UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

ESCUELA DE INGENIERIA ELÉCTRICA

Modificación de prototipo de medidor trifásico de consumo de energía eléctrica.

PRESENTADO POR:

DANIEL ALEJANDRO FLORES ABREGO

PARA OPTAR AL TITULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, AGOSTO 2013

UNIVERSIDAD DE EL SALVADOR

RECTOR :

ING. MARIO ROBERTO NIETO LOVO

SECRETARIA GENERAL :

DRA. ANA LETICIA ZAVALETA DE AMAYA

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO :

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL

SECRETARIO :

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERIA ELÉCTRICA

DIRECTOR :

ING. JOSÉ WILBER CALDERÓN URRUTIA

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

ESCUELA DE INGENIERIA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

Modificación de prototipo de medidor trifásico de consumo de energía eléctrica.

Presentado por :

DANIEL ALEJANDRO FLORES ABREGO

Trabajo de Graduación Aprobado por:

Docente Director :

Dr. CARLOS EUGENIO MARTÍNEZ CRUZ

San Salvador, Agosto 2013

Trabajo de Graduación Aprobado por:

Docente Director :

Dr. CARLOS EUGENIO MARTÍNEZ CRUZ

ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, 15 de agosto de 2013, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 5:30 horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. José Wilber Calderón Urrutia
   Director
   Firma: _____

2. Ing. Salvador de Jesús Germán
   Secretario
   Firma: _____

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

1- Ing. Jonathan Alberto Zaldaña Firma: _____

2- Ing. Walter Leopoldo Zelaya Chicas _____

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

Modificación de prototipo de medidor trifásico de consumo de energía eléctrica.

A cargo del Bachiller:

- Daniel Alejandro Flores Abrego

Habiendo obtenido el presente Trabajo una nota final, global de: 9.2

(NUEVE PUNTO DOS

Agradecimientos

Primero agradecer a Dios todo poderoso por darme la bendición de haber culminado mi carrera universitaria, gracias por darme la fuerza, sabiduría e inteligencia necesaria para alcanzar este éxito.

A mi mamá Reina Elizabeth Abrego de Flores y mi papá José Rigoberto Flores García, por haber sido el apoyo incondicional a lo largo de estos años de preparación académica, pero en especial gracias por todo su inmenso amor.

A mis hermanos Rigo y Diego, por su cariño, comprensión, apoyo y amor durante todo este tiempo, que con enojos, tristezas pero sobre todo alegrías, puedo decir junto a ustedes lo logre porque sé que este triunfo es de ustedes también.

Mis Abuelitos Luis, Lola y Dora (Dios tenga en su Gloria Eterna) que con su amor y comprensión siempre confiaron en mí, por su apoyo incondicional los amo.

A Heysel y Jonathan, primos, tíos y tías, a todos y cada uno de ustedes gracias por ese apoyo, confianza y amor a lo largo de estos años.

Mis amigos que estuvieron conmigo durante todos estos años, por todas esas noches de desvelos, preocupaciones y nerviosismo que pasamos ciclo tras ciclo en cada materia juntos, a todos ustedes gracias.

A mi Asesor de tesis el Dr. Carlos Martínez, por haber sido un gran maestro en mi formación profesional.

Y a todos que de alguna u otra manera me ayudaron a concluir esta etapa de mi vida muchas gracias a todos.

Daniel Alejandro Flores Abrego.

ACRÓNIMOS.

-EIE: Escuela de Ingeniería Eléctrica.

-AP: Punto de acceso (del inglés access point).

-MCU: Unidad Microcontroladora (del inglés microcontroller unit).

-PIC: Controlador de interfaz periférico (del inglés peripheral interface controller).

-SPUD: Panel de control simple unificado (del inglés simple unified dashboard).

-GUI: Interfaz gráfica de usuario (del inglés graphical user interface).

-PCB: Tarjeta de Circuito Impreso (del inglés printed circuit board).

- ADC: Convertidor analógico digital (del inglés analog to digital converter).

- BCD: Decimal codificado en binario (del inglés binary-coded decimal).

- UART: Transmisor-receptor asíncrono universal (del inglés universal asynchronous receiver-transmitter).

- I2C: Circuito Inter-Integrados (del inglés inter-integrated circuit).

- RTC:Reloj de Tiempo Real (del inglés Real Time Clock).

- WIFI: Fidelidad inalámbrica (del inglés wireless fidelity).

- NTP: Protocolo de tiempo de red (del inglés network time protocol).

- RRDtool: Herramienta de base de datos de round robin (del inglés round robin database Tool).

- B.A.T.M.A.N.: Better Approach To Mobile Adhoc Networking.

- B.A.T.M.A.N.D.: Better Approach To Mobile Adhoc Networking Daemon.

- S.E.P.: Sistema Eléctrico de Potencia.

- U.C.A.: Universidad Centro Americana "José Simeón Cañas".

- U.D.B.: Universidad Don Bosco.

- U.E.S.: Universidad de El Salvador.

Objetivo general:

- Modificar mediante su simplificación, uno de los prototipos de medidor de consumo de energía eléctrica construido en la Escuela de Ingeniería Eléctrica [4] [5].

Objetivos específicos:

- Reducir el tamaño del circuito sensor de energía del prototipo manufacturado en la EIE [4] [5].
- Reducir el costo de manufactura del circuito sensor de energía.
- Realizar un conjunto de experimentos conducentes a evaluar las prestaciones del medidor.

Índice:

Índice de Tablas:

Pag.

Indice de Figuras:

Pag.

Capítulo1:

1.1.Introducción.

En los últimos 25 años, en la Universidad de El Salvador se han realizado diferentes trabajos de graduación, con la finalidad de construir un dispositivo capaz de medir parámetros eléctricos y ser mostrados en un computador.

En mayo de 1990 se presentó el trabajo de graduación [1]. Este tomó como base la normativa de la distorsión armónica presente en la mayoría de países industrializados de la época. Muchos de estos países se vieron obligados a disponer de tres normas básicas tendientes a regular los niveles de producción de distorsión armónica en los sistemas de potencia. Primero, la necesidad de controlar la distorsión de la forma de onda del voltaje y de la corriente en los sistemas de potencia a niveles que los sistemas y componentes asociados puedan tolerar, Segundo, proveer a los consumidores conectados al sistema de potencia, que tenga una forma de onda adecuada para las necesidades particulares de estos, Tercero, asegurar que los sistemas de potencia no interfieran con la operación de otros sistemas, tales como los sistemas de comunicación y, en particular, con las redes telefónicas.

Por lo anterior expuesto y en vista que para los años 90 en el país no existía una instrumentación adecuada para el estudio de este fenómeno, mucho menos podía existir una normativa que lo regulara y de los altos costos de instrumentos desarrollados en otros países, se planteó en [1], el desarrollo de un instrumento que debía representar una opción real que se adaptara a las necesidades técnicas y económicas de nuestro país. Este instrumento consistía en tres bloques generales, el primero un circuito de potencia que permitía el acceso a la red de potencia, esto para censar la corriente y voltaje, transduciendo los valores sensados a señales apropiadas para su manejo. El segundo, el circuito de interfaz, que constaba de las puntas necesarias para realizar las pruebas al bus de baja tensión de la carga no lineal de un sistema de potencia y transformar a información digital las señales proporcionadas por el circuito de potencia. El tercero, y último bloque del instrumento, consistió en la utilización de la

microcomputadora COMMODORE 128 como elemento base para el desarrollo del instrumento. Así también mediante rutinas realizadas en Basic y Lenguaje de maquina se desarrolló un software, que se encargaba de controlar el funcionamiento de la interfaz, procesar la información y dar un resultado.

En el año 1991 se presentó el trabajo de graduación [2].Éste introdujo una interfaz para la comunicación entre un computador personal AT y un circuito sensor capaz de tomar información necesaria en redes AC de baja tensión. Se entiende por distorsión armónica, la deformación que sufren las ondas senoidales puras, cuando están “contaminadas”, por armónicos de diferentes órdenes, de la señal fundamental. Este trabajo de graduación tuvo como objetivo el diseño y construcción de un dispositivo que permitiera diagnosticar las formas de onda (corriente y voltaje) en una red AC de baja tensión, de manera que se pudiera obtener información acerca de los armónicos presentes.

La investigación se planteó para que el instrumento realizara cuatro funciones: Tomar la información del mundo real, almacenar la información, analizar la información y presentar la información.

Para la primera y segunda función, se diseñó un sistema de adquisición de datos, este se trabajó en dos partes: un módulo muestreador convertidor analógico a digital y un módulo de almacenamiento digital en memoria de la información. En el módulo muestreador se acondicionaban las señales reales de voltaje y corriente de la red a un nivel que era posible ser manipulados por los circuitos integrados.

Para la tercer y cuarta función se trató la comunicación entre el computador y el circuito sensor se dio por medio de una interfaz capaz de comunicar al computador con el exterior. El diseño se basó en la interfaz PPI 82C55A (Peripheral Programable Interface) de Intel. Este dispositivo de entrada/salida, poseía 3 puertos paralelos. Posterior a esta etapa se diseñó un programa en Turbo Pascal llamado Programa Análisis de Distorsión Armónica (PADA). El programa era el encargado de analizar y mostrar los resultados obtenidos. El programa PADA realizaba análisis de Fourier a las señales de voltaje y corriente.

Con ello se obtenía el espectro de frecuencias que permitía observar la presencia de armónicos en las señales.

En abril de 1996 se presentó el trabajo de graduación [20]. El principio fundamental de los multiplicadores usados por los watihorímetros electrónicos es llamado división por tiempo o modulación por ancho de pulso (PWM), a cualquiera de las dos señales voltaje o corriente, se le aplica una modulación por ancho de pulso y la otra una modulación por amplitud para así obtener la multiplicación de voltaje y corriente. Este trabajo de graduación consistió en el desarrollo de un diseño de un circuito electrónico muy simplificado. Este se basó en tres etapas principales. En la primera etapa se realizó el estudio de conceptos generales de mediciones de energía eléctrica y una breve historia de los principales movimientos en torno a medidores de energía en donde se describen las partes y mecanismos del medidor convencional. En la segunda etapa se realizó el diseño de un circuito muy simplificado, este se basó en una alternativa de medidor eléctrico propuesto por The Japan Electric Meters Inspection Coorporation, el diseño era especialmente apropiado para fase trifilar basándose en la conversión de potencia a frecuencia. La tercera etapa consistió en el análisis y presentación de los resultados de la implementación del circuito. Esto se llevó a cabo realizando pruebas de laboratorio y experimentales a cada una de las partes del circuito.

El circuito sensor constó de diferentes etapas, que le permitían tomar señales de voltaje y corriente, estas eran acondicionadas para aplicarle una modulación PWM, con ello se determinaba una señal proporcional de la potencia de entrada, que a su vez producía una serie de pulsos a través de un convertidor. Los resultados obtenidos eran mostrados en un contador digital de 7 segmentos.

En diciembre de 1996 se presentó el trabajo de graduación [21]. Éste consistió en la utilización de instrumentos virtuales, así como el desarrollo de una aplicación capaz de analizar la energía en redes trifásicas. Esto se logró con la ayuda y utilización de una tarjeta de adquisición de datos industrial, la PCL-718.Éste dispositivo estaba ligado a un computador PC XT/AT. Todo el proceso de

adquisición de tratamiento de las señales y de presentación se realizó con la herramienta LabWindows.

En febrero de 2005 se presentó el trabajo de graduación [3]. Este consistió en la construcción de un medidor de energía trifásica de estado sólido, por su precisión, exactitud y automatización en el campo de las mediciones eléctricas, se utilizó el IC ADE7754.

El medidor se diseñó con la capacidad de medir la potencia reactiva trifásica. Para lo cual contaba con 6 canales analógicos, 3 de corriente y 3 de voltaje. La capacidad nominal de medición trifásica era de 240 voltios línea-línea y 5 amperios rms. Como ya se dijo, este equipo medía las variables de corriente y voltaje rms en cada una de las fases, mediante la interfaz serial síncrona (SPI) del IC. Este equipo medidor contó también con la adición de un microcontrolador PIC16f877, este posee la capacidad de realizar funciones que antes hubiese sido necesario situar o realizar externamente con otros circuitos electrónicos, además de contar con una comunicación síncrona (SPI) y comunicación asíncrona (SCI).

Otra característica que resaltaba de este dispositivo es que contó con un reloj de tiempo real. Este era usado para efectuar las mediciones periódicas y a su vez ser almacenados en la memoria RAM y posteriormente ser visualizados en una interfaz gráfica de usuario desarrollado en LabView.

Para la adquisición y acondicionamiento de las señales para los canales de corriente y voltaje en el IC ADE7754, fue necesario la utilización de transductores de corriente y filtros anti-alias, los cuales sirvieron para evitar el traslape de las muestras de la señal de corriente de interés. Para las señales de voltaje fue necesaria la utilización de un circuito de atenuación de voltaje que consistió en una red de resistencia o divisor de voltaje, y luego de filtros anti-alias al igual que los canales de corriente.

En 2010 se inició la investigación sobre un medidor de bajo costo desarrollado por FLUKSO [7].Esta empresa, con sede en Bélgica, ofrece una forma de controlar el consumo energético residencial por medio de la monitorización de la potencia

eléctrica consumida. La monitorización y la visualización del consumo energético se presentan al usuario a través de un servicio web. Inicialmente el medidor utilizado por FLUKSO se basaba en un circuito sensor adherido a un router WAP-2102 de ABOCON. FLUKSO utiliza actualmente un router diseñado por ellos.

FLUKSO se basa en software libre. Sus desarrolladores han compartido la información necesaria para que usuarios sean capaces de poder modificar este medidor. A partir del año 2010 en la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador se desarrollan prototipos que tienen como base el medidor de Flukso. Los cambios principales hechos en la EIE fueron: Sustituir el Router utilizado por Flukso (WAP-2102 de ABOCON) por un router comercial y de bajo costo de venta en el país el cual es el router DIR-300 de la marca DLINK, la utilización de un microcontrolador PIC-16F876 para la realización de cálculos. También, se desarrolló un servicio web propio.

Para hacer posible lo antes mencionados se realizaron diversos trabajos de investigación. La primera etapa del diseño para un medidor monofásico tomó alrededor de dos años. Durante este tiempo se logró reducir el tamaño del circuito medidor, identificar sitios en internet para la compra e importación de los elementos electrónicos necesarios para la manufacturación del sensor, así como también conocer fabricantes de PCB's.

El primer prototipo medía instalaciones monofásicas [4], al igual que el medidor de FLUKSO, este mide únicamente corriente. El voltaje se fija a un valor constante de 120V. El transductor de corriente utilizado posee un valor máximo de 50 Amperios con una resolución de 1 Amp-AC/ 100 mV-DC (Se utilizó el mismo que utiliza el medidor Flukso).El valor de voltaje DC transformado se transfería a una entrada ADC del microcontrolador para ser digitalizado. Luego, el valor de corriente se multiplica por un valor constante de 120V, y así se obtiene el valor de potencia. Estos datos obtenidos eran enviados a través del puerto UART del microcontrolador (PIC16F876A) hacia el puerto UART del router DIR-300. Dentro del router los datos eran capturados por el demonio FLUKSO.lua que después de hacer unos cálculos, enviaba los datos hacia el servidor a través del protocolo de

red XML/RPC. Otra característica que poseía este circuito era la inclusión de un reloj de tiempo real RTC-1307, esto para dar a conocer la hora y fecha de cada dato enviado.

El formato utilizado para enviar los datos por medio del puerto UART se aprecia en la siguiente tabla:

COMANDO	ID	MEDICIÓN	FIN	PARAMETRO
Pwr	*2d4f76edc4dcf577fa0a49b8335l3bfe*	*:0000000120*	*0A*	*Potencia*
Pls	*2d4f76edc4dcf577fa0a49b8335l3bfe*	*:0000000300*	*0A*	*Energía*
Tim	*2d4f76edc4dcf577fa0a49b8335l3bfe*	*:1110011057*	*0A*	*Tiempo*

Tabla 1.1.1 Formato de datos enviados por el PIC al puerto UART [8] [9].

El router DIR-300, al igual que el router utilizado por FLUKSO WAP-2102,se configura en modo STA, por lo que era necesario otro router inalámbrico o un AP (Access Point por sus siglas en inglés) para enviar los datos al servidor de forma satisfactoria.

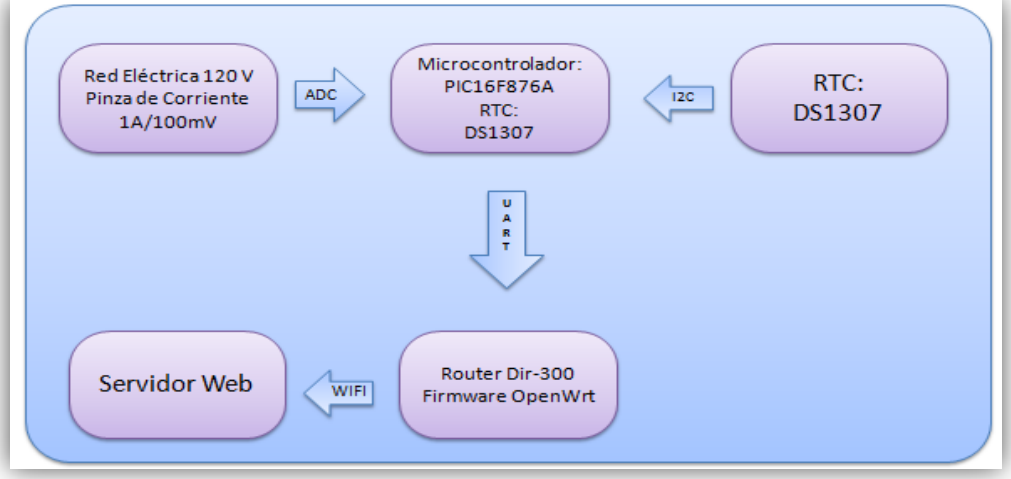


Figura 1.1: Bloques Medidor de Consumo eléctrico de bajo costo [8].

Como se aprecia en el diagrama de bloques de la Figura 1.1, la forma de trabajar del medidor diseñado en la EIE es idéntica al realizado por Flukso. En el primer bloque de la figura se aprecia que la corriente es sensada por medio de los

transductores de corriente. El voltaje es transformado a voltaje DC, que posterior es digitalizado por el MCU. El MCU toma la fecha y hora del RTC-1307 usando el pin i2c del microcontrolador. Posterior a esto se enviaban los datos calculados por medio del puerto UART hacia el router DIR-300. Este envía cada 5 minutos los datos promediados por medio de protocolo WIFI y XML/RPC hacia el servidor web para su almacenamiento y visualización.

En el año 2012, se trabajó en un nuevo dispositivo medidor de energía eléctrica [5].Se buscó que este nuevo dispositivo fuera capaz de medir el consumo energético en instalaciones trifásicas [5]. Se le dotó al router de la capacidad de trabajar en un entorno de red tipo malla. Se implementó un servidor web capaz de almacenar y visualizar los datos adquiridos por los medidores. El servicio web tiene una GUI donde se pueden visualizar los clientes de una red con tipología Malla y los enlaces activos entre ellos además posee la característica de dar la información de cada nodo (cliente) presente en la red.

Con la finalidad de mejorar el medidor se buscó en este trabajo de graduación: La reducción del tamaño del circuito sensor de potencia y energía eléctrica, la sustitución del circuito RTC, así como, la modificación de la etapa de alimentación, la sustituyó del RTC por un servicio NTP, así como también hacer mejoras y modificaciones del servidor de visualización de datos y gráficos en SPUD.

Capítulo 2:

2. Diseño del Circuito Medidor Trifásico y Mejoramiento en la Visualización de Datos en el Servidor Web.

2.1.Diseño del circuito medidor y código de microcontrolador.

El nuevo diseño del circuito medidor tuvo como objetivo principal utilizar la menor cantidad de hardware posible. El circuito, al igual que sus predecesores [4][5], hace uso de transductores de corriente. Estos datos son enviados a través de tres pines ADC del microcontrolador. No se utilizan transductores de voltaje, se fija un valor constante de 120 V.

Figura 2.1.1. Acople de los Transductores de Corriente en el Medidor.

Para lograr reducir los costos del medidor de energía trifásico se identificó que las etapas del regulador de voltaje de 5V y de Reloj de tiempo real RTC, son partes de los diseños que se pueden omitir. La alimentación para el sensor medidor se puede realizar mediante el puerto UART del router DIR-300. Por otra parte el reloj RTC puede ser sustituido por un servicio web NTP (Network Time Protocol).

Con ello se logra una reducción en el tamaño de la tarjeta impresa, reducción de elementos electrónicos y en general reducción significante en los costos de fabricación del sensor de energía trifásica.

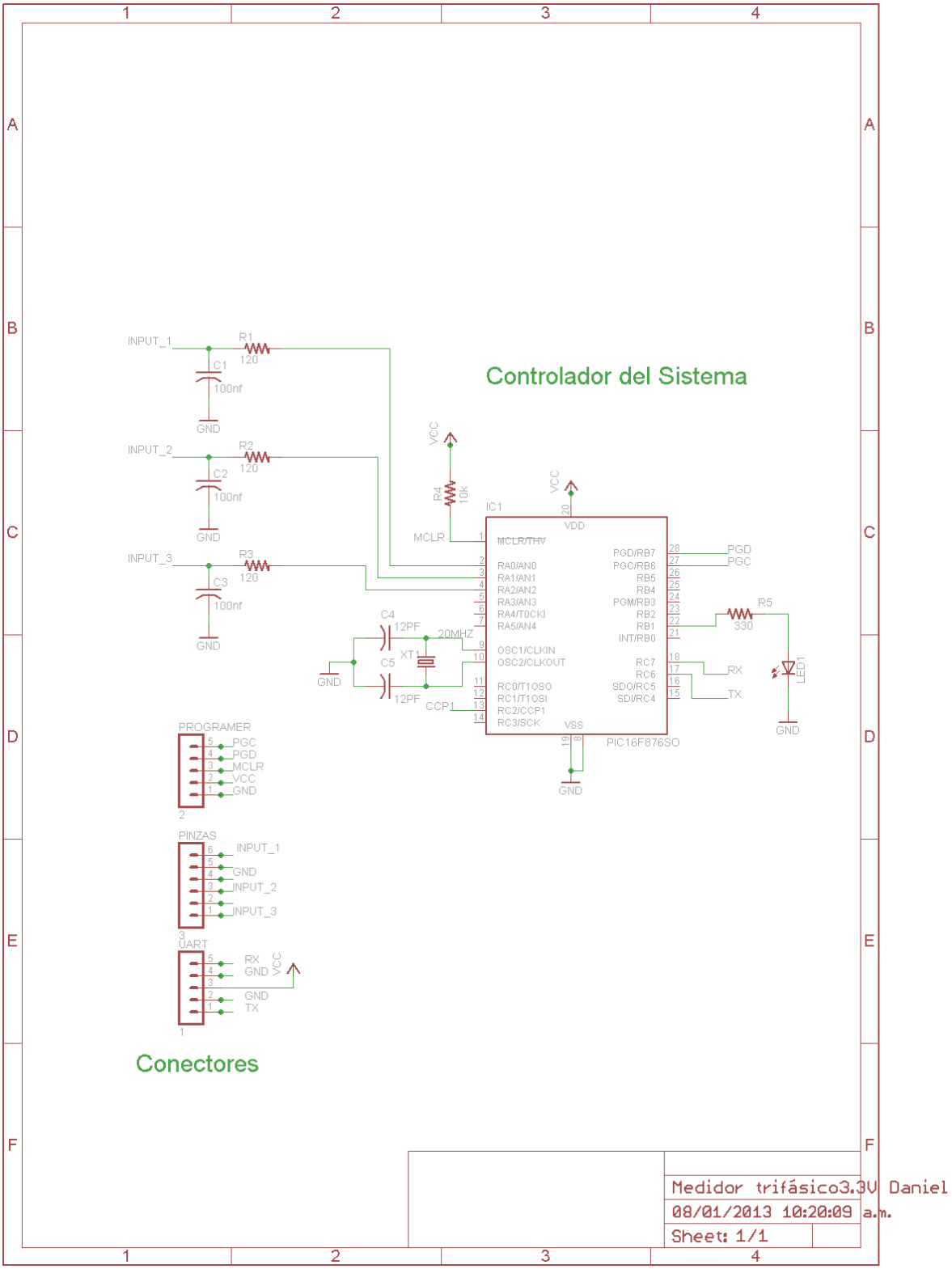


Figura 2.1.2. Diagrama esquemático del nuevo circuito medidor de potencia.

En la figura 2.1.2 se muestra el diagrama esquemático del nuevo circuito medidor de potencia, en este diagrama se puede apreciar que las etapas de regulador de voltaje y de reloj RTC presentes en investigaciones previas [4] [5] ya no están presentes. Así como también se reduce el número de elementos electrónicos a utilizar.

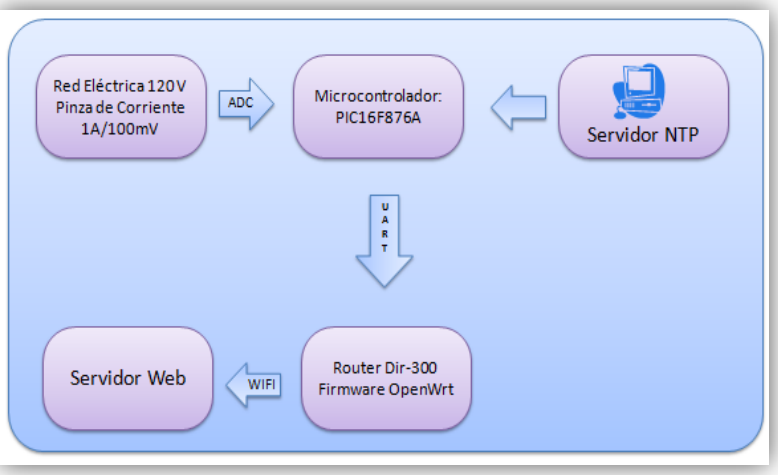


Figura 2.1.3. Bloques del nuevo prototipo medidor de consumo eléctrico de bajo costo.

En la Figura 2.1.3 se observa el nuevo diagrama de bloques del circuito medidor, la carencia de un circuito RTC se resuelve mediante la utilización de un servidor NTP en la computadora configurada como servidor. Es importante mencionar que los medidores de energía fabricados por FLUKSO trabajan de la misma forma, estos no poseen un RTC en su circuito medidor.

Este servidor NTP (Network Time Protocol) es un protocolo de internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes. Utiliza UDP como su capa de transporte, usando el puerto 123, y está diseñado para resistir los efectos de la latencia variable.

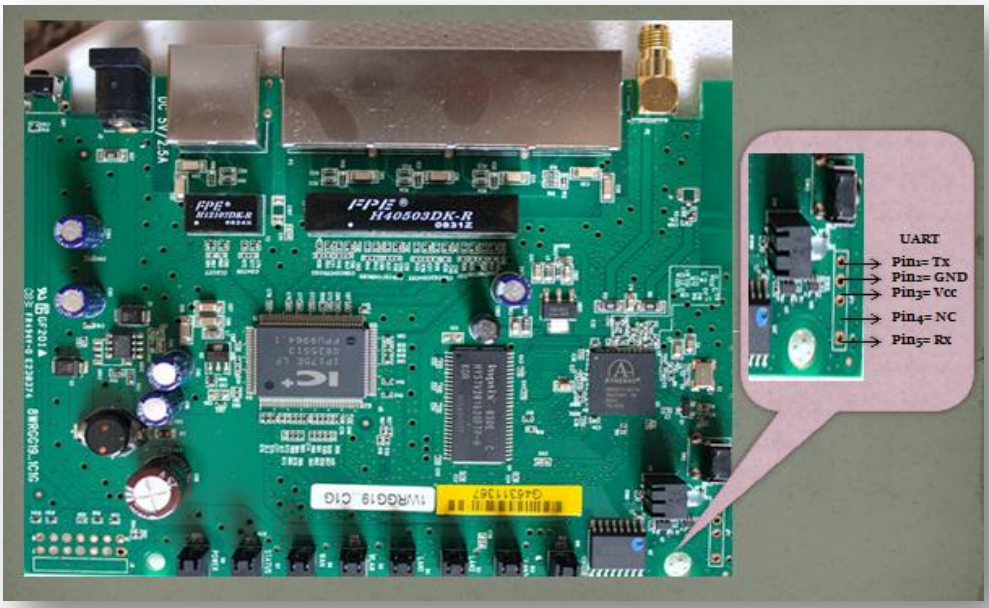


Figura 2.1.4: Ubicación del puerto UART en el router Dir-300 [19].

En un reciente estudio del puerto UART del router DIR-300 [8] [19], se identificó la distribución de pines de la Figura 2.1.4 Donde: **Tx** es el pin de transmisión serie, este es capaz de transmitir hasta un máximo de 57600 baudios. **GND** es el punto común del circuito. **Vcc** es salida de alimentación a un voltaje de 3.3V. **NC** es un pin flotante y no tiene conexión con el circuito impreso del router DIR-300. **Rx** es el pin de recepción del puerto UART. El Pin3 del J1 tiene presente una fuente de energía de 3.3 voltios, con este valor de energía puede alimentarse el microcontrolador para su funcionamiento.

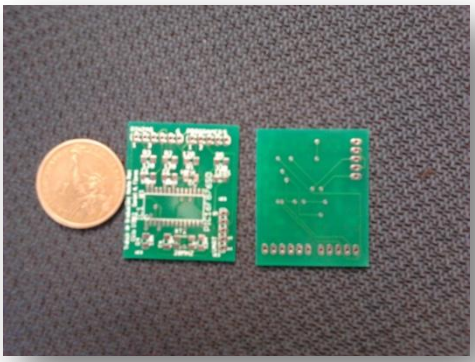


Figura 2.1.5: Distribución física de los elementos en el PCB.

Al modificar la fuente de energía que alimenta al microcontrolador, el voltaje de referencia que el convertidor ADC utiliza debe ser modificado esto en comparación al voltaje de referencia utilizado en el dispositivo monofásico [4] y trifásico [5] anteriores, y por tanto la resolución a utilizar es la que se muestra en la ecuación 2.1.1:

$$resolución = \frac{3.3\ V}{1024} = 3.22\ mV$$

Con una carga de 1AAC, se tendrá 100mV-DC en la entrada del ADC, el valor digitalizado por el PIC se calcula usando la ecuación 2.1.2:

$$Valor_{ADC} = \frac{100mV}{3.22mV} = 31.05$$

Para obtener el valor de la corriente hacemos el cálculo presente en la ecuación 2.1.3:

$$Corriente = Valor_{ADC} * \frac{3.3V}{1024} = 0.1000635\ Amp$$

Este valor tiene que ser representado como $1\ Amp$.Para lograrlo solo falta multiplicarlo por 10 unidades: $0.1000635 * 10 = 1.000635\ Amp$. Como resultado obtenemos un voltaje de referencia positivo del ADC.

Con este cambio de alimentación realizado al dispositivo medidor, es necesario cambiar también el código utilizado del microcontrolador [5].El voltaje de referencia de 5V pasa a 3.3V.Por lo consiguiente hay que realizar dicha modificación en las líneas de programación del módulo ADC del microcontrolador. En lenguaje de programación en "C" esto se traduce en lo siguiente:

```
corriente = 3.3*readADC/1024; //Escala de 0 a 3.3 voltios.
```

Figura 2.1.6: Voltaje de referencia ADC.

La línea de código de la Figura 2.1.6 usa por defecto el voltaje de alimentación como valores de referencia del ADC. Además, con la supresión del RTC DS1307 del circuito sensor ya no se generará una señal de 1Hz, por ello es necesario sustituir la interrupción externa del microcontrolador.

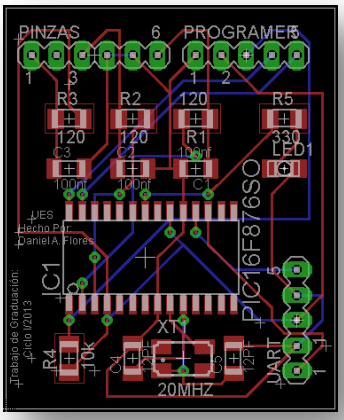


Figura 2.1.7: PCB generado para el sensor del medidor de potencia.

En la Figura 2.1.7, se observa el diseño final implementado en este trabajo. Éste diseño ya no incluye el RTC DS-1307, ni el regulador de voltaje. Comparando este nuevo diseño trifásico con su antecesor [4] [5], el área física del sensor reduce en aproximadamente un 55% del tamaño original.

2.2.Instalación del Servidor NTP:

Los datos medidos deben incluir la fecha y el tiempo. En los diseños anteriores tanto monofásico [4] y trifásico [5], el encargado de llevar este sincronismo era el RTC DS-1307.Éste generaba una señal de 1 Hz, para generar una interrupción externa necesaria para activar y leer los 3 pines ADC del microcontrolador en donde estaba representada la corriente que se media. El nuevo circuito ya no presenta esta etapa. En su lugar se utiliza un servidor web NTP (Network Time Protocol). Éste se encarga de que cada dato enviado de potencia y energía posea la fecha y hora en el que fueron tomados y enviados. Para sustituir la señal de 1Hz que era generada mediante el RTC DS1307 y poder leer los 3 pines ADC del microcontrolador, se hace uso de la interrupción timer1 del PIC, se sabe que este se activa cada 10 ms por lo que al código se agrega una variable que cuenta 100 interrupciones del timer1 equivalente a 1 segundo para poder leer los datos de los 3 pines ADC y luego ser enviados mediante el puerto UART.

Para poder utilizar el servidor web NTP en nuestros medidores, necesitamos instalar ciertos paquetes y configuraciones en nuestro servidor (PC) y router Dir-300 con firmware Backfire. Para instalar el firmware Backfire en nuestro Dir-300 es necesario flashear nuestro router para poder realizar esto se debe seguir uno a uno los pasos dados en el trabajo [12].

Como primer paso se debe instalar el servidor NTP en nuestro servidor (PC), abrimos un terminal de Ubuntu, una vez se despliegue la ventana terminal digitamos lo siguiente:

```
daniel@daniel:~$ apt-get install ntp
```

Figura 2.2: Comando de instalación del servidor NTP.

Una vez instalado todas las dependencias y paquetes necesarios, observamos el estado del servidor NTP en nuestra PC:

```
daniel@daniel:~$/etc/init.d/ntp status
 * NTP server is running
```

Figura 2.2.1: Comprobación del estado del servidor NTP.

Observamos que todo está en perfecto estado y nuestro servidor NTP funciona, luego procedemos a modificar el archivo de configuración ntp.conf en nuestro servidor, haciendo lo siguiente:

```
daniel@daniel:/etc# vi /etc/ntp.conf
```

Figura 2.2.2: Editamos el archivo ntp.conf mediante el editor de texto vi.

Al ejecutar el comando anterior se nos muestra el archivo de configuración *ntp.conf*, se selecciona todas las líneas presentes en el archivo y se reemplaza por lo siguiente:

```
#ntp server config file
restrict default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict 192.168.255.0 mask 255.255.255.0
server  127.127.1.0
fudge   127.127.1.0 stratum 0
driftfile /var/lib/ntp/drift
```

Figura 2.2.3: Archivo de configuración del servidor ntp.

En la Figura 2.2.3 se muestra las líneas de configuración necesarias para el servidor NTP. En él se listan dos direcciones IP (Internet Protocol) o servidores y su máscara de enlace. La primera es con la que se desea hacer el sincronismo remoto, y la siguiente dirección es una Pseudo-IP para él mismo (en este caso 127.127.1.0). La Pseudo-IP se utiliza en el caso de errores en la red o si cae el servidor NTP remoto. NTP sincronizará consigo mismo hasta que pueda empezar a sincronizar de nuevo con el servidor remoto.

2.3.Modificación realizada a SPUD:

Para esta investigación se utilizó SPUD como software de visualización de la red tipo malla [9], ya que recientes trabajos realizados en Escuela de Ingeniería Eléctrica de la Universidad de El Salvador mostraron su confiabilidad al ser de fácil instalación, manipulación y portabilidad con el sistema Ubuntu 12.10. Una característica que resalta SPUD es ser una herramienta web usada por empresas como Village Telco para monitorizar redes en configuración tipo malla, así como la configuración utilizada en esta investigación.

En el presente trabajo se realizaron ciertas modificaciones a la configuración de SPUD. En trabajos anteriores ya había pasado en una etapa previa de depuración que facilitó la visualización de gráficos de potencia y energía.

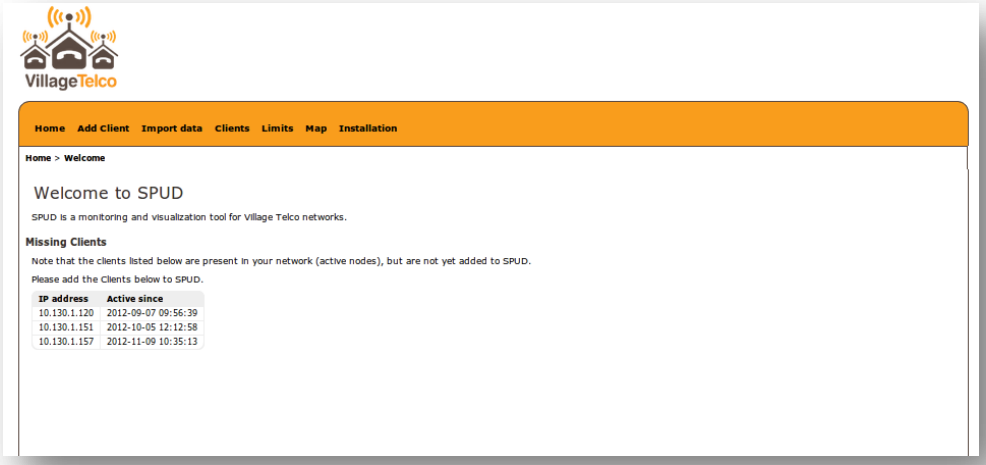


Figura 2.3. Página de inicio de SPUD.

SPUD es una interfaz gráfica sencilla implementada en Cake-PHP (una pequeña introducción a Cake php se encuentra en el Anexo B de [5]) para poder gestionar la red tipo malla. Consta principalmente de tres funciones: añadir nodos, lista de nodos y mapa. La Figura 2.3 muestra la pantalla de inicio de SPUD esta se puede acceder desde el navegador web con la URL *http:///localhost/spud/.* Abajo del texto de bienvenida SPUD muestra los nodos que han sido vistos dentro de la red malla que él está gestionando y aún no han sido dados de alta en el servidor. En la parte superior se encuentra el menú principal se observan las tres funciones mencionadas antes (add client, clients y map) y además incluye una opción poder importar datos de nodos desde un archivo de texto (import data), otra para introducir las métricas de B.A.T.M.A.N (limits), una guía rápida de instalación (instalation) y un enlace a la página de inicio (home).

Los pasos detallados para la instalación de SPUD (con el código fuente codificado) pueden encontrarse [5].

2.3.1. Modificaciones en la forma de visualizar los datos en SPUD.

La forma de visualizar los datos enviados por los medidores hasta nuestro servidor en nuestra PC es generando gráficos por medio de la base de datos utilizados en nuestro servidor, esto se logro haciendo modificaciones al código fuente de SPUD. Para comprender la forma de trabajar de las bases de datos en SPUD se puede consultar las modificaciones realizadas en una investigación previa [5].

La forma de trabajar y manipular los datos por medio de SPUD es mediante base de datos, una implementada en MYSQL y otra en RRDTOOLS, aquí son almacenados los datos de energía y potencia enviados por los nodos medidores de la red. Las bases de datos implementadas para cada medidor en el sistema son archivos creados con RRDTool con extensión .rrd y con el ID del medidor por nombre, estos archivos están almacenados en la dirección */usr/cgi-bin/rrd_database* de nuestro sistema operativo.

2.3.2. Extracción de datos de los gráficos.

La forma de visualizar los datos enviados por los medidores es por medio de gráficos generados a partir de las bases de datos. Estas se encuentran ubicados en la dirección*/usr/cgi-bin/rrd_database* y tienen por extensión.*rrd* y el nombre ID del medidor respectivo.

Para hacer posible lo anterior y lograr una integración entre los datos enviados por los Medidores (Protocolo XML-RPC) y la creación de una base de datos con RRDTOOLS, se utilizaron dos scripts desarrollados en el trabajo de graduación [8].Estos poseen la característica de crear una base de datos RRDTOOLS para las mediciones de energía y otro que es el encargado de almacenar los datos o mediciones de la potencia eléctrica.
Los scripts antes mencionados son: create_rrd.sh y createE_rrd.sh ambos tienen una ubicación en la dirección*/usr/lib/cgi-bin/.*

Posterior a la creación de las bases de datos, ya sea de Energía o Potencia Eléctrica, generados por los script antes mencionados, se implementaron dos script que son los encargados de mostrar en forma de gráficos las mediciones almacenadas en los archivos con extensión .rrd. Dichos script están ubicados en la dirección */usr/lib/cgi-bin/*. Estos llevan por nombre *"graficar.sh"* y *"graficarE.sh"*. Estos archivos generan una imagen PNG con los datos almacenados en los archivos .rrd. Los script utilizados son los mismos que se implementaron en [4].

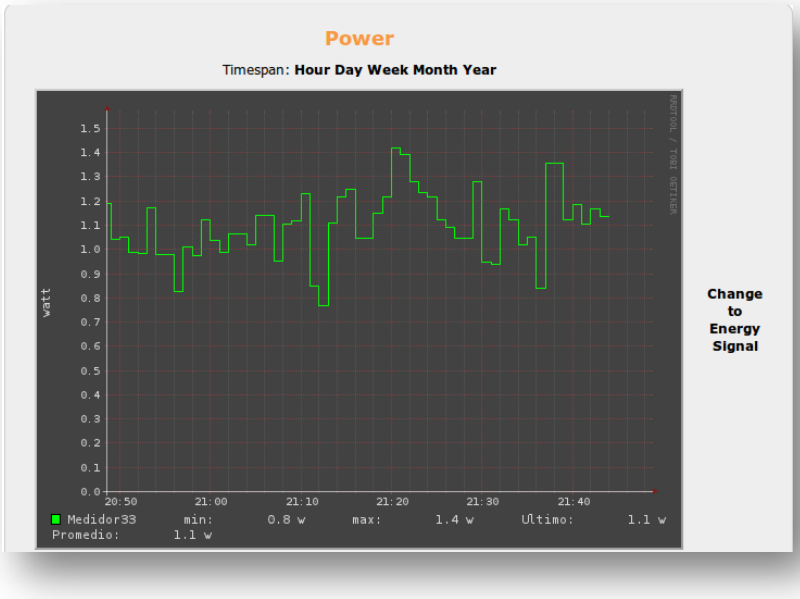


Figura 2.3.1.Gráfico de potencia generado por RRDTOOLS.

La Figura 2.3.1 nos muestra la forma en que los datos almacenados en los archivos .rrd son representados gráficamente, como se observa en este gráfico (eje vertical) se muestra una tendencia de la potencia consumida en un periodo de tiempo (Eje horizontal).

Lo que se pretende en esta sección es facilitar el manejo de los datos y la forma de ser representados, no solo por medio de gráficos sino también manipular cada dato enviado por los medidores según su hora y fecha.

En esta investigación se han implementado dos script, "*búsqueda.sh*" y "*busquedaE.sh*", estos tienen una ubicación en la dirección */usr/lib/cgi-bin/*. Estos

script se encargan de realizar una búsqueda de los archivos.*rrd* creados por la base de datos RRDTOOLS, la búsqueda de estos archivos se realiza en la dirección */usr/lib/cgi-bin/*de nuestro sistema operativo*,* sean estos de energía o potencia yel ID del medidor que este posea y se haya indicado, al localizar el archivo necesario se hace uso de la función XPORT presente en las librerías de RRDTOOLS, esta función se encarga de transformar nuestro archivo .rrd y generar un archivo XML (.xml).Los parámetros que esta función necesita para hacer la búsqueda y generar el archivo es una fecha de inicio, una fecha final y el ID del medidor a buscar. Esto se realiza mediante la implementación de un script hecho en PHP que lleva por nombre *"búsqueda.php*" el cual se encarga de enviar los parámetros necesarios a los script*busqueda.sh* o*busquedaE.sh.* Al momento que el script recibe los parámetros de fecha y hora se realiza la ejecución de la instrucción `strtotime($limite_s)` de php, este se encarga de hacer una conversión de tiempo UTC (tiempo estándar) a tiempo UNIX que está definido como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970, la forma grafica en SPUD de enviar los parámetros es como se aprecia en la Figura 2.3.2:

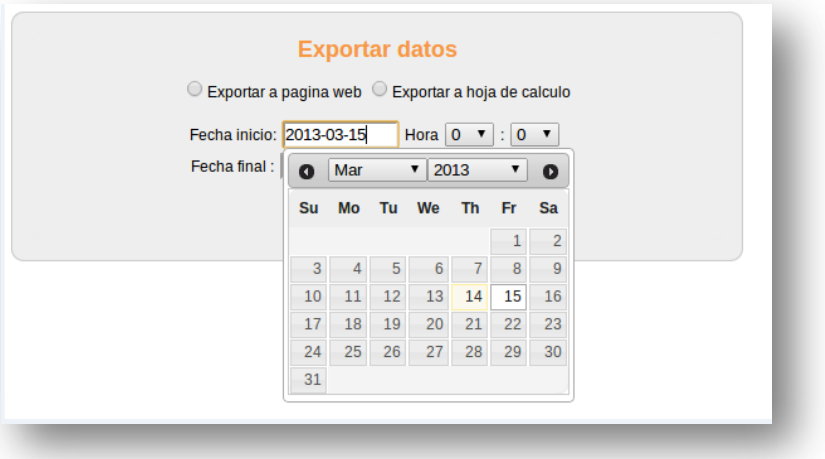


Figura 2.3.2: Representación gráfica del script implementado en php para el ingreso de fecha y hora.

El archivo *"ID_Medidor.xml"* generado es almacenado en la dirección */var/www/xml/* del sistema operativo. Como se observa en la figura 2.3.2 los

archivos .xml generados por el script búsqueda.php, también nos permite exportar los datos almacenados en los archivos .xml a un formato o viñeta de pagina web o a un archivo con formato o extensión .xlsx (Excel) y asi poder hacer más fácil su manipulación y visualización de los datos presentes en los gráficos desde otros equipos, esto se realiza mediante la implementación de dos script "*to_web.php*" y "*to_excel.php*".

Para ver el código fuente de cada uno de los script mencionados e implementados en esta sección ver los Anexos E.

Capítulo 3:

Resultados del Circuito Medidor:

En esta sección se pretende dar a conocer la forma de configurar y trabajar de los nodos medidores, también observar las modificaciones realizadas al servidor de visualización SPUD.

3.1 Configuración y sincronización del servidor NTP.

Como primer punto en la configuración de los router medidores es necesario configurar toda la red tipo Mesh (Malla).Para ello se hace uso de las indicaciones dadas en [9].Teniendo todo perfectamente configurado se hace uso del script "*configEth0.sh*".Este script es el encargado de la configuración automática del servidor. Es decir el servidor de la red necesita acceder a dos redes diferentes. La primera red es la que provee servicio de internet, la cual posee una dirección IP que se configura de forma dinámica por el servidor DHCP del Gateway, esta dirección se encuentra ubicada en el rango *192.168.1.xx/24*. La segunda red es un alias y es la que se encarga de comunicar el servidor en nuestro ordenador con los medidores de la red tipo Malla. Las líneas de configuración de la red se muestran en la Figura 3.1:

```
sudo ifconfig eth0:1 10.30.1.10/24 up

sudo route add -net 10.130.1.0/24 gw 10.30.1.1
sudo route add -net 10.30.11.0/24 gw 10.30.1.1
sudo route add -net 10.30.12.0/24 gw 10.30.1.1
sudo route add -net 10.30.13.0/24 gw 10.30.1.1
sudo route add -net 10.30.14.0/24 gw 10.30.1.1
sudo route add -net 10.30.31.0/24 gw 10.30.1.1
sudo route add -net 10.30.32.0/24 gw 10.30.1.1
sudo route add -net 10.30.33.0/24 gw 10.30.1.1
sudo route add -net 10.30.34.0/24 gw 10.30.1.1
sudo route add -net 10.30.35.0/24 gw 10.30.1.1
sudo route add -net 10.30.36.0/24 gw 10.30.1.1
sudo route add -net 10.30.37.0/24 gw 10.30.1.1
```

Figura 3.1: Script para configurar el servidor.

Una vez ejecutado el script "*configEth0.sh*", procedemos a ingresar mediante el protocolo de intérprete de orden seguro SSH (por sus siglas en ingles Security Shell) al SuperNodo de la Red. El SuperNodo es el encargado de enrutar los datos de la red hacia el servidor. El SuperNodo no cuenta con la capacidad de medir energía eléctrica, pero si con la característica de monitorizar mediante el servidor VIS los nodos y la calidad de su conexión WiFi entre cada uno de ellos por medio del servidor SPUD.

Ingresamos al SuperNodo mediante la instrucción *ssh root@10.130.2.1*en un terminal y cuando se pide ingresar el password digitamos "root" y presionamos un enter, de la misma forma procedemos a ingresar a un medidor de energía mediante ssh a la dirección IP 10.130.2.33 con password "root", que es el Medidor #33 de la red en forma de ejemplo.

Dentro de la configuración del Medidor 33 hacemos el sincronismo del servidor NTP entre el servidor del ordenador y el router, en primera instancia el medidor al introducir el comando "date" en el terminal, nos mostrara una fecha y hora que no es la actual en nuestro ordenador, esto se debe a que se sincroniza o muestra la fecha de creación de los sistemas Unix, luego de verificar esto ingresamos la instrucción *ntpclient –s –h 10.30.1.10* en el terminal, con esto hacemos el sincronismo mediante la red a la zona horaria de nuestro ordenador, así como se observa en la figura siguiente:

```
root@flukso:~# date
Wed Dec 31 18:09:10 CST 1969
root@flukso:~# ntpclient -s -h 10.30.1.10
25567 00611.052 2092.0 126.5 1368327913322894.9 11474.6 0
root@flukso:~# date
Sat May 11 21:15:35 CST 2013
root@flukso:~#
```

Figura 3.1.1: Configuración y Sincronismo con el Servidor NTP.

Para facilitar la configuración y el sincronismo entre nuestros medidores y el servidor NTP hacemos uso del archivo oculto */home/usuario/.profile* de OpenWrt.

Este archivo permite especificar una serie de comandos que se ejecutarán cuando el usuario acceda a su cuenta. En caso de que el archivo no exista se puede crear mediante el comando *vim .profile* e ingresar las siguientes instrucciones.

```
Echo “Actualizando Fecha” > log.txt
date >> log.txt
sleep 5m
ntpclient –s –h 10.30.1.10
echo “Su fecha fue Actualizada” >> log.txt
date >> log.txt
```

Figura 3.1.2: Configuración y Sincronismo entre los nodos medidores y el Servidor NTP mediante la utilización del archivo .profile de OpenWrt.

En la Figura 3.1.2 se observa las instrucciones necesarias en el archivo */usuario/.profile* para la configuración automática entre los medidores y el servidor ntp de nuestro ordenador. En la primera y segunda línea se nos genera un archivo *.txt* (log.txt). En este se almacena la fecha actual presente en nuestro medidor. En la tercer y cuarta instrucción se ejecuta un tiempo aproximado de sincronismo entre los medidores y el servidor ntp de la red. Este tiempo es aproximadamente de 5 minutos (*sleep 5m).* Transcurrido este tiempo se ejecuta el comando ntpclient –s –h 10.30.1.10, el cual finaliza la sincronización entre los medidores y el servidor ntp. En la quinta y sexta línea se almacena un registro en un archivo .txt de la fecha y hora en donde se realizó el sincronismo.

Mediante la instrucción *batmand –cd1*en el terminal del SuperNodo podemos observar en la Figura3.1.3 los enlaces y comunicación entre los nodos de la red y el SuperNodo, a manera de ejemplo en la figura siguiente se observa los enlaces entre el SuperNodo y los medidores 33 y 34 de la Red:

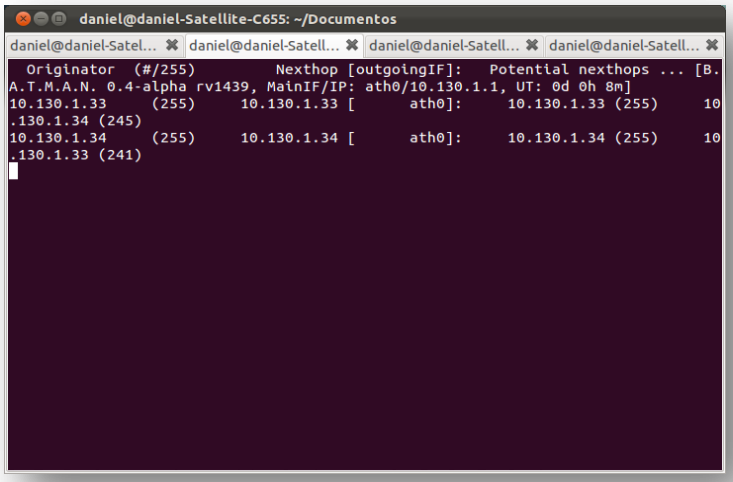


Figura 3.1.3: Visualización de los nodos medidores mediante el comando batmand.

Procedemos a hacer una prueba de enrutamiento entre los nodos de la red mediante la instrucción ping de protocolos de comunicación entre el SuperNodo y el medidor 33:

```
root@flukso:~# ping -c 4 10.130.2.33
PING 10.130.1.33 (10.130.1.33) 56(84) bytes of data.
64 bytes from 10.30.31.2: icmp_req=1 ttl=64 time=2.08 ms
64 bytes from 10.30.31.2: icmp_req=2 ttl=64 time=1.73 ms
64 bytes from 10.30.31.2: icmp_req=3 ttl=64 time=1.89 ms
64 bytes from 10.30.31.2: icmp_req=4 ttl=64 time=1.74 ms
--- 10.130.1.33 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.733/1.865/2.081/0.146 ms
```

Figura 3.1.4: Prueba ping desde el SuperNodo hasta el medidor con la dirección IP 10.130.2.33.

El sensor medidor de energía envía los datos del PIC a través del puerto UART hacia el router DIR-300. Ejecutando el comando *cat /dev/ttyS0*desde la consola de OpenWRT podemos visualizar los datos que el sensor está enviando. Estos datos son procesados por el router y enviados por medio inalámbrico WIFI hasta el servidor.

```
root@flukso:/# cat /dev/ttyS0
 pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
 pls 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000500
 pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
 pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
 pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
 pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
 pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
^C
 root@flukso:/#
```

Figura 3.1.5: Ejecución del comando cat en el fichero ttyS0 para ver la impresión de datos de potencia y energía.

En la Figura 3.1.5 se puede apreciar los datos ejecutados mediante el comando cat en el fichero *ttyS0*, estos datos enviados por el medidor de energía son en primer lugar transportado al servidor del ordenador mediante el SuperNodo y posterior se visualizan como se explico en el capitulo anterior mediante gráficos generados por RRDTOOLS en SPUD, así como se observa en la figura siguiente:

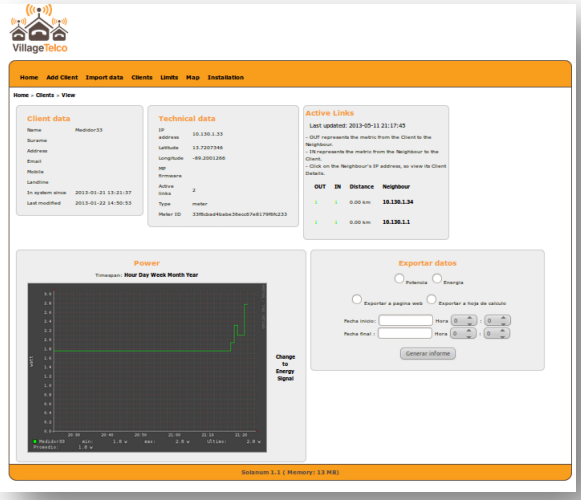


Figura 3.1.6: Página donde se presenta la grafica de potencia en SPUD.

En la Figura 3.1.6 se observa la nueva opción presente e implementada en este trabajo de investigación. Este consiste en la exportación de los datos presentes y observados en el grafico de potencia o energía respecto al tiempo generados por RRDTOOLS. Esta opción permite exportar los datos a un nueva viñeta en pagina web o exportarlo a un archivo de una hoja de datos como se muestra a continuación, la forma de representar los datos es mediante un dato en promedio por minuto almacenado, es decir el PIC envía cada segundo un dato de potencia mediante el puerto UART transcurrido un minuto de tiempo se extrae mediante la función XPORT de rrdtools un dato en promedio de los 60 almacenados en un minuto, este dato se muestra y es almacenado en un archivo xml como se menciono anteriormente.

Figura 3.1.7: Utilización del Script que ingresa los parámetros necesarios para la fecha y hora de los datos almacenados en los archivos .xml.

En la Figura 3.1.7 se observa la forma de ingresar los parámetros necesarios para extraer y exportar los datos, para tener un mayor control de los datos. Se trabaja mediante rangos de tiempo, es decir si se desea los datos o tendencia en un día u hora en especifico de los datos almacenados, se envía una fecha de inicio y la hora deseada, así como también una fecha final y hora final, estos parámetros se envían mediante un script implementado en php para facilitar el manejo y envió de los parámetros necesarios.

Figura 3.1.8: Datos exportados a una Viñeta en formato de página web.

En la Figura 3.1.8 se muestran una porción de datos almacenados y exportados a una viñeta de pagina web, en este caso son datos de potencia representados en notación científica los que se muestran.

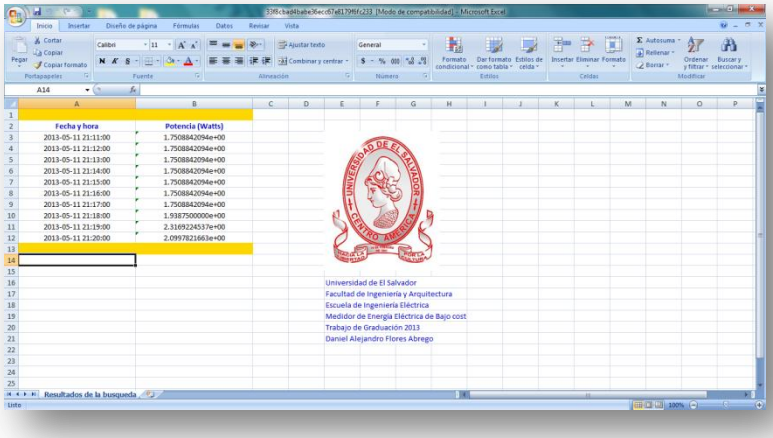


Figura 3.1.9: Datos exportados a una Hoja de datos.

En la Figura 3.1.9 se muestran una porción de datos almacenados y exportados a una hoja de datos, en este caso son datos de potencia representados en notación científica los que se muestran. Cabe mencionar que los datos a exportar pueden ser de potencia o energía generados en un rango de tiempo, y según el ID del medidor que está siendo monitorizado.

3.2Comparación de las mediciones tomadas con el medidor Shark y medidor diseñado en EIE.

En el mercado existen una gran variedad de medidores de potencia, energía y parámetros eléctricos. La mayoría de ellos poseen una gran exactitud en la toma de datos obtenidos, esto debido a que toman muestras de corriente y voltaje de la red eléctrica. En nuestro caso el medidor solo toma muestras de la magnitud de la corriente de la red eléctrica, fijando en una variable constante el voltaje nominal de la red en el microcontrolador PIC16F876A. Para poder obtener un margen de error en cuanto a las mediciones reales de una red y las mediciones dadas por nuestro medidor hecho en la EIE, se realizaron pruebas y comparaciones de nuestro medidor con uno de los medidores SHARK conectados a la red eléctrica de la UES.

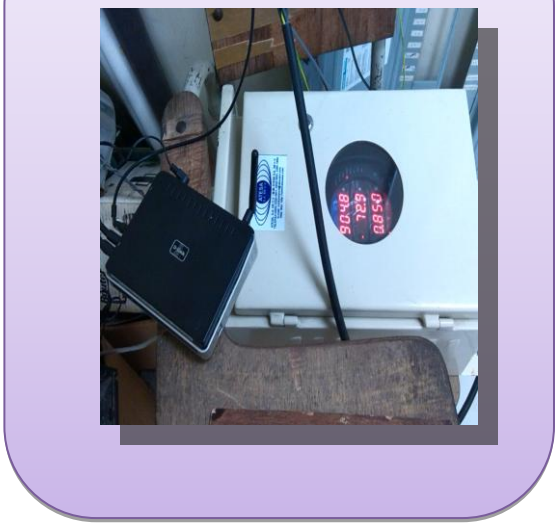
3.2.a) Medidor de Potencia SHARK

3.2.b) Medidor de Potencia Diseñado en la Escuela de Ingeniería Eléctrica.

Figura 3.2.1:a) Medidor de Potencia SHARK y b) Medidor de Potencia Diseñado con la Escuela de Ingeniería Eléctrica.

En la Figura 3.2.1 se observa el equipo utilizado para realizar mediciones con carga controlada utilizando los sub-tableros trifásicos ubicados en los laboratorios de Electrónica y Telecomunicaciones de la EIE-UES.

Conexión de los equipos utilizados a un tablero eléctrico



Medidor Dir-300 y Medidor Shark Analizando la red eléctrica.

Figura 3.2.2: a) Conexión de los equipos utilizados a un tablero eléctrico y b) Medidor Dir-300 y Medidor Shark Analizando la red eléctrica.

La Figura 3.2.2 Muestra la conexión que se realizo de los medidores y sus transductores de corriente y/o Voltaje a la red eléctrica en un sub tablero trifásico de la EIE. Ambos equipos estuvieron censando la carga demandada durante una semana en los laboratorios de la EIE y se hizo una comparación grafica de ambos equipos. Para el caso de los medidores manufacturados en la EIE se compararon los datos obtenidos por el medidor #33 y Medidor # 35 con los datos obtenidos con el medidor Shark propiedad del campus Universitario, dando como resultado lo siguiente:

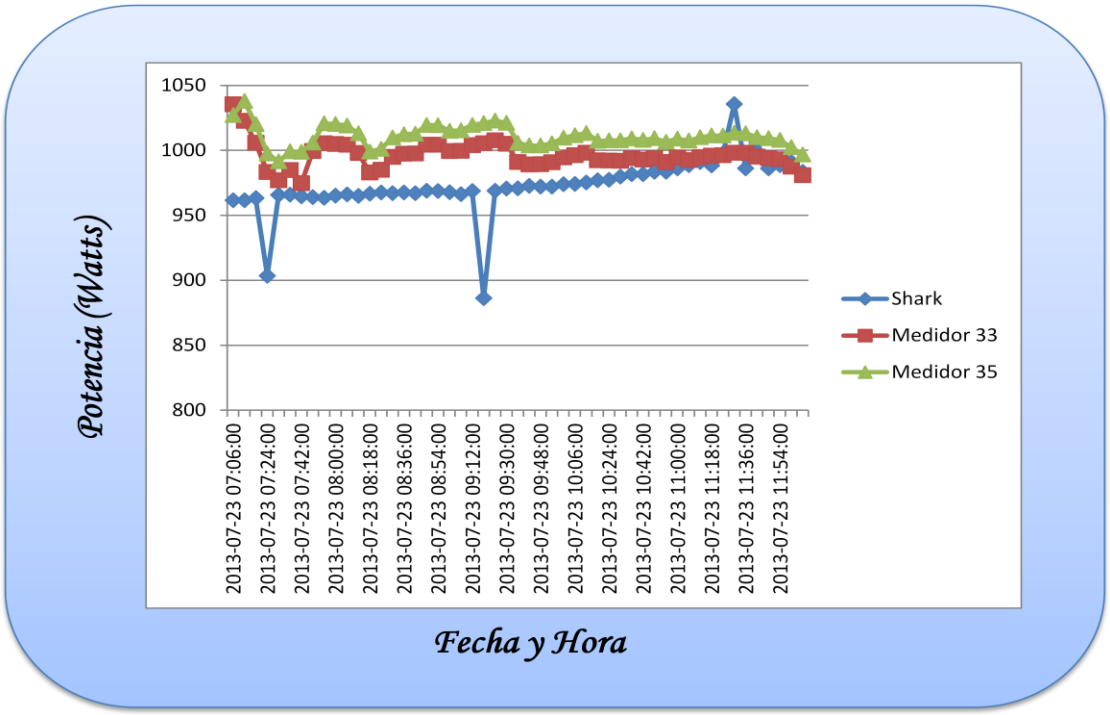


Figura 3.2.3: Gráficos comparativos de los datos de potencia trifásica obtenidos mediante los transductores de Corriente y/o voltaje en la red Eléctrica de la EIE.

En la Figura 3.2.3 se puede observar la tendencia del consumo de potencia obtenida en la red eléctrica. El Gráfico de color azul muestra los datos obtenidos por el medidor Shark y es la referencia base de la potencia trifásica consumida. Las líneas color verde y marrón muestran las gráficas de la potencia consumida través de cada una de las pinzas de corriente de los medidores 33 y 35 manufacturados en este trabajo de graduación, con esta gráfica se puede apreciar que cada pinza entrega un valor muy aproximado al valor ideal que el fabricante propone con una resolución de 1AAC/100mVDC.

Medidor 33			Medidor 35		Medidor SHARK
Fecha y hora	Potencia (Watts)	Error % Medidor 33	Potencia (Watts)	Error % Medidor 35	Potencia (Watts)
2013-07-23 07:06:00	1.04E+03	-7.67%	1.03E+03	-6.82%	961.6686401
2013-07-23 07:12:00	1.02E+03	-6.35%	1.04E+03	-7.95%	961.6010132
2013-07-23 07:18:00	1.01E+03	-4.43%	1.02E+03	-5.92%	963.2318726
2013-07-23 07:24:00	9.83E+02	-8.84%	9.98E+02	-10.42%	903.4683838
2013-07-23 07:30:00	9.77E+02	-1.16%	9.92E+02	-2.67%	965.7260132
2013-07-23 07:36:00	9.85E+02	-1.97%	9.99E+02	-3.45%	965.8722534
2013-07-23 07:42:00	9.75E+02	-1.05%	9.99E+02	-3.56%	964.6417847
2013-07-23 07:48:00	9.99E+02	-3.68%	1.01E+03	-4.35%	964.048645
2013-07-23 07:54:00	1.01E+03	-4.35%	1.02E+03	-5.94%	963.5164185
2013-07-23 08:00:00	1.00E+03	-4.11%	1.02E+03	-5.70%	965.237793
2013-07-23 08:06:00	1.00E+03	-3.96%	1.02E+03	-5.50%	966.0668335
2013-07-23 08:12:00	9.98E+02	-3.41%	1.01E+03	-4.99%	964.9530029
2013-07-23 08:18:00	9.83E+02	-1.69%	9.99E+02	-3.35%	966.7127075
2013-07-23 08:24:00	9.85E+02	-1.80%	1.00E+03	-3.47%	967.6212769
2013-07-23 08:30:00	9.95E+02	-2.88%	1.01E+03	-4.44%	967.1351318
2013-07-23 08:36:00	9.97E+02	-3.08%	1.01E+03	-4.63%	967.6317139
2013-07-23 08:42:00	9.98E+02	-3.17%	1.01E+03	-4.73%	967.0360107
2013-07-23 08:48:00	1.00E+03	-3.66%	1.02E+03	-5.22%	968.9368896
2013-07-23 08:54:00	1.00E+03	-3.69%	1.02E+03	-5.23%	968.6594849
2013-07-23 09:00:00	1.00E+03	-3.27%	1.02E+03	-4.87%	967.8612671
2013-07-23 09:06:00	1.00E+03	-3.45%	1.02E+03	-5.08%	966.3619995

Tabla 3.2.1: Tabla comparativa de Error entre los Medidores 33 y 35 con el Medidor Shark.

En la Tabla 3.2.1 se observa que con los datos obtenidos se calculó el error de la potencia registrada por un medidor, este error varía desde un 3% hasta un 9% del dato real, como el objetivo es mostrar de forma gráfica una tendencia del consumo energético, el porcentaje de error se considera aceptable.

El % de error se calculo tomando como referencia el dato obtenido por el medidor Shark.

$$\%\ de\ Error_{medidor-EIE} = \left(\frac{Medidor\ Shark - \ Medidor\ EIE}{Medidor\ Shark}\right) * 100$$

Ecuación 3.2.1: Ecuación cálculo del % de Error.

Capitulo 4:

Conclusiones y Líneas a Futuro.

1) En la configuración de la red mesh no se ha considerado la necesidad de tener acceso a internet encada medidor. Para lograr una red mesh con más prestaciones, se puede pensar en configurar los nodos medidores para que se pueda dotar a la red y que estos puedan ofrecer un servicio de internet.

2) El uso de los router para la transmisión de los datos de forma inalámbrica es de gran versatilidad, con esto se da paso a investigaciones con distintos circuitos sensores, por ejemplo podrían diseñarse equipos que midan temperatura, niveles de humedad en el ambiente o cualquier otro instrumento que mida otra variable de interés al mismo tiempo que este pueda estar analizando los parámetros eléctricos de la red.

3) Dado que el router Dir-300 ha sido descontinuado por D'Link es necesario que se experimente usando otro tipo de router o dispositivo por ejemplo el router Dragino, Raspberry Pi o Arduino para la implementación de Medidores de Potencia y Energía.

4) La incorporación de una memoria propia en el router para el almacenamiento interno de los datos que son adquiridos por los transductores de corriente del circuito medidor cuando se estén analizando los parámetros eléctricos de la red eléctrica.

5) Reducción del margen de error en la adquisición de datos, mediante la manipulación del dato de voltaje utilizado constantemente en el microcontrolador, esto se puede lograr utilizando los pines ADC del microcontrolador que no son usados en estos momentos y con ayuda de un transductor de voltaje, asi como también mediante la comparación de un voltaje promedio que se tenga en la red al momento de estar analizando sus parámetros mediante instrucciones comparativas en el microcontrolador.

ANEXO A:

Importación de los Elementos.

Todos elementos utilizados y necesarios para la manufacturación de los medidores de energía fueron importados al país mediante tres proveedores online. Todos los elementos de soldadura superficial fueron suministrados por la empresa DIGIKEY, los impresos PCB de los sensores y los conectores fueron manufacturados e importados por FUTURLEC y los Transductores de Corriente (Pinzas) fueron importados por FLUKSO.

Elemento	Dirección Online de compra
PIC-16F876A	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=PIC16F876A-I/SO-ND
XT-20Mhz	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=478-4363-1-ND
LED-rojo	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=160-1167-1-ND
C-100nF (0.1uF)	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=490-1775-1-ND
C-12pF	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=478-1469-1-ND
R-10KOhm	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=RMCF1206JT10K0CT-ND
R-330Ohm	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=RMCF1206FT330RDKR-ND
R-120Ohm	http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=RMCF1206FT120RCT-ND
PCB	http://futurlec.com/PCBService.shtm
Bases Macho 40 Pines	http://futurlec.com/Connectors/HEADS40pr.shtml
Bases Hembra 40 Pines	http://futurlec.com/Connectors/FHEADS40pr.shtml
Pinzas 50A	https://www.flukso.net/shop

Tabla A.1: Enlaces a las tiendas en línea de los proveedores.

ANEXO B:

Utilización del fichero *.profile* en distribuciones Linux.

En los sistemas LINUX/UNIX existe un fichero de configuración de nuestro perfil que es el ***.profile***. Éste fichero siempre se encuentra en el directorio **HOME** de cada usuario. Se carga cada vez que nos ingresamos de la siguiente manera en la consola: ***su - nombre_usuario*** La otra forma de ingresar es: ***su nombre_usuario*** De ésta manera estaremos cambiando de usuario pero sin cargar el fichero **.profile**. El archivo ***.profile*** permite realizar tareas que deseemos hacer cada vez que ingresemos a nuestro sistema, en nuestro caso OpenWrt y puede resultar extremadamente útil para realizar diversas actividades mediante la implementación de comandos necesarios en Linux/UNIX.

Una vez hecha dada esta breve introducción, se explicará cómo se edita el fichero de perfil de usuario y veremos la configuración del sincronismo entre nuestros medidores y el servidor ntp.

Para ver el ***.profile*** de un determinado usuario, nos situamos en su directorio home, es decir, aquel en el que se encuentra el usuario nada más al ingresar al sistema, y una vez allí, ejecutaremos en la consola el siguiente comando: ***ls -lrta | grep .profile.*** De ésta manera se mostrará un listado largo, apareciendo por orden de abajo a arriba los ficheros ordenados por fecha (el más reciente se encuentra abajo de todo) que contengan ***".profile"***.

En caso no exista el fichero ***".profile"***, se puede crear y editar mediante el comando ***vim .profile.*** Al ejecutar el comando anterior se nos desplegara un editor de texto, en el cual ingresaremos las siguientes instrucciones para la configuración y sincronismo entre nuestros medidores y el servidor NTP.

```
Echo “Actualizando Fecha” > log.txt
date >> log.txt
sleep 5m
ntpclient –s –h 10.30.1.10
echo “Su fecha fue Actualizada” >> log.txt
date >> log.txt
```

Tabla B.1: Instrucción de configuración y sincronización entre los Router Medidores y Servidor NTP.

Luego de ingresar y editar las instrucciones dadas en la Tabla B.1, se procede a guarda y reiniciar los servicios de nuestro sistema OpenWrt con el comando *wq* y *reboot.*

ANEXO C:

Presupuesto

El siguiente presupuesto incluye la comparación de precios de materiales entre los tres prototipos realizados en la EIE para 10 medidores de Energía.

	2011			2012			2013		
ELEMENTO	CANTIDAD	PRECIO	PRECIO TOTAL	CANTIDAD	PRECIO	PRECIO TOTAL	CANTIDAD	PRECIO	PRECIO TOTAL
PC DOBLE CARA	10	$ 10.23	$ 102.30	10	$ 10.23	$ 102.30	10	$ 7.40	$ 74.00
ROUTER DIR 300 RESTAURADOS	10	$ 25.00	$ 250.00	10	$ 15.00	$ 150.00	10	$ 15.00	$ 150.00
REFERENCIA 5V REF-195	10	$ 6.45	$ 64.50	10	$ 3.39	$ 33.90	0	$ -	$ -
MICROCONTROLADOR PIC 16F876	10	$ 8.80	$ 88.00	10	$ 6.16	$ 61.60	10	$ 6.16	$ 61.60
RELOJ DE TIEMPO REAL DS-1307	10	$ 5.32	$ 53.20	10	$ 3.35	$ 33.50	0	$ -	$ -
BATERIA CON BASE 3V	10	$ 2.76	$ 27.60	10	$ 1.93	$ 19.30	0	$ -	$ -
CRISTAL 20MHZ	10	$ 2.18	$ 21.80	10	$ 1.58	$ 15.80	10	$ 1.58	$ 15.80
CRISTAL 32.768 KHZ	10	$ 0.60	$ 6.00	10	$ 0.42	$ 4.20	0	$ -	$ -
CAPACITOR 10 uf	10	$ 0.80	$ 8.00	10	$ 0.44	$ 4.40	0	$ -	$ -
CAPACITOR 220 uF	10	$ 0.80	$ 8.00	10	$ 0.82	$ 8.20	0	$ -	$ -
CAPACITOR 12 pF	20	$ 0.60	$ 12.00	20	$ 0.23	$ 4.66	20	$ 0.23	$ 4.66
CAPACITOR 0.1 uf	40	$ 0.16	$ 6.40	30	$ 0.23	$ 6.75	30	$ 0.23	$ 6.75
RESISTENCIA 120 Ω	20	$ 0.07	$ 1.40	30	$ 0.04	$ 1.32	30	$ 0.04	$ 1.32
RESISTENCIA 330 Ω	20	$ 0.40	$ 8.00	30	$ 0.10	$ 3.00	10	$ 0.10	$ 1.00
RESISTENCIA 5.1 KΩ	20	$ 0.05	$ 1.00	20	$ 0.06	$ 1.20	0	$ -	$ -
RESISTENCIA 100 Ω	10	$ 0.05	$ 0.50	10	$ 0.06	$ 0.60	0	$ -	$ -
RESISTENCIA 10 KΩ	10	$ 0.05	$ 0.50	10	$ 0.02	$ 0.22	10	$ 0.02	$ 0.22
DIODO LED COLOR ROJO	20	$ 0.39	$ 7.80	30	$ 0.32	$ 9.45	10	$ 0.32	$ 3.15
BASES HEMBRA 40 PINES	4	$ 0.45	$ 1.80	4	$ 0.45	$ 1.80	4	$ 0.45	$ 1.80
BASE MACHO 40 PINES	2	$ 0.95	$ 1.90	2	$ 0.95	$ 1.90	2	$ 0.95	$ 1.90
PINZAS 50 A	4	$ 28.00	$ 112.00	4	$ 22.30	$ 89.20	4	$ 22.30	$ 89.20
COSTOS DE IMPORTACION DIGIKEY Y FUTURLEC	1	$ 106.25	$ 106.25	1	$ 106.25	$ 106.25	1	$ 106.25	$ 106.25
COSTOS DE IMPORTACION PINZAS 50A FLUKSO	1	$ 118.50	$ 118.50	1	$ 118.50	$ 118.50	1	$ 118.50	$ 118.50
TOTAL MEDIDORES			$ 1,007.45			$ 778.05			$ 636.15

Tabla C.1: Presupuesto para 10 medidores.

Prototipo	Medidor 2011	Medidor 2012
Medidor 2013 ($ 636.15)	$ 1,007.45	$ 778.05
Ahorro	36.86%	18.24%

Tabla C.2: Comparación del porcentaje de ahorro entre los prototipos manufacturados en la EIE.

Haciendo una comparación de los gastos realizados en la manufactura de los medidores de Energía y Potencia de la EIE como se muestra en la Tabla C.1. Se observa que los gastos realizados en cada uno de los prototipos, se ha obtenido una considerable reducción de los elementos y precios entre el prototipo del año 2011, 2012 y 2013. En la Tabla C.2 se observa el ahorro en la manufactura de 10 medidores del prototipo presentado en este trabajo de Graduación. Este ahorro es de un 36.86% en comparación del prototipo del año 2011 y un ahorro de 18.24 % con el prototipo del año 2012.

ANEXO D.

Modificaciones al código de SPUD

En este Anexo se incluyen los segmentos de código que fueron implementados para exportar los datos de los gráficos generados por rrdtools en SPUD cada uno tiene una pequeña explicación y la ruta del archivo original de SPUD.

```
<?php
/*****************************************************************
***********
 * details.ctp  - Display node details
 * version      - 1.0
 *
 * Version: MPL 1.1
 *
 * The contents of this file are subject to the Mozilla Public
License Version
 * 1.1 (the "License"); you may not use this file except in
compliance with
 * the License. You may obtain a copy of the License at
 * http://www.mozilla.org/MPL/
 *
 * Software distributed under the License is distributed on an "AS
IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See
the License
 * for the specific language governing rights and limitations
under the
 * License.
 *
 *
 * The Initial Developer of the Original Code is
 *   Louise Berthilson <louise@it46.se>
 *
 *
*****************************************************************
*********/

     echo $html->addCrumb('Clients', '/nodes');
     echo $html->addCrumb('View', '/nodes/details');


        $links = array();

        if(array_key_exists('Link', $details)){
```

```
            $links = $details['Link'];

        }

        //*** Personal data ***//
        $table1[] = array(array($html-
>div('table_sub_header',__('Client data',true)),array('colspan'=>
2)));
        $table1[] =
array(__('Name',true),$details['Node']['name']);
        $table1[] =
array(__('Surame',true),$details['Node']['surname']);
        $table1[] =
array(__('Address',true),$details['Node']['address']);
        $table1[] =
array(__('Email',true),$details['Node']['email']);
        $table1[] =
array(__('Mobile',true),$details['Node']['mobile']);
        $table1[] =
array(__('Landline',true),$details['Node']['landline']);
        $table1[] = array(__('In system
since',true),$details['Node']['created']);
        $table1[] = array(__('Last
modified',true),$details['Node']['modified']);

        //*** Technical data ***//
        $table2[] = array(array($html-
>div('table_sub_header',__('Technical
data',true)),array('colspan'=> 2)));
        $table2[] = array(__('IP
address',true),$details['Node']['ip_addr']);
        $table2[] =
array(__('Latitude',true),$details['Node']['lat']);
        $table2[] =
array(__('Longitude',true),$details['Node']['long']);
        $table2[] = array(__('MP
firmware',true),$details['Node']['firmware']);
        $table2[] = array(__('Active links',true),sizeof($links));
        $table2[] =
array(__('Type',true),$details['Node']['type']);
        if($details['Node']['type']=="meter")
           $table2[] = array(__('Meter
ID',true),$details['Node']['comment']);

        echo $html->div('frameLeft');
        echo "<table class='blue'>";
        echo $html->tableCells($table1,array('class'=>
'blue'),array('class'=> 'blue'));
echo "</table></div>";

        echo $html->div('frameLeft');
```

```
echo "<table class='blue' width='220px'>";
        echo $html->tableCells($table2,array('class'=>
'blue'),array('class'=> 'blue'));
echo "</table></div>";

        echo $html->div('frameLeft');
echo $html->div('table_sub_header',__('Active Links',true));


        //*** Links ***//
        if(sizeof($links)){

     echo $this->Html->div('info-message',__('Last
updated',true).': '.$links[0]['modified']);
     echo $this->Html->div('information',__('- OUT represents the
metric from the Client to the Neighbour.',true),array('style'
=>'width:350px;'));
     echo $this->Html->div('information',__('- IN represents the
metric from the Neighbour to the Client.',true),array('style'
=>'width:350px;'));
     echo $this->Html->div('information',__('- Click on the
Neighbour\'s IP address, so view its Client
Details.',true),array('style' =>'width:350px;'));


        $min   = $limits[0];
        $max   = $limits[1];
        $color = $limits[2];

           foreach($links as $key => $link){

                  if($link['label'] < $min) { $color_in =
$color['max'];} elseif($link['label']< $max) { $color_in =
$color['med'];} else { $color_in = $color['min'];}
                  if($link['label_reverse']< $min) { $color_out =
$color['max'];} elseif($link['label_reverse']< $max) { $color_out
= $color['med'];} else { $color_out = $color['min'];}

                  $in  = $html->para(false,$link['label'],
array('style' => 'color:#'.$color_in.';'));
                  $out = $html->para(false,$link['label_reverse'],
array('style' => 'color:#'.$color_out.';'));
            $distance = $this->Number-
>precision($link['distance'],2)." km";

                  $user_link  = $html->link($link['neighbour'],
array('controller' => 'nodes', 'action' => 'details',
$link['neighbour_id'] ), array('title' => 'Neighbour details',
'onclick' => "Modalbox.show(this.href, {title: this.title, width:
950}); return false;"),null,false,false);
                  $table3[] = array($in, $out, $distance,
```

```
$user_link);
           }

        echo "<table class='blue'>";
        echo $html->tableHeaders(array(
__("OUT",true),__("IN",true),__("Distance",true),__("Neighbour",tr
ue) ));
        echo $html->tableCells($table3,array('class'=>
'blue'),array('class'=> 'blue'));
        echo "</table>";

        } else {

        echo $html->div('info-message',__("This client has no
active links",true));

        }
     echo "</div>";
     if($details['Node']['type']=="meter"){
          $rnd=rand(1,10000);
     $tg=$_GET['tg'];
          $timespan=$_GET['timespan'];
          $meterid=$details['Node']['comment'];
          $metername=$details['Node']['name'];
          $nodeid=$details['Node']['id'];
          if($tg=="Energy"){
               system("/usr/lib/cgi-bin/graficarE.sh $meterid.rrd
$meterid.png $timespan $metername");
               $ntg="Power";
               }
          if($tg=="Power"){
               system("/usr/lib/cgi-bin/graficar.sh $meterid.rrd
$meterid.png $timespan $metername");
               $ntg="Energy";
               }
          echo
"<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><
br><br><br><br>";
          echo "<div class='frameLeft'><table class='blue'
width='550px'>";
          echo "<tr class='blue'><td align='center'><div
class='table_sub_header'>$tg</div></td><tr>";
          echo "<tr class='blue'><td align='center'>Timespan:
<b><a href='./$nodeid?tg=$tg&timespan=h'>Hour </a><a
href='./$nodeid?tg=$tg&timespan=d'>Day</a><a
href='./$nodeid?tg=$tg&timespan=w'> Week</a><a
href='./$nodeid?tg=$tg&timespan=m'>Month</a><a
href='./$nodeid?tg=$tg&timespan=y'>Year</a></b></td><tr>";
          echo "<tr class='blue'><td align='center'><img
src='/powermeter/graficas_power/$meterid.png?$rnd'></td>";
          echo "<td align='center'><b><a
```

```
href='./$nodeid?tg=$ntg&timespan=$timespan'>Change to $ntg
Signal</b></tr>";
          echo "</table></div>";

     //#########################################################
##################################################
          //echo "<br><br><br><br><b>MEDIDOR ID = </b> $meterid
<br><br><br><br><br> ";
          /*
          echo "
          <link rel=\"stylesheet\" href=\"/libs/jquery-ui.css\" />
          <script src=\"/libs/jquery-1.js\"></script>
          <script src=\"/libs/jquery-ui.js\"></script>
          <!-- <link rel=\"stylesheet\"
href=\"/resources/demos/style.css\" /> -->

          <script>
               $(function(){
                    $( \"#datepicker\" ).datepicker({
                       dateFormat: \"yy-mm-dd\",
                       changeMonth: true,
                       changeYear: true,
                       autoSize: true
                    });
               });
          </script>
          <script>
               $(function(){
                    $( \"#datepicker2\" ).datepicker({
                       dateFormat: \"yy-mm-dd\",
                       changeMonth: true,
                       changeYear: true,
                       autoSize: true
                    });
               });
          </script>
          ";
          */
          include_once "mis_funciones.php";

          echo "<div class='frameLeft'><table class='blue'
width='550px'>";
          echo "<tr class='blue'><td align='center'><div
class='table_sub_header'>Exportar datos</div></td><tr>";
          echo "<tr class='blue'><td align='center'>";
          formulario ("../../../../busqueda.php", $meterid);

          echo "</td><tr>";
          echo "</table></div>";

     }
```

```
?>
```

En la figura anterior se muestra el código fuente del archivo details.ctp, este archivo es el encargado de dar los parámetros del Medidor ID (meterid) en spud a los script que así lo soliciten, genera una clase que se encarga de sincronizar SPUD y los script, se encuentra ubicado en /var/www/spud/app/views/nodes.

```
<?php

function formulario ($archivo, $meterid) {
          echo "
          <link rel=\"stylesheet\" href=\"/libs/jquery-ui.css\"
/>
          <script src=\"/libs/jquery-1.js\"></script>
          <script src=\"/libs/jquery-ui.js\"></script>
          <!-- <link rel=\"stylesheet\"
href=\"/resources/demos/style.css\" /> -->

          <script>
               $(function(){
                    $( \"#datepicker\" ).datepicker({
                       dateFormat: \"yy-mm-dd\",
                       changeMonth: true,
                       changeYear: true,
                       autoSize: true
                    });
               });
          </script>
          <script>
               $(function(){
                    $( \"#datepicker2\" ).datepicker({
                       dateFormat: \"yy-mm-dd\",
                       changeMonth: true,
                       changeYear: true,
                       autoSize: true
                    });
               });
          </script>
          ";

     echo "<form name=prueba action=".$archivo." method=GET
target=\"_blank\"> \n";

     echo "
               <input type=radio name=tipo_dato value=uno
```

```
/><label>Potencia</label> \n
                <input type=radio name=tipo_dato value=dos
/><label>Energia</label> \n
                <br><br> \n
            ";

     echo "
                <input type=hidden name=medidor_ID value=$meterid
/>
                <input type=radio name=seleccion value=uno
/><label>Exportar a pagina web</label> \n
                <input type=radio name=seleccion value=dos
/><label>Exportar a hoja de calculo</label> \n
                <br><br> \n
                <label for=fecha1>Fecha inicio:</label><input
type=text id=datepicker name=fecha_1 /> \n
                <label for=hora>Hora</label> \n
                <select id=hora name=hora_1> \n
            ";

     for($i=0; $i<24; $i++){
          echo "<option value=",$i, ">", $i, "</option>";
     }

     echo "
                </select> \n
                <label for=min>:</label> \n
                <select id=min name=min_1> \n
            ";
     for($i=0; $i<60; $i++){
          echo "<option value=",$i, ">", $i, "</option>";
     }

     echo "
                </select> \n
                <br> \n
                <label for=fecha2>Fecha final : </label><input
type=text id=datepicker2 name=fecha_2 /> \n
                <label for=hora2>Hora</label> \n
                <select id=hora2 name=hora_2> \n
            ";

     for($i=0; $i<24; $i++){
          echo "<option value=",$i, ">", $i, "</option>";
     }

     echo "
                </select> \n
                <label for=min2>:</label> \n
                <select id=min2 name=min_2> \n
            ";
```

```
      for($i=0; $i<60; $i++){
            echo "<option value=",$i, ">", $i, "</option>";
      }

      echo "
                  </select> \n
                  <br><br> \n
                  <input type=submit value='Generar informe'> \n
                  </form> \n
               ";

}


?>
```

En la Figura anterior se muestra el código fuente del archivo implementado en php “mis_funciones.php”, este segmento de código es el script que genera la visualización del formulario presente en spud para el ingreso de las fecha de inicio y fin, así como también las horas deseado para exportar ese rango de datos, este script hace el llamado a unas funciones ya implícitas programadas en javascript, dichos script de java son los encargados de desplegar el calendario para poder facilitar la forma de ingreso de las fechas deseadas y así no generar confusión en la forma de enviar los parámetros, se encuentra ubicado en la dirección del ordenador /var/www/spud/app/views/nodes.

```
#!/usr/bin/env bash

#Esta parte es solo para asegurarnos que el script recibe todos
los arguementos
if [ "$#" != "3" ]; then
      echo "Uso: $0 <fecha de inicio><fecha final><medidor_ID>"
         echo
      echo "Ejemplo: $0 1353327780 1353328320
33f8cbad4babe36ecc67e8179f6fc233"
      exit
fi;
```

```
DB="/usr/lib/cgi-bin/rrd_database/$3"
OUTPUT="/var/www/xml/potencia/$3"


## en $0 se guarda el nombre del archivo sh
#echo "El script $0"
## en $1,2,3 van los argumentos con los que se llamo el script
#echo "Recibe los argumentos $1 $2 $3 "

#USO DE XPORT:
#rrdtool xport --start 1353327780 --end 1353328320
DEF:xx=33f8cbad4babe36ecc67e8179f6fc233.rrd:watt:AVERAGE
XPORT:xx:watt > salida.xml

#se busca en el archivo .rrd deacuerdo a los limites, a la salida
se recibe un xml y se guarda con el ID del medidor en www/xml

rrdtool xport --start $1 --end $2 DEF:xx=$DB.rrd:watt:AVERAGE
XPORT:xx:watt > $OUTPUT.xml
```

El código fuente que se aprecia en la figura pertenece al script implementado cuyo nombre es “búsqueda.sh” y es el encargado de hacer la búsqueda de las bases de datos almacenadas y generadas por el servidor rrdtools de los datos de potencia, cuando este genera la búsqueda y encuentra el archivo .rrd que se le fue indicado mediante el script mis_funciones.php, genera y transforma el archivo .rrd a un archivo .xml para la facilitar el manejo de los datos enviados por los medidores, esto se hace mediante la utilización de la función de rrdtools de XPORT, este script tiene una ubicación en la dirección /usr/lib/cgi-bin.

```
#!/usr/bin/env bash

#Esta parte es solo para asegurarnos que el script recibe todos
los arguementos
if [ "$#" != "3" ]; then
     echo "Uso: $0 <fecha de inicio><fecha final><medidor_ID>"
        echo
     echo "Ejemplo: $0 1353327780 1353328320
33f8cbad4babe36ecc67e8179f6fc233"
     exit
fi;
```

```
DB="/usr/lib/cgi-bin/rrd_database/Energy/$3"
OUTPUT="/var/www/xml/energia/$3"


#USO DE XPORT:
#rrdtool xport --start 1353327780 --end 1353328320
DEF:xx=33f8cbad4babe36ecc67e8179f6fc233.rrd:watt:AVERAGE
XPORT:xx:watt > salida.xml

#se busca en el archivo .rrd deacuerdo a los limites, a la salida
se recibe un xml y se guarda con el ID del medidor en www/xml

rrdtool xport --start $1 --end $2 DEF:xx=$DB.rrd:watt:AVERAGE
XPORT:xx:watt > $OUTPUT.xml
```

El código fuente que se aprecia en la figura pertenece al script implementado cuyo nombre es “búsquedaE.sh” y es el encargado de hacer la búsqueda de las bases de datos almacenadas y generadas por el servidor rrdtools de los datos de energía, cuando este genera la búsqueda y encuentra el archivo .rrd que se le fue indicado mediante el script mis_funciones.php, genera y transforma el archivo .rrd a un archivo .xml para la facilitar el manejo de los datos enviados por los medidores, esto se hace mediante la utilización de la función de rrdtools de XPORT, este script tiene una ubicación en la dirección /usr/lib/cgi-bin.

```
<?php

include "to_excel.php";
include "to_web.php";

$seleccion = $_GET['seleccion'];
$tipo_dato = $_GET['tipo_dato'];

// OBTENGO LOS DATOS DE LA FECHA DEL LIMETE INFERIOR
$fecha_1 = $_GET['fecha_1'];
$hora_1= $_GET['hora_1'];
$min_1 = $_GET['min_1'];

// OBTENGO LOS DATOS DE LA FECHA DEL LIMETE SUPERIOR
$fecha_2 = $_GET['fecha_2'];
$hora_2 = $_GET['hora_2'];
$min_2 = $_GET['min_2'];

// MEDIDOR ID
```

```
$medidor_ID = $_GET['medidor_ID'];
//$medidor_ID = "33f8cbad4babe36ecc67e8179f6fc233";
//$medidor_ID = $medidor_ID.".rrd";

// Uno fecha y hora en formato humado y lo transformo a tiempo
epoch para el limite inferior
$limite_i = $fecha_1." ".$hora_1.":".$min_1.":00";
$limite_i = strtotime($limite_i);

// Uno fecha y hora en formato humado y lo transformo a tiempo
epoch para el limite superior
$limite_s = $fecha_2." ".$hora_2.":".$min_2.":00";
$limite_s = strtotime($limite_s);

// LEER LOS .RRD Y GENERAL XML CON busqueda.sh

if($tipo_dato == "uno") {
      #Genero el XML para datos de potencia
      $command =  "/usr/lib/cgi-bin/busqueda.sh ".$limite_i."
".$limite_s." ".$medidor_ID;
}
else {
      #Genero el XML para datos de energÃa
      $command =  "/usr/lib/cgi-bin/busquedaE.sh ".$limite_i."
".$limite_s." ".$medidor_ID;
}

system($command);

if ($seleccion == "uno" ){
      #Exporto los datos a una pagina web.
      to_web($medidor_ID, $tipo_dato);
}
else {
      #Exporto los datos a una hoja de calculo.
      to_excel($medidor_ID, $tipo_dato);
}

?>
```

El código fuente que se aprecia en la figura pertenece al script implementado “búsqueda.php”, este script recibe los parámetros detallados en mis_funciones.php es decir recibe los parámetros de fecha y hora asignada mediante los script de java mencionados anteriormente, con estos atributos el indica el script a utilizar o activar, este puede ser la presentación de los datos de

potencia o energía, mostrarlos en una viñeta de pagina web o una hoja de datos, se encuentra en la dirección /usr/lib/cgi-bin.

```
table {
      margin: 2em 0;
      text-align:center;
      padding: 10px;
      spacing: 2px;
      width: 460px;
      border: #FF9900 solid 2px;
      #background: #ffffaa;
      background: #192d40;
}

.td1 {
      #background: #ffffaa;
      background: #192d40;
}

.td2 {
      background: #696969;
}

* { margin:0; padding:0; }
body, div, span, p, a, img, ul, ol, li, hr, form, fieldset,
legend, dl, dt, dd, blockquote, applet, object { border:0; }
body {
  padding: 0 0;
  background: #192d40 url('../libs/img/bg.jpg') repeat-x;
  font-family: "arial",sans-serif;
  font-size: 13px;
  line-height:24px;
  color: #e8eaeb;
  text-align: center;
}

/* links ----------------------------------------------------------
------ */
a {color:#fff;}
a:link {color:#fff;}
a:hover {color:#fff; text-decoration: none;}

/* headings -------------------------------------------------------
--------------- */
h1, h2, h3, h4, h5, h6 { margin:15px 0 10px 0; }
h1 { font-size:200%; }
h2 { font-size:160%; font-family: 'Trebuchet MS', 'Geneva CE',
```

```
lucida, sans-serif; font-weight: normal; }
h3 { font-size:120%; margin-top: 25px; font-family: 'Trebuchet
MS', 'Geneva CE', lucida, sans-serif;}
h4 { font-size:120%; }
h5 { font-size:100%; }

/* layout ------------------------------------------------------
------------- */
#layout {
  width:790px;
  margin:0 auto;
  text-align:left;
}
  #container {
  padding-top: 12px;
  }

     /* header ------------------------------------------------
--------------- */
     #header { position:relative; padding-left: 7px;}

      #logo {
        font-family: Impact, 'Techno CE', sans-serif;
        font-weight: normal;
        color: #fff;
        position: relative;
        overflow: hidden;
        height: 110px;
        font-size:260%;
        margin: 0;
        line-height: 130px;
      }
      #logo a{
        color: #fff;
        text-decoration: none;
      }

      #logo .leaf {
        position: absolute;
        left: 0; top: 20px;
        z-index: 1;
        width: 120px; height: 80px;
        background: url('../libs/img/leaf.gif') 45px 5px no-
repeat;
cursor: pointer;
      }

      #logo .light {
        color: #3f576e;
}
```

```
    /* navigation ----------------------------------------------
--- */
    #nav {position:relative; z-index:2; border: 1px solid #1a2735;
border-right: none; border-left: none; padding: 7px 0;}
      #nav ul {margin:0; padding:0; list-style:none;}
     #nav ul li {float:left; display:inline; margin:0; padding:0;}
     #nav ul li a {
       font-size: 19px;
       font-weight: bold;
          display: block;
          color: #fff;
          text-decoration: none;
          float: left; /*\*/ float:none;
          padding: 7px 25px 7px 7px;
          line-height: 18px;
        }

        #nav ul li a span {
       font-size: 11px;
       font-weight: normal;
       color: #67707a;
        }

        #nav ul li a:hover {
          background: #0a151f;
        }
     #nav ul li#active a {
        }



    .content {
      float: right;
      padding: 0 0 0 0;
      width:570px;
      z-index: 10;
    }
    .content .in{
      padding: 0 0 0 10px;
    }

     /* columns -------------------------------------------------
--------------- */
     #panel-left { float:left; width:200px; }
     .panel .in { padding: 10px; }

    #panel-left span  { color: #515f6a; }

    #panel-left p  { margin-top: 5px; }

     /* footer --------------------------------------------------
```

```
-------------- */
     #footer {
      clear:both;
      height: 77px;
      margin-top: 20px;
      border-top: 1px solid #324455;
      padding:10px 3px;
      color: #a0a9b0;
      font-size: 90%;
    }

    #footer a {
      color: #a0a9b0;
    }

      #footer p { margin:0; padding-top:10px; }



/* paragraphs ----------------------------------------------------
---------------- */
p { margin:15px 0; }

/* lists ---------------------------------------------------------
------------ */
ul, ol { display:block; margin:15px 0 15px 40px; }
ul ul, ul ul ul, ol ol, ol ol ol { margin:0; margin-left:20px; }
ol { list-style-type:decimal; }
ol ol { list-style-type:upper-alpha; }
ol ol ol {list-style-type:lower-alpha; }
li { display:list-item; }
ul li a { text-decoration:underline; }

/* definitions ---------------------------------------------------
----------------- */
dl { margin:15px 0; }
dt { font-weight:bold; }
dd { margin-left:30px; }

/* universal -----------------------------------------------------
---------------- */
fieldset { margin:15px 0; padding:10px; border:1px solid #CCC; }
legend { margin-left:10px; font-size:100%; font-weight:bold;
color:#000; }
abbr, acronym, .help { border-bottom:1px dotted #CCC; cursor:help;
}
blockquote { margin:15px 20px; font-style:italic; }
del, .del { text-decoration:line-through; }
strong, .strong { font-weight:bold; }
cite, em, q, var { font-style:italic; }
```

```
code, kbd, samp  {font-family:monospace; font-size:110%; }
hr { display:block; height:1px; margin:10px 0; padding:0; border:0
solid #CCC; background:#CCC; color:#CCC;}
.f-left {float:left;}
.f-right {float:right;}
.a-left, tr.a-left td {text-align:left;}
.a-center, tr.a-center td {text-align:center;}
.a-right, tr.a-right td {text-align:right;}
.a-justify {text-align:justify;}
.va-top {vertical-align:top;}
.va-middle {vertical-align:middle;}
.va-bottom {vertical-align:bottom;}
.clear { clear:both; }
.box { min-height:1px; }
.box:after { display:block; visibility:hidden; clear:both; line-
height:0; font-size:0; content:"."; }
.noscreen { display:none; }

.foto{  display: block;}
```

El código fuente que se aprecia en la figura pertenece al script cuyo nombre es “estilos.css”, esta implementado en un lenguaje de estilo que define la presentación de los documentos HTML. Abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas. Este script se usa para dar formato al contenido previamente estructurado para la creación de la viñeta encargada de representar los datos que son extraídos de los gráficos de rrdtools, tiene ubicación en la dirección /var/www/libs.

```
<?php

function to_excel ($medidor_ID, $tipo_dato) {

     // LECTURA DEL ARCHIVO XML GENERADO CON LA BUSQUEDA
     // creamos un array con la funcion simpleXML el cual contiene
le informacion del XML
     if($tipo_dato == "uno") {
          $datos =
simplexml_load_file('xml/potencia/'.$medidor_ID.'.xml');
          $leyenda = "Potencia (Watts)";
     }
```

```
      else {
            $datos =
simplexml_load_file('xml/energia/'.$medidor_ID.'.xml');
            $leyenda = "EnergÃa (Jules)";
      }

      // obtengo la cantidad de filas
      $cantidad_filas = count($datos->data->row);

      // EXPORTAR LOS DATOS A EXCEL

      // PHPExcel
      require_once 'Classes/PHPExcel.php';

      // PHPExcel_IOFactory
      require_once 'Classes/PHPExcel/IOFactory.php';

      // Create new PHPExcel object
      $objPHPExcel = new PHPExcel();

      // ####### MOSTRAMOS LOS DATOS
#####################################
      // aqui van los datos en concreto, hay q incluirlos en un
ciclo para que se muestren todos los datos

      $objPHPExcel->getActiveSheet()->getStyle('A1:B1')->getFill()-
>setFillType(PHPExcel_Style_Fill::FILL_SOLID)->getStartColor()-
>setARGB('FFFFD700');

      for ($i=0; $i<$cantidad_filas; $i++)
      {
            $ii = $i + 3;
            $epoch = $datos->data->row[$i]->t; $epoch =
(int)$epoch;
            $human_time = date("Y-m-d H:i:s", $epoch);
            $objPHPExcel->setActiveSheetIndex(0) // comenzar en la
primera fila y primera columna
                              ->SetCellValue('A2', 'Fecha y hora')
                        ->SetCellValue('B2', $leyenda)
                        ->setCellValue('A'.$ii, $human_time)  //
Imprimo la hora y fecha en tiempo humano
                        ->setCellValue('B'.$ii, $datos->data-
>row[$i]->v);   // Imprimo el dato de potencia o energÃa segÃºn el
caso

      }
      $final = $ii+1;
      $objPHPExcel->getActiveSheet()-
>getStyle('A'.$final.':B'.$final)->getFill()-
>setFillType(PHPExcel_Style_Fill::FILL_SOLID)->getStartColor()-
>setARGB('FFFFD700');
```

```
     //#### LOGO DE LA MINERVA ##############################
     $minerva = new PHPExcel_Worksheet_Drawing();
     $minerva->setName('Logo');
     $minerva->setDescription('Logo');
     $minerva->setPath('./libs/img/minerva_2.jpg');
     $minerva->setHeight(250);
     $minerva->setCoordinates('E3');
     $minerva->setWorksheet($objPHPExcel->getActiveSheet());

     //### PORTADA ##########################################
     $objPHPExcel->setActiveSheetIndex(0)
     ->setCellValue('E16', 'Universidad de El Salvador')
     ->setCellValue('E17', 'Facultad de IngenierÃa y
Arquitectura')
     ->setCellValue('E18', 'Escuela de IngenierÃa ElÃ©ctrica')
     ->setCellValue('E19', 'Medidor de EnergÃa ElÃ©ctrica de Bajo
costo')
     ->setCellValue('E20', 'Trabajo de GraduaciÃ³n 2013')
     ->setCellValue('E21', 'Daniel Alejandro Flores Abrego');

     // Uno las celdas de la portada para mejorar la presentaciÃ³n
     $objPHPExcel->getActiveSheet()->mergeCells("E16:H16");
     $objPHPExcel->getActiveSheet()->mergeCells("E17:H17");
     $objPHPExcel->getActiveSheet()->mergeCells("E18:H18");
     $objPHPExcel->getActiveSheet()->mergeCells("E19:H19");
     $objPHPExcel->getActiveSheet()->mergeCells("E20:H20");
     $objPHPExcel->getActiveSheet()->mergeCells("E21:H21");

     //######### setear que las celdas autoajusten el
ancho#################################

     $objPHPExcel->getActiveSheet()->getColumnDimension('A')-
>setWidth(30);
     $objPHPExcel->getActiveSheet()->getColumnDimension('B')-
>setWidth(30);

     //$objPHPExcel->getActiveSheet()->getColumnDimension('A')-
>setAutoSize(true);
     //$objPHPExcel->getActiveSheet()->getColumnDimension('B')-
>setAutoSize(true);

     // Renombrando la hoja
     $objPHPExcel->getActiveSheet()->setTitle('Resultados de la
busqueda');

     $objPHPExcel->getActiveSheet()->getStyle('A2:B2')->getFont()-
>getColor()->setARGB('FF0000CD');
     $objPHPExcel->getActiveSheet()->getStyle('E16:H21')-
>getFont()->getColor()->setARGB('FF0000CD');
```

```
     $_boldLabels = array(
          'font' => array('bold' => true,),
     );

     //seteo del encabezado de la tabla
     $styleArray = array(
          'alignment' => array('horizontal' =>
PHPExcel_Style_Alignment::HORIZONTAL_CENTER,),

          'fill' => array(
               'type' =>
PHPExcel_Style_Fill::FILL_GRADIENT_LINEAR,
               'rotation' => 90,
               'startcolor' => array('argb' => 'FF00FF00',),
               'endcolor' => array('argb' => 'FF32CD32',),
          ),
     );

     $objPHPExcel->getActiveSheet()->getStyle('A2:B'.$ii)-
>applyFromArray($styleArray);
     $objPHPExcel->getActiveSheet()->getStyle('A2:B2')-
>applyFromArray($_boldLabels);



     // Redirigir la salida hacia el navegador del cliente
(Excel5)
     header('Content-Type: application/vnd.ms-excel');
     header('Content-Disposition:
attachment;filename='.$medidor_ID.'.xls');
     header('Cache-Control: no-cache, must-revalidate');

     $objWriter = PHPExcel_IOFactory::createWriter($objPHPExcel,
'Excel5');
     $objWriter->save('php://output');
     exit;
}
```

El código fuente que se aprecia en la figura pertenece al script implementado cuyo nombre es *"to_excel.php*", este se encarga mediante funciones realizadas en programación con php, hacer una lectura de los datos almacenados en los archivos con extensión *.xml* y exportarlos o transformarlo en datos almacenados en una hoja de datos, el archivo se encuentra en la dirección */var/www/*.

```
<?php

function to_web ($medidor_ID, $tipo_dato) {

      //LECTURA DEL ARCHIVO XML GENERADO CON LA BUSQUEDA
      //creamos un array con la funcion simpleXML el cual contiene
le informacion del XML
      if($tipo_dato == "uno") {
            $datos =
simplexml_load_file('xml/potencia/'.$medidor_ID.'.xml');
            $medicion = "Potencia (Watt)";
      }
      else {
            $datos =
simplexml_load_file('xml/energia/'.$medidor_ID.'.xml');
            $medicion = "Energ&#237;a ";
      }

      //obtengo la cantidad de filas
      $cantidad_filas = count($datos->data->row);

      echo "<html>
      <head>
            <title>Resultados de la busqueda</title>
            <link rel=\"stylesheet\" type=\"text/css\"
href=\"libs/estilos.css\">
      </head>
      <body style=\"padding: 0 0; background: #192d40; font-family:
\"arial\",sans-serif; font-size: 13px; line-height:24px; color:
#e8eaeb; text-align: center;\">
      ";

      echo "

<div id=\"layout\">

<div id=\"header\">

      <h1 id=\"logo\">
                        Medidor de Energ&#237;a El&#233;ctrica de
Bajo Costo. <span class=\"leaf\"> </span>
                  </h1>
<hr class=\"noscreen\" />

<hr class=\"noscreen\" />

<div id=\"nav\" class=\"box\">
<ul>
                  <li><a href=\"#\">Medidor
ID<br/><span>$medidor_ID</span></a></li>
```

```
          </ul>
<hr class=\"noscreen\" />
</div>

<div id=\"container\" class=\"box\">

<div id=\"obsah\" class=\"content box\">
<div class=\"in\">

<h2>Medici&#243;n de $medicion</h2>
<p>
     ";
          // se muestra el resultado de la busqueda en la pagina
     echo "<table>";
     echo "<pre>";
     echo "<tr>
               <td>Tiempo</td><td>", $medicion."</td>
               </tr>
               ";
     for ($i=0; $i<$cantidad_filas; $i++){

          $epoch = $datos->data->row[$i]->t;
          $epoch = (int)$epoch;
          $human_time = date("Y-m-d H:i:s", $epoch);

          /**
          echo "tiempo= ", $human_time,
                    " ".$medicion."= ", $datos->data->row[$i]-
>v, "<br>"; **/

          if( $i % 2 == 0){$td_class="td1";}
          else{$td_class="td2";}

          echo "<tr>";
          echo "<td class=".$td_class.">", $human_time."</td>";
          echo "<td class=".$td_class.">", $datos->data->row[$i]-
>v, "</td>";
          echo "</tr>";

     }
     echo "</pre></table>";
     echo "
     </p>

</div>
</div>
          <!-- #### PANEL LATERAL IZQUIERDO
####################################### -->
<div id=\"panel-left\" class=\"box panel\">
<div class=\"in\">
```

```
<p>
                <!--
<span>03/07/2008</span><br/>
                -->
                <img src=libs/img/minerva.gif><br>
                <h6>
                <font size=1>
                     UNIVERSIDAD DE EL SALVADOR.<br>
                     FACULTAD DE INGENIERIA Y ARQUITECTURA.<br>
                     ESCUELA DE INGENIERIA ELECTRICA.
                </font>
                </h6>
</p>

</div>
</div>

</div>
<!-- ##### PIE DE PAGINA ################################### -->
<div id=\"footer\">
<span class=\"f-left\">&copy; 2013</span><span class=\"f-
right\">Daniel Alejandro Flores Abrego | Trabajo de
Graduaci&#243;n</span>
</div>
</div></div>
</body>
</html>


     ";

}
?>
```

El código fuente que se aprecia en la figura pertenece al script implementado cuyo nombre es "*to_web.php*", este se encarga mediante funciones realizadas en programación con php, hacer una lectura de los datos almacenados en los archivos con extensión .xml y exportarlos a una página web para una mejor lectura de los datos representados en los gráficos de rrdtools, el archivo se encuentra en la dirección */var/www/.*

Anexo E:

Guía para la instalación de SPUD y Configuraciones.

Para facilitar la instalación de SPUD con las modificaciones realizadas en este trabajo de graduación se implementaron tres script con comandos para el shell con una maquina Ubuntu 11.10. En el CD adjunto a este documento se encuentra un paquete que contiene el sistema SPUDinstall.tar.gz, este paquete descarga las dependencias del sistema y el servicio NTP para los router medidores. por lo que al momento de la instalación debe de existir una conexión a internet.

Los pasos para instalar SPUD modificado en una maquina con sistema operativo Ubuntu 11.10, a la cual no tiene instalada la distribución oficial de SPUD, son los siguientes:

1. Descomprimir SPUDinstall.tar.gz en cualquier directorio del sistema operativo.
2. Abrir una terminal de comandos y acceder a la ruta donde se descomprimió el directorio
SPUDinstall con la ayuda del comando cd
3. correr el siguiente script instalar con el comando sh instalar.sh
4. Introducir el password del usuario root cuando se necesite
5. Introducir el password que tendrá el usuario root de MySQL cuando este sea solicitado.
6. La contraseña por defecto para el usuario root de MySQL incluida en este paquete es 'root' si en el paso 5 se introdujo una distinta, el script conector de Cake-php con MySQL debe modificarse cuando instalar.sh haya terminado de ejecutarse. El archivo es el siguiente: /var/www/spud/app/config/database.php y el password del usuario root de MySQL debe estar escrito en las siguientes líneas de código donde para este ejemplo es '**root'**

```
var $default = array(
                    'driver' => 'mysql',
                    'persistent' => false,
                    'host' => 'localhost',
                    'login' => 'root',
                    'password' => 'root',
                    'database' => 'spud',
                    'prefix' => '',
);
```

7. El último paso debe hacerse manualmente escribiendo la siguiente sentencia en la línea de comandos: crontab -e cuando se abra el editor de textos se agrega la siguiente línea hasta el final:

```
*/5 * * * * root /usr/bin/wget -O - -q -t 1 http://localhost/spud/nodes/update >/dev/null 2>&1
```

Los pasos para modificar la versión oficial de SPUD con los cambios hechos en este proyecto en una maquina con Ubuntu 11.10 son los siguientes (SPUD previamente instalado y funcionando correctamente).

1. Descomprimir SPUDinstall.tar.gz en cualquier directorio del sistema operativo.
2. Abrir una terminal de comandos y acceder a la ruta donde se descomprimió el directorio SPUDinstall con la ayuda del comando cd.
3. correr el siguiente script parche con el comando sh parche.sh
4. Introducir el password del usuario root cuando se necesite
5. Descomprimir Modificación a SPUD 2013 en cualquier directorio del sistema operativo.
6. Abrir un terminal de comandos y acceder a la ruta donde se descomprimió el directorio Modificación a SPUD 2013 con la ayuda del comando cd.
7. Correr el script parche2 con el comando sh parche2.sh
8. introducir el password del usuario root cuando se necesite.

Referencias:

[1] “Diseño y construcción de un analizador de espectros para redes de potencia utilizando un microcomputador de bajo costo”, Reinoldo Cáceres Silva, Universidad de El Salvador. [Mayo 1990]

[2] “Diseño y Construcción de un dispositivo analizador de distorsión armónica en redes de baja tensión”, Marco Antonio Platero Saravia, Herberth Alirio Escalante Cordova, Universidad de El Salvador. [Octubre 1991]

[3] **“**Diseño e implementación de un medidor trifásico multifunción utilizando el IC ADE7754”, Daniel Antonio Cortez Franco, Douglas Alberto López Hernández, Universidad de El Salvador. [Febrero 2005]

[4] “Medidor inalámbrico de consumo de energía eléctrica de bajo costo”, Jonathan Alberto Zaldaña, Universidad de El Salvador. [Diciembre 2011]

[5]”Construcción de medidores trifásicos inalámbricos de consumo de energía”, Álvaro Antonio Calderón, Carlos Enrique Molina, Universidad de El Salvador. [Agosto 2012]

[6] http://informatica-practica.net/tutoriales/linux/profile.php

[7] Website de Flukso http://www.flukso.net[Ultima fecha de revisión: 06/08/2013]

[8] Eduardo García Breijo. Compilador C CCS y simulador PROTEUS para microcontroladores PIC.

[9] Andrés Cánovas López. Manual de usuario del compilador PCW de CCS.

[10] Hoja de datos microcontrolador microchip PIC-16F87X.

[11] Firmware Flukso V1: https://github.com/flukso/flm01

[12] “Telefonía inalámbrica y red de acceso a internet para los municipios de salcoatitán, Juayúa, Apaneca y Ataco”. Luis Alonso Colocho Susaña y Román Abad Tobías Vides. [Septiembre 2011]

[13] http://dili.villagetelco.org/index.php5?title=Main_Page. Página oficial del proyecto Dili Village Telco.

[14] Proyecto de ingeniería EIE UES: Redes Mesh y voz sobre IP. Román Abad Tobias Vides y Rony Stalyn Sánchez Morales.

[15] http://www.open-mesh.org/; protocolo de enrutamiento dinámico B.A.T.M.A.N. para redes mesh WiFi.

[16] http://book.cakephp.org/1.3/es/view/879/Comenzando-con-CakePHP. Cake PHP.

[17] http://www.dd-wrt.com/wiki/index.php/Serial_port_pinouts

[18] http://downloads.openwrt.org/backfire/10.03.1/atheros/packages/

[19] “Datalogger implementado en un medidor de potencia con Arduino y Router Dir-300”, Josué Gabriel Deras Campos, Juan Ulises Fuentes Barrera y Fredy Jonathan Quinteros Calzadia. [CONESCAPAN 2012]

[20] “Diseño de un Medidor Kilowatt hora Analógico de Bajo Costo para redes Trifilares Monofásicas”, Carlos Enrique Gómez Benítez, Universidad de El Salvador. [Abril 1996]

[21] “Aplicación de Instrumentos Virtuales en Sistemas de Instrumentación Electrónica”, Marvin Mauricio Flores García, Eduin Ruye Mendoza Maldonado, Universidad de El Salvador. [Diciembre 1996]