

Diseño e Implementación de Software y Hardware de un Registrador de Variables Eléctricas con Comunicaciones Ethernet Basado en Tecnología Arduino y Sistema de Supervisión HMI

Gustavo J. López F., David R. Rivas L., Milton F. Pérez G.

Departamento de Eléctrica y Electrónica, Escuela Politécnica del Ejército Sede Latacunga,

Latacunga, Ecuador

gustavofriasl@hotmail.com

drrivas@espe.edu.ec

mfperez3@espe.edu.ec

ABSTRACT: This project is about the design of software and hardware of an electrical variables logger which allows to obtain precise information from electrical power consumption of a home. Also transmits data stored through Ethernet communication. The boards used for the development logger are ARDUINO technology which are based on the philosophy of open hardware and software, the logger hardware was made with the following boards: Arduino UNO SMD, Arduino Nano v3.0 and Arduino Ethernet Shield.

RESUMEN: Este proyecto se trata del diseño de software y hardware de un registrador de variables eléctricas el cual permite obtener información precisa del consumo de energía eléctrica de una vivienda. Además transmite los datos almacenados mediante comunicación Ethernet. Las tarjetas utilizadas para el desarrollo del registrador son de tecnología ARDUINO las cuales se basan en la filosofía de software y hardware abierto; el hardware del registrador se lo realizó con las siguientes tarjetas: Arduino UNO SMD, Arduino Nano v3.0 y Arduino Ethernet Shield.

PALABRAS CLAVES: registrador, sensores, voltaje, corriente, Ethernet, kilovatios, Arduino, variables eléctricas.

I. INTRODUCCIÓN

La implementación del registrador de variables eléctricas con transmisión de datos a través del protocolo Ethernet permite realizar un monitoreo de dichas variables en cualquier lugar con tan solo acceder a la IP asignada a la tarjeta Arduino.

El costo de los registradores de variables eléctricas es alto por el uso de software y hardware propietario, es por ese motivo que se desarrolló un registrador de bajo costo con características de software y hardware de arquitectura abierta.

En Ecuador el desarrollo de tecnología es escaso, por lo que es conveniente fomentar la investigación en nuevos dispositivos para realizar aplicaciones de uso común que permitan cubrir una necesidad en la sociedad.

El hardware del registrador se lo realizó con las siguientes tarjetas: Arduino Uno SMD, Arduino Nano v3.0 y Arduino Ethernet Shield.

II. DESARROLLO

2.1. Arduino UNO SMD

Arduino UNO (figura 1) es una placa electrónica basada en el microcontrolador ATmega 328, la misma que cuenta con las características descritas en la tabla 1.

La placa electrónica tiene todo lo necesario para proporcionar soporte al microcontrolador, basta con conectar la placa a una computadora con un cable USB o a un adaptador AC-DC o a una batería para empezar.

Esta tarjeta difiere de todas las placas anteriores ya que no utiliza el chip FTDI USB serie. Por el contrario cuenta con el ATmega16U2 programado como un convertidor de USB a serie.



Fig. 1 Arduino UNO SMD.

TABLA I
ESPECIFICACIONES DE LA TARJETA ARDUINO UNO SMD

Microcontroller	ATmega328
Operating Voltage	5V

Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 Hz

2.2. Arduino Nano

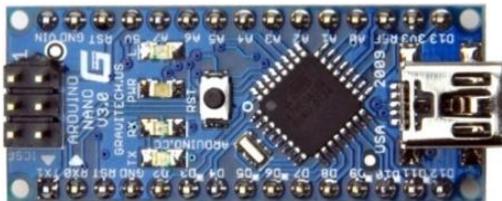


Fig. 2 Arduino Nano.

La tarjeta Arduino Nano es una placa pequeña, completa basada en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x). Tiene más o menos la misma funcionalidad de la tarjeta Arduino Duemilanove, pero en un paquete diferente, como se puede observar en la figura 2. Esta tarjeta funciona con un cable USB tipo mini B en lugar de uno estándar y sus características se encuentran en la tabla 2.

TABLA II
ESPECIFICACIONES DE LA TARJETA ARDUINO NANO

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 Hz

2.3. Arduino Ethernet Shield

El Ethernet Shield (figura 3) tiene un estándar de conexión RJ-45, con un transformador de línea integrado.

Posee una ranura para tarjetas micro SD, que puede ser usada para almacenar archivos para proporcionarlos a través de la red. Es compatible con las tarjetas Arduino Uno y Mega (usando la librería Ethernet). Al lector de tarjetas microSD a bordo se puede acceder a través de la librería SdFat. Las características de la tarjeta se encuentran en la tabla 3.

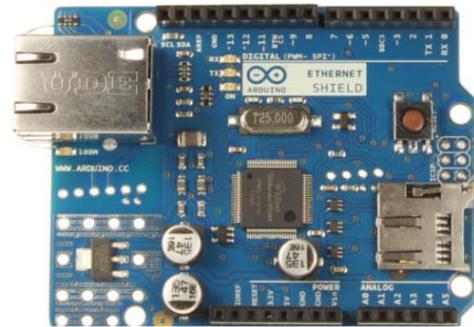


Fig. 3 Arduino Ethernet Shield.

TABLA III
ESPECIFICACIONES DE LA TARJETA ETHERNET SHIELD

Microcontrolador	Requiere una tarjeta Arduino UNO o MEGA
Voltaje de entrada	5 V (suministrados desde la placa Arduino)
Controlador Ethernet	W5100 con buffer interno de 16K
Velocidad de conexión	10/100 Mb
Conexión con Arduino	Mediante el puerto SPI

2.4. Bus I2C

Fue diseñado por Philips al inicio de la década de 1980. Su nombre viene de Inter Integrated Circuit. Este bus tiene una velocidad de transmisión de 100Kbits por segundo en el modo estándar, también permite velocidades de 3.4 Mbit/s. La principal característica de I2C es que utiliza dos líneas para transmitir la información: una para los datos y por otra la señal de reloj como se puede observar en la figura 4.

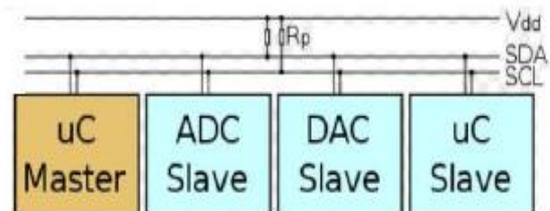


Fig. 4 Bus I2C.

SDA: Es la línea de datos serie (Serial Data, en inglés), semidireccional. Eléctricamente se trata de una

señal a colector o drenado abierto. Es gobernada por el emisor, sea este un maestro o un esclavo.

SCL: Es la señal de sincronía (reloj serie, o Serial Clock en inglés). Eléctricamente se trata de una señal a colector o drenado abierto. En un esclavo se trata de una entrada, mientras que en un maestro es una salida. Esta señal es gobernada única y exclusivamente por el maestro; un esclavo sólo puede retenerla para forzar al maestro a ralentizar su funcionamiento.

2.5. Bus SPI

Es un estándar establecido por Motorola que utiliza un bus de 4 líneas para interconectar dispositivos periféricos de baja y media velocidad como se indica en la figura 5. La comunicación se realiza siguiendo un modelo maestro/esclavo donde el maestro selecciona al esclavo y comienza el proceso de transmisión/recepción de información. SPI constituye un bus full duplex.

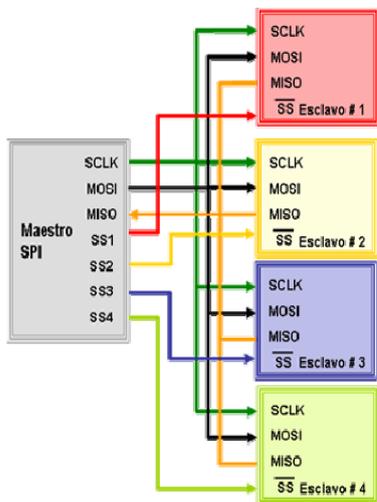


Fig. 5 Conexión de varios dispositivos SPI.

SCLK: Reloj del bus.

MOSI (Master Output Slave Input): Salida de datos del maestro y entrada de datos a los esclavos.

MISO (Master Input Slave Output): Salida de datos de los esclavos y entrada de datos al maestro.

SS (Slave Select): Habilitación del esclavo por parte del maestro.

2.6. Dirección IP y MAC

Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP (32 bits que suele ser mostrado en cuatro grupos de números decimales de 8 bits).

En redes de computadoras la dirección MAC (Media Access Control Address o dirección de control de acceso al medio) es un identificador de 48 bits (6 bytes) que corresponde de forma única a una tarjeta o interfaz de red.

2.7. Diferencias entre UDP y TCP

TABLA IV
DIFERENCIAS ENTRE UDP Y TCP

UDP	TCP
Servicio sin conexión; no establece una sesión entre los hosts.	Servicio orientado a la conexión; se establece una sesión entre los hosts.
UDP no garantiza ni confirma la entrega, y no secuencia los datos.	TCP garantiza la entrega mediante el uso de confirmaciones y la entrega secuenciada de datos.
Los programas que utilizan UDP son responsables de proporcionar la confiabilidad necesaria para el transporte de datos.	Los programas que utilizan TCP proporcionan la seguridad del transporte de datos confiable.
UDP es rápido, tiene requisitos de carga pequeños y puede admitir la comunicación punto a punto y de un punto a varios puntos.	TCP es más lento, tiene requisitos de carga mayores y sólo admite la comunicación punto a punto.

UDP y TCP utilizan puertos para identificar las comunicaciones para cada programa TCP/IP.

El campo de puerto tiene una longitud de 16 bits, lo que permite un rango que va desde 0 a 65535, pero no todos los puertos son de libre uso por lo que se detalla algunas normas sobre ellos:

- El puerto 0 es un puerto reservado, pero es un puerto permitido si el emisor no espera recibir mensajes del receptor.
- Los puertos que van del 1 al 1023 reciben el nombre de puertos “bien conocidos”, y en sistemas operativos tipo Unix es necesario acceder como súper usuario para enlazarse con uno de estos puertos.
- Los puertos que van desde 1024 al 49151 son los llamados puertos “registrados” y son de libre utilización.
- Los puertos del 49152 al 65535 son puertos “efímeros”, de tipo temporal y se utilizan sobre todo por los clientes al conectarse con el servidor.

III. DISEÑO DEL HARDWARE DEL REGISTRADOR DE VARIABLES ELÉCTRICAS

Para tener una concepción general del hardware empleado en el diseño del registrador de variables eléctricas, se lo ha dividido en varias partes otorgando una mejor visión del mismo. En la figura 6 se indica el diagrama de bloques del hardware del registrador.

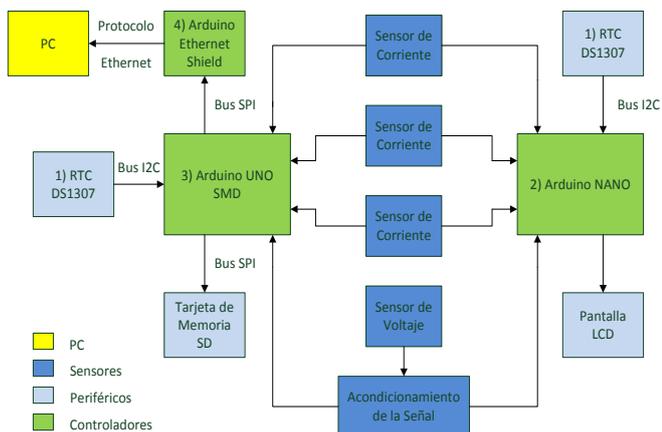


Fig. 6 Diagrama de bloques del hardware.

De acuerdo a la figura anterior el hardware consta de las siguientes partes:

- 1) Reloj de tiempo Real (RTC): Es un integrado diseñado para dar la hora y fecha sin perder su información.
- 2) Tarjeta Arduino Nano: Realiza la lectura de las mediciones de los sensores, separa los argumentos, controla fecha y hora del RTC y permite al usuario la visualización de la hora, fecha y de las mediciones realizadas en una pantalla LCD 20x4.
- 3) Tarjeta Arduino UNO: Realiza la lectura de las mediciones de los sensores cada cierto intervalo (5 segundos), separa los argumentos, crea un archivo con extensión “.CSV” en la tarjeta de memoria SD, controla hora y fecha del RTC, envía datos a la tarjeta de memoria SD y envía datos a la tarjeta Arduino Ethernet Shield.
- 4) Tarjeta Arduino Ethernet Shield: Permite enviar a través de una conexión Ethernet los datos adquiridos en la tarjeta Arduino UNO.

3.1. Sensor de Voltaje

Para adquirir la señal de voltaje se seleccionó un transformador de tensión (figura 7) con las siguientes características:

- Voltaje de entrada: 220V/110V
- Voltaje de salida: 12V – 0V – 12V
- Frecuencia: 50/60 Hz.
- Corriente: 300mA.

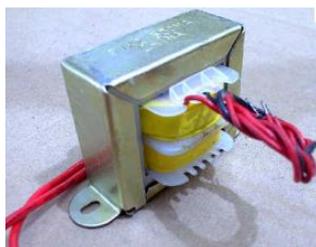


Fig. 7 Transformador de tensión.

3.1.1. Acondicionamiento de la señal de voltaje

Mediante el siguiente circuito que se muestra en la figura 8 se acondiciona la señal de voltaje para que pueda ingresar al convertor análogo-digital de las tarjetas Arduino UNO y Arduino Nano.

El transformador empleado en el circuito acondicionador de la señal de voltaje proporciona una tensión en el secundario de 12Vca para 220Vca a la entrada, y como en la entrada se está aplicando una tensión de 110Vca por eso la tensión en el secundario es de 6Vca, la cual al ser rectificadas es de 4.6Vcd ya que el puente de diodos consume 1.4V. Pero este voltaje de 4.6Vcd a 110Vca es muy alto aún para ser ingresado al convertor A/D de las tarjetas Arduino UNO y Arduino Nano, por lo que se diseñó un divisor de voltaje con el que se obtiene 2.3Vcd a 110Vca aproximadamente.

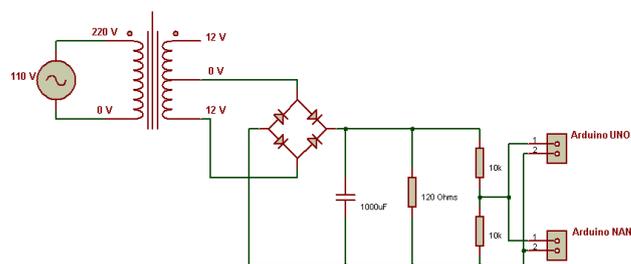


Fig. 8 Circuito acondicionador de la señal de voltaje.

3.2. Sensor de corriente

El sensor de corriente seleccionado para la implementación del registrador fue el sensor de efecto hall ACS712 20A (figura 9) ya que no presenta un error de repetitividad alto, además puede sensar corrientes menores a los 50 mA, mide corriente alterna y continua, y aparte ofrece aislamiento eléctrico al resto del sistema de adquisición separándolo de la red eléctrica.

Las características del sensor de corriente ACS712 20A son las siguientes:

- Pequeño y compacto.
- Error total de salida del $\pm 1.5\%$ a $T_A=25^\circ\text{C}$
- Rango de medida desde -20 a 20 Aca.
- Voltaje de alimentación 5Vcd.
- Corriente de alimentación 10mA.
- El ancho de banda del dispositivo se ajusta mediante el pin de filtro.
- Ancho de banda 80 kHz.
- 1.2 m Ω de resistencia conductora interna.
- Voltaje offset de salida extremadamente estable.
- Sensibilidad de salida 100mV/A.
- Voltaje de salida a 0A=2.5Vcd.

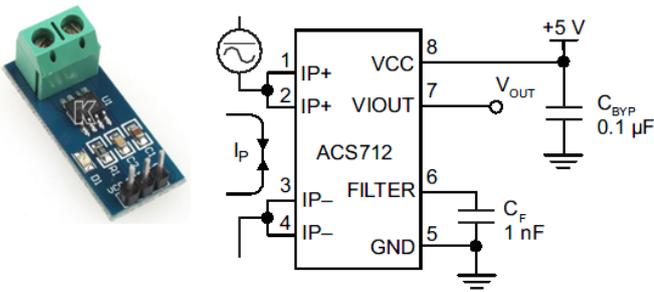


Fig. 9 Sensor de corriente ACS712 20A.

3.2.1. Acondicionamiento de la señal de corriente

El siguiente circuito que se indica en la figura 10 acondiciona la señal de corriente para que pueda ingresar al convertor análogo-digital de las tarjetas Arduino UNO y Arduino Nano.

El sensado de las tres líneas de corriente se realizó a través del sensor ACS712 20A. El rango de medición del sensor de corriente va desde -20A hasta 20A. Además, este sensor requiere para su funcionamiento una alimentación de 5Vcd y a su salida entrega un voltaje proporcional a la corriente de entrada en un rango de 0 a 5Vcd (nominalmente 2.5Vcd a una corriente de 0A). Este rango de voltaje se ajusta perfectamente al rango dinámico del convertor análogo-digital de las tarjetas Arduino UNO y Arduino Nano, por lo que el circuito de acondicionamiento de la señal de corriente resulta bastante simple.

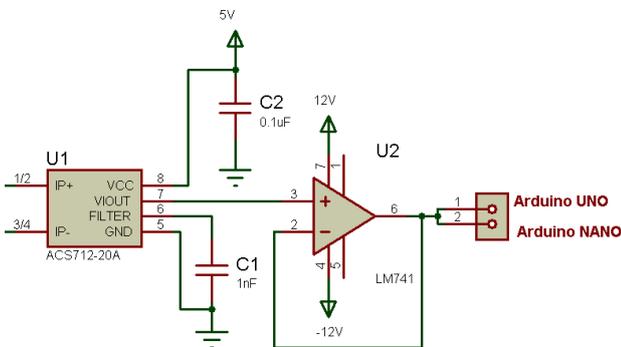


Fig. 10 Circuito acondicionador de la señal de corriente.

3.3. Reloj en tiempo real (RTC)

RTC es acrónimo de Real Time Clock o reloj en tiempo real, el mismo que se ha constituido en una herramienta de mucha utilidad para sistemas que usan aplicaciones de temporización industrial o en sistemas en los cuales se debe detectar eventos y la hora en la que ocurrieron. En la figura 11 se muestra físicamente al temporizador RTC.



Fig. 11 Temporizador RTC (DS1307).

Algunas de sus características principales son:

- El DS1307 es un RTC serial que procesa la información de los segundos, minutos, horas, día del mes, día de la semana, mes y año.
- El ajuste para la duración de los meses, incluso en los años bisiestos, es realizado por el propio circuito y es válido hasta el año 2100.
- Contiene 56 bytes de NVRAM (memoria RAM no volátil) para almacenamiento de datos.
- Permite operación en modo de 24 y 12 horas.
- Al presentar fallos en la alimentación (Vcc) el DS1307 cambia automáticamente al modo de operación con batería para no perder su configuración. Su consumo en esta condición, es menor de 500nA.
- Puede generar una señal de onda cuadrada de frecuencia programable.
- Rango de temperatura industrial opcional: -40 °C a +85 °C.
- Interface serie I2C.
- Detector automático de fallo de energía y circuito de conmutación.
- El pin 7 es una salida de colector abierto, que oscila a 1Hz. Esto permite la colocación de un led como indicador de segundos en aplicaciones de reloj.
- Los pines X1 y X2 son utilizados para la conexión de un cristal de cuarzo estándar de 32.768 kHz.

IV. DISEÑO DEL SOFTWARE DEL REGISTRADOR DE VARIABLES ELÉCTRICAS

El entorno de desarrollo para Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús como se indica en la figura 12. Este software permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.



Fig. 12 Entorno de desarrollo para Arduino.

Arduino utiliza para escribir la programación lo que denomina "sketch" (programa), los programas son escritos en el editor de texto. En el entorno de desarrollo de Arduino existe la posibilidad de cortar/pegar y buscar/remplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra los errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie.

4.1. Programación de la tarjeta Arduino Nano

El programa que se diseñó en la tarjeta Arduino Nano realiza las siguientes tareas:

- Graba la hora y la fecha actualizada en un Real-Time-Clock DS1307 mediante la interfaz de comunicación I2C.
- Muestra los datos de los sensores de corriente ACS712 20A tomando 5000 muestras cada 500 milisegundos.
- Acondiciona digitalmente la señal de los sensores de corriente ACS712 20 A en un rango de 0A a 20A.
- Acondiciona digitalmente la señal del sensor de voltaje en un rango que va desde 30V hasta 170V.
- Calcula los kWh.
- Presenta en la pantalla LCD los valores del voltaje, la corriente y los kWh. Además presenta la hora y fecha actualizada.

4.2. Programación de la tarjeta Arduino UNO

Para cumplir con el objetivo planteado se diseñó tres programas para la tarjeta Arduino UNO y estos son:

- a) Registrador de variables eléctricas remoto.

- b) Registrador de variables eléctricas con supervisión HMI diseñada en LabVIEW.
- c) Registrador de variables eléctricas remoto con monitoreo a través de un navegador web.

4.2.1. Registrador de variables eléctricas remoto

Este programa realiza las siguientes actividades:

- Graba la hora y la fecha actualizada en un Real-Time-Clock DS1307 mediante la interfaz de comunicación I2C.
- Crea el archivo de almacenamiento "ESPE-L00.CSV" en la tarjeta de memoria SD.
- Crea un nuevo archivo "ESPE-L00.CSV" cada vez que se reinicia la tarjeta Arduino UNO, el mismo que se enumera de 0 hasta 99 automáticamente.
- Creación de la cabecera del archivo "ESPE-L00.CSV" de almacenamiento (ID, HORA, FECHA, VOLTAJE, IL1, IL2, IL3, IT).
- Muestra los datos de los sensores de corriente ACS712 20A tomando 5000 muestras cada 500 milisegundos.
- Acondiciona digitalmente la señal de los sensores de corriente ACS712 20 A en un rango de 0A a 20A.
- Acondiciona digitalmente la señal del sensor de voltaje en un rango que va desde 30V hasta 170V.
- Almacena la información procesada de los sensores de corriente y del sensor de voltaje con hora y fecha en una tarjeta de memoria SD cada cinco segundos en el archivo "ESPE-L00.CSV".

4.2.2. Registrador de variables eléctricas con supervisión HMI diseñada en LabVIEW

Para realizar esta aplicación se desarrolló dos programas: uno en el software de Arduino y otro en LabVIEW. A continuación se describen las actividades que realiza el programa diseñado en el software de Arduino:

- Envía los datos de los sensores de corriente y del sensor de voltaje a LabVIEW mediante una comunicación Ethernet UDP, por lo que se le asigna una dirección MAC, una dirección IP y un puerto local a las tarjetas Arduino UNO y Arduino Ethernet Shield, los mismo que son:
 - Dirección MAC: 0xDE.0xAD.0xBE.0xEF.0xFE.0xED
 - Dirección IP: 192.168.1.177
 - Puerto local: 62024
- Muestra los datos de los sensores de corriente ACS712 20A tomando 5000 muestras cada 500 milisegundos.
- Lee los datos del sensor de voltaje.
- Lee el puerto local y la dirección IP de la PC con la cual se estable la comunicación Ethernet UDP.

- Convierte los datos de tipo “int” de los sensores de corriente y del sensor de voltaje a datos de tipo “char”, para poder ser enviados en una sola cadena de caracteres a LabVIEW, a través de la comunicación Ethernet UDP.

El programa diseñado en LabVIEW realiza las siguientes actividades:

- Abre la comunicación Ethernet UDP con la función de LabVIEW UDP Open. En la función UDP Open se debe asignar la dirección IP y el puerto local de la PC.
 - Dirección IP: 192.168.1.166
 - Puerto local: 49124
- Crea un archivo denominado “Datos.txt” en el escritorio de la computadora, en el cual se van a guardar los datos registrados de los sensores de corriente y del sensor de voltaje.
- Divide el string enviado por la tarjetas Arduino UNO y Arduino Ethernet Shield en cuatro substrings de cuatro caracteres mediante la función de LabVIEW String Subset, para luego convertir estos strings en valores decimales utilizando la función de LabVIEW Decimal String To Number.
- Acondiciona digitalmente la señal de los sensores de corriente ACS712 20 A en un rango de 0A a 20A.
- Acondiciona digitalmente la señal del sensor de voltaje en un rango que va desde 30V hasta 170V.
- Presenta las líneas de consumo de los sensores de corriente y del sensor de voltaje
- Guarda la información procesada de los sensores de corriente y del sensor de voltaje con hora y fecha cada dos segundos en el archivo “Datos.txt”.

4.2.3. Registrador de variables eléctricas remoto con monitoreo a través de un navegador web

El programa diseñado para esta aplicación realiza las siguientes tareas:

- Como su nombre lo dice esta aplicación permite monitorear en tiempo real los datos registrados en la tarjeta de memoria SD a través de cualquier navegador web, ya que la tarjeta Arduino UNO y la tarjeta Arduino Ethernet Shield están programadas para funcionar como un servidor web que puede soportar hasta cuatro conexiones simultáneas. Para que las tarjetas permitan el acceso a su información mediante un navegador web es necesario habilitar el puerto 80 (HTTP); y, además hay que asignar una dirección IP y una dirección MAC a las tarjetas Arduino UNO y Arduino Ethernet Shield, las mismas que son:
 - Dirección MAC:
0xDE.0xAD.0xBE.0xEF.0xFE.0xED
 - Dirección IP: 192.168.1.177

- Crea el archivo de almacenamiento “ESPE-L00.CSV” en la tarjeta de memoria SD.
- Crea un nuevo archivo “ESPE-L00.CSV” cada vez que se reinician las tarjetas Arduino UNO y Arduino Ethernet Shield, el mismo que se enumera de 0 hasta 99 automáticamente.
- Las tarjetas Arduino UNO y Arduino Ethernet Shield pueden ser reiniciadas desde la página web diseñada, mediante el botón de RESET.
- Creación de la cabecera del archivo “ESPE-L00.CSV” de almacenamiento (ID, VOLTAJE, IL1, IL2, IL3, IT).
- Muestra los datos de los sensores de corriente ACS712 20A tomando 5000 muestras cada 500 milisegundos.
- Acondiciona digitalmente la señal de los sensores de corriente ACS712 20 A en un rango de 0A a 20A.
- Acondiciona digitalmente la señal del sensor de voltaje en un rango que va desde 30V hasta 170V.
- Almacena la información procesada de los sensores de corriente y del sensor de voltaje en una tarjeta de memoria SD cada cinco segundos en el archivo “ESPE-L00.CSV”.
- Actualiza los datos registrados en la tarjeta de memoria SD cada vez que se refresca la página web y también permite visualizar todos los archivos que se encuentran almacenados en la tarjeta de memoria SD.

V. PRUEBAS Y RESULTADOS

A continuación se presenta el análisis del comportamiento práctico del dispositivo, en un ambiente real de trabajo. Las pruebas realizadas al prototipo ayudaran a conocer los errores que se producen durante el funcionamiento para mejorar el sistema y realizar las respectivas correcciones. En tanto que los resultados de las pruebas ayudaran a la determinación de importantes conclusiones.

5.1. Pruebas realizadas con los sensores de voltaje y corriente

Una vez realizado el respectivo escalamiento de los sensores en el software del registrador de variables eléctricas se efectuó las pruebas para determinar el grado de confiabilidad que brindará el prototipo.

Para el efecto de las pruebas en voltaje se utilizó el multímetro Fluke 117 y para las pruebas en corriente se utilizó el multímetro GW Instek GDM-8246. La forma en la que se llevó a cabo esta prueba fue mediante la comparación de las mediciones de los dispositivos anteriormente mencionados y el valor mostrado en el registrador de variables eléctrica. En las tablas 5 y 6 se indican las mediciones realizadas para esta prueba.

TABLA V
COMPARACIÓN DE LAS MEDICIONES REALIZADAS
CON EL SENSOR DE VOLTAJE

Fluke 117 (V)	Sensor de voltaje (V)	Desviación	Error (%)
30	30,114	0,114	0,38
40	40,067	0,067	0,1675
50	50,021	0,021	0,042
60	59,974	-0,026	-0,0433
70	70,923	0,923	1,3186
80	80,877	0,877	1,0962
90	91,826	1,826	2,0289
100	100,094	0,094	0,094
110	110,023	0,023	0,0209
120	119,952	-0,048	-0,04
130	129,881	-0,119	-0,0915
140	139,81	-0,19	-0,1357
150	149,739	-0,261	-0,174
160	159,669	-0,331	-0,2069
170	169,598	-0,402	-0,2365
		ERROR	0,4051%

TABLA VI
COMPARACIÓN DE LAS MEDICIONES REALIZADAS
CON EL SENSOR DE CORRIENTE ACS712 20A

GW Instek GDM-8246 (A)	Sensor ACS712 20A (A)	Desviación	Error (%)
0,525	0,53	0,0050	0,9524
1,172	1,162	-0,0100	-0,8532
1,47	1,5	0,0300	2,0408
2,12	2,13	0,0100	0,4717
2,639	2,64	0,0010	0,0379
3,088	3,1	0,0120	0,3886
3,485	3,5	0,0150	0,4304
4,12	4,15	0,0300	0,7282
4,616	4,6	-0,0160	-0,3466
5,098	5,01	-0,0880	-1,7262
5,568	5,555	-0,0130	-0,2335
6,135	6,14	0,0050	0,0815
6,675	6,67	-0,0050	-0,0749
7,245	7,25	0,0050	0,0690
		ERROR	0,6025%

Para el cálculo de la desviación se realizó una resta entre las dos medidas, la que marca el registrador de variables eléctricas menos el valor de los dispositivos de medida (Fluke 117 y GW Instek GDM-8246). En tanto

que para calcular el error de la medida del prototipo se divide la desviación para el valor verdadero y se multiplica por 100 para obtener el error en porcentaje. Como se puede observar el error de las mediciones de voltaje y corriente que marca el registrador de variables eléctricas para el sensor de voltaje es de 0,4051%, mientras que, para el sensor de corriente ACS712 20A es de 0,6025%, por lo que se puede determinar como válidas las mediciones realizadas por el registrador de variables eléctricas.

5.2. Análisis de la adquisición de datos del registrador de variables eléctricas

En las figuras 13, 14, 15, 16 y 17 se muestra el comportamiento y variación de la carga eléctrica de las variables consideradas por el registrador. La adquisición de datos se realizó en el laboratorio de Circuitos Electrónicos.

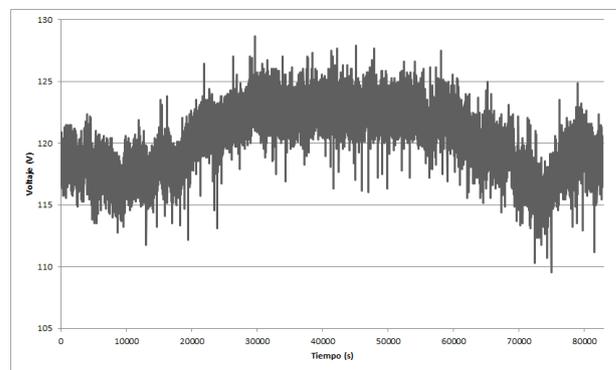


Fig. 13 Variación de voltaje versus tiempo.

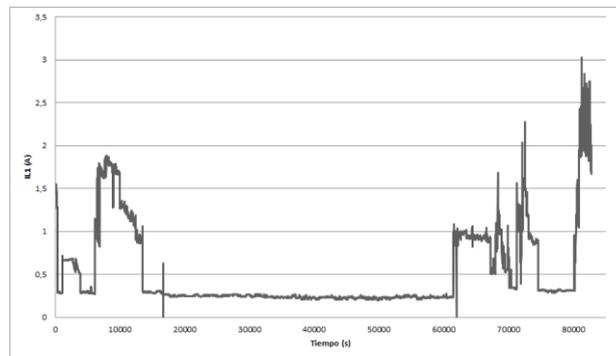


Fig. 14 Variación de corriente IL1 versus tiempo.

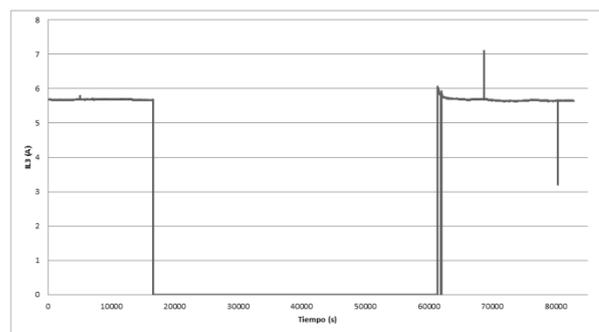


Fig. 15 Variación de corriente IL2 versus tiempo.

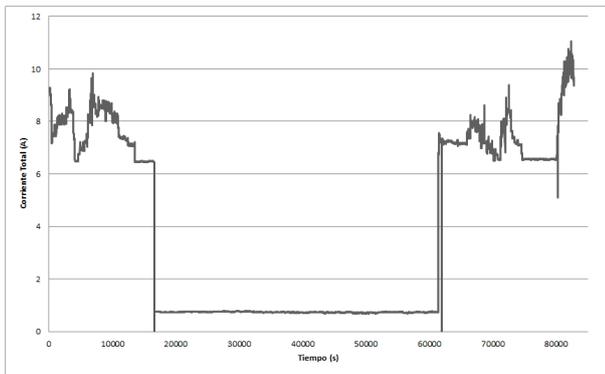


Fig. 16 Variación de corriente IL3 versus tiempo.

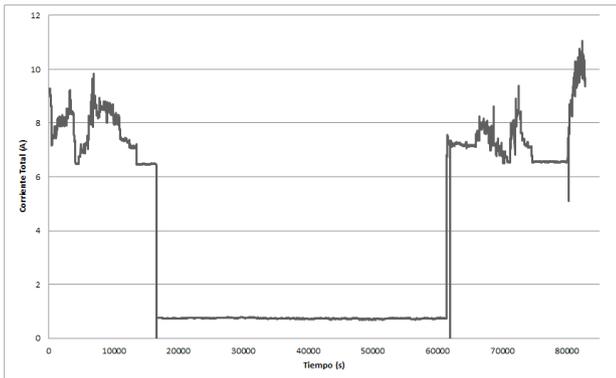


Fig. 17 Variación de corriente total versus tiempo.

La figura 15 corresponde al circuito de iluminación del laboratorio de electrónica en el cual se encuentran instaladas 16 lámparas fluorescentes de 40 W a 110 V, por lo que la potencia total del mismo es de 640 W y al dividir la potencia por el voltaje, se deduce que la corriente ideal es de 5.81 A, en tanto que la medición tomada por el registrador es de 5.69 A. Como se puede observar al restar la corriente ideal de la medición obtenida hay un error de 128 mA, el cual se produce por el deterioro de las lámparas o la variación del voltaje de alimentación, demostrándose así el correcto funcionamiento del dispositivo desarrollado.

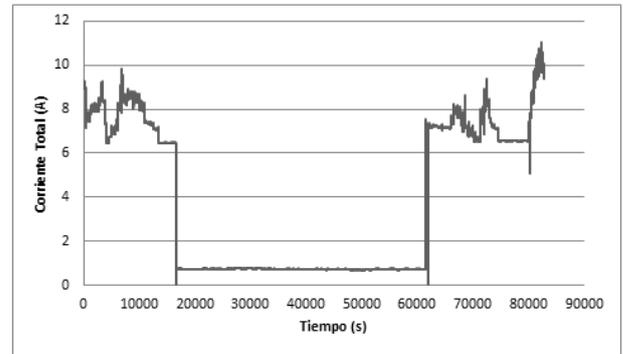
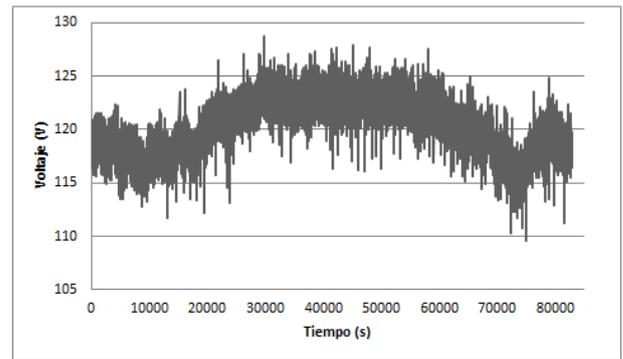


Fig. 18 Gráfica del voltaje y de la corriente total.

Como se puede observar en la figura 18 la tensión medida varía mucho, ya que al aumentar la demanda de corriente el voltaje tiende a disminuir, lo cual valida el funcionamiento del registrador.

Las gráficas anteriores se generaron a partir del archivo de almacenamiento del registrador de variables eléctricas. La adquisición de datos se realizó durante 23 horas por lo que se obtuvieron 13800 muestras; el registrador almacena una muestra cada 6 segundos, el archivo de almacenamiento ocupó un espacio de 453 KB en la tarjeta de memoria SD por lo que se puede determinar que una tarjeta de memoria de 2 GB permite almacenar la información de 4629 días aproximadamente, lo cual es suficiente para el propósito del registrador.

VI. CONCLUSIONES

- El almacenamiento de datos del registrador de variables eléctricas permite realizar comparaciones numéricas o visuales entre datos antiguos y datos actuales.
- Los valores medidos difieren de los valores calculados porque la tolerancia de las resistencias internas de los dispositivos utilizados no son exactas.
- El error de las mediciones del sensor de voltaje es de 0,4051% con relación al multímetro Fluke 117, mientras que, para el sensor de corriente ACS712 20A es de 0,6025% con relación al multímetro GW Instek GDM-8246.
- La exactitud del prototipo es directamente proporcional a la calibración realizada, por lo que

requieren patrones de referencia adecuados para su correcto funcionamiento.

- Las tarjetas Arduino UNO y Arduino Nano soportan perfectamente las comunicaciones SPI e I2C, lo cual facilita la conexión con diferentes periféricos.
- El software de programación de Arduino es una herramienta muy poderosa, en especial por la facilidad que tiene para agregar librerías, las mismas que proporcionan el soporte para realizar todo tipo de aplicaciones.
- El entorno de Arduino no presenta escalabilidad en sus diferentes versiones lo cual es una desventaja al momento de programar ya que el nombre de las funciones cambia, lo que complica el uso de las librerías del software.
- El dispositivo desarrollado puede ser utilizado para registrar el comportamiento de la carga de una residencia o de una industria, siempre y cuando sea un sistema monofásico.
- El desarrollo de este registrador de variables eléctricas permitirá en un futuro el estudio del comportamiento de la carga de una familia promedio, lo cual facilitará el desarrollo de múltiples proyectos de generación eléctrica utilizando energías renovables.
- Al finalizar este proyecto se concluye que este dispositivo es de beneficio para las empresas; pues permite comparar el consumo de kWh con el de la planilla eléctrica, así como poder determinar las horas de mayor consumo a través de las líneas de carga, lo cual facilitaría determinar el uso de cierta maquinaria para aprovechar las diferentes tarifas eléctricas y así generar ahorro monetario para la empresa y consumidor.

VII. RECOMENDACIONES

- Apagar el Bluetooth del computador cuando se vaya a utilizar el software de programación de Arduino para evitar el error launch4j, el cual es un error de java.
- Es importante conectar los sensores de corriente después de las protecciones del sistema eléctrico para evitar el daño de los mismos.
- Para acoplar el registrador de variables eléctricas a un sistema trifásico es necesario aumentar dos sensores de voltaje y modificar ligeramente su programación.
- Se recomienda para futuros temas relacionados utilizar la tarjeta Arduino MEGA para obtener una mayor capacidad de memoria de programación y así poder incluir el estudio de armónicos.
- Colocar el registrador en un sitio donde no este contacto con líquidos, calor excesivo entre otros. El sitio donde se lo coloque tiene que estar al alcance del usuario para que puede visualizar y configurar al dispositivo.
- Se recomienda la difusión del uso de las tarjetas Arduino ya que cuentan con herramientas de

desarrollo que facilitan la programación, permitiendo desarrollar un sin número de aplicaciones.

- Leer detenidamente las hojas técnicas de cada uno de los elementos utilizados en el desarrollo del registrador de variables eléctricas.

VIII. REFERENCIAS

- [1]. ARDUINO. "Arduino". [Documento HTML]. 4 de agosto del 2012. <http://www.arduino.cc/es/>.
- [2]. ARDUINO. "Arduino UNO". [Documento HTML]. 5 de agosto del 2012. <http://arduino.cc/en/Main/ArduinoBoardUno>.
- [3]. ARDUINO. "Arduino Nano". [Documento HTML]. 6 de agosto del 2012. <http://arduino.cc/en/Main/ArduinoBoardNano>.
- [4]. ARDUINO. "Arduino Ethernet Shield". [Documento HTML]. 7 de agosto del 2012. <http://arduino.cc/en/Main/ArduinoEthernetShield>.
- [5]. HISPAVILA. "DS1307". [Documento PDF]. 10 de agosto del 2012. http://www.hispavila.com/3ds/atmega/descargas/rtc_ds1307.pdf.
- [6]. LÓPEZ Q., JUAN A. "Diferencias entre UDP y TCP". 18 de agosto del 2012. http://dis.um.es/~lopezquesada/documentos/IES_0506/RAL_0506/doc/UT10.pdf.