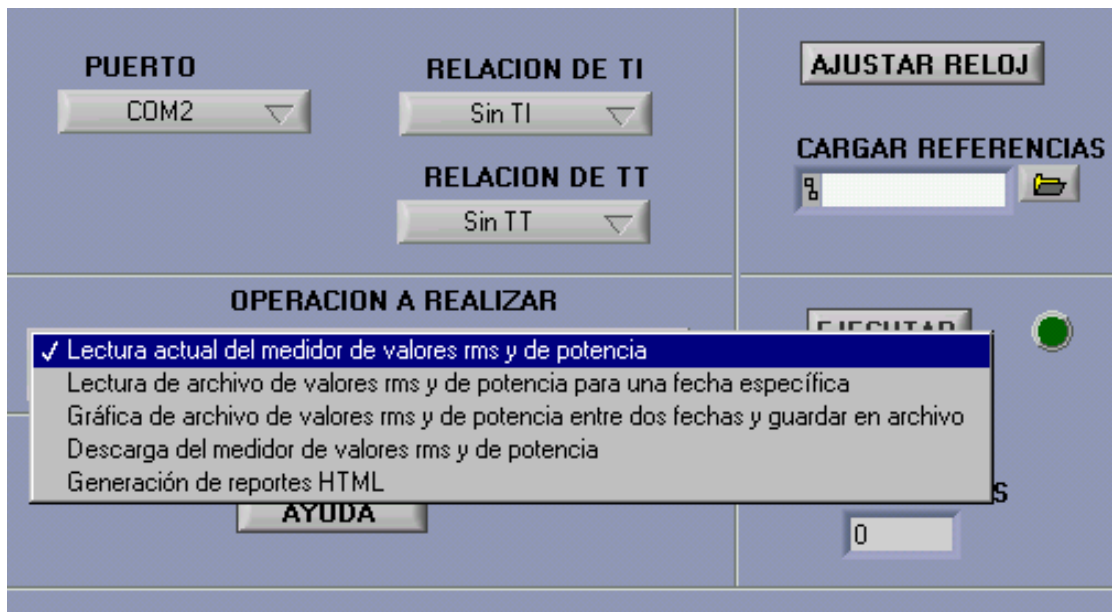


Figura A.1.24 Menú de operaciones a realizar

6. Se presiona el botón EJECUTAR, aparecerá un VI emergente correspondiente a la operación que se eligió. Cuando el VI emergente se cierra, inmediatamente regresa al VI principal.

7. Después de que el VI emergente regresa al VI principal, se tienen dos opciones se realiza otra operación, repitiendo todos los pasos anteriores, con lo cual el campo de texto N° DE PROCESOS literal 10 Figura A.1.22 se incrementa en 1; o se presiona el botón CERRAR literal 9 Figura A.1.22, y se cierra la aplicación.

4.2.1 VI DE LECTURA ACTUAL DE POTENCIA Y DE VALORES RMS

Cuando se selecciona la operación de lectura actual de potencia y de valores rms, y se presiona el botón EJECUTAR del programa principal; aparece este VI emergente

PROCEDIMIENTO DE LECTURA ACTUAL DE POTENCIA Y DE VALORES RMS

1. Se presiona el botón LEER literal 2 Figura A.1.25, luego aparecen los valores medidos por el medidor en ese momento, en los cuadros de salida:

Hora y fecha	literal 1 Figura A.1.25
Potencia activa trifásica	literal 5 Figura A.1.25
Potencia aparente trifásica	literal 6 Figura A.1.25
Potencia reactiva trifásica	literal 13 Figura A.1.25
Factor de potencia trifásica	literal 14 Figura A.1.25
Voltaje rms de la fase A	literal 7 Figura A.1.25
Voltaje rms de la fase B	literal 9 Figura A.1.25
Voltaje rms de la fase C	literal 12 Figura A.1.25
Corriente rms de la fase A	literal 8 Figura A.1.25
Corriente rms de la fase B	literal 10 Figura A.1.25
Corriente rms de la fase C	literal 11 Figura A.1.25

2. Si se desea leer otra vez, se presiona el botón DE NUEVO literal 3 Figura A.1.25; de lo contrario se puede presionar el botón CERRAR literal 4 Figura A.1.25.

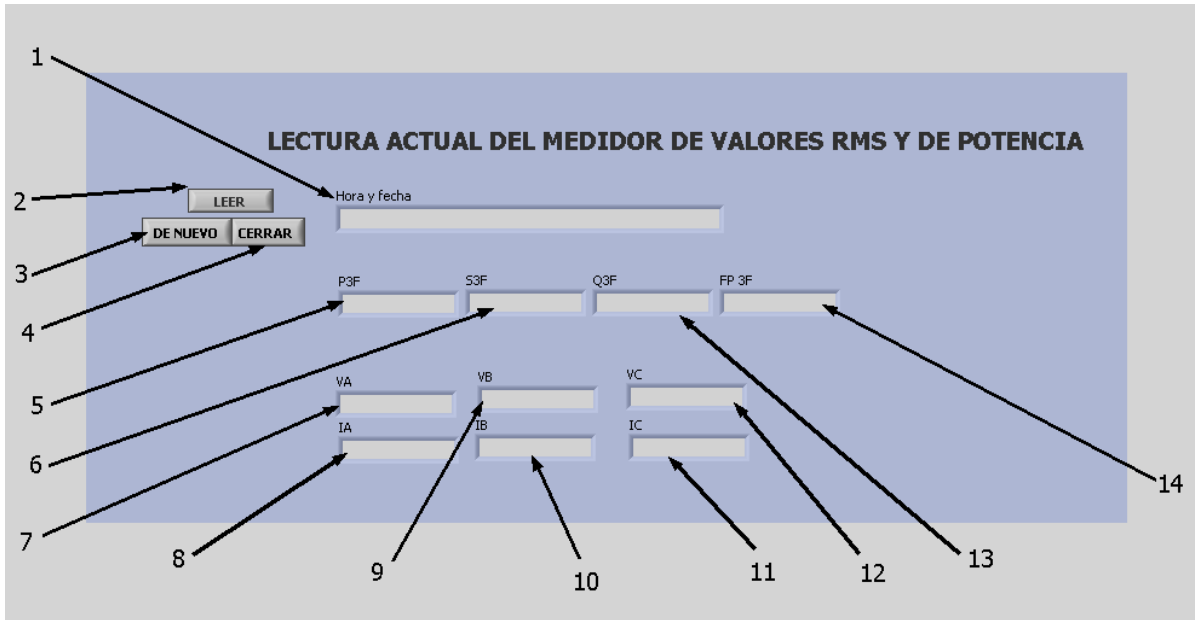


Figura A.1.25 VI de lectura actual de potencia y valores rms

4.2.2 VI DE LECTURA DE ARCHIVO DE POTENCIA Y DE VALORES RMS PARA UNA FECHA ESPECÍFICA

Con este VI es posible determinar el comportamiento de la carga para una hora y fecha específica durante el intervalo de medición.

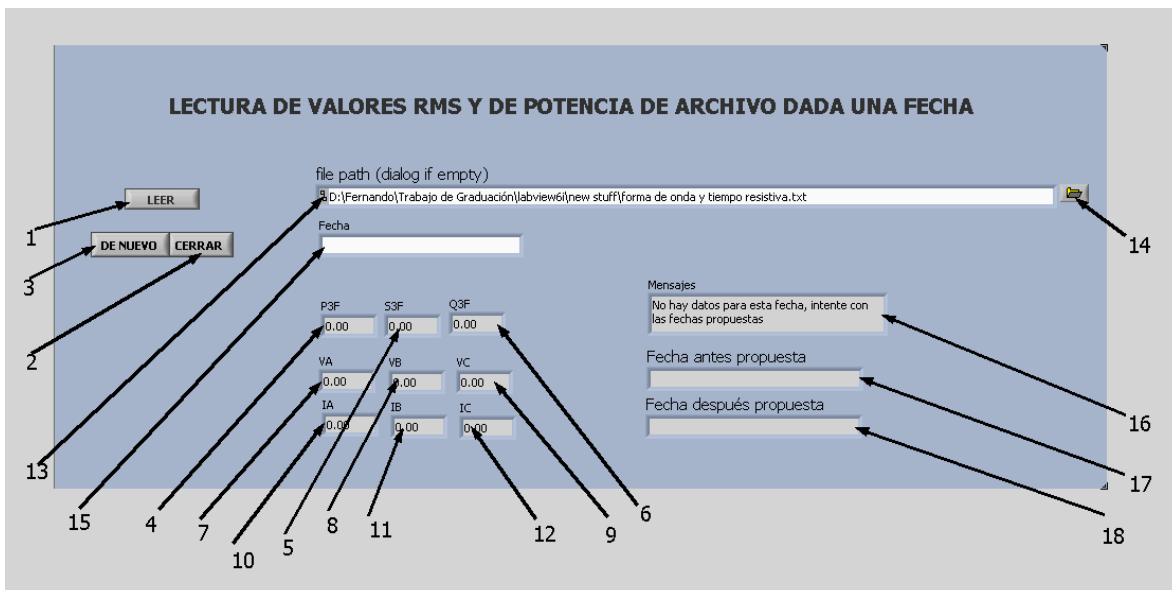


Figura A.1.26 VI de lectura de archivo de potencia y de valores rms dada una fecha

PROCEDIMIENTO DE LECTURA DE ARCHIVO DE POTENCIA Y DE VALORES RMS DADA UNA FECHA

1. Se introduce el nombre y la ubicación del archivo de texto que contiene los datos previamente descargados en el campo de entrada literal 13 Figura A.1.26, o se presiona el botón literal 14 Figura A.1.26 para cargar el archivo. Este archivo de texto debe ser para gráficos. Ver PROCEDIMIENTO DE VI DE DESCARGA.
2. Se introduce la hora y fecha de interés, en el campo de entrada literal 15 Figura A.1.26 ; como se explicó en MODIFICAR RELOJ.
3. Se presiona el botón LEER literal 1 Figura A.1.26, y aparecen los datos en los campos de salida:

Potencia activa trifásica	literal 4 Figura A.1.26
Potencia aparente trifásica	literal 5 Figura A.1.26
Potencia reactiva trifásica	literal 6 Figura A.1.26
Voltaje rms fase A	literal 7 Figura A.1.26
Voltaje rms fase B	literal 8 Figura A.1.26
Voltaje rms fase C	literal 9 Figura A.1.26
Corriente rms fase A	literal 10 Figura A.1.26
Corriente rms fase B	literal 11 Figura A.1.26
Corriente rms fase C	literal 12 Figura A.1.26

Si no hay datos en los campos de salida, luego aparecerá un mensaje en el campo de salida literal 16 Figura A.1.26, y aparecerán fechas propuestas antes y después, en los campos de salida literal 17 Figura A.1.26 y literal 18 Figura A.1.26, respectivamente, si las hay.

4. Si se desea hacer otra lectura presione el botón DE NUEVO literal 3 Figura A.1.26, y repita los pasos anteriores; en caso contrario, presione el botón CERRAR literal 2 Figura A.1.26.

4.2.3 VI DE LECTURA DE ARCHIVO DE POTENCIA Y DE VALORES RMS ENTRE DOS FECHAS

Este VI muestra el comportamiento de la carga entre dos fechas, mediante el despliegue de datos en gráficas, y permitiendo la posibilidad de guardar los datos si se desea.

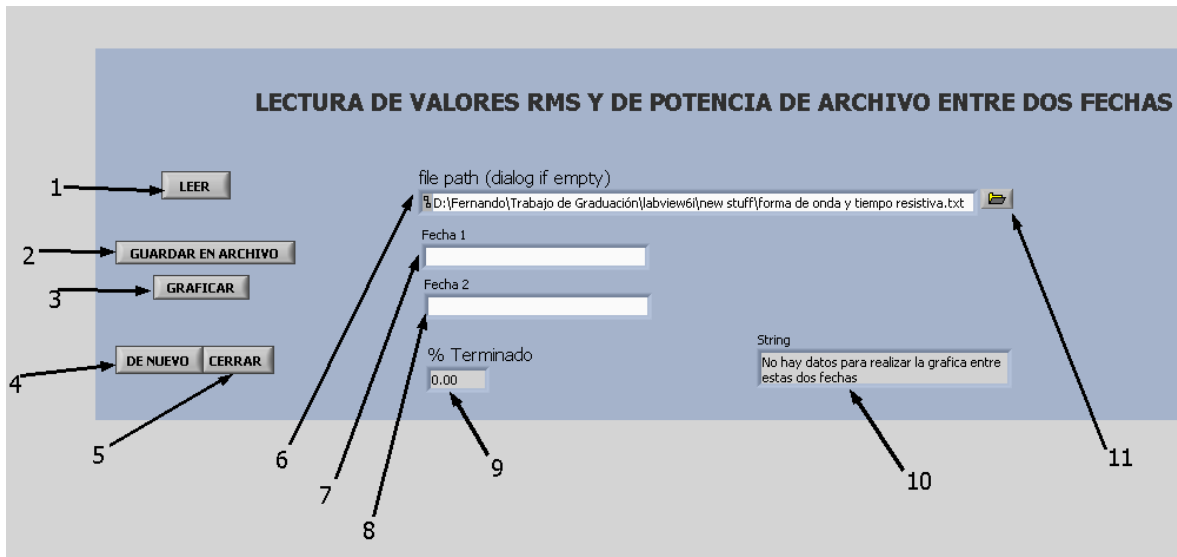


Figura A.1.27 VI de lectura de archivo de potencia y de valores rms entre dos fechas

PROCEDIMIENTO PARA LECTURA DE ARCHIVO DE POTENCIA Y VALORES RMS ENTRE DOS FECHAS

1. Se introduce el nombre y la ubicación del archivo de texto que contiene los datos previamente descargados en el campo de entrada literal 6 Figura A.1.27, o se presiona el botón literal 11 Figura A.1.27 para cargar el archivo. Este archivo de texto debe ser para gráficos. Ver PROCEDIMIENTO DE VI DE DESCARGA.
2. Se introduce la primera hora y fecha de interés, en el campo de entrada literal 7 Figura A.1.27 ; como se explicó en MODIFICAR RELOJ. Luego, se introduce la segunda hora y fecha de interés, en el campo de entrada literal 8 Figura A.1.27.
3. Se presiona el botón GUARDAR EN ARCHIVO literal 2 Figura A.1.27, si se desea guardar los datos entre las dos fechas en un archivo de texto; se presiona el botón GRAFICAR literal 3 Figura A.1.27, si se desea graficar los datos entre las dos fechas. Este gráfico es magnitud contra muestras. Ver Figura A.1.28, muestra en el literal 15 el gráfico de voltaje contra muestra, y en el literal 16 el gráfico de corriente contra muestra; muestra las tres fases por gráfico. Ver Fig. A.1.29, muestra en el literal 12 el gráfico de potencia activa trifásica contra muestra, en el literal 13 el gráfico de potencia aparente trifásica contra muestra, y en el literal 14 la potencia reactiva trifásica contra muestra.

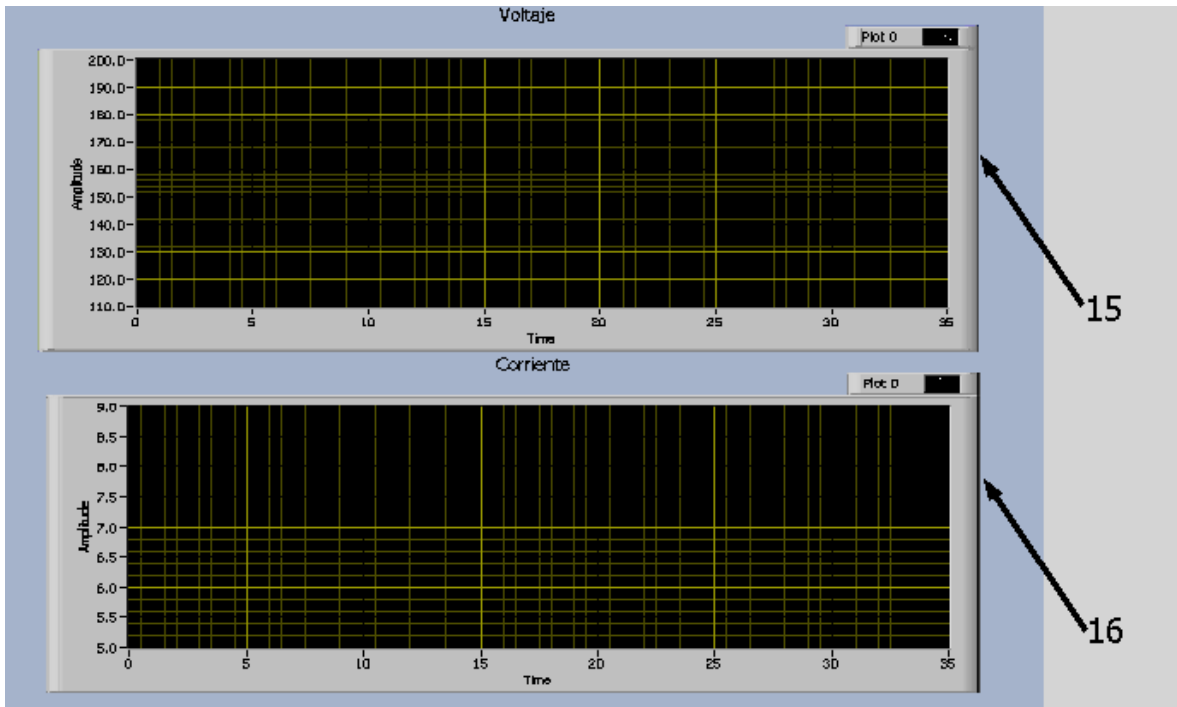


Figura A.1.28 VI de lectura de archivo de potencia y de valores rms entre dos fechas-salidas de gráficos de voltaje y corriente

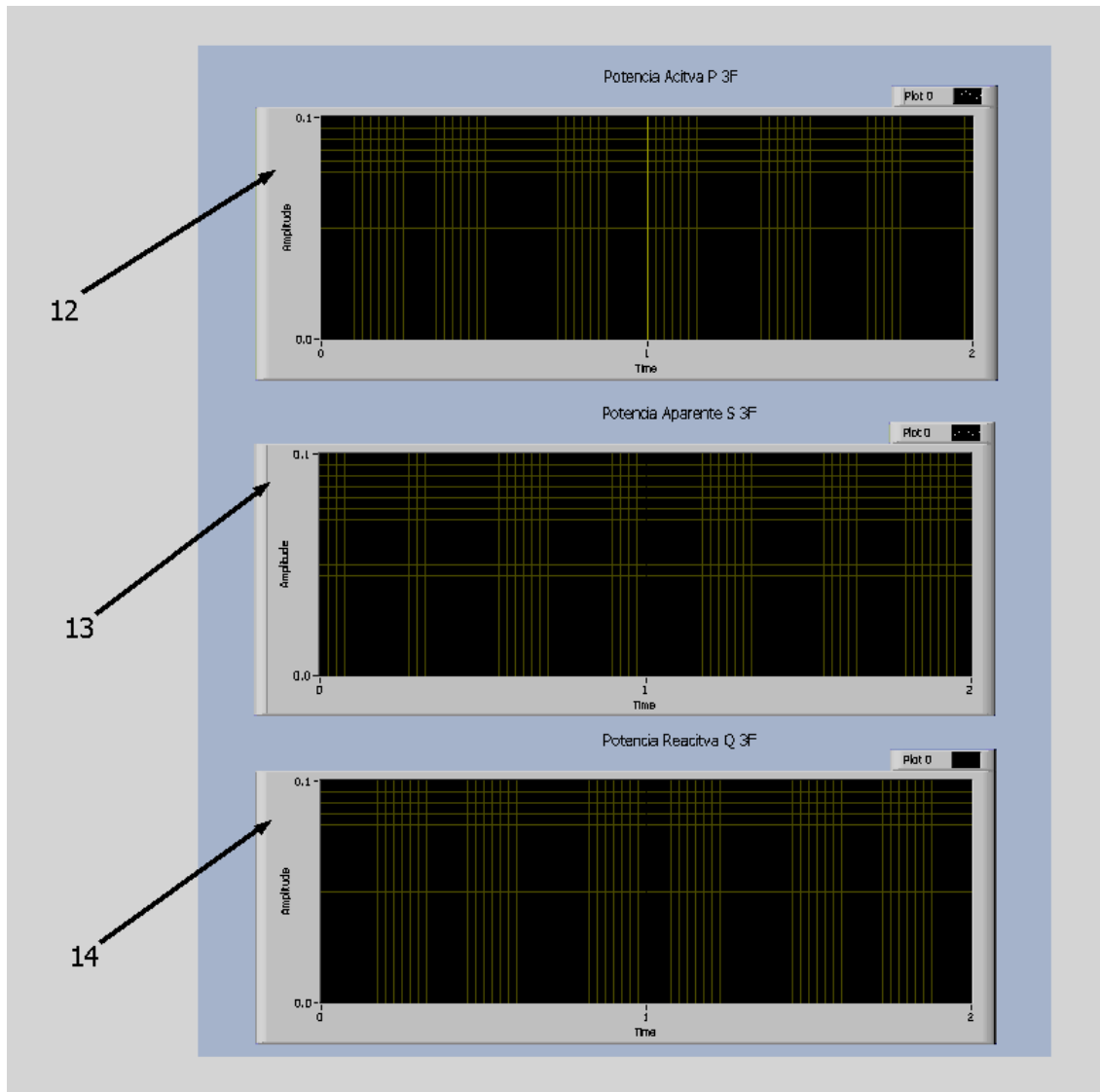


Figura A.1.29 VI de lectura de archivo de potencia y de valores rms entre dos fechas-salidas de gráficos de potencia trifásica

4. Se presiona el botón LEER literal 1 Figura A.1.27, y aparecen los datos en las gráficas campos de salida, si se eligió la opción; o aparece un menú emergente que solicita introducir un nombre de archivo y su ubicación, si se eligió esta opción. De lo contrario aparecerá un mensaje en el campo de salida literal 10 Figura A.1.21. Mientras el VI realiza la tarea de guardar los datos o de graficar, el VI muestra el progreso mediante el campo de salida %TERMINADO literal 9 Figura A.1.27.
5. Si se desea hacer otra lectura presione el botón DE NUEVO literal 4 Figura A.1.27, y repita los pasos anteriores; en caso contrario, presione el botón CERRAR literal 5 Figura A.1.27.

4.2.4 VI DE DESCARGA DE MEMORIA

Este VI descarga el contenido de la memoria en un archivo de texto.

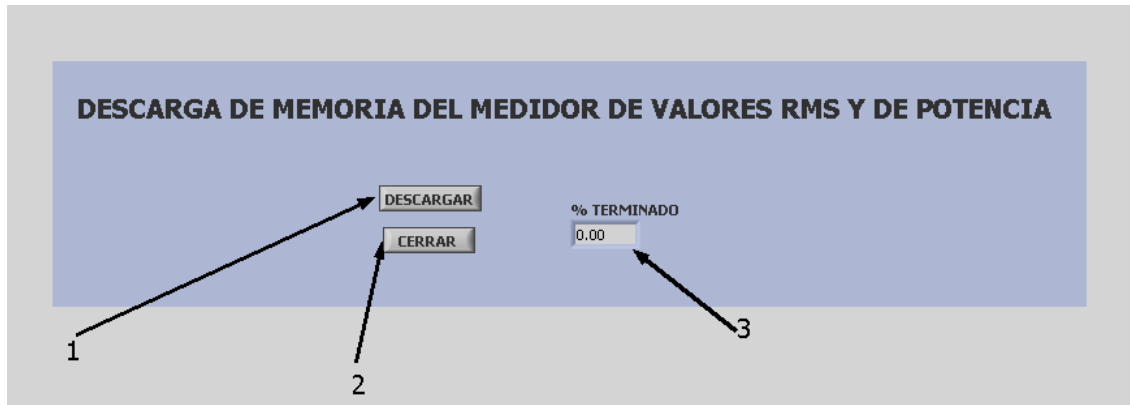


Figura A.1.30 VI de Descarga de memoria de potencia y de valores rms

PROCEDIMIENTO DE DESCARGA DE MEMORIA

1. Se presiona el botón DESCARGAR literal 1 Figura A.1.30, y aparecen dos menús emergentes, el primero solicita donde se desea guardar el archivo con formato de fecha y aparece el nombre "form-rms-pot-día-mes-año.txt"; el segundo solicita donde se desea guardar el archivo para graficar y aparece el nombre "graf-rms-pot-día-mes-año.txt". Este último es de utilidad cuando se desea utilizar los VI de lectura de archivo para una fecha, y el VI de lectura de archivo entre dos fechas.

La información del archivo viene en el siguiente formato:

P S Q VA VB VC IA IB IC HORA Y
FECHA

Donde:

P, es la potencia activa trifásica
S, es la potencia aparente trifásica
Q, es la potencia reactiva trifásica
VA, es el voltaje rms de la fase A
VB, es el voltaje rms de la fase B
VC, es el voltaje rms de la fase C
IA, es la corriente rms de la fase A
IB, es la corriente rms de la fase B
IC, es la corriente rms de la fase C

2. Se puede observar el indicador que muestra el progreso de la descarga mediante la salida % TERMINADO literal 3 Figura A.1.30.
3. Se presiona el botón CERRAR literal 2 Figura A.1.30.

4.2.5 VI DE GENERACIÓN DE REPORTE HTML

Este VI crea archivos HTML a partir de archivos de texto descargados de la memoria del medidor previamente.

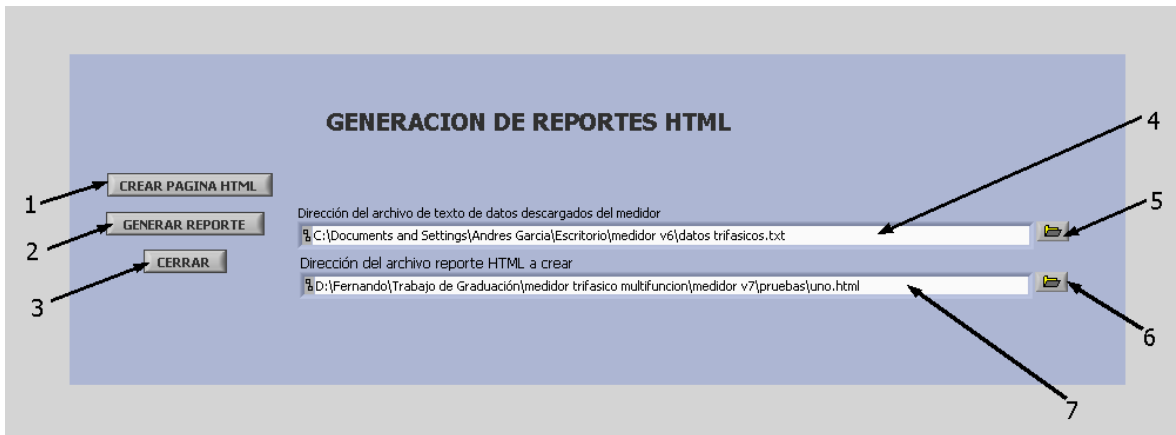


Figura A.1.31 VI de generación de reportes HTML

PROCEDIMIENTO DE GENERACIÓN DE REPORTE HTML

1. Presione el botón CREAR PAGINA HTML literal 1 Figura A.1.31, y aparecerá un menú emergente solicitando el nombre y la dirección del archivo HTML en blanco a crear.
2. Digitar el nombre y la ubicación e del archivo de texto con los datos de interés en el campo de texto de entrada literal 4 Figura A.1.31, o presionar el botón literal 5 Figura A.1.31 para cargar el archivo.
3. Digitar el nombre y la ubicación del archivo HTML que se creo en el paso 1 en el campo de texto de entrada literal 7 Figura A.1.31, o presionar el botón literal 6 Figura A.1.31 para cargar el archivo.
4. Presionar el botón GENERAR REPORTE literal 2 Figura A.1.31, luego aparecerá un menú emergente con dos botones ver Figura A.1.32. Se presiona el botón Replacé de la Figura A.1.32.

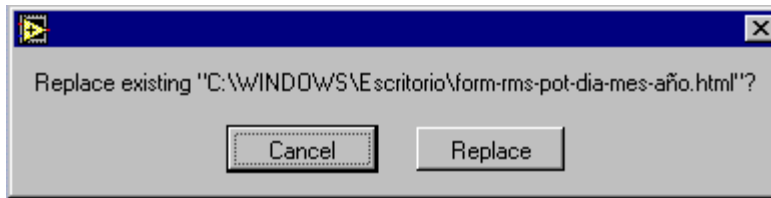


Figura A.1.32 Menú emergente para reemplazar archivo

5. Se presiona el botón CERRAR literal 3 Figura A.1.31.

4.2.6 VI DE AYUDA

Este VI aporta una breve referencia de que son los controles de cada VI utilizado en el programa principal. Ver Figura A.1.33.

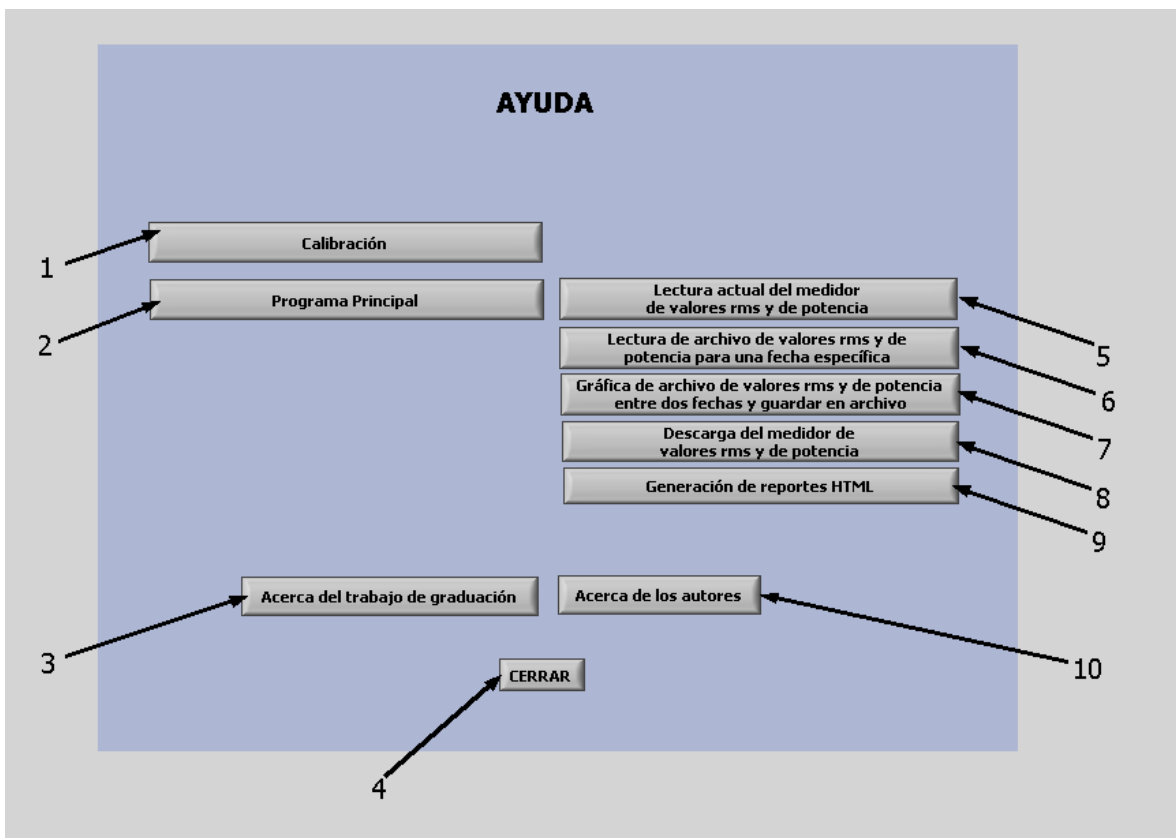


Figura A.1.33 VI de ayuda

Cuando se presiona un botón aparece un VI emergente con información acerca de la función asociada al botón presionado.

Se tienen los siguientes botones:

Calibración	literal 1 Figura A.1.33
Programa principal	literal 2 Figura A.1.33
Lectura actual de potencia y de valores rms	literal 5 Figura A.1.33
Lectura de archivo dada una fecha	literal 6 Figura A.1.33
Lectura de archivo entre dos fechas	literal 7 Figura A.1.33
Descarga de memoria	literal 8 Figura A.1.33
Generación de reportes HTML	literal 9 Figura A.1.33
Acerca del trabajo de graduación	literal 3 Figura A.1.33
Acerca de los autores	literal 10 Figura A.1.33
Cerrar	literal 4 Figura A.1.33

SECCIÓN 5: DETECCIÓN Y RESOLUCIÓN DE PROBLEMAS

¿A que niveles de voltaje, puedo conectar las entradas analógicas?

El medidor esta diseñado para que sus entradas analógicas de voltaje puedan ser conectadas a 240V AC de línea a línea, mientras que las entradas analógicas de corriente soportan una corriente de 5A por fase. Estas características deben de ser tomadas en cuenta al momento de conectar el medidor, dado que si no se respetan estos niveles el medidor sufrirá un serio deterioro.

Las lecturas de energía y potencia no son las correctas

Revise la configuración del medidor para ver si es la adecuada para el tipo deservicio al que este esta conectado.

Las lecturas de energía activa son negativas

Revise la conexión de los canales analógicos de corriente, ya que se tiene que respetar las terminales punteada (entradas de corriente) y las terminales no-punteada (salidas de corriente), revise también la conexión del canal de voltaje

¿Cómo saber si el medidor esta midiendo correctamente sin usar una PC?

Para determinar si el medidor esta funcionando en forma normal, se debe observar el led IRQ del ADE, este debe emitir un pulso cada 3 o 5 minutos según la configuración hecha en el medidor, además, el led del RELOJ estará pulsando cada 1 segundo. También es recomendable revisar la conexión de los canales analógicas de voltaje y corriente.

¿Qué hago si el led del IRQ y del RELOJ no funcionan correctamente?

Se deberán revisar las conexiones del medidor con el sistema trifásico a medir. Si las conexiones esta bien, re-inicie el medidor y verifique la configuración del medidor.

¿Cómo puedo conectar el medidor a un sistema con niveles de voltajes y corrientes que exceden los niveles especificados de las entradas analógicas del medidor?

Si se desea conectar el medidor a un sistema trifásico con niveles superiores a los especificados para los canales analógicos del medidor, se recomienda que se utilice transformadores de medición para acondicionar los niveles de voltaje y corriente.

¿Que pasa cuando la energía de alimentación de 120Vac falla?

Cuando la energía de alimentación falla, la energía de la batería de respaldo es conmutada para que el medidor siga funcionando, esta batería proporcionara energía por 2 días; después este tiempo los datos almacenados en la memoria se perderán. Para solucionar este inconveniente se debe de reparar la falla del sistema de alimentación prontamente.

¿El medidor no funciona?

Si el medidor no enciende se debe revisar que el jumper 36 este cortocircuitado, ya que este deshabilita la fuente de alimentación AC a sí como a la batería de respaldo.

SECCIÓN 6: CIRCUITOS ESQUEMÁTICOS

TARJETA DEL MEDIDOR DE ENERGIA

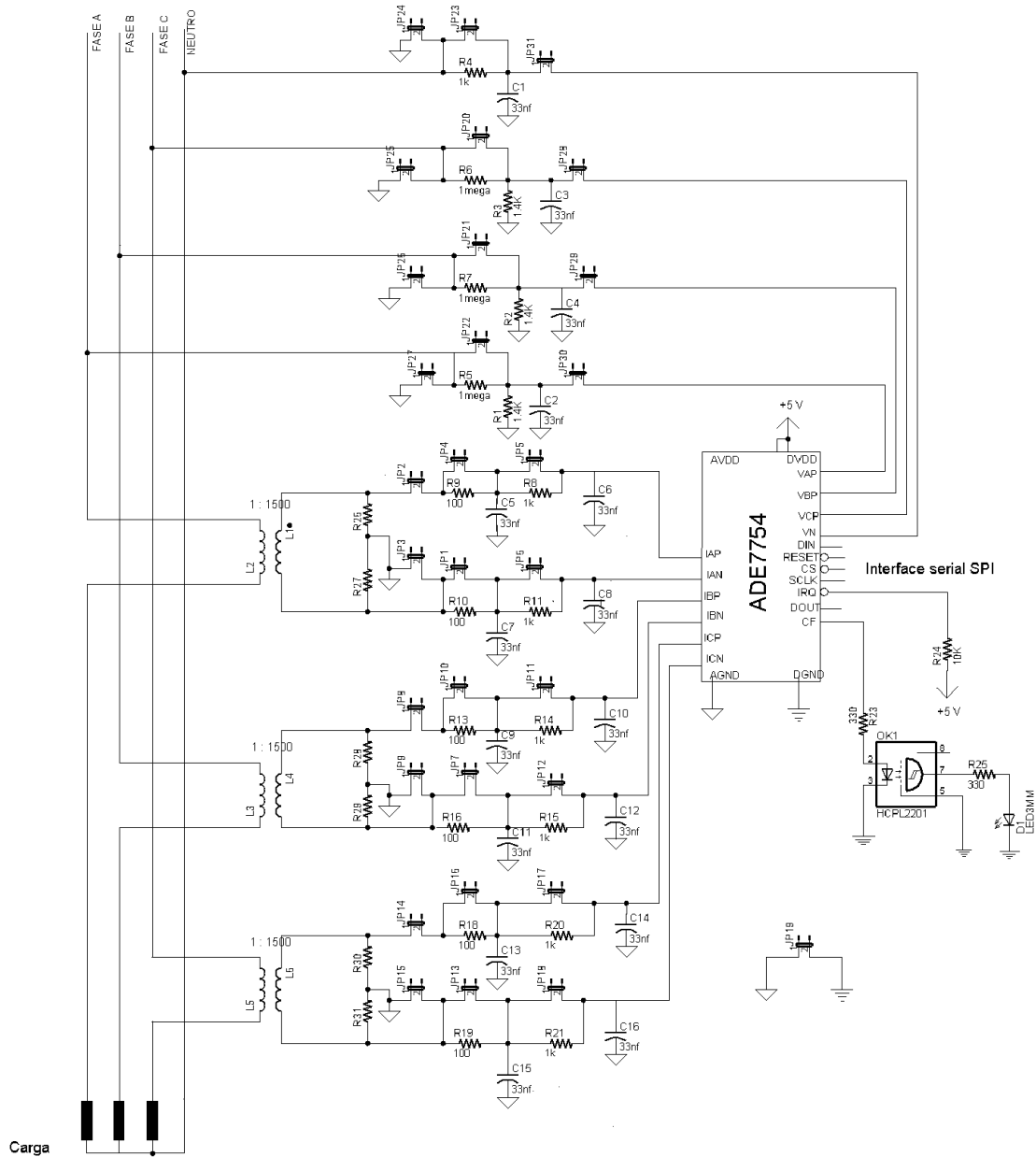


Figura A.1.30 Tarjeta del IC ADE7754

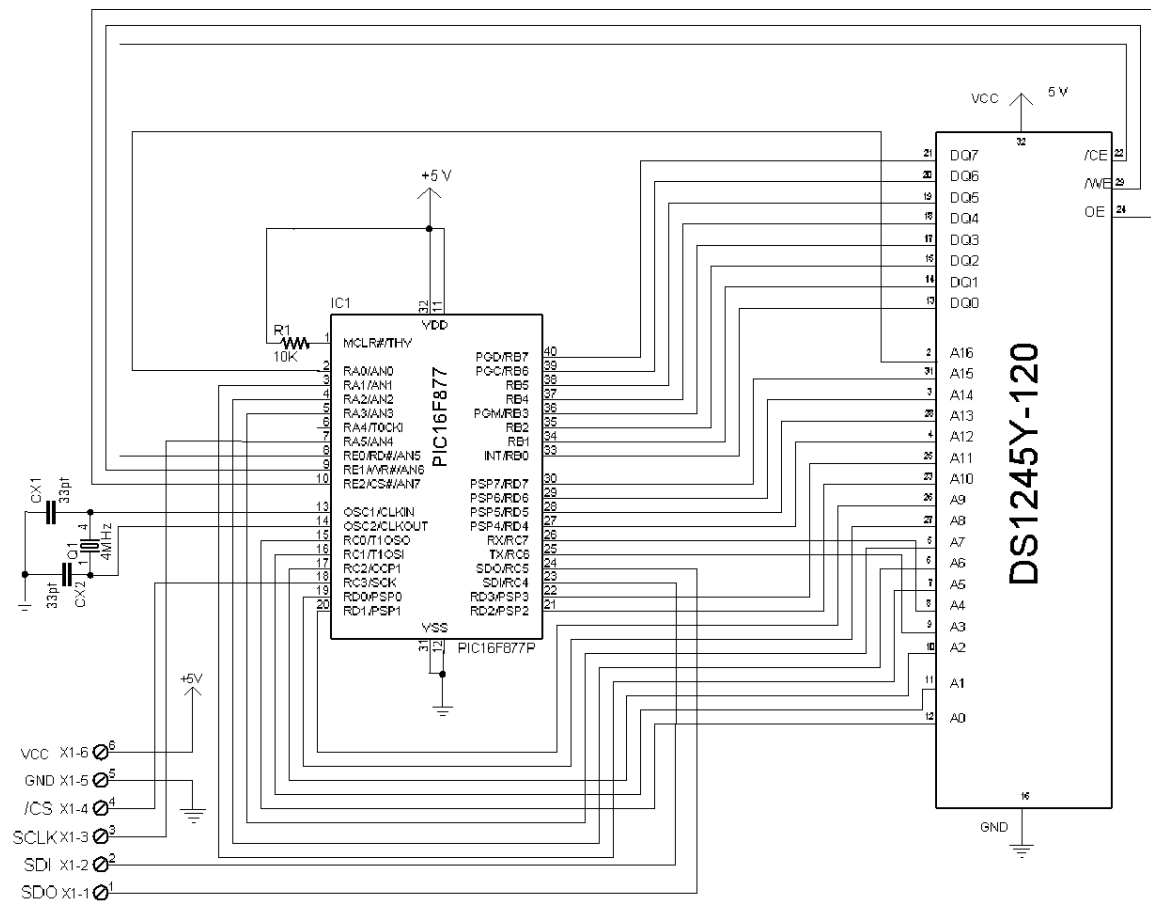


Figura A.1.31 Diagrama de implementación de una memoria con interfase SPI.

TARJETA DE TEMPORIZACION DEL MEDIDOR Y MICROCONTROLADOR MAESTRO

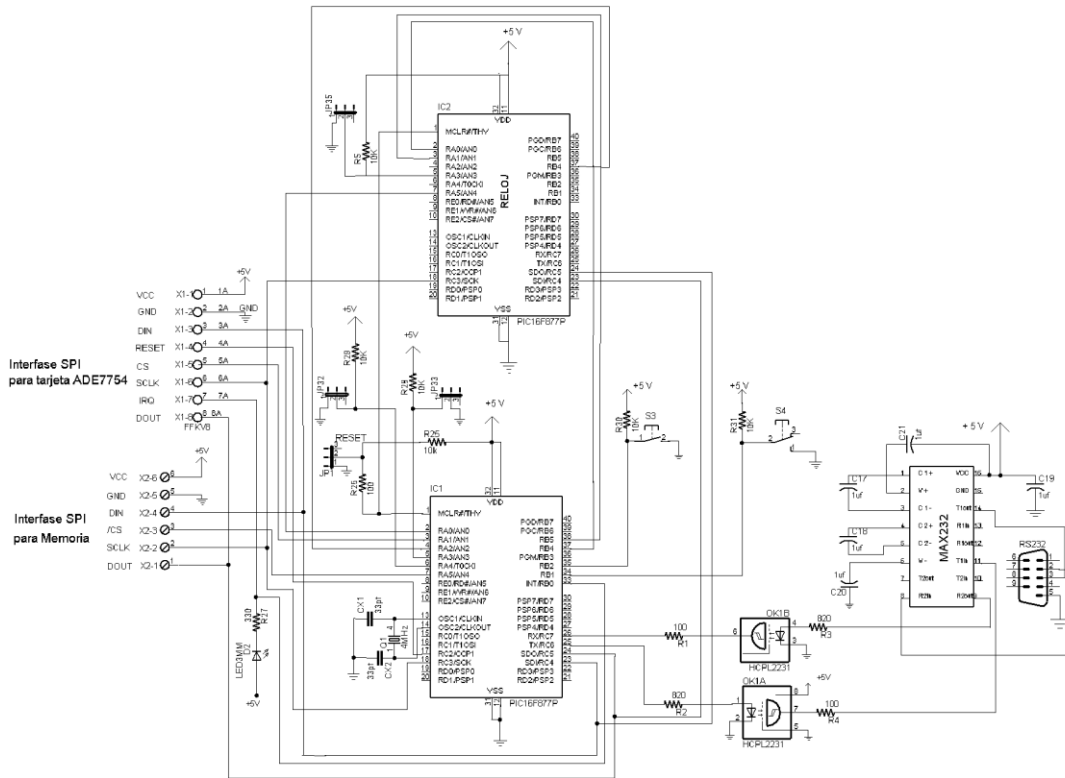


Figura A.1.32 Tarjeta de Temporización donde se incluyen también el microcontrolador maestro y el convertidor TTL-RS232

FUENTE DE VOLTAJE CON BATERIA DE RESPALDO

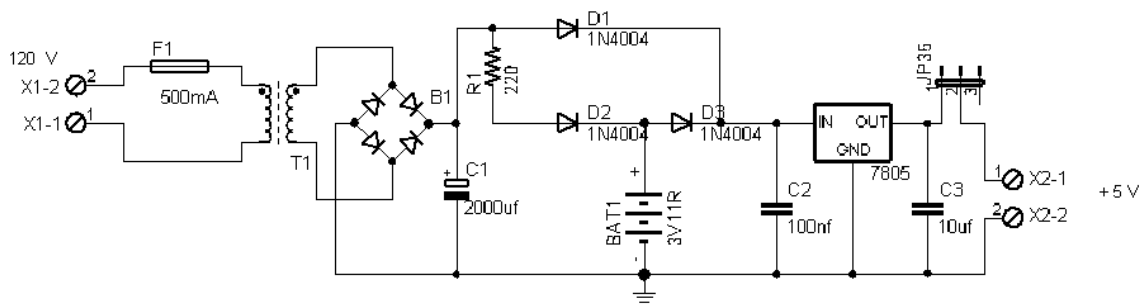


Figura A.1.33 Tarjeta de fuente de voltaje con batería de respaldo

A.2 TARJETA PROGRAMADORA DE MICROCONTROLADORES

Con esta tarjeta es posible programar la memoria Flash-EEPROM y EEPROM del PIC16F877 maestro, lugar donde residen las instrucciones que posteriormente ejecutarán los microcontroladores maestros y esclavos de manera autónoma. Ver Figura A.2.1.

Tarjeta programadora de PIC16f8XX

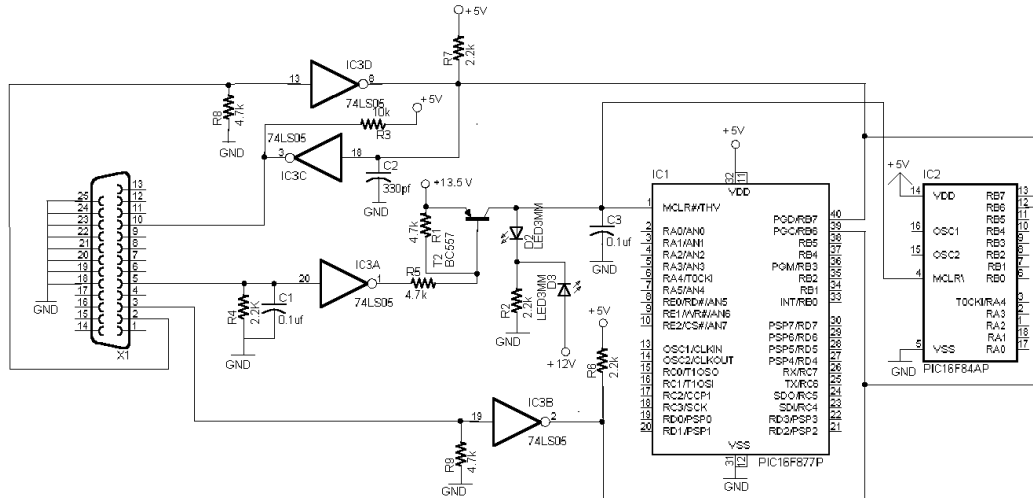


Figura A.2.1 Diagrama de la tarjeta programadora del microcontrolador PIC16F877

Este programador de PIC es del tipo serial controlado a través del puerto paralelo de la computadora y funciona para toda la familia de microcontroladores de PIC de la compañía MICROCHIP TECHNOLOGIES.

Para programar el microcontrolador es necesario efectuar los siguientes pasos:

- Compilar el código fuente .ASM con el compilador MPASM que es un software proporcionado por la compañía MICROCHIP para la plataforma Windows, para obtener el archivo ensamblado .HEX. Ver figura A.2.2.
- Descargar el archivo .HEX en el microcontrolador, mediante el programa ICPROG. Ver figura A.2.3.

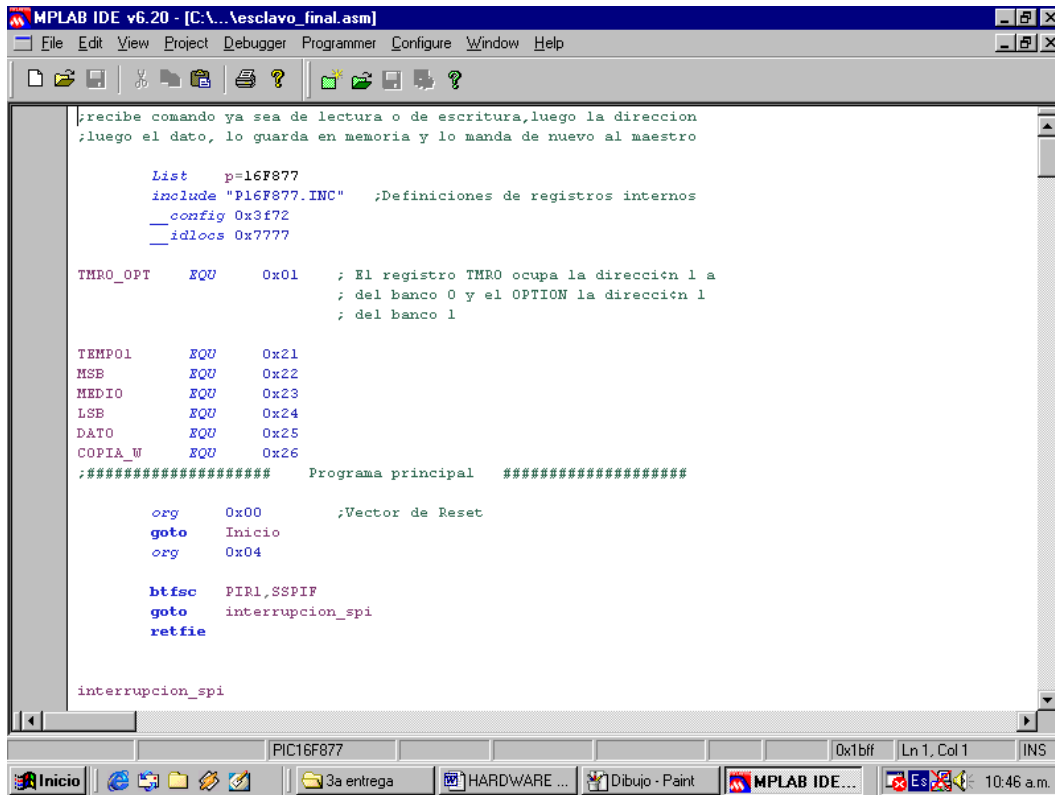


Figura A.2.2 Interfase grfica del compilador MPLAB IDE.

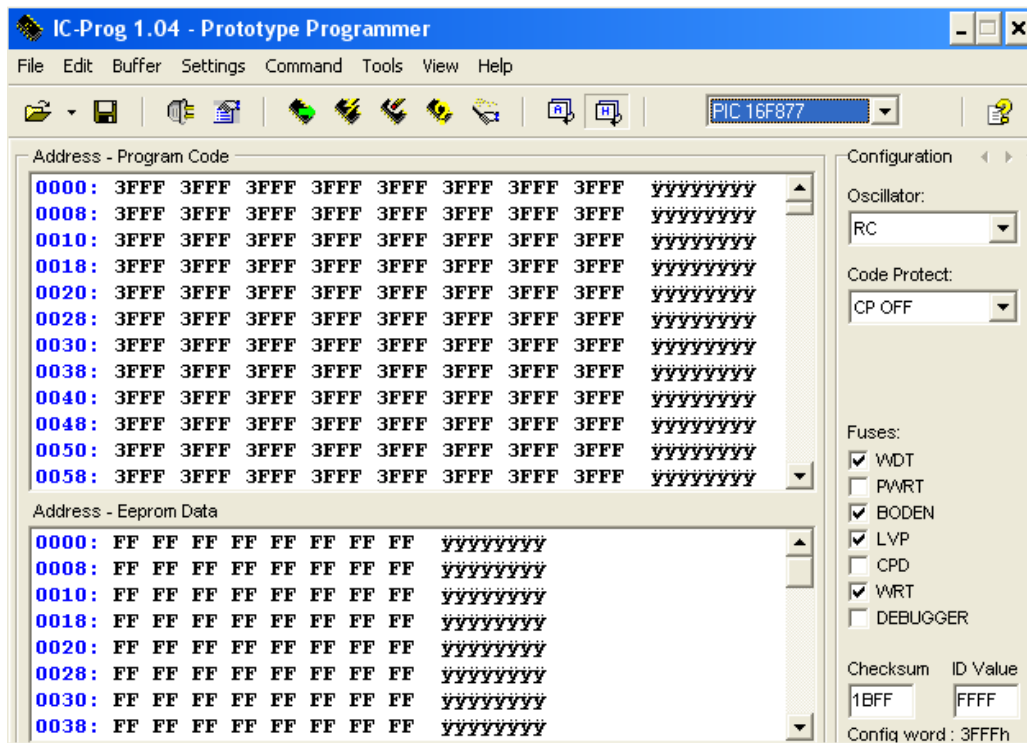


Figura A.2.3 Interfase grfica del programador de microcontroladores IC-PROG

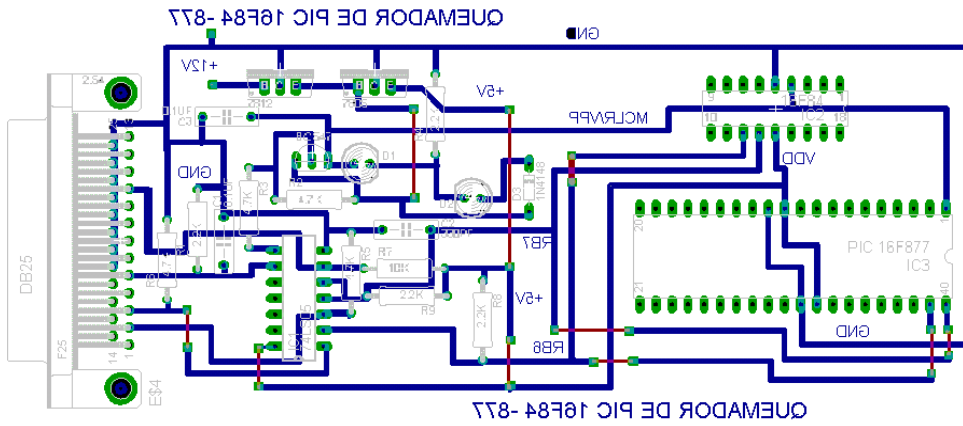


Figura A.2.4 Diagrama del impreso de la tarjeta programadora del microcontrolador PIC16F877

A.3 LISTA DE COMPONENTES Y PRECIOS

Lista de precios de los componentes de la tarjeta de memoria.				
Parte	Descripción	P. Unitario	cantidad	totales (\$)
C1	capacitor de 33pf /50V cerámico 5%	0.12	2	0.24
CN2	conector recto de 6 pines	0.28	1	0.28
IC1	microcontrolador PIC16F877/P	18.00	1	18.00
IC2	memoria SRAM DS1245Y-120	20.00	1	20.00
R1	resistencia de 10K ohm 1/4 W 1%	0.12	1	0.12
XT	cristal de 4MHz	6.00	1	6.00
	Tableta de cobre de 15x20 cm	2.00	1	2.00
	Bases de 20 pines por lado	0.46	2	0.92
P. Total tarjeta				47.56

Tabla A.3.1 Lista de precios de los componentes de la tarjeta de memoria

Lista de precios de los componentes de la tarjeta del microcontrolador ADE7754.				
Parte	Descripción	P. Unitario	cantidad	totales (\$)
C1	capacitor de cerámica de 22pF/50V	0.12	2	0.24
C2,24,26,29,30	capacitor de tantalio 10uF/16v	0.46	5	2.30
C3,C22,23,25,27,28	capacitor cerámico de 100nF/100V	0.46	6	2.76
C5,C21	capacitor de tantalio de 33nF/50V	0.46	2	0.92
CN1,CN3-CN11	conector macho 3 líneas	0.17	10	1.70
IC1	ADE7754-AR	10.00	1	10.00
JP1-JP32	jumper de 1 posición	0.12	32	3.84
LED	diodo led de 5mm	0.23	2	0.46
R1,R6,R7,R9,R40,R42	resistencia de precisión 100/1/4 W 0.1%	1.00	6	6.00
R2,R3-R5,R8,R10,R17,R41	resistencia de precisión 1K ohm/1/4 W 0.1%	1.00	8	8.00
R11-R13	resistencia de precisión 1.4K ohm/1/4 W 0.1%	1.00	3	3.00
R18	resistencia de precisión 10K ohm/1/4 W 0.1%	1.00	1	1.00
R19	resistencia de 820 ohm/1/4 W 1%	0.12	1	0.12
R38	resistencia de precisión 1M ohm/1/4 W 0.1%	1.00	1	1.00
R43	resistencia de 10 ohm/1/4 W 1%	0.12	1	0.12
R44-R49	resistencia de precisión 42.2 ohm/1/4 W 0.1%	1.00	6	6.00
XT	Cristal de 10MHz	6.00	1	6.00
P. Total tarjeta				53.46

Tabla A.3.2 Lista de precios de los componentes de la tarjeta del microcontrolador ADE7754

Lista de precios de los componentes de la tarjeta del microcontrolador maestro.				
Parte	Descripción	P. Unitario	cantidad	totales (\$)
C1-C4	capacitor de 33pF/50V de cerámica	0.12	4	0.48
C5-C9	capacitor electrolítico de 1uF/50v	0.35	5	1.75
CN1-CN6	conector recto macho	0.17	6	1.02
DB9	conector tipo DB9 macho para chasis.	0.60	1	0.6
IC1-IC2	microcontrolador PIC16F877/P	18.00	2	36
IC3-IC4	optoacoplador HCPL2232-2	9.42	2	18.84
IC5	convertidor TTL-RS232 MAX232	4.00	1	4
LED1-LED2	diodo led de 5mm	0.23	2	0.46
P1-P2	Pulsador normalmente abierto	0.46	2	0.92
R1,R4-R7,R9	resistencia de 10K 1/4 W 1%	0.12	6	0.72
R2,R10,R11	resistencia de 100 1/4 W 1%	0.12	3	0.36
R3,R8	resistencia de 330 1/4 W 1%	0.12	2	0.24
R12 , R13	resistencia de 1K 1/4 W 1%	0.12	2	0.24
RV1	potenciometro de 10Kohm	0.46	1	0.46
S1-S4	Switch	0.35	4	1.4
X1, X2	crystal de 4MHz	6.00	2	12
	tableta de cobre de 15x20 cm	2.00	1	2
	conector hembra para cable de poder	1.50	1	1.5
	cable de poder para fuente de PC	2.00	1	2
	chasis de 25 anchox25 profundidad x 8cm alto con tapadera	30.00	1	30
P. Total tarjeta				114.99

Tabla A.3.3 Lista de precios de los componentes de la tarjeta del microcontrolador maestro

Lista de precios de los componentes de la fuente de voltaje del medidor.				
Parte	Descripción	P. Unitario	cantidad	totales (\$)
C2	capacitor 2000u/16v	0.35	1	0.35
C3	capacitor 10u /16v	0.35	1	0.35
C4	capacitor 0.1u /16v	0.30	1	0.30
D5-D11	diodo 1N4001	0.12	7	0.84
IC2	regulador de voltaje de 5V 7805	0.80	1	0.80
R1	resistencia 220ohm 1/4w 1%	0.12	1	0.12
R2	resistencia 330ohm 1/4W 1%	0.12	1	0.12
B1	Batería recargable de 12V 1.2Ah	28.50	1	28.50
T1	transformador de 120/12v 0.5Amp	4.50	1	4.50
	tarjeta impresa perforada	1.30	1	1.30
P. Total tarjeta				37.18

Tabla A.3.4 Lista de precios de los componentes de la fuente de voltaje del medidor

Precio total tarjeta de memoria	\$ 47.56
Precio total tarjeta del microcontrolador ADE7754	\$ 53.46
Precio total tarjeta de microcontrolador maestro	\$ 114.99
Precio total tarjeta de fuente de voltaje del medidor.	\$ 37.18
Precio total de los componentes del medidor.	\$ 253.19

Tabla A.3.5 Precio total de los componentes del medidor

A.4 CIRCUITOS IMPRESOS DE LAS ETAPAS

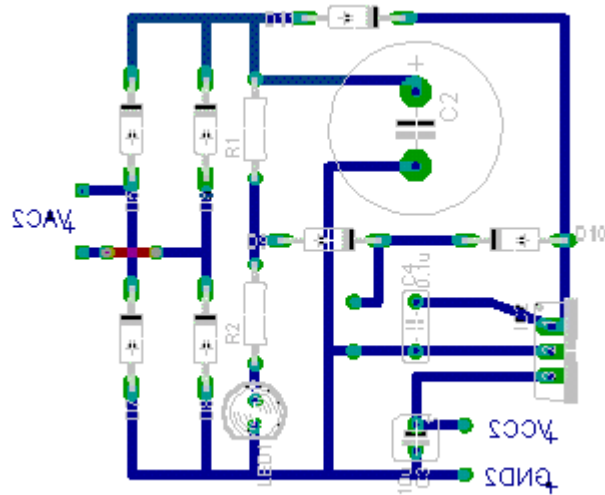


Figura A.4.1 Circuito impreso de fuente de voltaje tipo UPS.

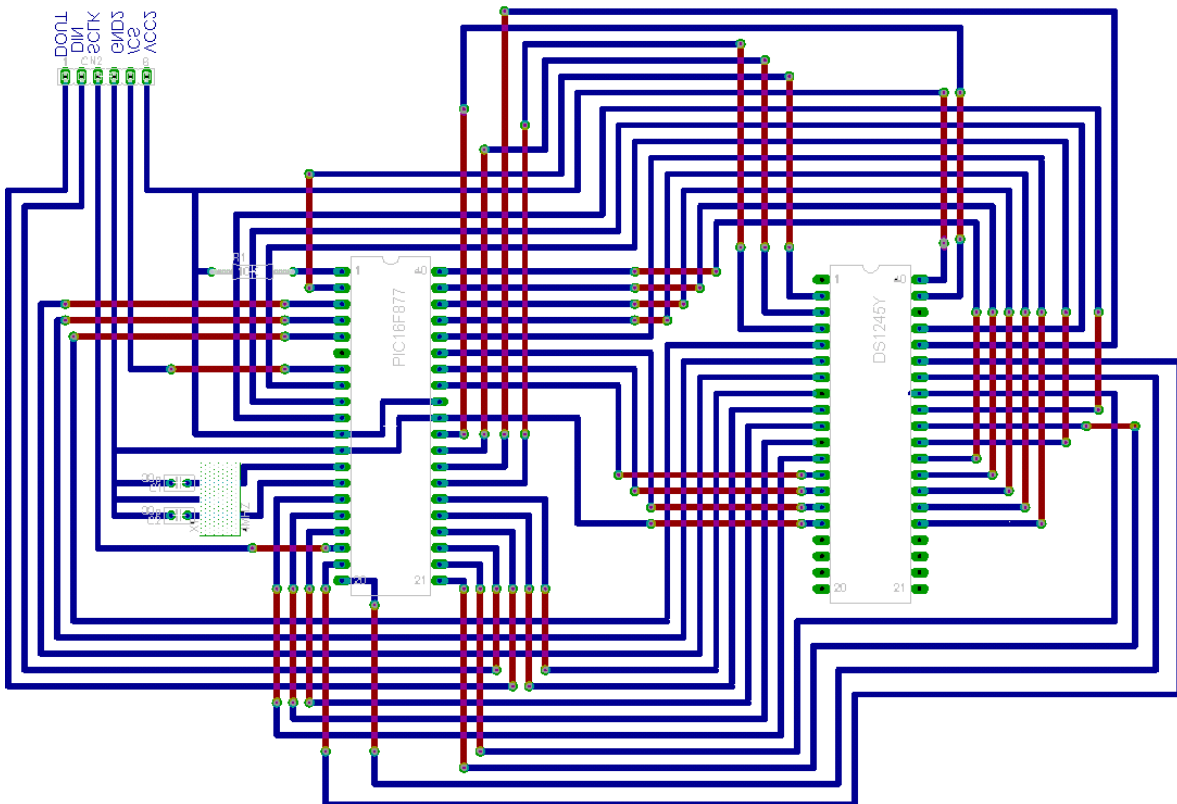


Figura A.4.2 Circuito impreso de microcontrolador memoria.

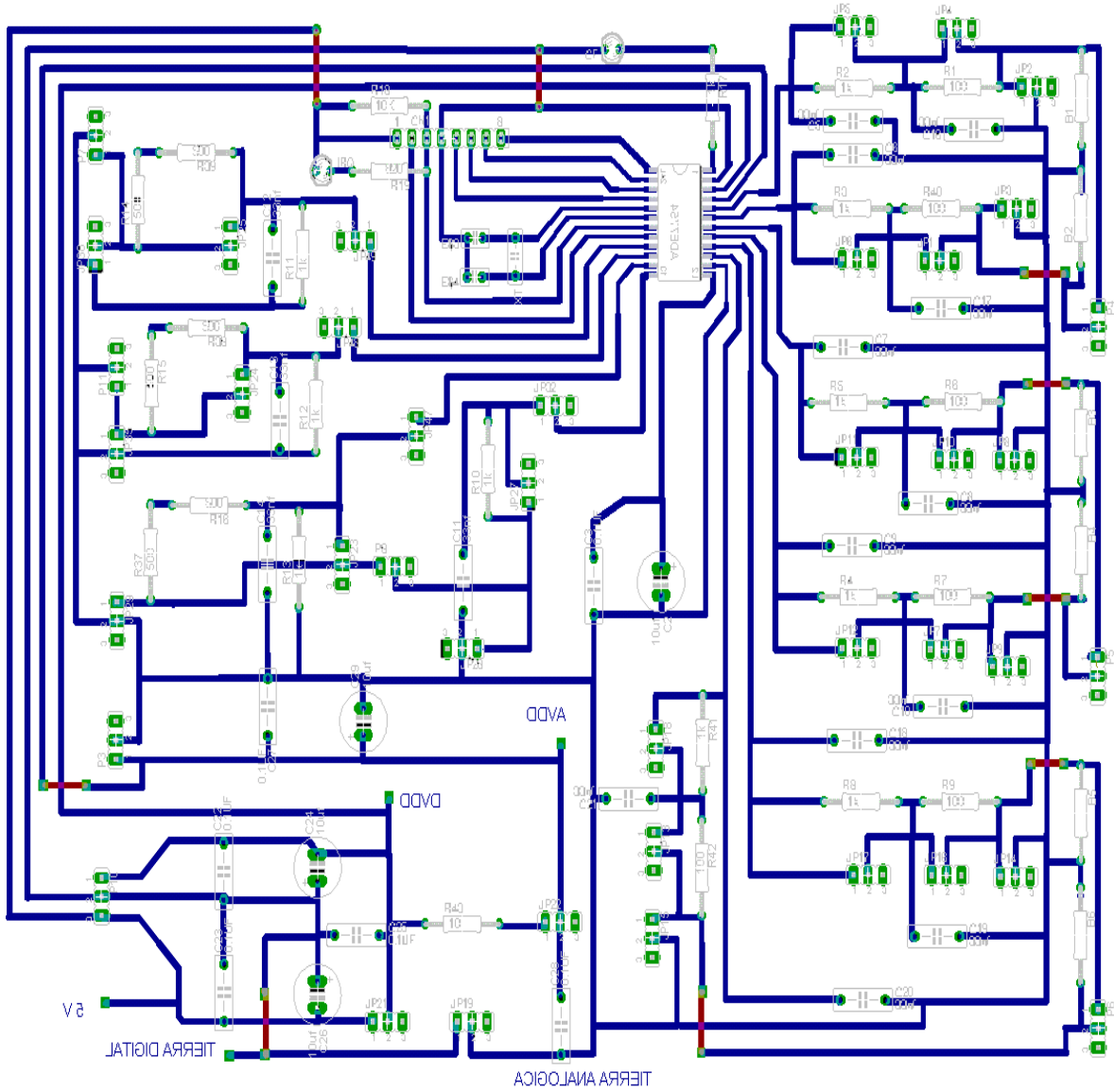


Figura A.4.3 Circuito impreso de microcontrolador ADE7754

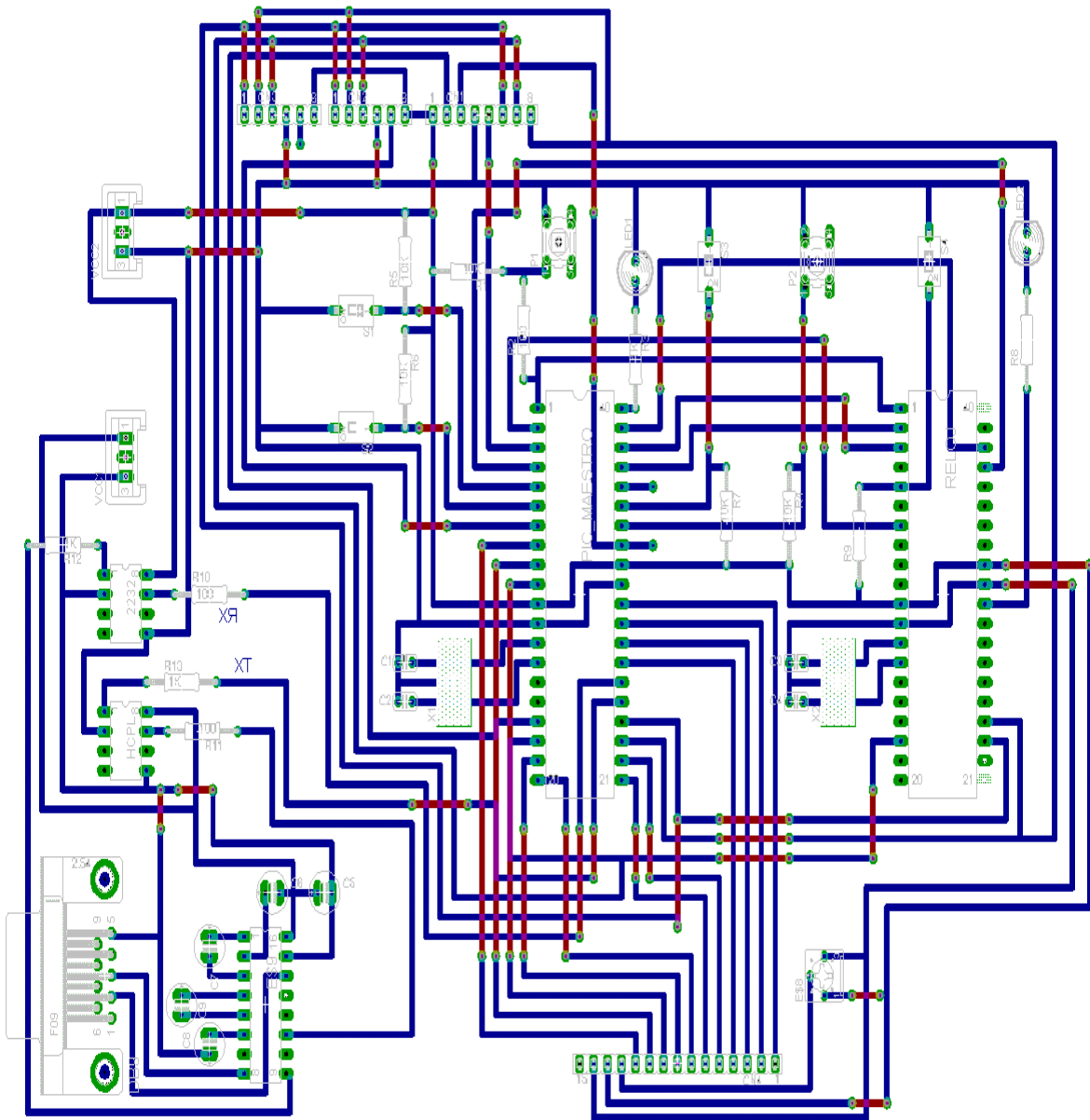


Figura A.4.4 Circuito impreso tarjeta microcontrolador maestro.

A.5 PROCESO DE FABRICACIÓN DE LOS CIRCUITOS IMPRESOS SOBRE TARJETAS DE COBRE

1. DISEÑO DEL CIRCUITO ESQUEMATICO EN EAGLE

El diseño de los circuitos esquemáticos es sencillo utilizando el programa EAGLE. Este programa posee una interfase gráfica con la cual cada circuito integrado (IC), elementos pasivos como resistencias, capacitores, inductores se pegan sobre una cuadrícula, y luego con líneas se forman lo que serían las pistas que conectan los pines de los IC con resistencias o cualquier otro elemento.

Este programa contiene una amplia biblioteca que contiene varios tipos de IC, de elementos pasivos así como activos.

2. DISEÑO DEL CIRCUITO IMPRESO EN EAGLE

El diseño del impreso se puede realizar utilizando el EAGLE, de dos maneras diferentes:

- Creando el impreso a partir de un circuito esquemático hecho previamente.
- Creando el impreso directamente

Si se hace con el primer método, se debe ordenar cada pista del impreso hasta obtener el impreso deseado; si se hace con el segundo método, el impreso se hace de manera similar que cuando se hace el circuito esquemático, a diferencia que cuando se selecciona un IC no aparecen las funciones que realiza cada pin, sino que aparece el IC con el número de pines que tiene, a lo cual sólo se debe unir cada pin como se haría físicamente.

Cabe destacar que se puede trabajar a doble pista con este programa, porque las pistas en el lado frontal son de color rojo y las pistas que van en la parte posterior con azul; a la hora de imprimir la pista es importante desactivar, las pistas de la parte posterior, si se desea imprimir la parte frontal, y viceversa.

3. LA IMPRESIÓN DEL IMPRESO SOBRE EL ACETATO

Cuando se tiene el diseño del impreso listo, se procede a imprimir teniendo especial cuidado de imprimir las pistas frontales y las posteriores por separado si se trabaja a doble pista.

Se recomienda imprimir directamente desde EAGLE, sobre un acetato con impresión láser a blanco y negro; pero sino se dispone de un impresor láser, se hace la impresión en papel bond habilitando la opción de espejo en EAGLE, y luego se saca una fotocopia de esta impresión en acetato.

4. LA IMPRESIÓN DEL ACETATO SOBRE LA TARJETA DE COBRE

En esta etapa pueden haber dos casos:

- En el primero, se utiliza una tarjeta de cobre normal con una cara o doble cara.
- En el segundo, se dispone de una tarjeta de cobre presensibilizada a la luz.

En el primer caso, se debe lijar la superficie de la tarjeta de cobre para eliminar impurezas, luego cuando se tiene el acetato se procede a la etapa de planchado. Se coloca el lado con tinta del acetato en contacto con la placa de cobre. Se coloca un pañuelo sobre el acetato y se calienta con una plancha de ropa hasta que la tinta del acetato se impregna sobre la tarjeta de cobre.

En el segundo caso, se coloca el acetato sobre la tarjeta presensibilizada, y a continuación se deposita sobre la máquina de rayos ultravioleta por quince minutos.

5. ELIMINANDO LAS IMPUREZAS DE LA TARJETA DE COBRE

Cuando se tiene el diseño del impreso sobre la tarjeta de cobre, se procede a introducir la tarjeta en una solución de percloruro con agua, de aproximadamente 1 litro de agua por 2 onzas de percloruro en un recipiente de plástico. Se deja reposar la tarjeta entre 30 minutos y una hora, o hasta que sólo queden las pistas de cobre del impreso. Si se desea más rapidez en esta etapa, se puede agregar más percloruro a la solución o agitar el recipiente de plástico constantemente.

Cuando se tienen las pistas de cobre únicamente, se procede a limpiar la tarjeta con agua. Luego se procede a abrir todos los agujeros de los IC, de los elementos pasivos. Después se lijan las pistas, y finalmente se limpia la tarjeta con solvente.

6. SOLDADURA DE LOS ELEMENTOS

Cuando la tarjeta ya está limpia, se procede a estañar los extremos de los elementos como resistencias y capacitores. Y luego se pueden soldar los elementos a la tarjeta.

A.6 GLOSARIO

Chip-Select pin que selecciona o deselecta un chip determinado, de entre todos los chips conectados a los buses de un sistema.

Comunicación asíncrona sistema de comunicación serial que no utiliza señal de sincronismo, pero implementa un protocolo a través de una serie de bit que son transmitidos con la información o el dato enviado.

Comunicación síncrona sistema de comunicación serial en donde los datos de envío y recepción de la comunicación son sincronizados utilizando un reloj que cronometra con precisión el tiempo que separa cada bit.

Comunicación full duplex característica de un sistema o dispositivo de comunicación de poder enviar y recibir datos al mismo tiempo.

Encapsulado SOIC Abreviatura de small outline IC (IC pequeño fuera de línea), un empaquetado de chips de montaje de superficie rectangular de plástico de pequeñas dimensiones que utiliza patas en forma de ala de gaviota que se extienden hacia fuera.

Encapsulado DIP Acrónimo de Dual Inline Package. Tipo de encapsulado consistente en un chip con dos filas de pines de conexión en cada lado

Fuente de energía de litio batería que utiliza litio en el electrodo negativo, sulfuro de metal en el electrodo positivo, y sal fundida en el electrolito

Instrumentos Virtuales (VI) instrumento personalizado; desarrollado con un lenguaje de programación gráfico, partiendo de un conjunto de características de un medidor real.

Memoria FLASH memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos energía y es mas pequeña. Es más rápida y de mayor densidad que la EEPROM.

Memoria EEPROM (Electrical Erasable Programmable Read OnIy Memory). memoria de sólo lectura, programable y bórrale eléctricamente

Memoria RAM memoria de tipo volátil, retiene información solo mientras esta energizada, el acceso a este tipo de memoria es muy rápido.

Memoria ROM memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip.

Memoria SRAM Abreviatura de static random-access memory (memoria estática de acceso aleatorio). El almacenamiento en RAM estática se basa en circuitos lógicos denominados flip-flop, que retienen la información almacenada en ellos mientras haya energía suficiente para hacer funcionar el dispositivo. Es normalmente mucho más rápida que la RAM dinámica. También es más cara, por lo que se reserva generalmente su uso en la memoria de caché.

Microcontrolador es un circuito integrado que incorpora una unidad central de procesamiento (CPU) y una serie de recursos internos, que le permiten funcionar como un computador básico.

Error de Medición define la diferencia entre el valor obtenido de la medición y el valor verdadero de la magnitud medida, después que todas las correcciones han sido hechas en el dispositivo usado para realizar la medición.

Periodo de la línea periodo de la señal de voltaje aplicado en dicha línea

Programación gráfico programación usando iconos o dibujos que en este lenguaje de programación realizan una acción determinada, las líneas de código o comandos convencionales son sustituidos por iconos.

Resistencia de Burden transductor de corriente a voltaje usado generalmente en el secundario de los transformadores de intensidad en un medidor.

Transductor un instrumento que convierte una forma de energía en otra o una cantidad física en otra.

Tecnología CMOS siglas para Semiconductor Complementario Óxido Metálico, una tecnología de semiconductores usada en muchos circuitos integrados. La tecnología CMOS es una extensión de la tecnología MOS.

SPI (Serial Peripheral Interfase) es un sistema de comunicación sincrónico, el cual utiliza una señal de reloj para establecer un sincronismo entre dos módulos SPI.

SCI (Serial Comunicación Interfase) es un receptor-transmisor universal asíncrono (UART), el cual no necesita de una señal de sincronismo (reloj), para establecer una comunicación. es un sistema de comunicación asíncrono,

Protocolo normas a seguir en una cierta comunicación; como por ejemplo el formato de los datos que debe enviar el emisor, cómo debe ser cada una de las respuestas del receptor, etc.

A.7 INTERFAZ DEL PUERTO SERIAL ASÍNCRONO SCI

1. INTRODUCCION

La transmisión de datos en serie es una de las más comunes para aquellas aplicaciones en las que la velocidad no es demasiado importante. Para simplificar el proceso de enviar los bits uno por uno han surgido circuitos integrados que realizan esta función.

Las comunicaciones serie se utilizan para enviar datos a través de largas distancias, ya que las comunicaciones en paralelo exigen demasiado cableado para ser operativas. Los datos serie recibidos desde un módem o otros dispositivos son convertidos a paralelo gracias a lo cual pueden ser manejados por el bus del PC.

Los equipos de comunicaciones serie se pueden dividir entre simplex, half-duplex y full-duplex. Una comunicación serie simplex envía información en una sola dirección. Half-duplex significa que los datos pueden ser enviados en ambas direcciones entre dos sistemas, pero en una sola dirección en un tiempo específico. En una transmisión full-duplex cada sistema puede enviar y recibir datos al mismo tiempo.

La línea que transmite los datos en serie está inicialmente en estado alto. Al comenzar la transferencia, se envía un bit a 0 ó **bit de inicio**. Tras él irán los 8 **bits de datos** a transmitir (en ocasiones son 7, 6 ó 5): estos bits están espaciados con un intervalo temporal fijo y preciso, ligado a la velocidad de transmisión que se esté empleando. Tras ellos podría venir o no un bit de paridad generado automáticamente por la UART. Al final, aparecerá un bit (a veces un bit y medio ó dos bits) a 1, que son los bits de parada o **bits de stop**.

Lo de medio bit significa que la señal correspondiente en el tiempo a un bit dura la mitad; La presencia de bits de inicio y parada permite sincronizar la estación emisora con la receptora, haciendo que los relojes de ambas vayan a la par.

A la hora de transmitir los bytes de datos unos tras otros, existe flexibilidad en los tiempos, de ahí que este tipo de comunicaciones se consideren **asíncronas**.

2. PROPIEDADES DEL HARDWARE

Los dispositivos que usan la comunicación serial se dividen en dos categorías estas son:

DCE (Data Communications Equipment) y DTE (Data Terminal Equipment). DCE son dispositivos como un módem, DTE es una Computadora Personal.

Las características eléctricas del puerto serial esta contenida en la EIA (Electronics Industry Association) RS232C standard. Este establece que las señales tienen características:

1. Un " Space" (cero lógico) se establece entre +3 y +25 Volts.
2. una "Mark" (uno lógico) se establece entre -3 y -25 Volts.
3. La región entre +3 y -3 volts esta indefinida.
4. Un voltaje de circuito abierto no debe de exceder nunca 25 V. (con referencia al GND)
5. Una corriente de corto circuito no debe de exceder nunca 500mA.

Lo anterior no cubre todo la lista de la EIA Standard. Para mas información por favor vea el EIA RS232-E Standard. Las ultimas revisiones del Standard, EIA-232D & EIA-232E fue realizada en 1987 & 1991 respectivamente.

Los conectores para el Puerto serial esta en dos tamaños, esta el DB 25 pin y el DB9 pin ambos son machos en la parte trasera de una Computadora Personal. La siguiente tabla muestra los pines de conexión de cada conector.

D-Type-25 Pin No.	D-Type-9 Pin No.	Abbreviation	Full Name
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

Tabla A.7.1 Pinado de los conectores seriales DB9 Y DB 25

Funciones de los pines

TD: Salida serial de datos (TXD)

RD: Serial de dato serial Data Input (RXD)

CTS: Esta línea indica que el módem está listo para intercambiar datos. El ordenador empezara a enviar datos al DCE.

CD: Cuando el MÓDEM detecta un "Carrier" de el MÓDEM en el otro extremo de la línea telefónica. La línea se activa.

DSR: Este le dice la UART que el módem esta listo para establecer un enlace

DTR: Este es lo opuesto a DSR. Este le dice al módem que el UART listo para un enlace.

RTS: Esta línea informa al módem que el UART esta listo para intercambiar datos.

RI: Esta línea se activa cuando el módem detecta una señal ringing de la PSTN.

3. NULL MÓDEM

Un Null Módem es usado para conectar dos DTE's juntas. Esto es comúnmente usado como un forma barata de transferir archivos entre dos Computadoras Personales usando un protocolo dado.

Este puede también ser usado en un sistema de desarrollo para microcontroladores.

En la siguiente figura se muestra la forma de construir el Null módem.

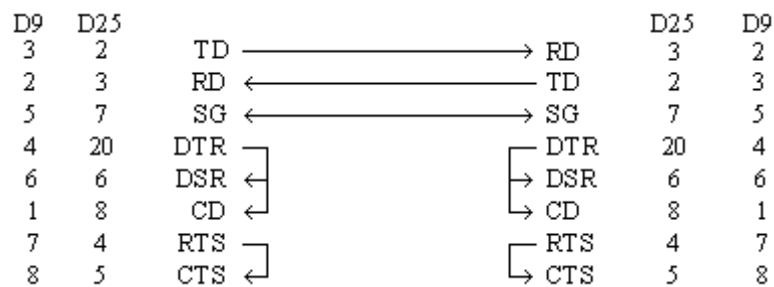


Figura A.7.1. Diagrama del alambrado Null Módem

A.8 INTERFAZ PERIFÉRICA SERIAL SPI

1. INTRODUCCIÓN

La interfaz periférica serial SPI (Serial Peripheral Interface), es un subsistema serial independiente de comunicaciones que permite a un MCU comunicarse sincrónicamente con dispositivos periféricos, tales como:

- Drivers de display de cristal líquido (LCD)
- Subsistemas convertidores de analógico a digital(A/D)
- Otros microprocesadores
- Reloj en tiempo real

La SPI es también capaz de establecer comunicaciones entre microprocesadores en un sistema con múltiples maestros. El sistema SPI puede ser configurado como un dispositivo maestro o como uno esclavo en un tiempo dado.

2. LINEAS DE DATOS Y DE CONTROL DE LA SPI

La SPI requiere dos líneas de control (CS y SCLK) y dos de datos (SDI y SDO). Motorola llamo a estas líneas MOSI (Master-Out-Slave-In) y MISO (Master-In-Slave-Out). La línea chip select es llamado SS (Slave-Select).

La función de cada línea es:

MOSI (Master-Out-Slave-In): Línea por donde circulan los datos que el maestro quiere enviar a los esclavos, por tanto será la señal de salida de datos de la unidad que funciona como maestro y la señal de entrada para los esclavos.

MISO (Master-In-Slave-Out): Línea por la cual viajan los datos que son enviados desde algún esclavo hacia el maestro, de esta forma será una señal de entrada para el maestro y las respectivas salidas para los esclavos.

SCLK (Serial Clock): Representa la señal de reloj con la que se producen la sincronización de las comunicaciones. Esta señal es generada solo por el maestro, por tanto para los esclavos en una señal de entrada mientras que para el maestro será una salida.

SS (Slave-Select): Esta línea es usada para seleccionar la unidad esclava con la que se quiere establecer una comunicación y deshabilitar la unidades esclavas restantes. La **SS** representa la entrada chip-enable de un modulo esclavo, En el maestro esta línea se puede utilizar para propósitos generales.

3. PROTOCOLO

Cuando el maestro tiene que mandar un mensaje a un modulo esclavo, debe proceder a realizar una selección del mismo, a través de los #SS, como si se trataran de chip-enable. De esta forma, al ser activado el esclavo, recibe el dato manteniendo el sincronismo a través de la señal de reloj continua generada por el maestro. Es posible que cuando un esclavo sea activado con el fin de recibir un dato, este desee enviar un dato al maestro. Esto será posible mientras su línea de activación (#SS) y la señal de reloj estén presentes, de modo que la transmisión se efectuara simultáneamente en los dos sentidos.

4. ENLACES USANDO SPI

Para activar el modulo SPI de un esclavo es necesario disponer del control sobre las líneas respectivas **SS**. Para una comunicación entre dos MCU o simplemente un MCU y un periférico, el control de la línea **SS** es sencilla ya que es posible conectar directamente la línea **SS** del esclavo a un pin entrada/salida, configurado como salida en el maestro, como se muestra en la figura A.8.1.

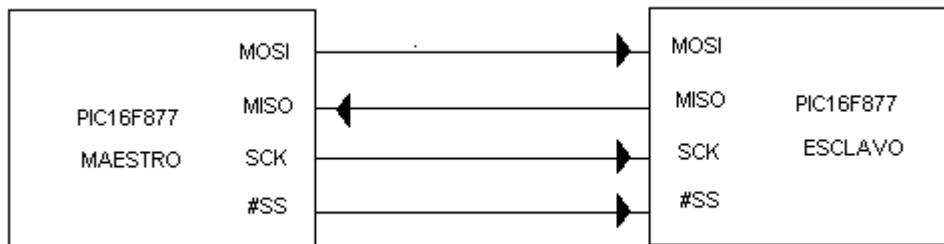


Figura A.8.1. Comunicación SPI entre dos Microcontroladores

Cuando se requiere comunicación SPI entre varios dispositivos, como un MCU con varios periféricos y/o con varios MCUs, la lógica de selección de un esclavo en particular se puede hacer por hardware externo o utilizando pines entrada/salida en la MCU maestro (selección por software), como se muestra en la figura A.8.2.

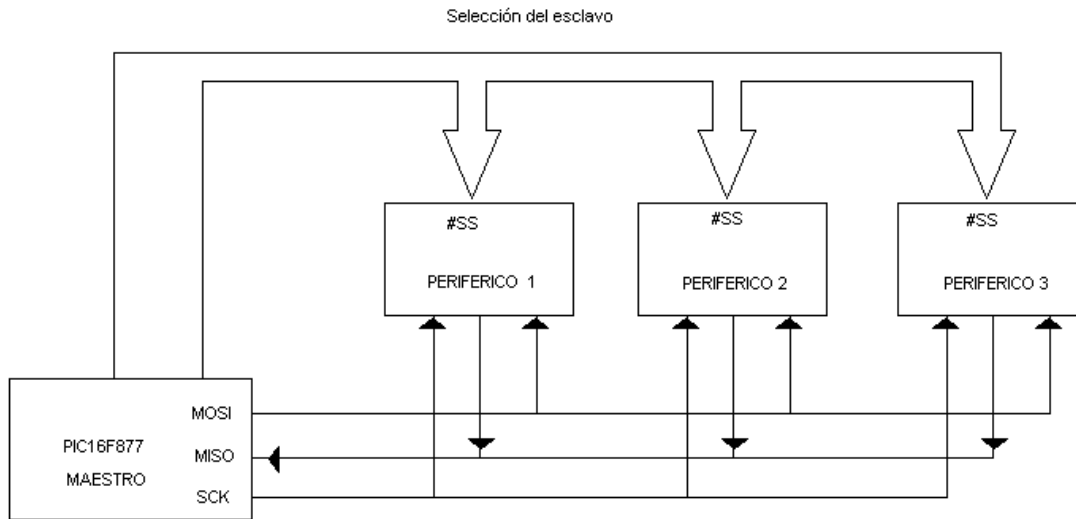
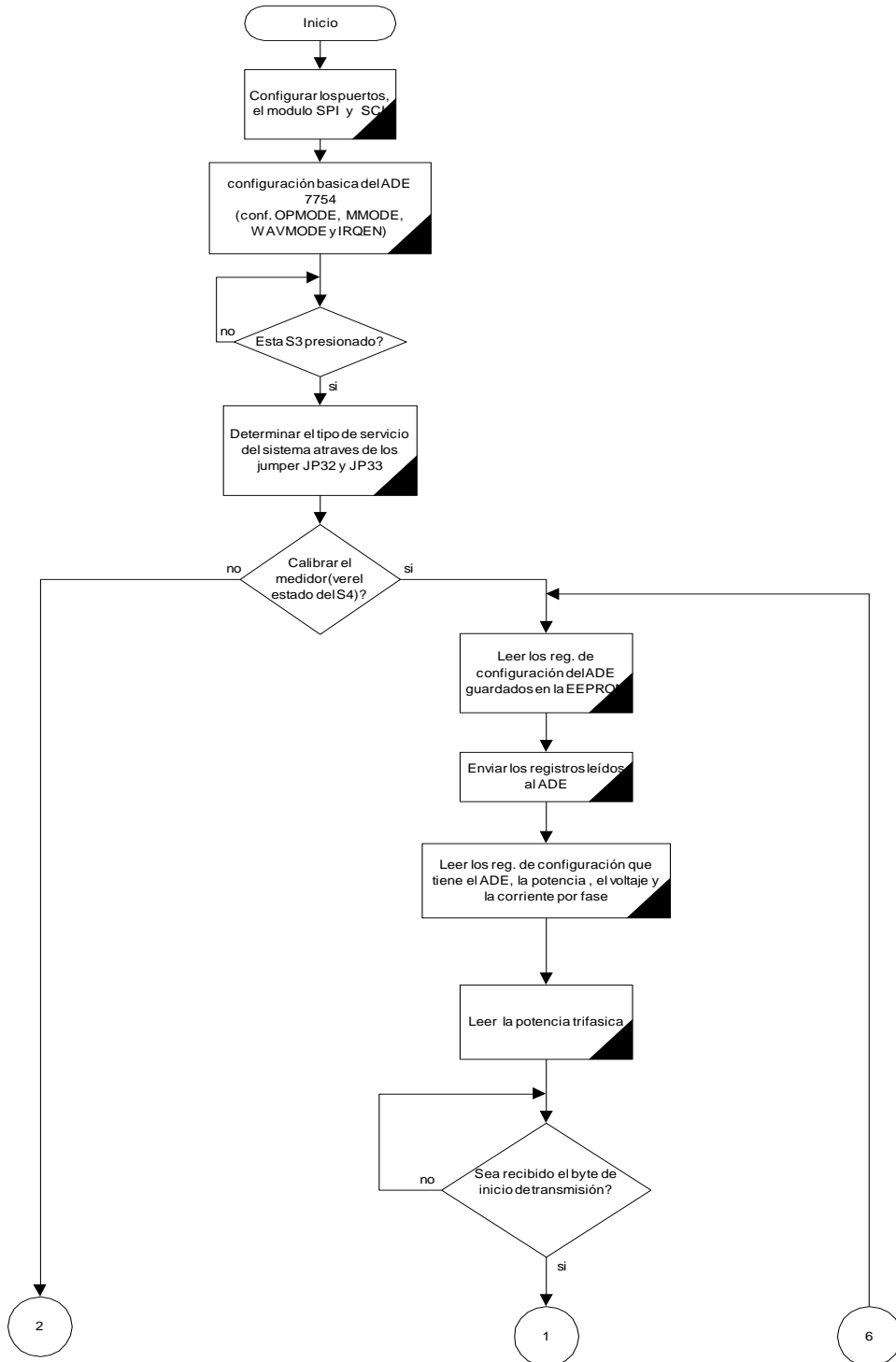
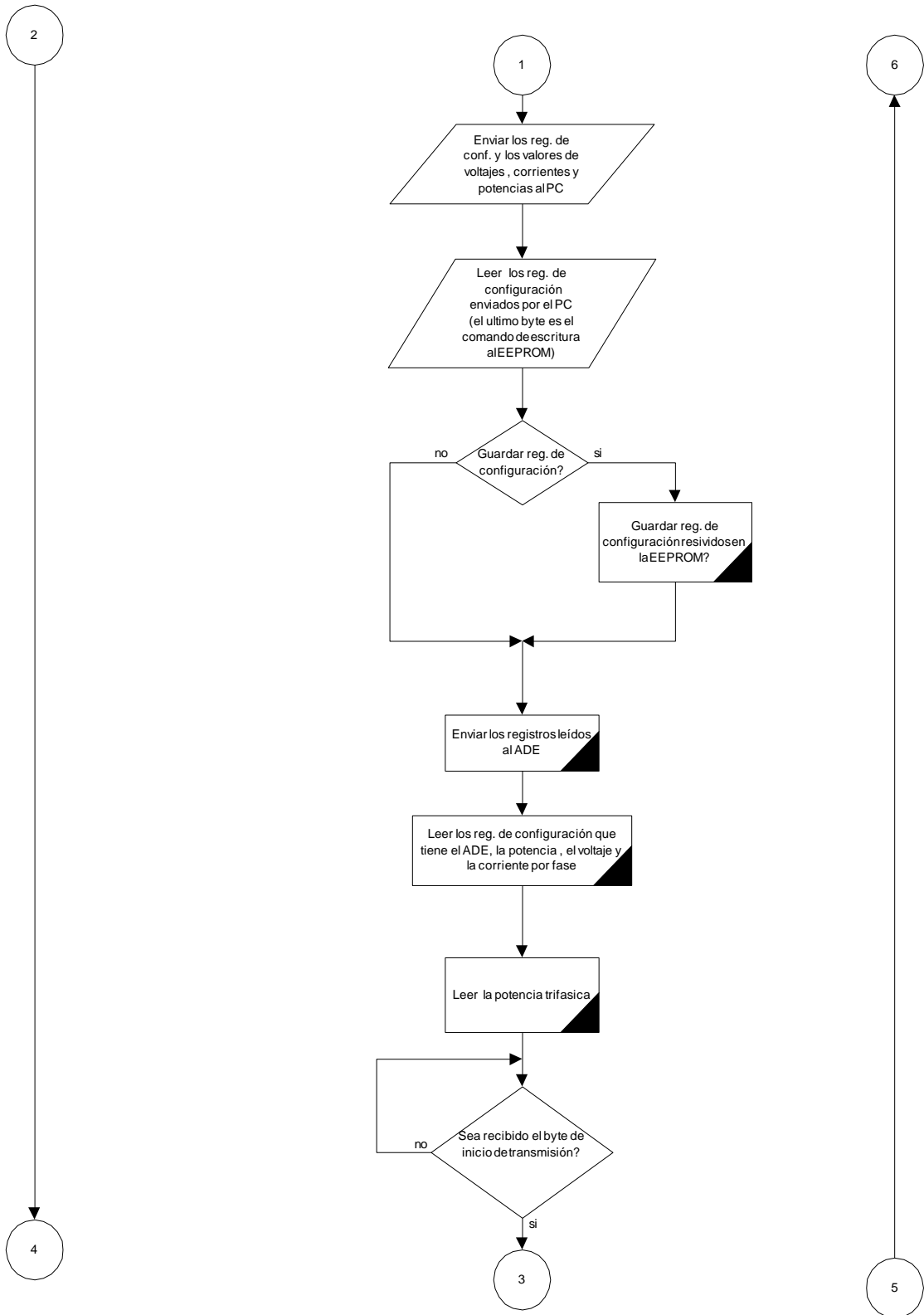


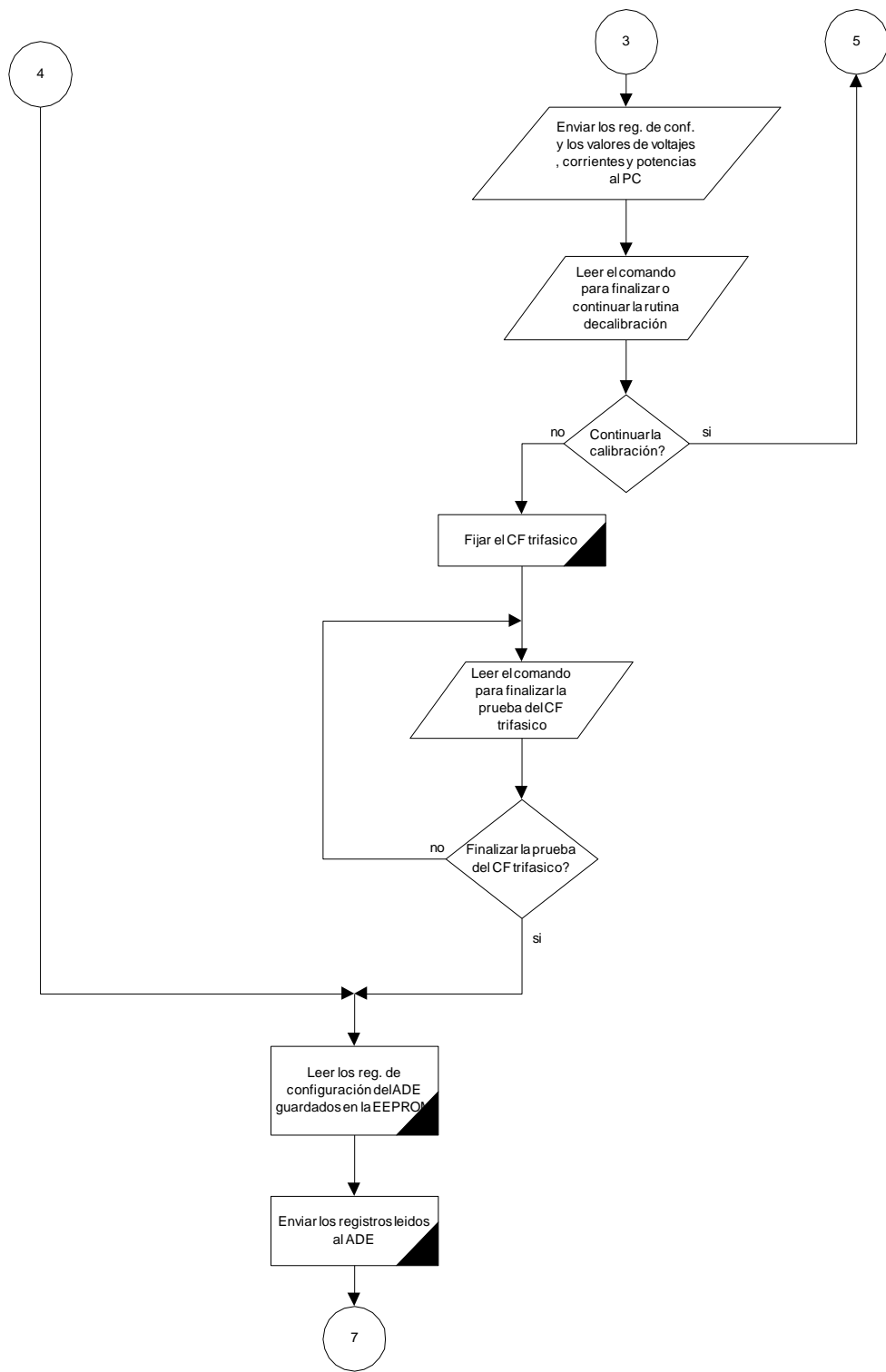
Figura A.8.2. Comunicación SPI entre varios periféricos

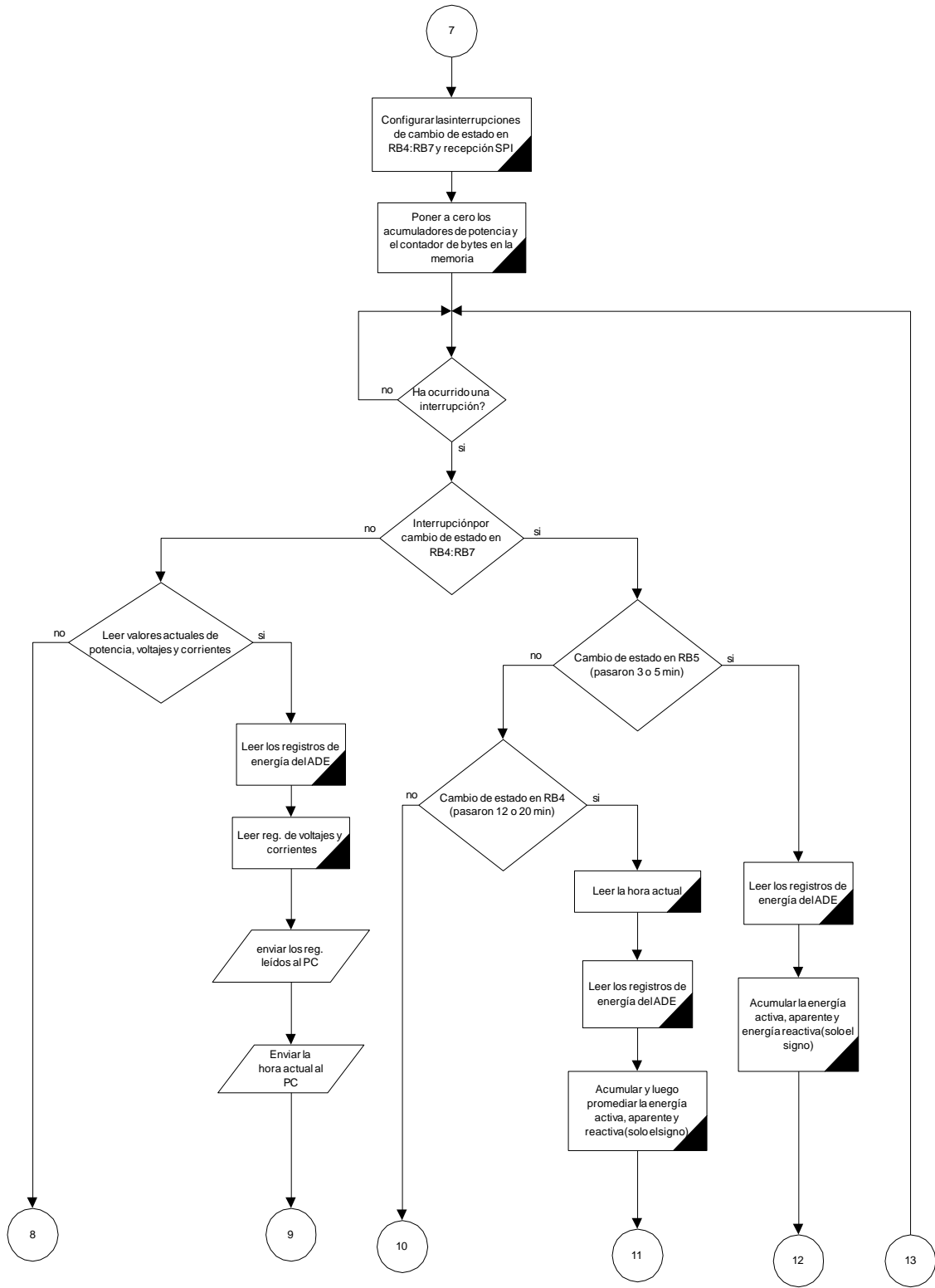
A.9 FLUJOGRAMAS DE LOS PROGRAMAS DEL MICROCONTROLADOR

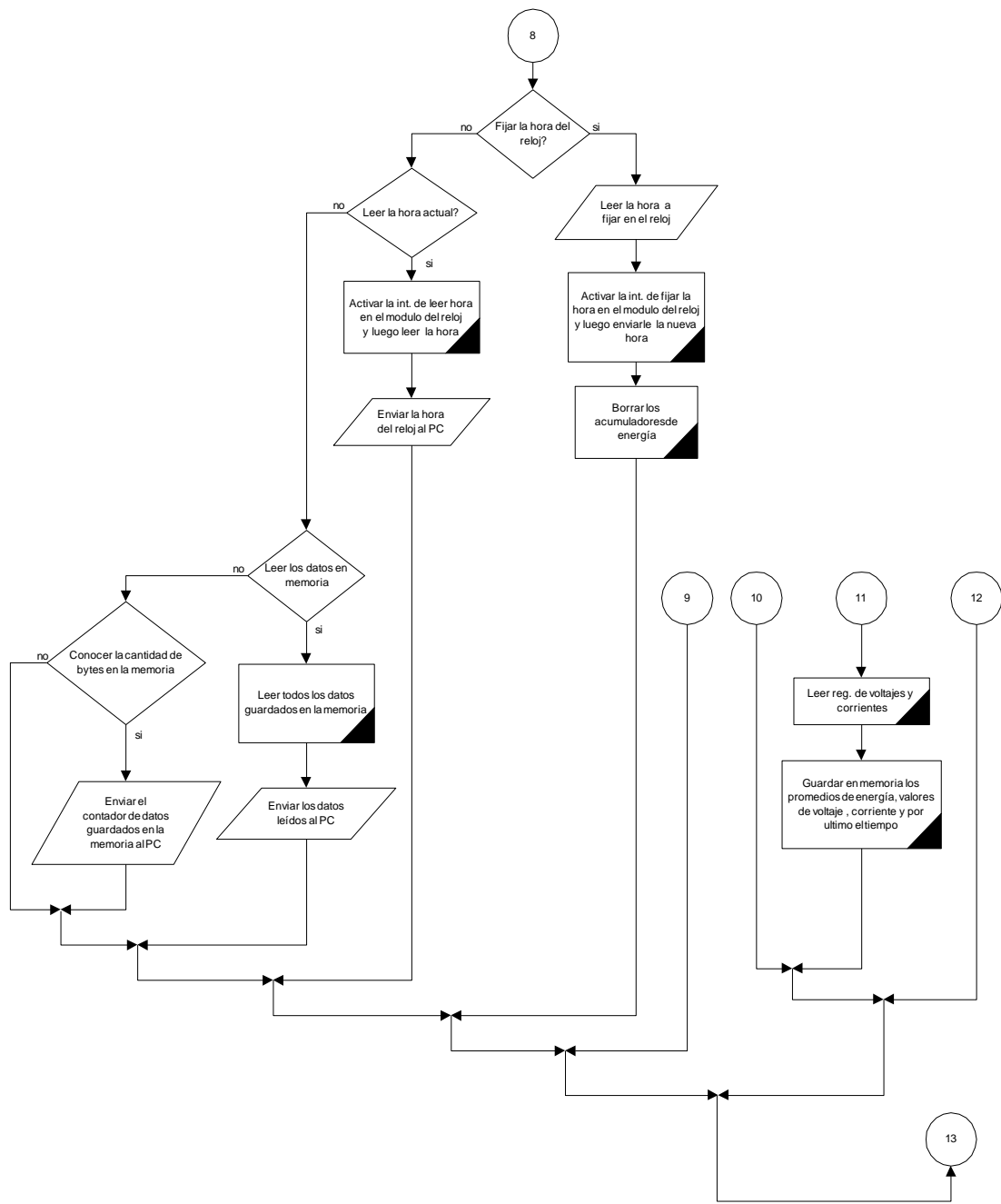
FLUJOGRAMA DEL MICROCONTROLADOR PRINCIPAL



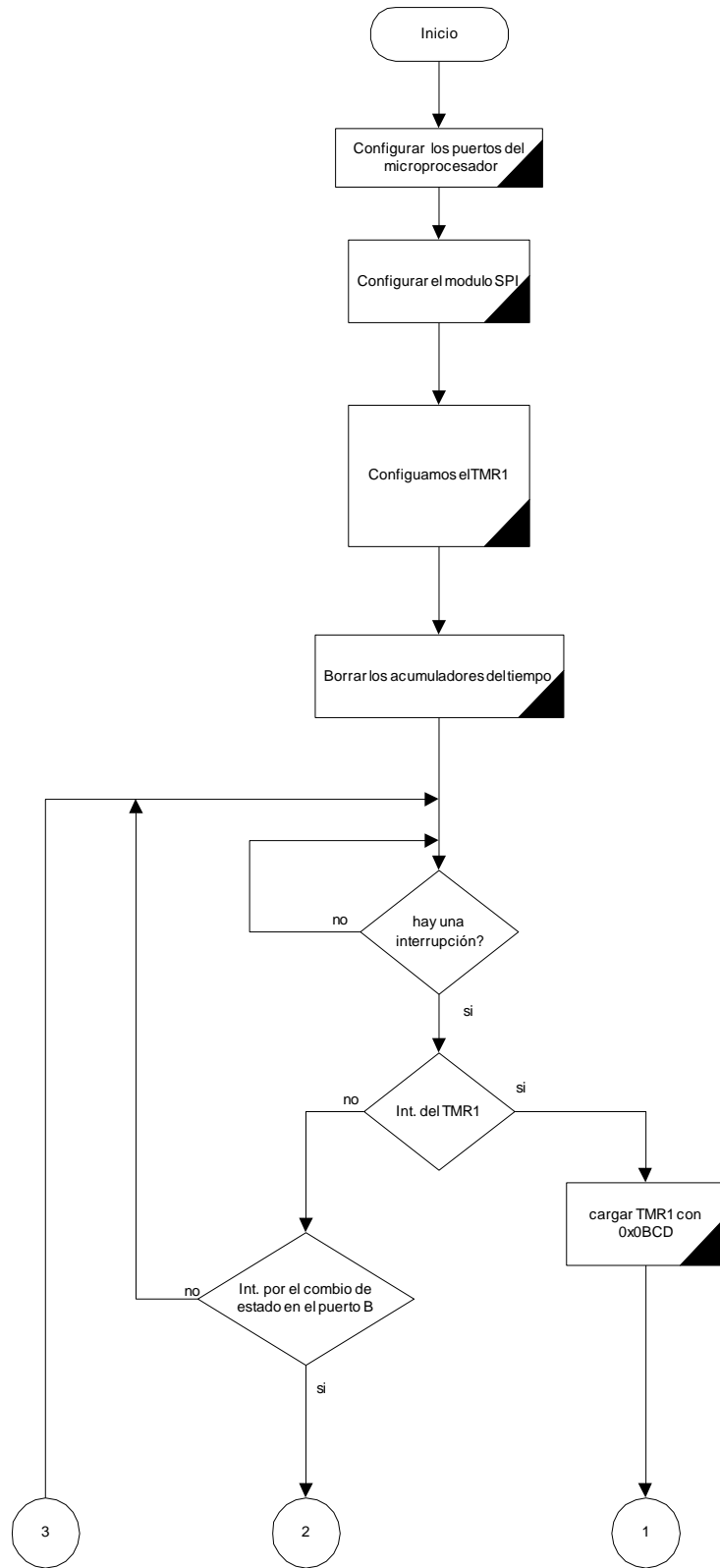


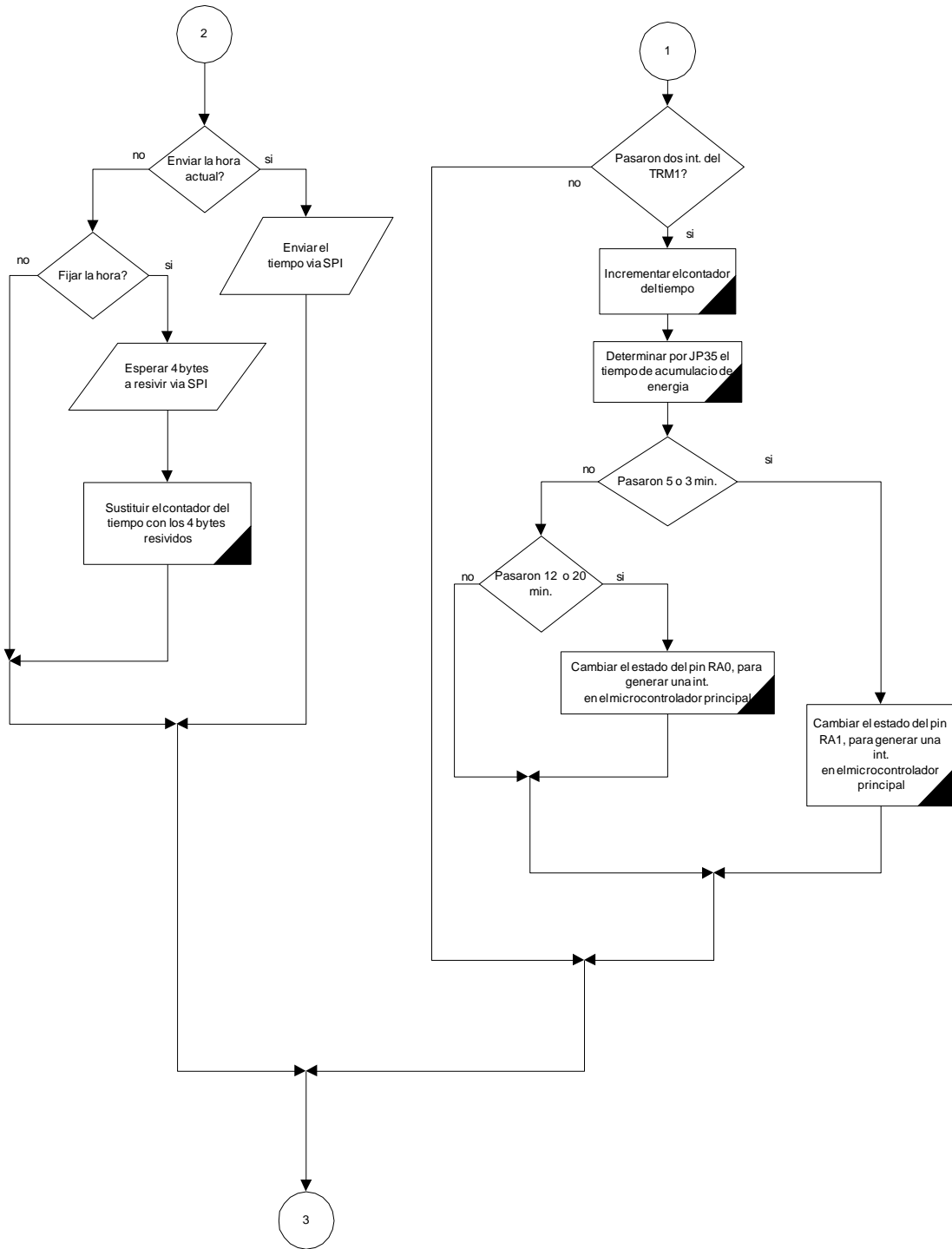




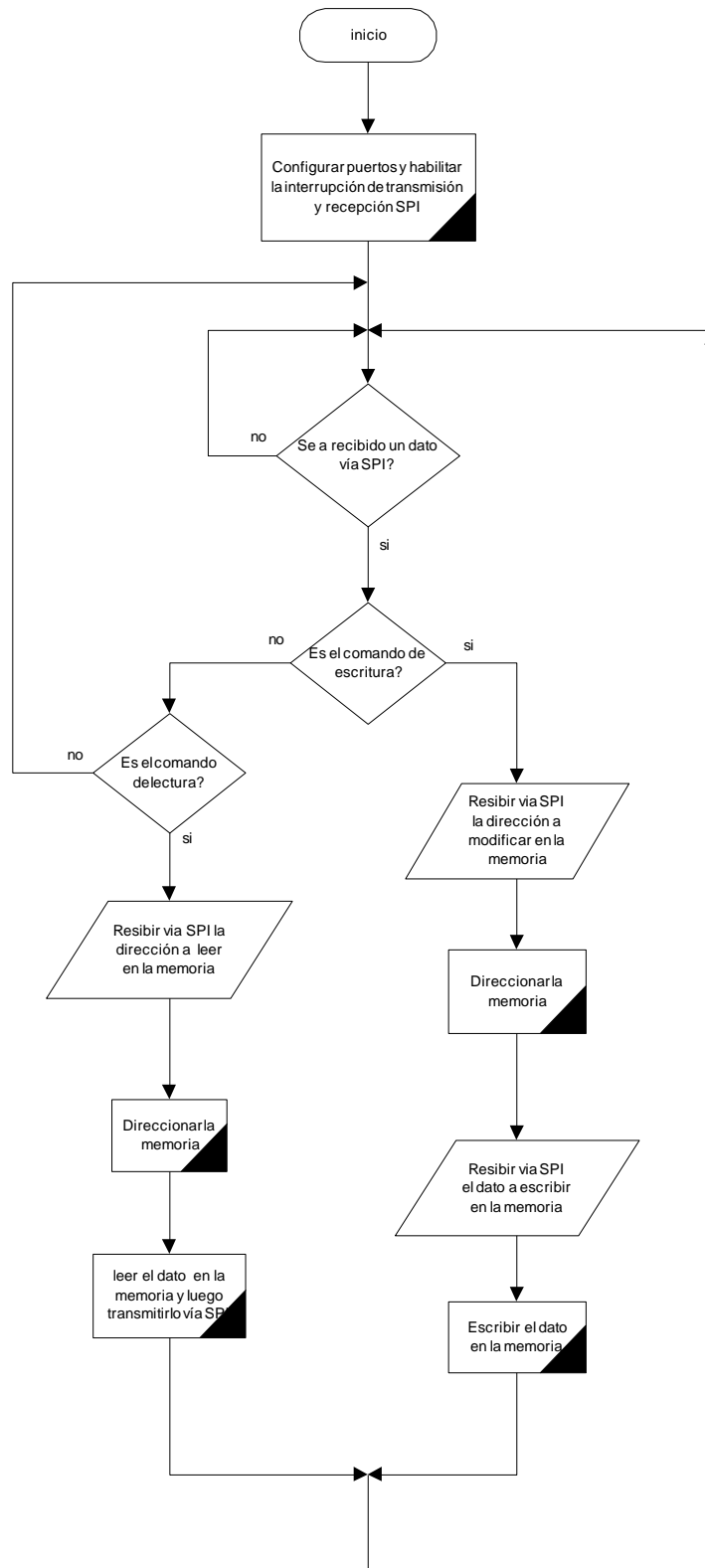


FLUJOGRAMA DEL RELOJ





FLUJOGRAMA DE LA MEMORIA



A.10 CÓDIGO DEL PROGRAMA DEL MICROCONTROLADOR

;PROGRAMA DEL MICROCONTROLADOR PRINCIPAL

;programa_principal.asm

```

;*****
;   Trabajo de Graduacion: Diseño e implementacion de un medidor trifasico
;                           multifunciones utilizando el IC ADE7754
;
;
;   Estudiantes:           - Daniel Antonio Cortez Franco
;                           - Douglas Alberto Lopez Hernandez
;                           - Fernando Alberto Arevalo Navas
;
;
;   fecha:13/01/2005
;*****
;
;   Este programa configura, controla, y lee datos del ADE5477,
;   el cual es un IC medidor de energia trifasica con interfaz SPI.
;   Los datos leidos son guardados en una memoria con interfaz SPI.
;
;
;   Este programa ademas lee y fija el tiempo de un reloj con interfaz SPI.
;   el codigo de este programa esta escrito para un PIC16F877

```

```

List      p=16F877           ;Tipo de procesador
include   "P16F877.INC"     ;Definiciones de registros internos
include   "F:\tesis\pic16f877\archivos_include\variables_trifasico.inc"
include   "F:\tesis\pic16f877\archivos_include\reg_ade7754.inc"
include   "F:\tesis\pic16f877\archivos_include\constantes_eeprom.inc"
include   "F:\tesis\pic16f877\archivos_include\macros.mac"
__config H'3f72'
__idlocs H'7777'

```

;##### Programa principal #####

```

org      0x00                ;Vector de Reset
goto     Inicio

org      0x04                ;Vector de interrupción
btfsc   INTCON,RBIF
goto     Inter_reloj ;la int. es por un cambio de estado en algun pin RB7:RB4
btfsc   PIR1,RCIF
goto     Inter_RX_dato_sci ;la int. es por recibir un dato via SPI
retfie                      ;int. desconocida, por tanto se ignora

```

Inter_RX_dato_sci

```

bcf     STATUS,RP1
bcf     STATUS,RP0          ;Selecciona banco 0
call    Rx_dato_sci         ;el dato recibido via SCI queda en W
movwf   GUARDAR_REG
movlw   0x32                ;comando para datos actuales de energia y RMS?
subwf   GUARDAR_REG,W
btfsc   STATUS,Z
goto    coman1              ;si, descargar datos actuales (energias, RMS'S y tiempo)
movlw   0x36                ;comando set clock?
subwf   GUARDAR_REG,W
btfsc   STATUS,Z
goto    coman2              ;si, cambiar el tiempo del reloj
movlw   0x37                ;comando para la hora actual?
subwf   GUARDAR_REG,W
btfsc   STATUS,Z
goto    coman3              ;si, leer el reloj
movlw   0x34
subwf   GUARDAR_REG,W       ;desea descargar datos almacenados en memoria          btfsc
STATUS,Z;(energias, RMS'S y tiempo)?
goto    coman4              ;si

```

```

movlw 0x38 ;comando para conocer la cantidad de memoria utilizada?
subwf GUARDAR_REG,W
btfsc STATUS,Z
goto coman5 ;si, en descargar el la cantidad de bytes usados.

retfie ;COMANO NO VALIDO!!

coman1 ;descargar datos actuales (energias, RMS'S y tiempo)
page_1 ;macro que selecciona la pagina 1 de la memoria de programa
call LEER_REG_ENERGY_TRIFASICA ;leemos energias trifasicas
page_0 ;macro que selecciona la pagina 0 de la memoria de programa
call LEER_REG_RMS ;lectura de corrientes y voltajes rms por fases
page_1 ;macro que selecciona la pagina 1 de la memoria de programa
call ENVIAR_REG_ENERGY_TRIFASICA_PC ;se envia la los datos leidos al PC
page_0 ;macro que selecciona la pagina 0 de la memoria de programa
call ENVIAR_REG_RMS_PC
set_spi_reloj ;macro que selecciona la spi del reloj

flash_led PORTA,0x02 ;invocamos la int. de cambia de estado en RB7:RB4 para
;el reloj, para leer tiempo actual
page_1 ;macro que selecciona la pagina 1 de la memoria de programa
call TIEMPO_DEL_RELOJ
page_0 ;macro que selecciona la pagina 0 de la memoria de programa

set_spi_ade7754 ;macro que selecciona la spi del ade7754

retfie ;FIN del int. por sci_RX

coman2 ;fijar el tiempo en el reloj
movlw 0x36
call Tx_Dato_sci ;enviamos un 0x36 a la Pc, para que nos envíe 4 bytes
call leer_un_byte ;Leemos el tiempo a fijar
movwf TIEMP4
call leer_un_byte
movwf TIEMP3
call leer_un_byte
movwf TIEMP2
call leer_un_byte
movwf TIEMP1

set_spi_reloj ;macro que selecciona la spi del reloj
flash_led PORTB,0x06 ;invocamos la int. de cambia de estado en RB7:RB4 para
;el reloj, para fijar el tiempo

;retardo para la comunicacion spi
retardo_x .2 ;retardo=2+(1+2)*2+1=9 ciclos de reloj

page_1
movf TIEMP4,W ;enviamos el tiempo a poner en el reloj, el byte MSB es
call Tx_dato_spi ;el primero y el LSB el ultimo en enviarse
movf TIEMP3,W
call Tx_dato_spi
movf TIEMP2,W
call Tx_dato_spi
movf TIEMP1,W
call Tx_dato_spi

call borrar_acumuladores_energia ;borramos los acumuladores de energia
page_0

set_spi_ade7754 ;macro que selecciona la spi del ade7754

retfie ;FIN del int. por sci_RX

coman3 ;peticion de la hora actual

```

```

set_spi_reloj          ;macro que selecciona la spi del reloj

flash_led PORTA,0x02   ;invocamos la int. de cambia de estado en RB7:RB4 para
                       ;el reloj, para leer tiempo actual

page_1
call   TIEMPO_DEL_RELOJ      ; leemos 4 bytes y los enviamos al PC
page_0

set_spi_ade7754       ;macro que selecciona la spi del ade7754

retfie   ;FIN del int. por sci_RX

coman4
        ;descargar datos almacenados en memoria (energias, RMS'S y tiempo)
set_spi_memoria       ;macro que selecciona la spi de la memoria RAM estatica

page_1
call   leer_todos_los_datos_guardados ;leemos todos los byte almacenados
page_0
        ; asta este momento en la memoria

set_spi_ade7754       ;macro que selecciona la spi del ade7754

retfie   ;FIN del int. por sci_RX

coman5
        ;cantidad de bytes almacenados en memoria

movf    CONTADOR_MSB,w ;enviamos al PC, la cantidad de bytes almacenados asta
call    Tx_Dato_sci     ;este momento empezando por el MSB y terminando con el
                       ;LSB

movf    CONTADOR_MEDIO,w
call    Tx_Dato_sci

movf    CONTADOR_LSB,w
call    Tx_Dato_sci

retfie   ;FIN del int. por sci_RX

Inter_reloj
        ;estas interrupciones son producidas por el reloj
movf    PORTB,W
bcf     INTCON,RBIF      ;ponemos la bandera RBIF nuevamente a cero
xorwff INT_DEL_RELOJ,F  ;vemos que pin del puerto B fue el que cambiado
btfsc  INT_DEL_RELOJ,0x05 ;de estado
goto   periodo_5min     ;ya pasaron 3 o 5 min. leer la energias y
                       ;acumularlas

btfsc  INT_DEL_RELOJ,0x04
goto   periodo_20min    ;ya pasaron 12 o 20 min. leer la energias,
                       ;acumularlas, promediarlas
                       ;y guardarlas en la memoria

retfie

periodo_5min
movwff INT_DEL_RELOJ      ;guardamos el estado actual del puerto B
page_1
call   LEER_REG_ENERGY_TRIFASICA ; lectura de las potencia potencias
call   SUMAR_ENERGIA_ACTIVA      ;se acumulan los tres tipos de energia
call   SUMAR_ENERGIA_APARENTE
call   SUMAR_ENERGIA_REACTIVA
page_0
retfie   ;FIN del int. por cambio de estado de RB4:RB7

periodo_20min
movwff INT_DEL_RELOJ      ;guardamos el estado actual del puerto B
;leemos el tiempo actual
set_spi_reloj           ;macro que selecciona la spi del reloj

```

```

flash_led PORTA,0x02      ;invocamos la int. de cambia de estado en RB7:RB4 para
                          ;el reloj, y poder leer tiempo actual

page_1                    ;macro que selecciona la pagina 1 de la memoria de programa
call TIEMPO_DEL_RELOJ
page_0                    ;macro que selecciona la pagina 0 de la memoria de programa

set_spi_ade7754          ;macro que selecciona la spi del ade7754

page_1
call LEER_REG_ENERGY_TRIFASICA      ; lectura de los 3 tipos de energia
call SUMAR_ENERGIA_ACTIVIA          ;se acumulan los tres tipos de energia
call SUMAR_ENERGIA_APARENTE
call SUMAR_ENERGIA_REACTIVA

call PROMEDIO_ENERGIA_ACTIVIA      ;promediamos la 4 lecturas hechas en los
call PROMEDIO_ENERGIA_APARENTE    ;12 o 20 minutos pasados.
call PROMEDIO_ENERGIA_REACTIVA
page_0

call LEER_REG_RMS              ;leemos los voltajes y corrientes RMS

set_spi_memoria          ;macro que selecciona la spi de la memoria RAM estatica

page_1
call GUARDAR_DATOS_A_MEMORIA      ;guardanos el promedio de las energias,
                                  ;los voltajes, corrientes y el tiempo.
call borrar_acumuladores_energia  ;borramos los acumuladores de energia
page_0

set_spi_ade7754          ;macro que selecciona la spi del ade7754

```

```

retfie                    ;fin de la Inter_reloj

```

Inicio

```

;-----
bcf STATUS,RP1
bcf STATUS,RP0          ;Selecciona banco 0

clrf PORTA
clrf PORTB
clrf PORTC
clrf PORTD

bsf STATUS,RP0          ;Selecciona banco 1

;Configura el puerto A
movlw 0x06              ;Configuramos todos los pines como I/O (cap. 11)
movwf ADCON1
movlw b'00011000'
movwf TRISA
;Configura el puerto B
movlw b'00110111'      ;configuramos el puerto B PORTB<0>=1 (para PB0/INT) y
                        ;PORTB<4:5>=1 (para int. por cambio de estado)
movwf TRISB

;Configura el puerto C
movlw b'10010000'
movwf TRISC            ;configuramos sck(TRISC<3>)=0, SDI(TRISC<4>)=1
                        ;SDO(TRISC<5>)=0, TX(TRISC<6>)=0, RX(TRISC<7>)=1
                        ;y los demas pines como salidas.

movlw b'10000111'
movwf OPTION_REG      ;Preescaler de 256 asociado al TMR0
                        ;PB0/INT (bit 6) se activa en flanco de caida

```



```

##### configuracion del modulo SPI MODO MAESTRO
#####

        bcf     STATUS,RP0      ;banco 0
        movlw   b'00000000'     ;configuramos WCOL=0, SSPOV=0, SSPEN=0, CKP=0,
                                ; SSPM3:SSPM0=0000 (sck=Fosc/4). SPI desactivada

        movwf   SSPCON

        bcf     STATUS,RP0      ;banco 1
        movlw   b'10000000'     ;configuramos SMP(SSPSTAT<7>)=1, CKE(SSPSTAT<6>)=0,
                                ;BF(SSPSTAT<0>)=0

        movwf   SSPSTAT

        bcf     STATUS,RP0      ;banco 0
        bsf     SSPCON,SSPEN     ;se activa la SPI

##### configuracion del modulo SCI #####

        bsf     STATUS,RP0      ;banco 1

        ;configuracion de USART en TX
        movlw   b'00100100'
        movwf   TXSTA           ;TX en On, modo asíncrono con 8 bits y alta velocidad

        ;configuracion de USART relacion de BAUDIOS
        movlw   .25             ;Cristal de 4 MHz -> SPBRG=25 para 9600 baudios
        movwf   SPBRG

        ;configuracion de USART en RX
        bcf     STATUS,RP0      ;Selecciona banco 0
        movlw   b'10010000'     ;configuramos el RX y USART ponemos en On
        movwf   RCSTA

        bcf     STATUS,RP0      ;banco 0

##### CONFIGURACION DEL ADE7754 #####

        call    DELAY2S         ;retardo para que establezca el ADE7754 (power up)

        bsf     PORTC,0x02      ;se habilita el ADE7754, sacamos al ADE del reset

        set_spi_ade7754        ;macro que selecciona la spi del ade7754

        call    DELAY1S         ;para que establezca el ADE7754

        ;conf. el registro OPMODE
        movlw   0x00             ;solo se activa el CF
        movwf   OUTD1
        movlw   OPMODE_W
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03 ;definimos la cantidad de bytes a escribir como 1
                                ;byte
        call    SPIDX            ;enviamos el comando de escritura (OPMODE_W)
                                ;el valor a escribir (OUTD1)

        ;conf. registro MMODE
        movlw   b'00010000'     ;se fija la fase A para medir el periodo y cruce por
        movwf   OUTD1           ;cero, picos de voltaje y corriente, etc.
        movlw   MMODE_W         ;comando de escritura para el ADE
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03
        call    SPIDX

        ;conf. registro WAVMODE
        movlw   b'00000000'     ;con datos por default
        movwf   OUTD1
        movlw   WAVMODE_W
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03

```

```

call    SPIDX

;conf. registro IRQEN
movlw  0x04          ;solo habilitamos la int. de LENERGY
movwf  OUTD2
movlw  0x00
movwf  OUTD1
movlw  IRQEN_W      ;comando de escritura al registro IRQEN
movwf  SPI_COMANDO
bcf    LENG_DATOS,0x03 ;definimos la cantidad de bytes a escribir como 2
                        ;byte
call    SPIDX          ;enviamos el comando de escritura (IRQEN_W)
                        ;el valor a escribir (OUTD2-OUTD1)

;----- configuracion del tipo de servicio en el ADE7754 -----
loop_boton1
    btfss PORTB,0x01    ;enter presionado (0V en PORTB<1>)
    goto tipo_de_servicio ;si
    goto loop_boton1    ;no, el usuario tiene que presionar el pulsador de
                        ;calibracion del medidor

tipo_de_servicio
    clrf TIPO_SERVICIO
    btfsc PORTA,0x03
    goto loop_boton3    ;es 8s/15s o 5s/13s?
    btfsc PORTA,0x04
    goto conf_6s        ;es 6s/14s
    goto conf_9s        ;es 9s/16s

loop_boton3
    btfsc PORTA,0x04
    goto conf_5s        ;es 5s/13s
    goto conf_8s        ;es 8s/15s

conf_5s
    clrf TIPO_SERVICIO
    bsf  TIPO_SERVICIO,0x00    ;si TIPO_SERVICIO=0x00 significa que es conf_5s
    goto tipo_de_servicio_fin

conf_6s
    clrf TIPO_SERVICIO
    bsf  TIPO_SERVICIO,0x01    ;si TIPO_SERVICIO=0x01 significa que es conf_6s
    goto tipo_de_servicio_fin

conf_8s
    clrf TIPO_SERVICIO
    bsf  TIPO_SERVICIO,0x02    ;si TIPO_SERVICIO=0x02 significa que es conf_8s
    goto tipo_de_servicio_fin

conf_9s
    clrf TIPO_SERVICIO
    bsf  TIPO_SERVICIO,0x03    ;si TIPO_SERVICIO=0x03 significa que es conf_9s
    goto tipo_de_servicio_fin

tipo_de_servicio_fin

;----- medo de conf. de los registros del ADE7754-----
loop_boton4
    btfsc PORTB,0x02    ;desea calibrar el ADE de forma manual
    goto loop_calibracion ;si
    goto calibracion_default ;no, usar valores almacenados en la eeprom del
                        ;pic16f877

loop_calibracion
    clrf GUARDAR_REG    ;bandera que indica si se desea guardar una nueva conf.
                        ;del ADE en la eeprom del pic16f877
    page_1
    call LEER_REG_CONF_EEPROM ;lee la conf. default de la EEPROM

```

```

page_0
call    ENVIAR_REG_CONF_AL_ADE ;conf. el ADE con valores de default

call    LEER_REG_CONF_ADE      ;leer reg. de conf., potencias y RMS del ADE

page_1
call    LEER_REG_ENERGY_TRIFASICA ;leemos energias trifasica
page_0

call    espera_dato_byte ;esperamos un byte (0x31) de la PC para enviar
                        ;los valores leidos del ADE (este byte es de
                        ;sincronismo) el dato leido queda en W

sublw   0x31              ;0x31-W
btfss   STATUS,Z;es (0x31-W)==0?
goto    $-3              ;no, entonces espere
                        ;si, continuar
call    ENVIAR_REG_CONF_PC    ;enviar reg., potencias y RMS a la PC

page_1
call    ENVIAR_REG_ENERGY_TRIFASICA_PC ;se envia la los datos leidos al PC
page_0
call    ENVIAR_REG_RMS_PC

call    LEER_REG_CONF_PC    ;leer reg. de configuracion enviados desde la
                        ;PC y por ultimo el registro GUARDAR_REG

btfsc   GUARDAR_REG,0x03 ;guardamos los reg. de configuracion en la eeprom?
goto    guardar_datos      ;si
goto    no_guardar_datos    ;no,

guardar_datos

page_1
call    ESCRIVIR_DATOS_EEPROM ;guardar los registros de conf. en EEPROM
page_0

no_guardar_datos

call    ENVIAR_REG_CONF_AL_ADE ;enviar reg. de configuracion enviados
                        ;desde la PC al ADE
call    LEER_REG_CONF_ADE      ;leer reg. de conf., potencias y RMS del ADE

page_1
call    LEER_REG_ENERGY_TRIFASICA ;leemos energias trifasica
page_0

call    espera_dato_byte ;esperamos un byte (0x31) de la PC para enviar
                        ;los valores leidos del ADE (este byte es de
                        ;sincronismo) el dato leido queda en W

sublw   0x31              ;0x31-W
btfss   STATUS,Z        ;es (0x31-W)==0?
goto    $-3              ;no, entonces espere

call    ENVIAR_REG_CONF_PC    ;si, enviar reg., potencias y RMS a la PC

page_1
call    ENVIAR_REG_ENERGY_TRIFASICA_PC ;se envia la los datos leidos al PC
page_0
call    ENVIAR_REG_RMS_PC

call    espera_dato_byte ;esperamos un byte (0x37) de la PC, para saber
                        ;si se a terminado el proceso de calibracion el
                        ;dato leido queda en W

sublw   0x37              ;0x37-W
btfss   STATUS,Z        ;(0x37-W)==0?

```

```

                goto    cf_trifasico          ;si, terminar la calibracion
                goto    loop_calibracion     ;no, continuar

cf_trifasico
                page_1
                call    set_trifasico
                page_0

loop_cf_trifasico
                call    espera_dato_byte    ;W=byte de la PC
                sublw  0x35                 ;0x35-W
                btfss  STATUS,Z            ;(0x35-W)!=0?
                goto    loop_cf_trifasico   ;si, continuar
                goto    calibracion_default ;no, terminar

calibracion_default
                page_1
                call    LEER_REG_CONF_EEPROM ;lee registros de conf. guardados en la EEPROM
                page_0

                call    ENVIAR_REG_CONF_AL_ADE ;enviar los registros de conf. al ADE

calibracion_salir ;fin de la calibracion de ADE

;----- ACTIVANDO LAS INT. EN EL PIC -----

                bsf     STATUS,RP0          ;Selecciona banco 1

                bsf     PIE1,RCIE          ;activamos la USART receive interrupt
                bsf     INTCON,RBIE        ;Activamos las int. de cambio de estado en los
                ;pines RB4-RB7

                page_1
                call    borrar_acumuladores_energia ;barramos los acumuladores de energia
                call    clear_contadores_datos_memoria ;inicializamos la memoria a cero bytes escritos
                page_0

                bcf     STATUS,RP1
                bcf     STATUS,RP0        ;Selecciona banco 0

                bsf     INTCON,PEIE        ;activa todas las interrupciones de los perifericos
                bsf     INTCON,GIE        ;activa todas las interrupciones globales

;##### FIN ACTIVANDO LAS INT. EN EL PIC #####

loop
                flash_led PORTB,0x07
                goto    loop

;##### DEFINICION DE RUTINAS #####

;-----
;
; Rutinas de retardos de 10ms, 1s y 2s.
; argumentos: - no hay
; nota: TMR0 debe tener un divisor de frec. de 256
;-----

DELAY10        bcf     INTCON,T0IF      ; Borra el flag de estado del TMR0
                movlw  0xD9
                movwf  TMR0             ;Carga del TMR0 para que cuente 39 que en complemento a 2 es
                ;D9
DELAY10_1      btfss  INTCON,T0IF      ; Espera el desbordamiento del TMR0
                goto    DELAY10_1
                decfsz TEMPO1,F

```

```

                goto    DELAY10
                return

DELAY1S movlw   .100           ;se carga con 200 decimal
                movwf   TEMPO1   ; Carga la variable TEMPO1, que sera decrementada
                call    DELAY10   ; en la rutina a la que se llama
                return

DELAY2S movlw   0xC8           ;se carga con 200 decimal
                movwf   TEMPO1   ; Carga la variable TEMPO1, que sera decrementada
                call    DELAY10   ; en la rutina a la que se llama
                return

```

```

;-----
;
; Rutina para Transmitir un byte via sci.
; argumentos: - el dato a transmitir debe estar en W
;
;-----

```

```

Tx_Dato_sci
                movwf   TXREG     ;Almacena el byte a transmitir
                bsf     STATUS,RP0

Tx_Dato_sci_Wait
                btfss   TXSTA,TRMT ;Byte transmitido ??
                goto    Tx_Dato_sci_Wait ;No, esperar

                bcf     STATUS,RP0
                return

```

```

;-----
;
; Rutina para Recibir un byte via sci.
; argumentos: - el dato recibido quedara en W
;
;-----

```

```

Rx_dato_sci
                movf    RCREG,W
                return

```

```

;-----
;
; Rutina que espera un byte via sci, asta que lo recibe continua.
; argumentos: - el dato recibido quedara en W
;
;-----

```

```

espera_dato_byte
                btfss   PIR1,RCIF
                goto    $-1
                call    Rx_dato_sci
                return

```

```

;-----
;
;lee del ADE un numero dato de de bytes (definidos en LENG_DATOS ), el comando de lectura
; esta SPI_COMANDO
;argumentos -la cantidad de bytes a leer se de fine en LENG_DATOS
; -el registro a leer se coloca en SPI_COMANDO
;argumentos de salida: INSD1 (lsb), INSD2, INSD3, INSD4(msb)
;
;nota: 4 bytes maximos de lectura
;-----

```

```

SPIRX:
                bcf     STATUS,RP1
                bsf     STATUS,RP0 ;Selecciona banco 1

                bcf     SSPSTAT,0 ;limpiamos la bandera del buffer spi

```

```

        bcf     STATUS,RP0      ;banco 0
        movf   SPI_COMANDO,w   ;enviamos la direccion a leer del ADE7754
        movwf  SSPBUF

LOOPSPI:
        bsf     STATUS,RP0      ;banco 1
        btfss  SSPSTAT,0       ;dato enviado?
        goto   LOOPSPI         ;no, seguir esperando

        bcf     STATUS,RP0      ;banco 0

        clrf   INSD1           ;limpiamos los reg. de entrada
        clrf   INSD2
        clrf   INSD3
        clrf   INSD4

        btfsc  LENG_DATOS,3    ;leer 4 bytes (32 bits)?
        GOTO   bit32
        btfsc  LENG_DATOS,2    ;leer 3 bytes (24 bits)?
        GOTO   bit24
        btfsc  LENG_DATOS,1    ;leer 2 bytes (16 bits)?
        GOTO   bit16
        btfsc  LENG_DATOS,0    ;leer 1 byte (8 bits)?
        GOTO   bit8

bit32:
        bsf     STATUS,RP0      ;banco 1
        bcf     SSPSTAT,0       ;limpiamos la bandera del buffer spi
        bcf     STATUS,RP0      ;banco 0
        movf   0x05,w          ;ponemos un dato cualquiera en el buffer
        movwf  SSPBUF          ;esto se hace para generar una señal de reloj
        bsf     STATUS,RP0      ; banco 1

LOOP32:
        btfss  SSPSTAT,0       ; dato recibido?
        GOTO   LOOP32         ;no, seguir esperando
        bcf   STATUS,RP0       ;si, banco 0
        movf  SSPBUF,w         ;cargamos el dato leído en un reg. de temporal
        movwf INSD4

bit24:
        bsf     STATUS,RP0      ;banco 1
        bcf     SSPSTAT,0       ;limpiamos la bandera del buffer spi
        bcf     STATUS,RP0      ;banco 0
        movf   0x05,w          ;ponemos un dato cualquiera en el buffer
        movwf  SSPBUF
        bsf     STATUS,RP0      ;banco 1

LOOP24:
        btfss  SSPSTAT,0       ;dato recibido?
        GOTO   LOOP24         ;no, seguir esperando
        bcf   STATUS,RP0       ;si, banco 0
        movf  SSPBUF,w         ;cargamos el dato leído en un reg. de temporal
        movwf INSD3

bit16:
        bsf     STATUS,RP0      ;banco 1
        bcf     SSPSTAT,0       ;limpiamos la bandera del buffer spi
        bcf     STATUS,RP0      ;banco 0
        movf   0x05,w          ;ponemos un dato cualquiera en el buffer
        movwf  SSPBUF
        bsf     STATUS,RP0      ;banco 1

LOOP16:
        btfss  SSPSTAT,0       ;dato recibido?
        GOTO   LOOP16         ;no, seguir esperando
        bcf   STATUS,RP0       ;si, banco 0

```

```

movf    SSPBUF,w      ;cargamos el dato leído en un reg. temporal
movwf   INSD2

bit8:
bsf     STATUS,RP0    ;banco 1
bcf     SSPSTAT,0     ;limpiamos la bandera del buffer spi
bcf     STATUS,RP0    ;banco 0
movf    0x05,w        ;ponemos un dato cualquiera en el buffer
movwf   SSPBUF
bsf     STATUS,RP0    ;banco 1

LOOP8:
btfss  SSPSTAT,0     ;dato recibido?
GOTO   LOOP8         ;no, seguir esperando
bcf     STATUS,RP0    ; si, banco 0
movf    SSPBUF,w     ;cargamos el dato leído en un reg. temporal
movwf   INSD1
return

;-----
;
;
; Rutina para escribir en el ADE7754
; argumentos: - numero de bits a escribir en LENG_DATOS
;
;               - registro en donde se escribira SPI_COMANDO
;
;               - OUTD2 byte superior
;
;               - OUTD1 byte inferior
;
;
;nota: escribe asta en registros de 2 bytes maximos de longitud
;-----
SPIDX:
bcf     STATUS,RP1
bsf     STATUS,RP0    ;Selecciona banco 1
bcf     SSPSTAT,0     ;limpiamos la bandera del buffer spi
bcf     STATUS,RP0    ;banco 0
movf    SPI_COMANDO,W ;cargamos la direccion a escribir
movwf   SSPBUF

bsf    STATUS,RP0 ;banco 1
TXLOOP1:
btfss  SSPSTAT,0     ;dato enviado?
goto   TXLOOP1      ;no, esperar

bsf    STATUS,RP0 ;banco 1
bcf    SSPSTAT,0   ;limpiamos la bandera del buffer spi
bcf    STATUS,RP0 ;banco 0
btfsc  LENG_DATOS,0x03 ;cuantos bit se escribiran? 8 o 16
goto   eight       ;son 8 bit
movf   OUTD2,w     ;son 16, cargamos el bit mas significativo
movwf  SSPBUF

bsf    STATUS,RP0 ;banco 1
TXLOOP2:
btfss  SSPSTAT,0     ;dato enviado?
goto   TXLOOP2      ;no, esperar

eight:
bsf    STATUS,RP0 ;banco 1
bcf    SSPSTAT,0   ;limpiamos la bandera del buffer spi
bcf    STATUS,RP0 ;banco 0
movf   OUTD1,w     ;cargamos los 8 bits LSB
movwf  SSPBUF

bsf    STATUS,RP0 ;banco 1
TXLOOP3:
btfss  SSPSTAT,0     ;dato enviado?
GOTO   TXLOOP3      ;no, esperar

```

```

bcf     STATUS,RP0      ;banco 0
return

```

```

-----
;
;
; Rutina para leer todos los registros de conf. del ADE7754
; argumentos: no hay
;
;
-----

```

LEER_REG_CONF_ADE

```

bcf     STATUS,RP1      ;Selecciona banco 0 y 1
bcf     STATUS,RP0      ;Selecciona banco 0

movlw   LINCYC_R        ;cargamos el comando para leer el reg.LINCYC
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf     LENG_DATOS,0x01 ;el reg. LINCYC es de 2 bytes
call   SPIRX
movf   INSD2,W
movwf   LINCYC1         ;guardamos el byte msb
movf   INSD1,W
movwf   LINCYC0         ;guardamos el byte lsb

movlw   AWG_R           ;cargamos el comando para leer el reg.AWG
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf   AWG1           ;guardamos el byte msb
movf   INSD1,W
movwf   AWG0           ;guardamos el byte lsb

movlw   BWG_R           ;cargamos el comando para leer el reg.BWG
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf   BWG1           ;guardamos el byte msb
movf   INSD1,W
movwf   BWG0           ;guardamos el byte lsb

movlw   CWG_R           ;cargamos el comando para leer el reg.CWG
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf   CWG1           ;guardamos el byte msb
movf   INSD1,W
movwf   CWG0           ;guardamos el byte lsb

movlw   APHCAL_R        ;cargamos el comando para leer el reg.APHCAL
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf     LENG_DATOS,0x00 ;leer 1 bytes
call   SPIRX
movf   INSD1,W
movwf   APHCAL0        ;guardamos el byte

movlw   BPHCAL_R        ;cargamos el comando para leer el reg.BPHCAL
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf     LENG_DATOS,0x00 ;leer 1 bytes
call   SPIRX

```



```

movf    INSD1,W
movwf   BPHCAL0           ;guardamos el byte

movlw   CPHCAL_R         ;cargamos el comando para leer el reg.CPHCAL
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x00 ;leer 1 bytes
call    SPIRX
movf    INSD1,W
movwf   CPHCAL0           ;guardamos el byte

movlw   AAPOS_R          ;cargamos el comando para leer el reg.AAPOS
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call    SPIRX
movf    INSD2,W
movwf   AAPOS1           ;guardamos el byte msb
movf    INSD1,W
movwf   AAPOS0           ;guardamos el byte lsb

movlw   BAPOS_R          ;cargamos el comando para leer el reg.BAPOS
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call    SPIRX
movf    INSD2,W
movwf   BAPOS1           ;guardamos el byte msb
movf    INSD1,W
movwf   BAPOS0           ;guardamos el byte lsb

movlw   CAPOS_R          ;cargamos el comando para leer el reg.CAPOS
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call    SPIRX
movf    INSD2,W
movwf   CAPOS1           ;guardamos el byte msb
movf    INSD1,W
movwf   CAPOS0           ;guardamos el byte lsb

movlw   CFNUM_R          ;cargamos el comando para leer el reg.CFNUM
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call    SPIRX
movf    INSD2,W
movwf   CFNUM1           ;guardamos el byte msb
movf    INSD1,W
movwf   CFNUM0           ;guardamos el byte lsb

movlw   CFDEN_R          ;cargamos el comando para leer el reg.CFDEN
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call    SPIRX
movf    INSD2,W
movwf   CFDEN1           ;guardamos el byte msb
movf    INSD1,W
movwf   CFDEN0           ;guardamos el byte lsb

movlw   AIRMSOS_R        ;cargamos el comando para leer el reg.AIRMSOS
movwf   SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01 ;leer 2 bytes
call    SPIRX
movf    INSD2,W

```

```

movwf AIRMSOS_1      ;guardamos el byte msb
movf INSD1,W
movwf AIRMSOS_0      ;guardamos el byte lsb

movlw BIRMSOS_R      ;cargamos el comando para leer el reg.BIRMSOS
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX
movf INSD2,W
movwf BIRMSOS_1      ;guardamos el byte msb
movf INSD1,W
movwf BIRMSOS_0      ;guardamos el byte lsb

movlw CIRMSOS_R      ;cargamos el comando para leer el reg.CIRMSOS
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX
movf INSD2,W
movwf CIRMSOS_1      ;guardamos el byte msb
movf INSD1,W
movwf CIRMSOS_0      ;guardamos el byte lsb

movlw AVRMSOS_R      ;cargamos el comando para leer el reg.AVRMSOS
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX
movf INSD2,W
movwf AVRMSOS_1      ;guardamos el byte msb
movf INSD1,W
movwf AVRMSOS_0      ;guardamos el byte lsb

movlw BVRMSOS_R      ;cargamos el comando para leer el reg.BVRMSOS
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX
movf INSD2,W
movwf BVRMSOS_1      ;guardamos el byte msb
movf INSD1,W
movwf BVRMSOS_0      ;guardamos el byte lsb

movlw CVRMSOS_R      ;cargamos el comando para leer el reg.CVRMSOS
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX
movf INSD2,W
movwf CVRMSOS_1      ;guardamos el byte msb
movf INSD1,W
movwf CVRMSOS_0      ;guardamos el byte lsb

movlw AAPGAIN_R      ;cargamos el comando para leer el reg.AAPGAIN
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX
movf INSD2,W
movwf AAPGAIN1      ;guardamos el byte msb
movf INSD1,W
movwf AAPGAIN0      ;guardamos el byte lsb

movlw BAPGAIN_R      ;cargamos el comando para leer el reg.BAPGAIN
movwf SPI_COMANDO
clrf LENG_DATOS

```

```

bsf     LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf  BAPGAIN1          ;guardamos el byte msb
movf   INSD1,W
movwf  BAPGAIN0          ;guardamos el byte lsb

movlw  CAPGAIN_R        ;cargamos el comando para leer el reg.CAPGAIN
movwf  SPI_COMANDO
clrf   LENG_DATOS
bsf    LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf  CAPGAIN1          ;guardamos el byte msb
movf   INSD1,W
movwf  CAPGAIN0          ;guardamos el byte lsb

movlw  AVGAIN_R         ;cargamos el comando para leer el reg.AVGAIN
movwf  SPI_COMANDO
clrf   LENG_DATOS
bsf    LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf  AVGAIN1          ;guardamos el byte msb
movf   INSD1,W
movwf  AVGAIN0          ;guardamos el byte lsb

movlw  BVGAIN_R         ;cargamos el comando para leer el reg.BVGAIN
movwf  SPI_COMANDO
clrf   LENG_DATOS
bsf    LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf  BVGAIN1          ;guardamos el byte msb
movf   INSD1,W
movwf  BVGAIN0          ;guardamos el byte lsb

movlw  CVGAIN_R        ;cargamos el comando para leer el reg.CVGAIN
movwf  SPI_COMANDO
clrf   LENG_DATOS
bsf    LENG_DATOS,0x01 ;leer 2 bytes
call   SPIRX
movf   INSD2,W
movwf  CVGAIN1          ;guardamos el byte msb
movf   INSD1,W
movwf  CVGAIN0          ;guardamos el byte lsb

call   LEER_REG_ENERGY ;lectura de los 3 tipos de energia fase
call   LEER_REG_RMS    ;lectura los voltajes y corrientes rms
return

```

```

;-----
;
; Rutina para leer todos los registros de energia (S, P y Q) por fase
; y RMS del ADE7754 (se usa en la calibracion del ADE)
; argumentos: no hay
;
;-----

```

```

LEER_REG_ENERGY
bcf    STATUS,RP1      ;Selecciona banco 0 y 1
bcf    STATUS,RP0      ;Selecciona banco 0

```

```

espera_1_fase_a_clear      ;esperamos a que la IRQ del ADE sea active (cambie de 1 a 0 logico)
btfsc  INTCON,INTF        ;y por tanto un cambio de estado en RB0
goto   salir_1_fase_a_clear ;ya estuvo
goto   espera_1_fase_a_clear ;seguir esperando

```

```

salir_1_fase_a_clear
    bcf     INTCON,INTF      ;ponemos a cero la bandera de peticion de int. externa
    movlw  0x11             ;lectura con reset del registro de STATUS de INT. del ADE
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf   LENG_DATOS,0x01 ;leer 2 bytes
    call  SPIRX

    btfss  INSD2,0x02      ;la fuente de la int. fue por el fin de la acumulacion
                                ;de energia de LAENERGY y LVAENERGY?

    goto   espera_1_fase_a_clear      ;no, seguir esperando

;leemos LAENERGY y LVAENERGY solo para limpiar estos reg.
    movlw  LAENERGY        ;cargamos el comando para leer el reg.LAENERGY
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf   LENG_DATOS,0x02 ;leer 3 bytes
    call  SPIRX            ;los valores leidos no se guardan

    movlw  LVAENERGY       ;cargamos el comando para leer el reg.LAENERGY
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf   LENG_DATOS,0x02 ;leer 3 bytes
    call  SPIRX            ;los valores leidos no se guardan

;configuracion de los acumuladores LAENERGY y LVAENERGY

    btfsc  TIPO_SERVICIO,0x03      ;tipo de servicio 9s ?
    goto   conf_9s_reg_4           ;si
    goto   tipo_deservicio_4_1 ;no

;----- configuramos las formulas para tipo de servicio 9S -----
conf_9s_reg_4
    movlw  b'00100100'           ;conf. la fase A para acumular P en LAENERGY (fijamos la
                                ;formula)
    movwf  OUTD1                 ;P=Va*Ia
    movlw  WATMODE_W             ;comando de escritura a WATMODE
    movwf  SPI_COMANDO
    bsf   LENG_DATOS,0x03 ;enviar solo un byte
    call  SPIDX

    movlw  b'00100100'           ;conf. la fase A para acumular S en LVAENERGY (fijamos la
                                ;formula)
    movwf  OUTD1                 ;S=Va*Ia
    movlw  VAMODE_W             ;comando de escritura a VAMODE
    movwf  SPI_COMANDO
    bsf   LENG_DATOS,0x03 ;enviar solo un byte
    call  SPIDX

    movlw  b'00000000'           ;conf. LAENERGY para que en este se acumule potencia activa
                                ;(P) y no Q
    movwf  OUTD1
    movlw  WAVMODE_W            ;comando de escritura a WAVMODE
    movwf  SPI_COMANDO
    bsf   LENG_DATOS,0x03 ;enviar solo un byte
    call  SPIDX
    goto   conf_reg_4           ;vamos a medir la energia consumida

tipo_deservicio_4_1
    btfsc  TIPO_SERVICIO,0x02      ;tipo de servicio 8s ?
    goto   conf_8s_reg_4           ;si
    goto   tipo_deservicio_4_2 ;no

```

```

;----- configuramos las formulas para tipo de servicio 8S -----
conf_8s_reg_4
movlw b'10100100' ;conf. la fase A para acumular P en LAENERGY (fijamos la
;formula)
movwf OUTD1 ; $P=V_a*(I_a-I_b)$ 
movlw WATMODE_W ;comando de escritura a WATMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

movlw b'10100100' ;conf. la fase A para acumular S en LVAENERGY (fijamos la
;formula)
movwf OUTD1 ; $S=V_a*I_a$ 
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

movlw b'00000000' ;conf. LAENERGY para que en este se acumule potencia activa
;(P) y no Q
movwf OUTD1
movlw WAVMODE_W ;comando de escritura a WAVMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX
goto conf_reg_4 ;vamos a medir la energia consumida

tipo_deservicio_4_2
btfsc TIPO_SERVICIO,0x01 ;tipo de servicio 6s ?
goto conf_6s_reg_4 ;si
goto tipo_deservicio_4_3 ;no

;----- configuramos las formulas para tipo de servicio 6S -----
conf_6s_reg_4
movlw b'01100100' ;conf. la fase A para acumular P en LAENERGY (fijamos la
;formula)
movwf OUTD1 ; $P=V_a*(I_a-I_b)$ 
movlw WATMODE_W ;comando de escritura a WATMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

movlw b'01100100' ;conf. la fase A para acumular S en LVAENERGY (fijamos la
;formula)
movwf OUTD1 ; $S=V_a*I_a$ 
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

movlw b'00000000' ;conf. LAENERGY para que en este se acumule potencia activa
;(P) y no Q
movwf OUTD1
movlw WAVMODE_W ;comando de escritura a WAVMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX
goto conf_reg_4 ;vamos a medir la energia consumida

tipo_deservicio_4_3
btfsc TIPO_SERVICIO,0x00 ;tipo de servicio 5s ?
goto conf_5s_reg_4
return ;servicio desconocido (este caso nunca se puede dar)

;----- configuramos las formulas para tipo de servicio 5S -----
conf_5s_reg_4
movlw b'00100100' ;conf. la fase A para acumular P en LAENERGY (fijamos la formula)

```

```

movwf OUTD1 ;P=Va*Ia
movlw WATMODE_W ;comando de escritura a WATMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

movlw b'00100100' ;conf. la fase A para acumular S en LVAENERGY (fijamos la formula)
movwf OUTD1 ;S=Va*Ia
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

movlw b'00000000' ;conf. LAENERGY para que en este se acumule potencia activa
; (P) y no Q

movwf OUTD1
movlw WAVMODE_W ;comando de escritura a WAVMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
call SPIDX

conf_reg_4 ;vamos a medir la energia consumida

;fase a
espera_1_fase_a
btfsc INTCON,INTF ;verificamos la bandera de PBO_int si esta activada?
goto salir_1_fase_a ;si
goto espera_1_fase_a ;no

salir_1_fase_a
bcf INTCON,INTF ;borramos la bandera de int. externa
movlw 0x11 ;lectura con reset del registro de estado de int. del ADE
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01 ;leer 2 bytes
call SPIRX ;no se guarda el resultado

btfss INSD2,0x02 ;la fuente de la int. fue por el fin de la acumulacion
;de energia de LAENERGY y LVAENERGY?
goto espera_1_fase_a ;no, esperar

;repetimos la operaciones anteriores para asegurar nos que tendremos datos validos
;en LAENERGY y LVAENERGY

espera_2_fase_a
btfsc INTCON,INTF
goto leer_P_S_fase_a
goto espera_2_fase_a
leer_P_S_fase_a

bcf INTCON,INTF
movlw 0x11 ;lectura con reset
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01
call SPIRX ;no se guarda el resultado

btfss INSD2,0x02
goto espera_2_fase_a

;leemos LAENERGY y LVAENERGY y los almacenamos en variables temporales
movlw LAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02 ;leer 3 bytes
call SPIRX
movf INSD3,W

```

```

    bsf     STATUS,RP0      ;banco 1
    movwf  P_A2             ;guardamos el MSB
    bcf     STATUS,RP0      ;banco 0
    movf   INSD2,W
    bsf     STATUS,RP0      ;banco 1
    movwf  P_A1
    bcf     STATUS,RP0      ;banco 0
    movf   INSD1,W
    bsf     STATUS,RP0      ;banco 1
    movwf  P_A0             ;guardamos el LSB
    bcf     STATUS,RP0      ;banco 0

    movlw  LVAENERGY
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf    LENG_DATOS,0x02 ;leemos 3 bytes
    call   SPIRX
    movf   INSD3,W
    bsf    STATUS,RP0      ;banco 1
    movwf  S_A2             ;guardamos el MSB
    bcf    STATUS,RP0      ;banco 0
    movf   INSD2,W
    bsf    STATUS,RP0      ;banco 1
    movwf  S_A1
    bcf    STATUS,RP0      ;banco 0
    movf   INSD1,W
    bsf    STATUS,RP0      ;banco 1
    movwf  S_A0             ;guardamos el LSB
    bcf    STATUS,RP0      ;banco 0

    ;conf. el ADE para leer Q
    movlw  b'00100000'     ;conf. LAENERGY para que en este se acumule Q,(WAVMODE<5>=1)
    movwf  OUTD1
    movlw  WAVMODE_W
    movwf  SPI_COMANDO
    bsf    LENG_DATOS,0x03
    call   SPIDX

espera_Q_1_fase_a
    btfsc  INTCON,INTF      ;verificamos la bandera de PB0_int
    goto   salir_Q_1_fase_a
    goto   espera_Q_1_fase_a
salir_Q_1_fase_a

    bcf    INTCON,INTF
    movlw  0x11             ;lectura con reset del registro de estado de int. del ADE
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf    LENG_DATOS,0x01
    call   SPIRX            ;no se guarda el resultado

    btfss  INSD2,0x02      ;la fuente de la int. fue por el fin de la acumulacion
                                ;de energia de LAENERGY y LVAENERGY?

    goto   espera_Q_1_fase_a ;no

    ;repetimos la operaciones anteriores para asegurar nos que tendremos datos validos
    ; en LAENERGY

espera_Q_2_fase_a
    btfsc  INTCON,INTF      ;verificamos la bandera de PB0_int
    goto   leer_Q_fase_a
    goto   espera_Q_2_fase_a
leer_Q_fase_a

    bcf    INTCON,INTF

```

```

movlw 0x11 ;lectura con reset de registro de estado de int. del ADE
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01
call SPIRX ;no se guarda el resultado

btfss INSD2,0x02 ;la fuente de la int. fue por el fin de la acumulacion
;de energia de LAENERGY y LVAENERGY?

goto espera_Q_2_fase_a ;no

movlw LAENERGY ;leemos el signo de Q en el reg.LAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02 ;leer 3 bytes
call SPIRX
movf INSD3,W
bsf STATUS,RP0 ;banco 1
movwf Q_A2 ;guardamos el MSB
bcf STATUS,RP0 ;banco 0
movf INSD2,W
bsf STATUS,RP0 ;banco 1
movwf Q_A1
bcf STATUS,RP0 ;banco 0
movf INSD1,W
bsf STATUS,RP0 ;banco 1
movwf Q_A0 ;guardamos el LSB
bcf STATUS,RP0 ;banco 0

;fase b
;conf.

btfsc TIPO_SERVICIO,0x03 ;tipo de servicio 9s ?
goto conf_9s_reg_2
goto tipo_deservicio_2_1
conf_9s_reg_2

movlw b'00010010' ;definimos P=Vb*Ib
movwf OUTD1
movlw WATMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'00010010' ; definimos S=Vb*Ib
movwf OUTD1
movlw VAMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'00000000' ;conf. el ADE para medir P y S (WAVMODE<5>=0)
movwf OUTD1
movlw WAVMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX
goto conf_reg_2

tipo_deservicio_2_1

btfsc TIPO_SERVICIO,0x02 ;tipo de servicio 8s ?
goto conf_8s_reg_2
goto tipo_deservicio_2_2
conf_8s_reg_2

movlw b'10000000' ;definimos P=0

```



```

movwf OUTD1
movlw WATMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'10010010' ;definimos S=Va*Ib
movwf OUTD1
movlw VAMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'00000000' ;conf. el ADE para medir P y S (WAVMODE<5>=0)
movwf OUTD1
movlw WAVMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX
goto conf_reg_2

```

tipo_deservicio_2_2

```

btfsc TIPO_SERVICIO,0x01 ;tipo de servicio 6s ?
goto conf_6s_reg_2
goto tipo_deservicio_2_3
conf_6s_reg_2

```

```

movlw b'01000000' ;definimos P=0
movwf OUTD1
movlw WATMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'01010010' ;definimos S=(Va+Vc)/2*Ib
movwf OUTD1
movlw VAMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'00000000' ;conf. el ADE para medir P y S (WAVMODE<5>=0)
movwf OUTD1
movlw WAVMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX
goto conf_reg_2

```

tipo_deservicio_2_3

```

btfsc TIPO_SERVICIO,0x00 ;tipo de servicio 5s ?
goto conf_5s_reg_2
goto conf_reg_2
conf_5s_reg_2

```

```

movlw b'00000000' ;definimos P=0
movwf OUTD1
movlw WATMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'00000000' ;definimos S=0
movwf OUTD1
movlw VAMODE_W

```

```

movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movlw b'00000000' ;conf. el ADE para medir P y S (WAVMODE<5>=0)
movwf OUTD1
movlw WAVMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

```

conf_reg_2

;lectura de P y S

```

espera_1_fase_b
btfsc INTCON,INTF ;verificamos la bandera de PB0_int
goto salir_1_fase_b
goto espera_1_fase_b
salir_1_fase_b
bcf INTCON,INTF
movlw 0x11 ;lectura con reset de registro de estado de int. del ADE
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01
call SPIRX ;no se guarda el resultado

btfss INSD2,0x02 ;la fuente de la int. fue por el fin de la acumulacion
;de energia de LAENERGY y LVAENERGY?
goto espera_1_fase_b ;no, esperar

espera_2_fase_b
btfsc INTCON,INTF ;verificamos la bandera de PB0_int
goto leer_P_S_fase_b
goto espera_2_fase_b
leer_P_S_fase_b

bcf INTCON,INTF
movlw 0x11 ;lectura con reset de registro de estado de int. del ADE
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01
call SPIRX ;no se guarda el resultado

btfss INSD2,0x02
goto espera_2_fase_b

;leer LAENERGY (P)
movlw LAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02
call SPIRX
movf INSD3,W
bsf STATUS,RP0 ;banco 1
movwf P_B2 ;guardamos el MSB
bcf STATUS,RP0 ;banco 0
movf INSD2,W
bsf STATUS,RP0 ;banco 1
movwf P_B1
bcf STATUS,RP0 ;banco 0
movf INSD1,W
bsf STATUS,RP0 ;banco 1
movwf P_B0 ;guardamos el LSB
bcf STATUS,RP0 ;banco 0

```

```

;leemos LVAENERGY (S)
movlw   LVAENERGY
movwfw SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02
call    SPIRX
movf    INSD3,W
bsf     STATUS,RP0           ;banco 1
movwfw S_B2                 ;guardamos el MSB
bcf     STATUS,RP0           ;banco 0
movf    INSD2,W
bsf     STATUS,RP0           ;banco 1
movwfw S_B1                 ;banco 0
bcf     STATUS,RP0           ;banco 0
movf    INSD1,W
bsf     STATUS,RP0           ;banco 1
movwfw S_B0                 ;guardamos el LSB
bcf     STATUS,RP0           ;banco 0

;conf. el ADE para leer Q
movlw   b'00100000'         ;conf. LAENERGY para que en este se acumule Q,(WAVMODE<5>=1)
movwfw OUTD1
movlw   WAVMODE_W
movwfw SPI_COMANDO
bsf     LENG_DATOS,0x03
call    SPIDX

espera_Q_1_fase_b
  btfsc INTCON,INTF         ;verificamos la bandera de PB0_int
  goto  salir_Q_1_fase_b
  goto  espera_Q_1_fase_b
salir_Q_1_fase_b

  bcf    INTCON,INTF
  movlw  0x11                ;lectura con reset de registro de estado de int. del ADE
  movwfw SPI_COMANDO
  clrf   LENG_DATOS
  bsf    LENG_DATOS,0x01
  call   SPIRX              ;no se guarda el resultado

  btfss  INSD2,0x02
  goto   espera_Q_1_fase_b

espera_Q_2_fase_b
  btfsc INTCON,INTF         ;verificamos la bandera de PB0_int
  goto  leer_Q_fase_b
  goto  espera_Q_2_fase_b
leer_Q_fase_b

  bcf    INTCON,INTF
  movlw  0x11                ;lectura con reset de registro de estado de int. del ADE
  movwfw SPI_COMANDO
  clrf   LENG_DATOS
  bsf    LENG_DATOS,0x01
  call   SPIRX              ;no se guarda el resultado

  btfss  INSD2,0x02
  goto   espera_Q_2_fase_b

  movlw  LAENERGY           ;leemos el signo de Q
  movwfw SPI_COMANDO
  clrf   LENG_DATOS
  bsf    LENG_DATOS,0x02
  call   SPIRX
  movf   INSD3,W
  bsf    STATUS,RP0           ;banco 1
  movwfw Q_B2               ;guardamos el MSB

```

```

bcf     STATUS,RP0           ;banco 0
movf   INSD2,W
bsf    STATUS,RP0           ;banco 1
movwf  Q_B1
bcf    STATUS,RP0           ;banco 0
movf   INSD1,W
bsf    STATUS,RP0           ;banco 1
movwf  Q_B0                 ;guardamos el LSB
bcf    STATUS,RP0           ;banco 0

;fase C
;conf.

btfsc  TIPO_SERVICIO,0x03    ;tipo de servicio 9s ?
goto   conf_9s_reg_3        ;si
goto   tipo_deservico_3_1    ;no
conf_9s_reg_3

movlw  b'00001001'          ;definimos P=Vc*Ic
movwf  OUTD1
movlw  WATMODE_W
movwf  SPI_COMANDO
bsf    LENG_DATOS,0x03
call   SPIDX

movlw  b'00001001'          ;definimos S=Vc*Ic
movwf  OUTD1
movlw  VAMODE_W
movwf  SPI_COMANDO
bsf    LENG_DATOS,0x03
call   SPIDX

movlw  b'00000000'          ;conf. el ADE para medir P y S (WAVMODE<5>=0)
movwf  OUTD1
movlw  WAVMODE_W
movwf  SPI_COMANDO
bsf    LENG_DATOS,0x03
call   SPIDX
goto   conf_reg_3

tipo_deservico_3_1

btfsc  TIPO_SERVICIO,0x02    ;tipo de servicio 8s ?
goto   conf_8s_reg_3        ;si
goto   tipo_deservico_3_2    ;no
conf_8s_reg_3

movlw  b'10001001'          ;definimos P=Vc*Ic
movwf  OUTD1
movlw  WATMODE_W
movwf  SPI_COMANDO
bsf    LENG_DATOS,0x03
call   SPIDX

movlw  b'10001001'          ;definimos S=Vc*Ic
movwf  OUTD1
movlw  VAMODE_W
movwf  SPI_COMANDO
bsf    LENG_DATOS,0x03
call   SPIDX

movlw  b'00000000'          ;conf. el ADE para medir P y S (WAVMODE<5>=0)
movwf  OUTD1
movlw  WAVMODE_W
movwf  SPI_COMANDO
bsf    LENG_DATOS,0x03
call   SPIDX

```

```

        goto    conf_reg_3

tipo_deservico_3_2

        btfsc  TIPO_SERVICIO,0x01      ;tipo de servicio 6s ?
        goto   conf_6s_reg_3
        goto   tipo_deservico_3_3
conf_6s_reg_3

        movlw  b'01001001'           ;definimos P=Vc(Ic-Ib)
        movwf  OUTD1
        movlw  WATMODE_W
        movwf  SPI_COMANDO
        bsf   LENG_DATOS,0x03
        call  SPIDX

        movlw  b'01001001'           ;definimos S=Vc*Ic
        movwf  OUTD1
        movlw  VAMODE_W
        movwf  SPI_COMANDO
        bsf   LENG_DATOS,0x03
        call  SPIDX

        movlw  b'00000000'           ;conf. el ADE para medir P y S (WAVMODE<5>=0)
        movwf  OUTD1
        movlw  WAVMODE_W
        movwf  SPI_COMANDO
        bsf   LENG_DATOS,0x03
        call  SPIDX
        goto  conf_reg_3

tipo_deservico_3_3

        btfsc  TIPO_SERVICIO,0x00      ;tipo de servicio 5s ?
        goto   conf_5s_reg_3
        goto   conf_reg_3
conf_5s_reg_3

        movlw  b'00001001'           ;definimos P=Vc*Ic
        movwf  OUTD1
        movlw  WATMODE_W
        movwf  SPI_COMANDO
        bsf   LENG_DATOS,0x03
        call  SPIDX

        movlw  b'00001001'           ;definimos S=Vc*Ic
        movwf  OUTD1
        movlw  VAMODE_W
        movwf  SPI_COMANDO
        bsf   LENG_DATOS,0x03
        call  SPIDX

        movlw  b'00000000'           ;conf. el ADE para medir P y S (WAVMODE<5>=0)
        movwf  OUTD1
        movlw  WAVMODE_W
        movwf  SPI_COMANDO
        bsf   LENG_DATOS,0x03
        call  SPIDX

conf_reg_3

        ;lectura de P y S

espera_1_fase_c

        btfsc  INTCON,INTF           ;verificamos la bandera de PB0_int
        goto  salir_1_fase_c
        goto  espera_1_fase_c

```

salir_1_fase_c

```
bcf      INTCON,INTF
movlw   0x11          ;lectura con reset de registro de estado de int. del ADE
movwfw SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01
call    SPIRX        ;no se guarda el resultado

btfss   INSD2,0x02
goto    espera_1_fase_c
```

espera_2_fase_c

```
btfsc   INTCON,INTF    ;verificamos la bandera de PB0_int
goto    leer_P_S_fase_c
goto    espera_2_fase_c
```

leer_P_S_fase_c

```
bcf      INTCON,INTF
movlw   0x11          ;lectura con reset de registro de estado de int. del ADE
movwfw SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x01
call    SPIRX        ;no se guarda el resultado

btfss   INSD2,0x02
goto    espera_2_fase_c
```

```
movlw   LAENERGY          ;leemos la potencia activa(P)
movwfw SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02
call    SPIRX
movf    INSD3,W
bsf     STATUS,RP0        ;banco 1
movwfw P_C2              ;guardamos el byte MSB
bcf     STATUS,RP0        ;banco 0
movf    INSD2,W
bsf     STATUS,RP0        ;banco 1
movwfw P_C1              ;banco 0
bcf     STATUS,RP0
movf    INSD1,W
bsf     STATUS,RP0        ;banco 1
movwfw P_C0              ;guardamos el byte LSB
bcf     STATUS,RP0        ;banco 0
```

```
movlw   LVAENERGY        ;leemos la potencia aparente (S)
movwfw SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02
call    SPIRX
movf    INSD3,W
bsf     STATUS,RP0        ;banco 1
movwfw S_C2              ;guardamos el byte MSB
bcf     STATUS,RP0        ;banco 0
movf    INSD2,W
bsf     STATUS,RP0        ;banco 1
movwfw S_C1              ;banco 0
bcf     STATUS,RP0
movf    INSD1,W
bsf     STATUS,RP0        ;banco 1
movwfw S_C0              ;guardamos el byte LSB
bcf     STATUS,RP0        ;banco 0
```

```
;conf. el ADE para leer Q
movlw   b'00100000'
movwfw OUTD1
```

```
;conf. LAENERGY para que en este se acumule Qr(WAVMODE<5>=1)
```

```

        movlw   WAVMODE_W
        movwf  SPI_COMANDO
        bsf    LENG_DATOS,0x03
        call   SPIDX

espera_Q_1_fase_c
        btfsc  INTCON,INTF      ;verificamos la bandera de PB0_int
        goto  salir_Q_1_fase_c
        goto  espera_Q_1_fase_c
salir_Q_1_fase_c

        bcf    INTCON,INTF
        movlw  0x11              ;lectura con reset de registro de estado de int. del ADE
        movwf  SPI_COMANDO
        clrf  LENG_DATOS
        bsf    LENG_DATOS,0x01
        call   SPIRX             ;no se guarda el resultado

        btfss  INSD2,0x02
        goto  espera_Q_1_fase_c

espera_Q_2_fase_c
        btfsc  INTCON,INTF      ;verificamos la bandera de PB0_int
        goto  leer_Q_fase_c
        goto  espera_Q_2_fase_c
leer_Q_fase_c

        bcf    INTCON,INTF
        movlw  0x11              ;lectura con reset de registro de estado de int. del ADE
        movwf  SPI_COMANDO
        clrf  LENG_DATOS
        bsf    LENG_DATOS,0x01
        call   SPIRX             ;no se guarda el resultado

        btfss  INSD2,0x02
        goto  espera_Q_2_fase_c

        movlw  LAENERGY         ;leemos el signo de Q
        movwf  SPI_COMANDO
        clrf  LENG_DATOS
        bsf    LENG_DATOS,0x02
        call   SPIRX
        movf  INSD3,W
        bsf    STATUS,RP0        ;banco 1
        movwf  Q_C2              ;guardamos el byte MSB
        bcf    STATUS,RP0        ;banco 0
        movf  INSD2,W
        bsf    STATUS,RP0        ;banco 1
        movwf  Q_C1
        bcf    STATUS,RP0        ;banco 0
        movf  INSD1,W
        bsf    STATUS,RP0        ;banco 1
        movwf  Q_C0              ;guardamos el byte LSB
        bcf    STATUS,RP0        ;banco 0

        return

;-----
;
; Rutina para leer todas las corrientes y los voltajes RMS del ADE7754
; argumentos: no hay
;
;-----
LEER_REG_RMS
        bcf    STATUS,RP1        ;Selecciona banco 0 y 1

```

```

bcf     STATUS,RP0      ;Selecciona banco 0

;VArms
movlw   AVRMS           ;leemos Varms
movwfw  SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02 ;leeremos 3 bytes
call    SPIRX
movf    INSD3,W
movwfw  AVRMS2          ;byte MSB
movf    INSD2,W
movwfw  AVRMS1
movf    INSD1,W
movwfw  AVRMS0          ;byte LSB

;VBrms
movlw   BVRMS           ;leemos Vbrms
movwfw  SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02 ;leeremos 3 bytes
call    SPIRX
movf    INSD3,W
movwfw  BVRMS2          ;byte MSB
movf    INSD2,W
movwfw  BVRMS1
movf    INSD1,W
movwfw  BVRMS0          ;byte LSB

;VCrms
movlw   CVRMS           ;leemos Vcrms
movwfw  SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02 ;leeremos 3 bytes
call    SPIRX
movf    INSD3,W
movwfw  CVRMS2          ;byte MSB
movf    INSD2,W
movwfw  CVRMS1
movf    INSD1,W
movwfw  CVRMS0          ;byte LSB

;AIRMS
movlw   AIRMS           ;leemos Iarms
movwfw  SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02
call    SPIRX
movf    INSD3,W
movwfw  AIRMS2
movf    INSD2,W
movwfw  AIRMS1
movf    INSD1,W
movwfw  AIRMS0

;BIRMS
movlw   BIRMS           ;leemos Birms
movwfw  SPI_COMANDO
clrf    LENG_DATOS
bsf     LENG_DATOS,0x02
call    SPIRX
movf    INSD3,W
movwfw  BIRMS2
movf    INSD2,W
movwfw  BIRMS1
movf    INSD1,W

```



```

movwf  BIRMS0

;CIRMS
movlw  CIRMS  ;leemos Icrms
movwf  SPI_COMANDO
clrf   LENG_DATOS
bsf    LENG_DATOS,0x02
call   SPIRX
movf   INSD3,W
movwf  CIRMS2
movf   INSD2,W
movwf  CIRMS1
movf   INSD1,W
movwf  CIRMS0

return

```

```

;-----
;
; Rutina para enviar todos los registros de conf. al ADE7754
; argumentos: no hay
;
; - LENG_DATOS
; - direccion en donde se escribira SPI_COMANDO
; - OUTD2 byte superior
; - OUTD1 byte inferior
;-----

```

```

ENVIAR_REG_CONF_AL_ADE
bcf    STATUS, RP1    ;banco 0 y 1
bcf    STATUS, RP0    ;banco 0

movf   LINCYC1,W      ;enviamos el numero de ciclos de la señal medida, utilizados
movwf  OUTD2          ;para la acumulacion de energia
movf   LINCYC0,W
movwf  OUTD1
movlw  LINCYC_W
movwf  SPI_COMANDO
bcf    LENG_DATOS,0x03 ;enviaremos 2 bytes
call   SPIDX

movf   AWG1,W         ;ganancia de potencia activa para la fase A
movwf  OUTD2
movf   AWG0,W
movwf  OUTD1
movlw  AWG_W
movwf  SPI_COMANDO
bcf    LENG_DATOS,0x03 ;enviaremos 2 bytes
call   SPIDX

movf   BWG1,W         ;ganancia de potencia activa para la fase B
movwf  OUTD2
movf   BWG0,W
movwf  OUTD1
movlw  BWG_W
movwf  SPI_COMANDO
bcf    LENG_DATOS,0x03
call   SPIDX

movf   CWG1,W         ;ganancia de potencia activa para la fase C
movwf  OUTD2
movf   CWG0,W
movwf  OUTD1
movlw  CWG_W
movwf  SPI_COMANDO
bcf    LENG_DATOS,0x03
call   SPIDX

movf   AVAG1,W        ;ganancia de potencia aparente para la fase A

```

```

movwf OUTD2
movf AVAG0,W
movwf OUTD1
movlw AVAG_W
movwf SPI_COMANDO
bcf LENG_DATOS,0x03
call SPIDX

movf BVAG1,W ;ganancia de potencia aparente para la fase B
movwf OUTD2
movf BVAG0,W
movwf OUTD1
movlw BVAG_W
movwf SPI_COMANDO
bcf LENG_DATOS,0x03
call SPIDX

movf CVAG1,W ;ganancia de potencia aparente para la fase C
movwf OUTD2
movf CVAG0,W
movwf OUTD1
movlw CVAG_W
movwf SPI_COMANDO
bcf LENG_DATOS,0x03
call SPIDX

movf APHCAL0,W ;enviar reg. calibracion de fase, para la fase A
movwf OUTD1
movlw APHCAL_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviaremos 1 byte
call SPIDX

movf BPHCAL0,W ;enviar reg. calibracion de fase, para la fase B
movwf OUTD1
movlw BPHCAL_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movf CPHCAL0,W ;enviar reg. calibracion de fase, para la fase C
movwf OUTD1
movlw CPHCAL_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
call SPIDX

movf AAPOS1,W ;enviar reg. calibracion de offset de potencia activa, para la
;fase A
movwf OUTD2
movf AAPOS0,W
movwf OUTD1
movlw AAPOS_W
movwf SPI_COMANDO
bcf LENG_DATOS,0x03
call SPIDX

movf BAPOS1,W ;enviar reg. calibracion de offset de potencia activa, para la
;fase B
movwf OUTD2
movf BAPOS0,W
movwf OUTD1
movlw BAPOS_W
movwf SPI_COMANDO
bcf LENG_DATOS,0x03
call SPIDX

```

```

movf    CAPOS1,W          ;enviar reg. calibracion de offset de potencia activa, para la
                                ;fase C
movwf   OUTD2
movf    CAPOS0,W
movwf   OUTD1
movlw   CAPOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    CFNUM1,W          ;enviar reg. numerado del escalamiento de CF
movwf   OUTD2
movf    CFNUM0,W
movwf   OUTD1
movlw   CFNUM_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    CFDEM1,W          ;enviar reg. denominador del escalamiento de CF
                                ;CF=(CFNUM/CFDEM)
movwf   OUTD2
movf    CFDEM0,W
movwf   OUTD1
movlw   CFDEM_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    AIRMSOS_1,W      ;enviar reg. calibracion de offset de corriente, para la fase A
movwf   OUTD2
movf    AIRMSOS_0,W
movwf   OUTD1
movlw   AIRMSOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    BIRMSOS_1,W      ;enviar reg. calibracion de offset de corriente, para la fase B
movwf   OUTD2
movf    BIRMSOS_0,W
movwf   OUTD1
movlw   BIRMSOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    CIRMSOS_1,W      ;enviar reg. calibracion de offset de corriente, para la fase C
movwf   OUTD2
movf    CIRMSOS_0,W
movwf   OUTD1
movlw   CIRMSOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    AVRMSOS_1,W      ;enviar reg. calibracion de offset de voltaje, para la fase A
movwf   OUTD2
movf    AVRMSOS_0,W
movwf   OUTD1
movlw   AVRMSOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    BVRMSOS_1,W      ;enviar reg. calibracion de offset de voltaje, para la fase B
movwf   OUTD2

```

```

movf    BVRMSOS_0,W
movwf   OUTD1
movlw   BVRMSOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    CVRMSOS_1,W ;enviar reg. calibracion de offset de voltaje, para la fase C
movwf   OUTD2
movf    CVRMSOS_0,W
movwf   OUTD1
movlw   CVRMSOS_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    AAPGAIN1,W ;ajuste de ganancia de potencia activa, para la fase A
movwf   OUTD2
movf    AAPGAIN0,W
movwf   OUTD1
movlw   AAPGAIN_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    BAPGAIN1,W ;ajuste de ganancia de potencia activa, para la fase B
movwf   OUTD2
movf    BAPGAIN0,W
movwf   OUTD1
movlw   BAPGAIN_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    CAPGAIN1,W ;ajuste de ganancia de potencia activa, para la fase B
movwf   OUTD2
movf    CAPGAIN0,W
movwf   OUTD1
movlw   CAPGAIN_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    AVGAIN1,W      ;ganancia de voltaje rms, para la fase A
movwf   OUTD2
movf    AVGAIN0,W
movwf   OUTD1
movlw   AVGAIN_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    BVGAIN1,W      ;ganancia de voltaje rms, para la fase B
movwf   OUTD2
movf    BVGAIN0,W
movwf   OUTD1
movlw   BVGAIN_W
movwf   SPI_COMANDO
bcf     LENG_DATOS,0x03
call    SPIDX

movf    CVGAIN1,W      ;ganancia de voltaje rms, para la fase C
movwf   OUTD2
movf    CVGAIN0,W
movwf   OUTD1
movlw   CVGAIN_W

```

```

movwf  SPI_COMANDO
bcf    LENG_DATOS,0x03
call   SPIDX

return
;-----
;
; Rutina para enviar todos los registros de conf. leidos del ADE7754 a la PC.
; argumentos: no hay
;
;NOTA: para los reg. de 2 bytes o mas, se envia siempre el MSB de primero
;       y por ultimo el LSB
;-----
ENVIAR_REG_CONF_PC
bcf    STATUS, RP1
bcf    STATUS, RP0      ;nos aseguramos que este en el banco 0

;registros de conf.
movf   AWG1,W           ;enviamos el reg. AWG
call   Tx_Dato_sci     ;enviamos el byte MSB
movf   AWG0,W
call   Tx_Dato_sci     ;enviamos el byte LSB

movf   BWG1,W           ;enviamos el reg. BWG
call   Tx_Dato_sci
movf   BWG0,W
call   Tx_Dato_sci

movf   CWG1,W           ;enviamos el reg. CWG
call   Tx_Dato_sci
movf   CWG0,W
call   Tx_Dato_sci

movf   AVAG1,W          ;enviamos el reg. AVAG
call   Tx_Dato_sci
movf   AVAG0,W
call   Tx_Dato_sci

movf   BVAG1,W          ;enviamos el reg. BVAG
call   Tx_Dato_sci
movf   BVAG0,W
call   Tx_Dato_sci

movf   CVAG1,W          ;enviamos el reg. CVAG
call   Tx_Dato_sci
movf   CVAG0,W
call   Tx_Dato_sci

movf   APHCAL0,W        ;enviamos el reg. APHCAL
call   Tx_Dato_sci

movf   BPHCAL0,W        ;enviamos el reg. BPHCAL
call   Tx_Dato_sci

movf   CPHCAL0,W        ;enviamos el reg. CPHCAL
call   Tx_Dato_sci

movf   AAPOS1,W         ;enviamos el reg. AAPOS
call   Tx_Dato_sci
movf   AAPOS0,W
call   Tx_Dato_sci

movf   BAPOS1,W         ;enviamos el reg. BAPOS
call   Tx_Dato_sci
movf   BAPOS0,W
call   Tx_Dato_sci

```

movf	CAPOS1,W	;enviamos el reg. CAPOS
call	Tx_Dato_sci	
movf	CAPOS0,W	
call	Tx_Dato_sci	
movf	LINCYC1,W	;enviamos el reg. LINCYC
call	Tx_Dato_sci	
movf	LINCYC0,W	
call	Tx_Dato_sci	
movf	CFNUM1,W	;enviamos el reg. CFNUM
call	Tx_Dato_sci	
movf	CFNUM0,W	
call	Tx_Dato_sci	
movf	CFDEN1,W	;enviamos el reg. CFDEN
call	Tx_Dato_sci	
movf	CFDEN0,W	
call	Tx_Dato_sci	
movf	AIRMSOS_1,W	;enviamos el reg. AIRMSOS
call	Tx_Dato_sci	
movf	AIRMSOS_0,W	
call	Tx_Dato_sci	
movf	BIRMSOS_1,W	;enviamos el reg. BIRMSOS
call	Tx_Dato_sci	
movf	BIRMSOS_0,W	
call	Tx_Dato_sci	
movf	CIRMSOS_1,W	;enviamos el reg. CIRMSOS
call	Tx_Dato_sci	
movf	CIRMSOS_0,W	
call	Tx_Dato_sci	
movf	AVRMSOS_1,W	;enviamos el reg. AVRMSOS
call	Tx_Dato_sci	
movf	AVRMSOS_0,W	
call	Tx_Dato_sci	
movf	BVRMSOS_1,W	;enviamos el reg. BVRMSOS
call	Tx_Dato_sci	
movf	BVRMSOS_0,W	
call	Tx_Dato_sci	
movf	CVRMSOS_1,W	;enviamos el reg. CVRMSOS
call	Tx_Dato_sci	
movf	CVRMSOS_0,W	
call	Tx_Dato_sci	
movf	AAPGAIN1,W	;enviamos el reg. AAPGAIN
call	Tx_Dato_sci	
movf	AAPGAIN0,W	
call	Tx_Dato_sci	
movf	BAPGAIN1,W	;enviamos el reg. BAPGAIN
call	Tx_Dato_sci	
movf	BAPGAIN0,W	
call	Tx_Dato_sci	
movf	CAPGAIN1,W	;enviamos el reg. CAPGAIN
call	Tx_Dato_sci	
movf	CAPGAIN0,W	
call	Tx_Dato_sci	
movf	AVGAIN1,W	;enviamos el reg. AVGAIN

```

call    Tx_Dato_sci
movf   AVGAIN0,W
call   Tx_Dato_sci

movf   BVGAIN1,W           ;enviamos el reg. BVGAIN
call   Tx_Dato_sci
movf   BVGAIN0,W
call   Tx_Dato_sci

movf   CVGAIN1,W           ;enviamos el reg. CVGAIN
call   Tx_Dato_sci
movf   CVGAIN0,W
call   Tx_Dato_sci

```

```

;-----
;
; Rutina para enviar todos los valores de energias, voltajes y corrientes RMS a la PC.
; argumentos: no hay
;
;-----

```

```

ENVIAR_REG_ENERGY_Y_RMS_PC   ;Funcion que envia los e tipos de potencias, voltajes y corrienteslos RMS

```

```

;potencias
;FASE A
bsf    STATUS,RP0         ;banco 1

movf   P_A2,W             ;byte MSB, de la potencia activa
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   P_A1,W
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   P_A0,W             ;byte LSB, de la potencia activa
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   S_A2,W             ;byte MSB, de la potencia aparente
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   S_A1,W
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   S_A0,W             ;byte LSB, de la potencia aparente
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   Q_A2,W             ;byte MSB, de la potencia reactiva
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

bsf    STATUS,RP0         ;banco 1
movf   Q_A1,W
bcf    STATUS,RP0         ;banco 0
call   Tx_Dato_sci

```

```

bsf STATUS,RP0 ;banco 1
movf Q_A0,W ;byte LSB, de la potencia reactiva
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

;FASE B
bsf STATUS,RP0 ;banco 1
movf P_B2,W ;byte MSB, de la potencia activa
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf P_B1,W
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf P_B0,W ;byte LSB, de la potencia activa
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf S_B2,W ;byte MSB, de la potencia aparente
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf S_B1,W
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf S_B0,W ;byte LSB, de la potencia aparente
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf Q_B2,W ;byte MSB, de la potencia reactiva
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf Q_B1,W
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf Q_B0,W ;byte LSB, de la potencia reactiva
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

;FASE C
bsf STATUS,RP0 ;banco 1
movf P_C2,W ;byte MSB, de la potencia activa
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf P_C1,W
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf P_C0,W ;byte LSB, de la potencia activa

```



```

bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

bsf     STATUS,RP0      ;banco 1
movf   S_C2,W           ;byte MSB, de la potencia aparente
bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

bsf     STATUS,RP0      ;banco 1
movf   S_C1,W
bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

bsf     STATUS,RP0      ;banco 1
movf   S_C0,W           ;byte LSB, de la potencia aparente
bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

bsf     STATUS,RP0      ;banco 1
movf   Q_C2,W           ;byte MSB, de la potencia reactiva
bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

bsf     STATUS,RP0      ;banco 1
movf   Q_C1,W
bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

bsf     STATUS,RP0      ;banco 1
movf   Q_C0,W           ;byte LSB, de la potencia reactiva
bcf     STATUS,RP0      ;banco 0
call   Tx_Dato_sci

```

```

;-----
;
;
;   Rutina para enviar al PC los voltajes y corrientes RMS leidos del ADE7754
;   argumentos: no hay
;
;-----

```

```

ENVIAR_REG_RMS_PC
;valores rms
movf   AVRMS2,W         ;byte MSB, de Va
call   Tx_Dato_sci

movf   AVRMS1,W
call   Tx_Dato_sci

movf   AVRMS0,W         ;byte LSB, de Va
call   Tx_Dato_sci

movf   BVRMS2,W         ;byte MSB, de Vb
call   Tx_Dato_sci

movf   BVRMS1,W
call   Tx_Dato_sci

movf   BVRMS0,W         ;byte LSB, de Vb
call   Tx_Dato_sci

movf   CVRMS2,W         ;byte MSB, de Vc
call   Tx_Dato_sci

movf   CVRMS1,W
call   Tx_Dato_sci

```

```

movf   CVRMS0,W      ;byte LSB, de Vc
call   Tx_Dato_sci

movf   AIRMS2,W      ;byte MSB, de Ia
call   Tx_Dato_sci

movf   AIRMS1,W
call   Tx_Dato_sci

movf   AIRMS0,W      ;byte LSB, de Ia
call   Tx_Dato_sci

movf   BIRMS2,W      ;byte MSB, de Ib
call   Tx_Dato_sci

movf   BIRMS1,W
call   Tx_Dato_sci

movf   BIRMS0,W      ;byte LSB, de Ib
call   Tx_Dato_sci

movf   CIRMS2,W      ;byte MSB, de Ic
call   Tx_Dato_sci

movf   CIRMS1,W
call   Tx_Dato_sci

movf   CIRMS0,W      ;byte LSB, de Ib
call   Tx_Dato_sci

return

```

```

;-----
;
;
;   Rutina para recibir todos los registros de conf. para el ADE7754 desde la PC.
;   argumentos: no hay
;
;NOTA: la PC envia siempre primero el MSB de un reg. y por ultimo el LSB
;-----

```

```

LEER_REG_CONF_PC
call   leer_un_byte   ;leemos el byte MSB de AWG enviado via sci
movwf  AWG1
call   leer_un_byte   ;leemos el byte LSB de AWG enviado via sci
movwf  AWG0

call   leer_un_byte   ;leemos el byte MSB de BWG enviado via sci
movwf  BWG1
call   leer_un_byte   ;leemos el byte LSB de BWG enviado via sci
movwf  BWG0

call   leer_un_byte   ;leer CWG
movwf  CWG1
call   leer_un_byte
movwf  CWG0

call   leer_un_byte   ;leer AVAG
movwf  AVAG1
call   leer_un_byte
movwf  AVAG0

call   leer_un_byte   ;leer BVAG
movwf  BVAG1
call   leer_un_byte
movwf  BVAG0

call   leer_un_byte   ;leer CVAG
movwf  CVAG1

```

```

call    leer_un_byte
movwf  CVAG0

call    leer_un_byte      ;leer APHCAL
movwf  APHCAL0

call    leer_un_byte      ;leer BPHCAL
movwf  BPHCAL0

call    leer_un_byte      ;leer CPHCAL
movwf  CPHCAL0

call    leer_un_byte      ;leer AAPOS
movwf  AAPOS1
call    leer_un_byte
movwf  AAPOS0

call    leer_un_byte      ;leer BAPOS
movwf  BAPOS1
call    leer_un_byte
movwf  BAPOS0

call    leer_un_byte      ;leer CAPOS
movwf  CAPOS1
call    leer_un_byte
movwf  CAPOS0

call    leer_un_byte      ;leer LINCYC
movwf  LINCYC1
call    leer_un_byte
movwf  LINCYC0

call    leer_un_byte      ;leer CFNUM
movwf  CFNUM1
call    leer_un_byte
movwf  CFNUM0

call    leer_un_byte      ;leer CFDEN
movwf  CFDEN1
call    leer_un_byte
movwf  CFDEN0

call    leer_un_byte      ;leer AIRMSOS
movwf  AIRMSOS_1
call    leer_un_byte
movwf  AIRMSOS_0

call    leer_un_byte      ;leer BIRMSOS
movwf  BIRMSOS_1
call    leer_un_byte
movwf  BIRMSOS_0

call    leer_un_byte      ;leer CIRMSOS
movwf  CIRMSOS_1
call    leer_un_byte
movwf  CIRMSOS_0

call    leer_un_byte      ;leer AVRMSOS
movwf  AVRMSOS_1
call    leer_un_byte
movwf  AVRMSOS_0

call    leer_un_byte      ;leer BVRMSOS
movwf  BVRMSOS_1
call    leer_un_byte
movwf  BVRMSOS_0

```

```

call    leer_un_byte    ;leer CVRMSOS
movwf  CVRMSOS_1
call   leer_un_byte
movwf  CVRMSOS_0

call    leer_un_byte    ;leer AAPGAIN
movwf  AAPGAIN1
call   leer_un_byte
movwf  AAPGAIN0

call    leer_un_byte    ;leer BAPGAIN
movwf  BAPGAIN1
call   leer_un_byte
movwf  BAPGAIN0

call    leer_un_byte    ;leer CAPGAIN
movwf  CAPGAIN1
call   leer_un_byte
movwf  CAPGAIN0

call    leer_un_byte    ;leer AVGAIN
movwf  AVGAIN1
call   leer_un_byte
movwf  AVGAIN0

call    leer_un_byte    ;leer BVGAIN
movwf  BVGAIN1
call   leer_un_byte
movwf  BVGAIN0

call    leer_un_byte    ;leer CVGAIN
movwf  CVGAIN1
call   leer_un_byte
movwf  CVGAIN0

call    leer_un_byte    ;leer GUARDAR_REG, este rig. es usado para saber si el usuario
movwf  GUARDAR_REG    ;desea guardar estos reg. Resididos desde la PC.
return

```

```

;-----
;
; Rutina para recibir un byte de labview (labview envia cada digito en codigo ASCII)
; argumentos: el byte unido queda en W
;
;-----

```

```

leer_un_byte
espere1_1    ;recibir los 4 bits mas significativos del byte, en ASCII
             btfss   PIR1,RCIF
             goto   espere1_1
             call   Rx_dato_sci    ;el dato que da en W
             movwf  INSD2

espere1_2    ;resivir los 4 bits menos significativos del byte, en ASCII
             btfss   PIR1,RCIF
             goto   espere1_2
             call   Rx_dato_sci    ;el dato que da en W
             movwf  INSD1

             call   ascii_to_bit_msb ;concatenar INSD2 y INSD1 en un solo byte
             page_1
             call   ascii_to_bit_lsb
             page_0
             movf   INSD2,W
             iorwf  INSD1,W

return

```

```

-----
;
; Rutina que pasa un dato en ASCII a binario y luego pone los 4 bits
; menos significativos en los 4 mas significativos.
; argumentos: INSD2
; dato de salida: INSD2
;
-----

```

ascii_to_bit_msb

```

page_1
movlw 0x30 ;es el dato ASCII un cero?
subwf INSD2,W
btfsc STATUS,Z
goto es_0 ;si

movlw 0x31 ;es el dato ASCII un 1?
subwf INSD2,W
btfsc STATUS,Z
goto es_1 ;si

movlw 0x32 ;es el dato ASCII un 2?
subwf INSD2,W
btfsc STATUS,Z
goto es_2 ;si

movlw 0x33 ;es el dato ASCII un 3?
subwf INSD2,W
btfsc STATUS,Z
goto es_3 ;si

movlw 0x34 ;es el dato ASCII un 4?
subwf INSD2,W
btfsc STATUS,Z
goto es_4 ;si

movlw 0x35 ;es el dato ASCII un 5?
subwf INSD2,W
btfsc STATUS,Z
goto es_5 ;si

movlw 0x36 ;es el dato ASCII un 6?
subwf INSD2,W
btfsc STATUS,Z
goto es_6 ;si

movlw 0x37 ;es el dato ASCII un 7?
subwf INSD2,W
btfsc STATUS,Z
goto es_7 ;si

movlw 0x38 ;es el dato ASCII un 8?
subwf INSD2,W
btfsc STATUS,Z
goto es_8 ;si

movlw 0x39 ;es el dato ASCII un 9?
subwf INSD2,W
btfsc STATUS,Z

```

```

goto    es_9            ;si

movlw   0x41            ;es el dato ASCII un 0xa?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_a            ;si

movlw   0x42            ;es el dato ASCII un 0xb?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_b            ;si

movlw   0x43            ;es el dato ASCII un 0xc?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_c            ;si

movlw   0x44            ;es el dato ASCII un 0xd?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_d            ;si

movlw   0x45            ;es el dato ASCII un 0xe?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_e            ;si

movlw   0x46            ;es el dato ASCII un 0xf?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_f            ;si

movlw   0x61            ;es el dato ASCII un 0xa?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_a            ;si

movlw   0x62            ;es el dato ASCII un 0xb?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_b            ;si

movlw   0x63            ;es el dato ASCII un 0xc?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_c            ;si

movlw   0x64            ;es el dato ASCII un 0xd?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_d            ;si

movlw   0x65            ;es el dato ASCII un 0xe?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_e            ;si

movlw   0x66            ;es el dato ASCII un 0xf?
subwf   INSD2,W
btfsc   STATUS,Z
goto    es_f            ;si

es_0    clrf           INSD2
        return

```

```

es_1      movlw 0x10
          movwf INSD2
          return

es_2      movlw 0x20
          movwf INSD2
          return

es_3      movlw 0x30
          movwf INSD2
          return

es_4      movlw 0x40
          movwf INSD2
          return

es_5      movlw 0x50
          movwf INSD2
          return

es_6      movlw 0x60
          movwf INSD2
          return

es_7      movlw 0x70
          movwf INSD2
          return

es_8      movlw 0x80
          movwf INSD2
          return

es_9      movlw 0x90
          movwf INSD2
          return

es_a      movlw 0xa0
          movwf INSD2
          return

es_b      movlw 0xb0
          movwf INSD2
          return

es_c      movlw 0xc0
          movwf INSD2
          return

es_d      movlw 0xd0
          movwf INSD2
          return

es_e      movlw 0xe0
          movwf INSD2
          return

es_f      movlw 0xf0
          movwf INSD2
          return

```

```

;-----
;
; Rutina que pasa un dato en ASCII a binario
; argumentos: INSD1
; dato de salida: INSD1
;
;-----
ascii_to_bit_lsb

```

```

movlw 0x30 ;es el dato ASCII un 0?
subwf INSD1,W
btfsc STATUS,Z
goto es_x0 ;si

movlw 0x31 ;es el dato ASCII un 1?
subwf INSD1,W
btfsc STATUS,Z
goto es_x1 ;si

movlw 0x32 ;es el dato ASCII un 2?
subwf INSD1,W
btfsc STATUS,Z
goto es_x2 ;si

movlw 0x33 ;es el dato ASCII un 3?
subwf INSD1,W
btfsc STATUS,Z
goto es_x3 ;si

movlw 0x34 ;es el dato ASCII un 4?
subwf INSD1,W
btfsc STATUS,Z
goto es_x4 ;si

movlw 0x35 ;es el dato ASCII un 5?
subwf INSD1,W
btfsc STATUS,Z
goto es_x5 ;si

movlw 0x36 ;es el dato ASCII un 6?
subwf INSD1,W
btfsc STATUS,Z
goto es_x6 ;si

movlw 0x37 ;es el dato ASCII un 7?
subwf INSD1,W
btfsc STATUS,Z
goto es_x7 ;si

movlw 0x38 ;es el dato ASCII un 8?
subwf INSD1,W
btfsc STATUS,Z
goto es_x8 ;si

movlw 0x39 ;es el dato ASCII un 9?
subwf INSD1,W
btfsc STATUS,Z
goto es_x9 ;si

movlw 0x41 ;es el dato ASCII un 0xa?
subwf INSD1,W
btfsc STATUS,Z
goto es_xa ;si

movlw 0x42 ;es el dato ASCII un 0xb?
subwf INSD1,W
btfsc STATUS,Z
goto es_xb ;si

movlw 0x43 ;es el dato ASCII un 0xc?
subwf INSD1,W
btfsc STATUS,Z
goto es_xc ;si

movlw 0x44 ;es el dato ASCII un 0xd?

```



```

subwf INSD1,W
btfsc STATUS,Z
goto es_xd ;si

movlw 0x45 ;es el dato ASCII un 0xe?
subwf INSD1,W
btfsc STATUS,Z
goto es_xe ;si

movlw 0x46 ;es el dato ASCII un 0xf?
subwf INSD1,W
btfsc STATUS,Z
goto es_xf ;si

movlw 0x61 ;es el dato ASCII un 0xa?
subwf INSD1,W
btfsc STATUS,Z
goto es_xa ;si

movlw 0x62 ;es el dato ASCII un 0xb?
subwf INSD1,W
btfsc STATUS,Z
goto es_xb ;si

movlw 0x63 ;es el dato ASCII un 0xc?
subwf INSD1,W
btfsc STATUS,Z
goto es_xc ;si

movlw 0x64 ;es el dato ASCII un 0xd?
subwf INSD1,W
btfsc STATUS,Z
goto es_xd ;si

movlw 0x65 ;es el dato ASCII un 0xe?
subwf INSD1,W
btfsc STATUS,Z
goto es_xe ;si

movlw 0x66 ;es el dato ASCII un 0xf?
subwf INSD1,W
btfsc STATUS,Z
goto es_xf ;si

es_x0 clrf INSD1
return

es_x1 movlw 0x01
movwf INSD1
return

es_x2 movlw 0x02
movwf INSD1
return

es_x3 movlw 0x03
movwf INSD1
return

es_x4 movlw 0x04
movwf INSD1
return

es_x5 movlw 0x05
movwf INSD1
return

```

```

es_x6      movlw  0x06
            movwf  INSD1
            return

es_x7      movlw  0x07
            movwf  INSD1
            return

es_x8      movlw  0x08
            movwf  INSD1
            return

es_x9      movlw  0x09
            movwf  INSD1
            return

es_xa      movlw  0x0a
            movwf  INSD1
            return

es_xb      movlw  0x0b
            movwf  INSD1
            return

es_xc      movlw  0x0c
            movwf  INSD1
            return

es_xd      movlw  0x0d
            movwf  INSD1
            return

es_xe      movlw  0x0e
            movwf  INSD1
            return

es_xf      movlw  0x0f
            movwf  INSD1
            return

```

```

;-----
;
;
;   Rutina que lee un byte de la memoria EEPROM del microcontrolador
;   argumentos: - direccion del dato a leer en ADDR ( en el banco 2 )
;   dato de salida: - el dato leido que da en W
;
;-----

```

```

LEER_DATO_EEPROM
    bsf    STATUS, RP1    ;banco 2 o 3
    bcf    STATUS, RP0    ;banco 2

    movf   ADDR,W        ;direccion a leer
    movwf  EEADR
    bsf    STATUS, RP0    ;banco 3
    bcf    EECON1,EEPGD   ;seleccionamos la memoria de datos para leer
    bsf    EECON1,RD      ;iniciamos la lectura
    btfsc  EECON1, RD     ;esperemos a que la lectura se realice
    goto   $-1
    bcf    STATUS, RP0    ;banco 2
    movf   EEDATA,W      ;ponemos el dato leido en W

    bcf    STATUS, RP1    ;banco 0

    return

```

```

;-----
;
;
;   Rutina para leer los registros de conf. guardados en la EEPROM
;   argumentos: no hay
;
;-----

```

```

LEER_REG_CONF_EEPROM
    bsf     STATUS, RP1      ;banco 2 o 3
    bcf     STATUS, RP0      ;banco 2

;leer LINCYC
    movlw   0x00
    movwf  ADDR              ;direccion 0x00
    call   LEER_DATO_EEPROM  ;el dato queda en W
    movwf  LINCYC1
    bsf    STATUS, RP1        ;banco 2
    incf   ADDR,F            ;direccion 0x01
    call   LEER_DATO_EEPROM  ;el dato queda en W
    movwf  LINCYC0
    bsf    STATUS, RP1        ;banco 2

;leer AWG
    incf   ADDR,F            ;direccion 0x02
    call   LEER_DATO_EEPROM
    movwf  AWG1
    bsf    STATUS, RP1        ;banco 2
    incf   ADDR,F            ;direccion 0x03
    call   LEER_DATO_EEPROM
    movwf  AWG0
    bsf    STATUS, RP1        ;banco 2

;leer BWG
    incf   ADDR,F            ;direccion 0x04
    call   LEER_DATO_EEPROM
    movwf  BWG1
    bsf    STATUS, RP1        ;banco 2
    incf   ADDR,F            ;direccion 0x05
    call   LEER_DATO_EEPROM
    movwf  BWG0
    bsf    STATUS, RP1        ;banco 2

;leer CWG
    incf   ADDR,F            ;direccion 0x06
    call   LEER_DATO_EEPROM
    movwf  CWG1
    bsf    STATUS, RP1        ;banco 2
    incf   ADDR,F            ;direccion 0x07
    call   LEER_DATO_EEPROM
    movwf  CWG0
    bsf    STATUS, RP1        ;banco 2

;leer AVAG
    incf   ADDR,F            ;direccion 0x08
    call   LEER_DATO_EEPROM
    movwf  AVAG1
    bsf    STATUS, RP1        ;banco 2
    incf   ADDR,F            ;direccion 0x09
    call   LEER_DATO_EEPROM
    movwf  AVAG0
    bsf    STATUS, RP1        ;banco 2

;leer BVAG
    incf   ADDR,F            ;direccion 0x0A
    call   LEER_DATO_EEPROM
    movwf  BVAG1
    bsf    STATUS, RP1        ;banco 2
    incf   ADDR,F            ;direccion 0x0B
    call   LEER_DATO_EEPROM
    movwf  BVAG0
    bsf    STATUS, RP1        ;banco 2

;leer CVAG

```

```

incf    ADDR,F          ;direccion 0x0C
call    LEER_DATO_EEPROM
movwf   CVAG1
bsf     STATUS, RP1     ;banco 2
incf    ADDR,F          ;direccion 0x0D
call    LEER_DATO_EEPROM
movwf   CVAG0
bsf     STATUS, RP1     ;banco 2

;leer APHCAL
incf    ADDR,F          ;direccion 0x0E
call    LEER_DATO_EEPROM
movwf   APHCAL0
bsf     STATUS, RP1     ;banco 2

;leer BPHCAL
incf    ADDR,F          ;direccion 0x0F
call    LEER_DATO_EEPROM
movwf   BPHCAL0
bsf     STATUS, RP1     ;banco 2

;leer CPHCAL
incf    ADDR,F          ;direccion 0x10
call    LEER_DATO_EEPROM
movwf   CPHCAL0
bsf     STATUS, RP1     ;banco 2

;leer AAPOS
incf    ADDR,F          ;direccion 0x11
call    LEER_DATO_EEPROM
movwf   AAPOS1
bsf     STATUS, RP1     ;banco 2
incf    ADDR,F          ;direccion 0x12
call    LEER_DATO_EEPROM
movwf   AAPOS0
bsf     STATUS, RP1     ;banco 2

;leer BAPOS
incf    ADDR,F          ;direccion 0x13
call    LEER_DATO_EEPROM
movwf   BAPOS1
bsf     STATUS, RP1     ;banco 2
incf    ADDR,F          ;direccion 0x14
call    LEER_DATO_EEPROM
movwf   BAPOS0
bsf     STATUS, RP1     ;banco 2

;leer CAPOS
incf    ADDR,F          ;direccion 0x15
call    LEER_DATO_EEPROM
movwf   CAPOS1
bsf     STATUS, RP1     ;banco 2
incf    ADDR,F          ;direccion 0x16
call    LEER_DATO_EEPROM
movwf   CAPOS0
bsf     STATUS, RP1     ;banco 2

;leer CFNUM
incf    ADDR,F          ;direccion 0x17
call    LEER_DATO_EEPROM
movwf   CFNUM1
bsf     STATUS, RP1     ;banco 2
incf    ADDR,F          ;direccion 0x18
call    LEER_DATO_EEPROM
movwf   CFNUM0
bsf     STATUS, RP1     ;banco 2

```

```

;leer CFDEN
incf   ADDR,F           ;direccion 0x19
call   LEER_DATO_EEPROM
movwf  CFDEN1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x1A
call   LEER_DATO_EEPROM
movwf  CFDEN0
bsf    STATUS, RP1     ;banco 2

;leer AIRMSOS
incf   ADDR,F           ;direccion 0x1B
call   LEER_DATO_EEPROM
movwf  AIRMSOS_1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x1C
call   LEER_DATO_EEPROM
movwf  AIRMSOS_0
bsf    STATUS, RP1     ;banco 2

;leer BIRMSOS
incf   ADDR,F           ;direccion 0x1D
call   LEER_DATO_EEPROM
movwf  BIRMSOS_1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x1F
call   LEER_DATO_EEPROM
movwf  BIRMSOS_0
bsf    STATUS, RP1     ;banco 2

;leer CIRMSOS
incf   ADDR,F           ;direccion 0x20
call   LEER_DATO_EEPROM
movwf  CIRMSOS_1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x21
call   LEER_DATO_EEPROM
movwf  CIRMSOS_0
bsf    STATUS, RP1     ;banco 2

;leer AVRMSOS
incf   ADDR,F           ;direccion 0x22
call   LEER_DATO_EEPROM
movwf  AVRMSOS_1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x23
call   LEER_DATO_EEPROM
movwf  AVRMSOS_0
bsf    STATUS, RP1     ;banco 2

;leer BVRMSOS
incf   ADDR,F           ;direccion 0x24
call   LEER_DATO_EEPROM
movwf  BVRMSOS_1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x25
call   LEER_DATO_EEPROM
movwf  BVRMSOS_0
bsf    STATUS, RP1     ;banco 2

;leer CVRMSOS
incf   ADDR,F           ;direccion 0x26
call   LEER_DATO_EEPROM
movwf  CVRMSOS_1
bsf    STATUS, RP1     ;banco 2
incf   ADDR,F           ;direccion 0x27
call   LEER_DATO_EEPROM

```

```

movwf  CVRMSOS_0
bsf    STATUS, RP1                ;banco 2

;leer AAPGAIN
incf   ADDR,F                    ;direccion 0x28
call   LEER_DATO_EEPROM
movwf  AAPGAIN1
bsf    STATUS, RP1                ;banco 2
incf   ADDR,F                    ;direccion 0x29
call   LEER_DATO_EEPROM
movwf  AAPGAIN0
bsf    STATUS, RP1                ;banco 2

;leer BAPGAIN
incf   ADDR,F                    ;direccion 0x2A
call   LEER_DATO_EEPROM
movwf  BAPGAIN1
bsf    STATUS, RP1                ;banco 2
incf   ADDR,F                    ;direccion 0x2B
call   LEER_DATO_EEPROM
movwf  BAPGAIN0
bsf    STATUS, RP1                ;banco 2

;leer CAPGAIN
incf   ADDR,F                    ;direccion 0x2C
call   LEER_DATO_EEPROM
movwf  CAPGAIN1
bsf    STATUS, RP1                ;banco 2
incf   ADDR,F                    ;direccion 0x2D
call   LEER_DATO_EEPROM
movwf  CAPGAIN0
bsf    STATUS, RP1                ;banco 2

;leer AVGAIN
incf   ADDR,F                    ;direccion 0x2E
call   LEER_DATO_EEPROM
movwf  AVGAIN1
bsf    STATUS, RP1                ;banco 2
incf   ADDR,F                    ;direccion 0x2F
call   LEER_DATO_EEPROM
movwf  AVGAIN0
bsf    STATUS, RP1                ;banco 2

;leer BVGAIN
incf   ADDR,F                    ;direccion 0x30
call   LEER_DATO_EEPROM
movwf  BVGAIN1
bsf    STATUS, RP1                ;banco 2
incf   ADDR,F                    ;direccion 0x31
call   LEER_DATO_EEPROM
movwf  BVGAIN0
bsf    STATUS, RP1                ;banco 2

;leer CVGAIN
incf   ADDR,F                    ;direccion 0x32
call   LEER_DATO_EEPROM
movwf  CVGAIN1
bsf    STATUS, RP1                ;banco 2
incf   ADDR,F                    ;direccion 0x33
call   LEER_DATO_EEPROM
movwf  CVGAIN0
;regresa con banco 0

return

```

-----;

```

;
; Rutina para escribe datos a la memoria EEPROM del microcontrolador
; argumentos: - ponemos la direccion del dato a escribir en ADDR ( en el banco 2 )
;
-----
ESCRIVIR_DATOS_EEPROM
    bsf     STATUS, RP1
    bcf     STATUS, RP0      ;banco 2

linc1  movlw  0x00           ;LINCYC, byte msb
        movwf ADDR
        bcf     STATUS, RP1      ;banco 0
        movf   LINCYC1,W
        bsf     STATUS, RP1      ;banco 2
        movwf  VALOR
        call   ENVIAR_REG_CONF_ADE ;escribir el byte

        call   LEER_DATO         ;leemos el byte escrito en la EEPROM para verificar
        subwf  VALOR,W           ;que se guardo correctamente
        btfss  STATUS,Z

        goto   linc1

linc2  movlw  0x01           ;LINCYC, byte lsb
        movwf  ADDR
        bcf     STATUS, RP1      ;banco 0
        movf   LINCYC0,W
        bsf     STATUS, RP1      ;banco 2
        movwf  VALOR
        call   ENVIAR_REG_CONF_ADE ;escribir el byte

        call   LEER_DATO         ;leemos el byte escrito en la EEPROM para verificar
        subwf  VALOR,W           ;que se guardo correctamente
        btfss  STATUS,Z
        goto   linc2

awg11  movlw  0x02           ;AWG, byte msb
        movwf  ADDR
        bcf     STATUS, RP1      ;banco 0
        movf   AWG1,W
        bsf     STATUS, RP1      ;banco 2
        movwf  VALOR
        call   ENVIAR_REG_CONF_ADE

        call   LEER_DATO         ;leemos el byte escrito en la EEPROM para verificar
        subwf  VALOR,W           ;que se guardo correctamente
        btfss  STATUS,Z
        goto   awg11

awg12  movlw  0x03           ;AWG
        movwf  ADDR
        bcf     STATUS, RP1      ;banco 0
        movf   AWG0,W
        bsf     STATUS, RP1      ;banco 2
        movwf  VALOR
        call   ENVIAR_REG_CONF_ADE

        call   LEER_DATO         ;verificamos que se guardo el dato
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   awg12

bwg11  movlw  0x04           ;BWG
        movwf  ADDR
        bcf     STATUS, RP1      ;banco 0
        movf   BWG1,W
        bsf     STATUS, RP1      ;banco 2

```

```

movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO ;verificamos que se guardo el dato
subwf VALOR,W
btfss STATUS,Z
goto bwg11

bwg12 movlw 0x05 ;BWG
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf BWG0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO ;verificamos que se guardo el dato
subwf VALOR,W
btfss STATUS,Z
goto bwg12

cwg11 movlw 0x06 ;CWG
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf CWG1,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO ;verificamos que se guardo el dato
subwf VALOR,W
btfss STATUS,Z
goto cwg11

cwg12 movlw 0x07 ;CWG
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf CWG0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO ;verificamos que se guardo el dato
subwf VALOR,W
btfss STATUS,Z
goto cwg12

avag11 movlw 0x08 ;AVAG
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf AVAG1,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO ;verificamos que se guardo el dato
subwf VALOR,W
btfss STATUS,Z
goto avag11

avag12 movlw 0x09 ;AVAG
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf AVAG0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

```



```

        call    LEER_DATO      ;verificamos que se guardo el dato
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   avag12

bvag11  movlw  0x0a           ;BVAG
        movwf  ADDR
        bcf   STATUS, RP1     ;banco 0
        movf  BVAG1,W
        bsf   STATUS, RP1     ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE
        call  LEER_DATO
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   bvag11

bvag12  movlw  0x0b           ;BVAG
        movwf  ADDR
        bcf   STATUS, RP1     ;banco 0
        movf  BVAG0,W
        bsf   STATUS, RP1     ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE
        call  LEER_DATO
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   bvag12

cvag11  movlw  0x0c           ;CVAG
        movwf  ADDR
        bcf   STATUS, RP1     ;banco 0
        movf  CVAG1,W
        bsf   STATUS, RP1     ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE
        call  LEER_DATO
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   cvag11

cvag12  movlw  0x0d           ;CVAG
        movwf  ADDR
        bcf   STATUS, RP1     ;banco 0
        movf  CVAG0,W
        bsf   STATUS, RP1     ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE
        call  LEER_DATO
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   cvag12

aphcal11  movlw  0x0e           ;APHCAL, este reg. es de un byte
        movwf  ADDR
        bcf   STATUS, RP1     ;banco 0
        movf  APHCAL0,W
        bsf   STATUS, RP1     ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE

        call  LEER_DATO
        subwf  VALOR,W
        btfss  STATUS,Z
        goto   aphcal11

```

```

bphcal11
    movlw 0x0f          ;BPHCAL, este reg. es de un byte
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf BPHCAL0,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto bphcal11

cphcal11
    movlw 0x10          ;CPHCAL, este reg. es de un byte
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf CPHCAL0,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto cphcal11

aapos11
    movlw 0x11          ;AAPOS, byte msb
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf AAPOS1,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto aapos11

aapos12
    movlw 0x12          ;AAPOS, byte lsb
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf AAPOS0,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto aapos12

bapos11
    movlw 0x13          ;BAPOS
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf BAPOS1,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO

```

```

        subwf  VALOR,W
        btfss STATUS,Z
        goto  bapos11

bapos12
        movlw  0x14          ;BAPOS
        movwf  ADDR
        bcf   STATUS, RP1    ;banco 0
        movf  BAPOS0,W
        bsf   STATUS, RP1    ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE

        call  LEER_DATO
        subwf  VALOR,W
        btfss STATUS,Z
        goto  bapos12

capos11
        movlw  0x15          ;CAPOS
        movwf  ADDR
        bcf   STATUS, RP1    ;banco 0
        movf  CAPOS1,W
        bsf   STATUS, RP1    ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE

        call  LEER_DATO
        subwf  VALOR,W
        btfss STATUS,Z
        goto  capos11

capos12
        movlw  0x16          ;CAPOS
        movwf  ADDR
        bcf   STATUS, RP1    ;banco 0
        movf  CAPOS0,W
        bsf   STATUS, RP1    ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE

        call  LEER_DATO
        subwf  VALOR,W
        btfss STATUS,Z
        goto  capos12

cfnun11
        movlw  0x17          ;CFNUM
        movwf  ADDR
        bcf   STATUS, RP1    ;banco 0
        movf  CFNUM1,W
        bsf   STATUS, RP1    ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE
        call  LEER_DATO
        subwf  VALOR,W
        btfss STATUS,Z
        goto  cfnun11

cfnun12
        movlw  0x18          ;CFNUM
        movwf  ADDR
        bcf   STATUS, RP1    ;banco 0
        movf  CFNUM0,W
        bsf   STATUS, RP1    ;banco 2
        movwf  VALOR
        call  ENVIAR_REG_CONF_ADE

```

```

call    LEER_DATO
subwf   VALOR,W
btfss   STATUS,Z
goto    cfnun12

cfden11
movlw   0x19           ;CFDEN
movwf   ADDR
bcf     STATUS, RP1   ;banco 0
movf    CFDEN1,W
bsf     STATUS, RP1   ;banco 2
movwf   VALOR
call    ENVIAR_REG_CONF_ADE

call    LEER_DATO
subwf   VALOR,W
btfss   STATUS,Z
goto    cfden11

cfden12
movlw   0x1a           ;CFDEN
movwf   ADDR
bcf     STATUS, RP1   ;banco 0
movf    CFDEN0,W
bsf     STATUS, RP1   ;banco 2
movwf   VALOR
call    ENVIAR_REG_CONF_ADE

call    LEER_DATO
subwf   VALOR,W
btfss   STATUS,Z
goto    cfden12

airmsos11
movlw   0x1b           ;AIrmsOS
movwf   ADDR
bcf     STATUS, RP1   ;banco 0
movf    AIRMSOS_1,W
bsf     STATUS, RP1   ;banco 2
movwf   VALOR
call    ENVIAR_REG_CONF_ADE

call    LEER_DATO
subwf   VALOR,W
btfss   STATUS,Z
goto    airmsos11

airmsos12
movlw   0x1c           ;AIrmsOS
movwf   ADDR
bcf     STATUS, RP1   ;banco 0
movf    AIRMSOS_0,W
bsf     STATUS, RP1   ;banco 2
movwf   VALOR
call    ENVIAR_REG_CONF_ADE

call    LEER_DATO
subwf   VALOR,W
btfss   STATUS,Z
goto    airmsos12

birmsos11
movlw   0x1d           ;BIrmsOS
movwf   ADDR
bcf     STATUS, RP1   ;banco 0
movf    BIRMSOS_1,W

```

```

    bsf     STATUS, RP1      ;banco 2
    movwf  VALOR
    call   ENVIAR_REG_CONF_ADE

    call   LEER_DATO
    subwf  VALOR,W
    btfss  STATUS,Z
    goto   birmsos11

birmsos12

    movlw  0x1e              ;BirmsOS
    movwf  ADDR
    bcf    STATUS, RP1      ;banco 0
    movf   BIRMSOS_0,W
    bsf    STATUS, RP1      ;banco 2
    movwf  VALOR
    call   ENVIAR_REG_CONF_ADE

    call   LEER_DATO
    subwf  VALOR,W
    btfss  STATUS,Z
    goto   birmsos12

cirmsos11

    movlw  0x1f              ;CirmsOS
    movwf  ADDR
    bcf    STATUS, RP1      ;banco 0
    movf   CIRMSOS_1,W
    bsf    STATUS, RP1      ;banco 2
    movwf  VALOR
    call   ENVIAR_REG_CONF_ADE

    call   LEER_DATO
    subwf  VALOR,W
    btfss  STATUS,Z
    goto   cirmsos11

cirmsos12

    movlw  0x20              ;CirmsOS
    movwf  ADDR
    bcf    STATUS, RP1      ;banco 0
    movf   CIRMSOS_0,W
    bsf    STATUS, RP1      ;banco 2
    movwf  VALOR
    call   ENVIAR_REG_CONF_ADE

    call   LEER_DATO
    subwf  VALOR,W
    btfss  STATUS,Z
    goto   cirmsos12

avrmsos11

    movlw  0x21              ;AVrmsOS
    movwf  ADDR
    bcf    STATUS, RP1      ;banco 0
    movf   AVRMSOS_1,W
    bsf    STATUS, RP1      ;banco 2
    movwf  VALOR
    call   ENVIAR_REG_CONF_ADE

    call   LEER_DATO
    subwf  VALOR,W
    btfss  STATUS,Z
    goto   avrmsos11

avrmsos12

    movlw  0x22              ;AVrmsOS

```

```

movwf ADDR
bcf STATUS, RP1 ;banco 0
movf AVRMSOS_0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto avrmsos12

bvrmsos11
movlw 0x23 ;BVrmsOS
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf BVRMSOS_1,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto bvrmsos11

bvrmsos12
movlw 0x24 ;BVrmsOS
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf BVRMSOS_0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto bvrmsos12

cvrmsos11
movlw 0x25 ;CVrmsOS
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf CVRMSOS_1,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto cvrmsos11

cvrmsos12
movlw 0x26 ;CVrmsOS
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf BVRMSOS_0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto cvrmsos12

```

```

aapgain1
    movlw 0x27          ;AAPGAIN
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf AAPGAIN1,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto aapgain1

aapgain12
    movlw 0x28          ;AAPGAIN
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf AAPGAIN0,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto aapgain12

bapgain11
    movlw 0x29          ;BAPGAIN
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf BAPGAIN1,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto bapgain11

bapgain12
    movlw 0x2a          ;BAPGAIN
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf BAPGAIN0,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO
    subwf VALOR,W
    btfss STATUS,Z
    goto bapgain12

capgain11
    movlw 0x2b          ;CAPGAIN
    movwf ADDR
    bcf STATUS, RP1    ;banco 0
    movf CAPGAIN1,W
    bsf STATUS, RP1    ;banco 2
    movwf VALOR
    call ENVIAR_REG_CONF_ADE

    call LEER_DATO

```

```

subwf VALOR,W
btfss STATUS,Z
goto capgain11

capgain12
movlw 0x2c ;CAPGAIN
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf CAPGAIN0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto capgain12

avgain11
movlw 0x2d ;AVGAIN
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf AVGAIN1,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto avgain11

avgain12
movlw 0x2e ;AVGAIN
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf AVGAIN0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto avgain12

bvgain11
movlw 0x2f ;BVGAIN
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf BVGAIN1,W
bsf STATUS, RP1 ;banco 2
movwf VALOR
call ENVIAR_REG_CONF_ADE

call LEER_DATO
subwf VALOR,W
btfss STATUS,Z
goto bvgain11

bvgain12
movlw 0x30 ;BVGAIN
movwf ADDR
bcf STATUS, RP1 ;banco 0
movf BVGAIN0,W
bsf STATUS, RP1 ;banco 2
movwf VALOR

```



```

call    ENVIAR_REG_CONF_ADE

call    LEER_DATO
subwf  VALOR,W
btfss  STATUS,Z
goto   bvgain12

cvgain11
movlw  0x31          ;CVGAIN
movwf  ADDR
bcf    STATUS, RP1  ;banco 0
movf   CVGAIN1,W
bsf    STATUS, RP1  ;banco 2
movwf  VALOR
call   ENVIAR_REG_CONF_ADE

call   LEER_DATO
subwf  VALOR,W
btfss  STATUS,Z
goto   cvgain11

cvgain12
movlw  0x32          ;CVGAIN
movwf  ADDR
bcf    STATUS, RP1  ;banco 0
movf   CVGAIN0,W
bsf    STATUS, RP1  ;banco 2
movwf  VALOR
call   ENVIAR_REG_CONF_ADE

call   LEER_DATO
subwf  VALOR,W
btfss  STATUS,Z
goto   cvgain12

bcf    STATUS, RP1
bcf    STATUS, RP0  ;banco 0

;retornamos en el banco 0
return

;-----
;
; Rutina que lee la memoria EEPROM del microcontrolador
; argumentos: - numero de bits a escribir en LENG_DATOS
;             - direccion en don de se escribira SPI_COMANDO
; salida: -el dato leido queda en W
;-----
LEER_DATO
bsf    STATUS, RP1
bcf    STATUS, RP0  ;banco 2
movf   ADDR,W      ;direccion a leer
movwf  EEADR
bsf    STATUS, RP0  ;banco 3
bcf    EECON1,EEPGD ;seleccionamos la memoria de datos
bsf    EECON1,RD    ;iniciamos la lectura
btfsc  EECON1, RD   ;esperemos que la lectura termine
goto   $-1
bcf    STATUS, RP0  ;banco 2
movf   EEDATA,W    ;ponemos el dato leido en W

;retornamos en el banco 2
return

;-----
;

```

```

; Rutina para enviar los reg. de conf. del ADE7754 a la eeprom del pic16f877
; argumentos: - el numero de bits a escribir en LENG_DATOS
;             - la direccion en donde se escribira esta en SPI_COMANDO
;-----
ENVIAR_REG_CONF_ADE
    bsf    STATUS, RP1
    bsf    STATUS, RP0      ;banco 3
    btfsc  EECON1, WR      ;esperemos si hay un proceso de escritura en progreso
    goto   $-1
    bcf    EECON1,WREN     ;desactivamos la funcion de escritura.
    bcf    STATUS, RP0     ;banco 2
    movf   ADDR,W
    movwf  EEADR           ;cargamos la direccion en la escribiremos
    movf   VALOR,W
    movwf  EEDATA
    bsf    STATUS,RP0      ;banco 3
    bcf    EECON1, EEPGD   ;seleccionamos la memoria de datos
    bsf    EECON1, WREN    ;activamos la funcion de escritura
    movlw  0x55
    movwf  EECON2         ;escribimos 0x55 en EECON2
    movlw  0xaa
    movwf  EECON2         ;escribimos 0xaa en EECON2
    bsf    EECON1,WR      ;iniciamos la operacion de escritura.
    bsf    STATUS, RP0     ;banco 3
    btfsc  EECON1, WR      ;esperemos a que el proceso de escritura termine
    goto   $-1
    bcf    EECON1,WREN     ;desactivamos la funcion de escritura.
    bcf    STATUS, RP0     ;banco 2
    return

```

```

;-----
;
; Rutina para leer la hora actual del reloj y enviarla al PC
; argumentos: no hay
;-----
TIEMPO_DEL_RELOJ

```

```

;retardo para la com. spi
retardo_x .29 ;call macro retardo = 2+(1+2)*29+1=90 ciclo de reloj

call    Rx_dato_spi      ;el Bye MSB que da en W
movwf  TIEMP4
page_0
call    Tx_Dato_sci      ;enviar el byte a la PC
page_1

call    Rx_dato_spi      ;el Bye Medio que da en W
movwf  TIEMP3
page_0
call    Tx_Dato_sci      ;enviar el byte a la PC
page_1

call    Rx_dato_spi      ;el Bye Medio que da en W
movwf  TIEMP2
page_0
call    Tx_Dato_sci      ;enviar el byte a la PC
page_1

call    Rx_dato_spi      ;el Bye LSB que da en W
movwf  TIEMP1
page_0
call    Tx_Dato_sci      ;enviar el byte a la PC
page_1

return

```

```

;-----
;
;
; Rutina para Recibir un byte por via SPI
; argumentos: no hay!!
; salida. el byte leido que da en W
;-----

```

```

Rx_dato_spi
    Bsf     STATUS,RP0      ; banco 1
    Bcf     SSPSTAT,0      ; limpiamos la bandera del buffer spi
    Bcf     STATUS,RP0      ; banco 0
    Movlw  0x05             ;enviamos un dato cualquiera
    Movwf  SSPBUF
    Bsf     STATUS,RP0      ; banco 1

```

```

Tx_Dato_Wait2
    Btfss  SSPSTAT,0      ; dato enviado?
    goto  Tx_Dato_Wait2  ; no

    bcf     STATUS,RP0      ; banco 0
    movf   SSPBUF, W      ;el byte recibido es puesto en W
    return

```

```

;-----
;
;
; Rutina para transmitir un byte por via SPI
; argumentos: el dato a transmitir esta en W
;-----

```

```

Tx_dato_spi
    Bsf     STATUS,RP0      ; banco 1
    Bcf     SSPSTAT,0      ; limpiamos la bandera del buffer spi
    Bcf     STATUS,RP0      ; banco 0
    Movwf  SSPBUF          ;cargamos el dato a transmitir en el buffer
    Bsf     STATUS,RP0      ; banco 1

```

```

Tx_dato_spi_LOOP:
    Btfss  SSPSTAT,0      ; dato enviado?
    Goto   Tx_dato_spi_LOOP ; no
    bcf   STATUS,RP0      ; banco 0

;retornamos en el banco 0
return

```

```

;-----
;
;
; Rutina para sumar potencia activa en los acumuladores
; argumentos: registros P_TRIFASICA_0, P_TRIFASICA_1, P_TRIFASICA_3
;
;
; datos de salida: P_PROM_0, P_PROM_1, P_PROM_2
;-----

```

SUMAR_ENERGIA_ACTIVA

```

    Bcf     STATUS, RP1
    bsf     STATUS, RP0      ;banco 1

;fase a
    movf   P_TRIFASICA_0,W ;potencia acumulada = potencia leida +
                           ;potencia acumulada
    addwf  P_PROM_0,F
    call   CARRYCHK
    movf   CARRY,W
    addwf  P_PROM_1,F
    call   CARRYCHK
    movf   CARRY,W

```

```

        addwf    P_PROM_2,F

        movf    P_TRIFASICA_1,W
        addwf   P_PROM_1,F
        call    CARRYCHK
        movf    CARRY,W
        addwf   P_PROM_2,F

        movf    P_TRIFASICA_2,W
        addwf   P_PROM_2,F

        bcf     STATUS, RP0      ;banco 0
        return

CARRYCHK
        clrf    CARRY            ;borramos el registro de ACARREO
        btfsc   STATUS,C         ;ha ocurrido un acarreo?
        incf    CARRY,F         ;si
        return

```

```

;-----
;
; Rutina para sumar potencia aparente en los acumuladores
; argumentos: registros S_TRIFASICA_0, S_TRIFASICA_1, S_TRIFASICA_3
;
; datos de salida: S_PROM_0, S_PROM_1, S_PROM_2
;-----
SUMAR_ENERGIA_APARENTE

```

```

        bcf     STATUS, RP1
        bsf     STATUS, RP0      ;banco 1

        ;fase a
        movf    S_TRIFASICA_0,W ;potencia acumulada = potencia leida + potencia
                                ;acumulada

        addwf   S_PROM_0,F
        call    CARRYCHK
        movf    CARRY,W
        addwf   S_PROM_1,F
        call    CARRYCHK
        movf    CARRY,W
        addwf   S_PROM_2,F

        movf    S_TRIFASICA_1,W
        addwf   S_PROM_1,F
        call    CARRYCHK
        movf    CARRY,W
        addwf   S_PROM_2,F

        movf    S_TRIFASICA_2,W
        addwf   S_PROM_2,F

        bcf     STATUS, RP0      ;banco 0
        return

```

```

;-----

```

```

;
;
; Rutina para sumar potencia reactiva en los acumuladores
; argumentos: registros S_TRIFASICA_0, S_TRIFASICA_1, S_TRIFASICA_3
;
; datos de salida: S_PROM_0, S_PROM_1, S_PROM_2
;

```

```

-----
SUMAR_ENERGIA_REACTIVA

```

```

    bcf     STATUS, RP1
    bsf     STATUS, RP0      ;banco 1

    ;fase a
    movf   Q_TRIFASICA_0,W
    addwf  Q_PROM_0,F
    call   CARRYCHK
    movf   CARRY,W
    addwf  Q_PROM_1,F
    call   CARRYCHK
    movf   CARRY,W
    addwf  Q_PROM_2,F

    movf   Q_TRIFASICA_1,W
    addwf  Q_PROM_1,F
    call   CARRYCHK
    movf   CARRY,W
    addwf  Q_PROM_2,F

    movf   Q_TRIFASICA_2,W
    addwf  Q_PROM_2,F

    bcf     STATUS, RP0      ;banco 0
    return

```

```

-----
;
;
; Rutina que calcula el promedio de la potencia activa
; argumentos: P_PROM_0, P_PROM_1, P_PROM_2
;
; datos de salida: los resg. de entrada modificados.
;

```

```

-----
PROMEDIO_ENERGIA_ACTIVIA

```

```

    bcf     STATUS, RP1
    bsf     STATUS, RP0      ;banco 1

    ;fase a
    movlw  .2                ;numero de veces a rotar el dato (2^2==4 o.k)
    movwf  DIV

divbin_fase_a      ;rotamos la potencia promedio a la derecha 2 veces
    bcf     STATUS,C
    btfsc  P_PROM_2,0x07    ;es positivo o negativo?
    bsf     STATUS,C        ;es negativo
    rrf    P_PROM_2,0x1
    rrf    P_PROM_1,0x1
    rrf    P_PROM_0,0x1
    decfsz DIV,0x01
    goto   divbin_fase_a

    bcf     STATUS, RP0      ;retornamos en el banco 0

    return

```

```

-----
;
;
; Rutina que calcula el promedio de la potencia activa
; argumentos: S_PROM_0, S_PROM_1, S_PROM_2
;

```

```

;      datos de salida: los resg. de entrada modificados.
;
;-----
PROMEDIO_ENERGIA_APARENTE
    bcf     STATUS, RP1
    bsf     STATUS, RP0      ;banco 1

    ;fase a
    movlw  .2      ;numero de veces a rotar el dato (2^2==4 o.k)
    movwf  DIV

divbin2_fase_a
    bcf     STATUS,C
    btfsc  S_PROM_2,0x07    ;es positivo o negativo?
    bsf     STATUS,C      ;es negativo
    rrf     S_PROM_2,0x1
    rrf     S_PROM_1,0x1
    rrf     S_PROM_0,0x1
    decfsz DIV,0x01
    goto   divbin2_fase_a

    bcf     STATUS, RP0      ;retornamos en el banco 0

    return

;-----
;
;      Rutina que calcula el promedio de la potencia activa
;      argumentos: Q_PROM_0, Q_PROM_1, Q_PROM_2
;
;      datos de salida: los resg. de entrada modificados.
;
;-----
PROMEDIO_ENERGIA_REACTIVA
    bcf     STATUS, RP1
    bsf     STATUS, RP0      ;banco 1

    ;fase a
    movlw  .2      ;numero de veces a rotar el dato (2^2==4 o.k)
    movwf  DIV

divbin3_fase_a
    bcf     STATUS,C
    btfsc  Q_PROM_2,0x07    ;es positivo o negativo?
    bsf     STATUS,C      ;es negativo
    rrf     Q_PROM_2,0x1
    rrf     Q_PROM_1,0x1
    rrf     Q_PROM_0,0x1
    decfsz DIV,0x01
    goto   divbin3_fase_a

    bcf     STATUS, RP0      ;banco 0 ;retornamos en el banco 0

    return

;-----
;
;      Rutina que borra los acumuladores de energia para un nuevo proceso
;      (esto se hace cada 12 o 20 min. segun la conf. del medidor)
;      argumentos: registros
;
;-----
borrar_acumuladores_energia

    bcf     STATUS, RP1      ;banco 0 o 1
    bsf     STATUS,RP0      ;Selecciona banco 1

```

```

clrf    P_PROM_0
clrf    P_PROM_1
clrf    P_PROM_2

clrf    S_PROM_0
clrf    S_PROM_1
clrf    S_PROM_2

clrf    Q_PROM_0
clrf    Q_PROM_1
clrf    Q_PROM_2

bcf     STATUS,RP0      ;Selecciona banco 0
return

```

```

;-----
;
; Rutina que guarda en la memoria externa via SPI, la potencia
; activa, aparente y reactiva trifasica y las corrientes y voltajes RMS
; argumentos: no hay
;
;nota: cada registro se guarda comenzando con el MSB y por ultimo el LSB
;-----

```

GUARDAR_DATOS_A_MEMORIA

```

bcf     STATUS, RP1      ;banco 0 o 1
bcf     STATUS, RP0      ;banco 0

set_spi_memoria ;macro que habilita el modulo SPI de la memoria

;guardamos la potencial activa
bsf     STATUS, RP0      ;banco 1
movf    P_PROM_2,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar          ;guardamos lo que esta en el reg. W

bsf     STATUS, RP0      ;banco 1
movf    P_PROM_1,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar          ;guardamos lo que esta en el reg. W

bsf     STATUS, RP0      ;banco 1
movf    P_PROM_0,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar          ;guardamos lo que esta en el reg. W

;guardamos la potencial aparente
bsf     STATUS, RP0      ;banco 1
movf    S_PROM_2,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar          ;guardamos lo que esta en el reg. W

bsf     STATUS, RP0      ;banco 1
movf    S_PROM_1,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar          ;guardamos lo que esta en el reg. W

bsf     STATUS, RP0      ;banco 1
movf    S_PROM_0,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar

;guardamos la potencial reactiva (solo el signo)
bsf     STATUS, RP0      ;banco 1
movf    Q_PROM_2,W       ;dato a guardar
bcf     STATUS, RP0      ;banco 0
call    guardar

```

```

bsf     STATUS, RP0      ;banco 1
movf   Q_PROM_1,W
bcf    STATUS, RP0      ;banco 0
call   guardar

bsf     STATUS, RP0      ;banco 1
movf   Q_PROM_0,W
bcf    STATUS, RP0      ;banco 0
call   guardar

;Guardar los voltajes
bcf    STATUS, RP0      ;banco 0

;fase a
movf   AVRMS2,W
call   guardar

movf   AVRMS1,W
call   guardar

movf   AVRMS0,W
call   guardar

;fase b
movf   BVRMS2,W
call   guardar

movf   BVRMS1,W
call   guardar

movf   BVRMS0,W
call   guardar

;fase c
movf   CVRMS2,W
call   guardar

movf   CVRMS1,W
call   guardar

movf   CVRMS0,W
call   guardar

;Guardar las corrientes

;fase a
movf   AIRMS2,W
call   guardar

movf   AIRMS1,W
call   guardar

movf   AIRMS0,W
call   guardar

;fase b
movf   BIRMS2,W
call   guardar

movf   BIRMS1,W
call   guardar

movf   BIRMS0,W
call   guardar

```



```

;fase c
movf   CIRMS2,W
call   guardar

movf   CIRMS1,W
call   guardar

movf   CIRMS0,W
call   guardar

;guardar el tiempo

movf   TIEMP4,W      ;byte MSB
call   guardar

movf   TIEMP3,W
call   guardar

movf   TIEMP2,W
call   guardar

movf   TIEMP1,W      ;byte LSB
call   guardar

set_spi_ade7754 ;macro que habilita el modulo SPI del ADE7754
return

;-----
;
; Rutina que guarda un byte en la memoria y conserva un apuntador a la
; siguiente localidad de memoria, tambien lleva el conteo de cuantos
; byte hay en la memoria
; argumentos: no hay
;-----
guardar
    bcf     STATUS, RP1    ;banco 0 o 1
    bcf     STATUS, RP0    ;banco 0
    movwf   DATO           ;se guarda el dato a escribir en W

;-----
retardo_x .6      ;call to macro de retardo =3+3*x ciclos de reloj, donde x=6

;ESCRITURA A LA MEMORIA
movlw   b'00000000'     ;comando de escritura es enviado a la memoria
movwf   SSPBUF

call    Tx_Dato_Wait     ;esperamos a que el comando se transmita

;-----
call    direccion        ;enviamos la direccion en donde guardaremos el dato
;-----
retardo_x .6      ;call to macro de retardo=3+3*x ciclos de reloj, donde x=6

movf   DATO,W           ;DATO A ESCRIBIR EN MEMORIA !!!!
movwf   SSPBUF

call    Tx_Dato_Wait     ;esperamos a que el byte se transmita

;incrementamos la direccion a guardar el siguiente byte, y incrementamos
;el contador de byte escritos en la memores

incf   LSB,f            ;incrementamos en 1 la siguiente direccion
incf   CONTADOR_LSB,f   ;incrementamos el contador
movf   LSB,w            ;el contenido de LSB se guarda en W
sublw  0x00
btfss STATUS,2;si bit Z es uno, saltar

```

```

return                                ;no fue cero

incf  MEDIO,f
incf  CONTADOR_MEDIO,f
movf  MEDIO,w
sublw 0x00
btfss STATUS,2;si bit Z es uno, saltar
return

incf  MSB,f
incf  CONTADOR_MSB,f
movf  MSB,w
sublw 0x02
btfss STATUS,2;si bit Z es uno, saltar
return

call clear_contadores_datos_memoria ;como la memoria ya se lleno,
                                     ;direccionamos al inicio de la memoria
return

```

```

;-----
;
; Rutina que espera a que un dato se trasmite a la memoria via spi
; argumentos: no hay
;
;-----

```

```

Tx_Dato_Wait
    bsf    STATUS,RP0    ;Banco 1

Rx_Dato2
    btfss  SSPSTAT,BF    ;si BF es uno se ha recibido el dato
    goto  Rx_Dato2;no

    bcf    STATUS,RP0    ;Selecciona banco 0

return

```

```

;-----
;
; Rutina que envia la direccion de la memoria en donde un dato se guardara
; argumentos: la direccion deseada MSB-MEDIO-LSB
;
;-----

```

```

direccion
    retardo_x .6    ;llama un macro de retardo

;envio de msb
    movf  MSB,W
    movwf SSPBUF

    call  Tx_Dato_Wait

;-----
    retardo_x .6    ;llama un macro 3+3*(x=20)=63uS

;envio de medio
    movf  MEDIO,W
    movwf SSPBUF

    call  Tx_Dato_Wait

;-----
    retardo_x .6    ; llama un macro 3+3*(x=20)=63

;envio de lsb
    movf  LSB,W
    movwf SSPBUF

```

```

        call    Tx_Dato_Wait
        return

;-----
;
; Rutina que lee todos los datos escritos en la memoria asta el momento
; argumentos: no hay
;-----
leer_todos_los_datos_guardados

        clrf   CONTADOR_MSB_LECTURA ;ponemos a cero el contador de datos leídos
        clrf   CONTADOR_MEDIO_LECTURA
        clrf   CONTADOR_LSB_LECTURA

        clrf   MSB ;direccionamos al inicio de la memoria (direccion cero)
        clrf   MEDIO
        clrf   LSB

loop1
        retardo_x .6 ; llama un macro retardo 3+3*(x=6)=21 ciclos de reloj

;LECTURA DE LA MEMORIA
        movlw  b'00000001' ;comando de lectura
        movwf  SSPBUF
        call   Tx_Dato_Wait

;-----
;
;-----
        call   direccion ;enviamos la direccion a ser leida
;-----

;LEER DEL SLAVE

        retardo_x .6 ; llama un macro 3+3*(x=6)=21

        call   recibir_spi ;recibimos el dato leido y lo ponemos en W
        page_0
        call   Tx_Dato_sci ;el dato en W es enviado a la pc.
        page_1

;vemos si ya se leyo toda la memoria (comparamos datos leidos con datos
;escritos)
        movf   CONTADOR_LSB,w
        subwf  CONTADOR_LSB_LECTURA,w
        btfss STATUS,2
        goto  salto1
        movf   CONTADOR_MEDIO,w
        subwf  CONTADOR_MEDIO_LECTURA,w
        btfss STATUS,2
        goto  salto2
        movf   CONTADOR_MSB,w
        subwf  CONTADOR_MSB_LECTURA,w
        btfss STATUS,2
        goto  salto3
        goto  salir

salto1
salto2
salto3

;direccionamos a la siguiente localidad de memoria y incrementamos el
;contador de datos leido
        incf   CONTADOR_LSB_LECTURA,F
        incf   LSB,f
        movf   LSB,w

```

```

        sublw    0x00
        btfss   STATUS,2          ;probara si bit Z es  cero
        goto    loop1

        incf    CONTADOR_MEDIO_LECTURA,F
        incf    MEDIO,f
        movf    MEDIO,w
        sublw   0x00
        btfss   STATUS,2          ;probara si bit Z es  cero
        goto    loop1

        incf    CONTADOR_MSB_LECTURA,F
        incf    MSB,f
        movf    MSB,w
        sublw   0x02
        btfss   STATUS,2          ;probara si bit Z es  cero
        goto    loop1          ;vamos a leer el siguiente byte

salir
        return

;-----
;
;   Rutina que espera un dato a resiva por la spi de la memoria
;   argumentos: registros
;
;-----

recibir_spi

        movlw   b'00010110'      ;enviar un dato cualquiera solo para generar reloj
        movwf   SSPBUF           ;Almacena el byte a transmitir en el buffer

        call    Tx_Dato_Wait     ;esperamos a que el dato se transmita
        movf    SSPBUF, W        ;el byte recibido es puesto en W
        return

clear_contadores_datos_memoria      ;inicializamos registros usados en memoria ext.
        clrf    MSB              ;direccionamos al inicio de la memoria
        clrf    MEDIO
        clrf    LSB

        clrf    CONTADOR_MSB     ;borramos el contador de datos escritos.
        clrf    CONTADOR_MEDIO
        clrf    CONTADOR_LSB

        return

;-----
;
;   Rutina para conf. la acumulacion trifasica en el ade, poniendo el cf trifasico
;   argumentos: registros
;
;-----

set_trifasico
        btfsc   TIPO_SERVICIO,0x03      ;tipo de servicio 9s ?
        goto    conf_9s_reg_1
        goto    tipo_deservico_1_1
conf_9s_reg_1

        movlw   b'00111111'          ;conf. la Potencia activa en LAENERGY (fijamos la formula)
        movwf   OUTD1                ; $P=V_a*I_a+V_b*I_b+V_c*I_c$ 
        movlw   WATMODE_W            ;comando de escritura a WATMODE
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03      ;enviar solo un byte
        page_0
        call    SPIDX
        page_1

```

```

movlw b'00111111' ;conf. la Potencia aparente en LVAENERGY (fijamos la formula)
movwf OUTD1 ;S=Va*Ia+Vb*Ib+Vc*Ic
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

goto conf_reg_1

tipo_deservico_1_1

btfsc TIPO_SERVICIO,0x02 ;tipo de servicio 8s ?
goto conf_8s_reg_1
goto tipo_deservico_1_2
conf_8s_reg_1

movlw b'10101101' ;conf. P=Va*(Ia-Ib)+0+Vc*Ic
movwf OUTD1
movlw WATMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
page_0
call SPIDX
page_1

movlw b'10111111' ;conf. S=Va*Ia+Vb*Ib+Vc*Ic
movwf OUTD1
movlw VAMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
page_0
call SPIDX
page_1

goto conf_reg_1

tipo_deservico_1_2

btfsc TIPO_SERVICIO,0x01 ;tipo de servicio 6s ?
goto conf_6s_reg_1
goto tipo_deservico_1_3
conf_6s_reg_1

movlw b'01101101' ;conf. P=Va*(Ia-Ib)+0+Vc*(Ic-Ib)
movwf OUTD1
movlw WATMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
page_0
call SPIDX
page_1

movlw b'01111111' ;conf. S=Va*Ia+(Va+Vc)/2*Ib+Vc*Ic
movwf OUTD1
movlw VAMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
page_0
call SPIDX
page_1
goto conf_reg_1

tipo_deservico_1_3

```

```

        btfsc    TIPO_SERVICIO,0x00      ;tipo de servicio 5s ?
        goto    conf_5s_reg_1
        goto    conf_reg_1
conf_5s_reg_1

        movlw   b'00101101'           ;conf. P=Va*Ia+0+Vc*Ic
        movwf   OUTD1
        movlw   WATMODE_W
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03
        page_0
        call    SPIDX
        page_1

        movlw   b'00101101'           ;conf. S=Va*Ia+0+Vc*Ic
        movwf   OUTD1
        movlw   VAMODE_W
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03
        page_0
        call    SPIDX
        page_1
conf_reg_1

        movlw   b'00000000'           ;conf. LAENERGY para que en este se acumule potencia activa (P) y no Q
        movwf   OUTD1
        movlw   WAVMODE_W
        movwf   SPI_COMANDO
        bsf     LENG_DATOS,0x03
        page_0
        call    SPIDX
        page_1

        return
;-----
;
; Rutina para leer todas las energias trifasicas (S, P y Q) del ADE7754
; argumentos: no hay
;
;-----
LEER_REG_ENERGY_TRIFASICA
        bcf     STATUS,RP1             ;Selecciona banco 0 y 1
        bcf     STATUS,RP0             ;Selecciona banco 0

espera_1_fase_a_clear_2 ;esperamos a que la IRQ del ADE sea active (cambie de 1 a 0 logico)

        btfsc   INTCON,INTF           ;y por tanto un cambio de estado en RB0
        goto    salir_1_fase_a_clear_2 ;ya estuvo
        goto    espera_1_fase_a_clear_2 ;seguir esperando

salir_1_fase_a_clear_2
        bcf     INTCON,INTF           ;ponemos a cero la bandera de peticion de int. externa
        movlw   0x11                  ;lectura con reset del registro de STATUS de INT. del ADE
        movwf   SPI_COMANDO
        clrf    LENG_DATOS
        bsf     LENG_DATOS,0x01 ;leer 2 bytes
        page_0
        call    SPIRX
        page_1

        btfss   INSD2,0x02             ;la fuente de la int. fue por el fin de la acumulacion
                                           ;de energia de LAENERGY y LVAENERGY?

        goto    espera_1_fase_a_clear_2 ;no, seguir esperando

;leemos LAENERGY y LVAENERGY solo para limpiar estos reg.

```

```

movlw LAENERGY ;cargamos el comando para leer el reg.LAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02 ;leer 3 bytes
page_0
call SPIRX ;los valores leidos no se guardan
page_1

movlw LVAENERGY ;cargamos el comando para leer el reg.LAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02 ;leer 3 bytes
page_0
call SPIRX ;los valores leidos no se guardan
page_1

;configuracion de los acumuladores LAENERGY y LVAENERGY

btfsc TIPO_SERVICIO,0x03 ;tipo de servicio 9s ?
goto conf_9s_reg_4_2 ;si
goto tipo_deservicio_4_1_2 ;no

;----- configuramos las formulas para tipo de servicio 9S -----
conf_9s_reg_4_2
movlw b'00111111' ;conf. la Potencia real en LAENERGY (fijamos la formula)
movwf OUTD1 ; $P=V_a*I_a+V_b*I_b+V_c*I_c$ 
movlw WATMODE_W ;comando de escritura a WATMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

movlw b'00111111' ;conf. la Potencia aparente en LVAENERGY (fijamos la formula)
movwf OUTD1 ; $S=V_a*I_a+V_b*I_b+V_c*I_c$ 
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

movlw b'00000000' ;conf. LAENERGY para que en este se acumule potencia activa
;(P) y no Q
movwf OUTD1
movlw WAVMODE_W ;comando de escritura a WAVMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1
goto conf_reg_4_2 ;vamos a medir la energia consumida

tipo_deservicio_4_1_2
btfsc TIPO_SERVICIO,0x02 ;tipo de servicio 8s ?
goto conf_8s_reg_4_2 ;si
goto tipo_deservicio_4_2_2 ;no

;----- configuramos las formulas para tipo de servicio 8S -----
conf_8s_reg_4_2
movlw b'10101101' ;conf. la Potencia real en LAENERGY (fijamos la
;formula)
movwf OUTD1 ; $P=V_a*(I_a-I_b)+0+V_c*I_c$ 
movlw WATMODE_W ;comando de escritura a WATMODE

```

```

movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

movlw b'10111111' ;conf. la Potencia aparente en LVAENERGY (fijamos la formula)
movwf OUTD1 ;S=Va*Ia+Vb*Ib+Vc*Ic
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

movlw b'00000000' ;conf. LAENERGY para que en este se acumule potencia activa
;(P) y no Q

movwf OUTD1
movlw WAVMODE_W ;comando de escritura a WAVMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1
goto conf_reg_4_2 ;vamos a medir la energia consumida

tipo_deservicio_4_2_2
btfsc TIPO_SERVICIO,0x01 ;tipo de servicio 6s ?
goto conf_6s_reg_4_2 ;si
goto tipo_deservicio_4_3_2 ;no

;----- configuramos las formulas para tipo de servicio 6S -----

conf_6s_reg_4_2
movlw b'01101101' ;conf. la Potencia real en LAENERGY (fijamos la formula)
movwf OUTD1 ;P=Va*(Ia-Ib)+0+Vc*(Ic-Ib)
movlw WATMODE_W ;comando de escritura a WATMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

movlw b'01111111' ;conf. la Potencia aparente en LVAENERGY (fijamos la formula)
movwf OUTD1 ;S=Va*Ia+(Va+Vc)/2*Ib+Vc*Ic
movlw VAMODE_W ;comando de escritura a VAMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1

movlw b'00000000' ;conf. LAENERGY para que en este se acumule potencia activa
;(P) y no Q

movwf OUTD1
movlw WAVMODE_W ;comando de escritura a WAVMODE
movwf SPI_COMANDO
bsf LENG_DATOS,0x03 ;enviar solo un byte
page_0
call SPIDX
page_1
goto conf_reg_4_2 ;vamos a medir la energia consumida

tipo_deservicio_4_3_2
btfsc TIPO_SERVICIO,0x00 ;tipo de servicio 5s ?
goto conf_5s_reg_4_2

```



```

return    ;servicio desconsido (este caso nunca se puede dar)

;----- configuramos las formulas para tipo de servicio 5S -----

conf_5s_reg_4_2
movlw    b'00101101'      ;conf. la Potencia real en LAENERGY (fijamos la formula)
movwf    OUTD1            ; $P=V_a*I_a+0+V_c*I_c$ 
movlw    WATMODE_W       ;comando de escritura a WATMODE
movwf    SPI_COMANDO
bsf      LENG_DATOS,0x03 ;enviar solo un byte
page_0
call     SPIDX
page_1

movlw    b'00101101'      ;conf. la Potencia aparente en LVAENERGY (fijamos la formula)
movwf    OUTD1            ; $S=V_a*I_a+0+V_c*I_c$ 
movlw    VAMODE_W        ;comando de escritura a VAMODE
movwf    SPI_COMANDO
bsf      LENG_DATOS,0x03 ;enviar solo un byte
page_0
call     SPIDX
page_1

movlw    b'00000000'      ;conf. LAENERGY para que en este se acumule potencia activa
                                ;(P) y no Q
movwf    OUTD1
movlw    WAVMODE_W       ;comando de escritura a WAVMODE
movwf    SPI_COMANDO
bsf      LENG_DATOS,0x03 ;enviar solo un byte
page_0
call     SPIDX
page_1

conf_reg_4_2                ;vamos a medir la energia consumida

;leer la energia trifasica
espera_1_fase_a_2
btfsc   INTCON,INTF       ;verificamos la bandera de PB0_int si esta activada?
goto    salir_1_fase_a_2  ;si
goto    espera_1_fase_a_2 ;no

salir_1_fase_a_2
bcf     INTCON,INTF       ;borramos la bandera de int. externa
movlw   0x11              ;lectura con reset del registro de estado de int. del ADE
movwf   SPI_COMANDO
clrf   LENG_DATOS
bsf    LENG_DATOS,0x01 ;leer 2 bytes
page_0
call   SPIRX              ;no se guarda el resultado
page_1

btfss   INSD2,0x02        ;la fuente de la int. fue por el fin de la acumulacion
                                ;de energia de LAENERGY y LVAENERGY?
goto    espera_1_fase_a_2 ;no, esperar

;repetimos la operaciones anteriores para asegurar nos que tendremos datos validos
;en LAENERGY y LVAENERGY

espera_2_fase_a_2
btfsc   INTCON,INTF
goto    leer_P_S_fase_a_2
goto    espera_2_fase_a_2
leer_P_S_fase_a_2

bcf     INTCON,INTF
movlw   0x11              ;lectura con reset

```

```

movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x01
page_0
call SPIRX ;no se guarda el resultado
page_1

btfss INSD2,0x02
goto espera_2_fase_a_2

;leemos LAENERGY y LVAENERGY y los almacenamos en variables temporales
movlw LAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02 ;leer 3 bytes
page_0
call SPIRX
page_1

movf INSD3,W
bsf STATUS,RP0 ;banco 1
movwf P_TRIFASICA_2 ;guardamos el MSB
bcf STATUS,RP0 ;banco 0

movf INSD2,W
bsf STATUS,RP0 ;banco 1
movwf P_TRIFASICA_1 ;guardamos el BYTE MEDIO
bcf STATUS,RP0 ;banco 0

movf INSD1,W
bsf STATUS,RP0 ;banco 1
movwf P_TRIFASICA_0 ;guardamos el LSB
bcf STATUS,RP0 ;banco 0

movlw LVAENERGY
movwf SPI_COMANDO
clrf LENG_DATOS
bsf LENG_DATOS,0x02 ;leemos 3 bytes
page_0
call SPIRX
page_1

movf INSD3,W
bsf STATUS,RP0 ;banco 1
movwf S_TRIFASICA_2 ;guardamos el MSB
bcf STATUS,RP0 ;banco 0

movf INSD2,W
bsf STATUS,RP0 ;banco 1
movwf S_TRIFASICA_1 ;guardamos el BYTE MEDIO
bcf STATUS,RP0 ;banco 0

movf INSD1,W
bsf STATUS,RP0 ;banco 1
movwf S_TRIFASICA_0 ;guardamos el LSB
bcf STATUS,RP0 ;banco 0

;conf. el ADE para leer Q
movlw b'00100000' ;conf. LAENERGY para que en este se acumule Qr(WAVMODE<5>=1)
movwf OUTD1
movlw WAVMODE_W
movwf SPI_COMANDO
bsf LENG_DATOS,0x03
page_0
call SPIDX

```

```

page_1
espera_Q_1_fase_a_2
    btfsc    INTCON,INTF        ;verificamos la bandera de PB0_int
    goto    salir_Q_1_fase_a_2
    goto    espera_Q_1_fase_a_2
salir_Q_1_fase_a_2

    bcf     INTCON,INTF
    movlw  0x11                ;lectura con reset del registro de estado de int. del ADE
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf   LENG_DATOS,0x01
page_0
    call   SPIRX                ;no se guarda el resultado
page_1

    btfss  INSD2,0x02          ;la fuente de la int. fue por el fin de la acumulacion
    ;de energia de LAENERGY y LVAENERGY?

    goto   espera_Q_1_fase_a_2    ;no

    ;repetimos la operaciones anteriores para asegurar nos que tendremos datos validos
    ;en LAENERGY
espera_Q_2_fase_a_2
    btfsc    INTCON,INTF        ;verificamos la bandera de PB0_int
    goto    leer_Q_fase_a_2
    goto    espera_Q_2_fase_a_2
leer_Q_fase_a_2

    bcf     INTCON,INTF
    movlw  0x11                ;lectura con reset de registro de estado de int. del ADE
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf   LENG_DATOS,0x01
page_0
    call   SPIRX                ;no se guarda el resultado
page_1

    btfss  INSD2,0x02          ;la fuente de la int. fue por el fin de la acumulacion
    ;de energia de LAENERGY y LVAENERGY?

    goto   espera_Q_2_fase_a_2    ;no

    movlw  LAENERGY            ;leemos el signo de Q en el reg.LAENERGY
    movwf  SPI_COMANDO
    clrf   LENG_DATOS
    bsf   LENG_DATOS,0x02 ;leer 3 bytes
page_0
    call   SPIRX
page_1

    movf   INSD3,W
    bsf   STATUS,RP0            ;banco 1
    movwf  Q_TRIFASICA_2        ;guardamos el MSB
    bcf   STATUS,RP0            ;banco 0

    movf   INSD2,W
    bsf   STATUS,RP0            ;banco 1
    movwf  Q_TRIFASICA_1        ;guardamos el BYTE MEDIO
    bcf   STATUS,RP0            ;banco 0

    movf   INSD1,W
    bsf   STATUS,RP0            ;banco 1
    movwf  Q_TRIFASICA_0        ;guardamos el LSB
    bcf   STATUS,RP0            ;banco 0

```

```

return

ENVIAR_REG_ENERGY_TRIFASICA_PC
;FASE A
page_0

bcf STATUS,RP1 ;banco 0 o 1
bsf STATUS,RP0 ;banco 1
movf P_TRIFASICA_2,W ;byte MSB, de la potencia activa
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf P_TRIFASICA_1,W ;byte medio, de la potencia activa
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf P_TRIFASICA_0,W ;byte LSB, de la potencia activa
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf S_TRIFASICA_2,W ;byte MSB, de la potencia aparente
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf S_TRIFASICA_1,W ;byte medio, de la potencia aparente
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf S_TRIFASICA_0,W ;byte LSB, de la potencia aparente
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf Q_TRIFASICA_2,W ;byte MSB, de la potencia reactiva
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf Q_TRIFASICA_1,W ;byte medio, de la potencia reactiva
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci

bsf STATUS,RP0 ;banco 1
movf Q_TRIFASICA_0,W ;byte LSB, de la potencia reactiva
bcf STATUS,RP0 ;banco 0
call Tx_Dato_sci
return
,*****
end ;FIN del codigo fuente

```

;CODIGO FUENTE DEL RELOJ

;reloj.asm

```
*****
;
;   Trabajo de Graduacion: Diseño e implementacion de un medidor trifasico
;                           multifunciones utilizando el IC ADE7754
;
;
;   Estudiantes:           -Daniel Antonio Cortez Franco
;                           - Douglas Alberto Lopez Hernandez
;                           - Fernando Alberto Arevalo Navas
;
;
;   fecha:13/01/2005
*****
;
;   Este programa es un reloj implementado con le TMR1, el TMR1 es cargado el
;   el complemento a dos de 0xf424 (62500 en decimal), por tanto el timer se
;   desborda cada medio segundo. y implementa la interfaz SPI; asi como peticion
;   de interrupcion para fijar la hora y solicitar la hora actual.
*****
```

```
List      p=16F877      ;Tipo de procesador
include   "P16F877.INC" ;Definiciones de registros internos
include   "F:\tesis\pic16f877\archivos_include\macros.mac"
```

```
__config H'3f72'
__idlocs H'7777'
```

```
TEMPO2      EQU      0x22      ;Contador de para el TMR1, su valor actual es 2
SEGUNDOS1   EQU      0x23      ;tiempo en segundos = SEGUNDOS4-SEGUNDOS3-
SEGUNDOS2   EQU      0x24      ;SEGUNDOS2-SEGUNDOS1
SEGUNDOS3   EQU      0x25
SEGUNDOS4   EQU      0x26
SEGUNDOS    EQU      0x27      ;reg. que cuenta 60 seg.
MINUTOS     EQU      0x28      ;reg. que cuenta 12 o 20 min.
INT_PORTB   EQU      0x29      ;reg. que almacena el estado del PORTB
```

```
##### Programa principal #####
```

```
org         0x00      ;Vector de Reset
goto       Inicio

org         0x04      ;Vector de interrupción
btfsc     PIR1,TMR1IF ;int. provocada por TMR1?
goto      Inter_TMR1 ;si
btfsc     INTCON,RBIF ;int. provocada por un cambio de esta de RB4:RB7
goto      Inter_RBIF ;SI
retfie    ;falsa alarma
```

```
;----- INT. del TMR1 -----
Inter_TMR1
```

```
bcf       PIR1,TMR1IF ;Restaura el flag del TMR1

movlw    0xdc          ;recargamos el valor del TMR1
movwf    TMR1L
movlw    0x0b
movwf    TMR1H

decfsz   TEMPO2,F ;Se han provocado 2 rebasamientos de 0.5 Seg. (1s.) ?
retfie   ;Todavía no
```

```
inc_segundos1
```

```
flash_led PORTD,0x07
movlw    .2
movwf    TEMPO2      ;Reponer variable de temporización con el valor 2
incf     SEGUNDOS1,F ;incrementamos el byte lsb del tiempo
movlw    .0
```

```

subwf SEGUNDOS1,W
btfsc STATUS,Z;ya se desbordo el byte lsb?
goto inc_segundos2 ;si
goto inc_segundos ;no

inc_segundos2
incf SEGUNDOS2,F ;incrementamos byte medio lsb
movlw .0
subwf SEGUNDOS2,W
btfsc STATUS,Z
goto inc_segundos3
goto inc_segundos

inc_segundos3
incf SEGUNDOS3,F ;incrementamos byte medio msb
movlw .0
subwf SEGUNDOS3,W
btfsc STATUS,Z
goto inc_segundos4
goto inc_segundos

inc_segundos4
incf SEGUNDOS4,F ;incrementamos byte msb

inc_segundos
incf SEGUNDOS,F ;incrementamos el contador de 60 seg.
movlw .60
subwf SEGUNDOS,W
btfsc STATUS,Z;pasaron 60 seg.?
goto inc_minutos ;si
retfie ;no

inc_minutos
clrf SEGUNDOS ;limpiamos el contador de 60 seg.

incf MINUTOS,F ;incrementamos el contador de 12 o 20 min.
btfsc PORTA,0x03 ;boton de conf. para contar asta 12 o 20 min.
goto cada_5min ;turn on (V=1)
goto cada_3min ;turn off (V=0)

cada_5min
movlw .5
subwf MINUTOS,W
btfsc STATUS,Z;pasaron 5 min?
call minutos_5 ;si, enviar señal al pic16f877 maestro

movlw .10
subwf MINUTOS,W
btfsc STATUS,Z;pasaron 10 min?
call minutos_5 ;si, enviar señal al pic16f877 maestro

movlw .15
subwf MINUTOS,W
btfsc STATUS,Z;pasaron 15 min?
call minutos_5 ;si, enviar señal al pic16f877 maestro

movlw .20
subwf MINUTOS,W
btfsc STATUS,Z;pasaron 20 min?
goto clear_minutos ;limpiamos el acumulador de 12 o 20 min.
retfie

cada_3min
movlw .3
subwf MINUTOS,W

```

```

    btfsc    STATUS,Z;pasaron 3 min?
    call     minutos_5      ;si, enviar señal al pic16f877 maestro

    movlw   .6
    subwf   MINUTOS,W
    btfsc   STATUS,Z;pasaron 6 min?
    call     minutos_5      ;si, enviar señal al pic16f877 maestro

    movlw   .9
    subwf   MINUTOS,W
    btfsc   STATUS,Z;pasaron 9 min?
    call     minutos_5      ;si, enviar señal al pic16f877 maestro

    movlw   .12
    subwf   MINUTOS,W
    btfsc   STATUS,Z;pasaron 12 min?
    goto    clear_minutos   ;limpiamos el acumulador de 12 o 20 min.
    retfie

clear_minutos
    clrf    MINUTOS        ;limpiamos el acumulador de 12 o 20 min.
    call    minutos_20     ;enviar señal al pic16f877 maestro, y luego enviar el
                          ;tiempo actual.
    retfie

;----- INT. de cambio de estado de RB4:RB7 -----
Inter_RBIF

    movf    PORTB,W        ;leemos el port. B actual
    bcf     INTCON,RBIF    ;ponemos la bandera nuevamente a cero
    xorwf   INT_PORTB,F    ;vemos que pin del puerto B a cambiado (comparar el
                          ;valor actual y el anterior del port. b)

    btfsc   INT_PORTB,0x05 ;cambio de estado el pin RB5?
    goto    set_datos      ;si, cambiar el la hora del reloj
    btfsc   INT_PORTB,0x04 ;cambio de estado el pin RB4?
    goto    datos_actual    ;si
    movwf   INT_PORTB
    retfie                  ;int. No valida

datos_actual
    movwf   INT_PORTB
    call    minutos_actual  ;enviar el tiempo actual
    retfie

set_datos ;modificar hora del reloj
    movwf   INT_PORTB
    call    Rx_dato_spi
    movwf   SEGUNDOS4      ; byte msb
    call    Rx_dato_spi
    movwf   SEGUNDOS3
    call    Rx_dato_spi
    movwf   SEGUNDOS2
    call    Rx_dato_spi
    movwf   SEGUNDOS1      ;byte lsb
    clrf   SEGUNDOS        ;limpiamos el contador de 60 seg.
    clrf   MINUTOS         ;limpiamos el contador de 12 o 20 min.
    retfie

;-----FIN de la interrupcion-----
Inicio

    bcf     STATUS,RP1
    bcf     STATUS,RP0     ;Selecciona banco 0

    clrf   PORTA
    clrf   PORTB

```

```

clrf    PORTC
clrf    PORTD

bsf     STATUS,RP0      ;Selecciona banco 1

;Configura el puerto A
movlw   0x06            ;Configuramos todos los pines como I/O (cap. 11)
movwf   ADCON1
movlw   b'00101000'    ; configuramos #SS(TRISA<5>)=1 segun especificaciones
movwf   TRISA

;Configura el puerto B
movlw   b'00110000'    ;configurmos el puertoB como salidas
movwf   TRISB

;Configura el puerto C
movlw   b'10011000'
movwf   TRISC          ;configuramos sck(TRISC<3>)=1, SDI(TRISC<4>)=1
                        ;SDO(TRISC<5>)=0, TX(TRISC<6>)=0, RX(TRISC<7>)=1
                        ;y los demas pines como salidas.

;Configura el puerto B
movlw   b'00000000'    ;configuramos el puertoB como salidas
movwf   TRISD

movlw   b'10000111'
movwf   OPTION_REG     ;divisor de frec. de 256 asociado al TMR0

##### configuracion del SPI MODO ESCLAVO #####

bcf     STATUS,RP0      ;banco 0
movlw   b'00000100'    ;configuramos WCOL=0, SSPOV=0, SSPEN=0, CKP=0,
                        ;SSPM3:SSPM0=0100 (esclavo). SPI desactivada
movwf   SSPCON

bsf     STATUS,RP0      ;banco 1
movlw   b'00000000'    ;configuramos SMP(SSPSTAT<7>)=0, CKE(SSPSTAT<6>)=0,
                        ;BF(SSPSTAT<0>)=0
movwf   SSPSTAT

bcf     STATUS,RP0      ;banco 0
bsf     SSPCON,SSPEN    ;se activa la SPI

bcf     STATUS,RP0      ;Selecciona banco 0

##### configuracion del TMR1 #####
movlw   .2
movwf   TEMPO2          ;reg. que cuenta 2 desbordes del TMR1
movlw   0x0b
movwf   TMR1H           ;cargamos el TMR1=0x0bdc
movlw   0xdc
movwf   TMR1L
movlw   b'00110000'
movwf   T1CON           ;divisor de frec. de 8 asociado al TMR1

bsf     STATUS,RP0      ;Selecciona banco 1

movf    PORTB,W          ;borramos la bandera RBIF
bcf     INTCON,RBIF
bsf     INTCON,RBIE      ;activamos la int. de cambio de estado en los pines
                        ;RB4:RB7
bsf     PIE1,TMR1IE     ;activamos la int. de sobre flujo del TMR1

#####
FIN DE LA CONFIGURACION #####

bcf     STATUS,RP0      ;Selecciona banco 0

```



```

;Borramos los acumuladores del tiempo conformado por 4 bytes
clrf    SEGUNDOS1    ;lsb
clrf    SEGUNDOS2
clrf    SEGUNDOS3
clrf    SEGUNDOS4    ;msb

;borramos los acumuladores de segundo y minutos
clrf    SEGUNDOS
clrf    MINUTOS

bsf     T1CON, TMR1ON ;activamos el TMR1
bsf     INTCON, PEIE   ;habilitamos la int. de perifericos
bsf     INTCON, GIE    ;habilitamos las int. globales

loop    nop           ;esperamos que ocurra una int.
        goto    loop

;##### DEFINICION DE RUTINAS #####

;-----
;
; Rutina para Transmitir un byte por medio SPI
; argumentos: dato en W
;-----
Tx_Dato_spi
        bcf     STATUS, RP0    ;Selecciona banco 0

        bcf     PIR1, SSPIF    ;se limpia la bandera de dato transmitido
        movwf   SSPBUF         ;Almacena el byte a transmitir

Tx_Dato_spi_Wait
        btfss   PIR1, SSPIF    ;Byte transmitido ?
        goto    Tx_Dato_spi_Wait ;No, esperar

        bcf     PIR1, SSPIF    ;se limpia la bandera de dato transmitido
        return

;-----
;
; Rutina para Recibir un byte por medio SPI
; argumentos: no hay!!
;-----
Rx_dato_spi

        movlw   0x05          ;ponemos un dato cualquiera en el buffer, para transmitirlo
        movwf   SSPBUF        ;este dato sera ignorado por el pic16f877 maestro.

Tx_Dato_Wait2
        btfss   PIR1, SSPIF    ;Byte transmitido ?
        goto    Tx_Dato_Wait2 ;No, esperar

        bcf     PIR1, SSPIF    ;se limpia la bandera de dato transmitido

        bsf     STATUS, RP0    ;Banco 1
Rx_Dato
        btfss   SSPSTAT, BF    ;si BF es uno se ha recibido el dato
        goto    Rx_Dato        ;no

        bcf     SSPSTAT, BF

```

```

        bcf      STATUS,RP0      ;Selecciona banco 0

        movf    SSPBUF, W        ;SSPBUF -> W
        return

minutos_5
        flash_led PORTA,0x01    ;activamos la int. de cambio de estado en PORTB
                                   ;(RB5) en el pic16f877 maestro
        return

minutos_20
        flash_led PORTA,0x00    ;activamos la int. de cambio de estado en PORTB
                                   ;(RB4) en el pic16f877 maestro
        return

minutos_actual
        movf    SEGUNDOS4,W      ;transmitimos la hora actual comenzando con el byte msb
        call    Tx_Dato_spi      ;y finalizando con el lsb
        movf    SEGUNDOS3,W
        call    Tx_Dato_spi
        movf    SEGUNDOS2,W
        call    Tx_Dato_spi
        movf    SEGUNDOS1,W
        call    Tx_Dato_spi

        return

;*****
end      ;FIN del codigo fuente

```

;CODIGO FUENTE LA MEMORIA
;memoria.asm

```

;*****
;
;      Trabajo de Graduacion: Diseño e implementacion de un medidor trifasico
;                               multifunciones utilizando el IC ADE7754
;
;
;      Estudiantes:      -Daniel Antonio Cortez Franco
;                       - Douglas Alberto Lopez Hernandez
;                       - Fernando Alberto Arevalo Navas
;
;
;      fecha:13/01/2005
;*****
;
;      Implementacion de una memoria con interfaz SPI.
;      Este programa recibe un comando ya sea de lectura o de escritura a memoria, luego la
;      direccion, luego el dato y lo guarda en memoria (si fue comando de escritura) o
;      envia el dato direccionado en la memoria al PC (si fue un comando de lectura).
;
;
;      List      p=16F877A
;      include   "P16F877A.INC"
;      __config 0x3f72
;      __idlocs 0x7777

TEMPO1      EQU      0x21      ;contador usado en las rutinas de retardo

```

```

MSB      EQU      0x22      ;apuntador a la memoria = MSB-MEDIO-LSB
MEDIO    EQU      0x23
LSB      EQU      0x24

DATO     EQU      0x25      ;reg. para almacenar el dato aguardar
COPIA_W  EQU      0x26      ;reg. que guarda el comando recibido de la PC
OUT1     EQU      0x27      ;reg. usado para poner el dato a transmitir por la SPI

##### Programa principal #####

org      0x00      ;Vector de Reset
goto     Inicio

org      0x04      ;Vector de int.

btfsc   PIR1,SSPIF      ;int. de la SPI
goto    interrupcion_spi ;si
retfie                          ;no

interrupcion_spi

bcf     PIR1,SSPIF      ;limpiamos la bandera de la interrupcion SPI
call    rx_dato_wait    ;retardo para leer el dato recibido por la SPI
movf    SSPBUF, W       ;guardamos el dato recibido
movwf   COPIA_W

sublw   0x00            ;el dato recibido es un comando de escritura?
btfsc   STATUS,2        ;probara si bit Z es cero
goto    fue_escritura   ;si
goto    no_es_escritura ;no

fue_escritura

call    escritura        ;fue uno
call    desabilitada_lectura_escritura
retfie

no_es_escritura

movf    COPIA_W,W
sublw   0x01            ;el dato recibido es un comando de lectura?
btfsc   STATUS,2        ;probara si bit Z es cero
call    lectura          ;si, es lectura
call    desabilitada_lectura_escritura
retfie

Inicio

bsf     STATUS,RP1
bsf     STATUS,RP0      ;Selecciona banco 1

;Configura el puerto A
movlw   0x06            ;Configuramos todos los pines como I/O (cap. 11)
movwf   ADCON1
movlw   b'00100000'    ; configuramos #SS(TRISA<5>)=1 segun especificaciones
movwf   TRISA          ; como entrada

;Configura el puerto C
movlw   b'00011000'    ;configuramos sck(TRISC<3>)=1, SDI(TRISC<4>)=1
movwf   TRISC          ;SDO(TRISC<5>)=0;y los demas pines como salidas.

;Configura el puerto D
movlw   b'00000000'    ;configuramos el puerto D como salidas
movwf   TRISD

;Configura el puerto E
movlw   b'00000000'    ;configuramos el puerto E como salidas
movwf   TRISE

```

```

        movlw    b'00000000'
        movwf   OPTION_REG      ;divisor de frec. Es 2 asociado al TMR0 es 2

;##### configuracion del SPI MODO ESCLAVO #####

        bcf     STATUS,RP0      ;banco 0
        movlw   b'00000100'     ;configuramos WCOL=0, SSPOV=0, SSPEN=0, CKP=0,
                                ;SSPM3:SSPM0=0100 (esclavo). SPI desactivada
        movwf   SSPCON

        bsf     STATUS,RP0      ;banco 1
        movlw   b'00000000'     ;configuramos SMP(SSPSTAT<7>)=1, CKE(SSPSTAT<6>)=0,
                                ;BF(SSPSTAT<0>)=0
        movwf   SSPSTAT

        bcf     STATUS,RP0      ;banco 0
        bsf     SSPCON,SSPEN    ;se activa la SPI

;#####

        bsf     STATUS,RP0      ;Selecciona banco 1

        bsf     PIE1,SSPIE     ;se habilita el bit de interrupcion para la bandera SSPIF

        bcf     STATUS,RP0      ;Selecciona banco 0

        bsf     INTCON,PEIE     ;se habilita el bit de interrupcion para los
                                ;perifericos
        bsf     INTCON,GIE      ;se habilita el bit de interrupcion para las
                                ;interrupciones globales

        bcf     STATUS,RP0      ;Selecciona banco 0
        call    desabilitada_lectura_escritura

loop    nop
        goto   loop

;##### DEFINICON DE RUTINAS #####

;-----
;
; Rutina que deshabilita la lectura y la escritura a la memoria DS1245Y/AB
; argumentos: no hay
;
; nota: vea el diagrama de conexion de la memora y el pic16f877
;-----
desabilitada_lectura_escritura
        movlw   b'00000111'     ;deshabilitada la lectura y escritura de la memoria
        movwf   PORTE          ;PE0=/CE ;PE1=/WE ;PE2=/OE
        return

;-----
;
; Rutina que habilita la escritura a la memoria DS1245Y/AB
; argumentos: no hay
;
;

```

```

;      nota: vea el diagrama de conexion de la memoria y el pic16f877
;-----
habilitada_escritura
    movlw    b'00000100'    ;deshabilitada la lectura y habilitada la escritura de
                           ;la memoria
    movwf    PORTE
    return

;-----
;
;      Rutina que habilita la lectura a la memoria DS1245Y/AB
;      argumentos: no hay
;
;      nota: vea el diagrama de conexion de la memoria y el pic16f877
;-----
habilitada_lectura
    movlw    b'00000010'    ;habilitada la lectura y deshabilitada la escritura de
                           ;la memoria
    movwf    PORTE
    return

;-----
;
;      Rutina que escribe a la memoria DS1245Y/AB un byte
;      argumentos: no hay
;
;      nota: vea el diagrama de conexion de la memoria y el pic16f877
;-----
escritura
    bsf      STATUS,RP0     ;Selecciona banco 1
    movlw    b'00000000'    ;configuramos el puerto B como salidas
    movwf    TRISB
    bcf      STATUS,RP0     ;Selecciona banco 0

    call     direccionamiento ;direccionamos la memoria
    call     habilitada_escritura ;habilitamos la escritura a la memoria

    call     rx_dato_wait    ;esperamos el dato a escribir en la memoria

    movf     SSPBUF, W
    movwf    PORTB          ;escribimos el dato (PORB es el buffer de datos de la memoria)

    return

;-----
;
;      Rutina que lee de la memoria DS1245Y/AB un byte
;      argumentos: no hay
;
;      nota: vea el diagrama de conexion de la memoria y el pic16f877
;-----
lectura
    bsf      STATUS,RP0     ;Selecciona banco 1
    movlw    b'11111111'    ;configuramos el puerto B como entradas
    movwf    TRISB
    bcf      STATUS,RP0     ;Selecciona banco 0

    call     direccionamiento ;direccionamos la memoria

```

```

call    habilitada_lectura    ;habilitamos la lectura a la memoria

movf    PORTB,W              ;leemos el buffer de la memoria y ponemos el dato en W

movwf   OUT1                 ;ponemos el dato leído en la variable de salida SPI
call    SPIDX                ;se envía por la spi lo que este en OUT1
return

```

```

;-----
;
; Rutina que direcciona una localidad de la memoria DS1245Y/AB
; argumentos: no hay
;
; nota: vea el diagrama de conexion de la memoria y el pic16f877
;-----

```

direccionamiento

;DIRECCIONAMIENTO PARA EL PUERTO A

```

call    rx_dato_wait        ;leemos el bit mas significativos
movf    SSPBUF, W
movwf   MSB

btfss   MSB,0
bcf     PORTA,0             ;direccion A16 correspondiente a PA0
btfsc   MSB,0
bsf     PORTA,0

```

;DIRECCIONAMIENTO PARA EL PUERTO D

```

call    rx_dato_wait        ;leemos el byte medio de direccionamiento
movf    SSPBUF, W
movwf   MEDIO              ;(A15-A7)

```

no_hay_MEDIO

```

btfss   PIR1,SSPIF
goto    no_hay_MEDIO

btfss   MEDIO,7
bcf     PORTD,7             ;direccion A15 correspondiente a PD7
btfsc   MEDIO,7
bsf     PORTD,7

btfss   MEDIO,6
bcf     PORTD,6             ;direccion A14 correspondiente a PD6
btfsc   MEDIO,6
bsf     PORTD,6

btfss   MEDIO,5
bcf     PORTD,5             ;direccion A13 correspondiente a PD5
btfsc   MEDIO,5
bsf     PORTD,5

btfss   MEDIO,4
bcf     PORTD,4             ;direccion A12 correspondiente a PD4
btfsc   MEDIO,4
bsf     PORTD,4

btfss   MEDIO,3
bcf     PORTD,3             ;direccion A11 correspondiente a PD3
btfsc   MEDIO,3
bsf     PORTD,3

btfss   MEDIO,2
bcf     PORTD,2             ;direccion A10 correspondiente a PD2
btfsc   MEDIO,2
bsf     PORTD,2

btfss   MEDIO,1

```

```

        bcf     PORTD,1      ;direccion A9 correspondiente a PD1
        btfsc  MEDIO,1
        bsf     PORTD,1

        btfss  MEDIO,0
        bcf     PORTD,0      ;direccion A8 correspondiente a PD0
        btfsc  MEDIO,0
        bsf     PORTD,0

;DIRECCIONAMIENTO PARA EL PUERTO C Y A
        call   rx_dato_wait
        movf   SSPBUF, W
        movwf  LSB

no_hay_LSB
        btfss  PIR1,SSPIF
        goto   no_hay_LSB

        btfss  LSB,7
        bcf     PORTA,3      ;direccion A7 correspondiente a PA3
        btfsc  LSB,7
        bsf     PORTA,3

        btfss  LSB,6
        bcf     PORTA,2      ;direccion A6 correspondiente a PA2
        btfsc  LSB,6
        bsf     PORTA,2

        btfss  LSB,5
        bcf     PORTA,1      ;direccion A5 correspondiente a PA1
        btfsc  LSB,5
        bsf     PORTA,1

        btfss  LSB,4
        bcf     PORTC,7      ;direccion A4 correspondiente a PC7
        btfsc  LSB,4
        bsf     PORTC,7

        btfss  LSB,3
        bcf     PORTC,6      ;direccion A3 correspondiente a PC6
        btfsc  LSB,3
        bsf     PORTC,6

        btfss  LSB,2
        bcf     PORTC,2      ;direccion A2 correspondiente a PC2
        btfsc  LSB,2
        bsf     PORTC,2

        btfss  LSB,1
        bcf     PORTC,1      ;direccion A1 correspondiente a PC1
        btfsc  LSB,1
        bsf     PORTC,1

        btfss  LSB,0
        bcf     PORTC,0      ;direccion A0 correspondiente a PC0
        btfsc  LSB,0
        bsf     PORTC,0

        return

```

```

;-----
;
; Rutina que espera asta que se a recibido un byte via SPI
; argumentos: no hay
;
;-----

```

```

rx_dato_wait
Rx_Dato2      bsf      STATUS,RP0      ;Banco 1
              btfss     SSPSTAT,BF      ;si BF es uno se ha recibido el dato
              goto     Rx_Dato2 ;no, esperar
              bcf      STATUS,RP0      ;Selecciona banco 0
              return

```

```

;-----
;
; Rutina que envia un byte via SPI
; argumentos: -OUTD1 byte a transmitir
;-----

```

SPIDX:

```

              bsf      STATUS,5        ;banco 1
              bcf      SSPSTAT,0       ;limpiamos la bandera del buffer spi
              bcf      STATUS,5        ;banco 0
              movf     OUT1,W          ;cargamos el la direccion a escribir
              movwf    SSPBUF
TXLOOP1:     bsf      STATUS,5        ;banco 1
              btfss     SSPSTAT,0       ;dato enviado?
              goto     TXLOOP1         ;no
              bcf      STATUS,5        ;banco 0
              return

```

```

;*****
end          ;FIN del codigo fuente

```

ARCHIVOS INC.

constantes_eeprom.inc


```

org          H'2100' ;definimos el contenido por defaul de la eeprom
eeprom1 de 0x00, .40 ;LINCYC
            de 0x00, 0x00 ;AWG
            de 0x00, 0x00 ;BWG
            de 0x00, 0x00 ;CWG
            de 0x00, 0x00 ;AVAG
            de 0x00, 0x00 ;BVAG
            de 0x00, 0x00 ;CVAG
            de 0x00, 0x00, 0x00 ;APHCAL, CPHCAL, CPHCAL
            de 0x00, 0x00 ;AAPOS
            de 0x00, 0x00 ;BAPOS
            de 0x00, 0x00 ;CAPOS
            de 0x00, 0x00 ;CFNUM
            de 0x00, 0x3f ;CFDEN
            de 0x00, 0x00 ;AIRmsOS
            de 0x00, 0x00 ;BIRmsOS
            de 0x00, 0x00 ;CIRmsOS
            de 0x00, 0x00 ;AVRmsOS
            de 0x00, 0x00 ;BVRmsOS
            de 0x00, 0x00 ;CVRmsOS
            de 0x00, 0x00 ;AAPGAIN
            de 0x00, 0x00 ;BAPGAIN
            de 0x00, 0x00 ;CAPGAIN
            de 0x00, 0x00 ;AVGAIN
            de 0x00, 0x00 ;BVGAIN
            de 0x00, 0x00 ;CVGAIN

```

variables.inc

```

LIST
NOLIST
TEMPO1 EQU 0x20 ;Variable contador para la rutinas rutinas de retardo
LAENERGY0 EQU 0x21
LAENERGY1 EQU 0x22
LAENERGY2 EQU 0x23
LVAENERGY0 EQU 0x24
LVAENERGY1 EQU 0x25
LVAENERGY2 EQU 0x26
PERIODO EQU 0x27
PERIOD1 EQU 0x28
LINCYC0 EQU 0x29
LINCYC1 EQU 0x2a
AWG0 EQU 0x2b
AWG1 EQU 0x2c
BWG0 EQU 0x2d
BWG1 EQU 0x2e
CWG0 EQU 0x2f
CWG1 EQU 0x30
APHCAL0 EQU 0x31
BPHCAL0 EQU 0x32
CPHCAL0 EQU 0x33
AAPOS0 EQU 0x34
AAPOS1 EQU 0x35
BAPOS0 EQU 0x36
BAPOS1 EQU 0x37
CAPOS0 EQU 0x38
CAPOS1 EQU 0x39
CFNUM0 EQU 0x3a
CFNUM1 EQU 0x3b
CFDEN0 EQU 0x3c
CFDEN1 EQU 0x3d
AIRMS0 EQU 0x3e
AIRMS1 EQU 0x3f
AIRMS2 EQU 0x40

```

BIRMS0	EQU	0x41	
BIRMS1	EQU	0x42	
BIRMS2	EQU	0x43	
CIRMS0	EQU	0x44	
CIRMS1	EQU	0x45	
CIRMS2	EQU	0x46	
AVRMS0	EQU	0x47	
AVRMS1	EQU	0x48	
AVRMS2	EQU	0x49	
BVRMS0	EQU	0x4a	
BVRMS1	EQU	0x4b	
BVRMS2	EQU	0x4c	
CVRMS0	EQU	0x4d	
CVRMS1	EQU	0x4e	
CVRMS2	EQU	0x4f	
AIRMSOS_0	EQU	0x50	
AIRMSOS_1	EQU	0x51	
BIRMSOS_0	EQU	0x52	
BIRMSOS_1	EQU	0x53	
CIRMSOS_0	EQU	0x54	
CIRMSOS_1	EQU	0x55	
AVRMSOS_0	EQU	0x56	
AVRMSOS_1	EQU	0x57	
BVRMSOS_0	EQU	0x58	
BVRMSOS_1	EQU	0x59	
CVRMSOS_0	EQU	0x5a	
CVRMSOS_1	EQU	0x5b	
AAPGAIN0	EQU	0x5c	
AAPGAIN1	EQU	0x5d	
BAPGAIN0	EQU	0x5e	
BAPGAIN1	EQU	0x5f	
CAPGAIN0	EQU	0x60	
CAPGAIN1	EQU	0x79	
AVGAIN0	EQU	0x61	
AVGAIN1	EQU	0x62	
BVGAIN0	EQU	0x63	
BVGAIN1	EQU	0x64	
CVGAIN0	EQU	0x65	
CVGAIN1	EQU	0x66	
SPI_COMANDO	EQU	0x67	
INSD1	EQU	0x68	
INSD2	EQU	0x69	
INSD3	EQU	0x6a	
LENG_DATOS	EQU	0x6b	
OUTD1	EQU	0x6c	
OUTD2	EQU	0x6d	
AVAG1	EQU	0x6e	
AVAG0	EQU	0x6f	
BVAG1	EQU	0x70	
CVAG0	EQU	0x71	
CVAG1	EQU	0x72	
BVAG0	EQU	0x73	
WDIV0	EQU	0x74	
VADIV0	EQU	0x75	
GUARDAR_REG	EQU	0x76	
INSD4	EQU	0x77	
TIPO_SERVICIO	EQU	0x78	
P_A0	EQU	0x20	;BANCO 1
P_A1	EQU	0x21	;BANCO 1
P_A2	EQU	0x22	;BANCO 1
S_A0	EQU	0x23	;BANCO 1

S_A1	EQU	0x24	;BANCO 1
S_A2	EQU	0x25	;BANCO 1
Q_A0	EQU	0x26	;BANCO 1
Q_A1	EQU	0x27	;BANCO 1
Q_A2	EQU	0x28	;BANCO 1
P_B0	EQU	0x29	;BANCO 1
P_B1	EQU	0x2a	;BANCO 1
P_B2	EQU	0x2b	;BANCO 1
S_B0	EQU	0x2c	;BANCO 1
S_B1	EQU	0x2d	;BANCO 1
S_B2	EQU	0x2e	;BANCO 1
Q_B0	EQU	0x2f	;BANCO 1
Q_B1	EQU	0x30	;BANCO 1
Q_B2	EQU	0x31	;BANCO 1
P_C0	EQU	0x32	;BANCO 1
P_C1	EQU	0x33	;BANCO 1
P_C2	EQU	0x34	;BANCO 1
S_C0	EQU	0x35	;BANCO 1
S_C1	EQU	0x36	;BANCO 1
S_C2	EQU	0x37	;BANCO 1
Q_C0	EQU	0x38	;BANCO 1
Q_C1	EQU	0x39	;BANCO 1
Q_C2	EQU	0x3a	;BANCO 1
P_A_PROM_0	EQU	0x3b	;BANCO 1
P_A_PROM_1	EQU	0x3c	;BANCO 1
P_A_PROM_2	EQU	0x3d	;BANCO 1
P_B_PROM_0	EQU	0x3e	;BANCO 1
P_B_PROM_1	EQU	0x3f	;BANCO 1
P_B_PROM_2	EQU	0x40	;BANCO 1
P_C_PROM_0	EQU	0x41	;BANCO 1
P_C_PROM_1	EQU	0x42	;BANCO 1
P_C_PROM_2	EQU	0x43	;BANCO 1
CARRY	EQU	0x44	;BANCO 1
S_A_PROM_0	EQU	0x45	;BANCO 1
S_A_PROM_1	EQU	0x46	;BANCO 1
S_A_PROM_2	EQU	0x47	;BANCO 1
S_B_PROM_0	EQU	0x48	;BANCO 1
S_B_PROM_1	EQU	0x49	;BANCO 1
S_B_PROM_2	EQU	0x4a	;BANCO 1
S_C_PROM_0	EQU	0x4b	;BANCO 1
S_C_PROM_1	EQU	0x4c	;BANCO 1
S_C_PROM_2	EQU	0x4d	;BANCO 1
Q_A_PROM_0	EQU	0x4e	;BANCO 1
Q_A_PROM_1	EQU	0x4f	;BANCO 1
Q_A_PROM_2	EQU	0x50	;BANCO 1
Q_B_PROM_0	EQU	0x51	;BANCO 1
Q_B_PROM_1	EQU	0x52	;BANCO 1
Q_B_PROM_2	EQU	0x53	;BANCO 1
Q_C_PROM_0	EQU	0x54	;BANCO 1
Q_C_PROM_1	EQU	0x55	;BANCO 1

```

Q_C_PROM_2    EQU            0x56    ;BANCO 1
DIV           EQU            0x57    ;BANCO 1 se usa en la rutina de div.

```

```

;variables a utilizar despues de la calibracion desde 0x27 a 0x3d

```

```

INT_DEL_RELOJ    EQU            0x27    ;BANCO 0

MSB             equ            0x28
MEDIO           equ            0x29
LSB            equ            0x2a

CONTADOR_LSB    equ            0x2b
CONTADOR_MEDIO equ            0x2c
CONTADOR_MSB    equ            0x2d

CONTADOR_LSB_LECTURA equ    0x2e
CONTADOR_MEDIO_LECTURA equ    0x2f
CONTADOR_MSB_LECTURA equ    0x30

DATO           equ            0x31

TIEMP1        equ            0x35
TIEMP2        equ            0x36
TIEMP3        equ            0x37
TIEMP4        equ            0x38

ADDR          EQU            0x20    ;BANCO 2 registro usado en el prog.
VALOR         EQU            0x21    ;BANCO 2 registro usado en el prog.

P_TRIFASICA_0 equ            0x22    ;BANCO 2
P_TRIFASICA_1 equ            0x23    ;BANCO 2
P_TRIFASICA_2 equ            0x24    ;BANCO 2

S_TRIFASICA_0 equ            0x25    ;BANCO 2
S_TRIFASICA_1 equ            0x26    ;BANCO 2
S_TRIFASICA_2 equ            0x27    ;BANCO 2

Q_TRIFASICA_0 equ            0x28    ;BANCO 2
Q_TRIFASICA_1 equ            0x29    ;BANCO 2
Q_TRIFASICA_2 equ            0x2a    ;BANCO 2

```

```

LIST

```

reg_ade7754.inc

```

LIST
; reg_ade7754.INC archivo de los reg. del ADE7754, Version 0.00
NOLIST

```

```

AENERGY        EQU 0x01
RAENERGY        EQU 0x02
LAENERGY        EQU 0x03
VAENERGY        EQU 0x04
RVAENERGY       EQU 0x05
LVAENERGY       EQU 0x06
PERIO           EQU 0x07
TEMP            EQU 0x08
WFORM           EQU 0x09
OPMODE_R        EQU 0x0A
OPMODE_W        EQU 0x8A
MMODE_R          EQU 0x0B
MMODE_W          EQU 0x8B
WAVMODE_R       EQU 0x0C
WAVMODE_W       EQU 0x8C
WATMODE_R       EQU 0x0D

```

WATMODE_W	EQU 0x8D
VAMODE_R	EQU 0x0E
VAMODE_W	EQU 0x8E
IRQEN_R	EQU 0x0F
IRQEN_W	EQU 0x8F
STATUS_R	EQU 0x10
STATUS_W	EQU 0x90
RSTATUS	EQU 0x11
ZXTOUT_R	EQU 0x12
ZXTOUT_W	EQU 0x92
LINCYC_R	EQU 0x13
LINCYC_W	EQU 0x93
SAGCYC_R	EQU 0x14
SAGCYC_W	EQU 0x94
SAGLVL_R	EQU 0x15
SAGLVL_W	EQU 0x95
VPEAK_R	EQU 0x16
VPEAK_W	EQU 0x96
IPEAK_R	EQU 0x17
IPEAK_W	EQU 0x97
GAIN_R	EQU 0x18
GAIN_W	EQU 0x98
AWG_R	EQU 0x19
AWG_W	EQU 0x99
BWG_R	EQU 0x1A
BWG_W	EQU 0x9A
CWG_R	EQU 0x1B
CWG_W	EQU 0x9B
AVAG_R	EQU 0x1C
AVAG_W	EQU 0x9C
BVAG_R	EQU 0x1D
BVAG_W	EQU 0x9D
CVAG_R	EQU 0x1E
CVAG_W	EQU 0x9E
APHCAL_R	EQU 0x1F
APHCAL_W	EQU 0x9F
BPHCAL_R	EQU 0x20
BPHCAL_W	EQU 0xA0
CPHCAL_R	EQU 0x21
CPHCAL_W	EQU 0xA1
AAPOS_R	EQU 0x22
AAPOS_W	EQU 0xA2
BAPOS_R	EQU 0x23
BAPOS_W	EQU 0xA3
CAPOS_R	EQU 0x24
CAPOS_W	EQU 0xA4
CFNUM_R	EQU 0x25
CFNUM_W	EQU 0xA5
CFDEN_R	EQU 0x26
CFDEN_W	EQU 0xA6
WDIV_R	EQU 0x27
WDIV_W	EQU 0xA7
VADIV_R	EQU 0x28
VADIV_W	EQU 0xA8
AIRMS	EQU 0x29
BIRMS	EQU 0x2A
CIRMS	EQU 0x2B
AVRMS	EQU 0x2C
BVRMS	EQU 0x2D
CVRMS	EQU 0x2E
AIRMSOS_R	EQU 0x2F
AIRMSOS_W	EQU 0xAF
BIRMSOS_R	EQU 0x30
BIRMSOS_W	EQU 0xB0
CIRMSOS_R	EQU 0x31
CIRMSOS_W	EQU 0xB1
AVRMSOS_R	EQU 0x32

```

AVRMSOS_W EQU 0xB2
BVRMSOS_R EQU 0x33
BVRMSOS_W EQU 0xB3
CVRMSOS_R EQU 0x34
CVRMSOS_W EQU 0xB4
AAPGAIN_R EQU 0x35
AAPGAIN_W EQU 0xB5
BAPGAIN_R EQU 0x36
BAPGAIN_W EQU 0xB6
CAPGAIN_R EQU 0x37
CAPGAIN_W EQU 0xB7
AVGAIN_R EQU 0x38
AVGAIN_W EQU 0xB8
BVGAIN_R EQU 0x39
BVGAIN_W EQU 0xB9
CVGAIN_R EQU 0x3A
CVGAIN_W EQU 0xBA
CHKSUM EQU 0x3E
VERSION EQU 0x3F

```

LIST

macros.mac

```

;***** definiconde macros *****

```

```

flash_led macro reg,bit ;cambia el estado de un pin cualquier de un puerto

```

```

    btfsc reg,bit
    goto $+3
    bsf reg,bit
    goto $+2
    bcf reg,bit
endm

```

```

;-----

```

```

retardo_x macro retardo ;retardo=2+(1+2)*retardo+1=? ciclos de reloj

```

```

    movlw retardo
    movwf TEMPO1
    decfsz TEMPO1,F
    goto $-1
endm

```

```

;-----

```

```

set_spi_reloj macro
    bsf PORTA,0x01 ;desactivamos la spi del ADE
    bsf PORTA,0x05 ;desactivamos la spi de la memoria
    bcf PORTA,0x00 ;activamos la spi del reloj
endm

```

```

;-----

```

```

set_spi_ade7754 macro
    bsf PORTA,0x05 ;desactivamos la spi de la memoria
    bsf PORTA,0x00 ;desactivamos la spi del reloj
    bcf PORTA,0x01 ;activamos la spi del ADE
endm

```

```

;-----

```

```

set_spi_memoria macro
    bsf PORTA,0x01 ;desactivamos la spi del ADE
    bsf PORTA,0x00 ;desactivamos la spi del reloj
    bcf PORTA,0x05 ;activamos la spi de la memoria
endm

```

```

;-----

```

```

page_0 macro
    bcf PCLATH,0x04
    bcf PCLATH,0x03
endm

```

```

;-----

```

```
page_1 macro
    bcf    PCLATH,0x04
    bcf    PCLATH,0x03
endm
```

```
;-----
page_2 macro
    bsf    PCLATH,0x04
    bcf    PCLATH,0x03
endm
```

```
;-----
page_3 macro
    bsf    PCLATH,0x04
    bcf    PCLATH,0x03
endm
```

```
;-----
```