

## **“Utilización de la Minicomputadora Raspberry Pi para la adquisición y evaluación de datos de consumo de energía eléctrica de equipos a 220 Voltios”**

Jairo Andrés Rayo Cedillo  
José Alfonso Chimbo Cusme  
Ing. Carlos Valdivieso  
Facultad de Ingeniería en Electricidad y Computación  
Escuela Superior Politécnica del Litoral  
Campus Gustavo Galindo Km 30.5, vía Perimetral, Apartado 09-01-5863, Guayaquil, Ecuador  
jrayo@fiec.espol.edu.ec, jchimbo@fiec.espol.edu.ec, cvaldiv@fiec.espol.edu.ec

### **Resumen**

*El siguiente artículo académico describe el funcionamiento de nuestro proyecto de graduación utilizando la ética del DIY (Do it yourself), tecnología Open Source, lenguajes como Python y el sistema operativo Linux. Para llevar a cabo la implementación de este proyecto se utilizó la minicomputadora de bolsillo Raspberry Pi y la tarjeta electrónica conocida como Arduino Uno. Estos dos elementos son los encargados de analizar los datos de cualquier equipo eléctrico, para ello hubo la necesidad de establecer comunicación entre ambos. Durante el proceso de adquisición de datos, los mismos son llevados a un servidor web en tiempo real, en el cual se muestra el consumo eléctrico del equipo. De igual manera los datos son recopilados para luego ser guardados en una herramienta que administra bases de datos. Los resultados del análisis a distintos equipos eléctricos como secadoras y lavadoras industriales muestran que el consumo por lo general oscila entre 700 y 4000 vatios, mientras que un acondicionador de aire promedio consume 250 a 2000 vatios en un tiempo de análisis de 10 minutos para ambos casos.*

**Palabras Claves:** Raspberry Pi; Arduino Uno; Linux.

### **Abstract**

*The following article describes the performance of our graduation project using the DIY ethics (Do it yourself), Open Source technology, programming languages such as Python, and Linux operative system. To implement this project, we used the Raspberry Pi minicomputer and the Arduino Uno electronic board. These devices are used to analyze the power consumption of any electrical equipment, so we needed to set communication between them. During the acquisition process, data is delivered to a web server on real time, where the power consumption of the electrical device is shown. Further, data is collected and saved in a database management tool. Analysis results of electrical devices such as industrial dryers and washing machines show that power consumption vary between 700 and 4000 watts, while a domestic air conditioner needs between 250 and 2000 watts in an analysis time of 10 minutes for both cases.*

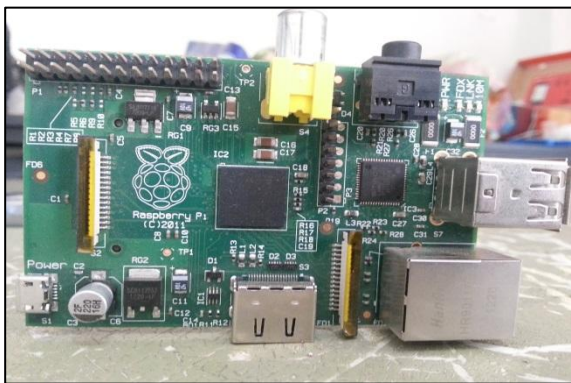
**Keywords:** Raspberry Pi; Arduino Uno; Linux.

## 1. Introducción

Durante muchos años la evolución de la electrónica y el campo de los computadores portátiles ha ido en gran aumento, siendo así que cada año aparece un nuevo dispositivo electrónico que reemplaza a otro, provocando de esta forma reducir el tiempo de vida de los dispositivos y a la vez facilitando cada vez más la vida de los seres humanos. La mayor parte de países de primer mundo son quienes fabrican este tipo de tecnologías, mientras que países como Ecuador solamente somos dependientes de ella, lo que retrasa nuestro avance tecnológico, pudiendo establecer que actualmente nos encontramos con unos treinta años de desfase con respecto a estos tipos de países de primer mundo.

Uno de los puntos más importantes dentro de este avance tecnológico ha sido el campo de los microcontroladores, que son pequeños dispositivos electrónicos los cuales realizan un sin número de funciones a través de diferentes instrucciones las cuales son almacenadas dentro de los mismos y codificadas en un tipo de lenguaje de programación.

Es el caso de la aparición de la minicomputadora Raspberry Pi en el año 2006, un gran avance tecnológico, que consiste en construir un dispositivo electrónico del tamaño de una tarjeta de crédito que posea la mayor parte de las características de un ordenador de escritorio en cuanto a puertos y periféricos, pero que tiene ciertas limitaciones de procesamiento. Dicha tarjeta o computador de bolsillo consta de puertos USB, auxiliar, HDMI, Ethernet, RGB y un slot para SD como sistema de almacenamiento. También se puede adaptar un hub USB para de esta manera expandir estos puertos y poder conectar otros dispositivos importantes como son mouse y teclado. Además consta de una interfaz gráfica amigable dependiendo del sistema operativo que se cargue en la misma. Debido a su tamaño ha tomado gran apertura muchas estudiantes utilizan la Raspberry Pi para un sin número de proyectos de bajo alcance pero de suma importancia. Podemos apreciar en la figura 1.1 la minicomputadora Raspberry Pi con cada uno de sus componentes.



**Figura 1.** Aspecto físico de la tarjeta Raspberry Pi

## 2. Componentes principales

Para efecto de este proyecto se utilizaron tres componentes de hardware principales e importantísimos como son el Raspberry Pi, el Arduino Uno y el sensor de corriente SCT-013-030, los cuales detallaremos a continuación cada una de sus características principales.

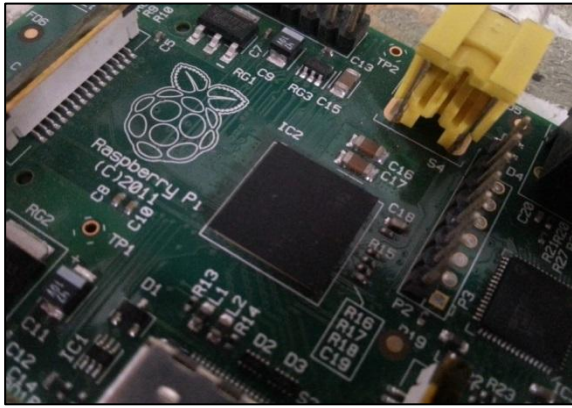
### 2.1 Raspberry Pi

La Raspberry Pi es una microcomputadora, como se muestra en la figura 1, que a diferencia de la mayoría de computadoras existentes trabaja con un sistema operativo propio de ella (Raspbian Wheezy). Posee entradas USB que generalmente son utilizadas para conectar el mouse y el teclado a la misma. Con respecto a la salida de video, el Raspberry Pi posee 2 alternativas para el usuario, la salida HDMI para televisores modernos o en su defecto la salida de video RCA para televisores analógicos. [1]

Entre sus partes físicas principales están: el procesador BMC2835, los puertos GPIO, las salidas de video, el puerto Ethernet, los puertos USB y el slot para la tarjeta de memoria.

#### Procesador BMC2835

Este procesador es considerado el corazón del Raspberry Pi debido que se encarga de los procesos del audio, video, procesamiento gráfico y comunicación de las interfaces del hardware (teclado y mouse). Todas estas tareas están repartidas en dos memorias especiales: la RAM y el GPU. La primera fue diseñada para el procesamiento de los datos, mientras que la segunda su misión es la del acelerador de video propio de la tarjeta. Un total de 256Mb\* está repartida entre las dos memorias, permitiendo al usuario seleccionar cuanta memoria desea distribuir entre ambos de acuerdo a sus necesidades, por ejemplo para la reproducción de videos en el Raspberry Pi el estándar utilizado es 192/64, es decir distribuir 192 Mb en el procesamiento de los datos y 64 Mb para el acelerador de videos. El procesador del Raspberry Pi se muestra en la figura 2. [2]



**Figura 2.** Procesador BMC2835 en la Raspberry Pi

### Salidas de Video

El Raspberry Pi está diseñado para soportar tres estándares diferentes de salida de video: RCA, HDMI, DSI. Los dos primeros son de fácil acceso al usuario ya que se encuentran en los televisores analógicos y digitales de la era actual, mientras que para el último estándar, es necesario un hardware adicional para su óptimo funcionamiento.

Para un televisor analógico es necesario utilizar la salida RCA, para un televisor digital o bien puede seguir usando esta salida o en su defecto usar la salida HDMI que se muestra en la figura 4. [3]



**Figura 3.** Salida de video RCA en la Raspberry Pi



**Figura 4.** Salida de video HDMI en la Raspberry Pi

### Conexión de mouse, teclado e Internet

EL Raspberry Pi (modelo B) posee 2 puertos USB, como se muestra en la figura 5, para la conexión de dispositivos como el teclado y el mouse, cabe recalcar que es posible adicionar un hub para alimentar a más dispositivos con salida USB para así conectarlos a la tarjeta, pero no utilice estos puertos como recarga de equipos electrónicos (como celulares) ya que reducen el tiempo de vida del Raspberry Pi. [4]

Además el modelo actual del Raspberry Pi, posee un puerto Ethernet para conectarse al internet, como se aprecia en la figura 5, pero si es la primera vez que enciende este microcomputador necesitará configurar la tarjeta para poder acceder a la red.



**Figura 5.** Conectores USB para el teclado, el mouse y puerto Ethernet para acceder a la red

### Fuente de Alimentación

Esta es una de las principales ventajas de este microcomputador, su procesador requiere un bajo consumo de energía, por lo tanto basta con un cargador que tenga para conectar un cable USB y tenga por salida un micro USB, como se muestra en la figura 6. Es importante tener en mente que el voltaje y amperaje ideal de alimentación sea de 5V y 1A, este consumo se justifica cuando se conectan varios elementos externos en la tarjeta a parte del teclado y mouse. [5]



**Figura 6.** Fuente de Alimentación de la Raspberry Pi



### Sistema Operativo Raspbian Wheezy

El Raspbian Wheezy es considerado el sistema operativo estándar del Raspberry Pi, tenga en cuenta que no solo es el único disponible en la red, pero sí el más usado por los usuarios que no poseen mucha experiencia utilizando Linux.

Este sistema operativo ocupa poco espacio en la memoria de la SD (2GB) debido a que apenas vienen instalados un mínimo de aplicaciones para el usuario. En otras palabras si el usuario necesita de algún programa extra u otra aplicación deberá instalarla manualmente usando líneas de comando en el LX Terminal del escritorio del Raspberry Pi.

### Puertos GPIO en la Raspberry Pi

El puerto de entrada/salida para propósitos generales (GPIO) son los pines que se aprecian en la figura 7, estos pines le permiten al Raspberry Pi comunicarse con otros componentes y circuitos electrónicos, además con una programación adecuada se dispone de la opción de censar temperatura, controlar servo motores y utilizando los protocolos adecuados como SPI e I2C comunicarse con otra computadora e incluso con otra Raspberry Pi. [6]



Figura 7. Localización del puerto GPIO en la Raspberry Pi

El propósito de cada uno de los pines de este puerto se muestra en la figura 8, no todos los pines se encuentran habilitados para uso del usuario, hay pines reservados, es decir que no deben ir conectados a ningún equipo externo o a la propia Raspberry Pi debido a que estos pines se encuentran conectados directamente con el procesador BMC2835 y cualquier conexión podría terminar dañando permanentemente la tarjeta. Otros de los pines tienen como función permitir el uso de los protocolos I2C, SPI, otros están habilitados para el uso del PWM, TX, RX, y un par de pines son de alimentación de 5 V y 3.3 V. [7]

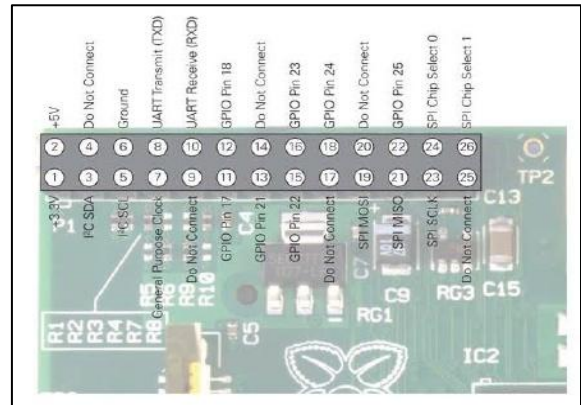


Figura 8. Designación de pines del puerto GPIO en la Raspberry Pi [8]

### 2.2 Arduino Uno

El Arduino es una placa basada en un microcontrolador Atmel AVR, que dispone de puertos de entrada y salida, además de un oscilador para trabajar a una frecuencia de reloj predeterminada de hasta 16 MHz. Para programar el Arduino se requiere utilizar el Arduino IDE (Integrated Development Environment) basado en lenguaje C. Tanto el hardware y software de Arduino son libres, es decir que los códigos, esquemático y diseño pueden ser utilizados libremente por cualquier persona. Al Arduino se pueden acoplar bases para ciertos tipos de sensores como son receptores de GPS, módulos Ethernet, LCD displays, etc., con la finalidad de añadir otras funcionalidades al Arduino. Existen diferentes modelos de Arduino, entre ellos el Arduino Uno que se puede observar en la figura 2.12, utilizado en este proyecto de graduación. Tal vez la variación más poderosa es el Arduino Mega 2560, el cual ofrece incremento de memoria y de puertos de entrada y salida. [8]



Figura 9. Apariencia física del Arduino Uno

El Arduino dispone de la facilidad de poder reemplazar el chip o microcontrolador AVR que contiene por uno nuevo en caso de que sea necesario.

### 2.3 Sensor de Corriente SCT-013-030

Es un sensor de energía eléctrica de gancho el cual puede soportar hasta 30 A de corriente como se muestra en la figura 10. Su función es sensar el campo electromagnético que atraviesa a través de un cable para luego referenciarlo en niveles de voltaje.



Figura 10. Sensor de corriente SCT-013-030

Entre sus características están:

- Resistencia interna de 64 ohmios.
- Corriente máxima de 30 A.
- Temperatura de operación entre -25 C – 70 C.
- Tensión de voltaje de salida entre 0V y 1V.

Aplicaciones: Adecuado para el monitoreo de mediciones de corriente en equipamientos de iluminación, compresores de aire, etc.

## 3. Materiales e Implementación

### 3.1. Materiales.

Detallamos los principales componentes utilizados en la implementación del proyecto:

Hardware:

- Arduino Uno.
- Raspberry Pi.
- Sensor de corriente SCT-013-030.
- Fuente de Alimentación micro-USB de 5 V con 0.7 A de salida como mínimo.
- Cable serial USB A/B utilizado para conexiones entre PC e impresora.
- Teclado y mouse USB.
- Hub-USB con alimentación propia.
- Monitor LCD como interfaz visual.

Software:

- Arduino IDE.
- Python 2.7 o superior.
- Librería Emonlib.h para Arduino Uno

- Librería py-serial y python-serial para Raspberry Pi.
- Tener una cuenta activa en Cosm.com.

Componentes adicionales:

- 1 Resistor de 33  $\Omega$ .
- 2 Resistores de 1k  $\Omega$
- Cables de conexión.

### 3.2. Implementación

Este proyecto consiste básicamente de tres partes importantes que son: primero, tratar la señal de voltaje interpretada por el sensor de corriente, la cual estaría entre valores de 0 a 5 V, la cual para asegurarnos que no existan valores negativos, ni mayores a 5 V diseñamos un pequeño circuito analógico el cual permitiría subir la señal con un nivel DC y centrarla en 2.5 V, lo que nos garantizaría no dañar el Arduino Uno; segundo, teníamos que procesar estos valores referenciales de voltaje y poder convertirlos a corriente haciendo uso de una librería muy importante que es Emonlib.h, la cual nos facilitó el procesamiento de la misma, para así de esta forma poder realizar el cálculo del consumo de energía consumida; tercero, estos dos datos van a ser enviados a la minicomputadora Raspberry Pi, que es la que nos va a permitir tener los accesos correspondientes para poder presentar los datos en el web server casero Cosm.com a través de una conexión a Internet por medio de un cable Ethernet o vía wifi, además de poder presentarlos en el terminal del Raspberry Pi, así como la creación de los archivos correspondientes para el almacenamiento de cada una de las variables, para de esta forma obtener un respaldo de los datos y poder tener nuestra propia base de datos la cual podríamos hacer uso en cualquier momento.

Podemos observar en la figura 11, la implementación total del proyecto, visualizando las tres etapas antes mencionadas.

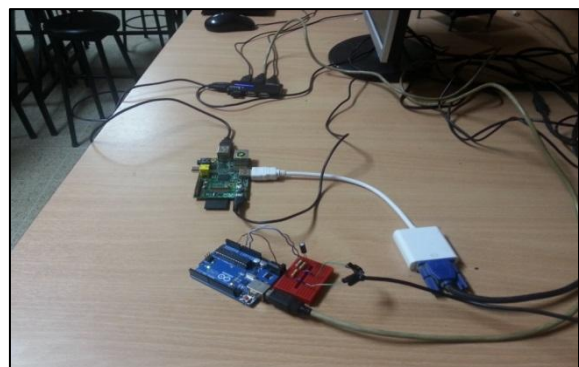


Figura 11. Componentes principales del proyecto

## 4. Funcionamiento

Para efecto de poner en funcionamiento nuestro proyecto de grado hemos llevado a cabo las pruebas experimentales en dos dispositivos eléctricos que funcionan a 220 V cada uno, el primero un aire acondicionado convencional de 24000 BTU y el segundo una secadora de ropa de hogar con diferentes ciclos de lavado.

En esta parte detallaremos el algoritmo utilizado tanto para el Arduino Uno, como para el Raspberry Pi, además del funcionamiento de nuestro proyecto en los dos dispositivos antes mencionados.

#### **4.1 Descripción del Algoritmo**

Para ambos compiladores el lenguaje de programación es parecido al del lenguaje C, el cual hemos adquirido conocimientos previamente dentro del pensum de nuestra carrera, por lo que no fue complicado establecer líneas de comandos que nos permitieran cumplir con nuestro objetivo.

##### **Algoritmo utilizado en Arduino IDE**

Se incluye la librería EmonLib.h que contiene internamente ya ciertas funciones implementadas para el cálculo de corriente a través de ciertos sensores, además de la calibración por si existe porcentajes de error en los mismos.

Se declaran las variables que van a ser utilizadas durante el programa.

Se declara la variable entera z y se la encera a la vez.

Se declara el procedimiento void setup() donde se fijan las configuraciones de la tasa de transmisión serial en 9600 y se calibra el porcentaje de error de la medición de corriente.

Se realiza un lazo infinito y dentro de él un lazo for en donde se incrementa la variable z.

Se calcula el valor de la corriente en ese instante y se lo asigna a la variable valor.

Se pregunta por la variable valor, si es mayor a 30 sale del lazo infinito.

Si la variable valor es menor a 30 envía serialmente el valor calculado de la corriente al Raspberry Pi mediante el comando serial.println(valor).

Se establece un retardo de 1s.

Finalmente se llama a la función reset(), donde se enceran todas las variables.

##### **Algoritmo utilizado en Python**

Se importan las librerías utilizadas en el proyecto que son las siguientes:

- eeml: permite la conexión con la dirección url cosm.com.
- serial: activa la comunicación vía USB con el Raspberry Pi.
- strptime y datetime: librerías de tiempo.

Se incluyen las llaves que permiten acceso con la cuenta creada en cosm.com.

Se define la variable y, que permite el control de la toma de datos cada 5 segundos.

Se define la variable ser que permite configurar la tasa de transmisión serial de la comunicación USB en 9600.

Se define la variable startTime y se le asigna el tiempo que registra el ordenador.

Se crea un lazo infinito mediante el comando while 1.

Se asigna a la variable x los datos de consumo eléctrico que están llegando vía USB en la comunicación serial.

Se asigna el valor de tiempo que registra el ordenador en ese momento a la variable now.

Se asigna la diferencia de tiempos entre las variables now y startTime a la variable elapsedTime.

Se imprimen los valores de tiempo, corriente y potencia consumida en el Lx Terminal de la minicomputadora Raspberry Pi.

Se crea un archivo con extensión .csv en modo de escritura y se almacenan los datos de tiempo, corriente y potencia consumida.

Se crea un archivo con extensión .txt en modo de escritura y se almacenan la hora, minutos y segundos como parte de tiempo, y como datos de consumo eléctrico almacenamos la corriente y la potencia consumida.

Se cierran los archivos con extensión .csv y .txt.

Se asignan los permisos para la conexión con la url cosm.com a la variable pac.

Se envían los datos en tiempo real a la url cosm.com, en el formato indicado de acuerdo a los parámetros establecidos.

Se realiza la gráfica en tiempo real mediante el comando pac.put().

Se encera la variable x.

Se consulta por la variable y, si es igual a 5 se deshabilita la comunicación serial, se establece un retardo de 1 segundo y se encera la variable y.

El lazo while terminará en el momento que exista una interrupción por teclado.

#### **4.2 Análisis del consumo de energía eléctrica para un aire acondicionado monofásico a 220V**

Hemos llevado a cabo la adquisición y evaluación de los datos de consumo de energía eléctrica en un aire acondicionado de hogar de 24000 BTU, con un consumo de corriente teórico de máximo 14.5 A. Al no conocer el factor de potencia real del aire acondicionado hemos estimado un valor aproximado de 0.85, este valor es el que utilizan ciertos fabricantes para esta clase de motores. Con ayuda de nuestro proyecto nos permitirá analizar el período de sleep que se produce en ciertos aires acondicionados, que se realiza cuando se ha llegado a la temperatura deseada a través del valor que censa el termostato, en este momento se apaga el compresor y disminuye el consumo de corriente, por lo cual también disminuirá el consumo de energía eléctrica. Esta prueba la realizamos conectando un cable Ethernet directamente al puerto LAN de la tarjeta, con lo cual no hubo ningún problema durante



el tiempo que se realizó la captura y presentación de los datos. También realizamos varias gráficas del consumo de energía eléctrica haciendo uso del programa Gnuplot.

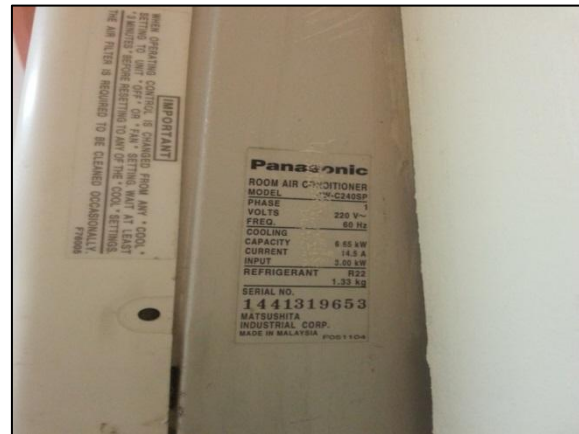
Tuvimos que realizar un pequeño corte en el cable de alimentación del aire acondicionado para poder separar los cables de línea viva y el neutro que se encontraban dentro del recubrimiento del mismo como se observa en la figura 12 y esto debido a que el sensor de corriente SCT-013-030 que utilizamos en nuestro proyecto solamente censa los niveles de corriente que circulan por uno de los cables de línea viva. Por el otro cable de línea viva circula la misma corriente siempre y cuando la carga se encuentre balanceada, mientras que por el neutro no circula corriente, aunque dependiendo de la instalación eléctrica pueden presentarse corrientes parásitas de muy baja magnitud.



**Figura 12.** Sensor SCT-013-030 de color azul y amperímetro de gancho

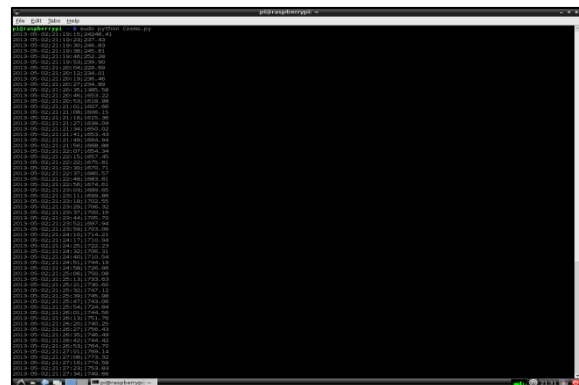
Una vez realizada todo el cableado y de haber colocado el sensor de corriente en uno de los cables de línea viva se procede a poner en marcha el proyecto para de esta forma poder monitorear el consumo de energía del aire acondicionado remotamente desde cualquier computadora o dispositivo con acceso a internet.

Podemos observar en la figura 13 algunas de las características del aire acondicionado especificadas por el fabricante como por ejemplo la fase, el voltaje de entrada, la frecuencia de operación, la corriente máxima, la potencia que consume, el tipo de refrigerante, etc. Todos estos valores son teóricos y esto lo mencionamos porque debido a los años que tiene el aire acondicionado hace que el aire acondicionado ya no entregue la misma capacidad que cuando era nuevo.



**Figura 13.** Características del aire acondicionado especificadas por el fabricante

Podemos observar en la figura 14 la adquisición de los datos de consumo de energía eléctrica del aire acondicionado que se actualizan cada tres segundos aproximadamente en el LX Terminal del Raspberry Pi, en el formato de fecha, hora, minutos, segundos, corriente y potencia consumida.



**Figura 14.** Presentación de los datos de consumo de energía eléctrica en el LX Terminal del Raspberry Pi

### 4.3 Análisis del consumo de energía eléctrica para una secadora de ropa de diferentes ciclos

Se llevó a cabo también la captura y procesamiento de los datos de consumo de energía eléctrica para una secadora de ropa a 220V durante un ciclo de secado de 60 minutos. Para esta prueba también se consideró un factor de potencia de 0.85, la corriente teórica máxima que podía entregar la secadora era de 20A. Se monitoreó remotamente los datos de corriente y potencia, visualizándolos gráficamente en el servidor Cosm. Para este caso conectamos inalámbricamente la tarjeta Raspberry Pi, al contrario del primer ejercicio, realizando esto para experimentar el correcto funcionamiento de la tarjeta Raspberry Pi debido a las limitaciones que esta posee, sin embargo nos dimos cuenta que el conectar muchos dispositivos al hub-USB que está directamente conectado a uno de los puertos USB de

la tarjeta hace que dejen de funcionar otros dispositivos por la sobrecarga de la misma. Igual que con la prueba del aire acondicionado también se realizaron varias gráficas en diferentes intervalos de tiempo haciendo uso del programa Gnuplot.

Trabajando ahora con una secadora de ropa casera con un consumo mayor al del aire acondicionado realizamos la conexión y cableado de todos los elementos de hardware del proyecto como se ilustra en la figura 15.



**Figura 15.** Conexión y cableado de los elementos de hardware del proyecto – secadora 220 V

Para esta prueba tuvimos que realizar una separación en el cable de alimentación de la secadora de ropa como se observa en la figura 16, para así poder adaptar el sensor de corriente SCT-013-030.



**Figura 16.** Adaptación del sensor de corriente SCT-013-030 al cable de alimentación de la secadora de ropa a 220 V

Dejamos funcionando el proyecto con la secadora configurada para un ciclo de trabajo de 60 minutos y a la vez monitoreando remotamente el consumo de energía eléctrica desde nuestra computadora personal.

## 5. Resultados

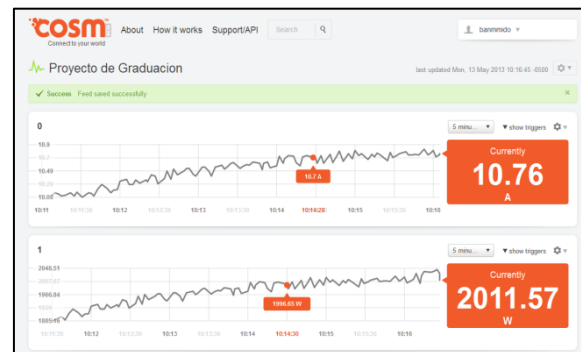
Cabe recalcar que todos los datos adquiridos se están enviando en tiempo real a una computadora

personal, desde la cual estamos monitoreando remotamente el consumo de energía eléctrica del aire acondicionado a través del web server casero Cosm, accediendo a nuestra cuenta a través de un usuario y contraseña.

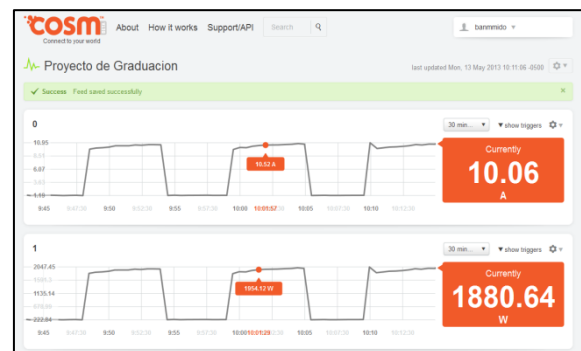
Una vez puesto en marcha el funcionamiento del proyecto tanto para el aire acondicionado como para la secadora de ropa obtuvimos los siguientes resultados:

### Consumo de energía eléctrica/aire acondicionado

En las figuras 17, 18 y 19 podemos ver las gráficas de corriente y potencia consumida durante los últimos cinco minutos, media hora y una hora, donde se puede apreciar que la corriente máxima durante ese intervalo de tiempo es de 10.95 A y la mínima es de 1.19 A, así mismo la amplitud máxima de potencia consumida es de 2047 Watts, mientras que la mínima es de 222 Watts. Nos preguntaremos a que se debe que exista un máximo y un mínimo de corriente y potencia consumida, y no sea constante como esperamos, la respuesta es sencilla debido a que ciertos aires acondicionados tienen un período de sleep en el cual su compresor se apaga trabajando al mínimo posible cuando el termostato censa que ha llegado a la temperatura indicada.



**Figura 17.** Gráfica del consumo de energía eléctrica del aire acondicionado en Cosm durante los últimos 5 minutos



**Figura 18.** Gráfica del consumo de energía eléctrica del aire acondicionado en Cosm durante los últimos 30 minutos





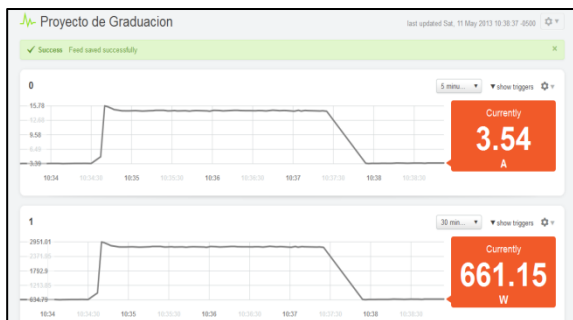
**Figura 19.** Gráfica del consumo de energía eléctrica del aire acondicionado en Cosm durante la última hora

Tanto la gráfica de corriente y potencia consumida tienen la misma tendencia debido a que son directamente proporcionales al permanecer el voltaje y el factor de potencia constantes.

Si queremos saber el costo del consumo de energía del aire acondicionado simplemente procedemos a sacar el área bajo la curva durante un intervalo de tiempo específico y lo multiplicamos el valor comercial al que se encuentre regulado el kilowatt/hora en el país.

#### Consumo de energía eléctrica/secadora de ropa

Durante un ciclo de lavado de 60 minutos obtuvimos los siguientes resultados:

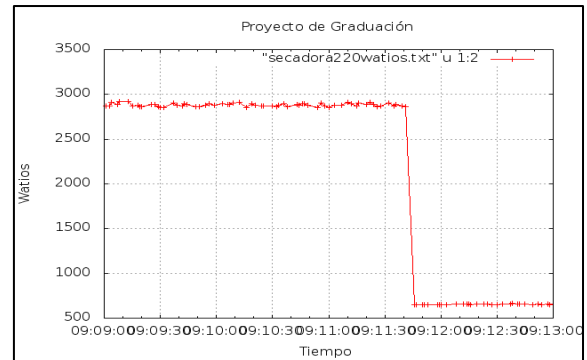


**Figura 20.** Gráfica del consumo de energía eléctrica de la secadora de ropa en Cosm durante 5 minutos

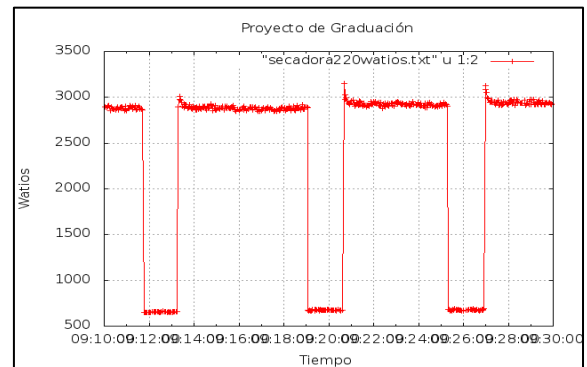
Como podemos apreciar en la figura 20 durante los últimos cinco minutos la corriente máxima fue de 15.78 A y la mínima de 3.39 A. Al igual que el aire acondicionado la secadora de ropa tiene un período de tiempo en donde su consumo de energía disminuye. Así mismo sucede con la potencia consumida, la máxima bordea los 2950 Watts y la mínima los 630 Watts.

Además analizamos los datos almacenados en nuestra base de datos para luego poder realizar gráficas con el programa Gnuplot y poder compararlas con las obtenidas en el web server.

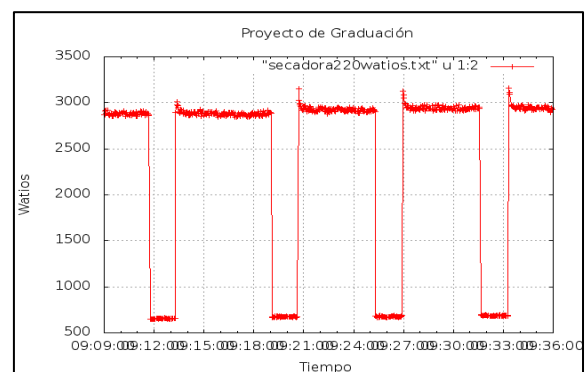
En las figuras 21, 22 y 23 podemos observar el consumo de corriente de la secadora de ropa en diferentes intervalos de tiempo utilizando el programa Gnuplot para la construcción de gráficas en 2D. Se puede apreciar que los valores son muy similares a los obtenidos en Cosm y esto es porque los valores que se envían a Cosm y los que se guardan en el archivo con extensión .txt son los mismos, el único cambio es el método de graficación.



**Figura 21.** Gráfica de potencia consumida de la secadora de ropa a 220 V realizada en Gnuplot durante 4 minutos



**Figura 22.** Gráfica de potencia consumida de la secadora de ropa a 220 V realizada en Gnuplot durante 20 minutos



**Figura 23.** Gráfica de potencia consumida de la secadora de ropa a 220 V realizada en Gnuplot durante 30 minutos

## 6. Conclusiones

El Raspberry Pi es una microcomputadora la cual a pesar de sus limitaciones es perfectamente capaz de realizar la mayoría de las tareas de cualquier ordenador actual, siempre y cuando el usuario que la posee le dé el mantenimiento y desarrollo necesario. Esto se debe a que la mayoría de actualizaciones, programas y software necesarios son open source, y existe una vasta cantidad de información acerca de ellos en foros por toda la red, lo cual le permitiría al usuario desarrollar más su Raspberry Pi con un poco de esfuerzo y dedicación.

Sin embargo si hablamos de open source, necesariamente debemos mencionar al Arduino Uno, el cual fue clave en la implementación de este proyecto, ya que este posee más tiempo en el mercado y debido precisamente a eso se han desarrollado códigos de programación que permiten comunicarse ya sea mediante el uso del puerto serial (Rx, Tx) o por el puerto USB con otro dispositivo externo, y es ahí donde está la participación del Raspberry Pi. Cabe recalcar que este tipo de comunicación no es posible lograr sino se posee los drivers adecuados para dicho fin, de hecho depende mucho de que programa se esté utilizando en el Raspberry Pi para lograr la comunicación con el Arduino IDE.

Una de las principales razones por la cual se escogió como plataforma de programación al Python 2.7, es que ya existen librerías adecuadas para la comunicación USB entre el Raspberry Pi y el Arduino Uno. Es importante destacar que dicha comunicación es factible en una versión inferior a la 3.0 del programa Python, la razón es que las librerías Python-serial y py-serial, que fueron utilizadas en el desarrollo del proyecto, fueron creadas en base a una versión del Python anterior a la 3.0. Además al intercomunicar ambos hardware (Raspberry Pi y Arduino Uno) mediante un cable USB la alimentación necesaria que requiere el Arduino Uno para su funcionamiento se provee a su vez por este medio físico.

Durante la toma de datos de consumo eléctrico, es necesario refrescar la comunicación USB entre ambos equipos, sin mencionar que es de vital importancia tomar en cuenta los retardos de transmisión y recepción de datos. Esto con la finalidad de evitar presentar datos extraños o erróneos los cuales darán una gráfica errónea con picos de corriente inexistentes que serán transmitidos vía streaming a la página Cosm.com que podrían dar mal interpretaciones del consumo de cierto aparato eléctrico que se esté analizando .

El web server Cosm.com posee una deficiencia a la hora de presentar los datos, esta es que el rango de

almacenamiento de los datos no es muy flexible en cuanto al rango de tiempo. Por lo tanto, el web server casero phpMyAdmin se instaló en el Raspberry Pi con el objetivo de preservar toda la información del proyecto y que esta fuera utilizada en un uso posterior para otro análisis de estudio más detallado.

## 7. Referencias

- [1] Simon Monk, Programing the Raspberry Pi Getting started with Python, McGraw Hill, 2013, p 1
- [2] Eben Upton and Gareth Halfacree, Raspberry Pi User Guide, John Wiley and Sons Ltd., 2012, pp 12-13
- [3] Eben Upton and Gareth Halfacree, Raspberry Pi User Guide, John Wiley and Sons Ltd., 2012, p 14
- [4] Eben Upton and Gareth Halfacree, Raspberry Pi User Guide, John Wiley and Sons Ltd., 2012, p 15
- [5] Simon Monk, Programing the Raspberry Pi Getting started with Python, McGraw Hill, 2013, p 7
- [6] Simon Monk, Programing the Raspberry Pi Getting started with Python, McGraw Hill, 2013, p 125
- [7] Eben Upton and Gareth Halfacree, Raspberry Pi User Guide, John Wiley and Sons Ltd., 2012, p 188
- [8] Michael McRoberts, Beginning Arduino, Springer Science+Business Media, 2010