# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

## FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

"REGISTRADOR DE DATOS (DATALOGGER) MEDIANTE UN DSPIC"

PROYECTO DE LA MATERIA DE GRADUACIÓN:

MICROCONTROLADORES AVANZADOS

PREVIA A LA OBTENCION DEL TÌTULO DE:

**INGENIERO EN ELÉCTRICIDAD ESPECIALIZACIÓN ELECTRONICA Y AUTOMATIZACIÓN INDUSTRIAL**

PRESENTADO POR:


**JULIO ARTEAGA PORTILLA**

**CARLOS NIETO DÍAZ**


GUAYAQUIL – ECUADOR

Año 2009

# A G R A D E C I M I E N T O

Agradezco a Dios por haberme brindado la oportunidad de culminar mis estudios universitarios. A mis padres y hermanos por apoyarme incondicionalmente durante mi vida. A la ESPOL por la formación académica recibida. A mis profesores por el conocimiento impartido durante mis años de estudio. A mis compañeros.

**Julio Arteaga Portilla.**

Agradezco a Dios por haberme dado fortaleza, constancia, y dedicación en el transcurso de mi vida estudiantil, a mis padres que sin su apoyo incondicional y sus consejos no hubiera logrado este objetivo y a todas aquellas personas que de una u otra manera aportaron a que este logro tan importante en mi vida sea posible.

**Carlos Nieto Díaz.**

# D E D I C A T O R I A

A mi madre por su gran esfuerzo y apoyo.

A mi padre por sus buenos consejos

Impartidos.

A mis hermanos por ser mi fuente de alegría.

**Julio Arteaga Portilla.**

A mi padre que gracias a su perseverancia y esfuerzos logro apoyarme incondicionalmente en el transcurso de mi vida estudiantil, con sus sabios consejos procuro inculcarme siempre el conocimiento.

**Carlos NietoDíaz.**

# TRIBUNAL DE GRADUACION

Ing. Carlos Valdivieso A.

DIRECTOR DE TESIS

Ing. Hugo Villavicencio

DELEGADO

# DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este proyecto de graduación nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

______________________ ______________________

Julio Arteaga Portilla. Carlos Nieto Díaz.

# RESUMEN

DATALOGGER, el proyecto consiste esencialmente en almacenar las variaciones de los parámetros de una señal deseada durante un intervalo de tiempo. La aplicación del proyecto mencionado con anterioridad se lo realizo utilizando un DsPic de la familia 30F4011, el proyecto registra la variación de la señal o de las señales adquiridas en un tiempo determinado, en nuestro caso las variables a ser analizadas son los voltajes de fase de una alimentación trifásica configurada en estrella, con sus respectivas corrientes de línea.

Para su correcto almacenamiento se procedió a almacenar la onda luego de un cruce por cero ascendentes para procurar registrar una señal cuasi semejante, la primera señal adquirida servirá como muestra para la siguiente y así sucesivamente, con esto obtenemos un muestreo continuo de la variación de la señal adquirida en un periodo de tiempo, logrando almacenar la variación de la señal deseada procurando no sobrescribir la señal previa, en el intervalo de tiempo determinado.

# INDICE GENERAL

# INDICE DE FIGURAS

# INDICE DE TABLAS



# LISTA DE SÍMBOLOS

| SIMBOLO | SIGNIFICADO |
|---|---|
| ºC | Grados Centigrados o Celsius |
| uA | Micro Amperios |
| Ma | Mili Amperios |
| A | Amperios |
| V | Voltios |
| mV | Mili Voltios |
| MHz | Mega Hertz |
| KHz | Kilo Hertz |
| Seg | Segundos |

| | |
|---|---|
| Pf | Pico Faradios |
| Hex | Hexadecimal |
| Vi | Voltaje de entrada |
| Vo | Voltaje de salida |
| VO1 | Voltaje de salida opan 1 |
| VO2 | Voltaje de salida opan 2 |
| Gnd | Tierra |
| VDC | Voltaje directo continuo |
| MΩ | Mega Ohmio |
| J1 | Jumper 1 |
| P1 | Potenciómetro 1 |
| P2 | Potenciómetro 2 |
| P3 | Potenciómetro 3 |
| IC2 | Circuito integrado 2 |
| IC3 | Circuito integrado 3 |
| Vref | Voltaje de referencia |
| Q1 | Transistor 1 |
| VCC | Alimentación |
| OUT | Salida |

# GLOSARIO

- Archivo hex.: Representación ASCII del código de máquina. Un archivo HEX está compuesto de registros que le especifican al microcontrolador datos o instrucciones que serán ubicados en un dispositivo de memoria programable.

- Circuito integrado: Dícese así, al conjunto de dispositivos que generan salidas en base al voltaje que se le aplica en sus entradas.

- Código fuente: Archivo de texto que es procesado por un lenguaje ensamblador o un compilador para producir un archivo de objeto intermedio, o código de máquina que pueda ejecutarse en un microcontrolador.

- Hardware: Partes o componentes físicos que integran una herramienta; inclusive ella misma como una unidad.

- Microcontrolador: Dispositivo electrónico que contiene todas las características de una computadora.

- Metodología: Es el proceso, técnicas o enfoques empleados en la solución de un problema o en la creación de algo: un procedimiento particular o un conjunto de procedimientos.

- Quemador: Es un circuito que permite cargar un programa al microcontrolador.

- Software: Es un programa de computadora que permite al programador interactuar con la computadora.

# 1 ANTECEDENTES

## 1.1 OBJETIVOS DEL PROYECTO

### 1.1.1 OBJETIVO GENERAL

Desarrollar un equipo electrónico capaz de realizar mediciones de voltaje y corriente, almacenar los datos medidos para un análisis posterior en donde se requiera el intercambio de información entre una aplicación desarrollada en Visual Basic y un dispositivo externo al computador.

### 1.1.2 OBJETIVOS ESPECÍFICOS

Realizar las mediciones de los voltajes de fase y corrientes de fase de un motor trifásico con alimentación de línea a línea de 220 voltios.

Almacenar los datos obtenidos tanto de voltajes como de corriente en el circuito implementado (Datalogger).

Desarrollar una interfaz gráfica amigable de interacción con el usuario que permita la visualización total de los datos grabados para su posterior análisis.

Garantizar totalmente la adquisición y almacenamiento de los datos durante todo el tiempo de muestreo.

## 1.2 IDENTIFICACIÓN DE LA PROBLEMÁTICA

Hoy en día ninguna industria puede darse el lujo de no contar con equipos de medición para monitorear su perfecto desempeño durante las horas de trabajo. El motor es una de las máquinas eléctricas más utilizadas en las industrias para diferentes etapas de un proceso e general, esto nos lleva a la necesidad de contar con un equipo que nos permita registrar el desarrollo del motor y verificar en lo posible en todo momento alguna anomalía con respecto a su funcionamiento en condiciones normales. El datalogger creado permite monitorear a un motor tanto monofásico como trifásico sin embargo su capacidad en memoria no es la suficiente para guardar tiempos considerables de muestras, para lo cual se lo ha diseñado de tal manera que se recojan los datos en un instante de tiempo e inmediatamente guardar los datos para ser analizados posteriormente en una PC.

### 1.2.1 TECNOLOGÍA UTILIZADA

La gran mayoría de los dataloggers que se encuentran disponibles en el mercado solo se encargan de registrar valores picos de la señal y en base a esos datos obtener su análisis, sin embargo su circuitería interna es reducida lo cual faculta la posibilidad de hacer cada vez más reducido el equipo.

Si bien nuestro equipo consta de circuitería eléctrica y electrónica básica, tiene la posibilidad de registrar la señal alterna del motor. Si bien es cierto la señal visualizada no coincide en forma exacta con la original la aproximación que se logró nos da como resultado una reproducción de la onda original en 70%, registrando en cada período los valores picos máximos y mínimos.

### 1.2.2 LIMITACIONES OPERACIONALES

Nuestro equipo como todo elemento de medición tiene sus limitaciones predeterminadas. Para las señales de voltaje el máximo voltaje que podrá ser leído es de 120Vrms. Para corriente no existe este problema pues la corriente máxima que puede ser leída es de 100A que es una cantidad de corriente considerable para motores industriales, sin embargo la limitación de la medición de corriente radica en el uso de un dispositivo de medida llamado FLUKE 80i – 110s AC/DC el cual genera según la selección una salida de 10mVAC por amperio ó 100mVAC por amperio leído.

Otra limitación de operación está sujeta la capacidad interna del DSPIC para almacenamiento de datos y por último el manejo del equipo debe ser el adecuado para ello se debe tener a mano el manual de funcionamiento del equipo.

## 1.3 JUSTIFICACIÓN PARA LOS CAMBIOS

Planteamos el desarrollo de un banco externo de memoria el cual nos permita almacenar mayor cantidad de datos y así poder alcanzar un mejor análisis de las ondas leídas.

Se plantea también la utilización de un programa más sofisticado para la simulación de un ESCADA con varios motores monitoreados por el equipo y en el cual se podría visualizar todo el proceso.

# 2 INTRODUCCIÓN

La medición de variables físicas tales como Voltaje, Corriente, Presión, Temperatura, Humedad, etc; es una tarea fundamental en la industria, en laboratorios, centros de investigación y en algunas actividades agrícolas, entre otros.

En algunos casos, dichas mediciones deben realizarse de forma secuencial para determinar el comportamiento de la variable correspondiente en un período de tiempo específico.

## 2.1 DESCRIPCIÓN DEL DATALOGGER

Es de vital importancia conocer la forma en que varían la corriente y el voltaje de un motor trifásico durante su desempeño, conocer sus valores máximos y mínimos así como su velocidad de variación. En este y en muchos otros casos, es necesario entonces contar con un equipo que pueda realizar dichas medidas durante un lapso de tiempo y a intervalos regulares seleccionados por el usuario.

Pensando en lo anterior, y con la intención de utilizar las modernas herramientas de diseño, como son los DSPIC, se inició la construcción de un equipo para realizar la adquisición y almacenamiento de datos de forma autónoma, es decir, que no tuviera como requisito la presencia de una computadora personal para realizar dicha labor.

El datalogger es un sistema que nos permite adquirir datos de una manera fácil y sencilla para luego ser procesados de manera gráfica paras su posterior análisis.

La idea principal  es que el instrumento pueda llevarse hasta el sitio donde se desean realizar las medidas y, una vez tomados y almacenados los datos, se pueda llevar dicho equipo hasta un sitio donde exista una computadora, la cual recibe a través del puerto serial los datos que se han tomado previamente. Este equipo recibe el nombre de “Registrador de Datos” ó “Datalogger”.

### 2.1.1 DIAGRAMA DE BLOQUES DEL DATALOGGER

En la figura 2.1.1 se muestra el diagrama en bloques del Datalogger. Cada uno de ellos está formado por un grupo específico de componentes y cumple una función determinada.

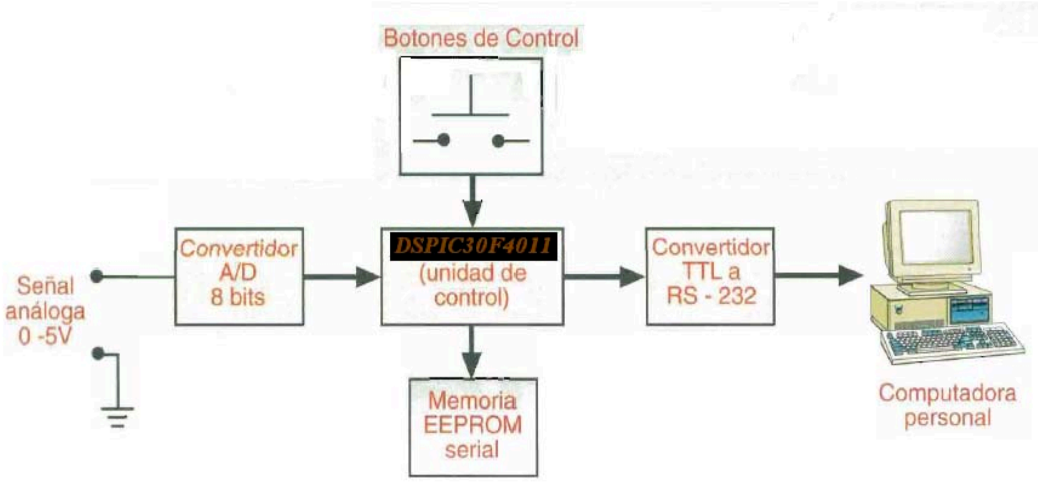



Fig. 2.1.1

La Unidad de Control está comandada por un DSPIC30F4011 y su respectivo oscilador interno FRC (8MHz). Se ha elegido este DSPIC porque posee una memoria EEPROM de fácil borrado y escritura y con una capacidad de grabación de datos suficientes para su utilización. Además presenta un menor desgaste por cada ciclo de lectura escritura, lo que implica un aumento en su vida útil.

Las funciones que desempeña el DSPIC30F4011 son leer el dato del convertidor analógico digital, leer los botones que indican la función que debe realizar, configurar el tiempo de muestreo, almacenar en la memoria EEPROM los datos leídos y recuperarlos para que sean enviados vía MAX – 232 hacia la computadora.

### 2.1.2 INTERRUPTORES DE CONTROL E INDICADORES

Como el circuito debe cumplir algunas funciones específicas, se han dispuesto algunos interruptores que el DSPIC debe revisar para determinar que tarea se realiza en un momento dado. El botón marcado START le indica que debe empezar a tomar los datos del convertidor A/D y almacenarlos secuencialmente en la memoria EEPROM destinada para tal fin; el botón CLEAR le indica al DSPIC que debe borrar el contenido de la memoria EEPROM para empezar nuevamente el proceso de registro de datos, y el botón TX_PC, sirve para ordenar el momento en el que el DSPIC debe empezar a transmitir los datos hacia la computadora los mismos que se encontraban en la memoria EEPROM.

Para conocer el estado del DSPIC en todo momento se ha dispuesto de LEDs indicadores del funcionamiento. Así cuando el DSPIC se encuentra en estado de muestreo (es decir que está tomando muestras y almacenando los datos en memoria) el LED enciende durante el período de muestreo. El sistema también posee otros dos LEDs, uno sirve para indicar que la fuente de alimentación del sistema está activada (POWER), y el otro sirve para indicar que se está haciendo transmisión de datos (TX_TEST) hacia la computadora y funciona gracias a que la señal varía entre niveles altos y bajos.

### 2.1.3 INTERFAZ DE COMUNICACIÓN MAX – 232.

Dado que el equipo debe transmitir los datos almacenados hacia una computadora, debe existir algún circuito que se encargue de convertir los niveles de voltaje TTL que maneja el DSPIC en niveles válidos para un puerto MAX – 232. En este caso, el nivel positivo se toma de la fuente de alimentación (+5V).

El datalogger envía la información  a través del puerto serial RS-232 a una velocidad de 57600BPS (bits por segundo) y con una configuración 8,n,1 (datos de 8 bits, sin paridad, 1 stop bit). El circuito de conversión TTL a MAX – 232 utilizado funciona bien en cortas distancias.

### 2.1.4 DIAGRAMA ESQUEMÁTICO DEL DATALOGGER

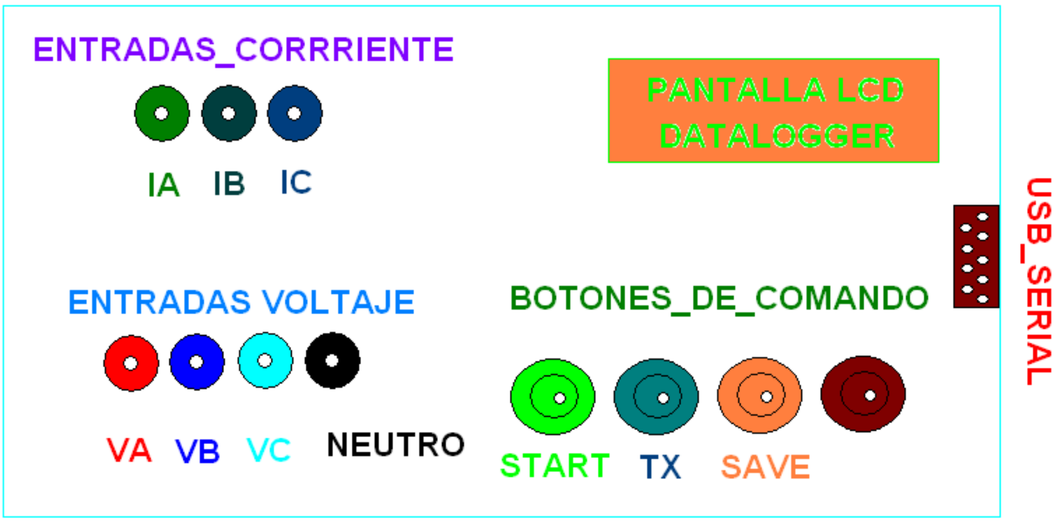
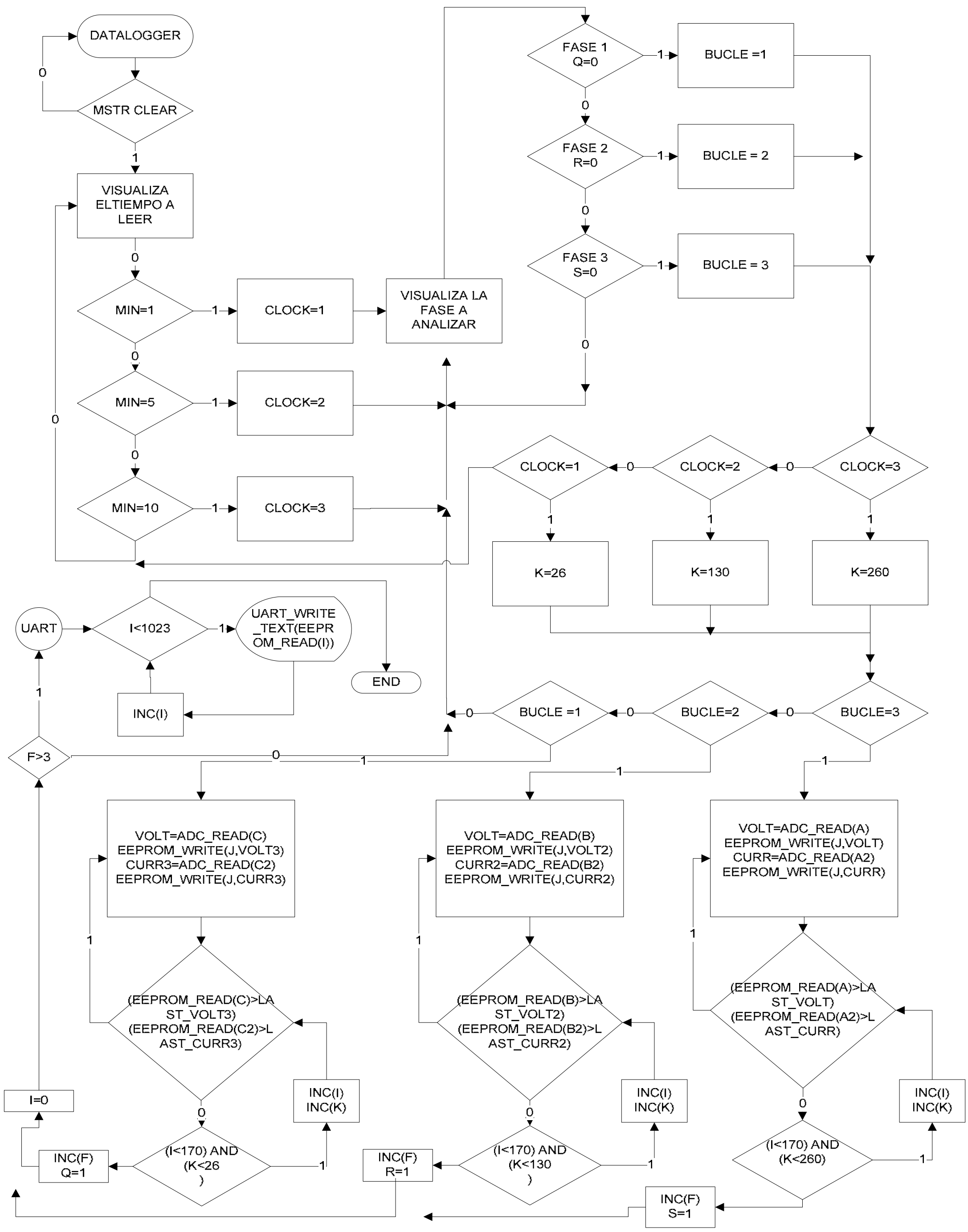



Fig. 2.1.4

## DIAGRAMA DE FLUJO DEL CÓDIGO FUENTE DEL DSPIC



# 2 DSPIC 30F4011

## 3.1 INTRODUCCIÓN A LOS DSPIC

Los DSPIC son posteriores a los DSP y a los MCU. En su diseño han participado expertos y especialistas de muchas áreas. Los DSPIC se han aprovechado de la experiencia acumulada por otros fabricantes.

Microchip, fabricante de los DSPIC, los ha bautizado con el nombre de DSC (Digital Signal Controller), que puede ser traducido como Controlador Digital de Señal.

## 3.2 PRINCIPALES CARACTERÍSTICAS DE LOS DSPIC

Entre las aportaciones típicas de los DSP que se han implementado en la arquitectura básica de los DSPIC destacan:

- Multiplicación MAC 16 x 16 en un ciclo.
- 2 acumuladores de 40 bits.
- Registro de desplazamiento de 40 bits para el escalado.
- Acceso simultáneo de dos operandos.
- Bucles con estructura DO y REPEAT.
- Bloque de registros de trabajo.
- Juego flexible de interrupciones.

- Perro guardián.
- Emulación en tiempo real.
- Optimización para programación en lenguaje C.

### 3.2.1 DESCRIPCIÓN DEL DSPIC30F4011

#### 3.2.1.1 RANGO DE FUNCIONAMIENTO

- DC – 30MIPS (30MIPS @ 4,5 – 5,5V, -40º a 85ºC).
- Voltaje de alimentación de 2,5 a 5,5V.
- Temperatura: interna de -40º a 85ºC y externa de -40º a 125ºC.

#### 3.2.1.2 CPU DE ALTO RENDIMIENTO

- Núcleo RISC con arquitectura Harvard modificada.
- Juego de instrucciones optimizado para el lenguaje C.
- Bus de datos de 16 bits.
- Bus de instrucciones de 24 bits.
- Repertorio de 84 instrucciones, la mayoría de una palabra de tamaño y ejecutable en un ciclo.
- Banco de 16 registros de propósito general de 16 bits.
- 2 acumuladores de 40 bits, con opciones de redondeo y saturación.

- Modos complejos de direccionamiento indirecto: modular o circular y de inversión de bits.
- Manejo de pila con software.
- Multiplicador para enteros y fraccionales de 16 x 16.
- División de 32/16 y 16/16.
- Operación de multiplicación y acumulación en un ciclo.
- Registro de desplazamiento de 40 bits.

#### 3.2.1.3 CONTROLADOR DE INTERRUPCIONES

- Latencia de 5 ciclos.
- Hasta 45 fuentes de interrupción, 5 externas.
- 7 niveles de prioridad, programables.
- excepciones especiales.

#### 3.2.1.4 ENTRADAS Y SALIDAS ANALÓGICAS/DIGITALES

- Hasta 40 patitas programables de E/S digitales.
- 25 mA de consumo por cada patita de E/S.

#### 3.2.1.5 MEMORIAS

- Memoria de programa FLASH de hasta 144KB con 100.000 ciclos de borrado/escritura.

- Memoria de datos EEPROM de hasta 4KB con 1.000.000 ciclos de borrado/escritura.
- Memoria de datos SRAM de hasta 8KB.

### 3.2.2 CONFIGURACIÓN DEL MÓDULO ANALÓGICO DIGITAL

La configuración comprende principalmente la selección del tiempo de muestreo y la de todos los pasos que se deben ejecutar para que el muestreo, retención y posterior codificación se realicen en los canales deseados, siguiendo el orden de la secuencia elegida.

Los pasos que se deben seguir para utilizar el módulo analógico digital a la hora de realizar una conversión son los siguientes:

Seleccionar las patitas como entradas analógicas, ADPCFG<15:0>.

Seleccionar la fuente de la tensión de referencia para que coincida el rango de conversión con el de las tensiones de entrada, ADCON2<15:13>.

Seleccionar el reloj de conversión analógico, respecto al reloj del procesador, ADCON3<5:0>, para que concuerde con la información a adquirir.

Determinar cuantos canales de muestreo y retención van a ser utilizados, ADCON2<9:8> y ADPCFG<15:0>.

Determinar como ocurrirá el muestreo ADCON1<3> y ADCSSL<15:0>.

Determinar cómo serán asignadas las entradas a los canales de muestreo y retención, ADCHS<15:0>.

Seleccionar la secuencia de muestreo y retención apropiada, ADCON1<7:0> y ADCON3<12:8>.

Seleccionar como se presentarán los resultados de la conversión en el buffer de resultado, ADCON1<9:8>.

Seleccionar el número de conversiones que se deben llevar a cabo para que se genere una interrupción, ADCON2<9:5>.

Encender el módulo analógico digital, ADCON1<15>.

### 3.2.3 CONVERSOR ANALÓGICO/DIGITAL

- Módulo conversor A/D de 10 bits y 500 Ksps, con 2 0 4 muestras simultáneas y hasta 16 canales de entrada. Conversión posible en el modo SLEEP.

- Módulo conversor A/D de 12 bits y 100 Ksps, con hasta 16 canales de entrada y conversión posible en el modo SLEEP como el anterior.

### 3.2.4 MÓDULO DE COMUNICACIÓN UART

El UART (Transmisor Receptor Universal Asíncrono) es un módulo para la comunicación serie asíncrona disponible en los DSPIC30F. Funciona como un sistema full – duplex o bidireccional asíncrono que puede adaptarse a multitud de periféricos, como ordenadores personales o interfaces RS – 232 y RS – 485.

Estas son las principales características del módulo UART:

- Funcionamiento en modo full – duplex con datos de 8 o 9 bits.
- Posibilidad de trabajar con paridad par, impar o sin paridad.
- Uno o dos bits de STOP.
- Contiene un generador de baudios con un prescaler de 16 bits que se encarga de generar la frecuencia de trabajo del módulo.
- Buffer de transmisión con capacidad para 4 caracteres.
- Buffer de recepción con capacidad para 4 caracteres.
- Posibilidad de emplear interrupciones para indicar la finalización de la transmisión o de la recepción.
- Patitas específicas TX y RX para transmitir y para recibir.

Las transferencias de información se realizan sobre dos líneas, UTX (transmisión) y URX (recepción), saliendo y entrando los bits por dichas líneas al ritmo de una frecuencia controlada internamente por el UART.

Cada palabra o dato de información se envía independientemente de los demás. Suele constar de 8 o 9 bits y van precedidos por un bit de START (inicio) y detrás de ellos se coloca un bit de STOP (parada), de acuerdo a las normas del formato estándar NRZ. Los bits son transmitidos a una frecuencia fija y normalizada.

El módulo UART está formado por estos tres grandes bloques que se muestran en la figura 3.2.4.

- Generador de baudios.
- Transmisor asíncrono.
- Receptor asíncrono.

Éste es el aspecto del diagrama de bloques que compone el módulo UART simplificado.

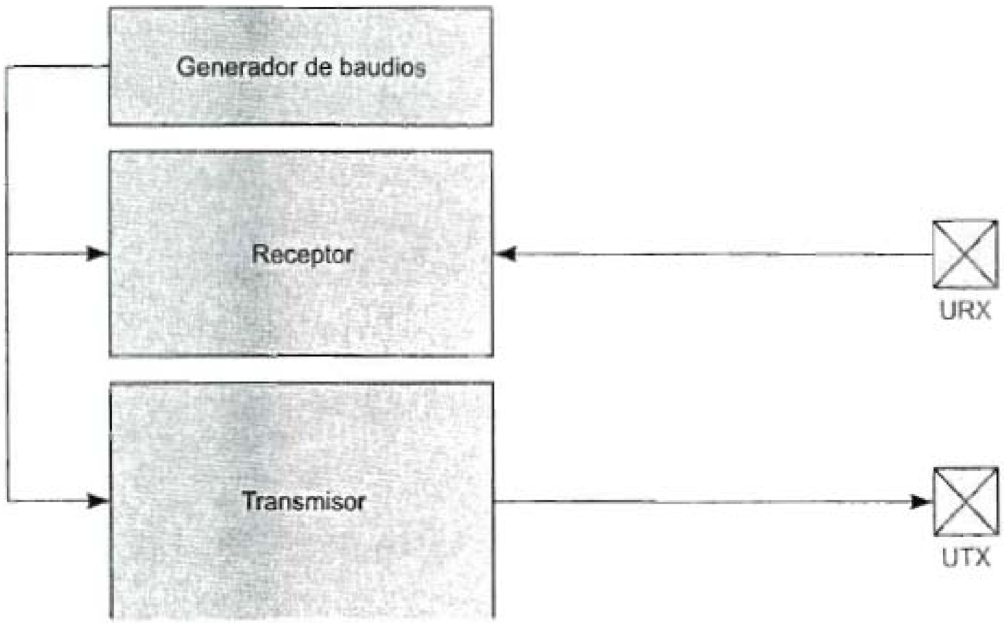



Fig. 3.2.4

### 3.2.5 MANEJO DE INTERRUPCIONES

Es posible deshabilitar interrupciones mediante la instrucción DISI que se ejecuta con más de 16.384 ciclos. Esto es efectivo cuando se deben ejecutar segmentos de código crítico.

La instrucción DISI solo desactiva interrupciones que estén en los niveles 1 – 6. Las interrupciones de nivel 7 y las excepciones se podrán atender aun estando activa la instrucción DISI.

Esta instrucción trabaja junto a un registro llamado DISICNT. Cuando éste no vale 0 las instrucciones de los niveles 1 – 6 se deshabilitan. Entonces el registro DISICNT comienza una cuenta atrás hasta que llega a 0 y habilita las interrupciones anteriores. El número de ciclos en que las interrupciones están deshabilitadas se puede modificar escribiendo en el registro un nuevo valor.

### 3.2.6 FUNCIONAMIENTO DE UNA INTERRUPCIÓN

Las instrucciones son atendidas en cada ciclo de instrucción. Cuando hay una petición de interrupción, IRQ, pendiente se señaliza poniendo el bit del flag a 1 en el registro IFS. Esta petición será atendida si el bit IECx del registro de permiso correspondiente está a 1.

Al atender la petición de interrupción ninguna instrucción abortará, sino que la que estaba en proceso cuando se le atendió la interrupción será completada cuando la interrupción termine.

Al atender a las interrupciones se evalúan los niveles de prioridad. Si hay una IRQ pendiente con un nivel de prioridad mayor que el proceso actual tiene la CPU, la interrupción se atenderá. Entonces el procesador deberá salvar la siguiente información, que permitirá al procesador el retorno a la instrucción en curso:

- El valor del Contador de Programa, PC.
- El byte bajo del Registro de Estado (SRI).
- El bit de estado IPL3.

La rutina de atención de la interrupción terminará con la instrucción RETFIE.

### 3.2.7 MEMORIA EEPROM

La memoria EEPROM de datos está mapeada en el espacio de la memoria de programa y está en palabras de 16 bits pudiendo alcanzar un tamaño de 2K palabras (4KB), según el modelo de DSPIC30F.

Para grabar la memoria EEPROM se usan las instrucciones de lectura y escritura de Tabla, pero como las posiciones de la EEPROM son de 16 bits, no serán necesarias las instrucciones TBLWTH y TBLRDH. Los procedimientos para la escritura y borrado de la EEPROM son similares a los del método RTSP de la memoria FLASH, pero en la EEPROM el acceso a los datos es más rápido y a diferencia de lo que ocurría con la memoria FLASH, la CPU no se detiene durante una operación de escritura/borrado de la EEPROM.

Las operaciones que puede realizar en la memoria EEPROM son:

- Borrar una palabra.
- Borrar una línea (16 palabras).
- Escribir una palabra.
- Escribir una línea (16 palabras)

El programa del usuario es responsable de esperar a que la operación de escritura/borrado se complete. Para ello, el fin de las operaciones de escritura/borrado en la EEPROM se puede detectar mediante uno de estos tres métodos:

- Explorar el bit WR por software. El bit WR será borrado cuando la operación se complete.
- Explorar el bit NVMF por software. Este bit será activado cuando la operación se complete.
- Habilitar las interrupciones NVM. La CPU recibirá una interrupción cuando la operación se complete.

# 4 ACONDICIONADORES DE SEÑAL

## 4.1 CIRCUITOS ACONDICIONADORES DE SEÑAL

### 4.1.1 TRANSFORMADOR DE VOLTAJE

El **transformador** es un dispositivo que se encarga de "transformar" el voltaje de corriente_alterna que tiene a su entrada en una amplitud diferente, que entrega a su salida.

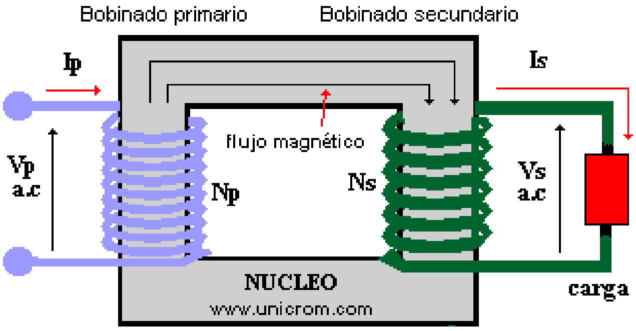



Fig. 4.1.1

Se compone de un núcleo de hierro sobre el cual se han enrollado varias espiras (vueltas) de alambre conductor.

Este conjunto de vueltas se llaman bobinas y se denominan:

Bobina primaria o "primario" a aquella que recibe el voltaje de entrada y Bobina secundaria o "secundario" a aquella que entrega el voltaje transformado.

- La Bobina primaria recibe un voltaje alterno que hará circular, por ella, una corriente alterna.

- Esta corriente inducirá un flujo magnético en el núcleo de hierro.

- Como el bobinado secundario está enrollado sobre el mismo núcleo de hierro, el flujo magnético circulará a través de las espiras de éste.

- Al haber un **flujo magnético** que atraviesa las espiras del "Secundario", se generará por el alambre del secundario un voltaje. En este bobinado secundario habría una corriente si hay una carga conectada (el secundario conectado por ejemplo a un resistor)

La razón_de_transformación del voltaje entre el bobinado "Primario" y el "Secundario" depende del número de vueltas que tenga cada uno. Si el número de vueltas del secundario es el triple del primario. En el secundario habrá el triple de voltaje. La fórmula:

Entonces: **Vs = Ns x Vp / Np**

Un **transformador** puede ser "elevador o reductor" dependiendo del número de espiras de cada bobinado. Si se supone que el **transformador** es ideal. (La potencia que se le entrega es igual a la que se obtiene de él, se desprecian las pérdidas por calor y otras), entonces:

Potencia de entrada (Pi) = Potencia de salida (Ps).

**Pi = Ps**

Si tenemos los datos de corriente y voltaje de un dispositivo, se puede averiguar su potencia usando la siguiente fórmula.

Potencia = voltaje x corriente

**P = V x I (en watts)**

Aplicando este concepto al **transformador** y como

P (bobinado primario) = P (bobinado secundario)

Entonces,

La única manera de mantener la misma potencia en los dos bobinados es que cuando el voltaje se eleve, la corriente se disminuya en la misma proporción y viceversa.

Entonces:

$$\frac{número\_espiras\_del\_primario(Np)}{número\_espiras\_del\_\sec undario(Ns)} = \frac{Corriente\_en\_el\_\sec undario(Is)}{Corriente\_en\_el\_primario(Ip)}$$

As

í, para conocer la corriente en el secundario (Is) cuando tengo:

- Ip (la corriente en el primario),
- Np (espiras en el primario) y
- Ns (espiras en el secundario)

Se utiliza siguiente fórmula:

**Is = Np x Ip / Ns**

### 4.1.2 FILTROS

La utilización de filtros capacitivos es necesaria para evitar los ruidos y estabilizar la señal adquirida. Además se utilizó las capacitancias como acople de impedancias tal como se muestra en la figura 4.1.5

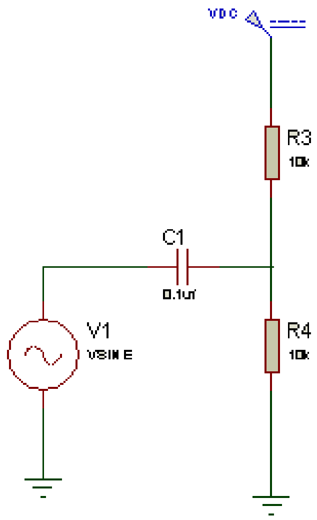



Fig. 4.1.5

### 4.1.3 ACONDICIONADOR DE SEÑAL DEL DISPOSITIVO

La señal de voltaje de fase del motor es reducida a través de un transformador reductor 110Vrms/6Vrms, luego esta a su vez es acondicionada entre un nivel máximo de voltaje DC de 5 voltios aplicando el principio de la resistencia de Thevenin al voltaje pico pico de la señal reducida. Una vez que la señal está en el rango deseado se colocó un capacitor como un acople de impedancia el cual mientras más grande es provoca que la reactancia tienda a cero, llegando a ser un cortocircuito lo cual le da estabilidad a la señal. Por último a esta señal le damos un nivel DC de 2,5 voltios para mantener la señal estable teniendo como eje horizontal 2,5 voltios, teniendo así una onda DC que puede ser leída por el DSPIC30F4011.

En la figura 4.1.3a se muestra el esquema del circuito utilizado para acondicionar la señal de voltaje del motor.

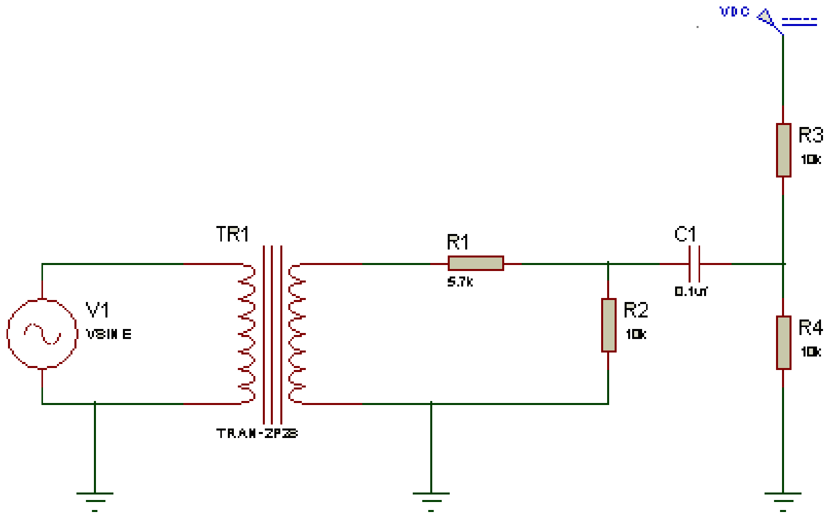



Fig. 4.1.3a

Teniendo en cuenta el análisis profundo de cada señal se ha procedido a realizar una simulación la cual es mostrada en la figura 4.1.3b en la cual se visualizan las señales de voltaje de la línea de entrada 170 voltios pico a pico o 120 Vrms (Señal de color amarilla), la señal de voltaje de la línea de entrada reducida 13,6 voltios pico a pico o 6 Vrms (Señal de color azul), y por último la señal DC deseada con un nivel de 2.5 voltios oscilando entre 1,7 voltios y 3,3 voltios (Señal de color roja).

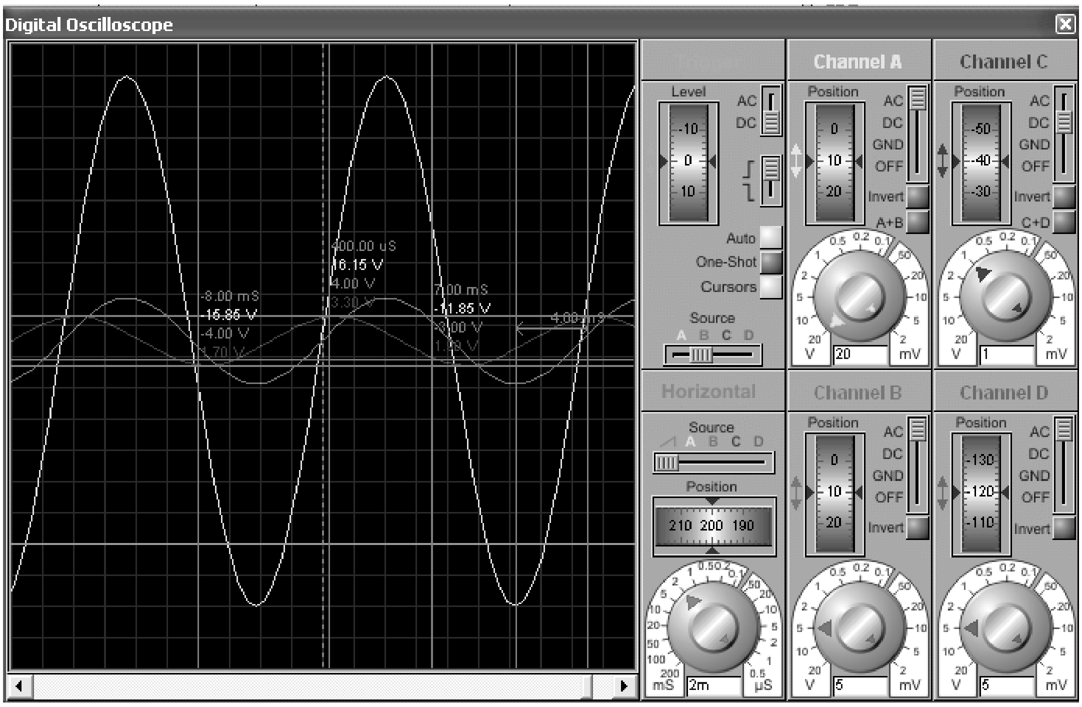

Digital Oscilloscope
Trigger
Channel A
Channel C
Horizontal
Channel B
Channel D
Level
AC
DC
Auto
One-Shot
Cursors
Source
A B C D
Position
GND
OFF
Invert
A+B
C+D
400.00 uS
16.15 V
4.00 V
3.30 V
-8.00 mS
-15.85 V
-4.00 V
1.70 V
7.00 mS
-11.85 V
-3.00 V
210 200 190
2m
20
1
5
5

**Fig. 4.1.3b**

# FLUKE 80i – 110s AC/DC

## INTRODUCCIÓN A LA SONDA DE CORRIENTE

La Fluke 80i-110s es una Sonda de corriente de CA/CC de pinza diseñada para reproducir formas de onda de corriente encontradas en sistemas comerciales e industriales modernos de distribución de energía eléctrica. El funcionamiento de la sonda se ha optimizado para la reproducción precisa de corrientes a la frecuencia de red y hasta el quincuagésimo armónico. La 80i-110s es compatible también con cualquier instrumento capaz de medir milivoltios. La Sonda de corriente (que se muestra en la Figura 5.1) presenta las siguientes ventajas:

• Mediciones de corriente exactas de CA, CC y CA+CC para aplicaciones eléctricas, electrónicas y de automoción.

• Blindada para proporcionar una alta inmunidad a las interferencias en las proximidades de dispositivos electrónicos de accionamiento por motor y sistemas de encendido.

• Amplio rango de medida entre 50 miliamperios y 100 amperios; útil para 10 miliamperios.

• Mandíbula diseñada con una forma que permite un fácil acceso a espacios estrechos.

• Conector BNC aislado para 600 voltios con diseño de seguridad – compatible con los ScopeMeter® de Fluke, analizadores de componentes armónicas de potencia, y osciloscopios.

• Salida seleccionable de 10 milivoltios por cada 1 amperio para el rango de 100 A, y 100 milivoltios por cada 1 amperio para el rango de 10 A.

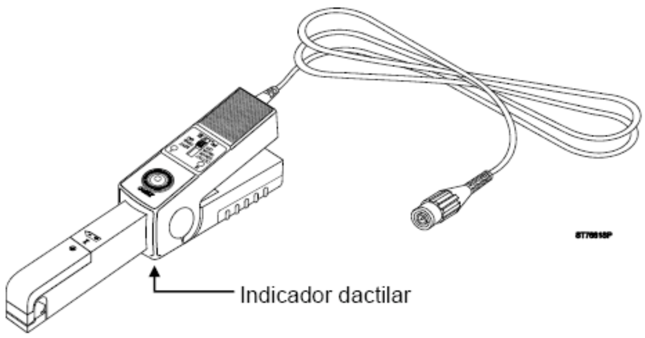



Fig. 5.1

## 4.2 DESEMBALAJE

En el estuche de la Sonda de corriente deben estar incluidos los siguientes elementos:

• Sonda de corriente de CA/CC 80i-110s

• Hoja de instrucciones (esta publicación)

• Batería de 9 voltios, modelo IEC 6LR61

Revise el contenido para comprobar si está completo. Si algún elemento contenido en el estuche ha sufrido daños o no ha sido incluido en el mismo, póngase en contacto inmediatamente con su distribuidor o con la oficina de ventas o centro de servicio FLUKE más próximo.

## 4.3 INSTALACIÓN DE A BATERÍAS

No olvide instalar la batería cuando utilice la sonda por primera vez.

Tomando como referencia la Figura 5.3, utilice el siguiente procedimiento para instalar la batería:

1. Asegúrese de que ha retirado la Sonda de corriente de cualquier conductor y ha desconectado el instrumento de medida.

2. Asegúrese de que la Sonda de corriente está apagada (posición OFF).

3. Localice la tapa del compartimiento de la batería en la empuñadura (véase la Figura 5.3). Afloje el tornillo con un destornillador de punta plana.

4. Haga deslizar la tapa del compartimiento de la batería retirándola de la Sonda de corriente.

5. Instale o sustituya la batería (modelo IEC 6LR61) como se indica en la Figura 5.3. Coloque los cables de la batería de forma que no queden aprisionados entre la parte inferior de la empuñadura y la tapa del compartimiento de la batería.

6. Vuelva a instalar la tapa del compartimiento de la batería y apriete el tornillo.

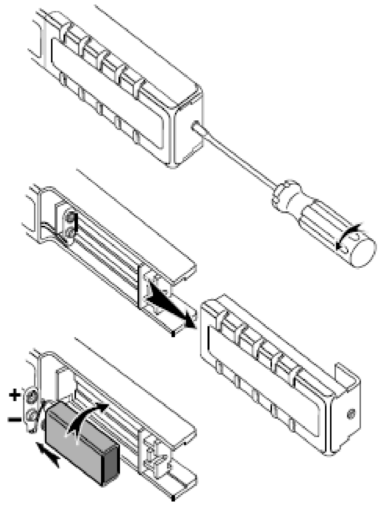



Fig. 5.3

## 4.4 INFORMACIÓN SOBRE SEGURIDAD

• Lea las instrucciones de uso antes de utilizar el instrumento y siga todas las instrucciones de seguridad.

• Emplee la sonda amperimétrica únicamente como se indica en las instrucciones de uso, de lo contrario, las características de seguridad de la sonda pueden no protegerlo.

• Siga los códigos de seguridad locales y nacionales. En lugares donde haya conductores vivos expuestos, debe utilizarse equipo de protección individual para evitar lesiones por descargas eléctricas y arcos.

• No sostenga la pinza amperimétrica desde la parte posterior a le indicador dactilar, consulte la figura 5.4.

• Antes de cada uso, examine el instrumento. Compruebe que no tenga grietas ni falten partes de la caja o aislamiento del cable de salida. Asegúrese también de que no haya componentes sueltos o flojos. Preste especial atención al aislamiento que rodea las mordazas.

• Verifique las superficies magnéticas de acoplamiento de la mordaza de la sonda. Estas no deberán tener polvo, suciedad, herrumbre ni otras materias extrañas.

• Nunca utilice la sonda amperimétrica en un circuito con tensiones superiores a 600 V CAT II o 300 V CAT III.

Los equipos CAT II están diseñados para proteger contra corrientes transitorias en circuitos conectados directamente a la instalación de baja tensión, como aparatos electrodomésticos, herramientas portátiles y equipos similares.

El equipo CAT III está diseñado para proteger contra corrientes transitorias en los equipos empleados en instalaciones de equipo fijo, tales como los paneles de distribución, alimentadores, circuitos de ramales cortos y los sistemas de iluminación de grandes edificios.

• Tenga extrema precaución al trabajar cerca de conductores sin aislamiento o barras colectoras. El contacto con el conductor podría producir una descarga eléctrica.

• Tenga cuidado al trabajar con tensiones superiores a 60 V CC, 30 V CA valor eficaz o 42 V CA pico. Estas tensiones representan peligro de descarga eléctrica.

• El uso de este equipo de manera no especificada aquí podría afectar la protección proporcionada por éste.

## 4.5 ESPECIFICACIONES ELÉCTRICAS

Todas las Especificaciones eléctricas son válidas a una temperatura de 23 °C ±3 °C (73 °F ±5 °F).

| | |
|---|---|
| **Rangos de corriente:** | 0 a 10 A cc o ca máx. |
| | 0 a 100 A cc o ca máx. |
| **Señales de salida:** | Rango de 10 A: 100 mV/A |
| | Rango de 100 A: 10 mV/A |

**Precisión básica (CC a 1 kHz):**

| Corriente de entrada (CC o CA máx.) | Error (después de la comprobación del cero): | |
|---|---|---|
| | 100 mV/A | 10 mV/A |
| 0 a 10 A | <3% de lectura +50 mA | - |
| 0 a 40 A | - | <4% de lectura +50 mA |
| 40 a 80 A | - | <12% de lectura +50 mA |
| 80 a 100 A | - | <15% de lectura |

**Precisión expandida:**

Para otras frecuencias, consulte el rango apropiado de corrientes de entrada y añada el error indicado más abajo al error de **"Precisión básica".**

Impedancia de carga de entrada (del instrumento principal):

| Frecuencias | Error adicional | |
|---|---|---|
| | 100 mV/A | 10 mV/A |
| 1 a 5 kHz | 3% | 3% |
| 5 a 20 kHz | 12% | 12% |
| >20 kHz | no especificado | no especificado |

>1 MΩ en paralelo con un máximo de 100 pF

**Ancho de banda útil (-3 dB):** 0 a 100 kHz

**Tiempo de subida o bajada:** <4 µseg.

**Nivel de ruido de salida:** 10 mV/A Valor típico: 480 µV pico a pico

**100 mV/A Valor típico:** 3 mV pico a pico

**Corriente máxima no destructiva:** 0 a 2 kHz 140 A máx.

2 a 10 kHz 110 A máx.

10 a 20 kHz 70 A máx.

20 a 50 kHz 30 A máx.

50 a 100 kHz 20 A máx

**Coeficiente de temperatura:**

2000 ppm/°C máx. para temperaturas entre 0 y 50 °C (32 y 132 °F).

## ESPECIFICACIONES DE SEGURIDAD

**Dimensiones:** 67 x 231 x 36 mm (2,6 x 9,1 x 1,4 pulgadas)

**Peso:** 330 g (11,6 onzas), incluida la batería

**Cable de salida:** 1,6 metros (63 pulgadas)

**Tamaño máximo de conductores**: 11,8 mm (0,46 pulgadas)

**Abertura máxima de mandíbulas**: 12,5 mm (0,49 pulgadas)

**Temperatura:** En funcionamiento: 0 a 50 °C (32 a 122 °F)

**En almacenamiento:** -30 a 70 °C (-22 a 158 °F)

**Humedad relativa (En funcionamiento):** 0 a 85% (0 a 35°C; 32 a 95°F) ó a 45% (35 a 50°C; 95 a 122°F)

**Altitud: En funcionamiento:** 0 a 2000 metros (0 a 6560 pies)

**En almacenamiento:** 0 a 12000 metros (0 a 40000 pies)

**Para desmagnetizar la sonda:** Abra y cierre varias veces las mordazas de la sonda

**INFORMACIÓN SOBRE LA BATERÍA**

**Batería:** 9 voltios, IEC 6LR61

**Consumo:** Valor típico: 8,6 mA

Valor máximo: 12 mA

**Vida útil:**

Valor típico: 55 horas, cuando se utiliza Alcalina IEC6LR61

Valor mínimo: 40 horas, cuando se utiliza Alcalina IEC 6LR61

**Indicador de batería (ON):**

La intensidad de iluminación del LED de color verde se reduce cuando la tensión de la batería es inferior a 6,5V.

**Indicador de sobrecarga (OL):** El LED de color rojo indica que la forma de onda o el impulso están fuera del rango.

# COMPATIBILIDAD CON INSTRUMENTOS

La 80i-110s es compatible con cualquier ScopeMeter, analizador de componentes armónicas de potencia, osciloscopio, multímetro u otro instrumento de medida de tensión que tenga las siguientes características:

• Un conector BNC. También se puede utilizar un adaptador banana a BNC (pedir la pieza PM9081/001 de Fluke) con entradas estándar en un multímetro digital (DMM).

• Precisión de entrada del 2% o mayor para aprovechar al máximo la precisión de la sonda.

• Impedancia de entrada igual o mayor de 1 M. en paralelo con un máximo de 100 pF.

• Banda de paso superior a cuatro veces la frecuencia de la forma de onda que se va a medir.

## 5.9 UTILIZACIÓN DE LA SONDA DE CORRIENTE

Para utilizar la sonda de corriente, observe las siguientes directrices al realizar mediciones:

1. Conecte la Sonda de corriente 80i-110s a la entrada deseada del instrumento de medida. Cuando se utilice el ScopeMeter o un osciloscopio, deben tener entrada acoplada en continua. Cuando se utilice un multímetro digital, use el adaptador banana a BNC (PM9081/001) para conectar la sonda a la entrada.

2. En la Sonda de corriente, seleccione el rango de menor sensibilidad (10 mV/A). Asegúrese de que se enciende el piloto de color verde de indicación ON. Consulte la Figura 5.9a para ver las posiciones del selector y del indicador ON de color verde.
3. En la Sonda de corriente, gire la rueda ZERO para ajustar la lectura de la sonda a cero. Consulte la Figura 5.9b para ver la posición del botón giratorio ZERO.
4. Seleccione la sensibilidad apropiada de la sonda en su ScopeMeter u osciloscopio.
5. Coloque la Sonda de corriente alrededor del conductor; asegúrese de que la flecha marcada en la mandíbula de la Sonda de corriente apunta hacia la orientación correcta (véase la Figura 5.9a).

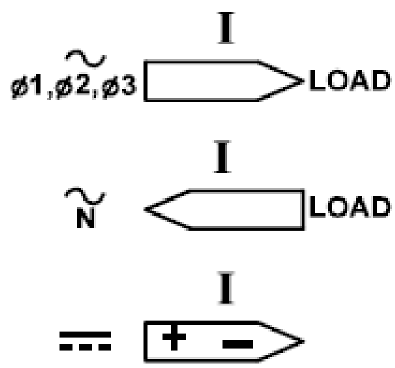



Fig. 5.9a

6. Observe el valor de la corriente o la forma de onda en la pantalla, o la lectura del valor de la corriente en el multímetro.
7. En el ScopeMeter, ajuste el mando de voltios/división y el mando de tiempo/división para conseguir la mejor presentación.
8. Si el indicador OL de color rojo se enciende, la Sonda de corriente está en situación de sobrecarga.
9. Al finalizar, asegúrese de ajustar el selector de rango en la posición OFF nuevamente.

En la Figura 5.9b se muestra una configuración de medida que utiliza la Sonda de corriente y un ScopeMeter.

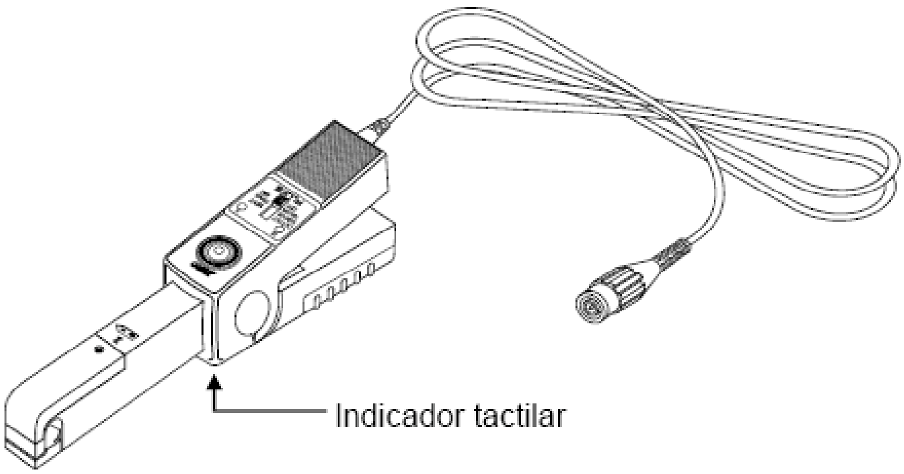



Fig. 5.9b

## 5.10 CONSIDERACIONES DE MEDICIÓN

Observe las siguientes directrices para colocar las mandíbulas de la sonda de corriente:

• Centre el conductor dentro de las mandíbulas de la sonda.

• Asegúrese de que la sonda está perpendicular al conductor.

• Asegúrese de que la flecha marcada en la mandíbula de la Sonda de corriente apunta hacia la dirección correcta. Observe las siguientes directrices:

• En la medida de lo posible, evite realizar mediciones cerca de otros conductores por los que circule corriente.

• En la Sonda de corriente, el rango de 100 mV/A ofrece la mayor precisión.

### 5.11 MANTENIMIENTO

Antes de cada utilización, asegúrese de que permanecen las condiciones de seguridad inspeccionando la sonda. Compruebe si existen grietas o faltan elementos del alojamiento de la sonda y el recubrimiento aislante del cable de salida, o bien, componentes sueltos o debilitados. Preste una atención especial al aislamiento que rodea las mandíbulas de la sonda. Si una sonda no supera esta inspección, manténgala en posición cerrada con cinta adhesiva para evitar que se utilice de forma involuntaria. Para comprobar el funcionamiento de la sonda, lleve a cabo el procedimiento de "Comprobación del funcionamiento".

Las reparaciones o tareas de mantenimiento no contempladas en este Manual de uso se deben realizar exclusivamente en un Centro de servicio Fluke. Una sonda que se encuentre bajo garantía será sustituida o reparada rápidamente (a criterio de Fluke) y se devolverá sin gastos.

## 5.12 LIMPIEZA Y ALMACENAMIENTO

Limpie periódicamente la caja del instrumento con un paño húmedo y detergente; no utilice materiales abrasivos ni disolventes. Abra las mandíbulas y limpie las piezas de los polos magnéticos con un paño ligeramente engrasado. No permita que se forme óxido o corrosión en los extremos de los núcleos magnéticos. Si no se va a utilizar la sonda durante períodos de tiempo superiores a 60 días, se debe retirar la batería y almacenarla en un lugar aparte.

## 5.13 COMPROBACIÓN DEL FUNCIONAMIENTO

Compruebe la precisión de la sonda con la configuración de prueba que se muestran en la Figura 5.13 El equipo de pruebas necesario se define en la Tabla1.

En la Figura 5.13 se muestra la estructura de la bobina toroidal.

Realice el siguiente procedimiento para comprobar la precisión de la sonda:

**Tabla 1. Equipo de pruebas necesario**

| NECESARIO | MODELO RECOMENDADO |
|---|---|
| Calibrador de CA/CC | Fluke modelo 5520A |
| Multímetro digital (DMM) | Fluke modelo 45 |
| Destornillador pequeño y aislado | Spectrol |
| Adaptador banana a BNC | Fluke modelo PM9081/001 |
| Bobina toroidal de 10 espiras | (véase la Figura 6) |

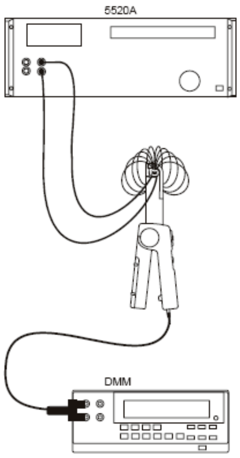



Fig. 5.1.6b

# 6 IMPLEMENTACIÓN DEL HARDWARE

## 6.1 DETALLES DE CONSTRUCCIÓN DEL DATALOGGER

### GRAFICAS DEL CIRCUITO IMPRESO DE LA TARJETA DE CONTROL.

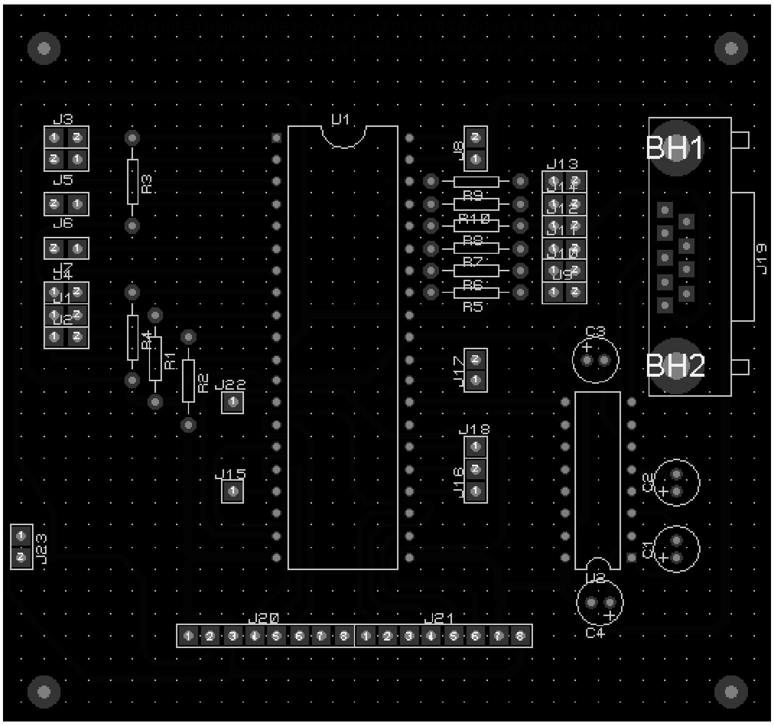
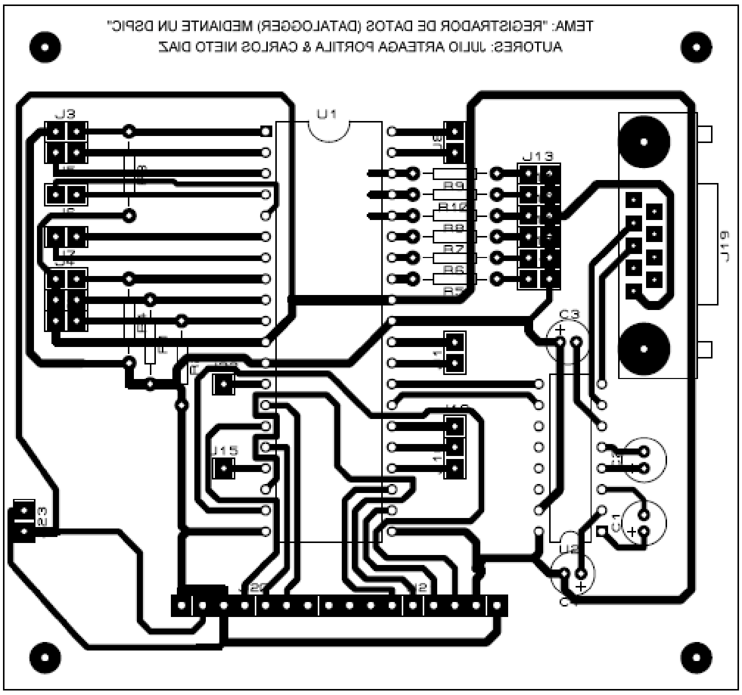
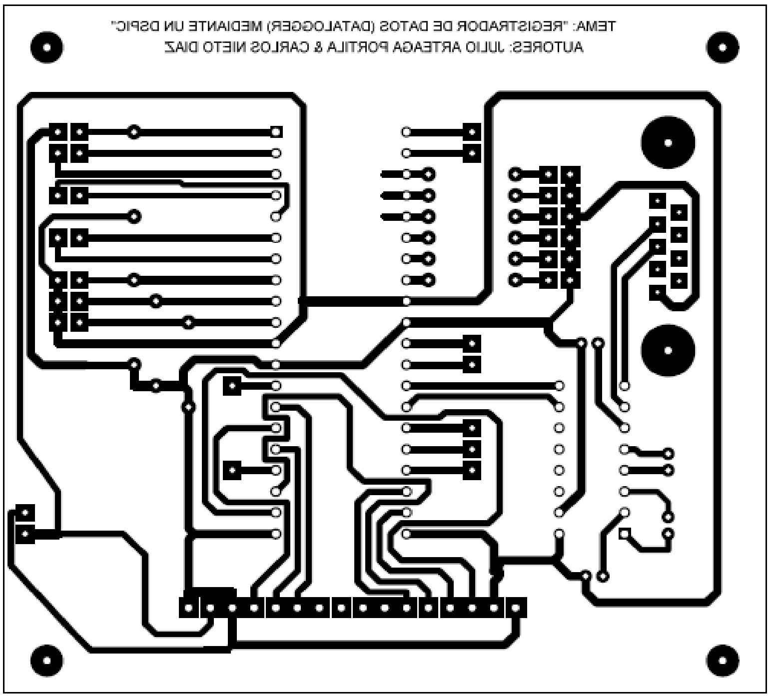



Fig. 6.1a


TEMA: "REGISTRADOR DE DATOS (DATALOGGER) MEDIANTE UN DSPIC"
AUTORES: JULIO ARTEAGA PORTILA & CARLOS NIETO DIAZ
U1
J3
J13
J19
J15
C3
U2
C1


TEMA: "REGISTRADOR DE DATOS (DATALOGGER) MEDIANTE UN DSPIC"
AUTORES: JULIO ARTEAGA PORTILA & CARLOS NIETO DIAZ

## GRAFICAS DEL CIRCUITO IMPRESO DE LA FUENTE DE ALIMENTACION

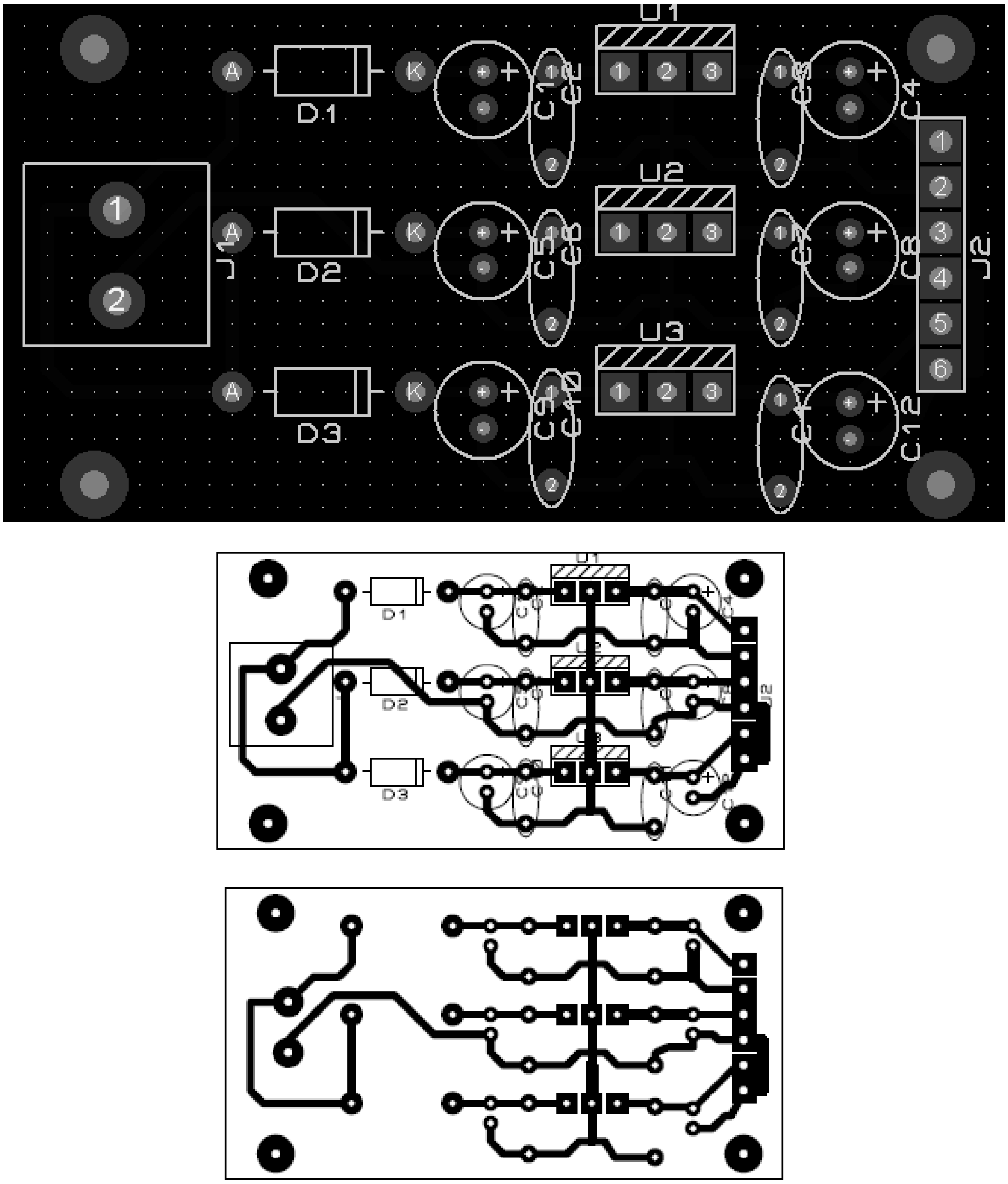
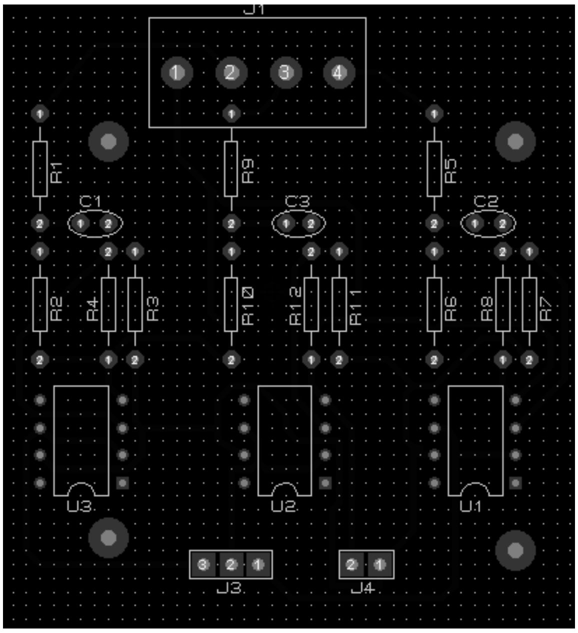
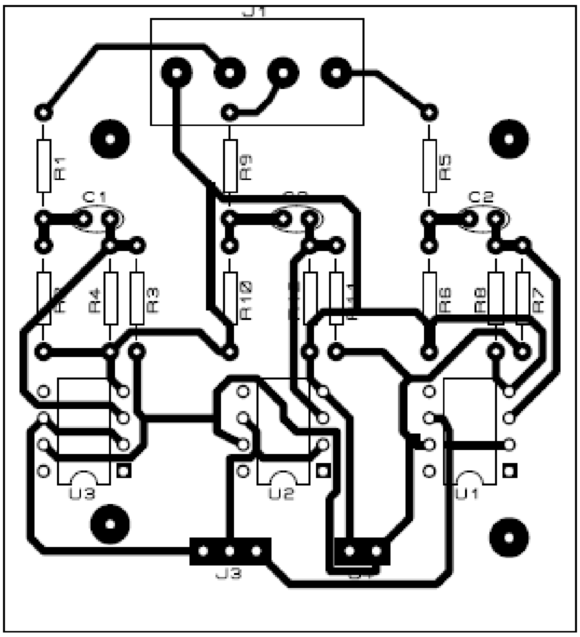



Fig. 6.1b

**GRAFICAS DEL CIRCUITO IMPRESO DEL ACONDICIONAMIETO DE LA SEÑAL DE VOLTAJE**



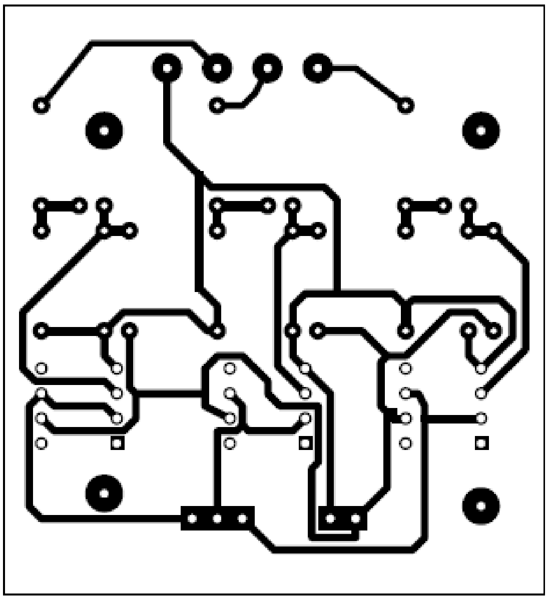
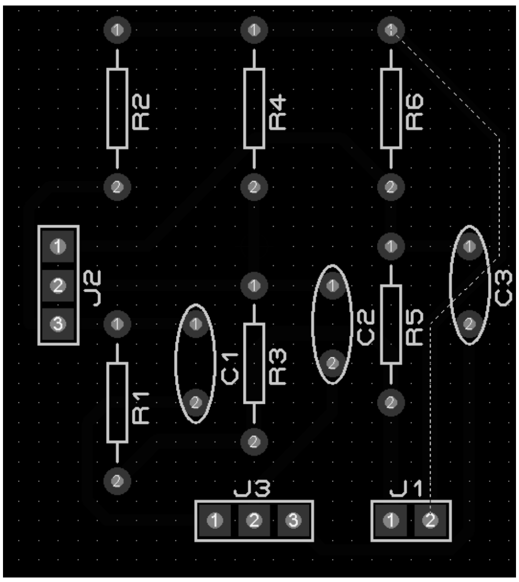


Fig. 6.1c

**GRAFICAS DEL CIRCUITO IMPRESO DEL ACONDICIONAMIETO DE LA SEÑAL DE CORRIENTE**



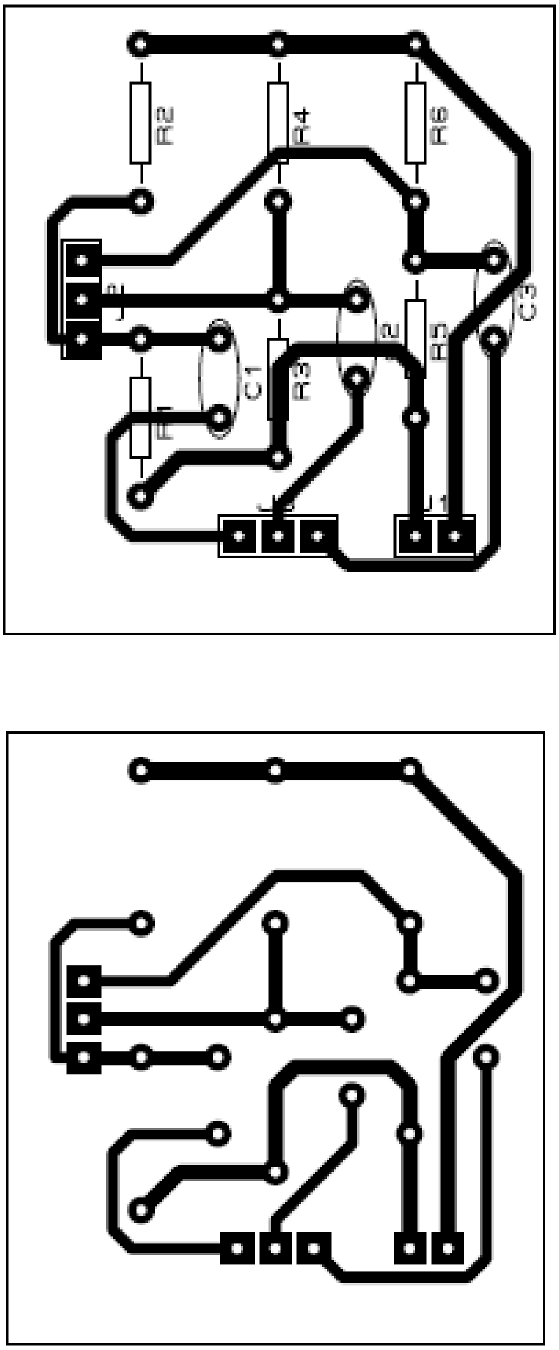



Fig. 6.1d

# 7 CONCLUSIONES Y RECOMENDACIONES

- Se logró implementar una tarjeta electrónica capaz de adquirir los datos de la prueba de medición de los voltajes de fase y corriente de fase de un motor trifásico.
- Con esta tarjeta se logró obtener una solución económica y confiable para este tipo de prueba que es practicada en la actualidad por industrias en general.
- Debido a las características internas del DSPIC30F4011, no fue necesario agregar módulos complementarios, tales como para la transferencias de datos. Logrando ahorro en la implementación de la tarjeta y que la PCB sea de menor coste debido a que sus dimensiones se redujeron.
- Se aplicaron las bases adquiridas y se mejoraron los conocimientos en mikrobasic y mikrobasic para DSPIC, que es un programador de gran utilidad y de manejo muy sencillo.
- Para la transferencia de los datos almacenados en la memoria EEPROM del DSPIC hacia la computadora fue de gran utilidad el integrado MAX 232 el cual nos brinda un buen soporte en la transmisión de datos garantizando así el envío correcto de los mismos.
- Mediante el desarrollo del proyecto se pudo mejorar los conocimientos adquiridos sobre electrónica tanto análoga como digital, además se trabajo con

algunos tipos de herramientas tanto eléctricas como mecánicas logrando así un mejor desempeño con el manejo de las mismas.

- Se logró profundizar el manejo de PROTEUS para la realización de circuitos impresos (PCB).
- Una modificación en el datalogger puede ser implementar una tarjeta de memoria flash, en donde se almacenaría mayor cantidad de datos de las pruebas, la misma que posteriormente sería retirada y llevada a un computador para el análisis posterior de los datos obtenidos.

# ANEXOS

## ANÁLISIS DE COSTOS DEL DATALOGGER

**LISTADO DE MATERIALES**
**MANUAL DE USUARIO**
**CODIGO FUENTE**

**TABLA DE COSTOS:**

**FUENTES**

| DESCRIPCIÓN | CANTIDAD | PRECIO UNITARIO | PRECIO TOTAL |
|---|---|---|---|
| **Componentes Electrónicos** | | | |
| Circuito Integrado LM7805 | 3 | $ 0,30 | $ 0,90 |
| Diodo 1N4007 | 3 | $ 0,04 | $ 0,12 |
| Capacitores Electrolíticos 10uF | 6 | $ 0,04 | $ 0,24 |
| Capacitores Cerámicos 0.1uF | 6 | $ 0,04 | $ 0,24 |
| Borneras | 2 | $ 0,20 | $ 0,40 |
| Espadines Macho | 6 | $ 0,02 | $ 0,12 |
| Espadines Hembra | 6 | $ 0,04 | $ 0,24 |
| PCB (FUENTE) | 1 | $ 2,00 | $ 2,00 |
| Portafusible | 1 | $ 0,22 | $ 0,22 |
| Fusible (3 Amp.) | 1 | $ 0,05 | $ 0,05 |
| Fusible (2 Amp.) | 1 | $ 0,05 | $ 0,05 |
| **TOTAL** | | | **$4,58** |

## ACONDICIONADOR DE SEÑAL DE VOLTAJES Y CORRIENTES

| DESCRIPCIÓN | CANTIDAD | PRECIO UNITARIO | PRECIO TOTAL |
|---|---|---|---|
| **Componentes Electrónicos** | | | |
| Amplifcador Operacional UA741 | 3 | $ 0,20 | $ 0,60 |
| Transformadores de Voltaje | 3 | $ 2,40 | $ 7,20 |
| Resistencias de 5.6K – ½ W | 3 | $ 0,05 | $ 0,15 |
| Resistencias de 10K – ½ W | 15 | $ 0,02 | $ 0,30 |
| Capacitores Cerámicos 0.1uF | 6 | $ 0,04 | $ 0,24 |
| Espadines Macho | 9 | $ 0,02 | $ 0,18 |
| Espadines Hembra | 9 | $ 0,04 | $ 0,36 |
| Socket (8 Pines) | 3 | $ 0,04 | $ 0,12 |
| Cable (Metros) | 3 | $ 0,65 | $ 1,95 |
| Conector Banana (Macho) | 2 | $ 0,15 | $ 0,30 |
| Conector Banana (Hembra) | 4 | $ 0,10 | $ 0,40 |
| Borneras | 4 | $ 0,10 | $ 0,40 |
| Conector Lagarto | 2 | $ 0,20 | $ 0,40 |
| Conetor BNC | 3 | $ 0,50 | $ 1,50 |
| PCB (SEÑAL DE VOLTAJE) | 1 | $ 2,40 | $ 2,40 |
| **TOTAL** | | | **$16,50** |

**DSPIC**

| DESCRIPCIÓN | CANTIDAD | PRECIO UNITARIO | PRECIO TOTAL |
|---|---|---|---|
| **Componentes Electrónicos** | | | |
| DSPIC30F4011 | 1 | $ 11,50 | $ 11,50 |
| Circuito Integrado Max232 | 1 | $ 3,00 | $ 3,00 |
| DB9 Serial (Conector Hembra) | 1 | $ 1,50 | $ 1,50 |
| Resistencias de 10K – ½ W | 4 | $ 0,02 | $ 0,08 |
| Resistencias de 330Ω – ½ W | 6 | $ 0,02 | $ 0,12 |
| Capacitores Electrolíticos 10uF | 4 | $ 0,04 | $ 0,16 |
| Botoneras | 4 | $ 0,10 | $ 0,40 |
| Leds Verdes | 4 | $ 0,06 | $ 0,24 |
| Leds Rojos | 2 | $ 0,06 | $ 0,12 |
| Socket (40 Pines) | 1 | $ 0,50 | $ 0,50 |
| Leds Amarillos | 2 | $ 0,06 | $ 0,12 |
| Espadines Macho | 53 | $ 0,02 | $ 1,06 |
| Espadines Hembra | 53 | $ 0,04 | $ 2,12 |
| PCB (DSPIC) | 1 | $ 5,90 | $ 5,90 |
| **TOTAL** | | | **$26,82** |

# ILUSTRACIONES

## GRAFICA DEL VOLTAJE ADQUIRIDO DE T1

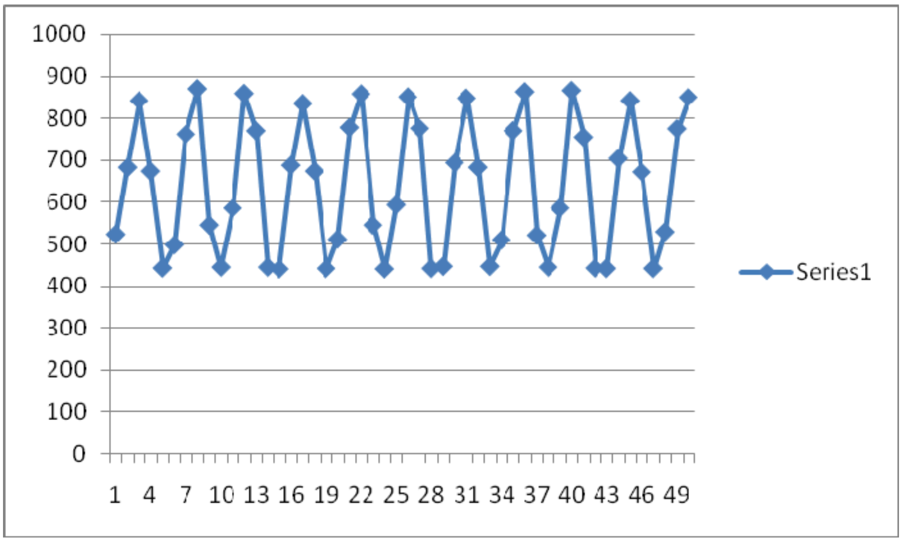



**Fig. A1**

## GRAFICA DEL VOLTAJE ADQUIRIDO DE T2

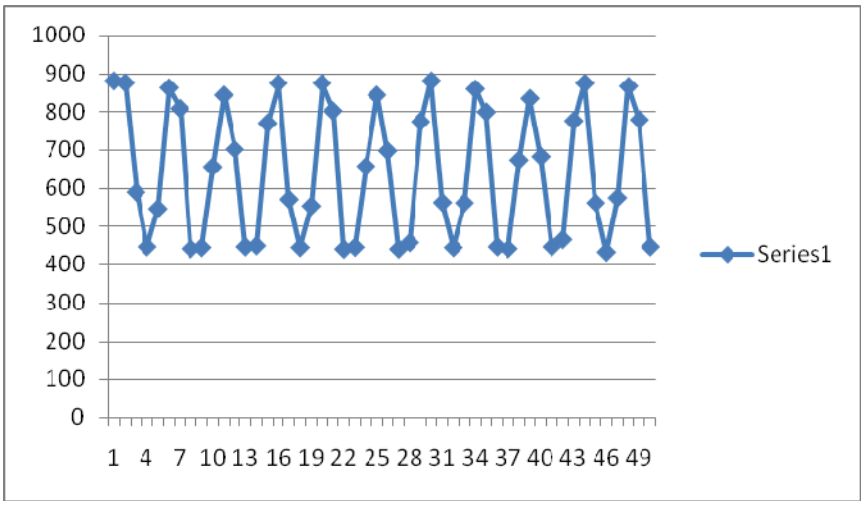



**Fig. A2**

**GRAFICA DEL VOLTAJE ADQUIRIDO DE T3**

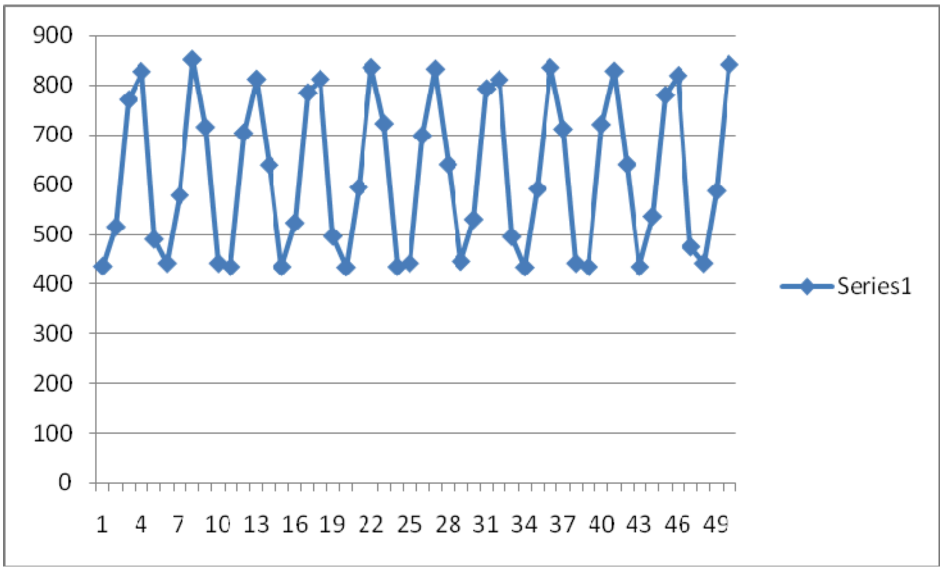



**Fig. A3**

**GRAFICA DE LA CORRIENTE ADQUIRIDA EN BNC1**

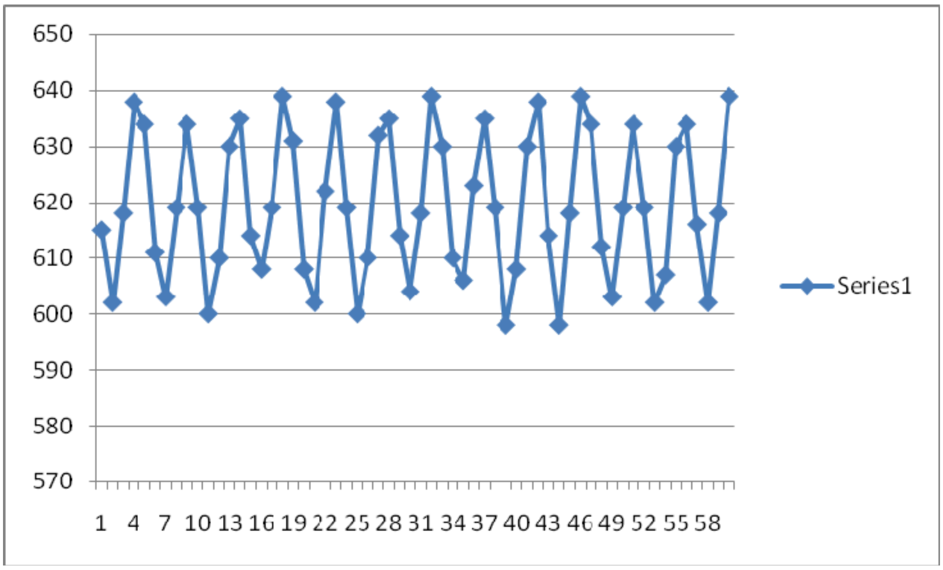



**Fig. A4**

## GRAFICA DE LA CORRIENTE ADQUIRIDA EN BNC2

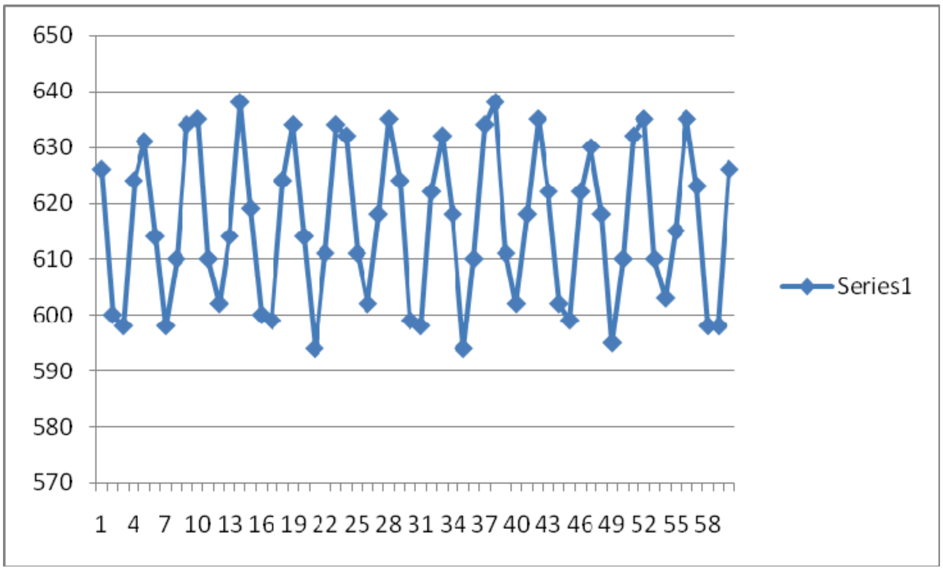



**Fig. A5**

## GRAFICA DE LA CORRIENTE ADQUIRIDA EN BNC3

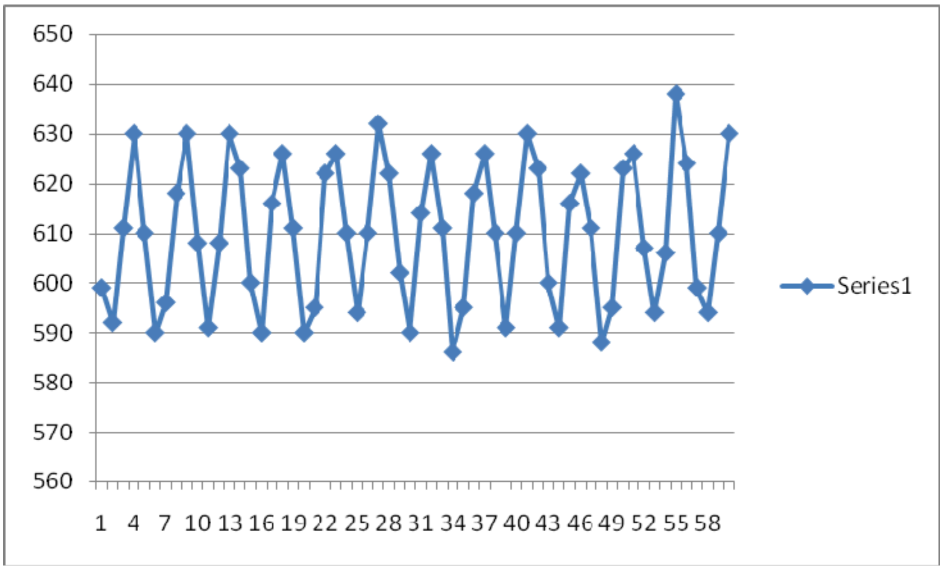



**Fig. A6**

## CÓDIGO FUENTE DE LA APLICACIÓN (SOFTWARE)

```
program data_logger

DIM I,J,K    AS WORD
DIM Q,R,S,F  AS BYTE

DIM M1,L1,O1  AS WORD
DIM M2,L2,O2  AS WORD

****NUMERO DE VECES QUE LAS SENALES FUERON ADQUIRIDAS****
DIM S_V1        AS WORD
DIM S_V2        AS WORD
DIM S_V3        AS WORD

DIM S_C1        AS WORD
DIM S_C2        AS WORD
DIM S_C3        AS WORD

'************ALMACENAN LAS SENALES ADQUIRIDAS*************
DIM VOLT                AS WORD
DIM VOLT_ANTERIOR       AS WORD
DIM VOLT_ANTERIOR2      AS WORD
DIM VOLT_ANTERIOR3      AS WORD
DIM CURRENT             AS WORD
DIM CURRENT_ANTERIOR    AS WORD
DIM CURRENT_ANTERIOR2   AS WORD
DIM CURRENT_ANTERIOR3   AS WORD
'**************ALMACENAN LAS VARIABLES SETEADAS***********
DIM CLOCK    AS BYTE
DIM BUCLE    AS BYTE
```

```
'************************************RANDOM FUNCTION*****************************
DIM EEADDR   AS LONGINT

DIM VOLT1    AS LONGINT
DIM VOLT2    AS LONGINT
DIM VOLT3    AS LONGINT

DIM CURRENT1 AS LONGINT
DIM CURRENT2 AS LONGINT
DIM CURRENT3 AS LONGINT
'******************************PROCEDIMIENTOS/FUNCIONES *********************
'*********************************** FUNCION ASSCII *********************************
'ESTA FUNCION NOS PERMITE CONVERTIR LOS DATOS A CARACTER
(STRING) PARA LUEGO SER
'ENVIADOS POR UART A TRAVES DEL PUERTO SERIAL

DIM DATO     AS WORD      ' DATO      : LECTURA DE LOS DATOS EN LA
EEPROM
DIM ENTERO   AS WORD    ' ENTERO    : DATO DEL PRIMER DIGITO
DIM DECIMAL1 AS WORD    ' DECIMAL1  : DATO DEL SEGUNDO DIGITO
DIM DECIMAL2 AS WORD    ' DECIMAL2  : DATO DEL TERCER DIGITO
DIM DECIMAL3 AS WORD    ' DECIMAL3  : DATO DEL CUARTO DIGITO

SUB FUNCTION ASCII(dim num as byte) as byte

   SELECT CASE NUM
      CASE 0   RESULT = $30     'NUMERO 0 EN ASSCII
      CASE 1   RESULT = $31     'NUMERO 1 EN ASSCII
      CASE 2   RESULT = $32     'NUMERO 2 EN ASSCII
      CASE 3   RESULT = $33     'NUMERO 3 EN ASSCII
      CASE 4   RESULT = $34     'NUMERO 4 EN ASSCII
      CASE 5   RESULT = $35     'NUMERO 5 EN ASSCII
      CASE 6   RESULT = $36     'NUMERO 6 EN ASSCII
      CASE 7   RESULT = $37     'NUMERO 7 EN ASSCII
      CASE 8   RESULT = $38     'NUMERO 8 EN ASSCII
```

```
      CASE 9   RESULT = $39     'NUMERO 9 EN ASSCII
   END SELECT
END SUB

'*********************************REAL CLOCK FUNCTION*************************

DIM CONT AS WORD

sub procedure Timer1Int org $1A
  '** it is necessary to clear manually the interrupt flag:
  IFS0 = IFS0 and $FFF7   ' Clear TMR1IF
  IF IFS0 <> $FFF7 THEN
    INC (CONT)                ' INCREMENTO DE CONTADOR
  END IF
end sub

'********************************** PROGRAMA PRINCIPAL*************************
main:

'******CONFIGURANDO PARAMETROS PARA UTILIZAR EL TIMER1 COMO
INTERUPCION*****

  CONT = 0                   'ENCERANDO EL CONTADOR (CLOCK)
  IPC0  = IPC0 or  $1000      ' Interrupt priority level = 1
  IFS0  = IFS0 and $FFF7      ' Clear TMR1IF
'*************************************************************************************
  ADPCFG = $FFC0            'CONFIGURACION DE LOS PUERTOS DE I/O
  UART2_INIT(57600)        'INICIALIZANDO LA COMUNICAION POR UART
```

```
'INICIALIZANDO Y CONFIGURANDO LA LCD, EN LOS PUERTOS "D" Y "C"
CON SUS PINES RESPECTIVOS
 LCD_INIT(PORTD, 3,2,1,0, PORTC, 13,14,15) 'RD3->MSB DE LA LCD
                                          ‘ RB13->RS
                                          'RD2->...        RB14->R/W
                                          'RD1->...        RB15->ENABLE
                                          'RD0->LSB DE LA LCD
 LCD_CMD(LCD_CURSOR_OFF)
 EEADDR   = $0
 VOLT1    = $7FFC00
 VOLT2    = $7FFD58
 VOLT3    = $7FFEB0

 CURRENT1 = $7FFCAC
 CURRENT2 = $7FFE04
 CURRENT3 = $7FFF5C

 TRISE = $0000      'DECLARACION DEL PUERTO "E" COMO SALIDAS
 PORTE = $0000      'ENCERANDO EL PUERTO E
'  TRISC = $0000     'CONFIGURANDO EL PUERTO "C" COMO SALIDAS
'  PORTC = $0000     'ENCERANDO EL PUERTO C
'****CONFIGURANDO LOS PINES DEL PUERTO "B" COMO ENTRADAS
ANALOGICAS****
 TRISB.0 = 1              'CONFIGURANDO EL PUERTO "B" PIN 0 COMO
ENTRADA ANALOGICA
 TRISB.1 = 1              'CONFIGURANDO EL PUERTO "B" PIN 1 COMO
ENTRADA ANALOGICA
 TRISB.2 = 1              'CONFIGURANDO EL PUERTO "B" PIN 2 COMO
ENTRADA ANALOGICA
 TRISB.3 = 1              'CONFIGURANDO EL PUERTO "B" PIN 3 COMO
ENTRADA ANALOGICA
 TRISB.4 = 1              'CONFIGURANDO EL PUERTO "B" PIN 4 COMO
ENTRADA ANALOGICA
 TRISB.5 = 1              'CONFIGURANDO EL PUERTO "B" PIN 5 COMO
ENTRADA ANALOGICA
```

```
'*****************************************************************************************
 F       = 0
 Q       = 0
 R       = 0
 S       = 0
 S_V1    = 0
 S_V2    = 0
 S_V3    = 0
 S_C1    = 0
 S_C2    = 0
 S_C3    = 0
'**************************************MENU PRINCIPAL*****************************
 PORTE.0 = 1                'LED INDICADOR DE ENDENDDO DEL
DATALOGGER(LED ROJO)

LCD_CMD(LCD_CLEAR)   'BORANDO LA PANTALLA DE LA LCD
LCD_OUT(1,5, "MATERIA") 'MOSTRANDO LA PALABRA "MATERIA" EN LA
LCD
LCD_OUT(2,2, "DE GRADUACION")'MOSTRANDO LA FRASE "DE
GRADUACION" EN LA LCD
DELAY_MS(2000)              'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
LCD_OUT(1,4, "DATALOGGER")  'MOSTRANDO LA PALABRA
"DATALOGGER" EN LA LCD
LCD_OUT(2,1, "ARTEAGA & NIETO")'MOSTRANDO LA PALABRA "ARTEAGA
& NIETO" EN LA LCD
DELAY_MS(2000)             'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

'****************MENU DE SELECCION DEL TIEMPO DE MUESTREO***********
TIME_CHANGE:
  LCD_CMD(LCD_CLEAR)         'BORANDO LA PANTALLA DE LA LCD
  LCD_OUT(1,4, "SELECT THE")  'MOSTRANDO LA PALABRA "SELECT THE"
EN LA LCD
```

```
  LCD_OUT(2,1, "TIME TO ANALIZE")'MOSTRANDO LA FRASE "TIME TO
ANALIZE" EN LA LCD
  DELAY_MS(1500)              'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
  LCD_OUT(1,5, "TO SKIP")     'MOSTRANDO LA FRASE "TO SKIP" EN LA LCD
  LCD_OUT(2,3, "PUSH CHANGE") 'MOSTRANDO LA FRASE "PUSH
CHANGE" EN LA LCD
  DELAY_MS(1000)              'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

DO
'MENU DE SELECCION TIEMPO 1 MINUTO
  DO
   LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

   LCD_OUT(1,7, "TIME")      'MOSTRANDO LA FRASE "TIME" EN LA LCD
   LCD_OUT(2,4, "ONE MINUTE")'MOSTRANDO LA FRASE "ONE MINUTE" EN
LA LCD
   DELAY_MS(1000)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

   LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD
   PORTE.1 = 1                'LED INDICADOR ------

   LCD_OUT(1,5, "TO SAVE")   'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
   LCD_OUT(2,4, "PUSH SAVE") 'MOSTRANDO LA FRASE "PUSH SAVE" EN
LA LCD
   DELAY_MS(800)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
   IF (PORTB.7 = 1) THEN
     CLOCK = 1
     GOTO TIME_OK
   END IF
  LOOP UNTIL (PORTB.8 = 1)
```

```
  PORTE.1 = 0                    'LED INDICADOR ------
'MENU DE SELECCION TIEMPO 5 MINUTOS
  DO
    LCD_CMD(LCD_CLEAR)        'BORANDO LA PANTALLA DE LA LCD

    LCD_OUT(1,7, "TIME")      'MOSTRANDO LA FRASE "TIME" EN LA LCD
    LCD_OUT(2,3, "FIVE MINUTE")'MOSTRANDO LA FRASE "FIVE MINUTE" EN
LA LCD
    DELAY_MS(1000)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

    LCD_CMD(LCD_CLEAR)        'BORANDO LA PANTALLA DE LA LCD
    PORTE.1 = 1                    'LED INDICADOR ------

    LCD_OUT(1,5, "TO SAVE")    'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
    LCD_OUT(2,4, "PUSH SAVE") 'MOSTRANDO LA FRASE "PUSH SAVE" EN
LA LCD
    DELAY_MS(800)             'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
    IF (PORTB.7 = 1) THEN
      CLOCK = 2
      GOTO TIME_OK
    END IF
  LOOP UNTIL (PORTB.8 = 1)
  PORTE.1 = 0                    'LED INDICADOR ------
'MENU DE SELECCION TIEMPO 10 MINUTOS
  DO
    LCD_CMD(LCD_CLEAR)        'BORANDO LA PANTALLA DE LA LCD

    LCD_OUT(1,7, "TIME")      'MOSTRANDO LA FRASE "TIME" EN LA LCD
    LCD_OUT(2,4, "TEN MINUTE")'MOSTRANDO LA FRASE "TEN MINUTE" EN
LA LCD
    DELAY_MS(1000)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
```

```
   LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD
   PORTE.1 = 1                 'LED INDICADOR ------

   LCD_OUT(1,5, "TO SAVE")   'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
   LCD_OUT(2,4, "PUSH SAVE") 'MOSTRANDO LA FRASE "PUSH SAVE" EN
LA LCD
   DELAY_MS(800)           'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
   IF (PORTB.7 = 1) THEN
     CLOCK = 3
     GOTO TIME_OK
   END IF
  LOOP UNTIL (PORTB.8 = 1)
  PORTE.1 = 0                 'LED INDICADOR ------
TIME_OK:

LOOP UNTIL (PORTB.7 = 1)
PORTE.1 = 1                'LED INDICADOR ------
DELAY_MS(500)               'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
PORTE.1 = 0                'LED INDICADOR ------

'*************************MENU DE SELECCION DEL CANAL**********************
CHANGE_FASE:
  LCD_CMD(LCD_CLEAR)         'BORANDO LA PANTALLA DE LA LCD
  LCD_OUT(1,6, "SELECT")  'MOSTRANDO LA PALABRA "SELECT" EN LA
LCD
  LCD_OUT(2,1, "PHASE TO ANALIZE")'MOSTRANDO LA FRASE "PHASE TO
ANALIZE" EN LA LCD
  DELAY_MS(1500)             'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
  LCD_OUT(1,5, "TO SKIP")     'MOSTRANDO LA FRASE "TO SKIP" EN LA LCD
```

```
  LCD_OUT(2,3, "PUSH CHANGE") 'MOSTRANDO LA FRASE "PUSH
CHANGE" EN LA LCD
  DELAY_MS(1000)              'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
DO

'MENU DE SELECCION DE LA FASE 1
IF (Q=0) THEN
   DO
     LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

     LCD_OUT(1,5, "PHASE 1")    'MOSTRANDO LA FRASE "FASE 1" EN LA
LCD
     LCD_OUT(2,1, "VOLT. - CURRENT.")'MOSTRANDO LA FRASE "VOLT. -
CURRENT." EN LA LCD
     PORTE.3 = 1              'LED INDICADOR DE LA FASE 1.
     DELAY_MS(1000)           'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

     LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

     LCD_OUT(1,5, "TO SAVE")   'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
     LCD_OUT(2,4, "PUSH SAVE") 'MOSTRANDO LA FRASE "PUSH SAVE" EN
LA LCD
     DELAY_MS(800)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
     IF (PORTB.7 = 1) THEN
       BUCLE = 1
       GOTO SELEC_OK
     END IF
   LOOP UNTIL (PORTB.8 = 1)
ELSE
   NOP
END IF
```

```
  PORTE.3 = 0              'LED INDICADOR DE LA FASE 1.

'MENU DE SELECCION DE LA FASE 2
IF (R=0) THEN
   DO
     LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

     LCD_OUT(1,5, "PHASE 2")    'MOSTRANDO LA FRASE "FASE 2" EN LA
LCD
     LCD_OUT(2,1, "VOLT. - CURRENT.")'MOSTRANDO LA FRASE "VOLT. -
CURRENT." EN LA LCD
     PORTE.4 = 1             'LED INDICADOR DE LA FASE 2.
     DELAY_MS(1000)           'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

     LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

     LCD_OUT(1,5, "TO SAVE")   'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
     LCD_OUT(2,4, "PUSH SAVE") 'MOSTRANDO LA FRASE "PUSH SAVE" EN
LA LCD
     DELAY_MS(800)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
     IF (PORTB.7 = 1) THEN
       BUCLE = 2
       GOTO SELEC_OK
     END IF
   LOOP UNTIL (PORTB.8 = 1)
ELSE
   NOP
END IF

  PORTE.4 = 0              'LED INDICADOR DE LA FASE 2.
```

```
'MENU DE SELECCION DE LA FASE 3

IF (S=0) THEN
   DO
     LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

     LCD_OUT(1,5, "PHASE 3")    'MOSTRANDO LA FRASE "FASE 3" EN LA
LCD
     LCD_OUT(2,1, "VOLT. - CURRENT.")'MOSTRANDO LA FRASE "VOLT. -
CURRENT." EN LA LCD
     PORTE.5 = 1              'LED INDICADOR DE LA FASE 3.
     DELAY_MS(1000)           'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA

     LCD_CMD(LCD_CLEAR)       'BORANDO LA PANTALLA DE LA LCD

     LCD_OUT(1,5, "TO SAVE")   'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
     LCD_OUT(2,4, "PUSH SAVE") 'MOSTRANDO LA FRASE "PUSH SAVE" EN
LA LCD
     DELAY_MS(800)            'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
     IF (PORTB.7 = 1) THEN
       BUCLE = 3
       GOTO SELEC_OK
     END IF
   LOOP UNTIL (PORTB.8 = 1)
ELSE
   NOP
END IF

 PORTE.5 = 0              'LED INDICADOR DE LA FASE 3.

SELEC_OK:
```

```
LOOP UNTIL (PORTB.7 = 1)

'*********************INICIO DEL BUCLE DE ALMACENAMIENTO*****************
PORTE.1 = 1                 'LED INDICADOR DE INICIO DEL
ALMACENAMIENTO.
LCD_CMD(LCD_CLEAR)           'BORANDO LA PANTALLA DE LA LCD
LCD_OUT(1,1, "SAVING...")   'MOSTRANDO LA FRASE "TO SAVE" EN LA LCD

SELECT CASE CLOCK
  CASE 1    K=26            'REPRESENTA 1 MINUTO EN TIEMPO REAL
  CASE 2    K=130           'REPRESENTA 5 MINUTO EN TIEMPO REAL
  CASE 3    K=260           'REPRESENTA 10 MINUTO EN TIEMPO REAL
  CASE ELSE
    GOTO   TIME_CHANGE
END SELECT

SELECT CASE BUCLE
  CASE 1    GOTO FASE_1
  CASE 2    GOTO FASE_2
  CASE 3    GOTO FASE_3
  CASE ELSE
    GOTO   CHANGE_FASE
END SELECT

'*****************************************FASE 1*************************************
FASE_1:
INC(F)
Q=1

J    = 0             ' NUMERO DE DATOS DE LA SANAL ADQUIRIDA
I    = 0             ' NUMERO DE VECES QUE LA ONDA SUPERO AL 10% DE LA
MUESTRA

M1    = 0             ' CONTADOR BASICO DE VOLTAJE
L1    = 0             ' CONTADOR BASICO 2 DE VOLTAJE
```

```
O1    = 0              ' CONTADOR BASICO 3 DE VOLTAJE

M2    = 0              ' CONTADOR BASICO DE CORRIENTE
L2    = 0              ' CONTADOR BASICO 2 DE CORRIENTE
O2    = 0              ' CONTADOR BASICO 3 DE CORRIENTE

S_V1  = 0              'NUMERO DSE VECES QUE LA SENAL DE VOLTAJE 1 FUE
ALMACENADA
S_C1  = 0              'NUMERO DSE VECES QUE LA SENAL DE CORRIENTE 1
FUE ALMACENADA

IFS0  = IFS0 and $FFF7  ' Clear TMR1IF
IEC0  = IEC0 or  $0008  ' Enable Timer1 interrupts
T1CON = $8020           ' Timer1 ON, internal clock FCY, prescaler 1:64

WHILE (((CONT <K) AND (S_V1<10)) OR ((CONT <K) AND (S_C1<10)))

L1=1+60*M1
INC(M1)

L2=1+60*M2
INC(M2)

'****ADQUIRIENDO UNA MUESTRA DE LA SENAL DE VOLTAJE Y
CORIENTE****
V1_SAVE:
INC(S_V1)
'SINCRONIZANDO LA SENAL DE VOLTAJE
   DO
       NOP
   LOOP UNTIL (ADC_READ(0)>512)
'ALMACENANDO LA SENAL DE VOLTAJE
   O1=L1
   WHILE ((J<17) AND (CONT<K) AND (S_V1<10))
       VOLT     = ADC_READ(0)
```

```
        EEPROM_WRITE(EEADDR+VOLT1+O1, VOLT)
        WHILE(NVMCON.15 <> 0)
           NOP       ' WAIT FOR WRITE TO FINISH
        WEND
        INC(J)
        O1=O1+2
    WEND
    J=0
    IF I>7 THEN
      GOTO OUT_V1
    ELSE
       NOP
    END IF

C1_SAVE:
INC(S_C1)
'SINCRONIZANDO LA SENAL DE CORRIENTE
    DO
        NOP
    LOOP UNTIL (ADC_READ(1)>512)
'ALMACENANDO LA SENAL DE CORRIENTE
    O2=L2
    WHILE ((J<17) AND (CONT<K) AND (S_C1<10))
        CURRENT  = ADC_READ(1)
        EEPROM_WRITE(EEADDR+CURRENT1+O2, CURRENT)
        WHILE(NVMCON.15 <> 0)
           NOP       ' WAIT FOR WRITE TO FINISH
        WEND
        INC(J)
        O2=O2+2
    WEND
    J=0
    IF I>7 THEN
      GOTO OUT_C1
    ELSE
```

```
      NOP
   END IF
'****VERRIFICANDO SI LA SENAL DE VOLTAJE Y CORRINTE ADQUIRIDAS
HAN VARIADO****

'SINCRONIZANDO LA SENAL DE VOLTAJE
   DO
        NOP
   LOOP UNTIL (ADC_READ(0)>512)
'VERIFICANDO LA SENAL DE VOLTAJE
   O1=L1
   WHILE ((J<17) AND (CONT<K) AND (S_V1<10))
        VOLT              = ADC_READ(0)
        VOLT_ANTERIOR      = EEPROM_READ(eeAddr+VOLT1+O1)
        IF ((VOLT > 1.1 * VOLT_ANTERIOR) OR (VOLT * 1.1 <
VOLT_ANTERIOR))  THEN
          I=I+1
        ELSE
           I=I
        END IF
        IF ((I>7) AND (J<10)) THEN
          J=0
          L1=1+60*M1
          INC(M1)
          GOTO V1_SAVE
        ELSE
          NOP
        END IF
        INC(J)
        O1=O1+2
   WEND
   OUT_V1:
   J=0
   I=0
'SINCRONIZANDO LA SENAL DE CORRIENTE
```

```
   DO
        NOP
   LOOP UNTIL (ADC_READ(1)>512)
'VERIFICANDO LA SENAL DE CORRIENTE
   O2=L2
   WHILE ((J<17) AND (CONT<K) AND (S_C1<10))
        CURRENT              = ADC_READ(1)
        CURRENT_ANTERIOR      = EEPROM_READ(eeAddr+CURRENT1+O2)
        IF ((CURRENT > 1.1 * CURRENT_ANTERIOR) OR (CURRENT * 1.1 <
CURRENT_ANTERIOR))  THEN
          I=I+1
        ELSE
           I=I
        END IF
        IF ((I>7) AND (J<10)) THEN
          J=0
          L2=1+60*M2
          INC(M2)
          GOTO C1_SAVE
        ELSE
           NOP
        END IF
        INC(J)
        O2=O2+2
   WEND
   OUT_C1:
   J=0
   I=0
WEND
'*****************************************************************************************
GOTO UART_SEND
```

```
'****************************************FASE 2**************************************
FASE_2:
INC(F)
R=1

J     = 0              ' NUMERO DE DATOS DE LA SANAL ADQUIRIDA
I     = 0              ' NUMERO DE VECES QUE LA ONDA SUPERO AL 10% DE LA
MUESTRA

M1     = 0              ' CONTADOR BASICO DE VOLTAJE
L1     = 0              ' CONTADOR BASICO 2 DE VOLTAJE
O1     = 0              ' CONTADOR BASICO 3 DE VOLTAJE

M2     = 0              ' CONTADOR BASICO DE CORRIENTE
L2     = 0              ' CONTADOR BASICO 2 DE CORRIENTE
O2     = 0              ' CONTADOR BASICO 3 DE CORRIENTE

S_V2  = 0              'NUMERO DSE VECES QUE LA SENAL DE VOLTAJE 1 FUE
ALMACENADA
S_C2  = 0              'NUMERO DSE VECES QUE LA SENAL DE CORRIENTE 1
FUE ALMACENADA

IFS0  = IFS0 and $FFF7  ' Clear TMR1IF
IEC0  = IEC0 or  $0008  ' Enable Timer1 interrupts
T1CON = $8020           ' Timer1 ON, internal clock FCY, prescaler 1:64

WHILE (((CONT <K) AND (S_V2<10)) OR ((CONT <K) AND (S_C2<10)))

L1=1+60*M1
INC(M1)

L2=1+60*M2
INC(M2)
'****ADQUIRIENDO UNA MUESTRA DE LA SENAL DE VOLTAJE Y
CORIENTE****
```

```
V2_SAVE:
INC(S_V2)
'SINCRONIZANDO LA SENAL DE VOLTAJE
   DO
       NOP
   LOOP UNTIL (ADC_READ(2)>512)
'ALMACENANDO LA SENAL DE VOLTAJE
   O1=L1
   WHILE ((J<17) AND (CONT<K) AND (S_V2<10))
       VOLT     = ADC_READ(2)
       EEPROM_WRITE(EEADDR+VOLT2+O1, VOLT)
       WHILE(NVMCON.15 <> 0)
          NOP       ' WAIT FOR WRITE TO FINISH
       WEND
       INC(J)
       O1=O1+2
   WEND
   J=0
   IF I>7 THEN
     GOTO OUT_V2
   ELSE
      NOP
   END IF

C2_SAVE:
INC(S_C2)
'SINCRONIZANDO LA SENAL DE CORRIENTE
   DO
       NOP
   LOOP UNTIL (ADC_READ(3)>512)
'ALMACENANDO LA SENAL DE CORRIENTE
   O2=L2
   WHILE ((J<17) AND (CONT<K) AND (S_C2<10))
       CURRENT  = ADC_READ(3)
       EEPROM_WRITE(EEADDR+CURRENT2+O2, CURRENT)
```

```
        WHILE(NVMCON.15 <> 0)
           NOP       ' WAIT FOR WRITE TO FINISH
        WEND
        INC(J)
        O2=O2+2
   WEND
   J=0
   IF I>7 THEN
     GOTO OUT_C2
   ELSE
      NOP
   END IF
'****VERRIFICANDO SI LA SENAL DE VOLTAJE Y CORRINTE ADQUIRIDAS
HAN VARIADO****

'SINCRONIZANDO LA SENAL DE VOLTAJE
   DO
        NOP
   LOOP UNTIL (ADC_READ(2)>512)
'VERIFICANDO LA SENAL DE VOLTAJE
   O1=L1
   WHILE ((J<17) AND (CONT<K) AND (S_V2<10))
        VOLT              = ADC_READ(2)
        VOLT_ANTERIOR2     = EEPROM_READ(eeAddr+VOLT2+O1)
        IF ((VOLT > 1.1 * VOLT_ANTERIOR2) OR (VOLT * 1.1 <
VOLT_ANTERIOR2))  THEN
          I=I+1
        ELSE
           I=I
        END IF
        IF ((I>7) AND (J<10)) THEN
          J=0
          L1=1+60*M1
          INC(M1)
          GOTO V2_SAVE
```

```
        ELSE
          NOP
        END IF
        INC(J)
        O1=O1+2
    WEND
    OUT_V2:
    J=0
    I=0
'SINCRONIZANDO LA SENAL DE CORRIENTE
    DO
        NOP
    LOOP UNTIL (ADC_READ(3)>512)
'VERIFICANDO LA SENAL DE CORRIENTE
    O2=L2
    WHILE ((J<17) AND (CONT<K) AND (S_C2<10))
        CURRENT              = ADC_READ(3)
        CURRENT_ANTERIOR2     = EEPROM_READ(eeAddr+CURRENT2+O2)
        IF ((CURRENT > 1.1 * CURRENT_ANTERIOR2) OR (CURRENT * 1.1 <
CURRENT_ANTERIOR2))  THEN
          I=I+1
        ELSE
           I=I
        END IF
        IF ((I>7) AND (J<10)) THEN
          J=0
          L2=1+60*M2
          INC(M2)
          GOTO C2_SAVE
        ELSE
           NOP
        END IF
        INC(J)
        O2=O2+2
    WEND
```

```
   OUT_C2:
   J=0
   I=0
WEND
'*******************************************************'**********************************
GOTO UART_SEND
'***************************************FASE 3****************************************
FASE_3:
INC(F)
S=1

J     = 0              ' NUMERO DE DATOS DE LA SANAL ADQUIRIDA
I     = 0              ' NUMERO DE VECES QUE LA ONDA SUPERO AL 10% DE LA
MUESTRA

M1    = 0              ' CONTADOR BASICO DE VOLTAJE
L1    = 0              ' CONTADOR BASICO 2 DE VOLTAJE
O1    = 0              ' CONTADOR BASICO 3 DE VOLTAJE

M2    = 0              ' CONTADOR BASICO DE CORRIENTE
L2    = 0              ' CONTADOR BASICO 2 DE CORRIENTE
O2    = 0              ' CONTADOR BASICO 3 DE CORRIENTE

S_V3  = 0              'NUMERO DSE VECES QUE LA SENAL DE VOLTAJE 1 FUE
ALMACENADA
S_C3  = 0              'NUMERO DSE VECES QUE LA SENAL DE CORRIENTE 1
FUE ALMACENADA

IFS0  = IFS0 and $FFF7  ' Clear TMR1IF
IEC0  = IEC0 or  $0008  ' Enable Timer1 interrupts
T1CON = $8020           ' Timer1 ON, internal clock FCY, prescaler 1:64

WHILE (((CONT <K) AND (S_V3<10)) OR ((CONT <K) AND (S_C3<10)))

L1=1+60*M1
```

```
INC(M1)

L2=1+60*M2
INC(M2)
'****ADQUIRIENDO UNA MUESTRA DE LA SENAL DE VOLTAJE Y
CORIENTE****
V3_SAVE:
INC(S_V3)
'SINCRONIZANDO LA SENAL DE VOLTAJE
   DO
        NOP
   LOOP UNTIL ((ADC_READ(4)>512))
'ALMACENANDO LA SENAL DE VOLTAJE
   O1=L1
   WHILE ((J<17) AND (CONT<K) AND (S_V3<10))
        VOLT     = ADC_READ(4)
        EEPROM_WRITE(EEADDR+VOLT3+O1, VOLT)
        WHILE(NVMCON.15 <> 0)
           NOP       ' WAIT FOR WRITE TO FINISH
        WEND
        INC(J)
        O1=O1+2
   WEND
   J=0
   IF I>7 THEN
     GOTO OUT_V3
   ELSE
      NOP
   END IF

C3_SAVE:
INC(S_C3)
'SINCRONIZANDO LA SENAL DE CORRIENTE
   DO
        NOP
```

```
   LOOP UNTIL ((ADC_READ(5)>512))
'ALMACENANDO LA SENAL DE CORRIENTE
   O2=L2
   WHILE ((J<17) AND (CONT<K) AND (S_C3<10))
        CURRENT  = ADC_READ(5)
        EEPROM_WRITE(EEADDR+CURRENT3+O2, CURRENT)
        WHILE(NVMCON.15 <> 0)
           NOP       ' WAIT FOR WRITE TO FINISH
        WEND
        INC(J)
        O2=O2+2
   WEND
   J=0
   IF I>7 THEN
     GOTO OUT_C3
   ELSE
      NOP
   END IF
'****VERRIFICANDO SI LA SENAL DE VOLTAJE Y CORRINTE ADQUIRIDAS
HAN VARIADO****

'SINCRONIZANDO LA SENAL DE VOLTAJE
   DO
        NOP
   LOOP UNTIL ((ADC_READ(4)>512))
'VERIFICANDO LA SENAL DE VOLTAJE
   O1=L1
   WHILE ((J<17) AND (CONT<K) AND (S_V3<10))
        VOLT              = ADC_READ(4)
        VOLT_ANTERIOR3     = EEPROM_READ(eeAddr+VOLT3+O1)
        IF ((VOLT > 1.1 * VOLT_ANTERIOR3) OR (VOLT * 1.1 <
VOLT_ANTERIOR3))  THEN
          I=I+1
        ELSE
           I=I
```

```
        END IF
        IF ((I>7) AND (J<10)) THEN
          J=0
          L1=1+60*M1
          INC(M1)
          GOTO V3_SAVE
        ELSE
          NOP
        END IF
        INC(J)
        O1=O1+2
    WEND
    OUT_V3:
    J=0
    I=0
'SINCRONIZANDO LA SENAL DE CORRIENTE
    DO
        NOP
    LOOP UNTIL ((ADC_READ(5)>512))
'VERIFICANDO LA SENAL DE CORRIENTE
    O2=L2
    WHILE ((J<17) AND (CONT<K) AND (S_C3<10))
        CURRENT              = ADC_READ(5)
        CURRENT_ANTERIOR3    = EEPROM_READ(eeAddr+CURRENT3+O2)
        IF ((CURRENT > 1.1 * CURRENT_ANTERIOR3) OR (CURRENT * 1.1 <
CURRENT_ANTERIOR3))  THEN
          I=I+1
        ELSE
           I=I
        END IF
        IF ((I>7) AND (J<10)) THEN
          J=0
          L2=1+60*M2
          INC(M2)
          GOTO C3_SAVE
```

```
        ELSE
           NOP
        END IF
        INC(J)
        O2=O2+2
    WEND
    OUT_C3:
    J=0
    I=0
WEND
'********************************************************************************************
GOTO UART_SEND
'***********************************ENVIO POR UART***********************************
UART_SEND:
IEC0.3  = 0
PORTE.1 = 0             'LED INDICADOR DE INICIO DEL ALMACENAMIENTO.
LCD_CMD(LCD_CLEAR)      'BORANDO LA PANTALLA DE LA LCD
LCD_OUT(1,4, "SAVE READY")'MOSTRANDO LA FRASE "TO SAVE" EN LA
LCD
DELAY_MS(500)           'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
IF F>=3 THEN
  LCD_CMD(LCD_CLEAR)   'BORANDO LA PANTALLA DE LA LCD
  LCD_OUT(1,1, "ALL PHASE READY")'MOSTRANDO LA FRASE "ALL PHASE
READY" EN LA LCD
  PORTE.3 = 0             'LED INDICADOR DE LA FASE 1.
  PORTE.4 = 0             'LED INDICADOR DE LA FASE 2.
  PORTE.5 = 0             'LED INDICADOR DE LA FASE 3.
  DELAY_MS(1500)        'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
  GOTO UARTZ
ELSE
  PORTE.3 = 0             'LED INDICADOR DE LA FASE 1.
  PORTE.4 = 0             'LED INDICADOR DE LA FASE 2.
  PORTE.5 = 0             'LED INDICADOR DE LA FASE 3.
```

```
  GOTO CHANGE_FASE
END IF
UARTZ:
DO
LCD_CMD(LCD_CLEAR)      'BORANDO LA PANTALLA DE LA LCD
LCD_OUT(1,3, "TO TRANSMIT")'MOSTRANDO LA FRASE "TO TRANSMIT" EN
LA LCD
LCD_OUT(2,4, "PUSH SEND")'MOSTRANDO LA FRASE "PUSH SEND" EN LA
LCD
DELAY_MS(800)          'TIEMPO DE RETARDO PARA VISUALIZIZACION
PREVIA
LOOP UNTIL PORTB.6=1
PORTE.2 = 1            'LED INDICADOR DE INICIO DE LA TRANSMICION.
Lcd_Cmd(LCD_CLEAR)
Lcd_Out(1,1, "SENDING...")
DELAY_MS(500)
'*****************************************************************************************
'*********************ENVIANDO LOS DATOS ALMACENADOS*********************
J =0
O1=0
   WHILE (J<170)
'         Uart_Write_Char($41)                ' ENVÍA EL CARACTER "A" COMO
CARACTER SEPARADOR DE DATOS
         DATO     = EEPROM_READ(EEADDR+VOLT1+O1)' DATO : LECTURA
DE LOS DATOS EN LA EEPROM

         ENTERO   = DATO DIV 1000             ' ENTERO : DIVISIÓN DE DATO
PARA 1000
         Uart_Write_Char(ASCII(ENTERO))       ' ENVÍA EL DATO DEL
CARACTER ENTERO
         DECIMAL1 = ((DATO DIV 10) DIV 10) MOD 10' DECIMAL1 : OBTENCIÓN
DEL 1ER DECIMAL
         Uart_Write_Char(ASCII(DECIMAL1))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL1
```

```
        DECIMAL2 = (DATO DIV 10) MOD 10       ' DECIMAL2 : OBTENCIÓN DEL
2DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL2))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        DECIMAL3 = DATO MOD 10               ' DECIMAL2 : OBTENCIÓN DEL
3DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL3))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2

        Uart_Write_Char($D)                  ' ENVÍA UN ENTER

        INC(J)
        O1=O1+2
   WEND

Uart_Write_Char($D)                          ' ENVÍA UN ENTER
Uart_Write_Char($42)                         ' ENVÍA EL CARACTER "B" COMO
CARACTER SEPARADOR DE GRAFICAS
Uart_Write_Char($D)                          ' ENVÍA UN ENTER

J =0
O1=0
   WHILE (J<170)
'       Uart_Write_Char($41)                 ' ENVÍA EL CARACTER "A" COMO
CARACTER SEPARADOR DE DATOS
        DATO     = EEPROM_READ(EEADDR+CURRENT1+O1)' DATO :
LECTURA DE LOS DATOS EN LA EEPROM

        ENTERO   = DATO DIV 1000             ' ENTERO : DIVISIÓN DE DATO
PARA 1000
        Uart_Write_Char(ASCII(ENTERO))        ' ENVÍA EL DATO DEL
CARACTER ENTERO
        DECIMAL1 = ((DATO DIV 10) DIV 10) MOD 10    ' DECIMAL1 :
OBTENCIÓN DEL 1ER DECIMAL
```

```
        Uart_Write_Char(ASCII(DECIMAL1))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL1
        DECIMAL2 = (DATO DIV 10) MOD 10       ' DECIMAL2 : OBTENCIÓN DEL
2DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL2))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        DECIMAL3 = DATO MOD 10                ' DECIMAL2 : OBTENCIÓN DEL
3DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL3))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2)

        Uart_Write_Char($D)                   ' ENVÍA UN ENTER
        INC(J)
        O1=O1+2
   WEND

Uart_Write_Char($D)                           ' ENVÍA UN ENTER
Uart_Write_Char($43)                          ' ENVÍA EL CARACTER "C" COMO
CARACTER SEPARADOR DE GRAFICAS
Uart_Write_Char($D)                           ' ENVÍA UN ENTER

J =0
O1=0
   WHILE (J<170)
'        Uart_Write_Char($41)      ' ENVÍA EL CARACTER "A" COMO CARACTER
SEPARADOR DE DATOS
        DATO     = EEPROM_READ(EEADDR+VOLT2+O1)       ' DATO :
LECTURA DE LOS DATOS EN LA EEPROM

        ENTERO   = DATO DIV 1000              ' ENTERO : DIVISIÓN DE DATO
PARA 1000
        Uart_Write_Char(ASCII(ENTERO))        ' ENVÍA EL DATO DEL
CARACTER ENTERO
        DECIMAL1 = ((DATO DIV 10) DIV 10) MOD 10    ' DECIMAL1 :
OBTENCIÓN DEL 1ER DECIMAL
```

```
        Uart_Write_Char(ASCII(DECIMAL1))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL1
        DECIMAL2 = (DATO DIV 10) MOD 10       ' DECIMAL2 : OBTENCIÓN DEL
2DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL2))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        DECIMAL3 = DATO MOD 10               ' DECIMAL2 : OBTENCIÓN DEL
3DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL3))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2

        Uart_Write_Char($D)                  ' ENVÍA UN ENTER
        INC(J)
        O1=O1+2
   WEND

Uart_Write_Char($D)                  ' ENVÍA UN ENTER
Uart_Write_Char($44)      ' ENVÍA EL CARACTER "D" COMO CARACTER
SEPARADOR DE GRAFICAS
Uart_Write_Char($D)                  ' ENVÍA UN ENTER

J =0
O1=0
   WHILE (J<170)
'        Uart_Write_Char($41)      ' ENVÍA EL CARACTER "A" COMO CARACTER
SEPARADOR DE DATOS
        DATO     = EEPROM_READ(EEADDR+CURRENT2+O1)       ' DATO :
LECTURA DE LOS DATOS EN LA EEPROM

        ENTERO   = DATO DIV 1000             ' ENTERO : DIVISIÓN DE DATO
PARA 1000
        Uart_Write_Char(ASCII(ENTERO))       ' ENVÍA EL DATO DEL
CARACTER ENTERO
        DECIMAL1 = ((DATO DIV 10) DIV 10) MOD 10    ' DECIMAL1 :
OBTENCIÓN DEL 1ER DECIMAL
```

```
        Uart_Write_Char(ASCII(DECIMAL1))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL1
        DECIMAL2 = (DATO DIV 10) MOD 10    ' DECIMAL2 : OBTENCIÓN DEL
2DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL2))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        DECIMAL3 = DATO MOD 10                 ' DECIMAL2 : OBTENCIÓN DEL
3DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL3))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2

        Uart_Write_Char($D)                          ' ENVÍA UN ENTER
        INC(J)
        O1=O1+2
   WEND

Uart_Write_Char($D)                                  ' ENVÍA UN ENTER
Uart_Write_Char($45)   ' ENVÍA EL CARACTER "E" COMO CARACTER
SEPARADOR DE GRAFICAS
Uart_Write_Char($D)                  ' ENVÍA UN ENTER

J =0
O1=0
   WHILE (J<170)
'        Uart_Write_Char($41)      ' ENVÍA EL CARACTER "A" COMO CARACTER
SEPARADOR DE DATOS
        DATO     = EEPROM_READ(EEADDR+VOLT3+O1)        ' DATO :
LECTURA DE LOS DATOS EN LA EEPROM

        ENTERO   = DATO DIV 1000             ' ENTERO : DIVISIÓN DE DATO
PARA 1000
        Uart_Write_Char(ASCII(ENTERO))        ' ENVÍA EL DATO DEL
CARACTER ENTERO
        DECIMAL1 = ((DATO DIV 10) DIV 10) MOD 10    ' DECIMAL1 :
OBTENCIÓN DEL 1ER DECIMAL
```

```
        Uart_Write_Char(ASCII(DECIMAL1))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL1
        DECIMAL2 = (DATO DIV 10) MOD 10       ' DECIMAL2 : OBTENCIÓN DEL
2DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL2))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        DECIMAL3 = DATO MOD 10                ' DECIMAL2 : OBTENCIÓN DEL
3DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL3))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2

        Uart_Write_Char($D)                   ' ENVÍA UN ENTER
        INC(J)
        O1=O1+2
   WEND

Uart_Write_Char($D)                   ' ENVÍA UN ENTER
Uart_Write_Char($46)      ' ENVÍA EL CARACTER "F" COMO CARACTER
SEPARADOR DE GRAFICAS
Uart_Write_Char($D)                   ' ENVÍA UN ENTER

J =0
O1=0
   WHILE (J<170)
'        Uart_Write_Char($41)      ' ENVÍA EL CARACTER "A" COMO CARACTER
SEPARADOR DE DATOS
        DATO     = EEPROM_READ(EEADDR+CURRENT3+O1)        ' DATO :
LECTURA DE LOS DATOS EN LA EEPROM

        ENTERO   = DATO DIV 1000              ' ENTERO : DIVISIÓN DE DATO
PARA 1000
        Uart_Write_Char(ASCII(ENTERO))        ' ENVÍA EL DATO DEL
CARACTER ENTERO
        DECIMAL1 = ((DATO DIV 10) DIV 10) MOD 10    ' DECIMAL1 :
OBTENCIÓN DEL 1ER DECIMAL
```

```
        Uart_Write_Char(ASCII(DECIMAL1))      ' CARACTER DECIMAL1
        DECIMAL2 = (DATO DIV 10) MOD 10       ' DECIMAL2 : OBTENCIÓN DEL
2DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL2))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        DECIMAL3 = DATO MOD 10                ' DECIMAL2 : OBTENCIÓN DEL
3DO DECIMAL
        Uart_Write_Char(ASCII(DECIMAL3))      ' ENVÍA EL DATO DEL
CARACTER DECIMAL2
        Uart_Write_Char($D)                   ' ENVÍA UN ENTER
        INC(J)
        O1=O1+2
   WEND
Uart_Write_Char($D)                   ' ENVÍA UN ENTER
Uart_Write_Char($47)      ' ENVÍA EL CARACTER "F" COMO CARACTER
SEPARADOR DE GRAFICAS
Uart_Write_Char($D)                   ' ENVÍA UN ENTER
J =0
O1=0
'*******************************************************************************************
delay_ms(1000)
PORTE.1 = 0
Lcd_Cmd(LCD_CLEAR)
Lcd_Out(1,1, "SEND READY")
DELAY_MS(500)
end.
```

# PROGRAMACION EN VISUAL BASIC

```
Dim ApExcel As Object
Dim objLibro As Object
Dim S As Integer
Dim ka As Integer
Dim arch
Dim aux(10) As Integer
Dim gra1(10) As Integer

Private Sub Form_Load()
 MSComm1.CommPort = 3
 MSComm1.Settings = "9600,n,8,1"
 MSComm1.RThreshold = 1
 MSComm1.PortOpen = True
 Randomize
End Sub
Private Sub k_Click()
 Me.gr1.MaxCols = 10
 Me.gr1.MaxRows = 0
 aux = 0
 S = 0
 Me.txt_in.Text = ""
End Sub

Private Sub k2_Click()
If aux(0) <> 0 Then
   Call Grabar
   Call gra
 Else
    MsgBox "INGRESE DATOS ", vbInformation, "INFORMACION"
 End If
End Sub
```

```
Private Sub k3_Click()
 Call llenargrid
End Sub

Private Sub k6_Click()
FRM_ACER.Show
End Sub

Private Sub k7_Click()
End
End Sub

Sub Grabar()
Dim J, c As Integer

'tomamos un comondialog y como filtro le decimo
'tomamos un comondialog y como filtro le decimos que es archivo de excel
CommonDialog1.Filter = "Libro de Microsoft Excel|*.xls"
CommonDialog1.ShowSave
If Err Then
' El usuario eligió Cancelar.
If Err = 32755 Then
Grabe.Value = 0
Grabar2.Caption = "Grabar"
Exit Sub
End If
End If
arch = CommonDialog1.FileName
'creamos el objeto de excel
Set ApExcel = CreateObject("Excel.application")
'añadimos un hoja nueva
ApExcel.Workbooks.Add
'*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-**
'enviamos datos a excel en las celdas qeu se indican
ApExcel.CELLS(1, 5).Formula = "DATALOGGER"
```

```
'se define la fuente para esta celda
ApExcel.CELLS(1, 1).Font.Size = 18
'se toma la fecha
ApExcel.CELLS(2, 1).Formula = "Fecha:  " & Date
ApExcel.CELLS(2, 1).Font.Size = 12
J = 1
Do While J <= 10
  ApExcel.CELLS(3, J).Formula = "DATOS"
  J = J + 1
Loop
'se toma la hora
S = 1: c = 0
Do While S <= gr1.MaxRows
  gr1.Row = S
  J = 1
  Do While J <= 10
    gr1.Col = J
    ApExcel.CELLS(c + Val(J), 25).Formula = gr1.Text
    ApExcel.CELLS(3 + Val(S), J).Formula = gr1.Text
    J = J + 1
  Loop
  c = c + (J - 1)
  S = S + 1
Loop
End Sub
Sub gra()
Dim t, t1, i, n As Integer
t1 = 1: i = 0
'///////////////////'

  'tomamos la hora 1 del libro como objeto
  Set objLibro = ApExcel.Worksheets(1)
  Do While Val(gra1(i)) <= ((S - 1) * 10)
    'seleccionamos otra vez la hoja uno para poder volver a tomar datos
    ApExcel.Sheets("Hoja1").Select
```

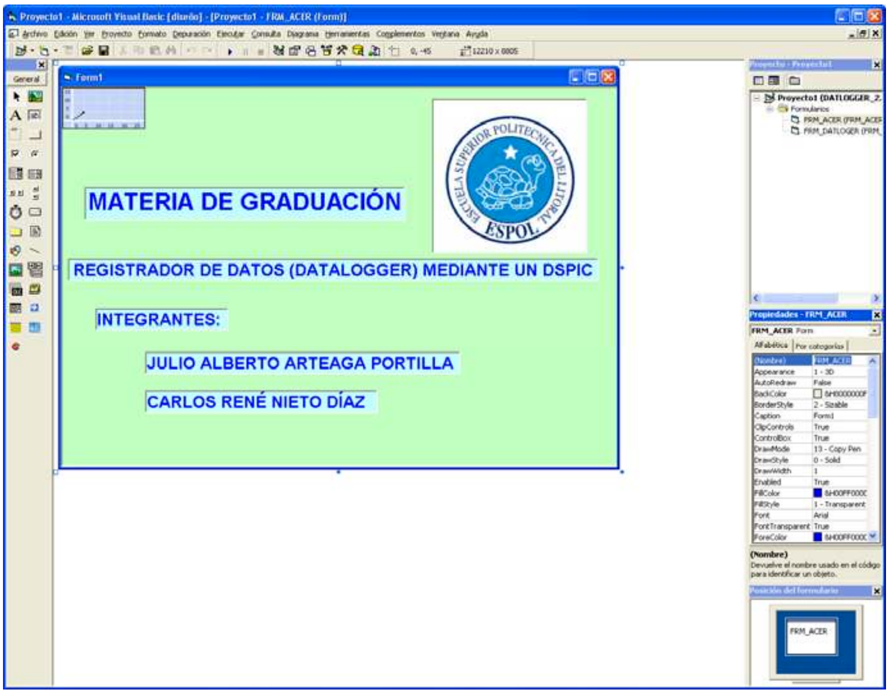
```
    'definimos el rango de celdas para graficar
    rango = "y" & Val(t1) & ":" & "y" & Val(gra1(i))
    'seleccionamos el rango adecuado
    'objLibro.Range(rango).Select
    ApExcel.Worksheets(1).Range(rango).Select
    'añadimos una grafica
    ApExcel.Charts.Add
    'le decimos qeu tipo de grafica es(el numero define cual es buscar hasta
encontrar la que necesita)
    ApExcel.ActiveChart.chartType = 73
    'donde va a estar ubicada
    '1 es para que sea en hoja nueva, 2 para la misma y que sea un objeto,
    'lo que está entre comillas es para el nombre

    If ((i + 1) Mod 7) = 0 Then
      ApExcel.ActiveChart.Location 1, "POTENCIA" & " " & "FASE " & Val(i / 6)
      ElseIf ((i + 1) Mod 7) <> 0 Then
    '/*/*/*/*/*/*/*/*/*/*/*/*/*/*
    If ((i + 1) Mod 2) = 0 Then
      ApExcel.ActiveChart.Location 1, "CORRIENTE" & " " & "FASE " & Val((i +
1) - ((i + 1) / 2))
      Else
      ApExcel.ActiveChart.Location 1, "VOLTAJE" & " " & "FASE " & Val((i + 1) -
(i / 2))
    End If
    End If
    If ((i + 1) Mod 8) = 0 Then
      ApExcel.ActiveChart.Location 1, "POTENCIA" & " " & "FASE " & Val((i * 2) /
i)
      ElseIf ((i + 1) Mod 8) <> 0 Then
     If ((i + 1) Mod 9) = 0 Then
      ApExcel.ActiveChart.Location 1, "POTENCIA" & " " & "FASE " & Val((i * 3) /
i)
     End If
     End If
```

```
'le decimos que no tiene leyenda
ApExcel.ActiveChart.HasLegend = False
'que tiene titulo
ApExcel.ActiveChart.HasTitle = True
'que no tiene lineas de division
ApExcel.ActiveChart.Axes(2).HasMajorGridlines = False
'el color del area de trazado
ApExcel.ActiveChart.PlotArea.Interior.Color = RGB(255, 255, 255)
'el titulo de la grafica
'/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/
If ((i + 1) Mod 7) = 0 Then
  ApExcel.ActiveChart.ChartTitle.Text = "POTENCIA" & " " & "FASE " & Val(i /
6) & " " & "vs TIEMPO"
ElseIf ((i + 1) Mod 7) <> 0 Then
If ((i + 1) Mod 8) = 0 Then
  ApExcel.ActiveChart.ChartTitle.Text = "POTENCIA" & " " & "FASE " & Val((i
* 2) / i) & " " & "vs TIEMPO"
ElseIf ((i + 1) Mod 8) <> 0 Then
 If ((i + 1) Mod 9) = 0 Then
  ApExcel.ActiveChart.ChartTitle.Text = "POTENCIA" & " " & "FASE " & Val((i
* 3) / i) & " " & "vs TIEMPO"
ElseIf ((i + 1) Mod 9) <> 0 Then

If ((i + 1) Mod 2) = 0 Then
   ApExcel.ActiveChart.ChartTitle.Text = "CORRIENTE " & "FASE " & Val((i +
1) - ((i + 1) / 2)) & " " & "vs TIEMPO"
Else
   ApExcel.ActiveChart.ChartTitle.Text = "VOLTAJE " & "FASE " & Val((i + 1)
- (i / 2)) & " " & "vs TIEMPO"
End If
End If
End If
End If

t1 = t1 + Val(aux(i)): i = i + 1
```

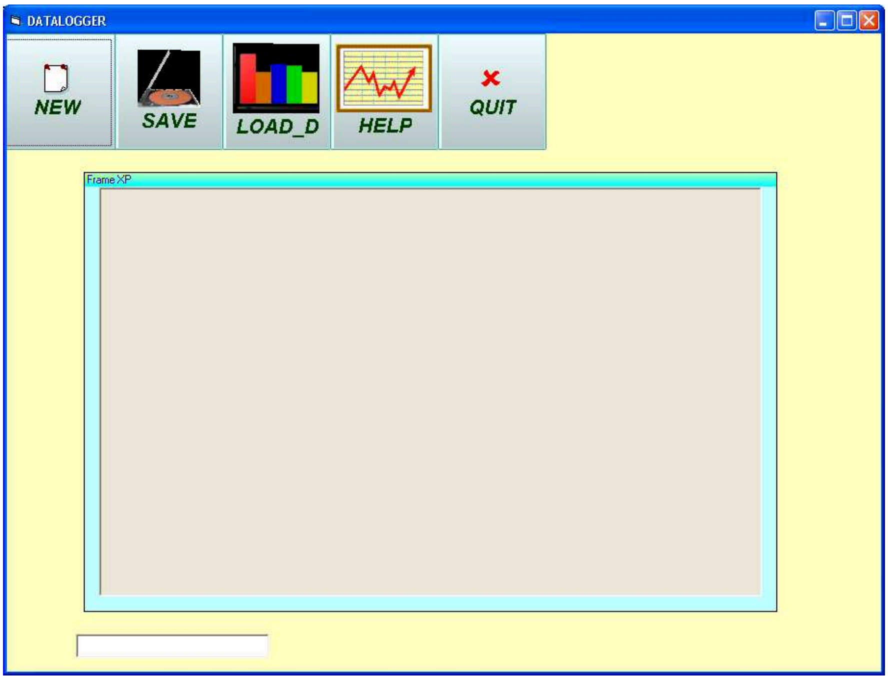
```
    If i = 9 Then gra1(i) = (S * 10) + 1

  Loop
    'seleccionamos otra vez la hoja uno para poder volver a tomar datos
    ApExcel.Sheets("Hoja1").Select
    rango = "a1:j" & Val(S + 2)
    ApExcel.Worksheets(1).Range(rango).Select

    'cambiamos el nombre de la hoja 1
    ApExcel.Worksheets(1).Name = "Datos"
    'borramos la hoja 2 y 3
    ApExcel.Worksheets("Hoja2").Delete
    ApExcel.Worksheets("Hoja3").Delete
    'guardamos cerramos y quitamos excel
    ApExcel.Worksheets(1).SaveAs arch
    ApExcel.Workbooks.Close
    ApExcel.Quit
    Set ApExcel = Nothing

S = 0
MsgBox "SUS DATOS FUERON EXPORTADOS A EXCEL ", vbInformation,
"INFORMACION"
End Sub

Public Sub llenargrid()
Dim c, J, ban, y, b As Integer
Dim valor1, con As String
' variables a cero
 c = 1: i = 0: X = 1: y = 0: ban = 0: b = 0
'*************** ojo********
'usando el puerto serial
 Do While y <> 10
  J = 1
  Me.gr1.MaxRows = Me.gr1.MaxRows + 1
  Me.gr1.Row = Me.gr1.MaxRows
```

```
Do While J <= 10
        Me.gr1.Col = J
        valor1 = Mid(Me.txt_in.Text, X, 1)

        Select Case valor1
              Case "A"
                   ban = ban + 1
              Case "B"
                   gra1(0) = c
                   aux(0) = c: c = 0: ban = ban + 1
              Case "C"
                   gra1(1) = c + gra1(0)
                   aux(1) = c: c = 0: ban = ban + 1
              Case "D"
                   gra1(2) = c + gra1(1)
                   aux(2) = c: c = 0: ban = ban + 1
              Case "E"
                   gra1(3) = c + gra1(2)
                   aux(3) = c: c = 0: ban = ban + 1
              Case "F"
                   gra1(4) = c + gra1(3)
                   aux(4) = c: c = 0: ban = ban + 1
              Case "G"
                   gra1(5) = c + gra1(4)
                   aux(5) = c: c = 0: ban = ban + 1
              Case "H"
                   gra1(6) = c + gra1(5)
                   aux(6) = c: c = 0: ban = ban + 1
              Case "I"
                   gra1(7) = c + gra1(6)
                   aux(7) = c: c = 0: ban = ban + 1
              Case "J"
                   gra1(8) = c + gra1(7)
                   aux(8) = c: ban = ban + 1
```

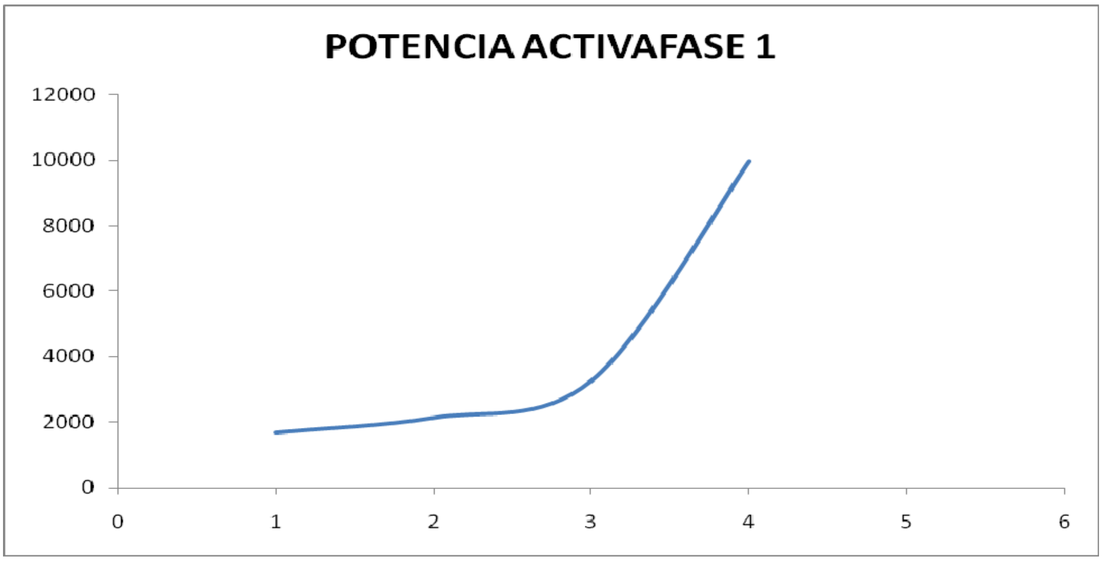
```
                ' otra letra
                 y = 10: J = 11

            Case "0" To "9"
                con = con & valor1
            Case Else
                b = 1
        End Select
        X = X + 1

        If ban = 2 And b <> 1 Then

          Me.gr1.Text = con
          J = J + 1: c = c + 1
          ban = 1: con = ""
        ElseIf b = 1 And ban = 2 Then
           ban = 1: con = "": b = 0:
        End If

  Loop

Loop

End Sub

Private Sub MSComm1_OnComm()

     Me.txt_in.Text = Me.txt_in.Text + MSComm1.Input

End Sub
```



MATERIA DE GRADUACIÓN
REGISTRADOR DE DATOS (DATALOGGER) MEDIANTE UN DSPIC
INTEGRANTES:
JULIO ALBERTO ARTEAGA PORTILLA
CARLOS RENÉ NIETO DÍAZ
ESCUELA SUPERIOR POLITECNICA DEL LITORAL
ESPOL

DATALOGGER
NEW
SAVE
LOAD_D
HELP
QUIT
Frame XP


POTENCIA ACTIVAFASE 1
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
6

# BIBLIOGRAFIA


- Microchip, Hoja de datos del dsPIC30F4011 http://ww1.microchip.com/downloads/en/DeviceDoc/70135F.pdf.

- Microchip, Manual de referencia de la familia dsPIC30F http://ww1.microchip.com/downloads/en/DeviceDoc/70046E.pdf.

- Microchip, Manual de usuario del Microchip PICkit 2 http://www.microchip.com/.

- Datasheetcatalog, Hoja de datos del LM324 http://www.datasheetcatalog.net/es/datasheets_pdf/L/M/3/2/LM324.shtml

- Monografias.com, Documento sobre componentes electrónicos http://www.monografias.com/trabajos16/componentes-electronicos/componentes-electronicos.shtml

- Ing. Ramos Ramos Guillermo, Ing. Hernández Jorge Eduardo, Castaño Juan Andrés. Curso práctico de Electrónica Industrial y Automatización: Proyectos, Tomo 3. 193-196, 202-208.

- Angulo Usategui José María, Begoña García Zapirain, Sáez Javie Vicente y Angulo Martínez Ignacio. Micro controladores Avanzados dsPIC: Diseño práctico de aplicaciones, 1era edición. Ed. McGraw-Hill. 54-55, 308-310, 339-342, 364-368, 444-448.