



Universidad Tecnológica de la Mixteca

“Diseño y construcción de un módulo de almacenamiento para el medidor de variables eléctricas PowerLogic PM650 basado en el microcontrolador 8031”

TESIS
para obtener el título profesional de:
INGENIERO EN ELECTRÓNICA

PRESENTA: Zhoraya López Villegas

:

ASESOR: M. C. Fermín Hugo Ramírez Leyva

RESUMEN

Este trabajo tiene como finalidad el diseño y construcción de un módulo de almacenamiento programable, el cual recolecta la información adquirida por el medidor de potencia trifásico marca *PowerLogic* modelo PM650 a través de una interfaz RS-485 y el protocolo de comunicaciones MODBUS. Las variables eléctricas a almacenar y el tiempo de muestreo son configurables a través de la interfaz de usuario del módulo. La información contenida en el modulo de almacenamiento se descarga hacia una computadora personal, mediante un programa desarrollado con el software de programación gráfica LabView de *National Instruments*. Este programa de descarga tiene la capacidad de graficar las variables eléctricas adquiridas y guardar la información para su posterior procesamiento.

A mis abuelos y mis padres:

Felipa y Magdaleno, Esther e Ignacio, Alba y Zaalathiel.

Gracias.

AGRADECIMIENTOS

Al M. C. Hugo F. Ramírez Leyva, asesor de esta tesis, por su apoyo, confianza, conocimientos compartidos y guía para la realización de este trabajo.

A los profesores que fueron los sinodales de mi tesis: C. M. C. Esteban Guerrero Ramírez, C. M. C. Enrique Guzmán Ramírez y C. Dr. Heriberto Hernández Martínez por sus valiosos consejos y el tiempo que dedicaron en la revisión de este documento.

A las autoridades y personal de la Universidad Tecnológica de la Mixteca por brindarme el apoyo necesario para realizar esta tesis en las instalaciones de la misma.

A los maestros que participaron en mi formación académica, en especial a los siguientes profesores: Gustavo Jiménez Santana, Hugo Suárez Onofre, José Antonio Moreno Espinoza, Raúl Cruz Barbosa, Jorge de Dios Domínguez y Vitaly Ryback, mi agradecimiento y admiración.

A Efraín por su amor, consejos y apoyo. A los amigos y compañeros que con sus palabras de aliento no me dejaron detenerme en esta carrera.

A mi familia: López Villegas, Villegas Córdova, Villegas Cruz, Vázquez Niño y Vázquez López Bautista.

CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1. MARCO TEÓRICO	5
1.1. LA ENERGÍA ELÉCTRICA	5
1.1.1. Parámetros de la energía eléctrica.....	6
1.1.2. Costo de la energía	8
1.1.3. Mediciones eléctricas	10
1.1.3.1. Medidores de potencia digitales	12
1.1.3.2. Medidor PM650	12
1.2. COMUNICACIONES SERIALES	13
1.2.1. Estándar RS-232	13
1.2.1.1. Convertidor ADM233	14
1.2.2. Estándar RS-485	15
1.2.2.1. Convertidor MC3486	17
1.2.2.2. Convertidor MC3487	17
1.3. PROTOCOLO MODBUS	18
1.3.1. Código CRC	19
1.3.2. Función de lectura de registros.....	20
1.4. LABVIEW	22
1.4.1. Instrumentos Virtuales	22
CAPÍTULO 2. DESCRIPCIÓN DEL MÓDULO DE ALMACENAMIENTO.....	25
2.1. DESCRIPCIÓN GENERAL.....	25
2.2. MEDIDOR PM650	26
2.2.1. Parámetros considerados para el almacenamiento.....	26
2.3. MÓDULO DE ALMACENAMIENTO.....	27
2.3.1. Submódulo de control.....	28
2.3.1.1. Respaldo de la información almacenada.....	31
2.3.2. Submódulo de interfaz de usuario.....	31
2.3.2.1. Teclado.....	32
2.3.2.2. Pantalla	34
2.3.3. Submódulo de comunicaciones seriales.....	36
CAPÍTULO 3. DESARROLLO DEL SOFTWARE	39
3.1. DESCRIPCIÓN GENERAL.....	39
3.2. PROGRAMA DEL MÓDULO DE ALMACENAMIENTO	40

3.2.1. Inicialización del módulo	41
3.2.2. Determinación de la velocidad de transmisión.....	42
3.2.3. Envío de paquete de petición.....	44
3.2.4. Verificación de memoria de almacenamiento.....	44
3.2.5. Petición de opción	45
3.2.5.1. Configuración del módulo	45
3.2.5.2. Inicio de almacenamiento de parámetros eléctricos.....	46
3.2.5.3. Envío de datos a la PC	47
3.3. PROGRAMA DE DESCARGA DE DATOS	47
3.3.1. Algoritmo del programa de descarga	48
3.3.2. inf_config.vi.....	50
3.3.3. indices.vi	50
3.3.4. archivo de salida.vi.....	50
3.3.5. modulo almacenamiento.vi.....	51
CAPÍTULO 4. RESULTADOS Y CONCLUSIONES	55
4.1. RESULTADOS	55
4.1.1. Prototipo del módulo de almacenamiento	55
4.1.2. Prueba de funcionamiento del módulo de almacenamiento	56
4.1.2.1. Ejemplo 1	59
4.1.2.2. Ejemplo 2	64
4.2. CONCLUSIONES	69
4.3. PERSPECTIVAS	70
BIBLIOGRAFÍA.....	73
ANEXO A. Tarifas para el suministro y venta de energía eléctrica (2001 - 2002) *	A-1
TARIFAS ESPECÍFICAS	A-1
TARIFAS GENERALES	A-1
TIPOS DE MODELOS DE RECIBOS DE LUZ	A-1
RECIBO DE LUZ PARA BAJA TENSIÓN, ALUMBRADO PÚBLICO, BOMBEO DE AGUA Y RIEGO AGRÍCOLA	A-2
Información del consumo	A-3
Cálculo del importe del consumo	A-3
Detalle de los cargos y créditos que integran la facturación	A-4
RECIBO DE LUZ PARA BAJA Y MEDIA TENSIÓN	A-5
Información del consumo	A-5
Cálculo del importe del consumo	A-6
Detalle de los cargos y créditos que integran la facturación	A-7

RECIBO DE LUZ PARA HORARIOS EN MEDIA Y ALTA TENSIÓN	A-7
Información del consumo	A-8
Cálculo del importe del consumo	A-8
Detalle de los cargos y créditos que integran la facturación	A-9
ANEXO B. Microcontrolador 8031	B-1
ANEXO C. Esquemático del módulo de almacenamiento	C-1
ANEXO D. Manual de usuario del módulo de almacenamiento	D-1
GENERALIDADES	D-1
Definición.....	D-1
Características	D-1
Conexiones.....	D-1
ALMACENAMIENTO DE LOS DATOS DEL PM650.....	D-2
Instalación del medidor PM650	D-2
Estado de funcionamiento del módulo de almacenamiento	D-2
Inicialización	D-2
Configuración.....	D-3
Inicia muestreo.....	D-6
Envía datos PC	D-7
Reinicia	D-8
DESCARGA DE DATOS.....	D-8
Ejecución del programa de descarga.....	D-8
Adquisición de los datos.....	D-9
Presentación de los datos	D-9
ERRORES COMUNES	D-10
Errores en el módulo de almacenamiento.....	D-10
Errores en el software de descarga.....	D-11
ANEXO E. Listado de programas	E-1
ANEXO E.1. Programa en lenguaje opal para decodificar la memoria y seleccionar el estándar de comunicaciones	E-1
ANEXO E.2. Programa en lenguaje opal para mapeo en memoria RAM del teclado y display.....	E-3
ANEXO E.3. Programa del módulo de almacenamiento.....	E-5

LISTA DE FIGURAS

Figura 1.1. Planta de energía de ciclo combinado. Cortesía, CFE.	6
Figura 1.2. Representación de la forma de onda de la señal eléctrica.	7
Figura 1.3. Amperímetro de gancho. Cortesía Wavetek Co.	10
Figura 1.4. Voltímetro. Cortesía Fluke Co.	11
Figura 1.5. Wattímetro. Cortesía Extech Co.	11
Figura 1.6. Analizador de redes. Cortesía Yokogawa Co.	11
Figura 1.7. Medidor PM650 Powerlogic con visualizador. Cortesía PowerLogic.	12
Figura 1.8. Conector macho DB9.	14
Figura 1.9. Diagrama lógico del ADM233 y sus conexiones.	15
Figura 1.10. Conexiones para el conector DB9.	16
Figura 1.11. Diagrama lógico del MC3486.	17
Figura 1.12. Diagrama lógico del MC3487.	18
Figura 1.13. Diagrama de flujo del algoritmo para calcular el código CRC.	20
Figura 1.14. Panel de control de <i>modulo almacenamiento.vi</i>	23
Figura 1.15. Diagrama de bloques de <i>inf_config.vi</i>	24
Figura 1.16. Icono y conectores del subVI: <i>inf_config.vi</i>	24
Figura 2.1. Diagrama a bloques del medido PM650 y el módulo de almacenamiento.	25
Figura 2.2. Medidor PM650 PowerLogic, cortesía SquaredD.	26
Figura 2.3. Diagrama a bloques del módulo de almacenamiento.	28
Figura 2.4. Distribución de la memoria externa del sistema mínimo del 8031.	29
Figura 2.5. Conexiones de la memoria externa del módulo de almacenamiento.	30
Figura 2.6. Diagrama lógico del ADM691.	31
Figura 2.7. Conexiones del submódulo de interfaz de usuario.	32
Figura 2.8. Diagrama lógico del MM74C923.	33
Figura 2.9. Diagrama de tiempos del MM74C923.	33
Figura 2.10. Conexiones del MM74C923.	34
Figura 2.11. Diagrama a bloques del AND491. Cortesía Purdy Electronics.	35
Figura 2.12. Conexiones del AND491.	36
Figura 2.13. Conexiones de las comunicaciones seriales del módulo de almacenamiento.	36
Figura 3.1 Diagrama a bloques del programa.	39
Figura 3.2. Diagrama a bloques del programa del módulo de almacenamiento.	40
Figura 3.3. Diagrama de flujo del algoritmo de determinación de velocidad de transmisión.	43
Figura 3.4 Diagrama de flujo del algoritmo de envío de paquete de petición de lectura.	44
Figura 3.5. Algoritmo de la rutina de inicio de almacenamiento.	46
Figura 3.6. Diagrama de flujo del programa de descarga.	49

Figura 3.7. Icono y conectores del subVI <i>inf_cfg.vi</i>	50
Figura 3.8. Icono y conectores del subVI <i>indices.vi</i>	50
Figura 3.9. Icono y conectores del subVI <i>archivo de salida.vi</i>	51
Figura 3.10. Icono del VI <i>modulo almacenamiento.vi</i>	51
Figura 3.11. Componentes del panel frontal del programa de descarga.	53
Figura 4.1. Prototipo del módulo de almacenamiento y sus componentes.....	55
Figura 4.2. Diagrama unifilar de la red eléctrica de la UTM.....	56
Figura 4.3. Módulo de almacenamiento instalado en la red eléctrica de la UTM.	57
Figura 4.4. Conexión de los voltajes y los transformadores de corriente.....	57
Figura 4.5. Conexión de comunicaciones RS485 del módulo de almacenamiento a las del PM650.....	58
Figura 4.6. Mensaje durante el almacenamiento en un tiempo automático y no automático.	58
Figura 4.7. Gráfica de los valores de frecuencia del ejemplo 1.	61
Figura 4.8. Gráfica de los valores de la corriente monofásica del ejemplo 1.	62
Figura 4.9. Gráfica del voltaje monofásico del ejemplo 1.....	62
Figura 4.10. Gráfica de los valores absolutos del factor de potencia del ejemplo 1.....	63
Figura 4.11. Gráfica de los valores con signo del factor de potencia del ejemplo 1.	63
Figura 4.12. Gráfica de los valores de distorsión armónica del voltaje del ejemplo 1.....	64
Figura 4.13. Gráfica de los valores de frecuencia del ejemplo 2.	66
Figura 4.14. Gráfica de los valores de corriente monofásica del ejemplo 2.....	67
Figura 4.15. Gráfica de los valores de voltaje monofásico del ejemplo 2.....	67
Figura 4.16. Gráfica de los valores absolutos del factor de potencia del ejemplo 2.....	68
Figura 4.17. Gráfica de los valores con signo del factor de potencia del ejemplo 2.	68
Figura 4.18. Gráfica de los valores de distorsión armónica del voltaje del ejemplo 2.....	69
Figura D.1. Conexiones de módulo de almacenamiento.....	D-1
Figura D.2. Cuadro de diálogo para el almacenamiento de datos.....	D-9
Figura D.3. Panel frontal del programa de descarga.....	D-10

LISTA DE TABLAS

Tabla 1.1. Tarifas de costo de energía eléctrica de CFE.	9
Tabla 1.2. Descripción de los pines del DB9 según el RS-232.	14
Tabla 1.3. Código de algunas funciones del protocolo Modbus.....	19
Tabla 1.4. Campos de la función de lectura de registros.....	20
Tabla 1.5. Campos del paquete de respuesta a la función de lectura de registros.....	21
Tabla 2.1. Parámetros considerados por el módulo de almacenamiento.....	27
Tabla 2.2. Señales empleadas en el direccionamiento de memoria del módulo de almacenamiento.....	29
Tabla 2.3. Distribución y asignación de memoria externa.	30
Tabla 2.4. Valores de los pines de salida del MM74C923.	34
Tabla 2.5. Descripción de las señales de control del AND491.	35
Tabla 3.1. Variables principales del programa del módulo de almacenamiento.....	42
Tabla 3.2 Descripción de los bloques de mediciones.	45
Tabla 3.3. Opciones de gráficas del programa de descarga.	52
Tabla 4.1. Archivo que almacena los datos de la medición del ejemplo 1.	60
Tabla 4.2. Archivo que almacena los datos de la segunda medición.	65
Tabla B.1 Integrantes de la familia MCS-51 y sus principales características.....	B-1

INTRODUCCIÓN

El desarrollo y evolución de la sociedad humana siempre ha estado ligado al uso de energía proveniente de diversas fuentes, la humanidad ha utilizado los recursos naturales como fuente de energía para la elaboración de bienes y servicios que satisfacen sus necesidades elementales. La energía de mayor uso en la actualidad es, sin duda, la energía eléctrica, la cual es generada principalmente usando hidrocarburos a pesar de ser fomentadas e impulsadas otras formas de generación (eólica, solar, nuclear, geotérmica, etc.), éstas no alcanzan porcentajes considerables frente al uso de hidrocarburos. Esta proporción hace inminente la búsqueda de nuevas formas de generación de energía, el desarrollo de las existentes, el fortalecimiento de una cultura de ahorro y el aprovechamiento racional de la energía disponible.

Un suceso determinante para el fomento de la cultura de ahorro de energía ocurrió en el año 1973 cuando, a partir de un enfrentamiento militar entre Egipto e Israel, el precio del petróleo se disparó provocando una crisis económica mundial, mostrando la dependencia de la economía mundial al precio del crudo. A partir de ese año países como Estados Unidos de América (EUA) desarrollaron y fomentaron la idea de una administración orientada a la demanda (DMS, *Demand-Side Management*), la cual tiene como fundamento la lógica económica de que es más barato invertir en la modificación de los consumos de los usuarios que construir y operar plantas nuevas de generación eléctrica.

En el panorama nacional, en 1990 se fundó el Fideicomiso para el Ahorro de Energía Eléctrica (FIDE) con el objeto de impulsar programas de ahorro de energía eléctrica. Además de este organismo se han creado normas que establecen formas de apoyo a empresas que ahorren y utilicen de manera eficiente la energía eléctrica suministrada por la Comisión Federal de Electricidad (CFE), organismo gubernamental que genera, distribuye y administra la energía eléctrica en México. Para aprovechar el apoyo otorgado por el gobierno mexicano a quien menos consume o mejor utiliza la electricidad, los consumidores de energía eléctrica, empresas principalmente, realizan análisis del comportamiento de su red eléctrica basados en historiales de los valores de las variables eléctricas más importantes para implementar medidas de ahorro de energía y programas de control de la demanda eléctrica.

Entre los instrumentos de medición de las variables eléctricas que se emplean en los análisis eléctricos se encuentra el medidor de potencia o analizador de redes eléctricas que tiene en general las siguientes características:

- Adquisición de manera instantánea de los valores de la corriente, el voltaje y la potencia eléctrica.
- Almacenamiento de la información obtenida en forma de historiales de mediciones.
- Impresión de los datos medidos en períodos preestablecidos.
- Programación de los intervalos de recolección de datos en períodos mayores y menores a 15 minutos.

En el mercado internacional y nacional existen varias compañías que fabrican y comercializan diversos modelos de medidores de potencia; su costo depende, de manera directamente proporcional, a las características presentes. A pesar de los beneficios económicos a largo plazo que genera un análisis eléctrico, la compra de un medidor de potencia, con todas las características listadas anteriormente, implica el desembolso de una cantidad considerable de dinero, lo cual no representa un ahorro económico a corto plazo.

La Universidad Tecnológica de la Mixteca (UTM) cuenta con un medidor de variables eléctricas; el medidor marca PowerLogic modelo PM650, con el que se realizan mediciones para el análisis del comportamiento de la red eléctrica de la universidad. Aunque este medidor puede medir hasta 31 variables eléctricas, registrar los valores de las mediciones instantáneamente y desplegar los valores en su pantalla, su utilidad se ve disminuida al carecer de una memoria que almacene los datos obtenidos en un período determinado de tiempo, debido a lo cual las mediciones deben ser anotadas manualmente y la calidad de la información obtenida disminuye.

El presente documento describe la implementación de un módulo de almacenamiento de parámetros eléctricos obtenidos por el medidor PM650, tanto el hardware como el software, además de la descripción de los principales resultados obtenidos durante el desarrollo de este trabajo.

OBJETIVOS

Objetivo general:

- Diseñar y construir un módulo de almacenamiento para el medidor de variables eléctricas PowerLogic PM650 basado en el microcontrolador 8031.

Objetivos particulares:

- Determinar la memoria mínima requerida para almacenar hasta 7 días en intervalos de 5 minutos las variables eléctricas más importantes en el análisis eléctrico.

- Realizar una investigación bibliográfica del estándar y protocolo de comunicaciones utilizados por el medidor PM650.
- Diseñar el sistema de comunicaciones con las especificaciones de comunicaciones del medidor PM650.
- Integrar las siguientes características al módulo de almacenamiento:
 - Configuración del número de variables a almacenar.
 - Elección del tiempo de muestreo.
 - Indicador de inicio de la recolección de los datos.
 - Descarga de los datos hacia la PC.
- Implementar un programa que obtenga, almacene y grafique en la PC los datos almacenados en el módulo de almacenamiento.
- Evaluar los resultados obtenidos a partir del funcionamiento del módulo de almacenamiento conectado al medidor PM650 y éste a su vez a una red eléctrica.

Descripción del contenido

El presente trabajo está dividido en cuatro capítulos con el siguiente contenido:

En el capítulo uno se definen las características, los parámetros y el costo de la energía eléctrica. Se explica la importancia de las mediciones de parámetros de las redes eléctricas, los instrumentos utilizados y en particular se describe el medidor de variables eléctricas PowerLogic PM650. Se describen las comunicaciones seriales, el estándar RS-485, el estándar RS-232 y los convertidores de niveles de voltaje utilizados en el módulo de almacenamiento. Además, se explica el protocolo de comunicaciones Modbus, utilizado por el medidor PM650 y se resume el uso y descripción del lenguaje LabView de National Instruments.

En el capítulo dos se listan y explican las características del módulo de almacenamiento correspondientes al diseño e implementación de los submódulos que lo integran.

El capítulo tres se expone la implementación del programa utilizado por el módulo de almacenamiento y el programa de descarga de los datos implementado en LabView.

El capítulo cuatro incluye los resultados y conclusiones obtenidos con el desarrollo del presente trabajo y las perspectivas del mismo.

CAPÍTULO 1. MARCO TEÓRICO

1.1. LA ENERGÍA ELÉCTRICA

La palabra *energía* denota movimiento, acción o capacidad de realizar una actividad; la energía eléctrica es aquella que permite que la mayoría de las máquinas que se utilizan cotidianamente realicen las funciones para las cuales fueron construidas. Este tipo de energía se encuentra presente en la Naturaleza, en fenómenos como las descargas atmosféricas conocidas como rayos, pero ésta energía no puede ser aprovechada y es necesario emplear otros medios para producirla y utilizarla en beneficio.

En el año de 1819, el físico danés Hans Oersted descubrió el principio para obtener energía eléctrica, el movimiento de una serie de espiras de cobre en el seno del campo magnético, producido por un imán, genera una diferencia de potencial en las terminales de la bobina que al final de cuentas es energía eléctrica que puede ser utilizada después de ciertas modificaciones. Al conjunto que forman el campo magnético y la bobina se denomina generador, el cual transforma la energía mecánica en energía eléctrica. Esta energía mecánica es obtenida a partir de diferentes fuentes: el movimiento de agua, el calor, la geotermia, la energía nuclear y las energías renovables como la eólica y de la biomasa (leña, carbón basura y rastrojos del campo).

En México el 75% de la electricidad se genera usando hidrocarburos, los cuales producen calor y vapor que mueven los generadores contenidos en las plantas o centrales termoeléctricas [URL 1]. La figura 1.1 muestra un diagrama de una planta generadora.

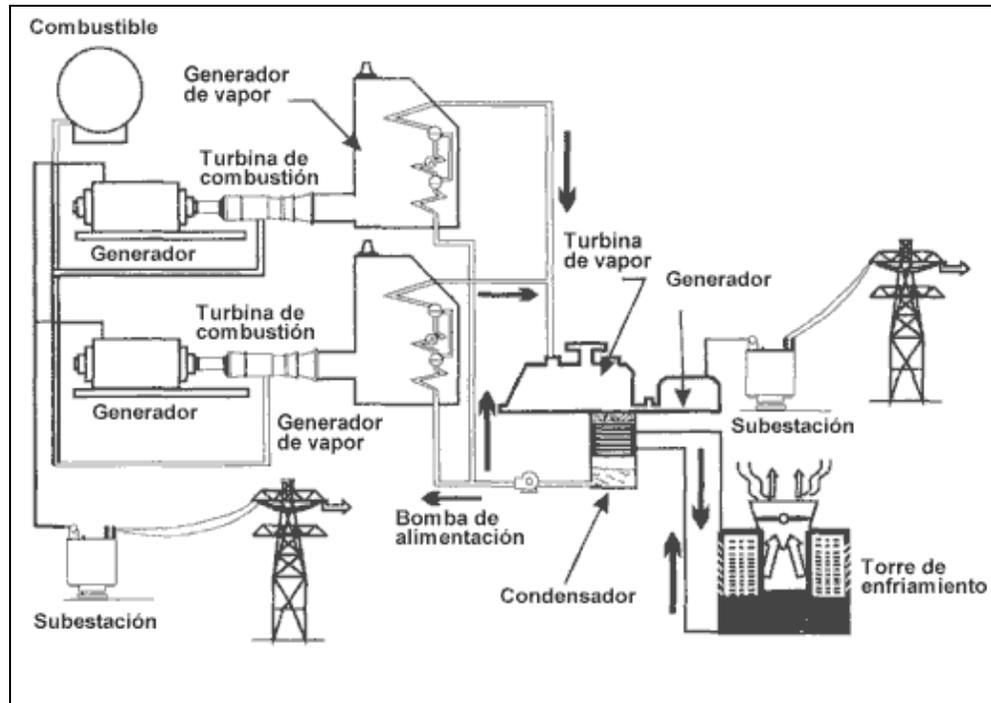


Figura 1.1. Planta de energía de ciclo combinado. Cortesía, CFE.

La electricidad producida en un generador alcanza unos 25,000 voltios (unidades de diferencia de potencial eléctrico). En la planta de energía el voltaje es elevado a 400,000 voltios para que la electricidad pueda viajar largas distancias a través de cables de alta tensión y después mediante transformadores se reduce el nivel del voltaje para ser utilizado en hogares o industrias.

En México, la energía eléctrica que se consume en los hogares tiene, aproximadamente, el valor de 120 voltios RMS (voltaje eficaz); en la industria y en otras actividades se emplean 220 voltios RMS e incluso voltajes superiores para mover maquinaria y grandes equipos. En países europeos lo normal es el uso de 220 voltios RMS para todos los aparatos eléctricos.

1.1.1. Parámetros de la energía eléctrica

La energía eléctrica se puede describir mediante sus parámetros más importantes, los cuales son indispensables para obtener un análisis del comportamiento eléctrico de un equipo, sistema o instalación. Las mediciones de los parámetros relacionados al consumo de energía (potencia activa y reactiva) se consideran indispensables para fines de un diagnóstico energético, mientras que los datos de intensidad de corriente y tensión aportan información complementaria sobre el comportamiento de la instalación y la calidad del suministro.

La energía eléctrica es representada como una onda senoidal con $120\sqrt{2}$ voltios de amplitud y 60 Hertz⁽¹⁾ de frecuencia. La forma de onda de la señal de la energía eléctrica,

⁽¹⁾ Hz., unidades ciclo por segundo

mostrada en la figura 1.2, revela que el valor de la onda del voltaje se repite en el sentido negativo de π hasta 2π y tiene el mismo valor máximo de $120\sqrt{2}$ voltios de amplitud, arriba y abajo del eje de los grados de rotación del generador.

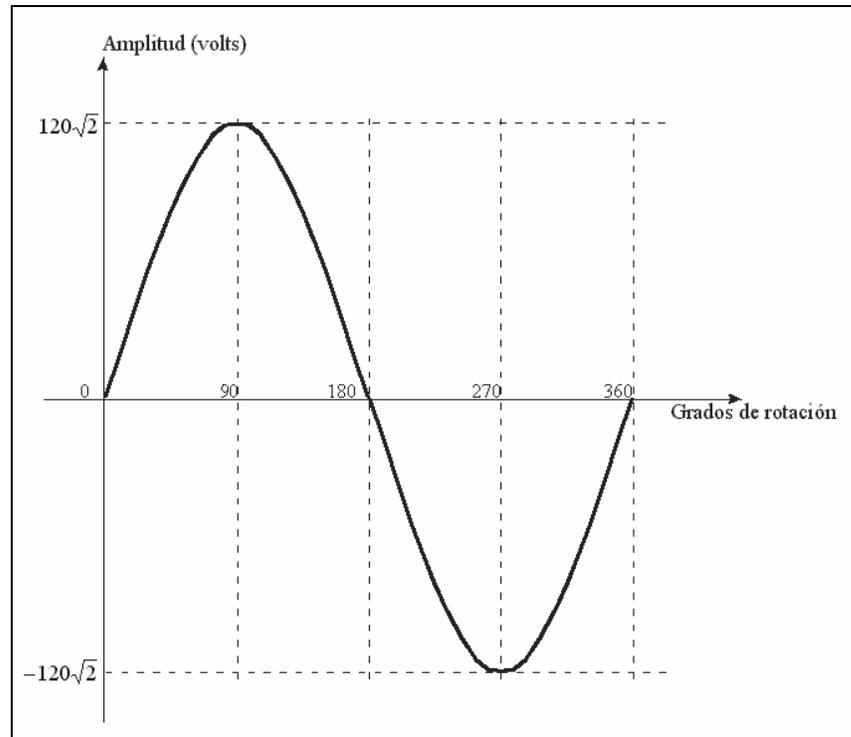


Figura 1.2. Representación de la forma de onda de la señal eléctrica.

El comportamiento del voltaje alterno se debe a la forma en que es generado; durante una revolución completa de la armadura del generador ocurre lo siguiente: el voltaje comienza en cero cuando la armadura no corta líneas magnéticas de fuerza, al girar la armadura, la tensión aumenta desde cero hasta un valor máximo en una dirección, luego disminuye otra vez hasta cero, en este punto el voltaje cambia de polaridad y aumenta hasta que llega a un máximo con ésta polaridad opuesta, para disminuir nuevamente a cero completando una revolución.

Además del voltaje y corriente, existen otros parámetros con los cuales se describe a la energía eléctrica. La frecuencia θ de la señal eléctrica es comúnmente de 60 Hz., la cual indica que se presentaron 60 ciclos de la señal en un segundo. La mayor parte de la energía eléctrica generada en América tiene una frecuencia de 60 Hz., en Europa y Sudamérica emplean energía eléctrica con una frecuencia de 50 Hz. [1].

En su forma matemática más simple, la ecuación de voltaje senoidal es:

$$V = V_{m\acute{a}x} \text{sen } \theta \quad (1.1)$$

donde:

V Es el valor del voltaje para θ grados de rotación del generador.

$V_{m\acute{a}x}$ Es el voltaje de pico de la onda senoidal, $120\sqrt{2}$ voltios.

- q Instante en un ciclo, de 0° a 360° .
 $q = \omega t$, donde ω es la fase de la señal y t el tiempo.

Potencia y factor de potencia son parámetros eléctricos considerados para la facturación de la energía por las empresas encargadas de distribución de la misma. El factor de potencia indica el rendimiento con que se utiliza la capacidad eléctrica y los contratos de tarifas de consumo de energía eléctrica pueden incluir términos referentes a recargos monetarios según los niveles del factor de potencia.

Si las ondas de voltaje y de corriente de un motor eléctrico se grafican, se puede apreciar que la onda de corriente está desfasada un ángulo ϕ con respecto a la onda de voltaje. El factor de potencia se define como el coseno del ángulo ϕ , siempre que no exista distorsión de onda. El factor de potencia es un valor numérico, en un rango de 0 a 1 (valor absoluto), que indica la relación entre la potencia activa y la potencia aparente. Los valores de factor de potencia pueden ser positivos o negativos, si es un valor negativo indica que la corriente se encuentra atrasada respecto al voltaje, la potencia reactiva es de origen inductivo; si es un valor positivo expresa que la corriente está adelantada respecto al voltaje, la potencia reactiva es de origen capacitivo. El factor de potencia puede definirse brevemente como un indicador del correcto aprovechamiento de la energía eléctrica [2].

La potencia o potencia aparente S de la energía eléctrica se define como la energía que un equipo requiere para poder operar [2]. La potencia aparente puede ser de dos tipos: potencia activa o potencia reactiva; la potencia activa P es la potencia que gasta el consumidor y transforma en otra forma de energía (calor, mecánica); la potencia reactiva Q es la potencia necesaria para la formación del campo electromagnético, por ejemplo motores eléctricos y transformadores, éste tipo de potencia no es transformado en ninguna forma de energía utilizable. Estos tipos de potencia se encuentran relacionados matemáticamente con la siguiente fórmula:

$$S^2 = P^2 + Q^2 \quad (1.2)$$

donde:

$$P = S \cos \phi \quad (1.3)$$

$$Q = S \sin \phi \quad (1.4)$$

1.1.2. Costo de la energía

La Comisión Federal de Electricidad (CFE) es la empresa que genera, transmite, distribuye y comercializa la energía eléctrica para la mayor parte del país, con excepción del Distrito Federal y algunas poblaciones cercanas, donde el servicio está a cargo de Luz y Fuerza del Centro.

La CFE como empresa encargada de suministrar energía eléctrica a industrias y hogares mexicanos, ha establecido treinta y tres tipos de tarifas eléctricas; 13 tipos de tarifas específicas y veinte de tarifas generales [URL 2]. La tabla 1.1 muestra las tarifas utilizadas clasificadas según su tipo y aplicación.

Tabla 1.1. Tarifas de costo de energía eléctrica de CFE.

Tipo	Aplicación	Tarifa
Específicas	Domésticas	1, 1A, 1B, 1C, 1D, 1E
	Domésticas alto consumo	DAC
	Servicios públicos	5, 5-A, 6
	Agrícolas	9, 9M
	Temporal	7
Generales	En baja tensión	2, 3
	En media tensión	O-M, H-M
	En alta tensión	HS, HS-L, HT, HT-L
	Servicio de respaldo	HM-R, HM-RF, HM-RM, HS-R, HS-RF, HS-RM, HT-R, HT-RF, HT-RM
	Servicio interrumpible	I-15, I-30

El precio total para el servicio eléctrico es la suma del cargo por el consumo más la demanda (aplicación de las tarifas mencionadas anteriormente), además de considerar otros conceptos como: cargo fijo, bonificación por alto factor de potencia o cargo por bajo factor de potencia, ajuste por variación el precio de los combustibles, inflación nacional, derecho de alumbrado público e impuesto, todo esto dependiendo del tipo de uso que se le dé a la energía eléctrica.

El costo de la energía consumida durante un período determinado se lista en un documento llamado recibo de luz. Para la CFE existen tres tipos de recibo de luz dependiendo del uso de la energía, los cuales son los siguientes:

- Recibo para consumo en baja tensión, alumbrado público, bombeo de agua y riego agrícola.
- Recibo para consumo en baja y media tensión.
- Recibo para consumo en media y alta tensión.

El recibo para baja tensión, alumbrado público, bombeo de agua y riego agrícola sólo considera el consumo de Kwh⁽²⁾, en cambio el recibo para baja y media tensión y el de media y alta tensión consideran además el consumo de kVarh⁽³⁾. El concepto de factor de potencia es tomado en cuenta en los recibos de baja y media tensión y de media y alta tensión para bonificar o aplicar un cargo al usuario dependiendo de su valor [URL 2]. La descripción más detallada de cada uno de estos recibos está contenida en el anexo A.

⁽²⁾ kwatts por hora, consumo de potencia real.

⁽³⁾ kVar por hora, consumo de potencia reactiva.

1.1.3. Mediciones eléctricas

Una medición eléctrica tiene como objetivo principal la obtención de los valores de los principales parámetros eléctricos para poder determinar el comportamiento eléctrico de una red al interpretar los valores obtenidos. También la medición eléctrica sirve para:

- Verificar los horarios y tiempo de operación de los equipos eléctricos.
- Detectar picos de demanda de energía.
- Detectar anomalías de factor de potencia.
- Detectar desbalanceo de fases.
- Tener antecedentes para la implantación de medidas de ahorro de energía y programas para controlar la demanda eléctrica.
- Comparar datos de medición eléctrica con los de facturación.

Además de los puntos antes mencionados, las mediciones eléctricas, que incluyen parámetros como: factor de potencia y distorsión armónica de voltaje, proporcionan información adicional del comportamiento de la red eléctrica. Dicha información puede ser utilizada para aplicar sistemas o dispositivos correctores del factor de potencia para reducir costos en el recibo de consumo energético.

Para realizar estas mediciones es necesario hacer uso de instrumentos que midan los parámetros que describen el comportamiento de la red eléctrica analizada, algunos de estos instrumentos son los siguientes [3].

Amperímetro: instrumento que permite medir la intensidad de corriente, en amperes, de una línea eléctrica. El tipo de amperímetro comúnmente utilizado es el de gancho (figura 1.3) debido a que la medición de corriente con un amperímetro normal debe hacerse en serie y para esto el cable que conduce la corriente debe cortarse en dos, lo cual en la realidad no es práctico. Este tipo de amperímetro no requiere la ruptura del cable pues el dispositivo mide la corriente mediante un campo magnético inducido.

Voltímetro: instrumento que mide el potencial eléctrico o voltaje de una línea eléctrica determinada, ver figura 1.4. Las puntas de este instrumento deben ser conectadas en paralelo con la carga a medir. Existen diversos tipos de presentaciones de acuerdo a la necesidad y calidad de las mediciones, pueden ser portátiles, fijo, analógicos o digitales.



Figura 1.3. Amperímetro de gancho. Cortesía Wavetek Co.

Factorímetro: indispensable cuando es necesario medir el factor de potencia, el cual es el nombre dado a la relación de la potencia activa usada en un circuito la potencia aparente que se obtiene de las líneas de alimentación. Este instrumento mide el ángulo de fase entre la corriente y el voltaje.



Figura 1.4. Voltímetro. Cortesía Fluke Co.

Wattímetro: utilizado generalmente en la medición comercial de energía eléctrica, es un dispositivo que mide la potencia promedio absorbida por una carga de dos terminales, ver figura 1.5. Está compuesto por dos bobinas, una de corriente con una impedancia muy baja y otra de voltaje con una impedancia muy alta. La bobina de corriente está conectada en serie con la carga y la bobina de voltaje está conectada en paralelo con la carga. La lectura de la potencia promedio se da cuando la corriente y el voltaje para la carga tienen ambos los mismos valores positivos o negativos.



Figura 1.5. Wattímetro. Cortesía Extech Co.

Medidor de demanda máxima: utiliza un integrador que totaliza los kilowatts demandados sobre períodos de 15 minutos. Las mediciones son almacenadas en la memoria del medidor.

Analizador de redes o medidor de potencia: instrumento capaz de obtener toda la información de los parámetros eléctricos de manera instantánea, adquiriendo y almacenando la información, ver figura 1.6.



Figura 1.6. Analizador de redes. Cortesía Yokogawa Co.

1.1.3.1. Medidores de potencia digitales

Un medidor de potencia o analizador de redes realiza mediciones de los parámetros eléctricos utilizados en el análisis de la red eléctrica. Las características de estos analizadores de redes, también conocidos como medidores de potencia, dependen del modelo y del fabricante; entre sus características más comunes se destacan las siguientes:

- Medición de parámetros eléctricos por fase o trifásica.
- Cálculo y/o presentación en pantalla de los principales parámetros eléctricos.
- Interfaz para la comunicación con una computadora.
- Registro de valores de forma remota utilizando un software destinado a ello.
- Almacenamiento de valores máximos y mínimos de todos los parámetros.

1.1.3.2. Medidor PM650

Uno de los medidores de potencia existentes en el mercado es el medidor Powerlogic PM650, mostrado en la figura 1.7. Este medidor es capaz de reemplazar medidores básicos análogos como amperímetros, voltímetros o wattíhorímetros, además es útil en aplicaciones industriales de monitorización de energía eléctrica. El medidor PM650 dispone de un visualizador opcional para observar las variables medidas y realizar ajustes localmente. Adicionalmente, este visualizador se utiliza como elemento independiente para la configuración de cualquier unidad instalada [4].

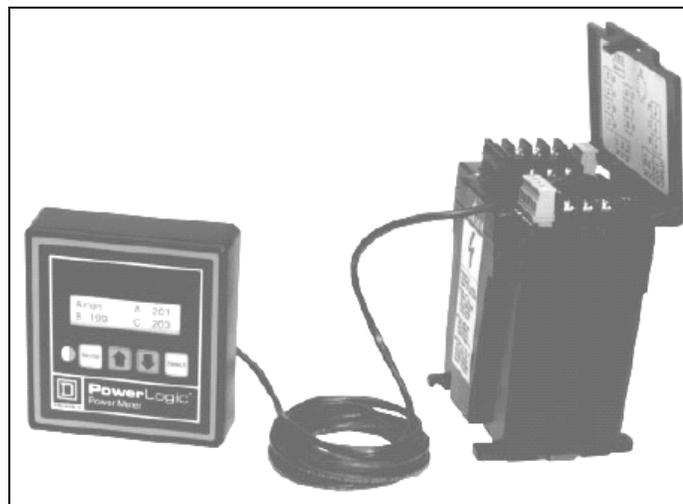


Figura 1.7. Medidor PM650 Powerlogic con visualizador. Cortesía PowerLogic.

El medidor PM650 realiza mediciones de los principales parámetros y pueden clasificarse como:

- *Lecturas en tiempo real:* corriente en cada fase, voltaje (fase-fase, fase-neutro), potencia activa (por fase, $3\emptyset$), potencia reactiva (por fase, $3\emptyset$), potencia aparente (por fase, $3\emptyset$), factor de potencia verdadera (por fase, $3\emptyset$) y frecuencia.

- *Lecturas de energía*: energía total acumulada activa, reactiva y aparente en las tres fases.
- *Lecturas de calidad de energía*: distorsión armónica de voltajes y corrientes.

Además de las características antes mencionadas, el medidor PM650 cuenta con un puerto RS-485 para comunicar el medidor con una PC y los protocolos de comunicaciones que soporta el medidor PM6550 son: Powerlogic, Modbus y Jbus [4].

1.2. COMUNICACIONES SERIALES

La información de un dispositivo electrónico es generalmente representada en forma lógica por un conjunto de *bits* (con valores de '1'⁽⁴⁾ o '0'⁽⁵⁾). Un dispositivo puede utilizar dos tipos de comunicaciones para enviar o recibir información: en forma paralela o en forma serial.

En la comunicación paralela la información se codifica en datos de 8 *bits*, los cuales son enviados en forma simultánea por el cable de transmisión, esta forma de transmitir información es rápida pero costosa ya que se requiere un cable para cada *bit*. La comunicación serial en cambio sólo necesita uno o un par de cables para enviar o recibir información a distancias mayores a 2 metros, dicha información debe enviarse bit a bit hasta completar toda la información [5].

La EIA (Asociación de Industrias Electrónicas, por sus siglas en inglés), es un organismo internacional encargado de establecer los estándares de comunicaciones, los cuales sirven para fijar las características de comunicación entre dos o más dispositivos. La EIA define las siguientes características:

- *Eléctricas*: Definición de los voltajes que representan los ceros y unos lógicos.
- *Mecánicas y de procedimiento*: El tipo de conector y la asignación de número a cada uno de los pines del conector.
- *Funcionales*: Descripción de las conexiones el transmisor y el receptor.

1.2.1. Estándar RS-232

El estándar RS-232 es definido por la EIA como un conjunto de normas para señales eléctricas utilizadas para la comunicación serial asíncrona [5]. Éste estándar se encuentra dentro de la categoría de comunicación de "single ended" (terminación única), debido a que ocupa una sola señal para el transmisor y otra para el receptor. Las características de este estándar son las siguientes:

- Tiene una terminal única para el transmisor y para el receptor.
- El número de elementos conectados son dos; un transmisor y un receptor.
- La distancia máxima es 15.24 metros a 19.2 Kbps.

⁽⁴⁾ Representado por un voltaje de 3.5V a 5V.

⁽⁵⁾ Representado por un voltaje de 0.5V a 1.5V.

- La máxima razón de datos es 19.2 Kbps para 15.24 metros.
- '1' lógico: -5 a -15 Voltios.
- '0' lógico: 5 a 15 Voltios.
- Su nivel mínimo de entrada es de ± 3 voltios.

Otra característica definida en este estándar es el tipo de conectores utilizados y la distribución de los pines en ellos. Existen dos tipos de conectores para el RS-232, el DB25 y el DB9 (DB, "conector miniatura D") con 25 y 9 pines de conexión respectivamente. El conector DB9 se muestra en la figura 1.8 y la descripción de sus pines es mostrada en la tabla 1.2.

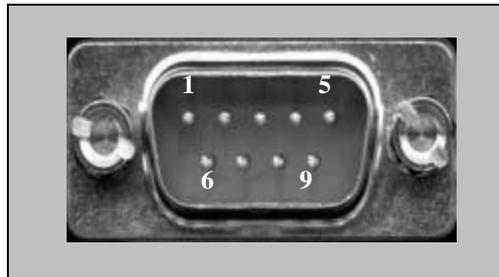


Figura 1.8. Conector macho DB9.

Tabla 1.2. Descripción de los pines del DB9 según el RS-232.

Pin del DB9	Descripción
1	\overline{CD} Detector de recepción
2	RXD Receptor
3	TXD Transmisor
4	\overline{DTR} Terminal de datos disponible
5	GND Tierra
6	\overline{DSR} Datos disponibles
7	\overline{RTS} Petición de transmisión
8	\overline{CTS} Listo para transmitir
9	Indicador de tono

1.2.1.1. Convertidor ADM233

El ADM233 pertenece a la familia de convertidores de voltaje ADM2xx de *Analog Devices Inc.*, la cual está diseñada para las interfaces de comunicaciones EIA-232-E [6]. Las características principales de este convertidor son:

- Voltaje de alimentación: 5 V.
- Velocidad de transmisión de datos: 120 kbps.
- Convertidores de niveles TTL a niveles RS-232: 2.
- Convertidores de niveles RS-232 a niveles TTL: 2.

- Voltaje de salida hasta de ± 9 V.
- Niveles de entrada del receptor: ± 30 V.

La figura 1.9 muestra la distribución de los pines y las conexiones del integrado ADM233.

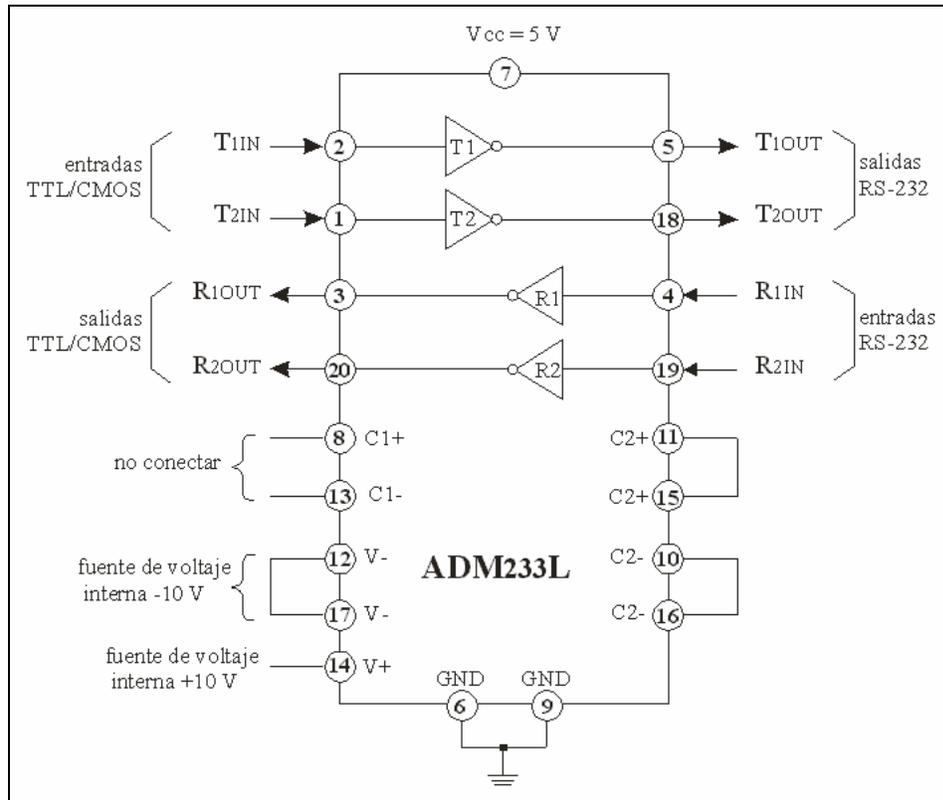


Figura 1.9. Diagrama lógico del ADM233 y sus conexiones.

1.2.2. Estándar RS-485

El RS-485 es un estándar establecido por la EIA para conectar equipos que intercambian datos en forma serial a distancias largas con un alto grado de inmunidad al ruido y alta velocidad de transmisión de datos. El estándar define la forma de transmisión de datos, la longitud máxima del cable utilizado, las características físicas del cable y las características eléctricas [5].

El antecesor del RS-485 es el estándar RS-422, el cual fue diseñado para solucionar el problema de falta de inmunidad al ruido que presenta el RS-232 a largas distancias y velocidades de transmisión altas. La eficacia de transmisión se consigue utilizando circuitos balanceados, es decir, se usan dos cables para cada señal. Las condiciones de 0 y 1 lógico son determinadas por cambios en la polaridad de los dos hilos, denotados por 'A' (señal positiva), y 'B' (señal negativa) por referencia del uno con el otro. El RS-422 sólo permite un transmisor y receptor por lo que se estableció un nuevo estándar, el RS-485, para aumentar el número de transmisores y receptores hasta 32 receptores y 32 transmisores.

Las características principales del RS-485 son:

- Señales utilizadas:
 - TXD (+), transmisión de datos (salida positiva).
 - TXD (-), transmisión de datos (salida negativa).
 - RXD (+), recepción de datos (salida positiva).
 - RXD (-), recepción de datos (salida negativa).
- Número de elementos conectados: 32 receptores y 32 transmisores por línea.
- Máxima distancia: 1,200 metros.
- Máxima razón de datos: 10 Mbps.
- '1' lógico: Cuando $5\text{ V} < V_{B-A} > 1.5\text{ V}$.
- '0' lógico: Cuando $5\text{ V} < V_{A-B} > 1.5\text{ V}$.
- Nivel mínimo de entrada: 0.2 voltios de voltaje diferencial.

El estándar RS-485 no especifica algún tipo de conector en especial, pero generalmente se usa un conector DB9 o un DB25. Una configuración para las conexiones RS-485 utilizando un conector DB9 es la que recomienda el manual del PM650, ver figura 1.10 [4].

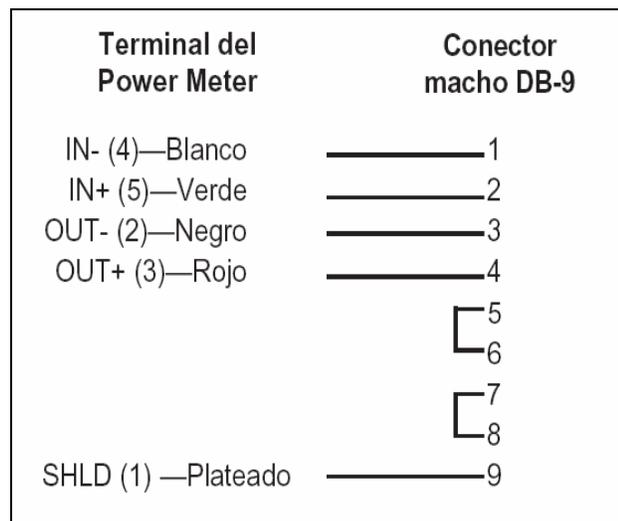


Figura 1.10. Conexiones para el conector DB9.

Para las comunicaciones RS-485, existen diferentes convertidores, según el estándar a que será convertido, puede ser RS-485 a RS-232 o RS-485 a TTL o viceversa. Un convertidor de niveles RS-485 a niveles TTL, debe ser capaz de aceptar entradas en modo común entre los -7 y los +12 V, con una sensibilidad de la entrada diferencial de $\pm 200\text{ mV}$ sobre el rango antes descrito. Un convertidor de niveles TTL a niveles RS-485 será capaz de ofrecer un voltaje de salida diferencial de 1.5 a 5 V con tensiones de línea en modo común entre -7 y +12 V. [5].

1.2.2.1. Convertidor MC3486

El MC3486 es un circuito integrado de *Texas Instruments Incorporated* que convierte niveles de voltaje RS-422 a niveles TTL. La figura 1.11 muestra la distribución de los pines y diagrama lógico del integrado MC3486 [7].

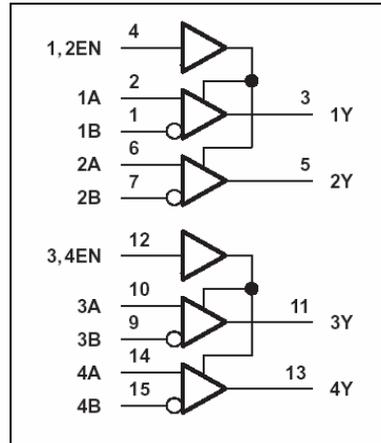


Figura 1.11. Diagrama lógico del MC3486.

Entre sus características principales se encuentran las siguientes:

- Voltaje de alimentación: 5 V.
- Voltaje diferencial máximo de entrada: ± 6 V.
- Voltaje de entrada en modo común: ± 7 V.
- Convertidores de niveles RS-422 a niveles TTL: 4.
- Voltaje diferencial⁽⁶⁾ para representar un nivel alto: voltaje menor a 0.2 V.
- Voltaje diferencial⁽⁶⁾ para representar un nivel bajo: voltaje menor a -0.2 V.
- Señal de habilitación de salida.

Este convertidor puede ser utilizado en la conversión de niveles RS-485 a niveles TTL debido al rango del voltaje diferencial permitido en sus entradas RS-422, el cual está dentro del rango establecido por el estándar RS-485.

1.2.2.2. Convertidor MC3487

El MC3487 es un circuito integrado de *Texas Instruments Incorporated* que convierte niveles de voltaje TTL a niveles RS-422. La figura 1.12 muestra la distribución de los pines y diagrama lógico del integrado MC3487 [8].

⁽⁶⁾ Voltaje de 'A' referenciado a 'B'

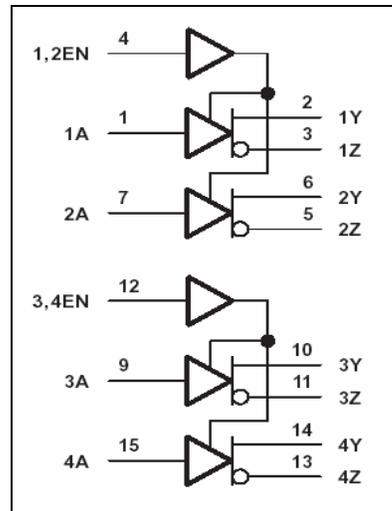


Figura 1.12. Diagrama lógico del MC3487.

Entre sus características principales se encuentran las siguientes:

- Voltaje de alimentación: 5 V.
- Voltaje de salida en modo común: 3 V.
- Voltaje diferencial⁽⁶⁾ de salida: ± 2 V.
- Voltaje de entrada de nivel alto: 2.5 V mínimo.
- Voltaje de entrada de nivel bajo: 0.5 V máximo.
- Convertidores de niveles TTL a niveles RS-422: 4.
- Señal de habilitación de salida.

Este convertidor puede ser utilizado en la conversión de niveles TTL a niveles RS-485 debido al rango del voltaje diferencial en sus salidas RS-422, el cual está dentro del rango establecido por el estándar RS-485.

1.3. PROTOCOLO MODBUS

El protocolo de comunicaciones seriales Modbus es un estándar informal desarrollado por la compañía Modicon en 1979 para establecer las reglas de comunicación y estructura de los mensajes entre dos o más dispositivos que utilizan dicho protocolo y comunicaciones RS-485 o RS-232 [9].

En el protocolo Modbus la comunicación de los datos consiste en un intercambio de información mediante el envío de un paquete de petición y la recepción de un paquete de respuesta. Dichos paquetes contienen la información necesaria o requerida en el intercambio y está organizada en cuatro campos:

⁽⁶⁾ Voltaje de 'A' referenciado a 'B'

Dirección del dispositivo	Código de la función	Datos		CRC
1 byte	1 byte	1 byte n bytes	2 bytes

Dirección del dispositivo, hace referencia a la dirección del dispositivo hacia el que va dirigido el paquete de petición. Esta dirección es un número de 1 byte de longitud cuyo valor está en el rango de 1 y 247, este número identifica a cada uno de los elementos de una red con protocolo Modbus. Cabe señalar que el número que corresponde a la dirección de registro debe decrementarse según las convenciones del protocolo Modbus.

Código de la función, corresponde al número de la función que el paquete representa. La tabla 1.3 muestra algunas de las funciones del protocolo Modbus que son soportadas por el medidor PM650 y su código correspondiente.

Tabla 1.3. Código de algunas funciones del protocolo Modbus.

Código	Función
03, 04	Lectura de registros
06	Escritura de un registro
16	Escritura de n palabras

Datos, este campo varía en longitud dependiendo si el mensaje es un paquete de petición o de respuesta. En el medidor PM650 los datos de las mediciones son almacenados en registros de dos bytes de longitud enviando el byte alto seguido del byte bajo.

CRC, Cyclic Redundancy Check. El código CRC es un campo de dos bytes de longitud. Se obtiene al ejecutar un algoritmo de detección de errores. El algoritmo utilizado es definido por el protocolo Modbus y será explicado a continuación.

1.3.1. Código CRC

Este código de detección de errores llamado también FCS (secuencia de verificación de tramas, por sus siglas en inglés), es un número calculado a partir de los campos de dirección de dispositivo, código de función más el de datos. El código CRC es obtenido multiplicando la cadena de 1's y 0's formada por el campo de dirección del dispositivo, de código de función y de datos por 2^{16} (1000000000000000) y después dividiéndolo entre el polinomio generador $x^{16} + x^{15} + x^2 + 1$, el cual puede ser expresado como un dato binario (11000000000000101). El cociente de esta operación es ignorado y los 16 bits de residuo son considerados el código CRC y añadidos al final del paquete de transmisión o recepción agregando el byte bajo seguido por el byte alto de dicho código. Si un dispositivo receptor calcula el CRC de un paquete recibido y el resultado no es el mismo que recibió se detecta un error de comunicación [9]. Lo anterior es implementado utilizando el algoritmo descrito en la figura 1.13, el cual hace uso de las operaciones aritméticas de **modulo dos** y **or exclusivo**. El argumento de entrada es una cadena de caracteres llamada **BYTE[]** formada por los campos de dirección del dispositivo, de código de función y de datos, siendo la salida la variable CRC de 2 bytes de longitud.

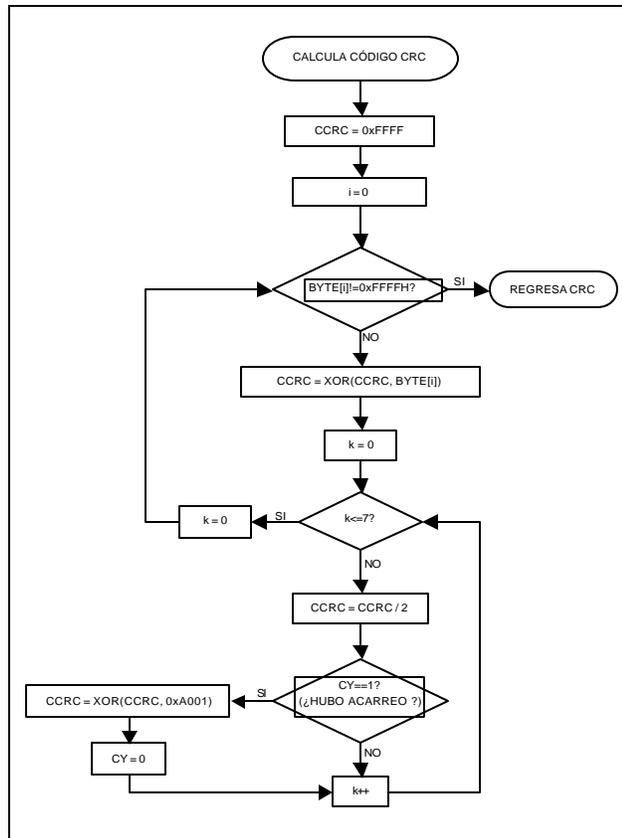


Figura 1.13. Diagrama de flujo del algoritmo para calcular el código CRC.

1.3.2. Función de lectura de registros

Ésta función obtiene el valor de uno o más registros. Los campos utilizados por la función de lectura se muestran en la tabla 1.4 según orden de envío.

Tabla 1.4. Campos de la función de lectura de registros.

Campo	Longitud
Dirección del dispositivo	1 byte
03 (Código de la función)	1 byte
Dirección de registro de inicio	2 bytes
Número de registros, n	2 bytes
Código CRC	2 bytes

Dirección del dispositivo, es el número con el cual se identifica un dispositivo en la red que utiliza el protocolo Modbus.

Función de código, en este caso el valor es 03H e indica la solicitud de lectura de la información contenida en los n número de registros empezando por la *dirección de registro de inicio*.

Dirección de registro de inicio, es un número de longitud de 2 bytes que indica la primera dirección que será leída.

Número de registros, debe contener el número de registros a leer.

Código CRC, obtenido al aplicar el algoritmo diseñado para el protocolo Modbus. Este campo es útil para detectar errores de comunicación; cuando se recibe el paquete se calcula el CRC con los valores recibidos y se coteja con el CRC recibido, si ambos coinciden significa que el paquete llegó incorrupto [9]. Los campos de dirección de registro de inicio como el de número de registros que constan de dos bytes, deben ordenarse según el byte alto seguido del byte bajo. El campo del código CRC se ordena del byte bajo seguido por el byte alto.

La respuesta a la petición de lectura es el envío de un paquete compuesto por los campos listados en la tabla 1.5 según su orden de recepción.

Tabla 1.5. Campos del paquete de respuesta a la función de lectura de registros.

Campo	Longitud
Dirección del dispositivo	1 byte
03 (Código de la función)	1 byte
Cuenta de bytes, 2 X no. registros	1 byte
Primer registro en rango	2 bytes
Segundo registro en rango	2 bytes
M	2 bytes
Código CRC	2 bytes

Dirección del dispositivo, es el identificador del dispositivo que envía la respuesta.

Función de código, envía el código de la función a la que es respuesta este paquete. En este caso 03H.

Cuenta de bytes, es el número de bytes que se envían, 2 por cada registro que se leyó.

N Registro en rango, es el valor del n registro; tomando en cuenta que cada registro tiene una longitud de 2 bytes, se envían los registros de forma ascendente en rango, del primero al n -enésimo registro.

Código CRC, código de detección de errores, conformado por dos bytes, se recibe primero el byte bajo seguido del byte alto.

Para explicar el uso de ésta función, supones que se quiere el registro 1001 del dispositivo con dirección 01H. El paquete de petición de lectura de éste registro se obtiene con los siguientes pasos.

1. Se construye la cadena de la cual se obtendrá el CRC, a partir de los valores de:

Dirección del dispositivo:	1, 01H
Código de la función:	3, 03H
Dirección de registro de inicio:	$1001 - 1 = 1000$, 03E8H
Número de registros:	1, 01H

La cadena de bytes resultante es:

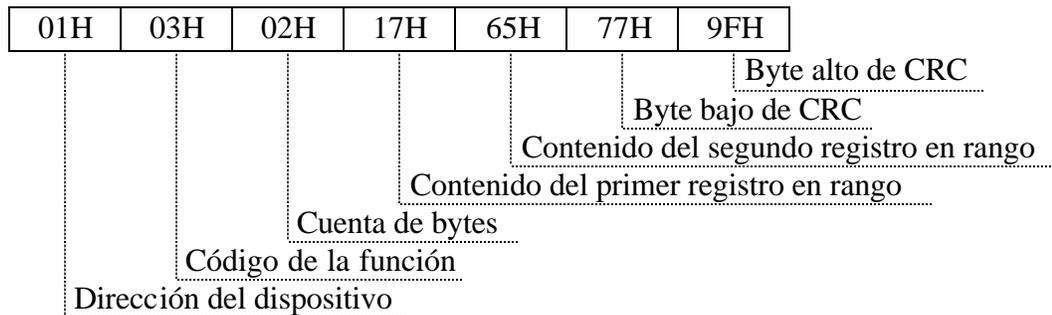
MSB						LSB
01H	03H	03H	E8H	00H	01H	

2. Se aplica el algoritmo del código de CRC y el resultado obtenido 7A04H se añade a la cadena⁽⁷⁾. La cadena resultante es:

MSB							LSB
01H	03H	03H	E8H	00H	01H	04H	7AH

3. Se envía el paquete.

4. Si el dispositivo recibe el paquete sin errores de transmisión envía el siguiente paquete de respuesta:



1.4. LABVIEW

LabView (Laboratory Virtual Instrument Engineering Workbench), es un producto de la firma National Instruments que tiene como principal propósito la adquisición de datos y el control de instrumentos enfocados a la automatización [URL 3].

LabView es diferente a otro lenguaje de programación C, BASIC o LabWindows de National Instruments, debido a que no utiliza un lenguaje basado en texto sino un lenguaje de programación gráfica [10], por lo que los programas desarrollados en LabView tienen la forma de diagrama de bloques.

1.4.1. Instrumentos Virtuales

Los programas desarrollados en LabView son conocidos como VIs (Instrumentos Virtuales, por sus siglas en inglés), y tienen una extensión *.vi. Un VI está compuesto por una interfaz de usuario interactiva, un diagrama de flujo de datos que es en sí el código de programa y conectores de iconos que son empleados cuando el código es utilizado en otro programa. A estos tres componentes se les denomina: panel frontal, diagrama de bloques e icono y conector respectivamente. LabView incluye librerías o funciones para el desarrollo de programas, los VIs pueden adoptar la apariencia y operación que tienen los instrumentos de medición utilizados en la adquisición y control de datos.

⁽⁷⁾ Recordando que en el campo del CRC, el byte bajo va primero que el byte alto.

Panel Frontal

Es la interfaz de usuario y puede adoptar la apariencia de un instrumento de medición, con sus respectivos botones de control, gráficas e indicadores, los cuales son llamados *controles* e *indicadores*; como su nombre lo indica, los *controles* sirven para operar el VI y los *indicadores* muestran la salida del dispositivo. La figura 1.14 muestra el panel frontal del *modulo almacenamiento.vi*.



Figura 1.14. Panel de control de *modulo almacenamiento.vi*.

Diagrama de bloques

Esta parte de un VI constituye su código fuente gráfico, el cual es construido uniendo objetos que representan variables, constantes o funciones para recibir o enviar datos a otros objetos. Los componentes del panel de control, indicadores o controles, tienen su representación en el diagrama de bloques y son utilizados como variables o constantes en el código del VI. Como en los lenguajes basados en texto, el lenguaje Labview utiliza estructuras condicionales o iterativas: estructuras **SEQUENCE** (secuencia de instrucciones), **CASE** (estructura *case*), **FOR LOOP** (estructura *for*), **WHILE LOOP** (estructura *while*).

El diagrama de bloques de la figura 1.15 corresponde al subVI utilizado en el programa de descarga de datos del módulo de almacenamiento llamado *inf_confg.vi*, en el cual se leen datos del puerto serial que se asigna y convierten en la información del almacenamiento de los datos obtenidos del medidor.

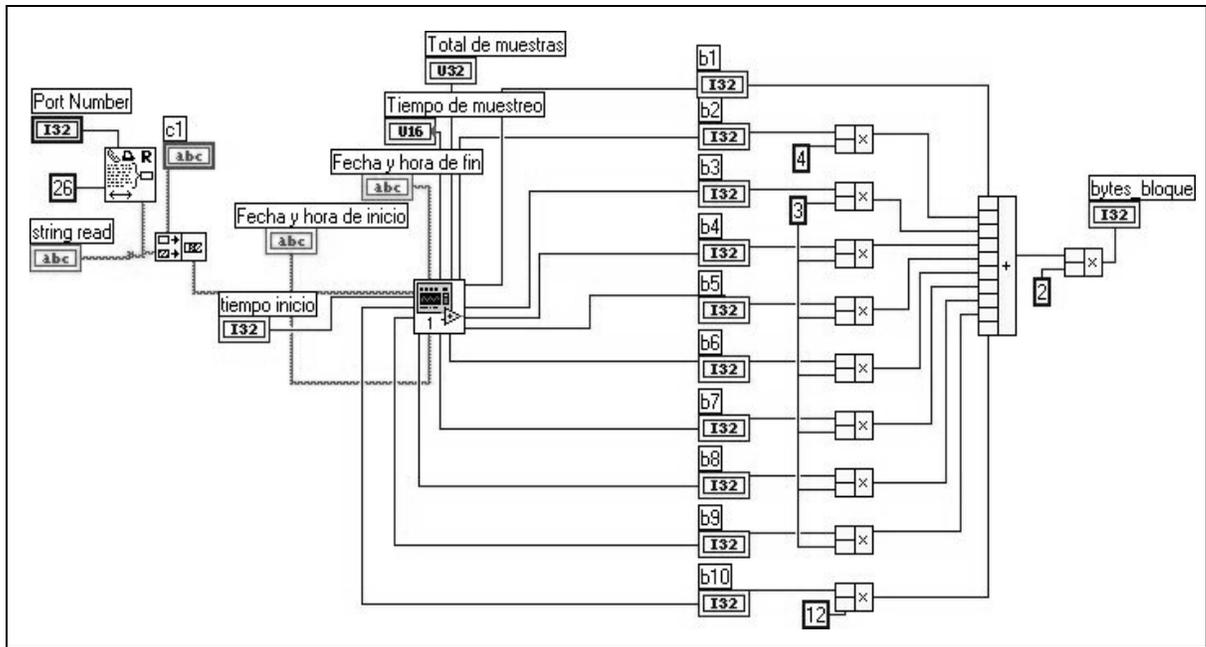


Figura 1.15. Diagrama de bloques de *inf_config.vi*.

Icono y conector

Un VI puede ser llamado dentro de otro VI, entonces el VI empleado se convierte en un subVI; de forma similar a las subrutinas de lenguaje basado en texto, los subVI pueden tener entradas y/o salidas de datos que corresponden a los conectores del icono del subVI. Para el caso del subVI *inf_config.vi*, los conectores e icono se muestra en la figura 1.16, los conectores de entrada se localizan a la izquierda del conector y en la derecha se ubican los conectores de salida.

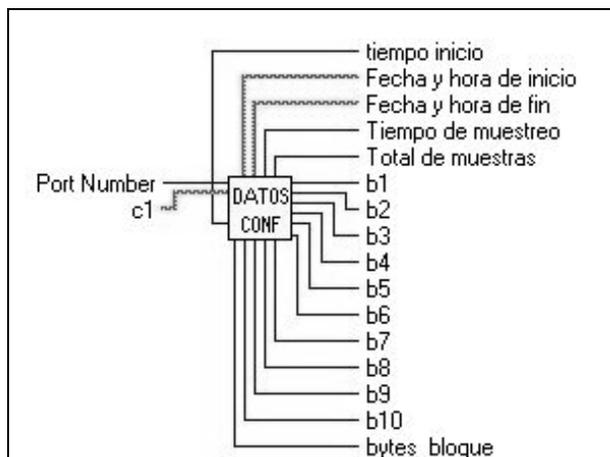


Figura 1.16. Icono y conectores del subVI: *inf_config.vi*.

CAPÍTULO 2. DESCRIPCIÓN DEL MÓDULO DE ALMACENAMIENTO

2.1. DESCRIPCIÓN GENERAL

Este capítulo describe el diseño del hardware del módulo para almacenar los datos obtenidos por el medidor PM650. Las funciones que realiza el módulo de almacenamiento son implementadas mediante componentes electrónicos agrupados en submódulos, cumpliendo cada uno de ellos una función específica; la figura 2.1 muestra las conexiones entre los submódulos, la conexión entre el módulo de almacenamiento y el medidor PM650.

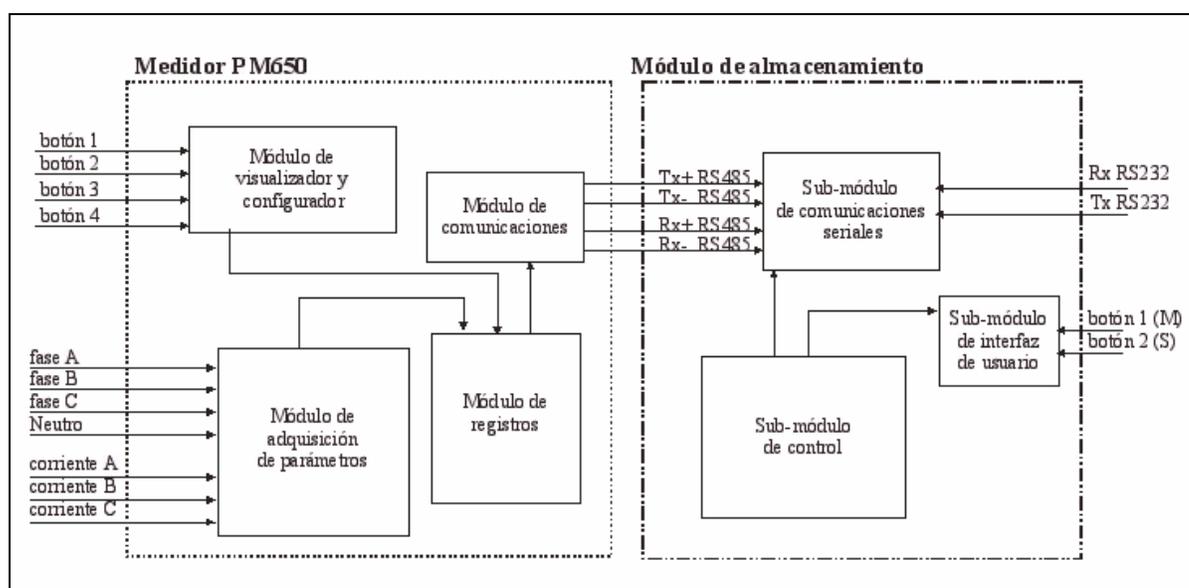


Figura 2.1. Diagrama a bloques del medido PM650 y el módulo de almacenamiento.

2.2. MEDIDOR PM650

El medidor PM650 de la marca PowerLogic, mostrado en la figura 2.2, se divide funcionalmente en cuatro módulos básicos: módulo de visualizador y configurador, módulo de adquisición de parámetros, módulo de los registros y módulo de comunicaciones.



Figura 2.2. Medidor PM650 PowerLogic, cortesía SquaredD.

- **Módulo de visualizador y configuración del medidor:** es la interfaz entre el medidor PM650 y el usuario para establecer la forma de trabajo del medidor, para ello cuenta con una pantalla de 16 caracteres por 2 filas y cuatro botones integrados al visualizador; también tiene la capacidad de mostrar los valores instantáneos obtenidos de los parámetros eléctricos.
- **Módulo de adquisición de parámetros:** es el encargado de recibir las señales de voltaje y corriente de cada fase de la red eléctrica y tomar lecturas de los parámetros eléctricos. Tiene siete terminales de entrada para cada uno de los voltajes (V_A , V_B , V_C y V_N) y tres para las corrientes (I_A , I_B y I_C).
- **Módulo de los registros del medidor:** es la parte donde se almacenan temporalmente los valores de las mediciones realizadas por el medidor mientras no se obtengan nuevos valores. El medidor PM650 contiene 1078 registros de almacenamiento, los cuales se almacenan los valores de la configuración del medidor y la información de las mediciones realizadas [7].
- **Módulo de comunicaciones del medidor:** el medidor PM650 tiene un puerto de comunicaciones seriales RS-485 que se emplea para leer los registros del medidor; el manejo del puerto RS-485 para establecer comunicación con el medidor se lleva a cabo mediante los protocolos de comunicaciones Powerlogic, Modbus o Jbus los cuales son protocolos de comunicaciones soportados por el medidor.

2.2.1. Parámetros considerados para el almacenamiento

Los parámetros considerados por el módulo de almacenamiento para su registro son: frecuencia, corriente en cada fase, voltaje en cada fase, factor de potencia, potencia activa, potencia reactiva, distorsión armónica de corriente en cada fase, distorsión armónica de voltaje en cada fase, energía aparente total, energía real total y energía reactiva total. Los valores de estos

parámetros son almacenados temporalmente en algunos de los 5,821 registros del medidor PM650; cada registro de almacenamiento es de dos bytes de longitud y tiene una dirección específica. La lectura de estos registros se efectúa a través del puerto de comunicaciones RS-485. Los registros de los parámetros mencionados y su descripción se enumeran y describen en la tabla 2.1.

Tabla 2.1. Parámetros considerados por el módulo de almacenamiento.

Nombre del registro	Dirección	Rango	Unidades
Frecuencia	1001	4500 a 6600, (45–66Hz)	0,01 Hertz / factor
Corriente de fase A	1003	Amps/Factor de escala A	0 a 32 767
Corriente de fase B	1004	Amps/Factor de escala A	0 a 32 767
Corriente de fase C	1005	Amps/Factor de escala A	0 a 32 767
Corriente calculada del neutro	1006	Amps/Factor	0 a 32 767
Voltaje de fase A a B	1014	Voltios/Factor de escala D	0 a 32 767
Voltaje de fase B a C	1015	Voltios/Factor de escala D	0 a 32 767
Voltaje de fase C a A	1016	Voltios/Factor de escala D	0 a 32 767
Voltaje de fase A a neutro	1018	Voltios/Factor de escala D	0 a 32 767
Voltaje de fase B a neutro	1019	Voltios/Factor de escala D	0 a 32 767
Voltaje de fase C a neutro	1020	Voltios/Factor de escala D	0 a 32 767
Factor de potencia real fase A	1031	En milésimas	-100 a +1000 a +100
Factor de potencia real fase B	1032	En milésimas	-100 a +1000 a +100
Factor de potencia real fase C	1033	En milésimas	-100 a +1000 a +100
Potencia activa fase A	1039	kW/Factor de escala E	0 a +/-32 767
Potencia activa fase B	1040	kW/Factor de escala E	0 a +/-32 767
Potencia activa fase C	1041	kW/Factor de escala E	0 a +/-32 767
Potencia reactiva fase A	1043	kW/Factor de escala E	0 a +/-32 767
Potencia reactiva fase B	1044	kW/Factor de escala E	0 a +/-32 767
Potencia reactiva fase C	1045	kW/Factor de escala E	0 a +/-32 767
THD/thd de corriente fase A	1051	% en décimas	0 a 10 000
THD/thd de corriente fase B	1052	% en décimas	0 a 10 000
THD/thd de corriente fase C	1053	% en décimas	0 a 10 000
THD/thd de voltaje fase A	1055	% en décimas	0 a 10 000
THD/thd de voltaje fase B	1056	% en décimas	0 a 10 000
THD/thd de voltaje fase C	1057	% en décimas	0 a 10 000
Energía aparente total de 3 fases	1617–1620	VAH	0 a 9 999 999 999 999 999
Energía real total de 3 fases	1621–1624	WH	0 a +/-9 999 999 999 999 999
Energía reactiva total de 3 fases	1625–1628	VArH	0 a +/-9 999 999 999 999 999

2.3. MÓDULO DE ALMACENAMIENTO

Las características del módulo de almacenamiento son:

- 128 kBytes de memoria RAM para almacenar los datos del medidor PM650.
- Puerto de comunicaciones RS-485 utilizado para la conexión hacia el medidor para la obtención de los valores de los parámetros elegidos por el usuario.
- Puerto de comunicaciones RS-232 que permite el envío de los datos almacenados hacia una computadora personal.

- Interfaz con el usuario constituida por una pantalla de 16 caracteres por 2 líneas y un teclado de dos botones para la elección de los datos a almacenar y el valor del tiempo de almacenamiento.

El módulo de almacenamiento se integra por los siguientes submódulos:

- Submódulo de control.
- Submódulo de interfaz con el usuario.
- Submódulo de comunicaciones seriales.

Cada sub -módulo cumple una o más tareas específicas que interactúan para que el módulo lea, almacene y envíe los datos obtenidos por el medidor PM650 conectado a la red eléctrica. La relación entre los submódulos del módulo de almacenamiento se muestra en la figura 2.3.

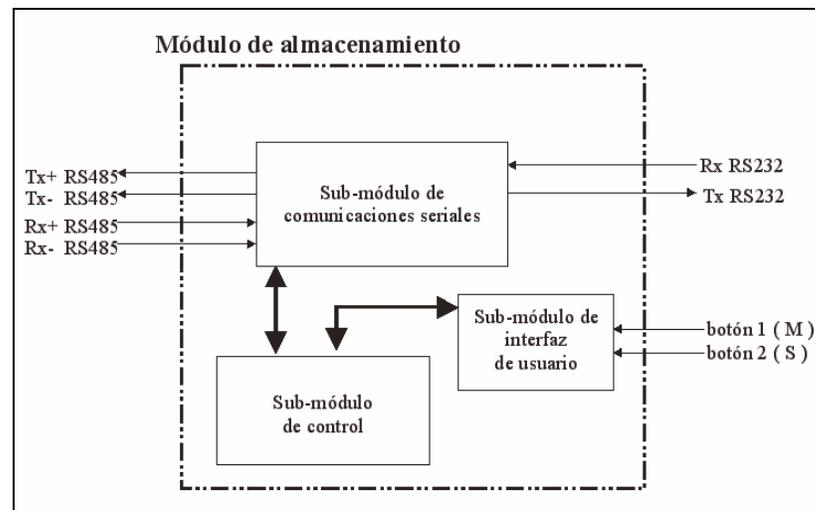


Figura 2.3. Diagrama a bloques del módulo de almacenamiento.

A continuación se describen las características particulares e implementación de cada uno de los submódulos.

2.3.1. Submódulo de control

Este submódulo es el cerebro del módulo de almacenamiento, en él se generan las instrucciones que se ejecutan para que el sistema cumpla con el objetivo de almacenar los datos provenientes del medidor según la configuración dada por el usuario. Este submódulo interactúa y administra las funciones de los demás submódulos del módulo de almacenamiento.

De forma conceptual el control se divide en dos partes, las cuales son:

- El sistema mínimo del microcontrolador 8031.
- El módulo de memoria de datos.

El sistema mínimo del microcontrolador 8031, descrito en el anexo B, está constituido por el microcontrolador, la memoria de programa, la memoria de datos y demás componentes necesarios para el funcionamiento del microcontrolador.

El módulo de almacenamiento cuenta con 128 kBytes de memoria, constituida por 5 memorias SRAM de 32 kBytes de capacidad cada una y destinada a almacenar los datos obtenidos en la lectura de los registros del PM650. La memoria del módulo de almacenamiento se divide en memoria para datos del microcontrolador 8031 y memoria para los datos del medidor. La distribución de la memoria, mostrada en la figura 2.4, es la siguiente:

- 32 kBytes, de la dirección 00000H a la 07FFFH, destinados a los datos del 8031.
- 128 kBytes, de la dirección 08000H a la 27FFFH, correspondientes a la memoria donde se almacenan los datos del medidor y las direcciones donde se mapean el teclado y la pantalla del módulo.

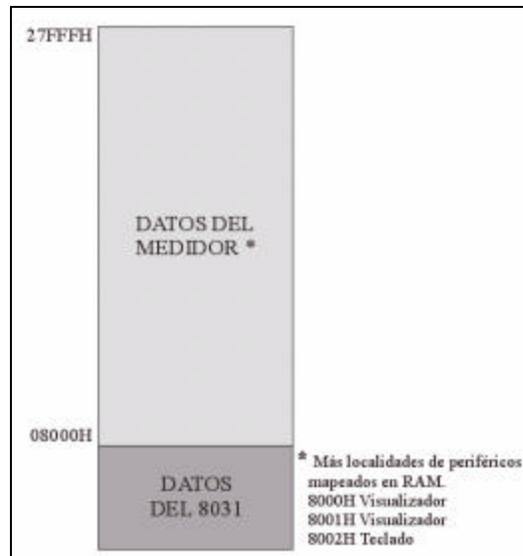


Figura 2.4. Distribución de la memoria externa del sistema mínimo del 8031.

El microcontrolador 8031 efectúa el direccionamiento de la memoria del módulo, debido a que sólo puede direccionar hasta 64 kBytes de memoria externa utilizando sus 16 líneas de direcciones ($A_0 - A_{15}$) se utilizan otras señales del microcontrolador, el rango del direccionamiento y las señales que el microcontrolador utiliza para el direccionamiento de la memoria del módulo de almacenamiento se listan y describen en la tabla 2.2.

Tabla 2.2. Señales empleadas en el direccionamiento de memoria del módulo de almacenamiento.

Señales	Descripción
$A_0 - A_{15}$	Bus de direcciones, capaz de direccionar hasta 64KBytes.
\overline{WR}	Se activa en bajo cuando se trata de escribir a memoria de datos externa
\overline{RD}	Se activa en bajo cuando se trata de leer a memoria de datos externa
P1.0, P1.1	Pines del puerto 1 del 8031, A_{16} y A_{17} , respectivamente

El direccionamiento de memoria, mostrado en la tabla 2.3, se realiza implementando un decodificador de memoria con una GAL22V10, en la cual se programaron las ecuaciones incluidas en el anexo E.1.

Tabla 2.3. Distribución y asignación de memoria externa.

P1.1 A17	P1.0 A16	A15	A14	A13	...	A1	A0	Señal	Descripción
0	0	0	X	X	X	X	X	$\overline{B_RAM}$	00000H a 07FFFH, Memoria de datos del 8031 *
0	0	1	X	X	X	X	X	$\overline{B1}$	08000H a 0FFFFH, Datos del PM650
0	1	0	X	X	X	X	X	$\overline{B2}$	10000H a 17FFFH, Datos del PM650
0	1	1	X	X	X	X	X	$\overline{B3}$	18000H a 1FFFFH, Datos del PM650
1	0	0	X	X	X	X	X	$\overline{B4}$	20000H a 27FFFH, Datos del PM650

* Incluyendo las localidades 07FF0H, 07FF01H y 07FF02H, para teclado y visualizador

X No importa

Las conexiones para el manejo de la memoria externa se muestran en la figura 2.5. A la entrada del decodificador se conectan las señales A0-A15, \overline{WR} , \overline{RD} , P1.0, P1.1, DI (señal de habilitación de la pantalla) y TE (señal de habilitación del teclado), las señales de salida son las señales de habilitación para cada una de las memorias SRAM: $\overline{B_RAM}$ es la habilitación de la memoria SRAM usada para los datos del 8031; $\overline{B1}$, $\overline{B2}$, $\overline{B3}$ y $\overline{B4}$ son las señales de habilitación para cada una de las memorias SRAM que integran la memoria de los datos de las mediciones del PM650.

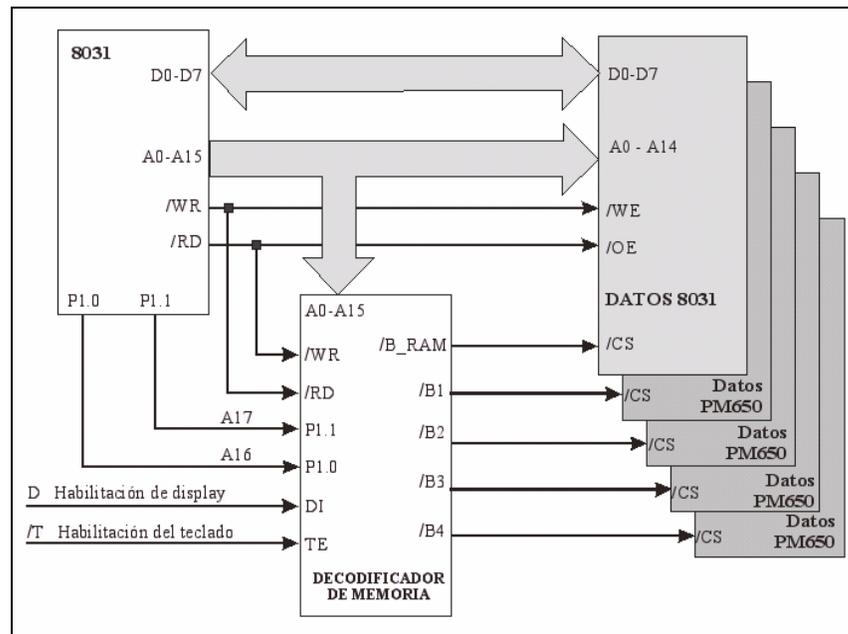


Figura 2.5. Conexiones de la memoria externa del módulo de almacenamiento.

2.3.1.1. Respaldo de la información almacenada

La información almacenada en la memoria del módulo se respalda mediante el uso del circuito integrado ADM691. El ADM691 es un circuito que compara el valor de dos voltajes (V_{CC} , voltaje de alimentación y V_{BATT} , voltaje de respaldo) y dirige el voltaje de valor mayor a su salida V_{OUT} , ver figura 2.6. Para respaldar la información del módulo de almacenamiento se conectó la salida V_{OUT} a los pines de voltaje de alimentación de las cinco memorias SRAM, que conforman la memoria del módulo, y del decodificador de memoria.

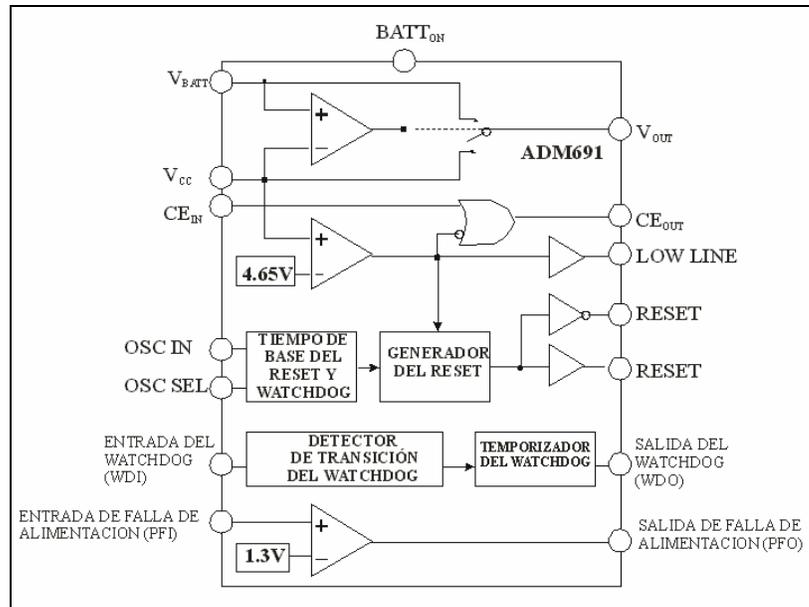


Figura 2.6. Diagrama lógico del ADM691.

2.3.2. Submódulo de interfaz de usuario

Integrado por el teclado y la pantalla del módulo de almacenamiento es el enlace entre el usuario y el módulo; la pantalla se emplea para enviar mensajes al usuario que responde utilizando el teclado, el cual tiene dos botones, M y S, el primero desplaza al usuario por las diferentes opciones y el segundo sirve para afirmar alguna elección o valor de la opción desplegada en la pantalla.

El teclado y la pantalla son mapeados en memoria RAM, esto significa que al leer o al escribir a una localidad determinada se lee o escribe al dispositivo que está mapeado en esa localidad. El controlador de periféricos que se utiliza para decodificar el teclado y el visualizador es implementado en una GAL22V10, en la cual están programadas las ecuaciones que hacen el mapeo de dichos componentes, ver anexo E.2. Las señales de habilitación del teclado y de la pantalla, /T y D respectivamente, están conectadas a la salida del controlador de periféricos y las señales de entrada del mismo son A0-A15, \overline{WR} , \overline{RD} , P1.0 y P1.1, ver figura 2.7.

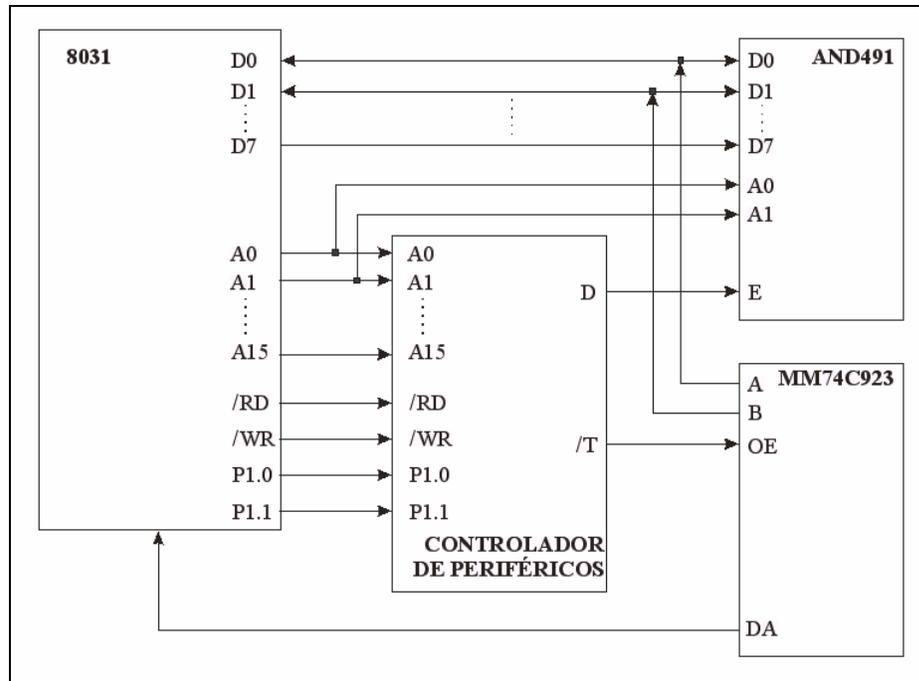


Figura 2.7. Conexiones del submódulo de interfaz de usuario.

2.3.2.1. Teclado

Los dos botones o teclas que conforman el teclado del módulo de almacenamiento son decodificados por el circuito integrado MM74C923, su diagrama lógico se muestra en la figura 2.8; el MM74C923 es un decodificador de teclas que codifica un arreglo de interruptores sin necesidad de dispositivos externos, además elimina el uso de diodos para evitar rebotes e incluye señales de control de lectura de los datos del teclado y habilitación del mismo [11].

El decodificador MM74C923, su diagrama lógico se muestra en la figura 2.8, lee el valor del arreglo de interruptores, cada uno representando una tecla, enviando el número correspondiente a la tecla oprimida representado en valor binario en sus cuatro pines de salida. El arreglo es formado con las entradas X_1 a X_4 y Y_1 a Y_5 , que corresponden a las columnas y filas del arreglo. La lectura de las teclas está controlada por un oscilador externo que activa un contador de dos bits, cuando se oprime una tecla se deshabilita el contador impidiendo que otra tecla pueda ser leída, se envía un '1' lógico a la señal *data available* (pin DA); cuando la señal *output enable* (pin \overline{OE}) tiene un valor de '0' lógico el valor del número correspondiente a la tecla oprimida es enviado a los pines A, B, C, D y E siendo E el bit más significativo.

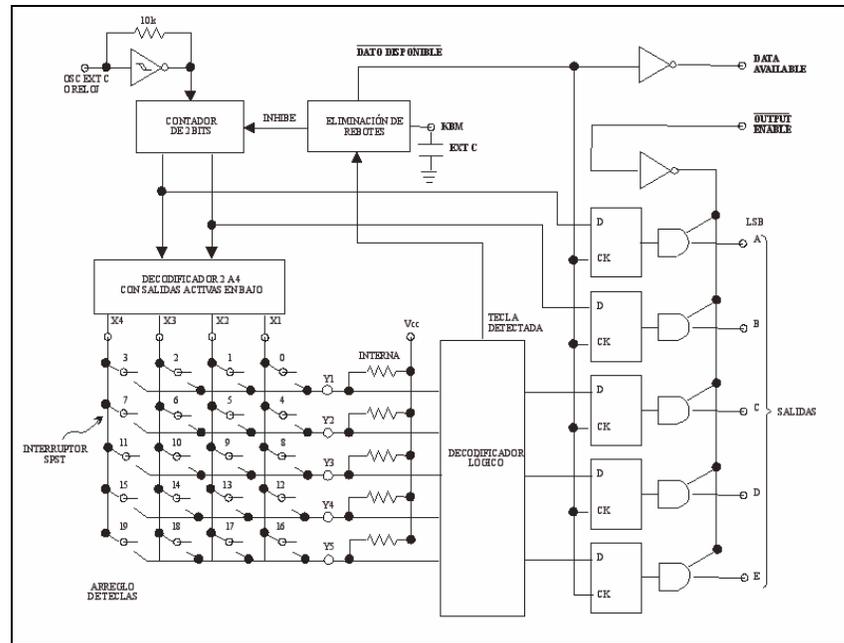


Figura 2.8. Diagrama lógico del MM74C923.

Lo descrito anteriormente se muestra utilizando el diagrama de tiempos del MM74C923 de la figura 2.9. Cuando se oprime una tecla, nivel bajo⁽¹⁾ en el interruptor correspondiente, la señal DA se activa en alto⁽²⁾ transcurrido un tiempo $T1$, los datos de salida están en un tercer estado hasta que la señal \overline{OE} tenga el valor de nivel bajo; cuando esto ocurre la señal DA vuelve al nivel bajo y el valor de la tecla oprimida se envía las salidas A, B, C, D y E [11].

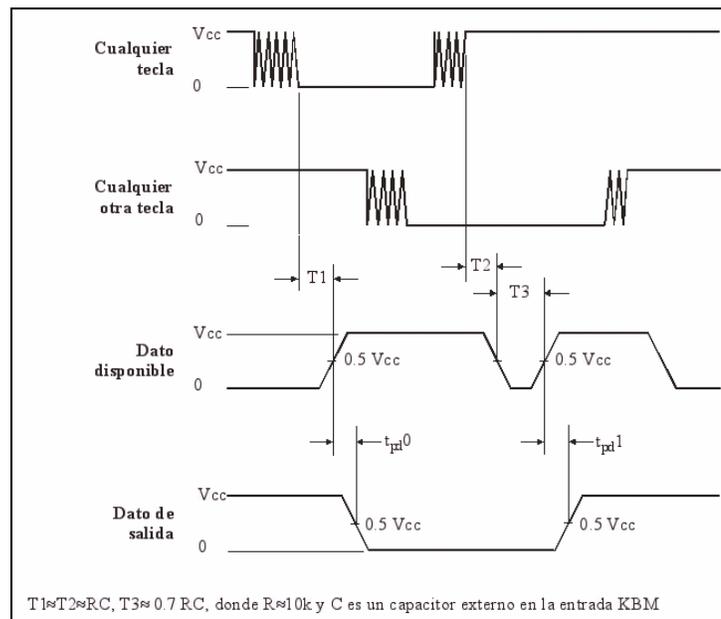


Figura 2.9. Diagrama de tiempos del MM74C923.

- (1) nivel bajo, '0' lógico.
 (2) nivel alto, '1' lógico.

Para la implementación del teclado del módulo de almacenamiento se considera lo siguiente: las salidas A y B del decodificador MM74C923 son conectadas al bus de datos del microcontrolador, la señal DA negada, usando una compuerta NOT, se conecta a $\overline{\text{INT0}}$ (interrupción externa 0 del 8031) y la señal $\overline{\text{OE}}$ a la salida $\overline{\text{/T}}$ del controlador de periféricos, éstas consideraciones son mostradas en la figura 2.10. Cuando el 8031 detecta la interrupción externa 0, entra a una rutina que lee la dirección 8002H, dirección donde está mapeado el teclado; cuando ésta dirección llega al controlador de periféricos, éste envía un nivel bajo a la entrada $\overline{\text{OE}}$ permitiendo que el valor de la tecla oprimida sea leído. Los valores correspondientes a las dos teclas del módulo de almacenamiento se listan en la tabla 2.4.

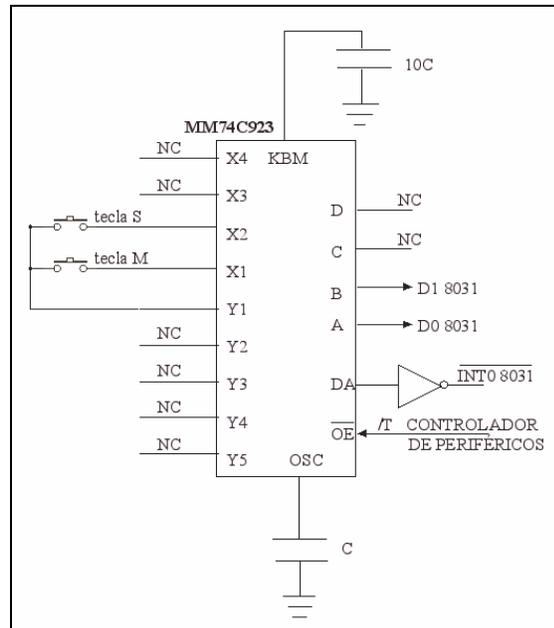


Figura 2.10. Conexiones del MM74C923.

Tabla 2.4. Valores de los pines de salida del MM74C923.

Valor de salida en el MM74C923	Tecla oprimida	
	M	S
A	0	1
B	0	0

2.3.2.2. Pantalla

La pantalla utilizada en el módulo de almacenamiento es el dispositivo AND491GST/GST-LED (AND491), constituido por una pantalla de cristal líquido de 2 líneas por 16 caracteres y un controlador que mediante comandos y líneas de control configura y despliega caracteres alfanuméricos en la pantalla. El diagrama a bloques del AND491, figura 2.11, muestra sus señales de entrada y de salida; éstas son divididas en señales de control (E, $\overline{\text{R/W}}$ y RS), señales de alimentación eléctrica (V_{DD} , V_{O} y GND) y señales del bus de datos del AND491 (DB0 - DB7) [12].

Las señales de control, descritas en la tabla 2.5, se utilizan para elegir el modo de operación de la pantalla, desplegar caracteres o leer caracteres desplegados en la pantalla; las señales de alimentación, como su nombre lo indica, envían el voltaje a su entrada para el funcionamiento de la pantalla y el controlador del AND491 y las señales de bus de datos reciben los valores de configuración o caracteres a desplegar en la pantalla o envían los caracteres leídos de la pantalla.

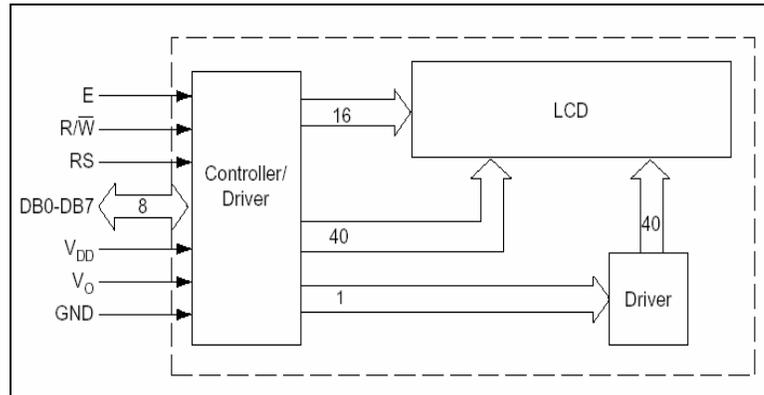


Figura 2.11. Diagrama a bloques del AND491. Cortesía Purdy Electronics.

Tabla 2.5. Descripción de las señales de control del AND491.

Señal de control	Descripción
E	Habilitación del controlador, '1' lógico.
R/ \overline{W}	'1' lógico: lectura. '0' lógico: escritura.
RS	'1' lógico: escritura de caracteres. '0' lógico: escritura de comandos.

Para el funcionamiento de la pantalla del AND491 se consideran los siguientes pasos:

1. Configuración del modo de operación de la pantalla: elección de la longitud del bus de datos (4 u 8 bits), tamaño de los caracteres desplegados (5 x 10 o 5 x 7 puntos) y número de líneas (1 o 2), mediante la escritura de comandos $R/\overline{W} = 0$, $RS = 0$ y $DB0 - DB7 =$ comando .
2. Envío de caracteres a la pantalla, $R/\overline{W} = 0$, $RS = 1$ y $DB0 - DB7 =$ caracter o $DB0 - DB3 =$ caracter , según longitud del bus de datos.

En el módulo de almacenamiento, el AND491 está mapeado en las direcciones 8000H y 8001H considerando lo siguiente: la señal R/\overline{W} se conectada directamente al pin A0 del 8031, la señal RS a A1 del 8031, la señal E se conecta al pin D del controlador de periféricos, los pines DB0 al DB7 son conectados al bus de datos del microcontrolador, éstas conexiones y las de las señales de alimentación se muestran en la figura 2.12.

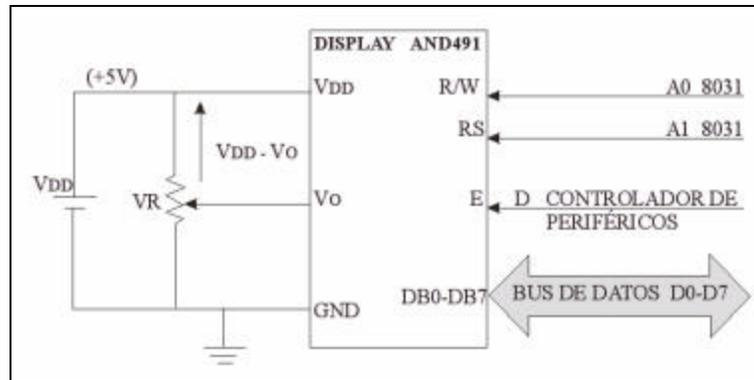


Figura 2.12. Conexiones del AND491.

2.3.3. Submódulo de comunicaciones seriales

El módulo de almacenamiento cuenta con la interfaz serial del 8031 para establecer comunicaciones seriales, pero no considera las características eléctricas de un estándar de comunicaciones en particular; además, la interfaz serial sólo maneja un receptor y un transmisor. El submódulo de comunicaciones seriales se encarga de la conversión adecuada de los niveles de voltaje, dependiendo del estándar y de la elección del puerto de comunicaciones, mediante el empleo de un multiplexor, un demultiplexor y convertidores de niveles TTL a RS-232, RS-232 a TTL, TTL a RS-485 y RS-485 a TTL, mostrados en la figura 2.13.

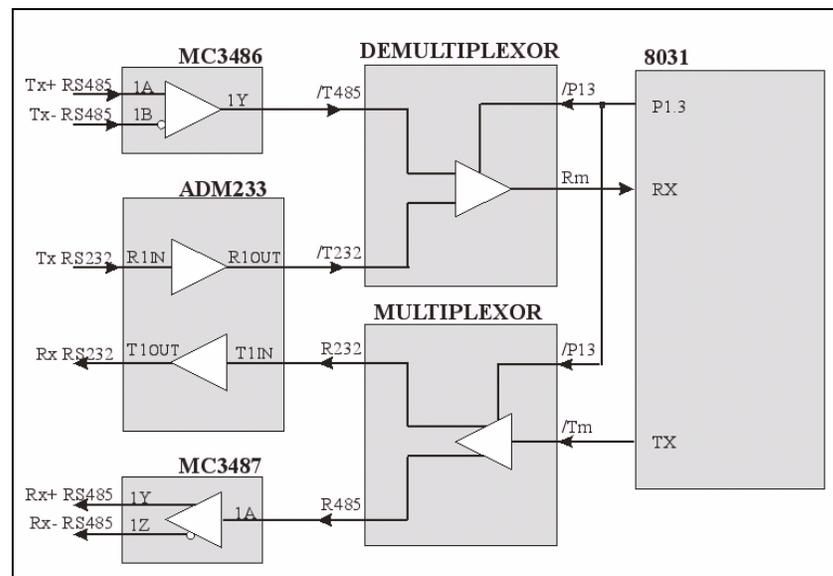


Figura 2.13. Conexiones de las comunicaciones seriales del módulo de almacenamiento.

El multiplexor y el demultiplexor son controlados por software poniendo en nivel alto o en bajo el pin 3 del puerto 1 del 8031 (P1.3). Si las señales (transmisión o recepción) deben ser RS-232, la señal de P1.3 debe ponerse en bajo y en alto si las señales son de tipo RS-485. El multiplexor y demultiplexor son implementados en una GAL22V10, las ecuaciones para la programación de la GAL22V10 se listan en el anexo E.1.

Las señales Tx y Rx son convertidas a niveles RS-232 y RS-485 utilizando los convertidores MC3486, MC3487 o ADM233. Las señales de transmisión del medidor (Tx + RS485, Tx - RS485) son transformadas a niveles TTL mediante el convertidor MC3486; la señal Tx RS232 es convertida a TTL con el ADM233, éstas señales convertidas son conectadas al demultiplexor, la salida de éste es conectada al pin de recepción del 8031, Rx. La señal de transmisión del 8031 (Tx) es conectada a la entrada del multiplexor, éste direcciona el valor de la señal Tx hacia el AMD233 o hacia el MC3487. Si la señal de P1.3 está en nivel bajo la salida del multiplexor se dirige hacia el convertidor ADM233 el cual convierte los niveles TTL de la señal a niveles RS-232. Si la señal de P1.3 está en nivel alto la salida del multiplexor se dirige hacia el convertidor MC3487 y la señal se transforma a niveles RS-485.

CAPÍTULO 3. DESARROLLO DEL SOFTWARE

3.1. DESCRIPCIÓN GENERAL

El módulo de almacenamiento realiza las tareas de configuración del almacenamiento, lectura de los registros del medidor, almacenamiento y envío de datos hacia la PC; estas tareas son ejecutadas a partir de dos programas: el programa del módulo de almacenamiento y el programa de descarga de datos hacia la PC.

El programa del módulo del almacenamiento está contenido en la EPROM del sistema mínimo del submódulo de control mientras que el programa de descarga está instalado en la PC donde se almacenan los datos provenientes del módulo, la relación entre estos programas está representada en la figura 3.1.

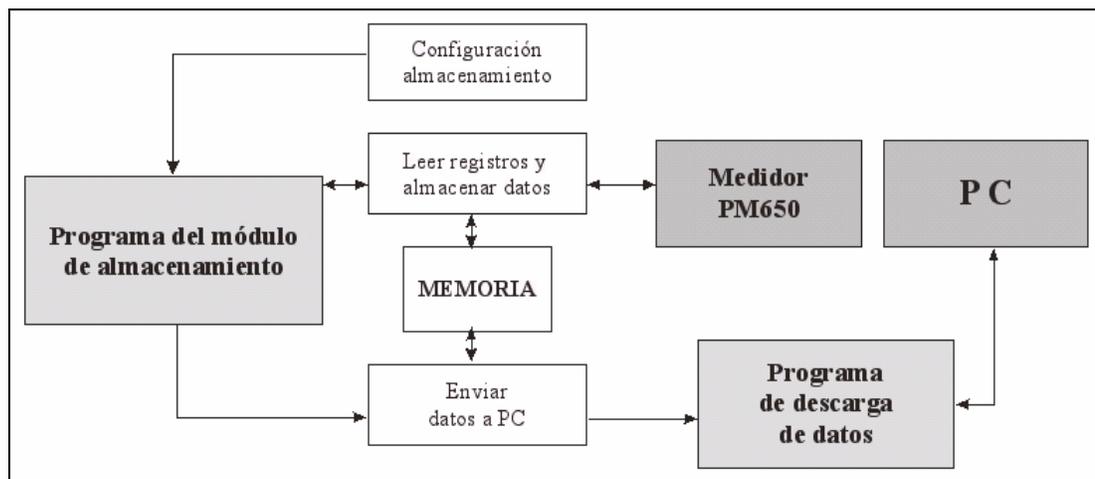


Figura 3.1 Diagrama a bloques del programa.

3.2. PROGRAMA DEL MÓDULO DE ALMACENAMIENTO

Las funciones principales del programa del módulo de almacenamiento son:

- Elección de los registros del medidor PM650 que serán almacenados.
- Cambio del tiempo de muestreo de lectura de los registros del medidor
- Almacenamiento de los datos obtenidos a partir de la lectura a los registros del medidor.
- Envío de los datos provenientes del medidor PM650 almacenados en la memoria del módulo de almacenamiento hacia el puerto serie de una PC.

Estas funciones son implementadas utilizando el diagrama de flujo mostrado en la figura 3.2., las funciones relevantes se explican a continuación de manera más detallada.

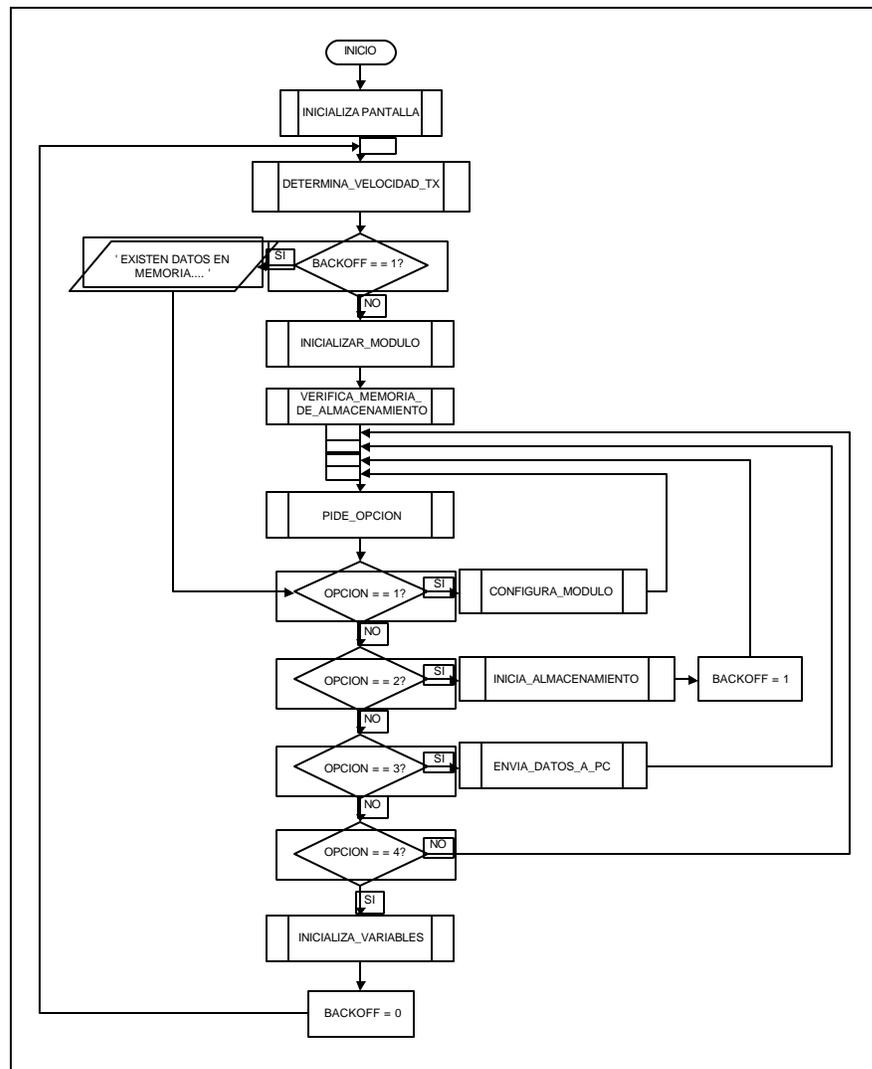


Figura 3.2. Diagrama a bloques del programa del módulo de almacenamiento.

3.2.1. Inicialización del módulo

Esta función es la encargada de configurar el modo de operación del AND491, de enviar un mensaje de bienvenida al usuario indicando el inicio del funcionamiento del módulo de almacenamiento y de asignar valores iniciales a las variables utilizadas en el programa del módulo, entre las cuales se pueden enumerar las que se usan para configurar la adquisición y las que intervienen en el almacenamiento de los datos generados por el medidor PM650.

b1, b2, ..., b10, Son diez variables que determinan si el bloque correspondiente debe ser almacenado, el valor de cada variable es elegido por el usuario en el modo configuración del módulo. Cada uno de los bloques contiene uno o más de los parámetros que usualmente se monitorean para el análisis de una red eléctrica, para mayor detalle ver la tabla 3.1.

backoff, es la variable que el módulo ocupa como bandera para indicar si hay datos de las mediciones en la memoria del módulo. Esta bandera es activada, se le asigna el valor de 1, cuando finaliza la recolección de los datos del medidor y desactivada, valor 0, cuando la opción reinicio del módulo es seleccionada.

baud, Contiene el valor de la velocidad de transmisión de datos expresado en bps⁽¹⁾ y se utiliza en las comunicaciones seriales del programa del módulo. El módulo de almacenamiento soporta tres valores de velocidad de transmisión: 1200, 2400 y 4800. No se utilizan velocidades mayores debido a las pérdidas de datos observadas en pruebas experimentales.

direccion_datos, Variable que indica la dirección donde se almacenan los datos de las mediciones realizadas por el PM650. El valor inicial de ésta variable es 8003H y se incrementa según se realiza el almacenamiento.

opcion, Es una variable que indica el modo en que opera el módulo de almacenamiento; configuración, inicio de almacenamiento, envío de datos a PC o reinicio.

PUERTO, Indica el tipo de comunicaciones que se utiliza, RS-232 o RS-485, con un valor de esta variable de '1' o '0' respectivamente, se asigna al pin 3 del puerto 1 del 8031 que se conecta a la entrada /P13 del multiplexor y demultiplexor utilizados en el submódulo de comunicaciones seriales.

tiempo_sapre, El valor de esta variable es elegida por el usuario e indica cuantos minutos debe esperar el módulo para iniciar una nueva adquisición de los valores de las mediciones de los bloques seleccionados por el usuario.

veces, Indica el número de mediciones que son almacenadas, su valor es el resultado de la división del número total de localidades de memoria, 131,072, entre la sumatoria de los bytes de cada uno de los bloques elegidos, ver tabla 3.1.

dd_1, dd_2, mm_1, mm_2, yy_1, yy_2, Contienen el valor del día, mes y año de inicio y de fin de la adquisición de los parámetros.

⁽¹⁾ número de bits transmitido por segundo.

hh_1, hh_2, mi_1, mi_2, ss_1, ss_2, Su valor representa hora, minutos y segundos en los cuales se inicio y finalizó la adquisición de los parámetros eléctricos. Su valor es adquirido cuando se inicia y finaliza el almacenamiento.

La tabla 3.1. lista el rango y valor inicial de las variables antes mencionadas.

Tabla 3.1. Variables principales del programa del módulo de almacenamiento.

Nombre	Rango	Valor inicial
b1, b2, ..., b10	0: no se almacena el bloque de mediciones. 1: se almacena el bloque de mediciones.	1
baud	1200, 2400, 4800 bps	1200
direccion_datos	8000H a 27FFFH	8000H
opcion	1: Configurar módulo. 2: Iniciar de almacenamiento. 3: Enviar datos a PC. 4: Reiniciar el módulo.	1
PUERTO	0: Comunicaciones RS-485. 1: Comunicaciones RS-232.	0
tiempo_sapre	0, 1, 2, 3, 4,... o 15 minutos.	0
veces	$\text{veces} = \frac{131072}{\sum_{i=1}^{n_i} b_i}$ $\sum_{i=1}^{n_i} b_i$ b_i es valor del bloque, 0 (almacenar) o 1 (no almacenar) n_i es el número de bytes por bloque, ver tabla 3.2.	$\text{veces} = \frac{131072}{76}$ 76 es el número total de bytes en cada muestra si todos los bloques son almacenados. 131072 son el número total de localidades de memoria disponibles para el almacenamiento
dd_1, dd_2	1, 2, o 31	ninguno
mm_1, mm_2	1, 2, o 12	ninguno
yy_1, yy_2	cualquiera	ninguno
hh_1, hh_2	0, 1, 2, ... o 23	ninguno
mi_1, mi_2	0, 1, 2, ... o 59	ninguno
ss_1, ss_2	0, 1, 2, ... o 59	ninguno

3.2.2. Determinación de la velocidad de transmisión

El intercambio de información entre el medidor PM650 y el módulo de almacenamiento se realiza de dos formas:

- Mediante un cable que une el puerto RS-485 del medidor con el puerto RS-485 del módulo de almacenamiento.
- Usando un convertidor RS-232 - RS-485, conectando a su entrada las señales provenientes del puerto RS-485 del medidor y su salida al puerto RS-232 del módulo.

Además, ambas formas de conexión deben considerar que la velocidad de transmisión del módulo debe ser la misma que la del medidor para que se efectúe el intercambio. Para evitar que el usuario tenga la responsabilidad de configurar la velocidad de transmisión del módulo de comunicaciones se implementó la rutina de determinación de la velocidad de transmisión, la cual se encarga de configurar la velocidad de transmisión del módulo de almacenamiento dependiendo de la utilizada por el medidor PM650.

El módulo de almacenamiento funciona correctamente cuando trabaja con 1200, 2400 o 4800 bps, aunque las velocidades de transmisión que puede utilizar el PM650 son 1200, 2400, 4800, 9600 y 19200 bps. Por esta razón, el usuario debe fijar el valor de velocidad de transmisión del PM650 considerando el rango utilizado por el módulo.

La rutina de determinación de la velocidad de transmisión verifica seis posibles combinaciones generadas por el tipo de estándar de comunicaciones, RS-232 o RS-485, y la velocidad de transmisión utilizado por el medidor, 1200, 2400 o 4800. Para cada una de las combinaciones, la rutina envía un paquete de petición de lectura al registro 1001 y espera un dato. Si el 8031 detecta un dato en la señal de recepción, RX, significa que la velocidad empleada por el 8031 es la misma que la del medidor y que el puerto de comunicaciones elegido es el correcto.

Lo anterior se resume en el diagrama de flujo de la rutina de determinación de velocidad de transmisión mostrado en la figura 3.3. Si la función **DETERMINA_VELOCIDAD_TX** envía el paquete de petición usando las seis combinaciones del estándar de comunicaciones y de velocidad de transmisión, el módulo entra en un ciclo infinito para permitir al usuario verificar conexiones o bien configurar al medidor en el rango de velocidad de transmisión permitidas.

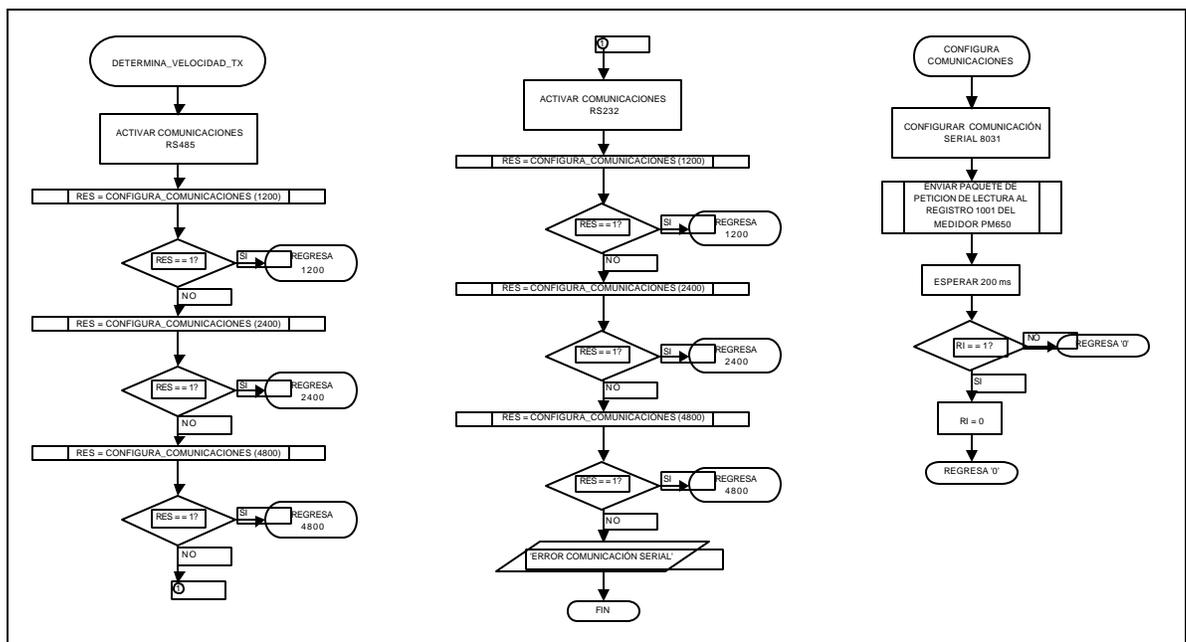


Figura 3.3. Diagrama de flujo del algoritmo de determinación de velocidad de transmisión.

3.2.3. Envío de paquete de petición

Como se explicó en la sección 1.3.2, para leer los registros del medidor es necesario utilizar la función 03 de lectura del protocolo Modbus, la cual se implementa mediante la función *envío de paquete de petición*, figura 3.4; se envían la **DIRECCIÓN DEL REGISTRO DE INICIO** y el **NÚMERO DE REGISTROS** que deben ser leídos, argumentos de entrada de la función de envío de paquete de petición.

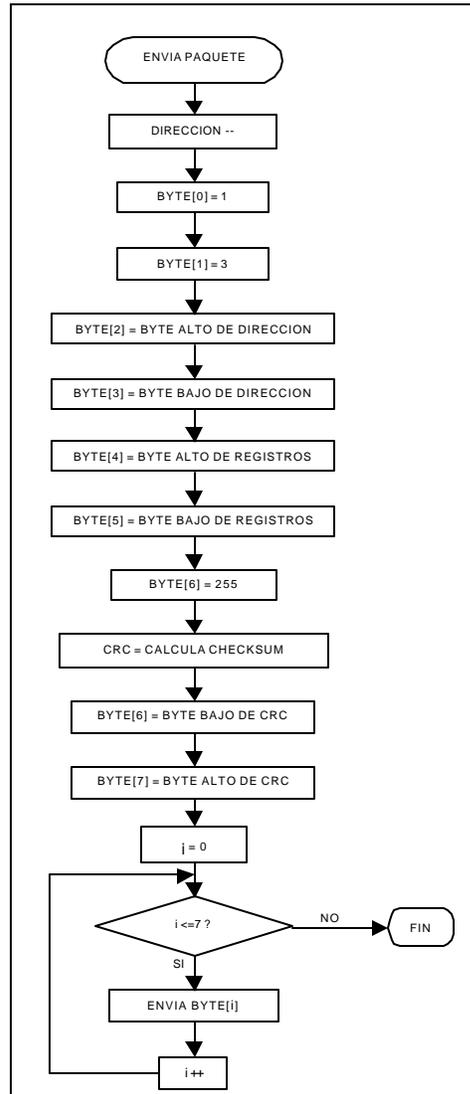


Figura 3.4 Diagrama de flujo del algoritmo de envío de paquete de petición de lectura.

3.2.4. Verificación de memoria de almacenamiento

Después de configurar las comunicaciones seriales, el programa pregunta al usuario si desea hacer una verificación a la memoria de almacenamiento, el cual consiste en escribir datos a las 32,768 localidades de cada una de las cuatro memorias RAM que integran la memoria de datos del medidor PM650; después de la escritura de este datos se hace una lectura y si en todas

las 32,768 localidades de cada memoria se lee el valor escrito anteriormente se envía el mensaje de “MEMORIA N OK”, de otro modo se envía el mensaje “ERROR EN MEMORIA” y el módulo entra a un ciclo infinito para que el usuario sustituya la memoria dañada.

3.2.5. Petición de opción

Después de configurar las comunicaciones seriales y de examinar la memoria de almacenamiento, el programa del módulo espera a que el usuario elija una de las cuatro opciones siguientes:

- Configurar el módulo. En esta opción el usuario modifica los valores de la forma de almacenamiento de los datos provenientes del medidor PM650.
- Iniciar almacenamiento de mediciones. Cuando se elige esta opción, el módulo lee los registros seleccionados y almacena los datos del PM650 en la memoria del módulo de almacenamiento.
- Enviar datos a la PC. Traslada los datos almacenados en la memoria hacia el puerto serie de la PC para su almacenamiento en un archivo de texto ASCII con extensión .txt.
- Reiniciar el módulo. El programa asigna valores iniciales a las variables utilizadas y comienza la ejecución del programa desde la primera línea.

3.2.5.1. Configuración del módulo

En este modo el programa permite al usuario elegir los bloques que serán almacenados y el tiempo de muestreo de las mediciones. Las mediciones que pueden ser almacenadas se agrupan en diez bloques, los cuales el usuario puede decidir almacenar o no. La tabla 3.2 muestra los diez bloques y los parámetros eléctricos que considera cada uno de ellos, ver tabla 2.1 para consultar la dirección del registro de cada parámetro.

Tabla 3.2 Descripción de los bloques de mediciones.

Bloques	Parámetros incluidos	No. bytes por bloque
1	Frecuencia	2
2	Corrientes en cada una de las tres fases (A, B, C y neutro).	8
3	Voltajes para las tres combinaciones de fases (A a B, B a C y C a A).	6
4	Voltajes entre el neutro y cada una de las tres fases.	6
5	Factor de potencia real para cada una de las tres fases.	6
6	Potencia real para cada una de las tres fases.	6
7	Potencia reactiva para cada una de las tres fases.	6
8	Distorsión armónica total de la corriente para cada una de las tres fases.	6
9	Distorsión armónica total del voltaje para cada una de las tres fases.	6
10	Energía del total de las tres fases: energía aparente, real y reactiva.	24

El tiempo de muestreo que puede elegir el usuario es: *automático*, *1 minuto*, *2 minutos*, *3 minutos* hasta *15 minutos*, este tiempo indica el tiempo que transcurre entre el inicio de una muestra y la siguiente. Los valores iniciales de los bloques es que todos se almacenaran y el tiempo de muestreo es automático.

3.2.5.2. Inicio de almacenamiento de parámetros eléctricos

Esta rutina, cuyo algoritmo se muestra en la figura 3.5, efectúa la lectura de los registros del medidor contemplados en los bloques seleccionados, almacena los datos obtenidos, espera el tiempo de muestreo, vuelve a leer y almacena los datos hasta que se cumple el número de muestras calculadas para ocupar toda la memoria de almacenamiento.

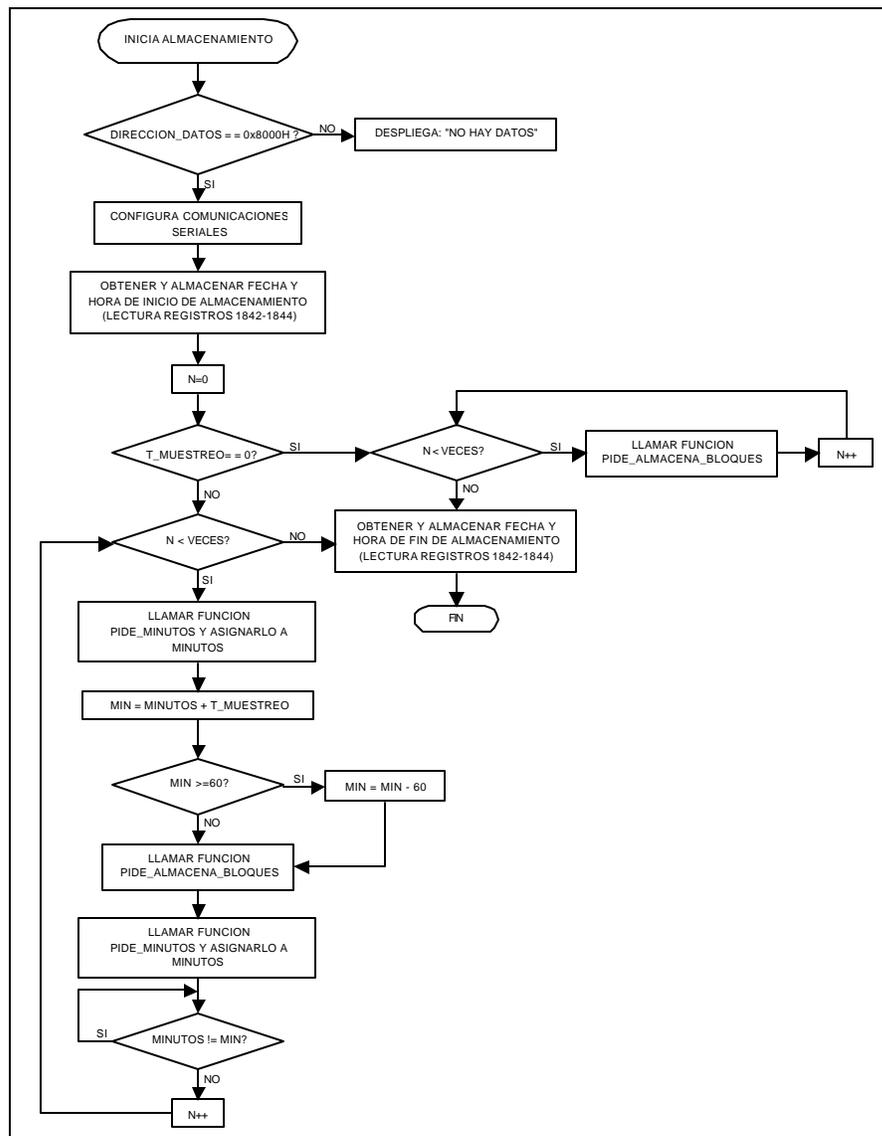


Figura 3.5. Algoritmo de la rutina de inicio de almacenamiento.

Cuando la respuesta a la petición de lectura de n registros llega al módulo de almacenamiento, el programa verifica si hubo errores de transmisión cotejando el CRC recibido con un CRC calculado a partir de los datos recibidos, si un error de transmisión se detecta se vuelve a pedir el paquete.

Los registros 1842, 1843 y 1844 del medidor PM650 contienen los valores de la fecha y hora [2]; estos registros son leídos antes y después del inicio de almacenamiento. Los valores obtenidos de los registros son considerados hora y fecha de inicio, hora y fecha de fin de almacenamiento, formando parte de la información enviada a la PC.

3.2.5.3. Envío de datos a la PC

Los datos almacenados son enviados al puerto serial de la PC cuando se elige esta opción, si no hay ningún dato en la memoria aparece el mensaje de “No existen datos para enviar” y regresa a pedir una opción. Si hay datos para enviar, el módulo configura las comunicaciones RS-232 a una velocidad de transmisión de 4800 bps. Los datos son enviados en el siguiente orden:

3. Datos de fecha y hora de inicio del almacenamiento, el programa envía las variables que contienen los datos del día, mes, año, hora, minutos y segundos del inicio del almacenamiento.
 4. Datos de fecha y hora de fin del almacenamiento, contiene las variables de día, mes, año, hora, minutos y segundos del fin del almacenamiento.
 5. Valor de los bloques seleccionados, valor de cada bloque ‘0’ o ‘1’ dependiendo si se almacenó o no.
 6. Tiempo de muestreo, número de minutos de espera entre cada almacenamiento.
 7. Número de muestras almacenadas, expresado en 4 bytes enviando primero el byte más significativo.
6. Datos de las mediciones realizadas por el medidor.

3.3. PROGRAMA DE DESCARGA DE DATOS

El programa de descarga de datos fue desarrollado utilizando el lenguaje Labview de National Instruments, este lenguaje gráfico tiene la ventaja de implementar aplicaciones de instrumentación de manera sencilla. En esta sección se describe la función del programa de descarga, su algoritmo, su panel frontal y diagrama a bloques a partir de sus subVI's (funciones dentro del programa principal) más importantes.

El programa de descarga de datos tiene la función de recibir los datos provenientes del módulo de almacenamiento, procesar la información del almacenamiento y graficar los datos para un posterior análisis. Las tareas que realiza el software de descarga son las siguientes:

- Configura el puerto serial de la PC para comunicarse con el medidor utilizando una velocidad de transmisión de 4800 bps.
- Establece comunicación con el módulo de almacenamiento a través de los puertos RS-232 de la PC y del módulo.
- Espera a que el módulo éste en modo *envío de datos a PC*.
- Recibe los datos enviados por el módulo de almacenamiento.
- Ordena y almacena los datos recibidos en un archivo de texto ASCII con extensión .txt en la ruta seleccionada por el usuario.
- Muestra las gráficas de las mediciones recibidas para facilitar la interpretación de los datos.

3.3.1. Algoritmo del programa de descarga

La función principal del programa de descarga es obtener los datos contenidos en la memoria del módulo de almacenamiento y lo hace en el siguiente orden:

El programa inicializa el puerto serial de la PC a una velocidad de 4800 bps. Debido a que no se utiliza un protocolo determinado, el programa de descarga envía el dato 0xFF y durante 1 segundo espera recibir un 0xFF, si esto ocurre indica que el módulo de almacenamiento envió la información. El software de descarga lee y almacena en una cadena los bytes que llegan al puerto serie hasta que se detecta un 0xFF; cuando esto sucede se procesa la cadena de bytes recibidos usando el subVI *inf_conf.vi*, el cual procesa los bytes enviados y los divide en *fecha y hora de fin*, *fecha y hora de inicio*, *tiempo de muestreo*, *total de muestras*, *tiempo de inicio*, *bytes_muestra*, *b1*, *b2*, *b3*, *b4*, *b5*, *b6*, *b7*, *b8*, *b9* y *b10*, ver sección 3.2.1.

Se lee un número de bytes, un total de n veces según el valor de *bytes_muestra*; dependiendo del valor de *total de muestras*, cada vez que se leen los n bytes se almacenan en un arreglo de *bytes_muestra* por *total de muestra* y al finalizar se tiene una matriz donde las filas corresponden a cada una de las muestras tomadas. Es necesario aclarar que los datos en este punto no pueden ser consultados debido a que son números de 16 bits, cuando los datos deben ser números de 32 bits. Para hacer este cambio se deben tomar los datos de cada fila y agruparlos en pares: tomar el primer byte, de cada par, multiplicarlo por 256 y sumarlo al segundo byte de cada par, con esto obtendremos el valor de cada parámetro. Al término de este proceso se obtiene una matriz cuyas columnas son cada uno de los parámetros y las filas cada una de las muestras tomadas.

El subVI *indices.vi* crea un arreglo de 29 elementos, utilizando los valores de *b1*, *b2*, *b3*, *b4*, *b5*, *b6*, *b7*, *b8*, *b9* y *b10*, en donde cada elemento corresponde a uno los 29 parámetros que se pueden almacenar, esto sirve para referenciar las columnas a cada uno de los parámetros.

El subVI *archivo de salida.vi* toma los datos obtenidos, convertidos en números de 32 bits, los almacena en un archivo de texto ASCII cuya ruta es especificada por el usuario. Además de tener los valores de las mediciones, este archivo contiene la información del almacenamiento: el tiempo de muestreo, la fecha y hora de inicio y fin del almacenamiento y el total de muestras adquiridas. Los datos de las mediciones son ordenados en forma de tabla donde la primer

columna corresponde a la hora en que fue realizada la muestra, las siguientes columnas corresponden a cada uno de los parámetros almacenados y las filas son cada una de las muestras adquiridas por el módulo de almacenamiento.

Por último, se elige el arreglo que contiene los datos de las mediciones y se grafican los datos correspondientes dependiendo de la gráfica que el usuario haya elegido.

Cada uno de los pasos explicados anteriormente se muestran en el diagrama de flujo del programa de descarga de la figura 3.6.

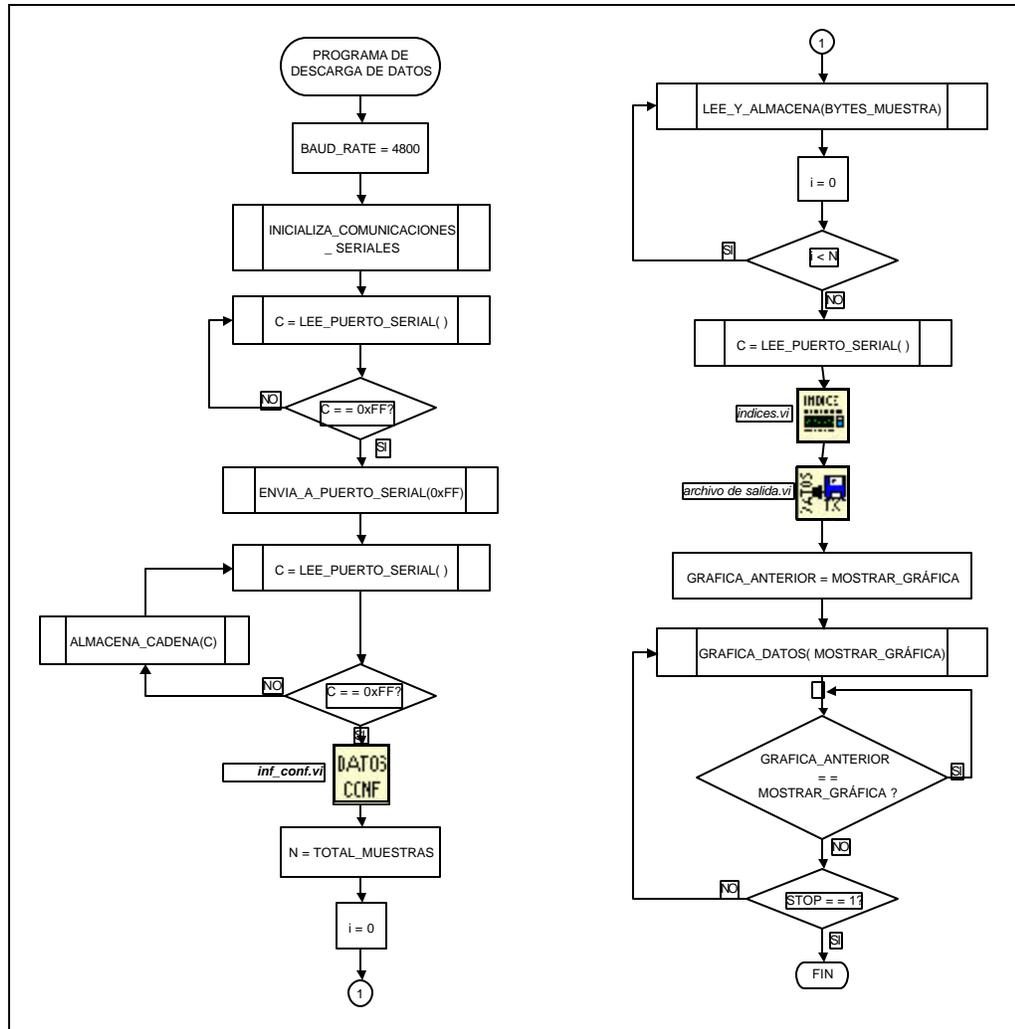


Figura 3.6. Diagrama de flujo del programa de descarga.

A continuación se describen los subVIs mostrados en el diagrama a bloques del software de descarga utilizando sus conectores e iconos correspondientes.

3.3.2. *inf_config.vi*

Recibe 26 bytes del puerto '**port number**' y lo añade a un carácter ya recibido '**c1**', esta nueva cadena se divide en '**fecha y hora de inicio**', '**fecha y hora de fin**', '**tiempo de muestreo**', '**total de muestras**', '**b1**', '**b2**', '**b3**', '**b4**', '**b5**', '**b6**', '**b7**', '**b8**', '**b9**' y '**b10**'. El submódulo realiza la sumatoria de los productos de la multiplicación del número de bytes por bloque por '0' o '1', dependiendo si se almacenó o no el bloque; el resultado de la sumatoria lo asigna a '**bytes bloque**'. Las variables descritas anteriormente se muestran en la figura 3.7. donde se muestra el icono y conectores de *inf_config.vi*.

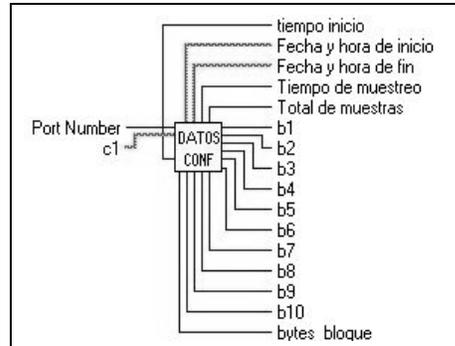


Figura 3.7. Icono y conectores del subVI *inf_config.vi*.

3.3.3. *indices.vi*

El subVI *indices.vi*, figura 3.8, muestra su icono y conectores; éste submódulo construye un arreglo de 29 elementos donde cada uno de ellos contiene un índice que corresponde a la columna de la matriz de los datos de las mediciones. Estos índices dependen de los valores de '**b1**', '**b2**', '**b3**', '**b4**', '**b5**', '**b6**', '**b7**', '**b8**', '**b9**' y '**b10**', si el valor de cualquier bloque es 1 se incrementa el índice; el primer elemento del arreglo tiene un valor de 0.

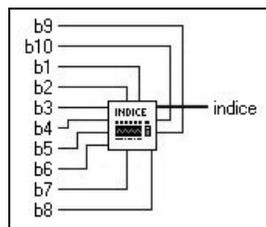


Figura 3.8. Icono y conectores del subVI *indices.vi*.

3.3.4. *archivo de salida.vi*

Este submódulo almacena los datos de las mediciones y la información del almacenamiento en un archivo de texto ASCII con extensión **.txt** en la ruta: '**ruta del reporte generado**'. El archivo de texto contiene la información del almacenamiento: total de muestras, tiempo de muestreo, fecha y tiempo de inicio y fin de almacenamiento y los datos de

las mediciones con sus respectivos nombres para cada uno de los parámetros almacenados. El icono y los conectores de este submódulo se muestran en la figura 3.9.

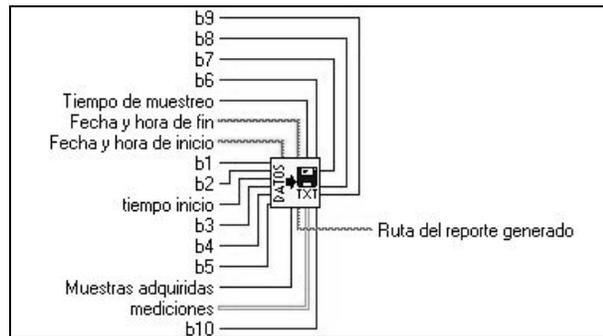


Figura 3.9. Icono y conectores del subVI *archivo de salida.vi* .

3.3.5. modulo almacenamiento.vi

Es el programa de descarga de datos del módulo de almacenamiento contiene el diagrama a bloques (sección 3.3.1), panel frontal, icono y conectores. Su icono se muestra en la figura 3.10, los conectores no son mostrados ya que *modulo almacenamiento.vi* se considera un VI y no un subVI.

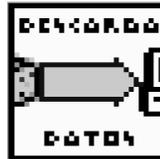


Figura 3.10. Icono del VI *modulo almacenamiento.vi*.

El panel frontal del programa de descarga está compuesto por:

Identificación. Muestra el nombre del programa.

Control de Puerto serial y botón de paro. El control de puerto permite al usuario seleccionar el número de puerto serial y el botón de paro finaliza la ejecución del programa de descarga.

Indicadores de adquisición y procesamiento. Muestran el estado de la adquisición de los datos así como el procesamiento de los mismos.

Datos del historial. Contienen los indicadores que muestran la fecha y hora de inicio, fecha y hora de fin, total de muestras y tiempo de muestreo.

Ruta del reporte generado. Es un indicador que muestra la ruta de ubicación del archivo, resultado de un cuadro de diálogo donde se pide dicha ruta al usuario.

Control de gráfica solicitada. Ubicado en la parte superior derecha e indica al software de descarga el parámetro que debe ser graficado.

Gráfica. Es un control que contiene las opciones gráficas disponibles en el programa de descarga, las opciones se listan en la tabla 3.3.

Tabla 3.3. Opciones de gráficas del programa de descarga.

Opción	Parámetros gráficos
Frecuencia.	Frecuencia.
Corriente monofásica.	Corriente fase A. Corriente fase B. Corriente fase C.
Voltaje bifásico.	Voltaje fase A a B. Voltaje fase B a C. Voltaje fase C a A.
Voltaje monofásico.	Voltaje fase A a neutral. Voltaje fase B a neutral. Voltaje fase C a neutral.
Factor de potencia, valores con signo.	Factor de potencia fase A. Factor de potencia fase B. Factor de potencia fase C.
Factor de potencia, valores absolutos.	Factor de potencia fase A Factor de potencia fase B Factor de potencia fase C
Potencia real.	Potencia real fase A. Potencia real fase B. Potencia real fase C.
Potencia reactiva.	Potencia reactiva fase A. Potencia reactiva fase B. Potencia reactiva fase C.
THD/thd de corriente.	THD/thd de corriente fase A. THD/thd de corriente fase B. THD/thd de corriente fase C.
THD/thd de voltaje.	THD/thd de voltaje fase A. THD/thd de voltaje fase B. THD/thd de voltaje fase C.
Energía total aparente.	Energía total aparente.
Energía total real.	Energía total real.
Energía total reactiva.	Energía total reactiva.

La figura 3.11. muestra cada de los componentes que integran al panel frontal del software de descarga. El funcionamiento y operación del software de descarga se detalla en el anexo D “Manual de usuario del Módulo de almacenamiento”.

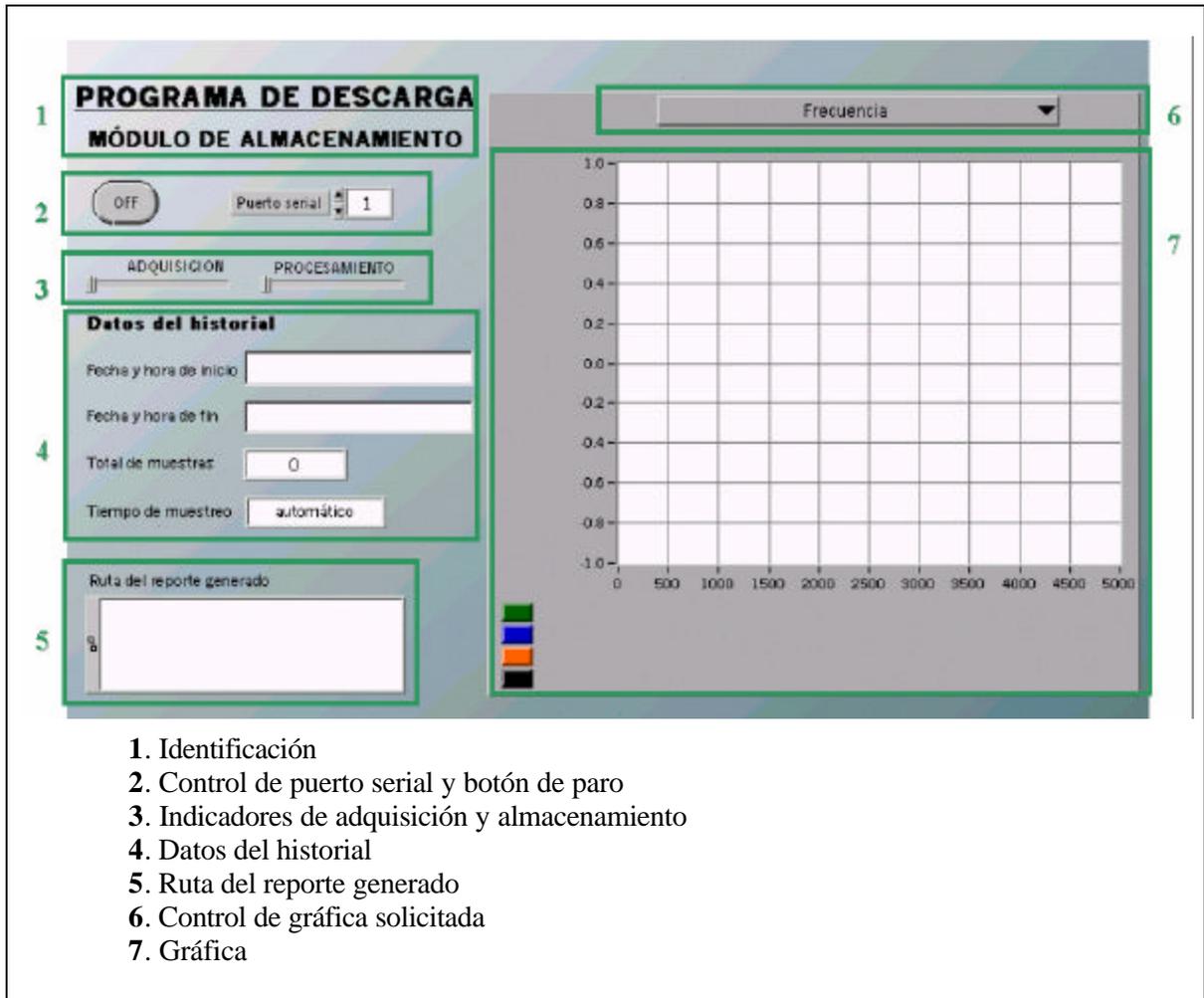


Figura 3.11. Componentes del panel frontal del programa de descarga.

CAPÍTULO 4. RESULTADOS Y CONCLUSIONES

4.1. RESULTADOS

En este capítulo se presentan los resultados de las pruebas de campo que se realizaron al módulo de almacenamiento incluyendo la descripción del prototipo del módulo de almacenamiento, los datos y gráficas obtenidas durante el proceso de verificación del módulo.

4.1.1. Prototipo del módulo de almacenamiento

El prototipo se construyó con la técnica de *Wire Wrap*⁽¹⁾, lo cual permitió rediseñar el prototipo durante todo el proceso de diseño y construcción. En la figura 4.1, se observa el prototipo final, el cual cuenta con dos puertos seriales; RS232 y RS485, el módulo mínimo del 8031, las memorias de almacenamiento, el visualizador, el teclado y demás componentes listados en el esquemático del anexo C.

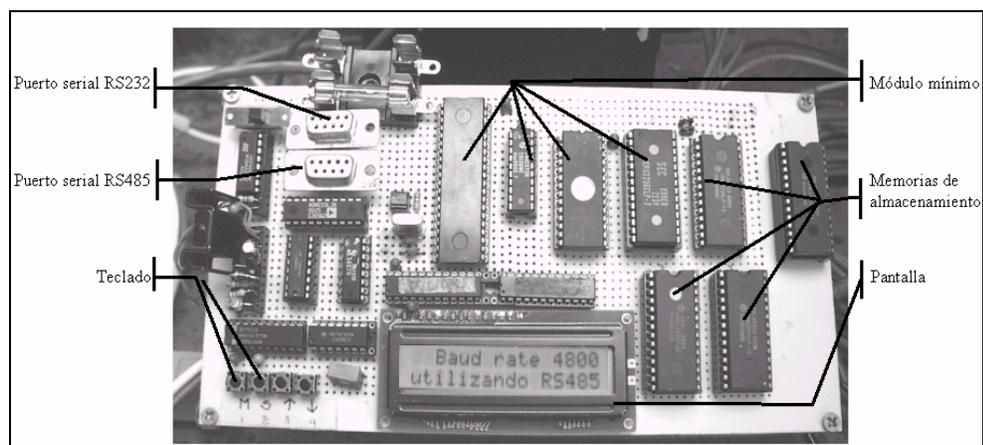


Figura 4.1. Prototipo del módulo de almacenamiento y sus componentes.

⁽¹⁾ Técnica empleada para construir circuitos electrónicos.

4.1.2. Prueba de funcionamiento del módulo de almacenamiento

Las pruebas de funcionamiento del módulo de almacenamiento se realizaron bajo el siguiente procedimiento:

- Conectar el medidor PM650 a la red eléctrica.
- Conectar el módulo de almacenamiento al medidor PM650.
- Configurar el módulo de almacenamiento.
- Iniciar almacenamiento de datos del medidor PM650.
- Enviar datos almacenados a la PC.

Estos pasos fueron realizados en dos ocasiones cambiando el tiempo de muestreo de los datos almacenados, automático y 1 minuto; los resultados se presentan en las secciones 4.1.2.1 y 4.1.2.2. Los pasos del procedimiento son descritos a continuación.

a. Conexión del medidor PM650 a la red eléctrica

Las entradas de voltajes y corrientes del medidor PM650 se conectan a la red eléctrica en el mismo punto donde CFE tiene conectado un medidor de potencia para registrar el consumo del usuario. Para las pruebas del módulo de almacenamiento, el medidor fue conectado a la red eléctrica de la Universidad Tecnológica de la Mixteca (UTM), ver anexo D para mayor detalle; en cuyo diagrama unifilar, mostrado en la figura 4.2, se observa que la línea eléctrica que CFE distribuye pasa por un transformador de voltaje que convierte los 13.8 kV a 120 V, a la salida de éste se encuentran instalados, el medidor CFE, un interruptor termomagnético principal y 6 interruptores secundarios conectados en paralelo.

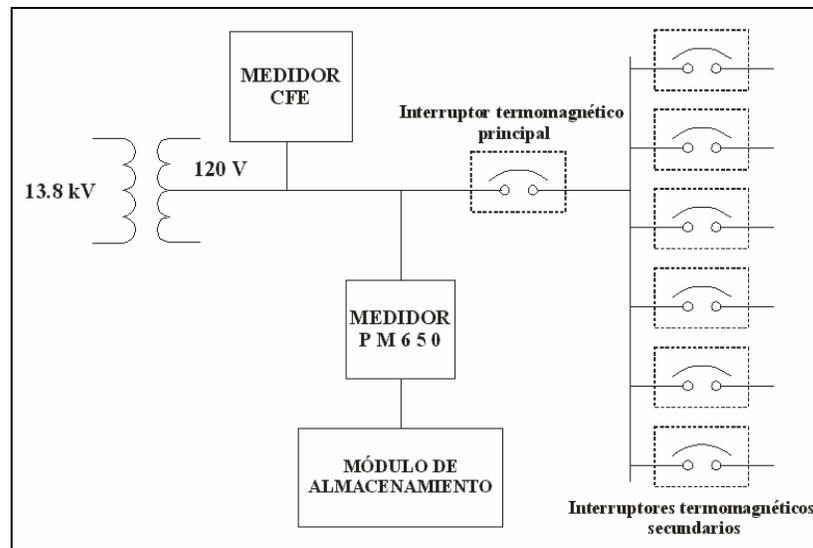


Figura 4.2. Diagrama unifilar de la red eléctrica de la UTM.

En la fotografía de la figura 4.3 se observa la instalación del medidor PM650 y el módulo de almacenamiento a la red eléctrica de la UTM para la realización de las pruebas presentadas en este documento.

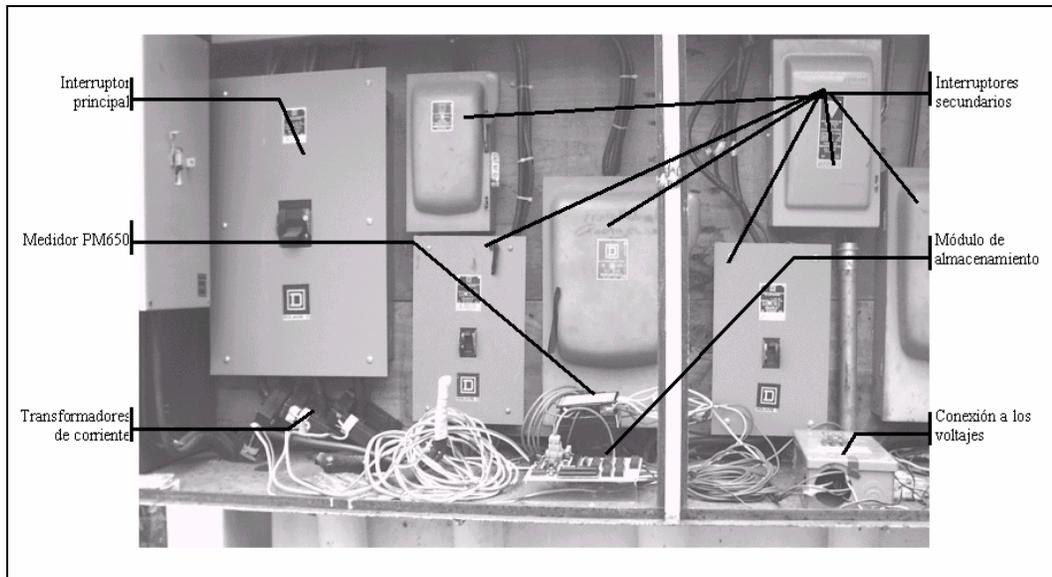


Figura 4.3. Módulo de almacenamiento instalado en la red eléctrica de la UTM.

Las entradas de voltajes del medidor PM650 se conectan a las tres fases y al neutro de voltaje de la red eléctrica utilizando la caja de interruptores termomagnéticos, figura 4.4. Las entradas de corrientes se conectan utilizando los transformadores de corriente (TC, o donas de corriente) a las salidas del interruptor electromagnético principal, figura 4.5.

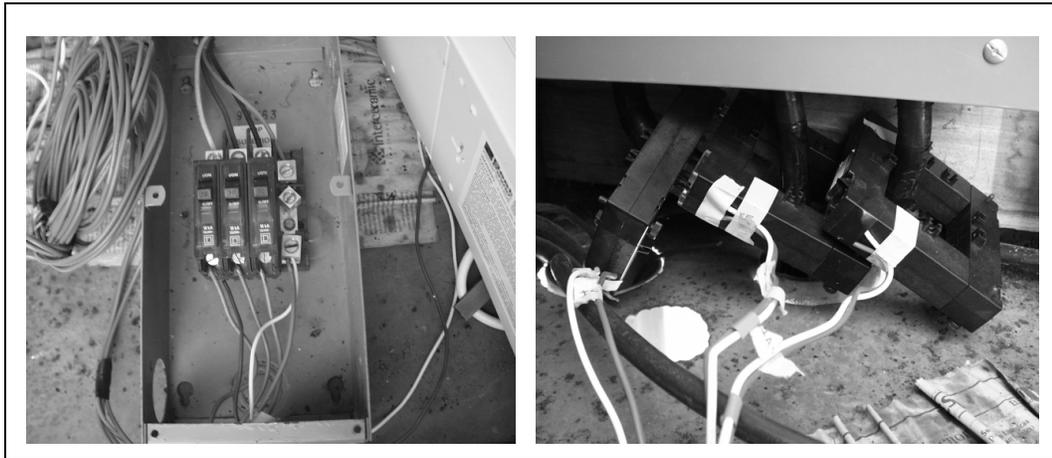


Figura 4.4. Conexión de los voltajes y los transformadores de corriente.

b. Conexión del módulo de almacenamiento al medidor PM650

La figura 4.5 muestra como las salidas y entradas de comunicaciones RS485 del medidor PM650 se conectan al puerto de comunicaciones RS485 del módulo de almacenamiento, utilizando un cable UTP con la configuración descrita en la figura 1.13. El convertidor de voltaje

se conecta a la fase A y el neutro y su salida se conecta a la entrada de voltaje del módulo de almacenamiento. Se enciende el módulo de almacenamiento utilizando el interruptor del prototipo; si la conexión es bien realizada y no existe algún corto circuito, el led del módulo se enciende indicando que el módulo se encuentra energizado.

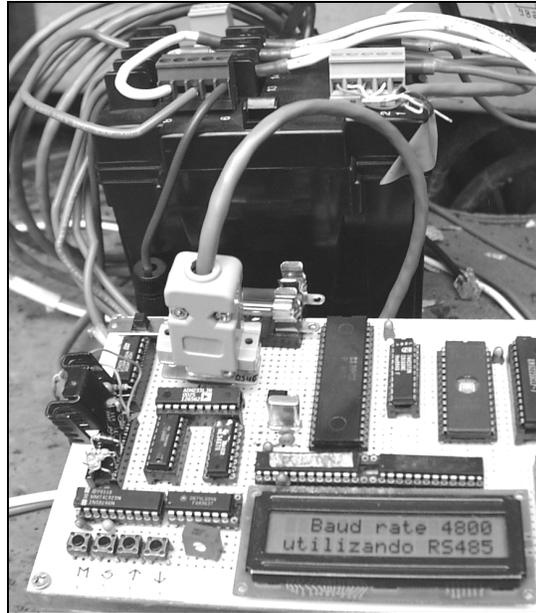


Figura 4.5. Conexión de comunicaciones RS485 del módulo de almacenamiento a las del PM650.

c. Configuración del módulo de almacenamiento

Después de la inicialización del módulo de almacenamiento⁽²⁾ se elige el modo **Configuración** y se seleccionan los parámetros que se van almacenar y el tiempo de muestreo.

d. Inicio del almacenamiento de datos

Habiendo configurado el almacenamiento de los parámetros se elige el modo **Inicia muestreo**, durante el cual el visualizador del módulo del almacenamiento indica el número de veces que restan al almacenamiento; en el caso de tiempo de muestreo automático, o el tiempo restante si el tiempo de muestreo no es automático, ver figura 4.6.

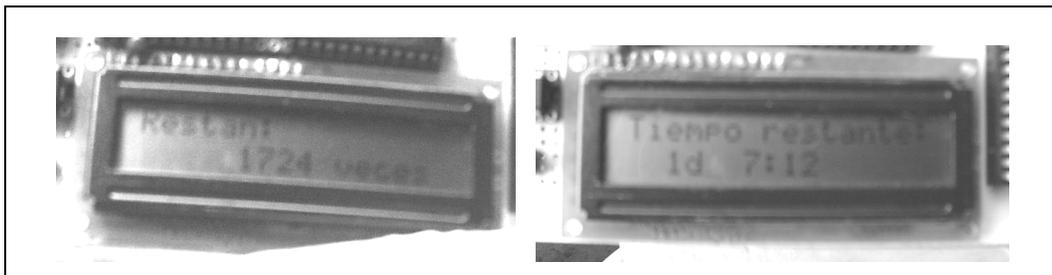


Figura 4.6. Mensaje durante el almacenamiento en un tiempo automático y no automático.

⁽²⁾ Más detalles en manual de usuario anexo D.

e. Envío de datos almacenados a la PC

Cuando concluye el almacenamiento de los datos, el módulo está en posibilidades de enviar los datos almacenados hacia el puerto serie de la PC. Para esto, el módulo de almacenamiento debe estar en modo **Envía dato a PC**, el puerto de comunicaciones RS232 debe estar conectado al puerto serie de la PC con un cable serial RS232 y el programa de descarga de datos debe estar activo.

Los datos son recibidos a través del puerto serial, son adquiridos y procesados automáticamente por el programa de descarga de módulo de almacenamiento. Al finalizar la adquisición y procesamiento el programa de descarga se pide al usuario la ruta para almacenar los datos en un archivo de texto ASCII con extensión .txt.

4.1.2.1. Ejemplo 1

Realizada con un tiempo de muestreo automático y seleccionando los bloques de los parámetros más utilizados en el análisis eléctrico: frecuencia, corriente monofásica, voltaje monofásico, distorsión armónica de voltaje y factor de potencia. La duración aproximada del almacenamiento fue de 34 minutos, iniciando a las 18:44:48 horas y culminando a las 19:18:40 horas del mismo día. Un extracto del archivo de texto que se obtuvo de esta primer prueba, se muestra en la tabla 4.1.

GRÁFICA DE FRECUENCIA

El rango establecido por CFE del valor de la frecuencia de la señal eléctrica distribuida es de 60 ± 4.8 Hz; en la figura 4.7 se observan los valores de frecuencia, los cuales se encuentran dentro del rango sin observar un cambio notable o brusco.

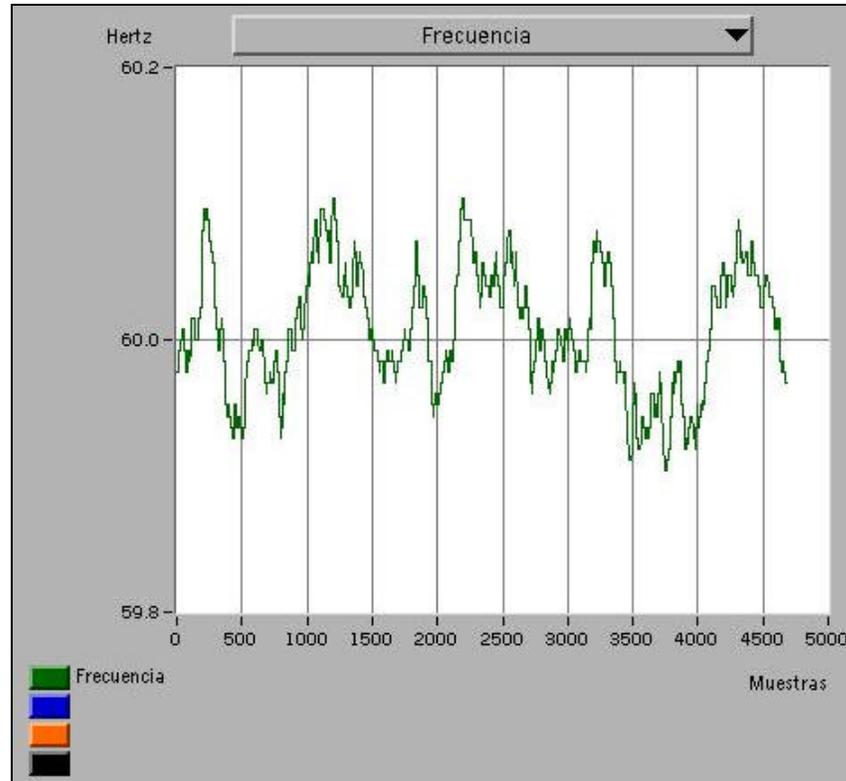


Figura 4.7. Gráfica de los valores de frecuencia del ejemplo 1.

GRÁFICA DE CORRIENTES

El análisis de los valores de las corrientes sirve para verificar el balanceo de las fases, esto se determina evaluando el valor de corriente de neutro, cuando el valor de éste tienda a cero indica que las fases tienden a estar balanceadas. La gráfica de los valores de las corrientes, figura 4.8, muestra que las fases se encuentran desbalanceadas en el rango de la muestra 2500 a la 3000; el valor de la corriente de neutro presenta los valores mínimos y si se observan los valores y el comportamiento de las corrientes de las fases en lo que se observa que estos se incrementan o decrecientan conjuntamente. Para el rango de la muestra 1 a la 1500, los valores para cada una de las fases no cambian de la misma forma por lo que el valor para la corriente del neutro presenta valores máximos.

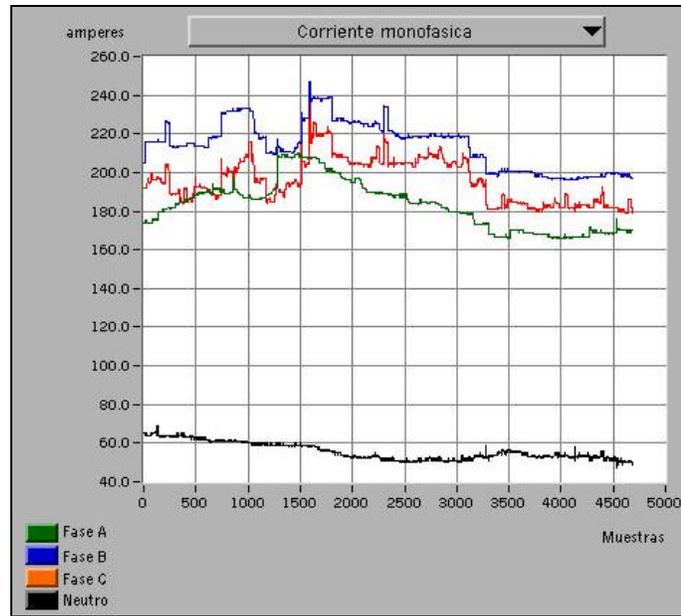


Figura 4.8. Gráfica de los valores de la corriente monofásica del ejemplo 1.

GRÁFICA DE VOLTAJE MONOFÁSICO DE LAS TRES FASES

Los valores del voltaje en cada fase son analizados para determinar si se encuentran dentro del rango establecido por CFE: 120 ± 12 V; para los datos de esta medición estos valores se encuentran en el rango de 126 a 118 V. Como se observa en la figura 4.9, el voltaje en cada una de las fases presenta un decremento en su valor, esto puede ser provocado a que en el lapso que duró el almacenamiento la línea de distribución de CFE es disminuida por el incremento en la carga general a la línea principal (encendido de alumbrado público, iluminación casa-habitación, etc).

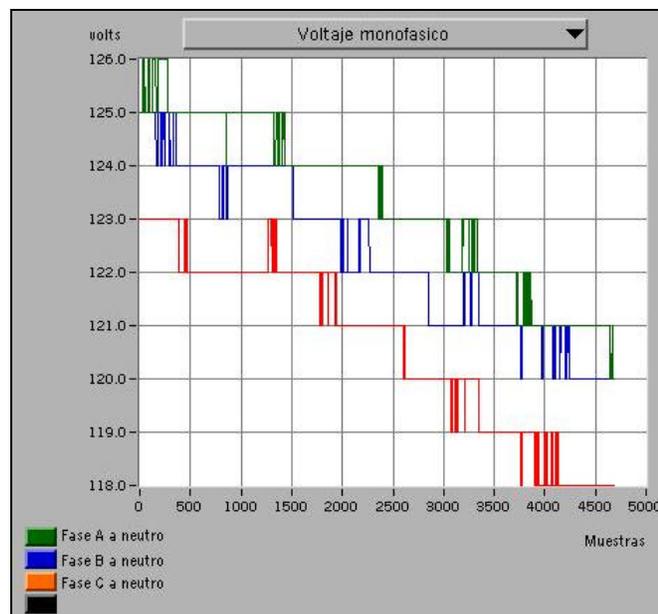


Figura 4.9. Gráfica del voltaje monofásico del ejemplo 1.

GRÁFICAS DE FACTOR DE POTENCIA

El programa de descarga proporciona dos formas de gráfica del factor de potencia (FP): con valores con signo o valores absolutos. La gráfica de la figura 4.10, correspondiente al valor absoluto de FP, muestra que el FP se encuentra en niveles muy cercanos a 0.9, lo cual es aceptable.

Los valores que presenta la gráfica en la figura 4.11, son los valores con signo del factor de potencia. Estos valores muestran que tipo de cargas fueron las que ocasionaron en mayor medida el factor de potencia, si el signo es positivo se trata de cargas capacitivas y negativo si son cargas inductivas. La fase C muestra mayor carga capacitiva que las dos restantes, en las cuales no se observa una carga dominante.

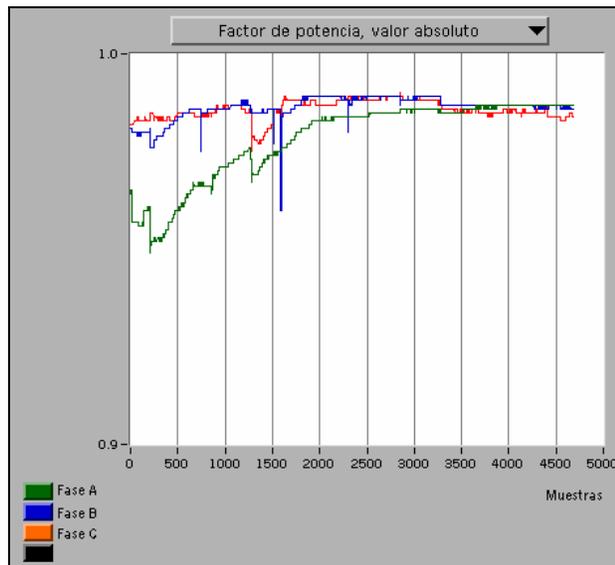


Figura 4.10. Gráfica de los valores absolutos del factor de potencia del ejemplo 1.

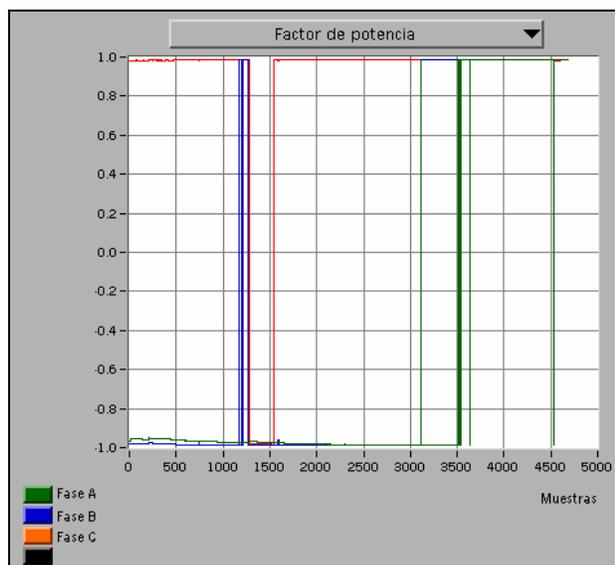


Figura 4.11. Gráfica de los valores con signo del factor de potencia del ejemplo 1.

GRÁFICA DE DISTORSION ARMÓNICA DE VOLTAJE

La norma IEEE 519 ‘Recomendaciones Prácticas y Requerimientos para el Control de armónicas en Módulos Eléctricos de Potencia’, especifica un límite de THD del 5% para módulos que manejen 69KV o menos. Los valores de THD para cada uno de los voltajes, mostrados en la gráfica de la figura 4.12, tienen un rango que va desde el 3% al 4.5%, lo cual indica que el módulo se encuentra dentro del límite admitido por la norma IEEE 519.

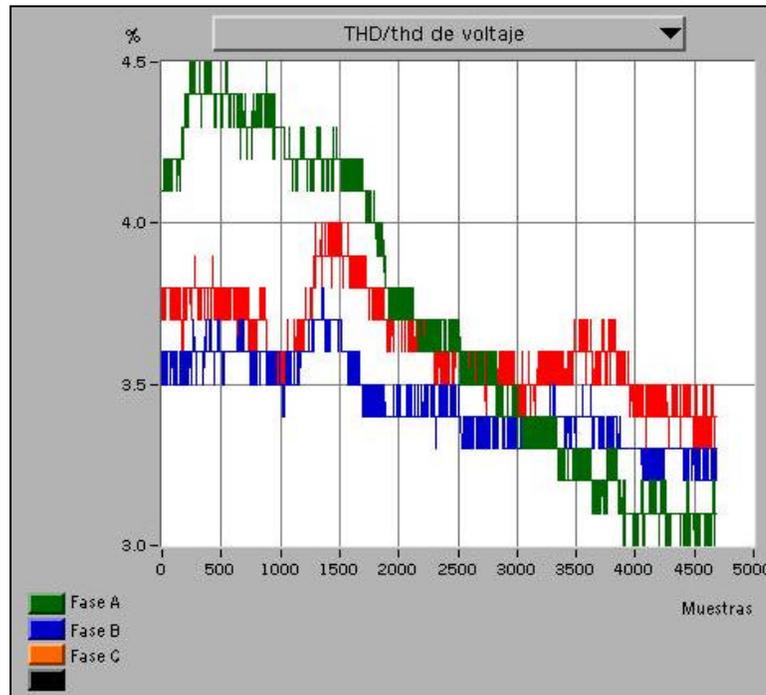


Figura 4.12. Gráfica de los valores de distorsión armónica del voltaje del ejemplo 1.

4.1.2.2. Ejemplo 2

Esta segunda prueba se realizó con un tiempo de muestreo de 1 minuto y seleccionando los bloques de frecuencia, corriente monofásica, voltaje monofásico, distorsión armónica de voltaje y factor de frecuencia. La tabla 4.2, muestra el encabezado y las primeras muestras extraídas del archivo resultado de la descarga de datos.

Tabla 4.2. Archivo que almacena los datos de la segunda medición.

SAPRE: Sistema de almacenamiento de parámetros de la red eléctrica														
Fecha y hora de inicio de almacenamiento: 4/4/2002 19: 45: 7														
Fecha y hora de fin de almacenamiento: 8/4/2002 1: 46: 0														
Tiempo de muestreo: 1 minuto														
Total de muestras: 4681														
Hora de almacenamiento	Frecuencia (Hz)	Corriente de Fase A (Amperes)	Corriente de Fase B (Amperes)	Corriente de Fase C (Amperes)	Corriente calculada de neutro (Amperes)	Voltaje fase A a neutro (Voltios)	Voltaje fase B a neutro (Voltios)	Voltaje fase C a neutro (Voltios)	Factor de potencia verdadera fase A	Factor de potencia verdadera fase B	Factor de potencia verdadera fase C	THD/thd del voltaje de fase A (%)	THD/thd del voltaje de fase B (%)	THD/thd del voltaje de fase C (%)
19:45:07	59.99	162	174	157	41	120	120	118	0.988	0.984	0.987	2.8	3	3
19:45:08	60.03	163	178	148	42	120	120	118	-0.988	0.985	-0.983	2.8	3	3
19:45:09	60.04	149	173	146	44	121	120	118	-0.984	0.986	-0.983	2.8	2.9	3.1
19:45:10	60.01	144	173	156	47	121	120	118	-0.982	0.986	-0.985	2.8	2.9	3
19:45:11	60.03	143	174	153	48	121	120	118	-0.982	0.986	-0.984	2.8	3	3
19:45:12	60.04	139	162	143	43	121	120	118	-0.982	-0.983	-0.983	2.8	2.9	3.1
19:45:13	59.99	138	162	138	44	121	120	118	-0.982	-0.983	-0.982	2.8	2.9	3.1
19:45:14	60.01	136	163	137	44	121	120	118	-0.983	-0.983	-0.982	2.7	3	3
19:45:15	60.05	133	164	147	47	121	120	118	-0.983	-0.983	-0.984	2.7	3	3
19:45:16	60.02	132	164	138	47	121	120	118	-0.983	-0.983	-0.982	2.7	3	3
19:45:17	60.05	131	163	133	47	121	120	118	-0.983	-0.983	-0.981	2.6	3	3
19:45:18	60.03	131	163	131	46	121	120	118	-0.984	-0.983	-0.982	2.6	3	2.9
19:45:19	60.01	131	162	132	47	121	120	118	-0.983	-0.983	-0.981	2.7	3	2.9
19:45:20	60.07	132	161	127	45	121	120	118	-0.983	-0.984	-0.982	2.7	2.9	2.8
19:45:21	60.04	132	161	124	45	121	120	118	-0.983	-0.984	-0.983	2.6	2.9	2.8
19:45:22	60.02	130	160	122	44	121	120	118	-0.982	-0.985	-0.982	2.6	2.9	2.8
19:45:23	60.08	132	160	125	45	121	120	118	-0.983	-0.985	-0.982	2.6	2.9	2.9
19:45:24	60.09	130	159	122	44	121	120	118	-0.983	-0.985	-0.983	2.5	2.9	2.8
19:45:25	60.09	129	158	124	42	121	120	118	-0.983	-0.986	-0.983	2.5	2.9	2.8
19:45:26	60.1	129	159	124	44	121	120	118	-0.983	-0.985	-0.983	2.5	2.9	2.7
19:45:27	60.13	129	159	132	42	122	121	118	-0.983	-0.985	-0.985	2.5	2.9	2.8
19:45:28	60.05	128	158	132	43	122	121	118	-0.983	-0.984	-0.983	2.5	2.9	2.8
19:45:29	60.02	128	160	124	46	122	121	118	-0.983	-0.984	-0.983	2.5	3	2.8
19:45:30	60.04	128	150	117	44	122	121	118	-0.983	0.983	-0.975	2.5	3	2.9
19:45:31	60.03	128	150	114	44	122	121	118	-0.983	0.983	-0.976	2.5	3	2.9
19:45:32	60.04	126	150	112	43	122	121	118	-0.983	0.983	-0.977	2.5	3	2.9
19:45:33	59.96	126	150	112	43	121	120	118	-0.983	0.983	-0.977	2.3	2.9	2.7
19:45:34	60.04	126	149	113	43	121	121	118	-0.983	0.983	-0.977	2.4	2.9	2.8
19:45:35	60.01	120	148	112	44	121	121	118	-0.982	0.984	-0.978	2.4	2.9	2.9
19:45:36	60.03	124	147	112	42	121	121	118	-0.983	0.984	-0.978	2.4	2.8	2.8
19:45:37	59.99	124	147	111	42	121	120	118	-0.983	0.984	-0.978	2.4	2.9	2.8
19:45:38	60.01	124	154	124	44	121	120	118	-0.983	-0.986	-0.981	2.4	2.8	2.8
19:45:39	60.03	124	153	121	42	121	120	118	-0.983	-0.986	-0.983	2.4	2.7	2.7
19:45:40	60	124	153	123	43	121	120	118	-0.983	-0.986	-0.982	2.4	2.7	2.8
19:45:41	60.05	124	153	124	40	122	121	118	-0.983	-0.986	-0.982	2.4	2.7	2.8
19:45:42	60.03	125	152	124	42	122	121	118	-0.983	-0.986	-0.982	2.5	2.7	2.7
19:45:43	60.02	124	151	119	39	121	120	118	-0.983	-0.986	-0.983	2.4	2.7	2.8
19:45:44	60.03	126	151	119	40	122	121	118	-0.983	-0.986	-0.983	2.5	2.7	2.8
19:45:45	60	124	150	119	39	121	120	118	-0.983	-0.986	-0.983	2.4	2.7	2.8
19:45:46	59.96	124	149	120	38	121	120	118	-0.983	-0.986	-0.983	2.4	2.7	2.8
19:45:47	60	119	149	118	40	121	120	118	-0.982	-0.986	-0.983	2.4	2.7	2.8
19:45:48	59.99	120	149	118	40	121	120	118	-0.982	-0.986	-0.983	2.3	2.7	2.7
19:45:49	59.96	119	142	111	41	121	121	118	-0.981	0.984	-0.977	2.4	2.8	2.8
19:45:50	60.04	119	141	114	41	122	121	118	-0.982	0.985	-0.976	2.4	2.7	2.8
19:45:51	60.03	119	141	112	40	122	121	118	-0.982	0.984	-0.977	2.4	2.8	2.8
19:45:52	60.03	119	141	114	40	122	121	118	-0.982	0.984	-0.976	2.3	2.8	2.8
19:45:53	60.05	119	141	113	40	122	121	118	-0.982	0.984	-0.977	2.3	2.8	2.7
19:45:54	60.02	119	149	123	42	122	121	118	-0.982	-0.985	-0.982	2.4	2.8	2.7
19:45:55	60.03	119	150	126	40	122	121	118	-0.982	-0.986	-0.985	2.4	2.8	2.7
19:45:56	60.03	122	150	121	39	122	121	118	-0.984	-0.986	-0.983	2.4	2.8	2.8
19:45:57	60.03	122	148	127	39	122	121	118	-0.984	-0.986	-0.981	2.4	2.7	2.8
19:45:58	60.02	122	146	121	37	122	121	118	-0.984	-0.987	-0.982	2.4	2.7	2.9
0:52: 0	60.03	122	146	120	38	122	121	118	-0.984	-0.987	-0.983	2.4	2.6	2.8
0:53: 0	60.07	122	146	120	38	122	121	119	-0.984	-0.987	-0.983	2.4	2.6	2.7
0:54: 0	60	122	148	125	40	122	121	118	-0.983	-0.987	-0.981	2.4	2.7	2.8
0:55: 0	59.99	122	147	123	39	122	121	118	-0.983	-0.987	-0.982	2.4	2.6	2.8
0:56: 0	59.97	122	147	123	39	122	121	118	-0.983	-0.987	-0.982	2.4	2.7	2.8
0:57: 0	59.99	122	146	123	39	122	121	118	-0.984	-0.987	-0.982	2.4	2.6	2.8
0:58: 0	60.03	122	146	123	39	122	121	118	-0.983	-0.987	-0.982	2.3	2.6	2.7
0:59: 0	60.01	122	139	111	37	122	121	118	-0.983	0.986	-0.978	2.4	2.7	2.7

A continuación se explican las gráficas de las muestras tomadas.

GRÁFICA DE FRECUENCIA

Los valores mostrados en la gráfica de la frecuencia, figura 4.13, indican que la línea no presenta valores fuera del rango establecido por CFE, 64.8 a 55.2 Hz.

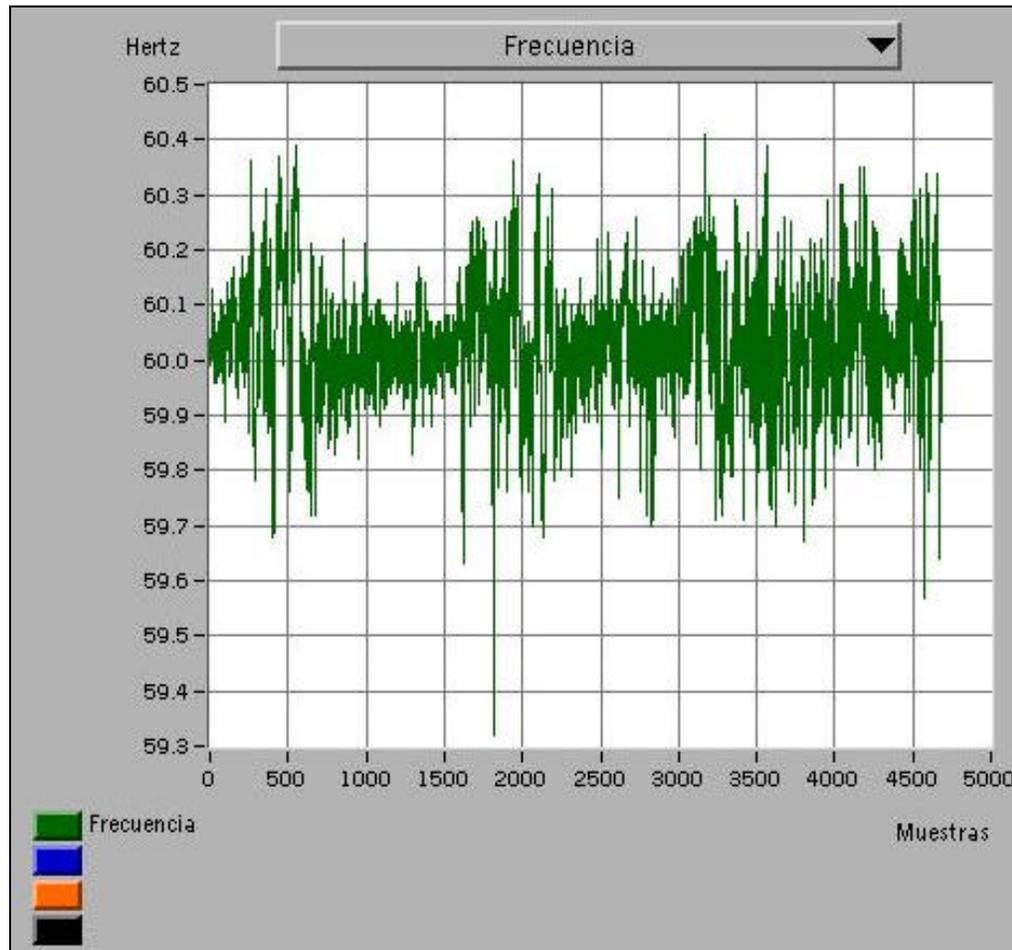


Figura 4.13. Gráfica de los valores de frecuencia del ejemplo 2.

GRÁFICA DE CORRIENTES

Los valores de la corriente monofásica graficados en la figura 4.14 indican que durante la toma de las muestras 1000 a 1500 se presentó una fuerte demanda de corriente lo que se puede interpretar a que en ese lapso se encendieron varios aparatos que produjeron dicha demanda. Esa demanda pudo ser ocurrida en la mañana del 5 de abril. Cabe señalar que el día 4 de abril de 2002 fue jueves, por lo tanto los picos donde la demanda de corriente se presentan ocurrieron uno por día, eso es lógico si observamos que los dos últimos picos de demanda corresponden al sábado y domingo, días donde la demanda es menor.

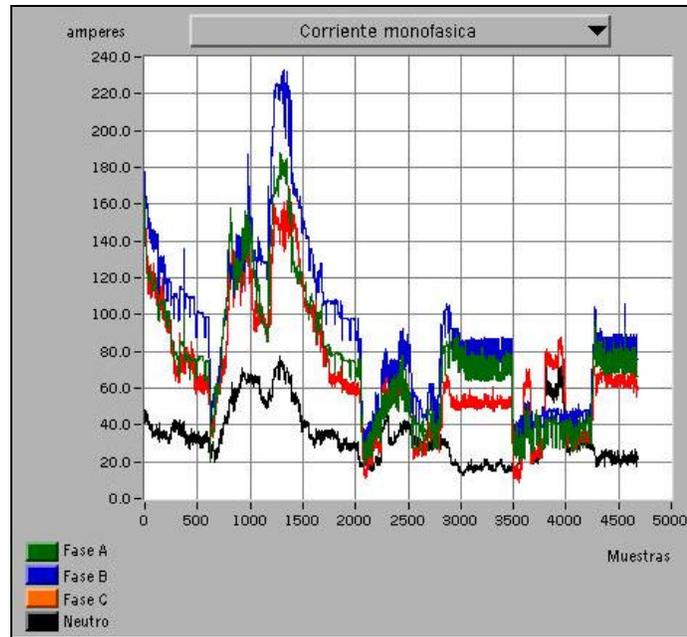


Figura 4.14. Gráfica de los valores de corriente monofásica del ejemplo 2.

GRÁFICA DE VOLTAJE MONOFÁSICO

La gráfica de voltaje monofásico, figura 4.15, presenta cuatro niveles de voltaje altos, los cuales pudieron ocurrir durante la mañana de cada día, debido a que esas horas de demanda total de la línea de CFE es mínima y los niveles más bajos de voltaje se deben a la demanda total es mayor provocado por el encendido del alumbrado.

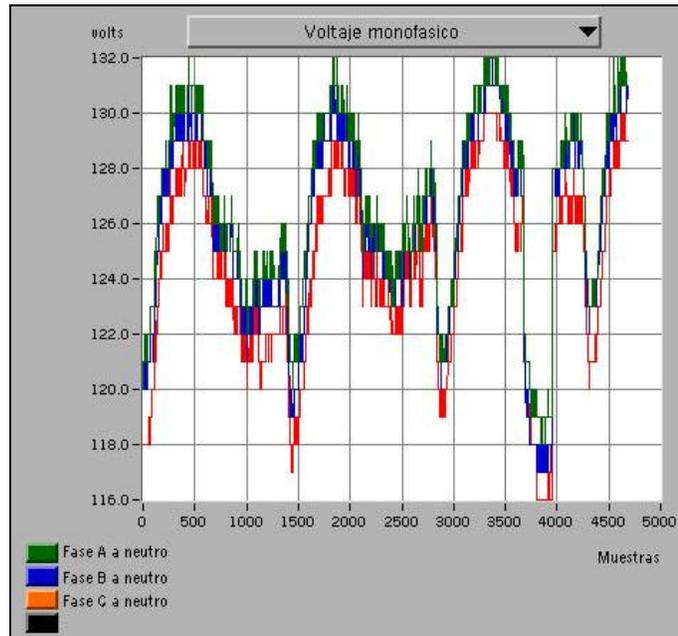


Figura 4.15. Gráfica de los valores de voltaje monofásico del ejemplo 2.

GRÁFICA DE FACTOR DE POTENCIA

Los valores mostrados en la gráfica de la figura 4.16 indican que la fase A y C presentan valores bajos de factor de potencia, lo cual se puede atribuir a un aumento en la carga inductiva o capacitiva. La fase B es la que muestra mejores valores de factor de potencia pues su rango es de 0.85 s 0.99 aproximadamente sin presentar cambios considerables.

La figura 4.17. muestra que los valores con signo de factor de potencia de la fase B son de tipo inductivo y capacitivo sin presentar una tendencia a alguno de ellos. Los valores mostrados en la gráfica para la fase A y C son predominantemente negativos lo que puede indicar que en esas fases se encuentran conectados equipos electrónicos que demandan un nivel alto de potencia reactiva tipo inductiva.

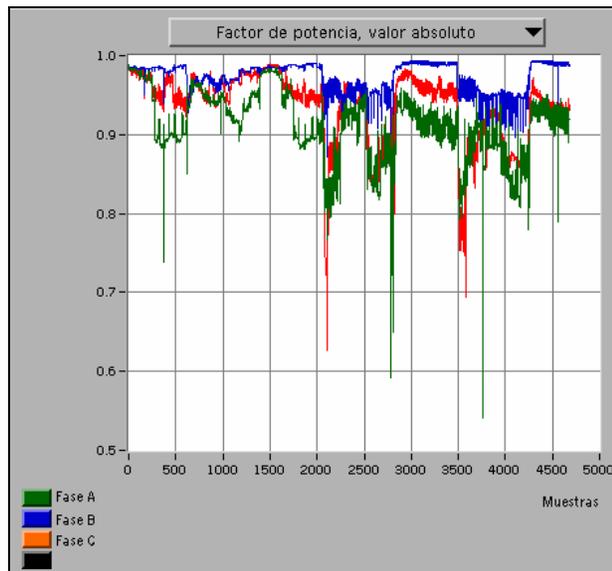


Figura 4.16. Gráfica de los valores absolutos del factor de potencia del ejemplo 2.

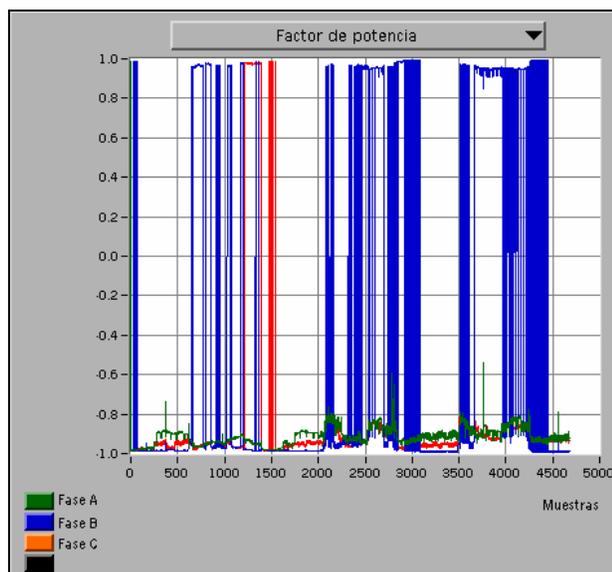


Figura 4.17. Gráfica de los valores con signo del factor de potencia del ejemplo 2.

El uso del lenguaje C para el desarrollo del programa que controla el funcionamiento del módulo de almacenamiento, redujo en un menor tiempo de implementación y en la generación de un código más entendible que si se hubiera usado lenguaje ensamblador.

El lenguaje de programación gráfico Labview tiene la enorme ventaja de poseer varias herramientas que lo hacen flexible para el desarrollo de aplicaciones de instrumentación, por lo cual las principales dificultades que se tuvieron durante el desarrollo del programa de descarga se debieron a la falta de conocimiento de dicho lenguaje. Se comprobó experimentalmente que el tiempo de ejecución de una aplicación en este ambiente es lento, por lo cual se recomienda su uso, en aplicaciones donde la velocidad de procesamiento de datos no sea un parámetro crítico.

El módulo de almacenamiento cumplió con el objetivo que se planteó y los productos finales de este trabajo, las gráficas generadas por el programa de procesamiento y el archivo con la información del almacenamiento, constituyen una herramienta para un Ingeniero eléctrico para plantear posibles mejoras e implementar medidas de ahorro de energía.

4.3. PERSPECTIVAS

Los posibles trabajos que se plantean a futuro, tomando como base el módulo de almacenamiento son:

- Implementar una rutina en el programa del módulo considerando el protocolo de comunicaciones Modbus para comunicar varios módulos de almacenamiento con su respectivo medidor PM650 y formar con ellos un red de información que pueda ser utilizada para analizar el comportamiento de una o más redes eléctricas al mismo tiempo.
- El programa de descarga puede ser modificado añadiendo gráficas que muestren los resultados de operaciones probabilísticas (media, promedio, varianza, etc.) para ayudar al usuario en su análisis.
- Los datos obtenidos y almacenados en formato de texto ASCII pueden analizarse desarrollando un sistema experto que permita un análisis automatizado de la red eléctrica estudiada.
- Publicar los resultados del trabajo en una revista de divulgación nacional o un congreso nacional de instrumentación.

BIBLIOGRAFÍA

REFERENCIAS BIBLIOGRÁFICAS

- [1] Mileaf, H. : Electricidad tres. Limusa Noriega Editores, 1997, México.
- [2] Hayt, W. y Kemmerly, J.: Análisis de circuitos en ingeniería. Editorial McGraw-Hill, 1993, México.
- [3] Mompin, J.: La Electrónica en la Industria. Ediciones Orbis, S.A, 1986, España.
- [4] Square, D: *Manual de instrucciones Power Meter Clase 3020*. Square D, 1998, EUA.
- [5] Hall, D: Microprocessors and interfacing programming and hardware. McGraw-Hill Editions, 1992, Singapur.
- [6] Analog Devices Inc.: *5 V Powered CMOS RS-232 Drivers/Receivers, Datasheet*. Analog Devices Inc, 2001, USA.
- [7] Texas Instruments Incorporated: *MC3486 Quadruple differential line receiver with 3-state output*. Texas Instruments Incorporated, 1995, USA.
- [8] Texas Instruments Incorporated: *MC3487 Quadruple differential line driver*. Texas Instruments Incorporate, 1995, USA.
- [9] Eurotherm Controls: *JBUS and MODBUS Digital communications handbook*. Eurotherm Controls, 1993, England.
- [10] National Instruments Corporation: *LabVIEW Help Version 5.1*. National Instruments Corporation, 1999.
- [11] Fairchild Semiconductor Corporation: *MM74C922 • MM74C923, 16-Key Encoder • 20-Key Encoder*. Fairchild Semiconductor Corporation, 1999, USA.
- [12] Purdy Electronics Corporation: *Intelligent Alphanumeric, Application Notes*. Purdy Electronics Corporation, 1999, USA.

URL's

- [URL 1] <http://www.conae.gob.mx>
“*Información básica sobre la electricidad de la Comisión Nacional de Ahorro de Energía*”. *Página principal: Desde el Hogar: ¿Qué es la electricidad?*.
- [URL 2] <http://www.cfe.gob.mx>
“*Página de la Comisión Federal de Electricidad, descripción del recibo de luz*”. *Página principal: ¿Qué es CFE?: Información: Preguntas más frecuentes: Conoce tu recibo*.
- [URL 3] <http://www.eelab.usyd.edu.au/labview/intro.html>
“*Introducción al uso de Labview*”.

ANEXOS

ANEXO A. Tarifas para el suministro y venta de energía eléctrica (2001 - 2002) *

TARIFAS ESPECÍFICAS

Domesticas aplicadas del 1° de enero al 7 de febrero del 2002	1, 1A, 1B, 1C, 1E
Domésticas a partir del 8 de febrero del 2002	1, 1A, 1B, 1C, 1E
Domesticas de alto Consumo	DAC
Servicios públicos	5, 5A, 6
Agrícolas	9, 9M
Temporal	7

TARIFAS GENERALES

En baja tensión	2, 3
En media tensión	O-M, H-M, H-MC
En alta tensión	HS, HS-L, HT, HT-L
Servicio de respaldo	HM-R, HM-RF, HM-RM, HS-R, HS-RF, HS-RM, HT-R, HT-RF, HT-RM
Servicio interrumpible	I-15, I-30

TIPOS DE MODELOS DE RECIBOS DE LUZ

Hay tres modelos de recibo.

- Para los servicios que destine la energía a cualquier uso en baja tensión, alumbrado público, bombeo de agua y riego agrícola. **Tarifas 2, 5, 5A, 6, 9 y 9M.**
- Para los servicios que destinen la energía a cualquier uso en baja y media tensión. **Tarifas 3 y OM.**
- Para los servicios horarios en media y alta tensión. **Tarifas HM, HS, HS-L, HT Y HT-L.**

Los tres tipos de modelos contienen la siguiente información:

- **Información de pagos y fechas.** Incluye el monto del adeudo anterior, pagos, cargos ó créditos, monto a pago, fecha límite de pago y fecha de corte.
- **Datos generales del servicio.** Ubicación del suministro: nombre o razón social, dirección, población o entidad federativa donde se localiza el suministro.

Domicilio fiscal: datos de la cédula de la identificación fiscal.

Tarifa: tipo de tarifa registrada por el consumidor.

* Información tomada de la dirección electrónica: <http://www.cfe.gob.mx>

Información del consumo

Período de consumo		Días	Tarifa	Hilos	Consumo kWh por día	Uso
29 JUN 01 a 29 AGO 01		61	01	1	8.73	Doméstico
Medidor	Lecturas		Multiplicador	Consumo		
Número	Actual	Anterior				
Z8384N	43577	43044	1	533		
COMPROMETIDOS CON LA HONESTIDAD						

Período de consumo: fecha inicial y final del período en que se registro el consumo de energía eléctrica.

Días: número de días que abarca el período.

Tarifa: identificación de la tarifa aplicada.

Hilos: número de cables con corriente eléctrica contratados.

Multiplicador: constante que depende de la tarifa aplicada al uso de energía eléctrica.

Consumo kWh por día: consumo de energía eléctrica que en promedio se utilizó por día.

Uso: descripción de la tarifa aplicada.

Número de medidor.

Lectura actual y lectura anterior.

Consumo kWh: diferencia entre la lectura actual y la anterior.

Cálculo del importe del consumo

CALCULO DEL IMPORTE DE SU FACTURACION					
COSTO		FACTURACION			
	Concepto	KWh	*	Precio	Total
Costo de Producción: \$715.90	Básico	150	*	0.451	67.65
	Intermedio	250	*	0.532	133.00
	Excedente	133	*	1.557	207.08
	Suma	533			407.73
		APORTACION GUBERNAMENTAL		\$308.17	

Costo: costo de generación, transmisión distribución y comercialización.

Facturación: resultado de aplicar las cuotas que fija la Secretaría de Hacienda y Crédito Público al consumo de energía eléctrica.

Aportación gubernamental: aportación del Gobierno Federal para cubrir la diferencia entre el costo de producción y la facturación.

Detalle de los cargos y créditos que integran la facturación

		CONCEPTOS	IMPORTE
<p style="text-align: center;">AVISOS IMPORTANTES</p> <p>>> Nos transformamos para servirle mejor. >> Gracias por su pago efectuado el 17 JUL 01 por \$446.00 >> Servicio a Clientes Teléfono 071</p> <p>Fecha y lugar de expedición: 31 AGO 01, SAN LUIS POTOSI, SLP Son: (CUATROCIENTOS SESENTA Y NUEVE PESOS 84/100 M.N.)</p> <p>Secretaría de Contraloría, Quejas y Denuncias al teléfono: 01 800 712 69 82</p>		Energía	407.73
		I.V.A.	61.15
		Fac. del Periodo	468.88
		Adeudo Anterior	446.96
		Su Pago	446.00 -
		Total	\$469.84

Conceptos: Conceptos que integran el importe total de la factura.

Importe: cargos o créditos que corresponden a cada concepto.

Fecha y lugar de expedición.

RECIBO DE LUZ PARA BAJA Y MEDIA TENSIÓN

Contiene la siguiente información.

- Información de pagos y fechas.
- Datos generales del servicio.
- Información del consumo.
- Cálculo del importe del consumo.
- Detalle de los cargos y créditos que integran la facturación.
- Historia del consumo.
- Talón de caja.



Información del consumo

Función	No. Medidor	Lectura actual	Lectura anterior	Diferencia	Totales
kWh	1234YZ	26307	25842	465	37,200
kW	1234YZ	1.0998	0.0	1.0998	88
kVArh	1234YZ	13110	12898	212	16,960

Función: Tipo de parámetro que registra el medidor.

Número de medidor: Letras y números que identifican al medidor.

Lectura actual y lectura anterior: Lecturas del medidor correspondientes al período en que se registro el consumo.

Diferencia: Resultado de restar la lectura anterior a la lectura actual.

Totales: Resultado de multiplicar la diferencia por el multiplicador para obtener el total a facturar de cada concepto.

Cálculo del importe del consumo

Mes	Días del mes	Consumo prom. diario	Energía kWh	Precios \$/kWh	Importes \$
0108	29	1,282.7586	37,200	0.4260	15,847,71
Mes	Factor de proporción	Demanda máxima kW	Precios \$/kW	Importes \$	Factor de potencia %
0108	0.9355	88	56.59	4,658.71	
					90.99

Mes: meses que comprende el período.

Días del mes: días de cada mes comprendidos en el período.

Consumo promedio diario: Energía eléctrica que en promedio se utilizó cada día del período de consumo. Resultado de dividir el consumo entre el número de días que abarca el período de consumo.

Energía kWh: Consumo de energía que se utilizó en el período indicado. Se obtiene de multiplicar el consumo promedio diario por los días de cada mes comprendidos en el período.

Precios \$/kWh: Cuotas de la tarifa que se aplica al consumo de energía.

Importes \$: Resultado de multiplicar los precios al consumo de energía de cada mes.

Factor de proporción: Es el resultado de dividir los días del mes que comprende el período entre los días calendarios de cada mes.

Demanda máxima kW: Indica la demanda máxima medida en cualquier intervalo de 15 minutos, en el cual el consumo de energía eléctrica sea mayor que cualquier otro intervalo de 15 minutos en el período de facturación del consumo.

Precios \$/kW: Cuotas de la tarifa que se aplican a la demanda máxima medida.

Importe \$: Resultado de multiplicar el factor de proporción por la demanda máxima medida por el precio de la demanda.

Factor de potencia: Factor del aprovechamiento del consumo de energía en trabajo útil o fuerza mecánica. El valor de Factor de Potencia mínimo aceptable es de 90%. Cuando el valor de Factor de Potencia tiene un valor superior al 90% se aplica una bonificación; mientras que un porcentaje menor al 90% significa energía que se desperdicia y se aplica un cargo.

Detalle de los cargos y créditos que integran la facturación

Conceptos	Importes \$
Cargo por Energía	15,847.20
Cargo por Demanda	4,658.71
Cargo 2% Baja Tensión	410.11
Bonificación Factor de Potencia	62.74-
Subtotal	20,853.28
IVA	3,127.99
Facturación del Periodo	23,981.27
Adeudo Anterior	26,722.46
Su Pago	26,722.00-
Total	\$23,981.73

Conceptos: Son los distintos conceptos que integran el importe total de la factura.

Importes \$: Cargos o créditos (-) que corresponden a cada concepto.

RECIBO DE LUZ PARA HORARIOS EN MEDIA Y ALTA TENSION

Contiene la siguiente información.

- Información de pagos y fechas.
- Datos generales del servicio.
- Información del consumo.
- Cálculo del importe del consumo.
- Detalle de los cargos y créditos que integran la facturación.
- Historia del consumo.
- Talón de caja.

COMISION FEDERAL DE ELECTRICIDAD

Asociación Patronal: \$3,677,000.00 | Página: \$3,677,000.00 | Cargos/creditos: \$0.428.48.35 | Muestra pagar: \$4,703,428.83

Fecha de emisión de pago: 12 ABRIL 01 | Fecha de corte de la: 03 ABRIL 01

Ubicación del suministro: INDA S.P.A. México, S.A. MARQUE INDUSTRIAL No. 5 O. CALLE 196 N. L. MONTERREY, N.L. MISA 55121 MEX

Numero de Servicio: 006 331 220 312

Periodo: 31 MAR 01 a 30 ABR 01 | **Carga contratada (kW):** 10,200 | **Tarifa:** -HSL

Historia del consumo:

Consumo	Consumo anterior	Diferencia	Importe	
WWh en carga	32,548	12,382	10,249	2,450,040
WWh en carga	22,515	15,976	14,038	3,361,501
WWh en carga	18,800	17,309	2,938	696,980
WWh en carga	41	0	41	0.842
WWh en carga	42	0	42	0.852
WWh en carga	43	0	43	0.862
WWh en carga	18,700	7,188	11,512	276,380
Factor de potencia %				91.27

Detalle de los cargos y créditos:

Concepto	Importe	Importe anterior	%	Tarifa
Energía en carga kWh	2,450,040	0.3228		
Energía en demanda kWh	3,361,501	0.3422		
Energía en demanda kWh	696,980	0.0813		
Cargos en demanda kWh	0.852	0.34		
Bonificación factor de potencia	0.842	0.34		
Subtotal	2,577,921.93			
IVA	814,232.48			
Facturación del periodo	3,392,154.41			
Su pago	3,392,154.41			
Total	33,725,442.27			

TOTAL A PAGAR: \$23,981.73

Información del consumo

Función y Período	Lectura actual	Lectura anterior	Diferencia	Totales
kWh base	22,548	12,302	10,246	2,459,040
kWh intermedia	29,913	15,905	14,008	3,361,920
kWh punta	14,880	11,976	2,904	696,960
kWh semipunta				
kW base	41	0	41	9,840
kW intermedia	42	0	42	10,080
kW punta	40	0	40	9,600
kW semipunta				
kVArh	18,790	7,198	11,592	278,280
Factor de potencia %				91.97

Función y período: Tipo de parámetro que registra el medidor en períodos de base, intermedia, punta y semipunta.

Lectura actual y lectura anterior: Lecturas del medidor correspondientes al período en que se registro el consumo.

Diferencia: Resultado de restar la lectura anterior a la lectura actual.

Totales: Cantidad total del consumo, demanda y reactivos utilizados en los períodos de base, intermedia, punta y semipunta. Resultado de multiplicar la diferencia por el multiplicador, para obtener el total a facturar de cada concepto.

Factor de potencia: Factor del aprovechamiento del consumo de energía en trabajo útil o fuerza mecánica. El valor de Factor de Potencia mínimo aceptable es de 90%. Cuando el valor de Factor de Potencia tiene un valor superior al 90% se aplica una bonificación; mientras que un porcentaje menor al 90% significa energía que se desperdicia y se aplica un cargo.

Cálculo del importe del consumo

Conceptos	Totales	Precios unitarios \$
Energía en base kWh	2,459,040	0.3052
Energía en intermedia kWh	3,361,920	0.3433
Energía en punta kWh	696,960	0.9653
Energía en semipunta kWh		
Demanda facturable kW	9,696	63.38

Conceptos: Conceptos que integran el importe total de la factura.

Totales: Cantidad de cada concepto en las unidades definidas.

Precios unitarios: Cuotas de la tarifas aplicables a cada concepto.

Detalle de los cargos y créditos que integran la facturación

Conceptos	Importes \$
Cargo por energía	2,577,421.62
Cargo por demanda	614,532.48
Cargo 2% baja tensión	63,839.08
Bonificación factor de potencia	16,278.96-
Subtotal	3,239,514.22
IVA	485,927.13
Facturación del período	3,725,441.35
Saldo anterior	3,617,985.85
Su pago	3,617,985.00-
Total	\$3,725,442.20

Conceptos: Son los distintos conceptos que integran el importe total de la factura.

Importes \$: Cargos o créditos (-) que corresponden a cada concepto.

ANEXO B. Microcontrolador 8031

El sistema de almacenamiento está basado en un microcontrolador de la familia MCS-51 de INTEL, el 8031. En la tabla B.1 se describen las características principales de los integrantes de la familia MCS-51.

La familia de microcontroladores MCS-51 posee características óptimas para diversas aplicaciones; fue diseñada, fabricada y comercializada, por la compañía INTEL, aunque en la actualidad otras compañías tienen licencia para producirlas y venderlas, tal como Siemens, Advanced Micro Devices, Fujitsu y Phillips entre otras .

Tabla B.1 Integrantes de la familia MCS-51 y sus principales características.

Dispositivo	Memoria interna		Temporizadores o contadores de eventos	Interrupciones
	Programa	Datos		
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16-Bit	6
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8032AH	ninguna	256 x 8 RAM	3 x 16-Bit	6
8031AH	ninguna	128 x 8 RAM	2 x 16-Bit	5
8031	ninguna	128 x 8 RAM	2 x 16-Bit	5
8751AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8751H-8	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5

Algunas características que posee el microcontrolador 8031 son las siguientes:

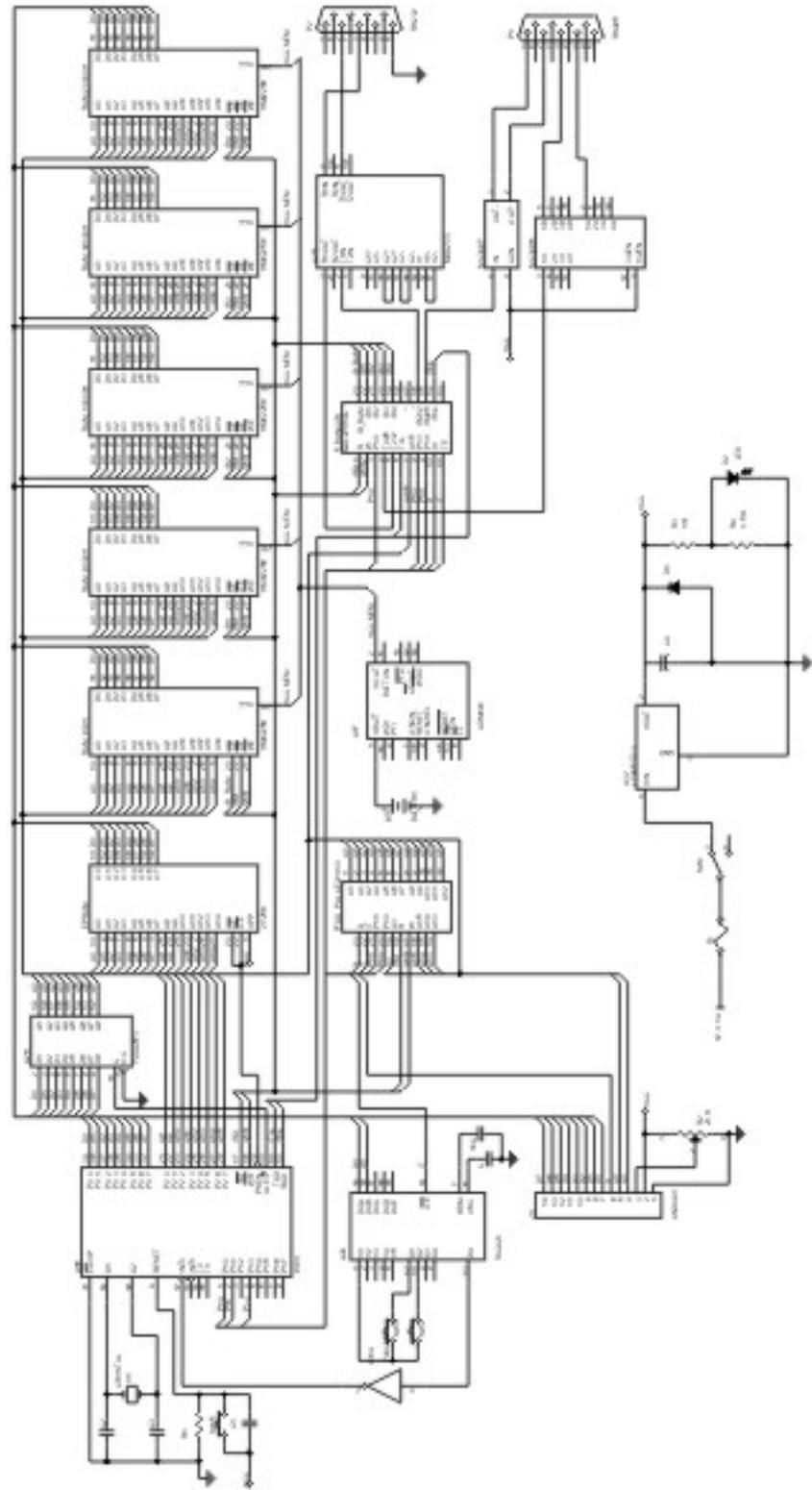
- 128 bytes de memoria RAM, incluidos cuatro bancos de ocho registros generales y una pila para las llamadas a subrutinas e interrupciones.
- Cuatro puertos de 8 bits de entrada/salida.
- Dos temporizadores de 16 bits cuya salida activa una interrupción.
- Interfaz serial.
- 64 K de direccionamiento de memoria de código.
- 64 K de direccionamiento de memoria de datos.
- Operación de bit a bit.
- 210 localidades direccionables por bit.
- Temporizadores e interrupciones de I/O con dos niveles de prioridad.
- Unidad Aritmética Lógica (ALU) con capacidad de realizar funciones aritméticas como suma, resta, multiplicación y división y las funciones lógicas AND, OR, XOR y complementos.
- Conjunto de 111 instrucciones de 51 funciones básicas.

- Frecuencia máxima de operación de 12 MHz, con un ciclo de máquina de 12 períodos de reloj.

Este microcontrolador no está contenido en un sistema embebido, es decir que es necesario agregar más dispositivos para su funcionamiento básico, estos dispositivos y el 8031 forman entonces lo que comúnmente se le denomina *sistema mínimo*, el cual tiene los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

ANEXO C. Esquemático del módulo de almacenamiento



Lista de componentes que integran al módulo de almacenamiento y su costo.

Componente	Identificador	Valor	Cantidad	*Costo unitario	Subtotal
ADM691	U17	-	1	\$22.30	\$22.30
CAPACITOR	C	0.001 mF	1	\$2.70	\$2.70
CAPACITOR	10C,C1	0.01 mF	2	\$2.70	\$5.40
CAPACITOR	C3,C2	33 pF	2	\$0.90	\$1.80
CAPACITOR	C4	0.1 mF	1	\$0.90	\$0.90
DIODE	D1	-	1	\$0.90	\$0.90
LED	D2	-	1	\$1.80	\$1.80
EPROM	EPROM	32 Kbytes	1	\$39.00	\$39.00
FUSE	F1	1 A	1	\$2.00	\$2.00
LCD AND491	J1	-	1	\$70.00	\$70.00
CONECTOR DB9 FEMALE	P1, P2	-	2	\$2.00	\$4.00
RAM	RAM_8000H RAM_8031 RAM_10000H RAM_18000H RAM_20000H	HM62256	5	\$44.00	\$220.00
RESISTOR	R1, R3	1 K Ohm	2	\$0.20	\$0.40
RESISTOR	R4	1.5 K Ohm		\$0.20	\$0.00
POTENCIOMETRO	R2	20 K Ohm		\$8.00	\$0.00
SW PUSHBUTTON	SW1,SW2	-	2	\$3.00	\$6.00
SW PUSHBUTTON-SPDT	SW3	-	1	\$3.50	\$3.50
Microcontrolador 8031	U16	-	1	\$30.00	\$30.00
MAX233	U25	-	1	\$63.00	\$63.00
74AC573	U26	-	1	\$3.00	\$3.00
MC3487	U27	-	1	\$9.00	\$9.00
MC3486	U28	-	1	\$6.00	\$6.00
74C923	U29	-	1	\$53.00	\$53.00
74AC05	U31	-	1	\$3.00	\$3.00
L7805/TO3	U32	-	1	\$17.00	\$17.00
CRISTAL	Y1	12 MHz	1	\$29.00	\$29.00
GAL22V10C	5_BANCOS 7FXX	-	2	\$31.00	\$62.00

SUBTOTAL: \$633.40

Costo de diseño y construcción: \$17,181.00

Costo de programa de descarga: \$10,308.60

TOTAL: \$28,123.00

* En pesos mexicanos.

Precios vigentes al 26 de mayo de 2002

Fuente: AG Electrónica S.A de C. V.

ANEXO D. Manual de usuario del módulo de almacenamiento

GENERALIDADES

Definición

El módulo de almacenamiento de datos del medidor PM650, es un sistema que almacena datos de mediciones realizadas por el medidor PM650 con la configuración del usuario.

Características

El módulo de almacenamiento cuenta con las siguientes características:

- 128 KBytes de memoria RAM utilizados para almacenar los datos del medidor PM650.
- Puerto de comunicaciones RS485 para la conexión hacia el medidor, mediante el cual se adquieren los datos de las mediciones.
- Puerto de comunicaciones RS232 para la conexión hacia una computadora personal que permite la descarga de datos almacenados.
- Interfaz con el usuario constituida por un visualizador y un teclado para la elección de los datos a almacenar, así como la configuración del tiempo de almacenamiento.

Conexiones

El medidor de potencia debe estar conectado a la red eléctrica y el módulo de almacenamiento se une al medidor utilizando comunicaciones RS485. En la figura D.1 se muestra el diagrama de conexiones del módulo de almacenamiento.

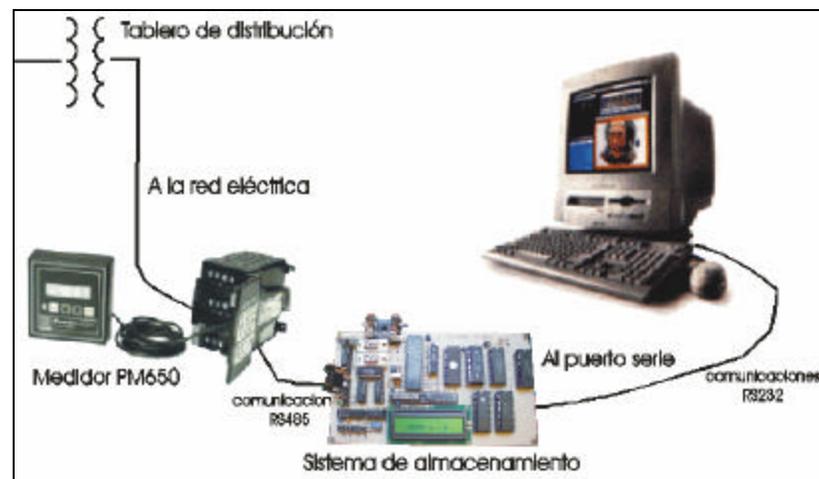


Figura D.1. Conexiones de módulo de almacenamiento.

ALMACENAMIENTO DE LOS DATOS DEL PM650

Instalación del medidor PM650

Para la instalación del medidor se considera lo siguiente:

- Utilizar vestimenta y calzado de seguridad.
- Realizar las conexiones en compañía de otra persona.
- Asegurar que la caja de interruptores termomagnéticos tenga los mismos en posición de apagado.
- Instalar los cables dispuestos para unir los voltajes y corrientes al módulo de almacenamiento.
- Conectar los cables de los voltajes a la caja de interruptores.
- Conectar las donas electromagnéticas considerando la dirección de la corriente, así mismo observar el orden de dichas corrientes.

Estado de funcionamiento del módulo de almacenamiento

- Después de la instalación de las entradas de voltajes y corrientes, encender el módulo de almacenamiento, oprimiendo el botón de ENCENDIDO ubicado en el panel frontal de módulo de almacenamiento.
- Asegurar de que módulo de almacenamiento funciona correctamente, esto es si aparece la velocidad de transmisión utilizado del medidor interno. De otro modo oprima el botón de RESET ubicado en el panel frontal de módulo de almacenamiento.

Inicialización

Al iniciar, módulo de almacenamiento envía el siguiente mensaje:

M O D U L O D E
A L M A C E N A M I E N T O

Después de esto, el módulo de almacenamiento indica la velocidad de transmisión que utiliza para comunicarse con el medidor PM650.

Para 1200 bps.

B a u d r a t e 1 2 0 0
u t i l i z a n d o R S 4 8 5

Para 2400 bps.

B a u d r a t e 2 4 0 0
u t i l i z a n d o R S 4 8 5

Para 4800 bps.

B a u d r a t e 4 8 0 0
u t i l i z a n d o R S 4 8 5

Si el módulo no estuviera conectado al medidor o la velocidad de transmisión utilizada por el medidor no es 1200, 2400 o 4800 bps, el módulo envía el siguiente mensaje de advertencia:

```
N O H A Y C O N E X I O N
C O N E L M E D I D O R
```

```
V e r i f i c a
c o n e x i o n e s ,
```

```
v e l o c i d a d t x d e l
P M 6 5 0 y r e i n i c i a
```

Si existieran datos en la memoria del módulo, es decir, hubo un corte en la energía eléctrica y la batería respaldó la información, el módulo envía el siguiente mensaje para informar al usuario que hay datos respaldados en la memoria del módulo. Después de este mensaje el módulo presenta directamente el menú de operación.

```
E x i s t e n d a t o s e n
l a m e m o r i a . . . .
```

Si no existieran datos en la memoria, el módulo de almacenamiento puede verificar la memoria de almacenamiento si el usuario lo desea, para esto envía el siguiente mensaje:

```
V e r i f i c a r m e m . ?
s i ( S ) / n o ( M )
```

Al cual el usuario puede contestar si o no; si elige verificar memoria, el módulo de almacenamiento escribe y lee un datos específico a las localidades de los cuatro bancos de memoria, si puede leer el valor escrito, al término de la última localidad de cada uno de los cuatro bancos envía el mensaje:

```
B a n c o 1
0 K
```

```
B a n c o 2
0 K
```

```
B a n c o 3
0 K
```

```
B a n c o 4
0 K
```

Configuración

Después de la inicialización, el módulo de almacenamiento espera a que el usuario elija uno de los cuatro modos: configuración, inicio de almacenamiento, envío de datos o inicialización del módulo.

El primer modo es el de configuración y en la pantalla del módulo de almacenamiento aparece:

```
M O D O :
C o n f i g u r a c i o n
```

En este modo se eligen los bloques que se desean almacenar así como el tiempo de muestreo, para esto se elige el modo configuración oprimiendo la tecla SELECCIÓN, en la pantalla del módulo aparece el siguiente mensaje.

```
F r e c u e n c i a
G U A R D A R ? S I
```

En la primer línea se lee el bloque que se está configurando, se hace la pregunta “GUARDAR?” a la cual por defecto el módulo responde “SI”, pero el usuario puede cambiar la respuesta oprimiendo la tecla MODO, a lo cual la pantalla muestra la respuesta, para confirmar la respuesta solo se oprime SELECCIÓN y módulo de almacenamiento muestra el siguiente bloque hasta completar los 10 bloques de mediciones. Los valores para la configuración son “SI” y “NO”.

Mensaje de configuración del bloque 1:

```
F r e c u e n c i a
G U A R D A R ? S I
```

Mensaje de configuración del bloque 2:

```
I m o n o f a s i c o
G U A R D A R ? S I
```

Mensaje de configuración del bloque 3:

```
V b i f a s i c o
G U A R D A R ? S I
```

Mensaje de configuración del bloque 4:

```
V m o n o f a s i c o
G U A R D A R ? S I
```

Mensaje de configuración del bloque 5:

```
F P m o n o f a s i c o
G U A R D A R ? S I
```

Mensaje de configuración del bloque 6:

```
P o t . r e a l
G U A R D A R ? S I
```

Mensaje de configuración del bloque 7:

```
P o t . r e a c t i v a
G U A R D A R ? S I
```

Mensaje de configuración del bloque 8:

```
T H D / t h d   d e   l a   I
G U A R D A R ?   S I
```

Mensaje de configuración del bloque 9:

```
T H D / t h d   d e l   V
G U A R D A R ?   S I
```

Mensaje de configuración del bloque 10:

```
E n e r g i a   t o t a l
G U A R D A R ?   S I
```

Después de elegir los bloques que serán almacenados, se muestra el mensaje que indica la elección del tiempo de muestreo. Utilizando el botón MODO se muestran los siguientes 16 valores para el tiempo de muestreo:

```
T i e m p o   m u e s t r e o
a u t o m a t i c o ?
```

```
T i e m p o   m u e s t r e o :
1   m i n u t o ?
```

```
T i e m p o   m u e s t r e o :
2   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
3   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
4   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o
5   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
6   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
7   m i n u t o s ?
```

```
T i e M p o   m u e s t r e o :
8   m I n u t o s ?
```

```
T i e M p o   m u e s t r e o :
9   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
1 0   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
1 1   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
1 2   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
1 3   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
1 4   m i n u t o s ?
```

```
T i e m p o   m u e s t r e o :
1 5   m i n u t o s ?
```

Cuando se visualiza el valor deseado para el tiempo de muestreo, se presiona el botón SELECCIÓN para confirmar la elección; después de esto y dependiendo del valor del tiempo de muestreo se muestra el tiempo aproximado de almacenamiento. Por ejemplo, con todos los bloques seleccionados y el tiempo de muestreo a 2 minutos, el tiempo total de muestreo será de 2 días, 9 horas y 28 minutos, para lo cual la pantalla mostrará lo siguiente:

```
T . A L M A C E N A M I E N T O
      2 d      9 : 2 8
```

Si la elección toma en cuenta todos los bloques de mediciones y el tiempo de muestreo automático, en la pantalla aparecera el siguiente mensaje:

```
T . A L M A C E N A M I E N T O
      a u t o m a t i c o
```

Después de este mensaje se debe oprimir cualquier botón para salir del modo de configuración.

Inicia muestreo

Para iniciar el almacenamiento de valores, se debe oprimir el botón SELECCIÓN cuando aparezca el siguiente mensaje.

```
M O D O :
I n i c i a   m u e s t r e o
```

Después de esto, en la pantalla del módulo se irá mostrando el avance del almacenamiento.

Para el caso de tiempo automático y todos los bloques seleccionados, se verá el siguiente mensaje:

```
R e s t a n :
          1 7 2 4   v e c e s
```

Al inicio, aparecerá el número total de veces que se tomaran las muestras dependiendo de los bloques de mediciones seleccionados, después cada vez que se almacena una muestra el número se decrementa en uno.

Para el caso de tiempo no automático, en este caso 2 minutos, y todos los bloques seleccionados se verá el siguiente mensaje; el tiempo se decrementará conforme la adquisición avanza.

```
T i e m p o   r e s t a n t e :
  2   d   9 : 2 8
```

Cuando el almacenamiento haya concluido, la pantalla del módulo mostrará el número de muestras adquiridas.

```
1 7 2 4   m u e s t r a s
          a d q u i r i d a s
```

Después de que el mensaje de finalización de almacenamiento ha aparecido basta con oprimir cualquier botón para salir del modo inicio de almacenamiento.

Envía datos PC

En este modo el módulo de almacenamiento envía los datos almacenados en su memoria hacia la PC utilizando las comunicaciones RS232 a una velocidad de transmisión de 4800 bps. La elección de este modo se efectúa oprimiendo el botón SELECCIÓN después de ver el mensaje:

```
M O D O :
E n v i a   d a t o s   P C
```

Al seleccionar este modo, el módulo de almacenamiento verifica que existan datos en su memoria, si esto no fuera así muestra el mensaje de alerta al usuario.

```
N o   e x i s t e n   d a t o s
          p a r a   e n v i a r
```

Si el módulo de almacenamiento verifica que existen datos en la memoria, envía un 0xFFH indicando al programa de descarga el inicio de la transmisión de los datos (ver sección 3.3.1.) y en la pantalla del módulo siguiente el mensaje.

```
D e t e c t a n d o   R x . .
```

Para iniciar la recepción de los datos, la PC debe estar conectada a través de su puerto serial al puerto serial del módulo de almacenamiento y ejecutar el programa de descarga de datos de módulo de almacenamiento. Cuando la comunicación se establece con el programa de descarga, el módulo de almacenamiento indica que se esta enviando datos hacia la PC.

```
E n v i a n d o   d a t o s . .
```

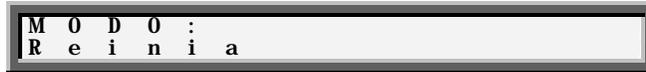
Al finalizar el envío de datos se visualiza el siguiente mensaje, que indica el número de bytes almacenados en la memoria de módulo de almacenamiento.

```
1 3 1 0 2 4
b y t e s   e n v i a d o s
```

Después de este mensaje, se debe oprimir cualquier botón para salir del modo envía datos a PC.

Reinicia

Este modo es útil, si se ha cambiado la configuración del medidor interno de módulo de almacenamiento o simplemente se quiere que el módulo tome los valores por defecto. Para elegir este modo, se oprime el botón SELECCIÓN después de que aparece el mensaje:



M O D O :
R e i n i c i a

Después de esto el módulo de almacenamiento es reiniciado y lo indica con el siguiente mensaje.



S i s t e m a
r e i n i c i a n d o . . .

DESCARGA DE DATOS

La descarga de los datos almacenados se lleva a cabo utilizando el archivo *módulo de almacenamiento.vi* contenido en el archivo *módulo almacenamiento.llb*

NOTA: Para poder ejecutar el archivo *módulo de almacenamiento.vi* es necesario tener instalado la aplicación LabView 5.1 de National Instruments bajo el sistema operativo Windows 95, 98 ó 2000.

Ejecución del programa de descarga

Para abrir el archivo del software de descarga se realiza lo siguiente:

1. Desde el programa explorador de Windows dar dos pulsaciones al botón izquierdo del ratón sobre el archivo *módulo almacenamiento.llb*
2. Al aparecer el cuadro de diálogo de archivo, seleccionar el archivo *módulo de almacenamiento.vi* y dar una pulsación en el botón OK.
3. O de otra forma:
4. Ejecutar el programa Labview.
5. Dar un clic en el boton **Open vi**.
6. Elegir la ruta donde el archivo *módulo almacenamiento.llb* se encuentra, señalar dicho archivo y dar una pulsación en el botón abrir.
7. Al aparecer el cuadro de diálogo de archivo, seleccionar el archivo *módulo de almacenamiento.vi* y dar una pulsación en el botón OK.

Cuando el panel de control aparezca hacer lo siguiente:

- Asegurarse que el cable serial esté conectado al puerto serial RS232 del módulo de almacenamiento y a un puerto serial de la PC.
- Elegir el número del puerto serial que se está utilizando para descargar los datos, para esto hacer uso del ratón para incrementar o decrementar el valor del indicador **Puerto serial**, ubicado en el extremo izquierdo superior del panel de control.
- Cuando las conexiones y la elección del puerto serial se hayan realizado, ejecutar el software dando una pulsación sobre el icono , o bien oprimir las teclas Ctrl+R.

Adquisición de los datos

Una vez que *módulo de almacenamiento.vi* ha obtenido y procesado los datos almacenados, pedirá al usuario la ruta y nombre del archivo en donde los datos serán almacenados. Los datos son procesados y almacenados en un archivo de texto ASCII con extensión .txt.

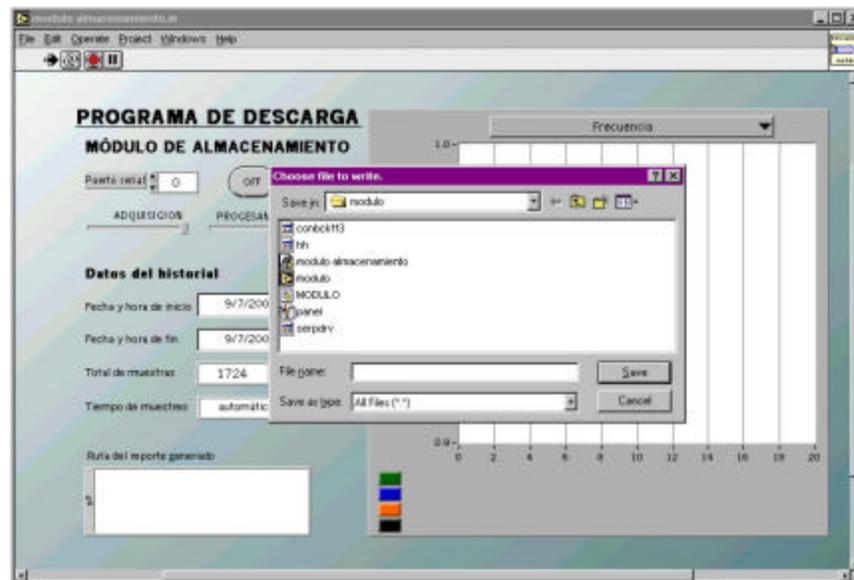


Figura D.2. Cuadro de diálogo para el almacenamiento de datos.

Presentación de los datos

Cuando el *módulo de almacenamiento.vi* ha almacenado los datos en el archivo, cuyo nombre y ruta el usuario eligió, grafica los datos según el indicador que se encuentre en la parte superior derecha del panel frontal de *módulo de almacenamiento.vi*.

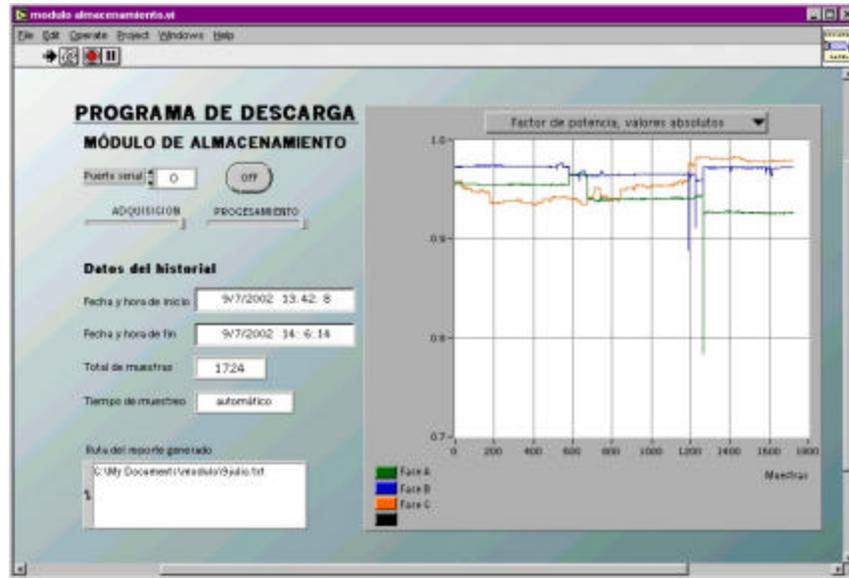


Figura D.3. Panel frontal del programa de descarga.

ERRORES COMUNES

Errores en el módulo de almacenamiento

E r r o r e n
l o c a l i d a d n

Mensaje que el módulo de almacenamiento envía cuando no puede leer un banco de memoria en la rutina de verificación de memoria, el módulo debe apagarse, examinar las memorias, cambiar memorias dañadas y reiniciar el módulo.

C O M U N I C A C I O N
S E R I A L C O N E R R O R

V E R I F I C A . .
C O N E X I O N E S Y V E L

D E T X D E P M 6 5 0 Y
R E I N I C I A S I S T .

Error que se presenta cuando el módulo de almacenamiento no puede establecer comunicación con el medidor; esto puede ocurrir cuando no hay conexión entre el módulo de almacenamiento y el medidor o el medidor esta configurado con una velocidad de transmisión diferente a 1200, 2400 o 4800 bps.

N O E X I S T E N D A T O S
P A R A E N V I A R

Mensaje para informarle al usuario que no existen datos en la memoria del módulo de almacenamiento. Debe ejecutarse primero el modo de adquisición de datos.

Errores en el software de descarga

Error 37 occurred at Serial Port INIT

Possible reasons:

LabVIEW: Device not found

Error en comunicación serial, verificar conexiones y ejecutar de nuevo el software.

ANEXO E. Listado de programas

ANEXO E.1. Programa en lenguaje opal para decodificar la memoria y seleccionar el estándar de comunicaciones

5_BANCOS.OPL

```

begin header
DECODIFICADOR QUE HABILITA 4 BANCOS
end header

begin definition
device gal22v10;
inputs A15=7,P10=3,P11=9,P13=8,T232=5,T485=4,Tm=6;
inputs W=1,R=2,DI=10,TE=11;
outputs (com) /B_RAM=23,/B1=22,/B2=21,/B3=20,/B4=19;
outputs (com) /Rm=14,/R485=15,/R232=16;
end definition

begin equations
WR      = /W*R+/R*W;

B_RAM  = /P11*/P10*/A15*WR;
B1     = /P11*/P10* A15*WR*/DI*TE;
B2     = /P11* P10*/A15*WR;
B3     = /P11* P10* A15*WR;
B4     = P11*/P10*/A15*WR;

Rm     = /P13*/T232 +
        P13*/T485 ;

R232   = /P13*/Tm;

R485   = P13*/Tm;

end equations

```

5_BANCOS.LOG

OPL2PLA - OPAL design entry compiler V008
 Copyright (c) National Semiconductor Corporation 1991
 Input Pins

```

=====
Pin name          pin number    arch    visible  default
-----          -
A15                7            com    visible
P10                 3            com    visible
P11                 9            com    visible
P13                 8            com    visible
T232                5            com    visible
T485                4            com    visible
Tm                  6            com    visible
W                   1            com    visible
R                   2            com    visible
DI                  10           com    visible
TE                  11           com    visible
  
```

Output Pins

```

=====
Pin name          pin number    arch    visible  default
-----          -
/B_RAM            23           com    visible  rst
/B1                22           com    visible  rst
/B2                21           com    visible  rst
/B3                20           com    visible  rst
/B4                19           com    visible  rst
/Rm                14           com    visible  rst
/R485              15           com    visible  rst
/R232              16           com    visible  rst
  
```

1 internal variables declared.WR

EQN2JED - Boolean Equations to JEDEC file assembler (Version V024)
 Copyright (c) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

W	1	24	VCC
R	2	23	/B_RAM
P10	3	22	/B1
T485	4	21	/B2
T232	5	20	/B3
Tm	6	19	/B4
A15	7	18	
P13	8	17	
P11	9	16	/R232
DI	10	15	/R485
TE	11	14	/Rm
GND	12	13	

ANEXO E.2. Programa en lenguaje opal para mapeo en memoria RAM del teclado y display

800X.OPL

```
begin header
SISTEMA DE ADQUISICION DE DATOS
8000 ENABLE DEL DISPLAY 491
8001 ENABLE DEL DISPLAY 491
8002 ENABLE TECLADO
end header

begin definition
device gal22v10;
inputs A0=1,A1=2,A2=3,A4=4,A5=5,A6=6,A7=7,A8=8,A9=9,
        A10=10,A11=11,A12=13,A13=14,A14=15,A15=16,W=17,R=18,A3=19;
inputs P10=20,P11=21;
outputs (com) D=23,/T=22;
end definition

begin equations

A  = A15*/A14*/A13*/A12*/A11*/A10*/A9*/A8*/A7*/A6*/A5*/A4*/A3*/A2;

D  = /P11*/P10*A*/A1*/W;

T  = /P11*/P10*A* A1*/A0*/R;

end equations
```

800X.LOG

OPL2PLA - OPAL design entry compiler V008
 Copyright (c) National Semiconductor Corporation 1991
 Input Pins

```

=====
Pin name          pin number    arch    visible  default
-----          -
A0                1            com     visible
A1                2            com     visible
A2                3            com     visible
A4                4            com     visible
A5                5            com     visible
A6                6            com     visible
A7                7            com     visible
A8                8            com     visible
A9                9            com     visible
A10              10           com     visible
A11              11           com     visible
A12              13           com     visible
A13              14           com     visible
A14              15           com     visible
A15              16           com     visible
W                17           com     visible
R                18           com     visible
A3              19           com     visible
P10             20           com     visible
P11             21           com     visible
  
```

Output Pins

```

=====
Pin name          pin number    arch    visible  default
-----          -
D                 23           com     visible  rst
/T               22           com     visible  rst
  
```

1 internal variables declared.A

EQN2JED - Boolean Equations to JEDEC file assembler (Version V024)

Copyright (c) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

A0	1	24	VCC
A1	2	23	D
A2	3	22	/T
A4	4	21	P11
A5	5	20	P10
A6	6	19	A3
A7	7	18	R
A8	8	17	W
A9	9	16	A15
A10	10	15	A14
A11	11	14	A13
GND	12	13	A12

ANEXO E.3. Programa del módulo de almacenamiento

MODULO_ALMACENAMIENTO.C

```

#include<absacc.h>
#include<math.h>
#include<reg51.h>

#define DC XBYTE[0x8000]
#define DD XBYTE[0x8001]
#define TE XBYTE[0x8002]
#define inicio_memoria 0x8003

void determina_baud(unsigned char *baud);
unsigned int configura_comunicaciones(unsigned char baud);
void delay(unsigned char d);
void envia_paquete(unsigned int direccion,unsigned int numero_registros);
void envia_c(unsigned char c);
unsigned int checksum(unsigned char *byte,unsigned char elementos);
void inicializa_LCD(void);
void dato_teclado(void);
void control(unsigned int d, unsigned int fst);
void interrupcion_ext0(void) interrupt 0 using 0;
void despliega(char *cadena);
void espera_n_segundos(char seg);
void limpia_LCD(void);
void linea(unsigned int ren, unsigned int col);
void pide_mod0(unsigned char *opcion);
void configura_sistema(void);
void configura_bloques(void);
void selecciona_bloque(char *bloque);
void despliega_numero(unsigned long numero);
void despliega_tiempo(unsigned int numero);
void inicia_almacenamiento(unsigned char baud);
unsigned char recibe_c(void);
void adquiere_bloque(unsigned int dir, unsigned char n_reg);
void almacena(unsigned long dir, char dato);
char lee_memoria(unsigned long dir);
void pide_almacena_bloques(void);
unsigned char pide_fecha(void);
void obtiene_fecha(void);
void envia_datos(void);
void checa_memoria(void);
void ram(long limite);

sbit PUERTO=0x90^3;
xdata unsigned char tecla,tiempo_sapre;
xdata unsigned char b1,b2;
xdata unsigned char b3,b4;
xdata unsigned char b5,b6;
xdata unsigned char b7;
xdata unsigned char b8;
xdata unsigned char b9,b10;
xdata unsigned long veces;
xdata unsigned char paquete[30]={0};
xdata unsigned char dd,mm;
xdata unsigned char yy,hh;
xdata unsigned char mi,ss;
xdata unsigned char dd_1,mm_1;
xdata unsigned char yy_1,dd_2;
xdata unsigned char mm_2,yy_2;
xdata unsigned char hh_1,mi_1;
xdata unsigned char ss_1,hh_2;
xdata unsigned char mi_2,ss_2;
xdata unsigned char pto_com;
xdata unsigned char backoff;

```

```

xdata unsigned long memoria_RAM= 0x28000-inicio_memoria;
xdata unsigned long direccion_datos;

void main(void)
{
  unsigned char baud,opcion;
  unsigned char baud_transmision;

  P1=0;

  inicializa_LCD();

  do{
    limpia_LCD();
    linea(1,0); despliega(" MODULO DE ALMACENAMIENTO");
    espera_n_segundos(10);

    determina_baud(&baud);

    IE=0;
    PS=0;
    PX0=0;

    limpia_LCD();
    if(baud!=26)
      {
        if(baud==12) despliega(" Baud rate 1200");
        if(baud==24) despliega(" Baud rate 2400");
        if(baud==48) despliega(" Baud rate 4800");
        linea(2,0);
        if(PUERTO==1)
          {
            despliega("utilizando RS485");
            else
              despliega("utilizando RS232");
          }
        espera_n_segundos(10);
      }
    pto_com=PUERTO;
    baud_transmision=baud;

    if(backoff!=1)
      {
        opcion=1;
        b1=1; b2=1; b3=1; b4=1; b5=1; b6=1; b7=1; b8=1; b9=1; b10=1;
        tiempo_sapre=0;
        veces=memoria_RAM/76;
        direccion_datos=inicio_memoria;
        dd_1=0; mm_1=0; yy_1=0; dd_2=0; mm_2=0; yy_2=0;
        hh_1=0; mi_1=0; ss_1=0; hh_2=0; mi_2=0; ss_2=0;
        checa_memoria();
      }
    else
      {
        limpia_LCD();
        despliega("Existen datos en memoria...");
        espera_n_segundos(10);
        limpia_LCD();
      }

    P1=0;

    do{
      pide_modos(&opcion);
      if(opcion==1)
        configura_sistema();
      if(opcion==2)
        inicia_almacenamiento(baud_transmision);
      if(opcion==3)
        envia_datos();
      dato_tecladb();
      espera_n_segundos(3);
    }while(opcion!=4);
  }
}

```

```

        limpia_LCD();
        despliega("Sistema      reiniciando..");
        espera_n_segundos(10);
        backoff=0;
        }while(opcion!=4);
    }
    /***** F U N C I O N E S *****/
    void ram(long limite)
    {
        long i;
        unsigned char c=0;

        if(limite==0x10000) i= 0x8003;
        if(limite==0x18000) i=0x10000;
        if(limite==0x20000) i=0x18000;
        if(limite==0x28000) i=0x20000;

        for(i<limite;i++)
        {
            almacena(i,0xAA);
            c=lee_memoria(i);
            if(c!=0xAA)
            {
                limpia_LCD();
                despliega("  Error en localidad");
                linea(2,15);
                despliega_numero(i);
                while(1);
            }
        }
    }
    /***** *****/
    void checa_memoria(void)
    {
        tecla=0;
        limpia_LCD();
        despliega("Checar memoria?");
        linea(2,0);
        despliega("si(S)/no(M)...");
        dato_teclado();
        if(tecla==1)
        {
            limpia_LCD(); despliega("Checando memoria");

            linea(2,0); despliega("Banco 1      "); ram(0x10000);
            linea(2,14); despliega("OK"); espera_n_segundos(7);

            linea(2,0); despliega("Banco 2      "); ram(0x18000);
            linea(2,14); despliega("OK"); espera_n_segundos(7);

            linea(2,0); despliega("Banco 3      "); ram(0x20000);
            linea(2,14); despliega("OK"); espera_n_segundos(7);

            linea(2,0); despliega("Banco 4      "); ram(0x28000);
            linea(2,14); despliega("OK"); espera_n_segundos(7);
        }
        else
            espera_n_segundos(3);
    }
    /***** *****/
    void envia_datos(void)
    {
        unsigned long dir_num_dir,aux;
        unsigned char c=0;
        unsigned char bytes_bloque=0;

        if(direccion_datos==inicio_memoria)
        {
            limpia_LCD();
            despliega("No existen datos  para enviar");
            dato_teclado();

```

```

}
else
{
    bytes_bloque=(b1*b2+b2*8+(b3+b4+b5+b6+b7+b8+b9)*6+b10*24);

    num_dir=(direccion_datos-inicio_memoria)/bytes_bloque;
    veces=num_dir;

    PUERTO=0; // comunicaciones RS232
    delay(3);

    SCON=0x52; TMOD=0x20; ES=1;
    PCON=0x80; TH1=0xF3; // BAUD RATE = 4800

    limpia_LCD();
    despliega("Detectando Rx..");
    RI=0;
    TR1=1; EA=1;
    do{
        envia_c(0xFF);
        delay(3);
        if(RI)
            c=SBUF;
        RI=0;
        espera_n_segundos(10);
    }while(c!=0xFF);

// envia fecha y tiempo de inicio
    limpia_LCD(); despliega("Enviando datos..");
    envia_c(dd_1); envia_c(mm_1); envia_c(yy_1);
    envia_c(hh_1); envia_c(mi_1); envia_c(ss_1);
// envia fecha y tiempo de fin
    envia_c(dd_2); envia_c(mm_2); envia_c(yy_2);
    envia_c(hh_2); envia_c(mi_2); envia_c(ss_2);
// envia valor de bloques seleccionados
    envia_c(b1); envia_c(b2);
    envia_c(b3); envia_c(b4);
    envia_c(b5); envia_c(b6);
    envia_c(b7); envia_c(b8);
    envia_c(b9); envia_c(b10);
// envia tiempo de muestreo y numero de veces almacenados
    envia_c(tiempo_sapre);
    envia_c(veces/0x1000000);
    envia_c(veces/0x10000);
    envia_c(veces/0x100);
    envia_c(veces);
// envia datos
    aux=inicio_memoria+(num_dir*bytes_bloque);
    for(dir=inicio_memoria;dir<aux;dir++)
        envia_c(lee_memoria(dir));

    IE=0; PS=0; PX0=0;
    limpia_LCD();
    linea(1,14); despliega_numero(dir);
    linea(2,1); despliega("bytes enviados");
    dato_teclado();
}
}
/*****/
void obtiene_fecha(void)
{
    unsigned int crc_1,crc_2,i;
    unsigned char fecha[22];

    do{
        envia_paquete(1842,3);
        fecha[0]=recibe_c0();
        if(fecha[0]==1)
            i=1;
        else
            i=0;
    }
}

```

```

    for(;i<9;i++)
        fecha[i]=recibe_c();
    fecha[9]=0xFF;
    crc_1=recibe_c();
    crc_2=recibe_c();
    crc_1=crc_1+(crc_2*0x100);
    crc_2=checksum(fecha,9);

}while(crc_1!=crc_2);

mm=fecha[3];
dd=fecha[4];
yy=fecha[5];
hh=fecha[6];
mi=fecha[7];
ss=fecha[8];
}
/*****/
char lee_memoria(unsigned long dir)
{
    bit comunicacion;
    char dato;

    comunicacion=PUERTO;

    if(dir<0x10000)
    {
        DPH=dir/0x100;
        DPL=dir;
        ACC=XBYTE[(DPH*0x100)+DPL];
    }
    else
        if(dir<0x18000)
        {
            DPH=dir/0x100;
            DPL=dir;
            P1=1;
            ACC=XBYTE[(DPH*0x100)+DPL];
        }
    else
        if(dir<0x20000)
        {
            DPH=dir/0x100;
            DPL=dir;
            P1=1;
            ACC=XBYTE[(DPH*0x100)+DPL];
        }
    else
        if(dir<0x28000)
        {
            DPH=dir/0x100;
            DPL=dir;
            P1=2;
            ACC=XBYTE[(DPH*0x100)+DPL];
        }
    P1=0;
    PUERTO=comunicacion;
    dato=ACC;
    return(dato);
}
/*****/
void almacena(unsigned long dir, char dato)
{
    bit comunicacion;
    unsigned char bajo, alto;

    bajo=dir;
    alto=dir/0x100;

    comunicacion=PUERTO;

```

```

if(dir<0x10000)
{
ACC=dato;
B=ACC;
DPL=B;
DPH=dir/0x100;
B=DPL;
DPL=dir;
ACC=B;
XBYTE[(DPH*0x100)+DPL]=ACC;
}
else
    if(dir<0x18000)
    {
        ACC=dato;
        B=ACC;
        DPL=B;
        DPH=dir/0x100;
        B=DPL;
        DPL=dir;
        ACC=B;
    }
    P1=1;
    XBYTE[(DPH*0x100)+DPL]=ACC;
}
else
    if(dir<0x20000)
    {
        ACC=dato;
        B=ACC;
        DPL=B;
        DPH=dir/0x100;
        B=DPL;
        DPL=dir;
        ACC=B;
        P1=1;
        XBYTE[(DPH*0x100)+DPL]=ACC;
    }
    else
        if(dir<0x28000)
        {
            ACC=dato;
            B=ACC;
            DPL=B;
            DPH=dir/0x100;
            B=DPL;
            DPL=dir;
            ACC=B;
            P1=2;
            XBYTE[(DPH*0x100)+DPL]=ACC;
        }
    P1=0;
    PUERTO=comunicacion;
}
/*****/
void adquiere_bloque(unsigned int dir, unsigned char n_reg)
{
char i;
unsigned int crc,crc_enviado;

do{
envia_paquete(dir,n_reg);
do{
paquete[0]=recibe_c();
}while(paquete[0]!=1);

for(i=1;i<=(2+(2*n_reg));i++) // 2+(2*n_reg)
paquete[i]=recibe_c();
paquete[i]='\0';
crc_enviado=recibe_c();
crc=recibe_c();
}

```

```

crc_enviado=crc_enviado+(crc*0x100);
crc=checksum(paquete,3+(2*n_reg));

if(crc==crc_enviado)
{
for(i=3;i<(3+(2*n_reg));direccion_datos++,i++) // 2*n_reg
almacena(direccion_datos,paquete[i]);
}
else
delay(3);
}while(crc!=crc_enviado);
delay(1);
}
/*****/
void pide_almacena_bloques(void)
{
if(b1==1) adquiere_bloque(1001,1);
if(b2==1) adquiere_bloque(1003,4);
if(b3==1) adquiere_bloque(1014,3);
if(b4==1) adquiere_bloque(1018,3);
if(b5==1) adquiere_bloque(1031,3);
if(b6==1) adquiere_bloque(1039,3);
if(b7==1) adquiere_bloque(1043,3);
if(b8==1) adquiere_bloque(1051,3);
if(b9==1) adquiere_bloque(1055,3);
if(b10==1) adquiere_bloque(1617,12);
}
/*****/
unsigned char pide_fecha(void)
{
unsigned int crc_1,crc_2,i;
unsigned char fecha[12];
do{
envia_paquete(1844,1);
fecha[0]=recibe_c();
if(fecha[0]!=1)
i=0;
else
i=1;
for(i<5;i++)
fecha[i]=recibe_c();
fecha[i]=0xFF;
crc_1=recibe_c();
crc_2=recibe_c();
crc_1=crc_1+(crc_2*0x100);
crc_2=checksum(fecha,5);
}while(crc_1!=crc_2);

return(fecha[3]);
}
/*****/
void inicia_almacenamiento(unsigned char baud)
{
unsigned char minutos,min;
unsigned long num_almacenados;
unsigned long tiempo_restante,time;

PUERTO=pto_com;
SCON=0x52; TMOD=0x20; ES=1;

if(baud==12) { PCON=0; TH1=0xE6;}
if(baud==24) { PCON=0; TH1=0xF3;}
if(baud==48) { PCON=0x80; TH1=0xF3;}

num_almacenados=0;
tiempo_restante=(memoria_RAM/(b1*2+b2*8+(b3+b4+b5+b6+b7+b8+b9)*6+b10*24))*tiempo_sapre;
direccion_datos=inicio_memoria;
TR1=1; EA=1;

limpia_LCD();
obtiene_fecha();

```

```

dd_1=dd; mm_1=mm; yy_1=yy;
hh_1=hh; mi_1=mi; ss_1=ss;

if(tiempo_sapre==0)
{
    linea(1,0); despliega("Restan.");
    linea(2,10); despliega("veces");
    linea(2,0); despliega(" ");
    despliega_numero(veces-num_almacenados);
    do{

        pide_almacena_bloques();
        if(num_almacenados==0)
            backoff=1;
        linea(2,0);
        despliega(" ");
            num_almacenados++;
        despliega_numero(veces-num_alm acenados);
    }while(num_almacenados<veces);
}
else
{
    linea(1,0); despliega("Tiempo restante.");
    minutos=pide_fecha();
    do{
        time=tiempo_restante;
        linea(2,0);despliega(" ");
        linea(2,2); despliega_numero(time/1440);
        time=time-((time/1440)*1440);
        linea(2,3); despliega("d");
        linea(2,6); despliega_numero(time/60);
        time=time-((time/60)*60);
        linea(2,7); despliega(":");
        linea(2,9); despliega_numero(time);
        min=minutos+tiempo_sapre;
        if(min>=60)
            min=min-60;
        pide_almacena_bloques();
        do{
            minutos=pide_fecha();
            delay(2);
        }while(minutos!=min);
        num_almacenados++;
        tiempo_restante=tiempo_restante-tiempo_sapre;
    }while(num_almacenados<veces);
}
}
obtiene_fecha();
dd_2=dd; mm_2=mm; yy_2=yy;
hh_2=hh; mi_2=mi; ss_2=ss;

IE=0; PS=0; PX0=0;

limpia_LCD();
linea(1,6); despliega_numero(veces);
linea(1,8); despliega("muestras");
linea(2,6); despliega("adquiridas");
}
/*****/
void retrocede_LCD(void)
{
    control(1,0x10);
    control(1,0x10);
}
/*****/
void selecciona_bloque(char *bloque)
{
    linea(2,0);
    despliega("GUARDAR?");

    do{
        linea(2,14);

```

```

        if(*bloque==1)
            despliega("SI");
        if(*bloque==0)
            despliega("NO");
        espera_n_segundos(1);
        dato_teclado();
        if(tecla==0)
            *bloque=!(*bloque);
        }while(tecla!=1);
    }
    /***/
void configura_bloques(void)
{
    limpia_LCD();

    linea(1,0); despliega("Frecuencia  ");
    selecciona_bloque(&b1);
    linea(1,0); despliega("I en cada fase ");
    selecciona_bloque(&b2);
    linea(1,0); despliega("V bifasico  ");
    selecciona_bloque(&b3);
    linea(1,0); despliega("V monofasico ");
    selecciona_bloque(&b4);
    linea(1,0); despliega("FP monofasico ");
    selecciona_bloque(&b5);
    linea(1,0); despliega("Pot. real  ");
    selecciona_bloque(&b6);
    linea(1,0); despliega("Pot. reactiva ");
    selecciona_bloque(&b7);
    linea(1,0); despliega("THD/thd de la I ");
    selecciona_bloque(&b8);
    linea(1,0); despliega("THD/thd del V ");
    selecciona_bloque(&b9);
    linea(1,0); despliega("Energia total ");
    selecciona_bloque(&b10);
}
/***/
void configura_sistema(void)
{
    unsigned long tiempo;

    configura_bloques();

    espera_n_segundos(1);

    limpia_LCD();
    despliega("Tiempo muestreo:");
    do{
        linea(2,3);
        despliega_tiempo(tiempo_sapre);
        dato_teclado();
        if(tecla==0)
            tiempo_sapre++;
        if(tiempo_sapre>15)
            tiempo_sapre=0;
        espera_n_segundos(1);
    }while(tecla!=1);

    tiempo=tiempo_sapre;
    veces=(memoria_RAM/(b1*2+b2*8+(b3+b4+b5+b6+b7+b8+b9)*6+b10*24));
    if(tiempo!=0)
        tiempo=veces*tiempo;
    limpia_LCD();
    despliega("T.ALMACENAMIENTO");
    if(tiempo!=0)
    {
        linea(2,2);
        despliega_numero(tiempo/1440);
        tiempo=tiempo-((tiempo/1440)*1440);
        linea(2,3);
        despliega("d");
    }
}

```

```

        linea(2,6);
        despliega_numero(tiempo/60);
        tiempo=tiempo-((tiempo/60)*60);
        linea(2,7);
        despliega(".");
        linea(2,9);
        despliega_numero(tiempo);
    }
    else
    {
        linea(2,3);
        despliega("automatico ");
    }
}
/*****/
void despliega_tiempo(unsigned int numero)
{
    if(numero==0) despliega("automatico?");
    if(numero==1) despliega(" 1 minuto? ");
    if(numero==2) despliega(" 2 minutos?");
    if(numero==3) despliega(" 3 minutos?");
    if(numero==4) despliega(" 4 minutos?");
    if(numero==5) despliega(" 5 minutos?");
    if(numero==6) despliega(" 6 minutos?");
    if(numero==7) despliega(" 7 minutos?");
    if(numero==8) despliega(" 8 minutos?");
    if(numero==9) despliega(" 9 minutos?");
    if(numero==10) despliega("10 minutos?");
    if(numero==11) despliega("11 minutos?");
    if(numero==12) despliega("12 minutos?");
    if(numero==13) despliega("13 minutos?");
    if(numero==14) despliega("14 minutos?");
    if(numero==15) despliega("15 minutos?");
}
/*****/
void despliega_numero(unsigned long numero)
{
    unsigned long modulo;

    do{
        modulo=numero/10;
        delay(1);
        DD=(numero-(modulo*10))+0x30;
        numero=modulo;
        retrocede_LCD();
    }while(modulo!=0);
}
/*****/
void despliega_modo(unsigned char opcion)
{
    linea(2,0);
    if(opcion==1)
        despliega("Configuracion ");
    if(opcion==2)
        despliega("Inicia muestreo");
    if(opcion==3)
        despliega("Envia datos PC ");
    if(opcion==4)
        despliega("Reinicia  ");
}
/*****/
void pide_modo(unsigned char *opcion)
{
    limpia_LCD();
    despliega("MODO:");
    do{
        despliega_modo(*opcion);
        dato_teclado();
        if(tecla==0)
            (*opcion)++;
        if((*opcion)>4)

```

```

        (*opcion)=1;
        espera_n_segundos(1);
    }while(tecla!=1);
}
/*****/
void linea(unsigned int ren, unsigned int col)
{
    control(1, (((ren-1)*0x40)+ col+0x80) );
}
/*****/
void limpia_LCD(void)
{
    control(1,1);
}
/*****/
void espera_n_segundos(char seg)
{
    int i;
    for(i=0;i<(seg*2);i++)
        delay(3);
}
/*****/
void dato_teclado(void)
{
    tecla=0;
    PX0=1;
    SCON=0x02;
    IE=0x81;
    while(tecla==0);
    IE=0;
    tecla=TE&0x03;
}
/*****/
void inicializa_LCD(void)
{
    control(2,0x38);
    control(1,0x38);
    control(3,0x38);
    control(1,0x38);
    // control(1,0x0E); // CON CURSOR
    control(1,0x0C); // SIN CURSOR
    control(1,0x01);
    control(1,0x06);
    control(1,0x01);
    control(1,0x01);
}
/*****/
void determina_baud(unsigned char *baud)
{
    unsigned char respuesta;

    SCON=0x52;
    TMOD=0x20;
    ES=1;

    PUERTO=1; // COMUNICACIONES RS485
    *baud=12;
    respuesta=configura_comunicaciones(*baud);
    if(!respuesta)
    {
        *baud=24;
        respuesta=configura_comunicaciones(*baud);
        if(!respuesta)
        {
            *baud=48;
            respuesta=configura_comunicaciones(*baud);
        }
    }

    if(respuesta==0)

```

```

{
  PUERTO=0;
  *baud=12;
  respuesta=configura_comunicaciones(*baud);
  if(!respuesta)
  {
    *baud=24;
    respuesta=configura_comunicaciones(*baud);
    if(!respuesta)
    {
      *baud=48;
      respuesta=configura_comunicaciones(*baud);
    }
  }
}

IE=0;
PS=0;
PX0=0;
if(respuesta==0)
{
  *baud=26;
  limpia_LCD();
  despliega("NO HAY CONEXION");
  linea(2,0); despliega("CON EL MEDIDOR");
  espera_n_segundos(12);
  limpia_LCD();
  despliega("Verifica");
  linea(2,0); despliega("conexiones,");
  espera_n_segundos(4);
  limpia_LCD();
  despliega("velocidad tx del");
  linea(2,0); despliega("PM650 y reinicia.");
  espera_n_segundos(7);
}
}
/*****/
unsigned int configura_comunicaciones(unsigned char baud)
{
  unsigned char c=0;
  if(baud==12)
    { PCON=0; TH1=0xE6;}
  if(baud==24)
    { PCON=0; TH1=0xF3;}
  if(baud==48)
    { PCON=0x80; TH1=0xF3;}
  TR1=1;
  EA=1;
  envia_paquete(1001,1);
  delay(3);
  delay(3);
  if(RI)
    c=SBUF;

  RI=0;
  if(c==1)
    return(1);
  else
    return(0);
}
/*****/
void delay(unsigned char d)
{
  unsigned int i;
  if(d==1)
    for (i=0; i<2500; i++); /* retardo de 5ms*/
  if(d==2)
    for (i=0; i<8500; i++); /*retardo de 15ms*/
  if(d==3)
    for (i=0; i<50400; i++); /*retardo de 100ms*/
}

```

```

/*****/
void envia_paquete(unsigned int direccion,unsigned int numero_registros)
{
  unsigned char byte[7];
  unsigned char epi;
  unsigned int crc;

  direccion--;
  byte[0]=1;
  byte[1]=3;
  byte[2]=((direccion/0x100)+(direccion*0x100))&0xFF;
  byte[3]=(direccion)&0xFF;
  byte[4]=((numero_registros/0x100)+(numero_registros*0x100))&0xFF;
  byte[5]=(numero_registros)&0xFF;
  // byte[6]=0xFF;

  crc=checksum(byte,6);

  byte[6]=crc&0xFF;
  byte[7]=((crc/0x100)+(crc*0x100))&0xFF;

  for(epi=0;epi<=7;epi++)
    envia_c(byte[epi]);

  delay(1);
}
/*****/
void envia_c(unsigned char c)
{
  while(TI==0);
  TI=0;
  SBUF=c;
}
/*****/
unsigned int checksum(unsigned char *byte,unsigned char elementos)
{
  unsigned int ccrc=0xFFFF;
  unsigned char i,j,bytes;
  //for(i=0;*(byte+i)!=0xFF;)
  for(i=0;i<elementos;)
  {
    bytes=*(byte+i);
    ccrc=ccrc^bytes; // ccrc= xor(ccrc,*(byte+i));
    j=0;
    while(j<=7)
    {
      ccrc=ccrc/2; // ccrc=shr(ccrc);
      if(CY==1)
        ccrc=ccrc^0xA001; //ccrc=xor(ccrc,0xA001);
      CY=0;
      j++;
    }
    i++;
  }
  return(ccrc);
}
/*****/
void control(unsigned int d, unsigned int fst)
{
  DC=fst;
  delay(d);
}
/*****/
void interrupcion_ext0(void) interrupt 0 using 0
{
  IE=0;
  tecla=1;
}
/*****/
void despliega(char *cadena)
{

```

```
char i;
i=0;
while(*(cadena+i)!=0x00)
{
    delay(1); //(1)
    if(*(cadena+i) == 0xF1 || *(cadena+i)==0xD1)
        DD=0xEE;
    else
        DD=*(cadena+i);
    if (i==15)
    {
        delay(1);
        DC=0xC0; // funcion selecciona segunda linea
    }
    i++;
}
}
/*****/
unsigned char recibe_c(void)
{
    while(RI==0);
    RI=0;
    return(SBUF);
}
/*****/
```