

Diseño de interfaz de control y adquisición de datos para un prototipo didáctico de refrigeración

Design of control interface and data acquisition for a didactic cooling prototype

CHAVARRÍA-DOMÍNGUEZ, Fernando†, JIMENEZ-SILVA, J. Isidro, HERNANDEZ-BARABATA, Fernando y NOBLE-PEREZ, Hugo I.

Universidad Veracruzana, Facultad de Ingeniería

ID 1^{er} Autor: *Fernando, Chavarría-Domínguez* / ORC ID: 0000-0002-0858-1431, arXiv Author ID: NS8OED-Y3SW7Q, CVU CONACYT ID: 475091

ID 1^{er} Coautor: *J. Isidro, Jimenez-Silva* / ORC ID: 0000-0002-4057-7889, CVU CONACYT ID: 597479

ID 2^{do} Coautor: *Fernando, Hernandez-Barabata* / ORC ID: 0000-002-4366-6350, CVU CONACYT ID: 1014029

ID 3^{er} Coautor: *Hugo I., Noble-Perez* / ORC ID: 0000-003-3474-9303, CVU CONACYT ID: 597455

DOI: 10.35429/JID.2019.7.3.1.7

Recibido 09 Marzo, 2019; Aceptado 30 de Junio, 2019

Resumen

El objetivo principal de este trabajo es presentar el diseño y construcción de un sistema de control y adquisición de datos para un prototipo de refrigeración didáctico, el prototipo didáctico consiste en un ciclo de refrigeración por evaporación en el cual es importante conocer las variables de voltaje, corriente en el compresor así como presión y temperatura en cada etapa del ciclo, para así poder comprender los fenómenos termodinámicos que ocurren en su operación. El sistema de control y adquisición de datos lee, registra y visualiza en un interfaz hombre maquina las variables de voltaje, corriente, temperatura y presión, a su vez controla la operación del compresor y ventilador del prototipo, cuando alguna variable del proceso excede los límites permisibles. En este sistema utilizamos la plataforma arduino, sensores de presión, voltaje, temperatura y corriente como parte del sistema de adquisición y acondicionamiento de señales, como entorno de instrumentación virtual el software labVIEW y un módulo de relevadores como etapa de potencia para controlar las salidas del sistema. Para acondicionar las lecturas del sensor de voltaje se utilizó una regresión polinomial de 3 orden. Para verificar el funcionamiento del sistema de adquisición de datos se compararon las lecturas contra mediciones de equipos de laboratorios calibrados.

Arduino, Regresión polinomial, Adquisición de datos

Abstract

The main objective of this work is to present the design and construction of a control system and data acquisition for a didactic refrigeration prototype, the didactic prototype consists of an evaporation cooling cycle in which it is important to know the voltage variables, current in the compressor, as well as pressure and temperature at each stage of the cycle, in order to understand the thermodynamic phenomena that occur in its operation. The data acquisition and control system reads, registers and displays the voltage, current, temperature and pressure variables on a machine-man interface, in turn it controls the operation of the prototype compressor and fan, when some process variable exceeds the limits permissible. In this system we use the Arduino platform, pressure, voltage, temperature and current sensors as part of the signal acquisition and conditioning system, as a virtual instrumentation environment the LabVIEW software, and a relay module as a power stage to control the outputs of the system. A polynomial regression of 3 orders was used to condition the voltage sensor readings. To verify the operation of the data acquisition system, the readings were compared against measurements of calibrated laboratory equipment.

Arduino, Polynomial regression, Data acquisition

Citación: CHAVARRÍA-DOMÍNGUEZ, Fernando, JIMENEZ-SILVA, J. Isidro, HERNANDEZ-BARABATA, Fernando y NOBLE-PEREZ, Hugo I. Diseño de interfaz de control y adquisición de datos para un prototipo didáctico de refrigeración. Revista del Diseño Innovativo. 2019. 3-7: 1-7

† Investigador contribuyendo como primer autor.

Introducción

En este trabajo se presenta una propuesta de instrumentación y control virtual de un sistema de refrigeración didáctico, construido en el laboratorio de energía de la Facultad de ingeniería de la universidad veracruzana, resultado de la necesidad de contar con equipos de práctica que faciliten del aprendizaje significativo de los alumnos. Los temas referentes a aire acondicionado y refrigeración son de vital importancia en la formación de técnicos e ingenieros, Según un estudio reciente llevado a cabo por el INEGI (Instituto Nacional de Estadística y Geografía), con el nombre “Primera encuesta nacional sobre consumo de energéticos en viviendas particulares” dado a conocer el 7 de noviembre del 2018, menciona que el 99% de las viviendas habitadas del país cuentan con electricidad, de ellas el 88% cuenta con algún tipo de refrigerador indicando que pudieran existir más de 28.9 millones de equipos en uso, mientras que el uso de aires acondicionados se contabiliza en un aproximado de 7 millones de equipos en viviendas particulares del país, distribuidos en las diferentes zonas del país, siendo su mayor concentración en la zona norte siendo está catalogada como “zona cálida extrema” donde el porcentaje de viviendas que cuentan con algún tipo de aire acondicionado es del 48.1%.

El prototipo didáctico de refrigeración por evaporación cuenta con 4 etapas básicas en su operación: regulación, evaporación, compresión y condensación, en los cuales es necesario conocer los valores de presión y temperatura, para poder comprender los fenómenos termodinámicos que ocurren en cada una de ellas. También es necesario conocer parámetros eléctricos como voltaje y corriente consumida por el compresor, para poder determinar el correcto funcionamiento de este componente. Inicialmente el prototipo contaba con termómetros y manómetros analógicos para poder observar el comportamiento termodinámico del ciclo.

La propuesta es integrar una interfaz hombre maquina (HMI) por sus siglas en inglés, por medio de un sistema de adquisición de datos de bajo costo basados en la plataforma arduino y vinculado a un entorno de instrumentación virtual en el software LabVIEW.

En lo subsecuente se presenta la selección de sensores, la programación de la plataforma de adquisición de datos, la programación de los instrumentos virtuales VI por sus siglas en inglés, y la integración de todo el sistema.

Planteamiento

Se colocaron sensores en las siguientes partes del ciclo de refrigeración:

1. Salida del evaporador y succión de compresor, se coloca un sensor de presión en este punto midiendo el comportamiento del refrigerante en esa porción del sistema (región de baja presión).
2. Compresor
3. Salida del compresor hacia el condensador, se coloca un sensor de presión en este punto midiendo el comportamiento del refrigerante en esa porción del sistema (región de alta presión).
4. Entrada al condensador, se coloca un sensor de temperatura para observar su comportamiento en esta región (región de alta temperatura).
5. Condensador, en este punto se colocan dos sensores, tanto el sensor de nivel de líquido como un sensor de temperatura, el sensor de nivel de agua tendrá el gobierno de la bomba de agua para mantener a nivel de líquido en un rango apropiado, mientras que el sensor de temperatura estará ahí para observar esta misma en esta área específica del proceso.
6. Salida del condensador, se coloca otro sensor de temperatura en este punto para observar este comportamiento en este punto del sistema, así como compararlo con el punto cuatro a la entrada del condensador.
7. Salida del condensador hacia la válvula de expansión región de alta temperatura en este punto se coloca otro sensor de presión para observar esta variable en este punto.
8. Salida de la válvula de expansión hacia el evaporador, en este punto se coloca otro sensor de presión para observar lo que sucede con esta variable en esta zona (región de baja presión).

9. Entrada al evaporador, en este punto se coloca otro sensor de temperatura para observar lo que ocurre con esta en este punto.
10. Zona del evaporador y lugar donde se coloca otro sensor de temperatura para el control de histéresis del compresor y ventilador.
11. Ventilador del evaporador.
12. Salida del evaporador, se coloca el último sensor de temperatura para observar el comportamiento de esta en este punto y compararla con el punto nueve.

Programación de la plataforma de adquisición de datos

Los sensores de corriente y voltaje se colocan justo en la fuente de alimentación del prototipo abarcando el energizado general y por tanto no dejando de censar ningún aparato del proceso.

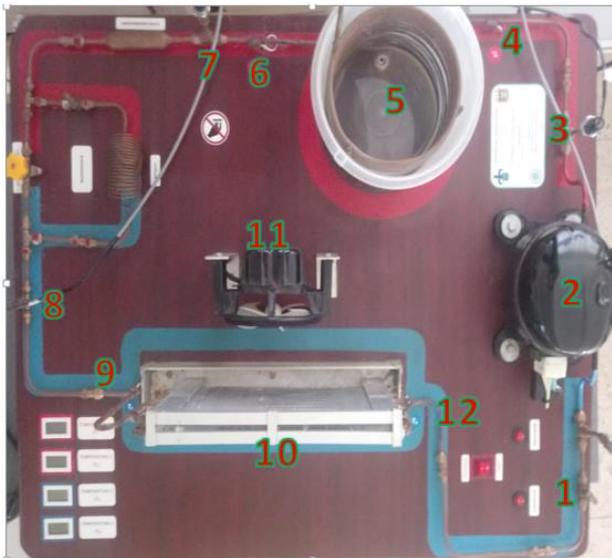


Figura 1 Distribución de sensores y partes del prototipo
Fuente: *Elaboración Propia*

Los sensores que se utilizaron para este elemento y la programación empleada en la plataforma arduino son los siguientes:

Para los sensores de temperatura se utiliza el dispositivo digital DS18B20 con la librería "onewire" este dispositivo permitirá obtener los datos de todos nuestros sensores a través de un solo cable, por ende, de un solo puerto de nuestra placa Arduino, se declara la librería y se ajustan los parámetros de entradas de datos, posteriormente se integran los sensores de temperatura que se utilizan.

```
#include <OneWire.h>
#include <DallasTemperature.h>

OneWire pinSensor(2);
DallasTemperature sensor(&pinSensor);

void setup() {

  Serial.begin(115200);
  sensor.begin();

}

void loop() {
  sensor.requestTemperatures();

  int temp = (sensor.getTempCByIndex(0));

  Serial.println("A" + String(temp));
}
```

Figura 2 Declaración de librería y ajuste de parámetros para sensor de temperatura
Fuente: *Elaboración Propia*

```
void temperaturas() {
  sensor.requestTemperatures();
  int temp = (sensor.getTempCByIndex(0));
  int temp2 = (sensor.getTempCByIndex(1));
  int temp3 = (sensor.getTempCByIndex(2));
  int temp4 = (sensor.getTempCByIndex(3));
  int temp5 = (sensor.getTempCByIndex(4));
  int temp6 = (sensor.getTempCByIndex(5));
  Serial.println("A" + String(temp));
  Serial.println("B" + String(temp2));
  Serial.println("C" + String(temp3));
  Serial.println("D" + String(temp4));
  Serial.println("E" + String(temp5));
  Serial.println("F" + String(temp6));
}
```

Figura 3 Integración de los sensores de temperatura
Fuente: *Elaboración propia*

Para la medición de presión se emplea el sensor el HK1100C, es un sensor transductor de presión análogo, transmite un rango de voltaje de 0.5 VCD de salida a 0 PSI y 4.5 VCD a 174 PSI, en tanto se lleva a cabo la conversión de los datos obtenidos. Para este sensor se declara la variable y a su vez esta indicara el pin de entrada análogo en la placa Arduino, seguido en el "setup" se da inicio a la comunicación serial, para en el "loop" dar lectura a la información proveniente del sensor de presión y la conversión interna que hace Arduino, sabiendo que Arduino convierte los valores análogos de 0 a 1023 puntos de medida en función del voltaje de entrada en la terminal seleccionada, se expresa la formula con la cual se obtiene el voltaje.

Una vez hecho esto se coloca la fórmula que nos permitirá saber la presión en PSI misma que recomienda el fabricante, de esta forma queda expresado nuestro algoritmo.

```
int SPI = A0;

void setup() {
  Serial.begin(115200);
}

void loop() {
  int VSP=analogRead(SPI);
  float VsensorP = (VSP * 5.0) /1024.0;
  float pres = 217.5*(VsensorP/5.0)-21.75;
  Serial.println(pres);
  delay(250);
}
}
```

Figura 4 Programación para sensor de presión *Fuente: Elaboración Propia*

Para la corriente se ocupa el sensor de corriente alterna no invasivo SCT – 013 – 030, Uno de los factores importantes con los que se cuenta es el número de espiras en el núcleo ferromagnético, En el caso del sensor con el que se cuenta SCT-013-030 el devanado primario es el cable que se desea medir y el número de vueltas es uno, mientras que, el devanado secundario tiene 2000 vueltas. Se utiliza un acondicionamiento de señal con el propósito de adecuar esta misma a parámetros que nuestra placa de adquisición de datos sea capaz de interpretar sin que esta sea dañada, el acondicionador está compuesto por una resistencia de carga (derecha) y un offset en DC (centro) como se muestra.

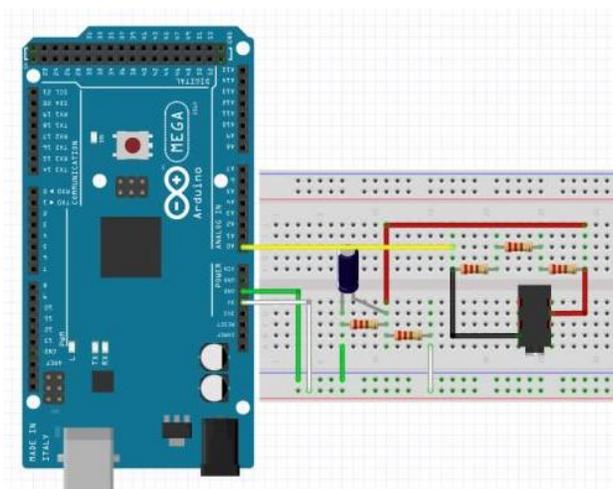


Figura 5 Conexión de acondicionamiento de señal *Fuente: Elaboración Propia*

Para programar el sensor de corriente en IDE de arduino se utiliza una librería conocida como “EmonLib”, esta librería se da a la tarea de acondicionar la señal de entrada a una lectura estable, la programación es sencilla y se puede observar en la figura

```
#include "EmonLib.h"
EnergyMonitor emon1;
void setup()
{
  Serial.begin(9600);

  emon1.current(1, 111.1);
}

void loop()
{
  double Irms = emon1.calcIrms(1480);

  Serial.print("J" + String(Irms));
}
}
```

Figura 6 Programación para sensor de corriente *Fuente: Elaboración propia*

Para censar el voltaje se utiliza el sensor ZMPT101B el cual tiene la capacidad de medir hasta un voltaje de 250 VCA cuenta con un ajuste de giro múltiple con potenciómetro para ajustar la salida a DC, ya que internamente reproduce una señal de onda adecuada para la lectura analógica de la tarjeta Arduino, por lo cual este ajuste nos permite situar un cero artificial para la medida de la onda.

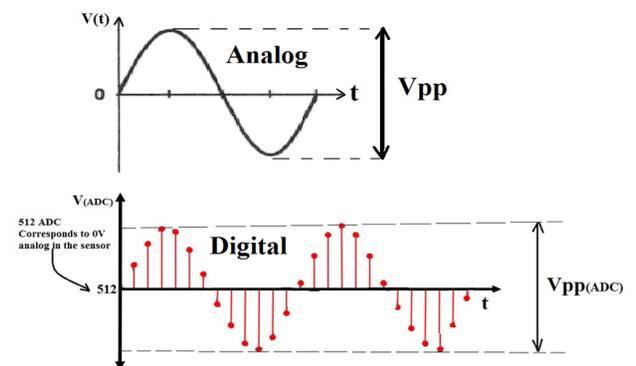


Figura 7 Ajuste de potenciómetro *Fuente: www.researchgate.net/publication*

La programación de este módulo sensor se basa en una publicación previa que consiste en una caracterización de los datos obtenidos de una placa Arduino a través del sensor, adecuando una regresión polinomial de 1ro a 5to orden siendo el polinomio de tercer orden el que mejor resultados presenta según este trabajo, mismo que cuenta con dos algoritmos de medición uno es de pico a pico y el otro es instantáneo [I].

Se decide usar el algoritmo de pico a pico con la regresión polinomial de tercer orden pues es la que muestra menos desviaciones en la medición, se realizaron algunos ajustes en el algoritmo para completarlo.

```
#define VOLTAJEPIN A0
int Vmax;
int Vmin;
int valorsensor;
int valorsensor1;
int valorsensor2;
int V;
int Vrms;
long contador = 0;

void setup() {
    Serial.begin(9600);
}

void loop(){
    if (millis()-contador > 1000) {
        contador=millis();
        Vmax = 0;
        Vmin = 1023;
        for (int i=0;i<1000;i++){
            valorsensor1 = Vmax;
            valorsensor2 = Vmin;
            valorsensor = analogRead(VOLTAJEPIN);
            if(valorsensor > valorsensor1){
                Vmax = valorsensor;
                goto bailout;
            }
            if (valorsensor < valorsensor2) {
                Vmin = valorsensor;
                goto bailout;
            }
            Vmax = valorsensor1;
            Vmin = valorsensor2;
            bailout:
            delay(0.1);
        }
        float x = (Vmax - Vmin);
        V = (0.00000412*x*x*x - 0.000857*x*x + 2.375*x - 3.198);
        Vrms = V/(2*sqrt(2));
        Serial.println("El Voltaje rms es: " + String(Vrms) + " Volts");
    }
}
```

Figura 8 Programación para modulo sensor ZMPT101B
Fuente: Elaboración Propia

Para establecer las señales de control se declaran los pines de salida que controlaran a los relevadores, en el setup se abre la comunicación serial al igual que se declara la funcionalidad de los pines en este caso de salida (OUTPUT), luego en el loop se declaran las condicionales if para cada salida de control, asignando una letra la cual es la que nos estará transmitiendo LabVIEW a manera de señales de control.

Entorno de instrumentación virtual

Para la programación en LabVIEW se utiliza la biblioteca “instrumentos/Serial”, tomando en cuenta que, las tarjetas Arduino son capaces de transmitir y recibir información a través de la comunicación serial con otros dispositivos, en este caso se estará comunicando con una computadora y LabVIEW, se hace una programación sencilla para leer todo el buffer de información en formato “string” que estará enviando la placa de adquisición Arduino Mega, se adiciona un discriminador, cada lectura que se obtiene e imprime en la comunicación serial se encuentra precedida por una letra en mayúscula, esta misma es la que se toma para hallar y **separar** todas las señales dentro del buffer, con ello se obtiene las señales de cada sensor que se ha colocado.

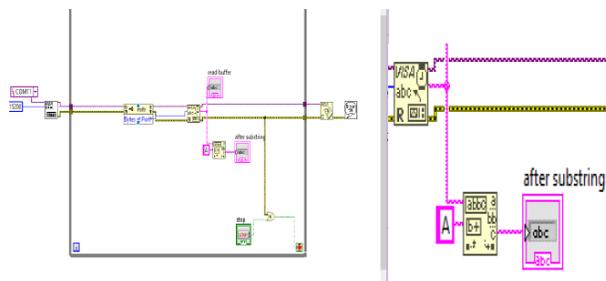


Figura 9 Discriminador
Fuente: Elaboración Propia

Los datos están en formato “String” por tanto es necesario una conversión a formato numérico para que estos datos alimenten de información a los instrumentos virtuales VIs, así para ellos se utiliza un conversor como se muestra en la figura.

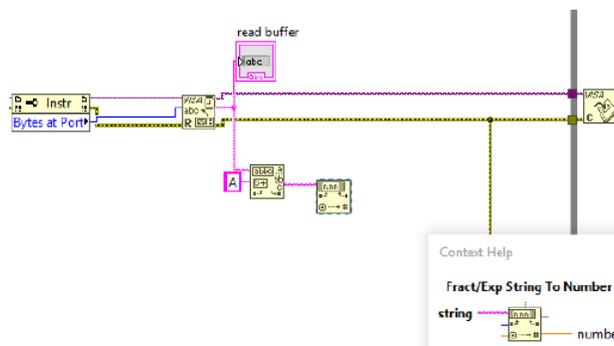


Figura 10 Conversión de cadena de caracteres a cadena numérica
Fuente: Elaboración Propia

Por necesidades de sincronización con LabVIEW y Arduino al VI se le da un tiempo de ejecución de cada 100 milisegundos, ello nos crea una perturbación en la lectura dado las velocidades desiguales de lectura entre Arduino y LabVIEW pero si modificamos el tiempo de ejecución es muy probable que haya otra clase de perturbaciones en las lecturas o controles, por tanto habrá que colocar una retroalimentación en el conversor de cadena de caracteres que se colocó, ellos nos ayudara a mantener una lectura estable y evitar otros problemas de sincronización, la programación queda como se muestra en la figura, y esto se replica para cada lectura.

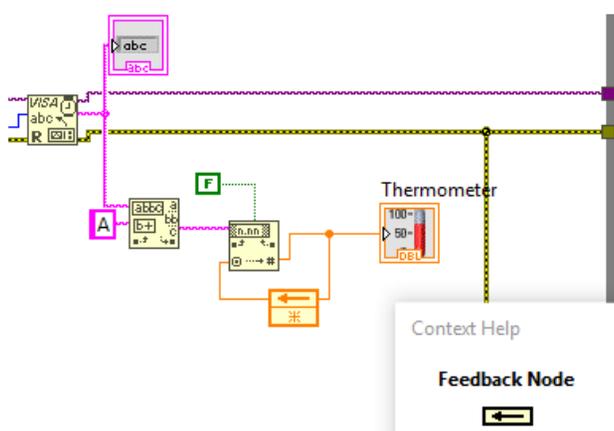


Figura 11 Colocación de la retroalimentación y VI de lector grafico

Fuente: *Elaboración Propia*

Para el apartado de control ya no será necesario el uso de discriminadores, sin embargo, es necesario el uso de selectores, la programación en este punto es más sencilla pues solo es necesario escribir en el puerto serial y mandar el código a la placa de control Arduino, misma que como se ha visto ya fue preparada para recibir tales comandos, la programación será como se muestra a continuación.

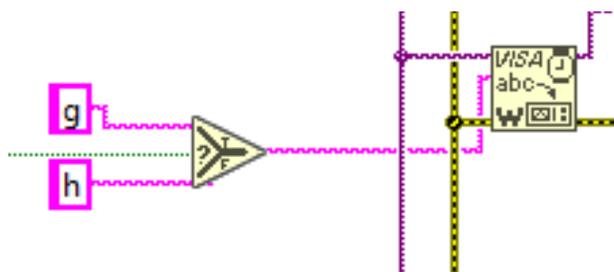


Figura 12 Selector y escritura serial

Fuente: *Elaboración Propia*

Esta operación se realiza por lo menos 4 veces, así tendremos control sobre cuatro puntos requeridos, energizado principal, control de ventilador, control de compresor y el control de la bomba de agua.

Resultados

Una vista final del panel visual de adquisición de datos y control se puede observar en la figura de izquierda a derecha en primera instancia se puede observar la botonera de paro y arranque general, debajo de ella se encuentran los “sets” de disparo para las diferentes protecciones, continuando con un indicador de cadena de caracteres que señalará el estado en el que se encuentra la máquina, debajo la animación de funcionamiento del ventilado junto a una lampara indicadora de estado general, esta solo mostrara si esta encendido o apagado el sistema, en el recuadro debajo de ellos se pueden observar a la izquierda las lámparas indicadoras precautorias las cuales encenderán al aproximarse a los niveles de disparo, junto a ellas (centro) indicadores de estado en cadena de caracteres y a la derecha las lámparas indicadoras de disparo, debajo del recuadro está el botón de “reset” el cual dará el permiso para reiniciar el sistema después de una falla, en la parte central de la imagen se encuentran los controles para el uso en automático o manual de la bomba de agua, el compresor y el ventilador del evaporador, y a la derecha de la imagen están todas las lecturas provenientes de los diferentes puntos de proceso, las seis de temperatura en grados centígrados, cuatro de presión en PSI, voltaje, corriente y potencia.

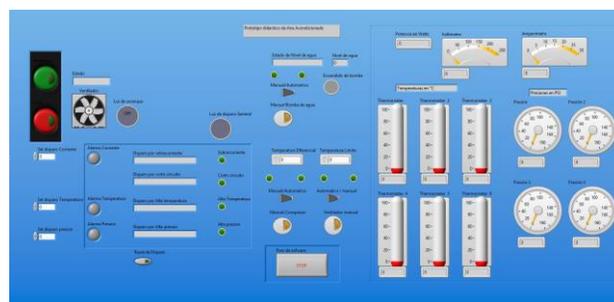


Figura 13 panel frontal final

Fuente: *Elaboración Propia*

Conclusiones

Se integraron todas las programaciones de los distintos sensores en el IDE de arduino para lograr una ejecución simultánea con una sola placa, esto permite tener un adquisidor de datos de bajo costo, la programación en LabVIEW permite tener una interfaz hombre máquina que monitorea el comportamiento de las variables, así como controlar ciertas funciones del ciclo de refrigeración, de manera automática o virtualmente activadas. Los ajustes de rango de operación también brindan una flexibilidad a la operación del sistema. Las lecturas tienen un rango de error aceptable, comparadas con instrumentos de medición, profesionales y calibrados.

Referencias

Abubakar, I., Khalid, S. N., Mustafa, M. W., Shareef, H., & Mustapha, M. (2017). Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring. *ARP Journal of Engineering and Applied Sciences*.

OpenEnergyMonitor. (s.f.). Learn OpenEnergyMonitor. Recuperado mayo, 2019, de <https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory>.

INEGI. (2018). PRIMERA ENCUESTA NACIONAL SOBRE CONSUMO DE ENERGÉTICOS EN VIVIENDAS PARTICULARES (ENCEVI) (COMUNICADO DE PRENSA NÚM. 541/18). Recuperado de <https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2018/EstSociodemo/ENCEVI2018.pdf>