

SISTEMA DE MONITOREO INALÁMBRICO DE BAJO COSTO PARA MÓDULOS FOTOVOLTAICOS EMPLEANDO RASPBERRY PI

Oscar May Tzuc

Instituto Tecnológico Superior Progreso
omay@itsprogreso.edu.mx

Juliana González Quijano

Instituto Tecnológico Superior Progreso
jgonzalez@itsprogreso.edu.mx

José Daniel Ortiz López

Instituto Tecnológico Superior Progreso
danielortizlopez9@gmail.com

Renán Quijano Cetina

Facultad de Ingeniería, Universidad Autónoma de Yucatán
rquijano@uady.mx

Resumen

El presente trabajo muestra el desarrollo de una red de monitoreo para sistemas fotovoltaicos basado en el uso de la tecnología inalámbrica ZigBee, microprocesadores ARM de 32 bits y una tarjeta de desarrollo Raspberry Pi. El sistema propuesto se encuentra formada por un módulo de sensado diseñados para la medición y transmisión de los parámetros de temperatura, voltaje y corriente de los paneles o arreglos fotovoltaicos. Dicho módulo se comunica con una tarjeta Raspberry Pi la cual realiza las funciones de sistema de coordinación central y servidor web. Así, el módulo ZigBee incorporado a la red es capaz de transmitir los parámetros a la tarjeta Raspberry Pi, la cual generará una base de datos con los valores recibidos, además de asignarles fecha y hora. Todos los

datos registrados pueden ser visualizados desde una aplicación web desarrollada, la cual se actualiza constantemente, mediante cualquier computadora o dispositivo móvil.

Palabra(s) Clave(s): PSoC, Raspberry Pi, red inalámbrica de sensores, sistemas embebidos, tecnología ZigBee.

1. Introducción

En la actualidad casi la totalidad de la energía consumida a nivel mundial proviene de la quema de combustible fósiles provocando la liberación de una gran cantidad de gases de efecto invernadero a la atmosfera. Esto aunado a la creciente necesidad de energía producto del constante incremento poblacional ha ocasionado que las tendencias energéticas a nivel mundial, hoy en día se enfoquen en la sustitución de las fuentes energéticas convencionales por alternativas sustentables [1,2].

De las diversas fuentes de energía renovables existentes, la energía solar es una de las más prometedoras a futuro debido a que es abundante, limpia, no genera ruido o alguna otra clase de contaminación en el ambiente [3]. Esto en conjunto con la constante reducción de los costos de producción en las celdas fotovoltaicas ha incentivado a los gobiernos de diversos países a la realización de políticas energéticas destinadas al incremento de consumo de sistemas fotovoltaicos tanto a nivel comercial, residencial e industrial [4,5].

La eficiencia de los paneles fotovoltaicos se ve seriamente afectada por el bloqueo de la radiación incidente sobre el panel, acumulación de suciedad sobre su superficie protectora así como por la acción de agentes degradantes del entorno [5]. Debido a esto monitoreo es considera un aspecto de vital importancia. Mediante él es posible llevar a cabo la observación de la estabilidad y desempeño del sistema proporcionando información que posibilite tomar medidas para optimizar el proceso de generación de energía [4]. Uno de los métodos más simple consiste en realizar la recolección de información y transferirla para su análisis a través de cables. Sin embargo, debido a los costos y limitaciones técnicas de la transmisión por cables, las estaciones de monitoreo requieren estar

razonablemente cerca de las plantas, lo cual trae consigo un incremento en el capital de inversión y costos de mantenimiento. Por lo tanto, para contrarrestar estos inconvenientes los sistemas de monitoreo basados en las tecnologías inalámbricas se han vislumbrado como una alternativa completamente viable para este tipo de aplicaciones bajo intemperie.

En la actualidad los sistemas de monitoreo existentes están asociados a inversores de algún fabricante en particular, son costosos y en la mayoría de los casos dependen de software exclusivo para obtener los datos desde una computadora. Por tal motivo en el presente trabajo se propone un sistema de monitoreo de bajo costo, que sea capaz de conectarse a diferentes sistemas fotovoltaicos independientemente del fabricante del inversor y los paneles. Además, este sistema puede operar de manera autónoma sin la necesidad de una computadora o software específico. Los datos serán accesibles para el usuario a través de una página web a la cual puede accederse desde cualquier computadora o dispositivo móvil con conexión a internet. La sección 2 del documento presenta las características de la tarjeta embebida empleada para el desarrollo del sistema de monitoreo. En la sección 3 se realiza una descripción de tecnología de comunicación inalámbrica empleada. En la sección 4 se describe la estructura de la red implementada. La sección 5 ofrece una descripción detallada de los componentes electrónicos del sistema. Luego, en la sección 6 se presenta algunos resultados de las mediciones experimentales realizadas. Finalmente, la sección 7 contiene las conclusiones del trabajo desarrollado.

2. Desarrollo

2.1 Sistemas Embebidos

Los sistemas embebidos son una combinación de hardware y software de computadoras y en algunas ocasiones piezas mecánicas u otras adicionales diseñados para realizar alguna función específica. Su principal característica es que al estar insertados dentro de un dispositivo que controlan están sujetos a cumplir requisitos de tamaño, fiabilidad, consumo y costo. La gran aplicabilidad de

estos sistemas en cualquier ámbito ha generado gran interés en la industria para su desarrollo [6].

Tarjeta de Desarrollo Raspberry Pi

La tarjeta Raspberry Pi es un ordenador de placa reducida o SBC de bajo costo, el cual opera bajo el sistema operativo Linux. En la actualidad ha adquirida una gran popularidad debido a su versatilidad de implementación en diversos proyectos como desarrollo de equipos media center, empleo micro controlador avanzado y la habilitación como servidores web, por citar algunos [7,8]. El modelo utilizado en este trabajo es el conocido como Raspberry Pi-B (figura 1).

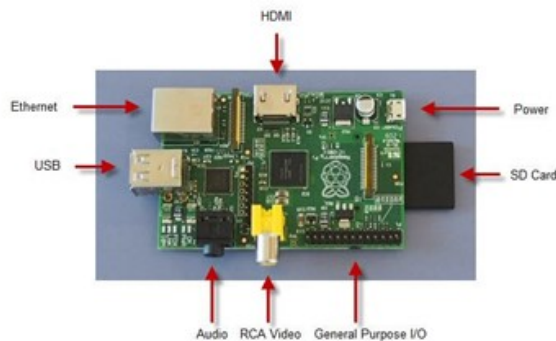


Figura 1 Tarjeta de desarrollo Raspberry Pi modelo B.

2.2 Dispositivos ZigBee

ZigBee es una tecnología de comunicación inalámbrica de bajo consumo y baja tasa de transferencia de datos, basada en las especificaciones del estándar IEEE 802.15.4, empleada para procesos de monitoreo y control de edificios [10,11]. Estos dispositivos poseen un rango de transmisión entre los 70 y 300 metros usando espectros de frecuencia en banda libre entre los 900 MHz y 2.4 GHz. Su estructura de red tiene atributos de Maestro/Esclavo y puede proporcionar funciones de comunicación en dos vías. Los ZigBee emplean niveles MAC mediante los cuales entablan su comunicación con mecanismos de prevención de errores incrementando la confiabilidad de la transmisión de datos [12].

Los dispositivos ZigBee presentan ciertas ventajas sobre otras tecnologías de comunicación inalámbrica enfocadas a la creación de redes de área y redes

personales. La tabla 1 muestra una comparativa de los puntos más relevantes entre las tecnologías ZigBee, Wi-Fi y bluetooth.

Tabla 1 Comparación entre diversas tecnologías inalámbricas [12].

Requerimiento	Wi-Fi	Bluetooth	ZigBee
Fuente de alimentación	Horas	Días	Años
Orden de complejidad de desarrollo	Muy complejo	Complejo	Simple
Número máximo de nodos	32	7	65536
Velocidad de segmento establecido	3 sec	10 sec	30 msec
Rango de transmisión	100m	10m	70m-300m
Transmisibilidad	11 Mbps	1 Mbps	250 Kbps
Seguridad	SSID	64 ó 128 bits	128 bits AES

La figura 2 muestra los módulos mediante los cuales se distribuye comercialmente la tecnología ZigBee conocidos como dispositivos XBee, El modelo de dispositivo empleado para el presente trabajo fue el XBee Pro S2.



Figura 2 Módulo comercial de comunicación inalámbrica XBee.

2.3 Red Inalámbrica de Sensores

Las redes inalámbricas de sensores (WSN, por sus siglas en inglés) comenzaron a desarrollarse a finales de los años 90's y para principios de la década de los 2000 se pusieron en marcha las primeras WSN funcionales. Las características de este tipo de redes, además de la transmisión de datos por medios no cableados, son: el bajo consumo energético, la velocidad de transmisión de datos de baja velocidad, la autonomía de la red y una topología dinámica, entre otras [13]. Para formar una WSN es necesario integrar diferentes componentes electrónicos que se enlacen a través de nodos especializados de

acuerdo con la función que se desea que desempeñen. Así, en una WSN típica es posible encontrar los denominados Nodos sensores, Nodos ruteadores, Nodos efectores y Nodos coordinadores. Algunos de estos nodos tienen la posibilidad de combinarse en un solo nodo que desempeñe más de una función. En la estructura del trabajo que aquí se presenta, se implementan nodos sensores, que operan en modo ruteador, así como un nodo coordinador que además funciona como puerto de enlace a Internet.

Nodo Sensor

Un nodo sensor está formado generalmente por uno o más sensores, un micro controlador, un módulo para la transmisión inalámbrica de las variables medidas y una fuente de alimentación [14]. Para optimizar el consumo de energía de los nodos sensores debe seleccionarse cuidadosamente cada uno de sus componentes y optimizar el código de programación del microcontrolador, con el propósito de mantener el dispositivo en modo de reposo durante los periodos que no transmite información [15].

Para el desarrollo del presente proyecto se seleccionó el Sistema en Chip Programable (PSoC, por sus siglas en inglés) CY8C4125AXI-483 de la compañía Cypress Semiconductors, el cual contiene un microcontrolador de última generación con características de bajo consumo energético y que integra módulos analógicos y digitales en un solo encapsulado. Por otra parte, el módulo para la transmisión inalámbrica de los datos obtenidos por los sensores utiliza la tecnología ZigBee.

Nodo Coordinador

El nodo coordinador administra la red solicitando a cada uno de los nodos sensores los datos medidos. Una vez que recibe los datos, los organiza, les asigna la fecha y hora en que fueron recibidos y los almacena en una base de datos para su visualización en la página web. El nodo coordinador ha sido implementado en una tarjeta de desarrollo Raspberry Pi, con capacidad de conexión a internet.

2.4 Diseño de la Red Inalámbrica de Sensores

La red inalámbrica desarrollada está conformada por tres elementos esenciales que operan de manera coordinada los cuales son el módulo sensor, el servidor web y el módulo de transmisión inalámbrica: (a) El módulo sensor se encuentra conformado por tres sensores diseñados para el sensado de las variables de temperatura (T), corriente (I) y voltaje (V) correspondientes a un arreglo o panel fotovoltaico; (b) Una tarjeta de desarrollo Raspberry Pi, funge como servidor web recabando la información medida por los módulos sensores para posteriormente procesarla, estructurar y cargarla a red para su visualización mediante una página web; (c) Módulos de comunicación inalámbricos los cuales emplean la tecnología ZigBee se encuentran conectados tanto a los módulos sensores, como al servidor web. Es a través de estos tres elementos que se establece la conexión inalámbrica a distancia para el monitoreo de los sistemas fotovoltaicos.

La red desarrollada presenta una configuración "Punto-Multipunto" permitiendo la comunicación de un dispositivo central (Nodo Coordinador) con todos los periféricos (Nodo Sensor) dentro de su rango de alcance mediante el empleo de broadcast [16], en la cual los Xbee (módulos comerciales que utilizan la tecnología ZigBee) conectados a los módulos de sensado operan en modo router y el Xbee coordinador se conecta directamente a la Raspberry Pi.

La figura 3 ilustra el proceso de comunicación para la adquisición de la temperatura, corriente y voltaje de los módulos fotovoltaicos. Este proceso comienza en el momento que el Coordinador solicita al Router, a través de su dirección MAC, la información recabado por cada uno de sus sensores; el Router almacena la dirección MAC del Coordinador, empaqueta la información solicitada en un *Frame data* [17] y realiza la transmisión de un promedio de las mediciones realizadas. De existir algún error en la comunicación el coordinador lleva a cabo la solicitud nuevamente hasta obtener una respuesta correcta. Una vez recibida la información por el Coordinador, la Raspberry Pi realiza el desempaquetamiento y conversión de la información contenida en el paquete de datos, y posteriormente la almacena en una base de datos incluyendo la fecha y hora de recepción para poder ser visualizado por cualquier dispositivo conectado a Internet. Este proceso

de solicitud y recepción de información se repite constantemente en intervalos de 10 minutos.

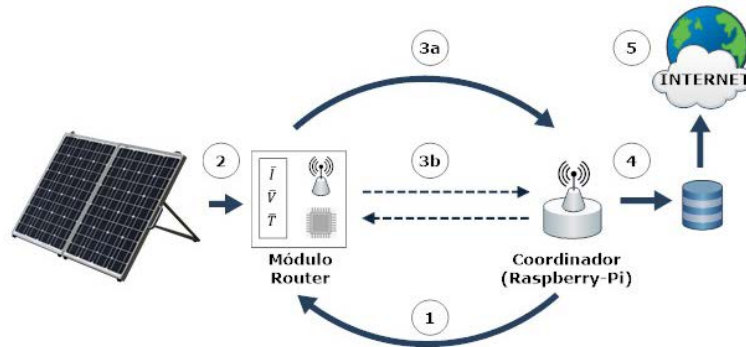


Figura 3 Proceso de comunicación y almacenamiento de información de la red de sensores inalámbricos para un par Coordinador-Router.

2.5 Componentes del Sistema

Sensores

Los parámetros medidos a través de los módulos de sensado (Voltaje, Corriente y Temperatura) corresponden a las variables esenciales presentes en los módulos fotovoltaicos. Para realizar estas mediciones fueron empleados dos sensores analógicos: un sensor lineal de efecto Hall (ACS712-05B) para la medición de corriente, un divisor de tensión con aislamiento galvánico para la medición de voltaje y un sensor digital DS18B20 para la medición de la temperatura del panel fotovoltaico. Los sensores envían la información recabada hacia el micro controlador, cuando este la solicita, en intervalos de 5 segundos. La figura 4 muestra el diagrama electrónico de acondicionamiento para la señal de los sensores hacia el microcontrolador.

Micro controlador PSoC 4

El micro controlador implementado en los módulos de sensado fue el PSoC CY 8C4128AXI-483 de la familia 4 de la compañía Cypress Semiconductor (figura 5), el cual posee una arquitectura ARM Cortex-M0 de 32 bits, 36 pines de propósito general (digital/analógico) programables y dos bloques reconfigurables para comunicación SPI, I2C o UART.

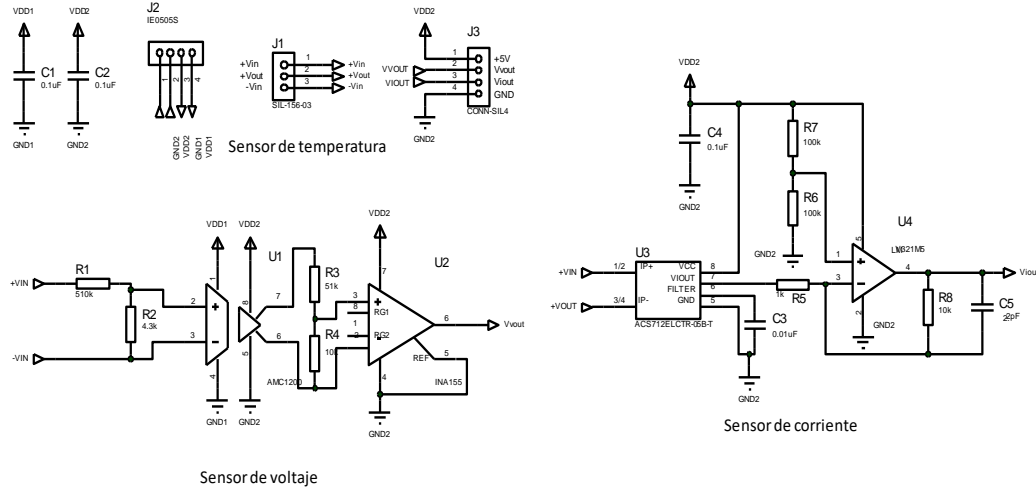


Figura 4 Diagrama eléctrico de los sensores de voltaje, corriente y temperatura.



Figura 5 Microcontrolador PSoC CY 8C4128AXI-483 sistema de monitoreo inalámbrico.

El código de programación implementado en los módulos de sensado es:

```

/*=====
                                     CODIGO DE COMUNICACIÓN DEL MÓDULO ROUTER
=====*/

//-----Importamos Librerías-----
#include <DallasTemperature.h>           //librería para medición de sensores DS18B20
#include <OneWire.h>                     //librería para protocolo OneWire

//-----Declaramos constantes-----
#define BUS_ONE_WIRE 3                   //pin de comunicación OneWire
#define TX_NUM 24                         //numero de bytes a enviar al coordinador
#define RX_NUM 17                         //numero de bytes recibidos
#define TX_BYTE 6                         //bytes a enviar en el frame de los sensores

//-----Declaramos variables-----
float Temp;                              //variable para almacenar temperatura
byte TX_DATA[TX_NUM];                    //creamos el frame para enviar datos al xbee
byte SENSORS[TX_BYTE];                   //array para almacenar los bytes de sensores a enviar
byte RX_DATA[RX_NUM];                    //array para almacenar los bytes recibidos
long int checksum=0;                     //variable para almacenar el checksum
short int Vol,Amp;                        //variable para almacenar valor de sensores

OneWire oneWire(BUS_ONE_WIRE);           //indicamos el pin a emplear en el protocolo
DallasTemperature sensors(&oneWire);     //vinculamos librería onewire con dallas

//Indicamos la dirección ROM del DS18B20:
DeviceAddress Sensor1= {0x28,0xE9,0x07,0x15,0x04,0x00,0x00,0xD7};
    
```

```

void setup(){
  Serial.begin(9600);           //establecemos comunicaci3n serial
  Serial1.begin(9600);         //establecemos comunicacion para el xbee
  sensors.begin();             //inicializamos la medicion en la libreria
  sensors.setResolution(Sensor1,10); //indicamos la resolucio del sensor1

  //Llenamos el frame de envio
  TX_DATA[0]=0x7E;
  TX_DATA[1]=0x00;
  TX_DATA[2]=0x14;
  TX_DATA[3]=0x10;
  TX_DATA[4]=0x00;           //indicamos el ID del frame
  TX_DATA[13]=0xFF;         //indicamos la direccio de red
  TX_DATA[14]=0xFE;
  TX_DATA[15]=0x00;         //cantidad de saltos
  TX_DATA[16]=0x00;}        //opciones de transmision

void loop(){
  Vol=analogRead(0);         //leemos el valor del sensor de voltaje
  Amp=analogRead(1);         //leemos el valor del sensor de corriente
  sensors.requestTemperatures(); //comando para leer la temp de dispositivos
  Temp=sensors.getTempC(Sensor1); //obtenemos la temp del sensor1

  SENSORS[0]=(int(Temp*100))>>8&0xFF; //obtenemos MSB de temp
  SENSORS[1]=(int(Temp*100)); //obtenemos LSB de temp
  SENSORS[2]=Vol>>8&0xFF; //obtenemos MSB de voltaje
  SENSORS[3]=Vol&0xFF; //obtenemos LSB de voltaje
  SENSORS[4]=Amp>>8&0xFF; //obtenemos MSB de corriente
  SENSORS[5]=Amp&0xFF; //obtenemos LSB de corriente

  if(Serial1.available(>0){ //si hay datos en el buffer de entrada

    for(int i=0;i<RX_NUM;i++){ //almacenamos todos los datos recibidos
      RX_DATA[i]=Serial1.read();
      Serial.print(RX_DATA[i],HEX);}
    Serial1.clear(); //descartamos la info que no fue almacenada

    if (RX_DATA[3]==0x90&RX_DATA[15]==0x01)//verificamos si recibimos lo deseado
    {Serial.println("ok");
      for(int i=17;i<TX_NUM-1;i++){ //almacenamos los valores de los sensores
        TX_DATA[i]=SENSORS[i-17];}

      for(int i=5;i<13;i++){ //almacenamos la direccio del xbbe con quien
        TX_DATA[i]=RX_DATA[i-1];} // nos vamos a comunicar

      for(int i=3;i<TX_NUM-1;i++){ //obtenemos la suma de los bits
        checksum=checksum+TX_DATA[i];}
      TX_DATA[TX_NUM-1]=0xFF-(checksum&0xFF);//Almacenamos el checksum

      for (int i=0;i<TX_NUM;i++){ //enviamos los datos solicitados
        Serial.print(TX_DATA[i],HEX);
        Serial1.write(TX_DATA[i]);}
      }
    Serial.print(' ');}
  checksum=0; //resetamos el checksum
  Serial1.clear(); //limpiamos el buffer de recepcio
  /*=====
                               FIN
  =====*/

```

Como se puede observar en el c3digo presentado, el microcontrolador ejerce dos funciones dentro del m3dulo de sensado. La primera de estas consiste en realizar la captura de informaci3n de las magnitudes de voltaje, corriente y temperatura, para posteriormente llevar a cabo un promedio de las mismas, proceso que se

repite iterativamente en intervalos de 5 segundos hasta que el Coordinador solicite la información. La segunda función se centra en establecer la comunicación inalámbrica con el coordinador a través del módulo Router Xbee, lo cual se lleva a cabo mediante la implementación del protocolo UART enviando un paquete de datos con un contenido de 24 Bytes a una tasa de transferencia de 96 kbps (figura 6).

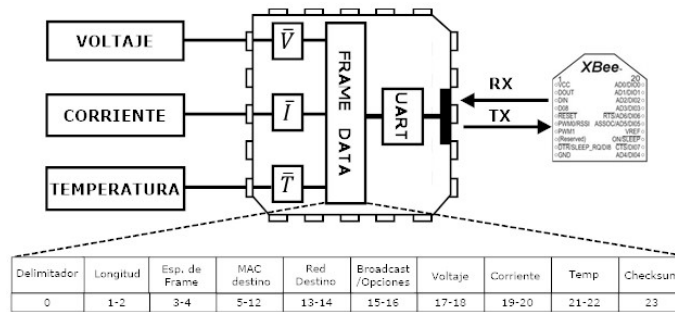


Figura 6 Funciones del microcontrolador CY8C4125AXI-483 en módulo de sensado.

Servidor Web

La implementación del servidor web se realiza a través de la tarjeta de desarrollo Raspberry Pi debido a sus ventajas respecto a las computadoras convencionales como lo son su bajo costo, pequeñas dimensiones y mínimo consumo energético [7].

Por otro lado, además de la función de servidor web, la Raspberry Pi también se encuentra encargada de la recepción de información proveniente de los módulos de sensado así como de establecer la conexión UART con el Xbee Coordinador. La programación realizada en la Raspberry Pi se llevó a cabo mediante el lenguaje de programación Phyton, permitiendo el almacenamiento de la información recibida. en una base de datos MySQL. Lo anterior cual en conjunto con el uso del servidor web HTTP de código abierto Apache, implementado en la tarjeta Raspberry Pi, permite la visualización de la información transmitida, y su respectiva hora de medición, a través de cualquier dispositivo conectado a la red [8,18], El código de programación es descrito a continuación e ilustrado en la figura 7:

```

"""----- IMPORTAMOS LIBRERIAS -----"""
import MySQLdb                               #importamos libreria para crear db en mysql
    
```

```
import serial                                #importamos libreria para comunicacion serial
import time                                  #importamos libreria para delays
import sys                                    #importamos libreria del sistema

"""----- DATOS PARA LA COMUNICACION LA BD ----- """
DB_HOST='localhost'                         #indicamos el host de la base de datos
DB_USER='xxxx'                               #indicamos el nombre de usuario de la bd
DB_PASS='xxxxxx'                             #indicamos la contraseña de la base de datos
DB_NAME='Ensayo1'                             #indicamos el nombre de la base de datos

"""-----CONSTANTES Y VARIABLES----- """
RX_NUM=22                                    #numero de bytes a recibir
SENSOR_NUM=6                                 #numeros de byte recibidos para el sensor
TxBuffer=[]                                  #bufer de transmision
SensorBuffer=[0,0,0,0,0,0]                  #buffer para almacenar el valor de los sensores
checksum=0                                    #creamos el checksum para sumar bits

"""----- INICIAMOS SERIAL Y DB ----- """
ser=serial.Serial('/dev/ttyAMA0',9600,timeout=.5) #iniciamos la comunicacion por el puerto serie
bd=MySQLdb.connect(DB_HOST,DB_USER,DB_PASS,DB_NAME) #establecemos conexion con la base de
datos
cursor=bd.cursor()
cursor.execute("DROP TABLE IF EXISTS XbeeTable")
cursor.execute("CREATE TABLE XbeeTable(id INT PRIMARY KEY AUTO_INCREMENT,Voltaje
DECIMAL(4,2),Corriente DECIMAL(4,2),Temperatura DECIMAL(5,3),Fecha VARCHAR(8),Hora VARCHAR(8))")

#llenamos los buffers de salida
TxBuffer.append(0x7E)
TxBuffer.append(0x00)
TxBuffer.append(0x0F)
TxBuffer.append(0x10)
TxBuffer.append(0x00) #cambiamos el ID para poder enviar a varios
TxBuffer.append(0x00) #cambiamos el MAC adress para poder en viar a
varios
TxBuffer.append(0x13)
TxBuffer.append(0xA2)
TxBuffer.append(0x00)
TxBuffer.append(0x40)
TxBuffer.append(0xA8)
TxBuffer.append(0x1F)
TxBuffer.append(0xDE) #fin de la direccion MAC
TxBuffer.append(0xFF)
TxBuffer.append(0xFE)
TxBuffer.append(0x00)
TxBuffer.append(0x00)
TxBuffer.append(0x01) #byte de solicitud
TxBuffer.append(0x00)

TX_NUM=len(TxBuffer)
CHECKNUM=TX_NUM-1

"""----- FUNCION PARA CONVERTIR BYTES A NUMERO ENTERO-----"""
def BytesToInt(x):
    for i in range(SENSOR_NUM):
        #convertimos los caracteres de entrada a enteros
        SensorBuffer[i]=int(x[i+15].encode('hex'),16)
    y=(SensorBuffer[0]<<8)|SensorBuffer[1]
    if y>=32768:
        y=-1*(y^0xffff)
    u=(SensorBuffer[2]<<8)|SensorBuffer[3]
    if u>=32768:
        u=-1*(u^0xffff)
    v=(SensorBuffer[4]<<8)|SensorBuffer[5]
    if v>=32768:
        v=-1*(v^0xffff)
    return y,u,v

try:
    while True:
```

```

#colocamos un switch case para cambiar los valores del xbee con el que
#nos queremos comunicar.

for i in range(3,CHECKNUM):          #obtenemos el checksum
    checksum=checksum+TxBuffer[i]
TxBuffer[CHECKNUM]=(0xFF-(checksum&0xFF))

print (TxBuffer)                    #visualizamos lo que vamos a enviar

for i in range(TX_NUM):              #enviamos la informacion
    ser.write(chr(TxBuffer[i]))
data=ser.read(RX_NUM)               #recibimos informacion

print data.encode('hex')            #visualizamos la informacion recibida

temp,volt,amp=BytesToInt(data)
temp=temp/100.0
volt=volt*80/1023
amp=amp*6/1023
print temp
print volt
print amp

fecha=time.strftime("%x")           #guardamos el valor actual de la fecha
hora=time.strftime("%X")           #guardamos el valor actual de la hora
cursor.execute("INSERT INTO XbeeTable(Voltaje,Corriente,Temperatura,Fecha,Hora)
VALUES(%f,%f,%f,%s,%s)"%(volt,amp,temp,fecha,hora))
# bd.commit()

checksum=0                          #reseteamos el checksum
ser.flushInput()                    #limpiamos el buffer de entrada
time.sleep(5)
bd.commit()

except KeyboardInterrupt:          #condicion de interrupcion
ser.flushInput()
ser.close()                        #cerramos la comunicacion serial
bd.commit()
    
```

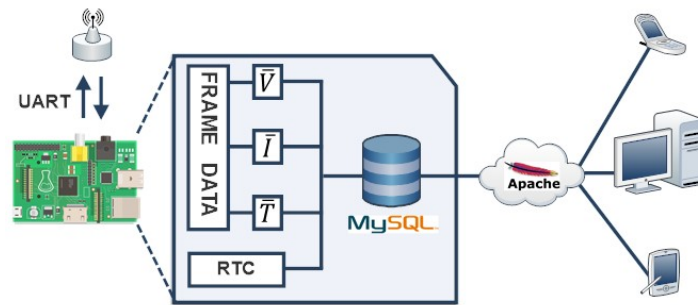


Figura 7 Implementación de la tarjeta de desarrollo Raspberry Pi como servidor Web.

3. Resultados

Para llevar a cabo las pruebas correspondientes el sistema de monitoreo fue probado en las instalaciones de la Facultad de Ingeniería de la Universidad Autónoma de Yucatán. Para establecer a la tarjeta Raspberry como servidor web, esta fue integrada al sistema de Red de Área Local (LAN por sus siglas en inglés). Del mismo modo las pruebas fueron realizadas empleando un módulo fotovoltaico

mono cristalino de 20 V y 4.5 A. Los resultados de la interfaz web tras el empleo del sistema de monitoreo fotovoltaico son mostrados en la figura 8.

Como se puede observar dicha figura presenta una toma de pantalla de la página web generada a partir de los resultados obtenidos una vez realizada la implementación del sistema de monitoreo inalámbrico dentro de la LAN. Las mediciones realizadas son presentadas en un formato de tabla, en el intervalo de tiempo especificado. Es importante indicar que durante la operación del sistema no se registraron pérdidas de datos, lo cual se puede corroborar con los gráficos presentados en figuras 9 a 11 correspondientes a mediciones de voltaje, corriente y temperatura durante un día estándar.


Instituto Tecnológico Superior Progreso
Universidad Autónoma de Yucatán
 "Sistema de Monitoreo de Paneles Fotovoltaicos"

FECHA	HORA	VOLTAJE (V)	CORRIENTE (A)	TEMPERATURA (°C)
13/05/15	10:43	14.1	3.3	68.8
13/05/15	10:53	14.8	3.5	71.4
13/05/15	11:03	15.1	3.5	72.8
13/05/15	11:13	15.4	3.45	74.6
13/05/15	11:23	16.2	4.00	77.0
13/05/15	11:33	17.5	4.82	79.2
13/05/15	11:43	17.9	4.80	80.2
13/05/15	11:53	18.1	4.65	82.7
13/05/15	12:03	19.3	4.27	83.8
13/05/15	12:13	19.5	4.30	83.7
13/05/15	12:23	19.2	4.15	84.8
13/05/15	12:33	19.5	4.15	83.8
13/05/15	12:43	19.6	4.23	84.0
13/05/15	12:53	19.6	4.20	83.9
13/05/15	12:53	19.3	4.10	78.9
13/05/15	13:03	19.4	4.15	81.8
13/05/15	13:13	19.0	3.99	85.4
13/05/15	13:23	18.9	3.89	83.2
13/05/15	13:33	19.1	3.95	85.0

Figura 8 Visualización de mediciones del panel fotovoltaico (página Web generada).

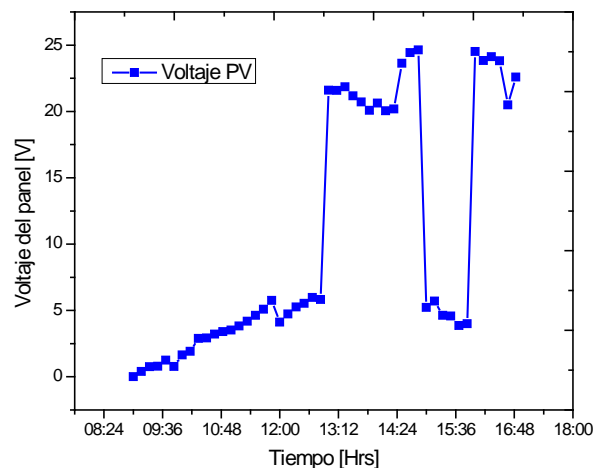


Figura 9 Mediciones de voltaje del panel fotovoltaico durante un día estándar.

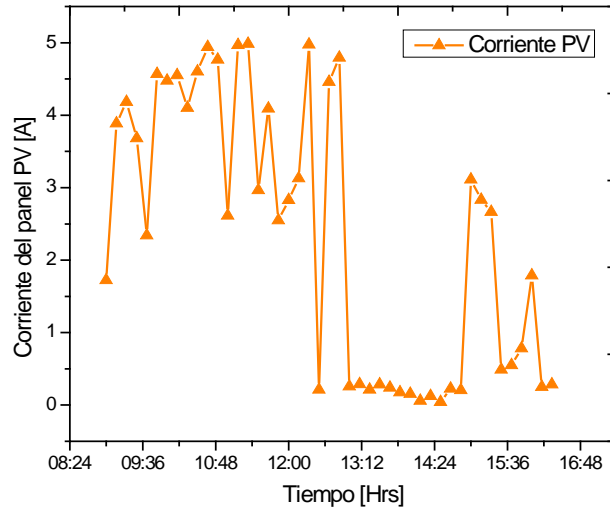


Figura 10 Mediciones de corriente del panel fotovoltaico durante un día estándar.

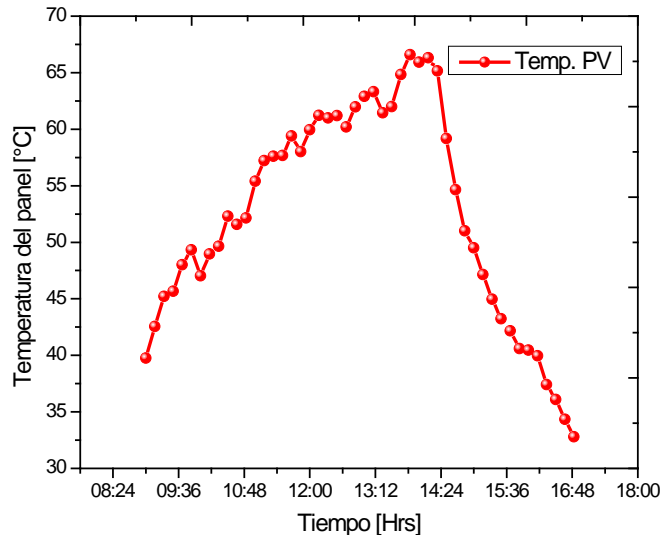


Figura 11 Mediciones de temperatura del panel fotovoltaico durante un día estándar.

Una vez comprobado el correcto funcionamiento del sistema, una evaluación económica del costo del mismo fue llevada a cabo. La tabla 2, presenta la información de costos de construcción de la red de monitoreo realizando una comparativa entre los componentes de última generación empleados como lo son la tarjeta Raspberry Pi y el micro controlador PSoC CY8C4125AXI-483 respecto a las opciones convencionales para el desarrollo en el ámbito de investigación de este tipo de sistemas. Como se puede observar la implementación de la Tarjeta Raspberry Pi disminuye ampliamente los costos de inversión y desarrollo del nodo

sensor aun considerando el empleo de la misma tecnología de transferencia inalámbrica. Por otra parte, el uso de micro controladores de última generación no convencionales supera en demasía los beneficios económicos y tecnológicos, respecto a empleo de equipos comercialmente aceptados en la actualidad como lo son las tarjetas de desarrollo Arduino. Por lo tanto se concluye que el uso de los componentes propuestos presenta una opción viable y mucho más económico que otras opciones disponibles para el desarrollo del sistema.

Tabla 2 Costos comparativos de construcción de los nodos sensor y coordinador.

	Elementos empleados	\$ (USD)	Elementos convencionales	\$ (USD)
Nodo sensor	Raspberry	29.00	Computadora	240.00
	Modulo Xbee	32.00	Módulo Xbee	32.00
	Total	61.00	Adaptador Xbee-Pc	20.50
			Total	292.50
Nodo coordinador	CY8C4125AXI-483	4.00	Arduino UNO	11.00
	Módulo Xbee	23.00	Módulo Xbee	32.00
	Sensores	19.00	Sensores	19.00
	Total	55.00	Total	62.00

4. Conclusiones

La red inalámbrica para el monitoreo de equipos fotovoltaicos propuesto fue desarrollado con éxito. El empleo del módulo con tecnología Zeegbee permitió establecer comunicación a grandes distancias con un protocolo de bajo nivel posibilitando controlar todos los aspectos de la transmisión y recepción de datos. Del mismo modo, la substitución de una computadora convencional por un sistema embebido Raspberry Pi y el empleo del micro controlador CY8C4125AXI-483 con arquitectura ARM de 32 bits permitieron alcanzar de manera eficiente los objetivos planteados para el presente proyecto, reduciendo los costos de los componentes y minimizando el consumo energético del mismo.

5. Bibliografía y Referencias

- [1] V. Tyagi, N. Rahim, J. Selvaraj, "Progress in solar PV technology: Research and achievement". Renewable and Sustainable Energy Review. Vol. 20. Pp. 443-461.

- [2] M. Tzuc, O. Bassam, A. Flota-Bañuelos, M. Ordoñez E, Ricalde-Cab, L. Quijano, R. Vega-Pasos, Thermal Efficiency Prediction of a Solar Low Enthalpy Steam Generating Plant Employing Artificial Neural Networks. 2016. *Intelligent Computing Systems*. Pp. 61-73.
- [3] K. Solangi, M. R. Islam, R. Saidur, N. A. Rahim, H. Fayaz, "A review on solar energy policy". *Renewable and Sustainable Energy Review*. Vol. 15. 2011. Pp. 2149-2163.
- [4] F. Shariff, N. A. Rahim, H. Wooi Ping, "Zigbee-based data acquisition system for online monitoring of grid-connected photovoltaic system". *Expert Systems with Applications*. Vol. 42. 2015. Pp. 1730-1742.
- [5] P. Papageorgas, D. Piromalis, K. Antonakoglou, G. Vokas, D. Tseles, K. G. Arvanitis, "Smart solar panels: In-situ monitoring of photovoltaic panels based on wired and wireless sensor networks". *Energy Procedia*. Vol. 36. 2013. Pp. 535-545.
- [6] R. Kaamal, *Embedded Systems*. 2008. Mc-GrawHill. USA.
- [7] O. May Tzuc, R. Quijano Cetina, F. Fernández Castillo, *Sistema Domótico de control de luces mediante una aplicación Android basada en una tarjeta Raspberry-Pi. Tendencias de Investigación y Desarrollo*. 1 Ed. 2015.
- [8] M. Romero A. *Implementación de servicios de red mediante una Raspberry Pi*. 2014. Zaragoza, España.
- [9] S.Monk, *Raspberry Pi cookbook*. 2014. O'Reilly. USA.
- [10] L-C. Huang, H-C. Chang, C-C. Chen, C-C. Kuo. "A ZigBee-based monitoring and protection system for building electrical safety". *Energy Build*. Vol. 43. 2011. Pp: 1418-1423.
- [11] NC. Batista, R Melício, JCO. Matias, JPS. Catalão, "Photovoltaic and wind energy system monitoring and building/home energy management using ZigBee devices within a smart grid". *Energy*. Vol. 49. 2013. Pp. 306-315.
- [12] W-T. Sung, Y-C. Hsu. "Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee". *Expert Systems with Applications*. Vol. 38. 2011. 4522-4529.

- [13] C. Buratti, A. Conti, D. Dardari, R. Verdone, "An Overview on Wireless Sensor Networks Technology and Evolution". Basel, Switzerland, *Sensors-Open Acces Journal*. Vol. 9. 2009. Pp. 6869-6896.
- [14] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson. *Wireless Sensor Networks for Habitat Monitoring*. Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications. 2002. New York, USA.
- [15] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, D. Ganesan, *Building Efficient Wireless Sensor Networks with Low-Level Naming*. Proceedings of the eighteenth ACM symposium on Operating systems principles. 2009. New York, USA.
- [16] A. Oyarce, *Guía de usuario Xbee serie 1*. MCI Ltda. Santiago de Chile. 2010.
- [17] R. Fallody, *Building Wireless Sensor Networks*. O'Reilly. USA. 2011.
- [18] R. González Duque, *Python para todos*. Creative Common. España.

6. Autores

M.I. Oscar de Jesús May Tzuc, estudió en la Universidad Autónoma de Yucatán obteniendo el grado de Ingeniero Físico con especialidad en Instrumentación y Control electrónico en 2014. En 2016 obtuvo el grado de Maestro en Ingeniería con especialidad en Energías Renovables por parte de la Universidad Autónoma de Yucatán. Actualmente es profesor del Instituto Tecnológico Superior Progreso, sus áreas de interés son los sistemas embebidos, instrumentación y control electrónico, sistemas fototérmicos y aplicación de herramientas de inteligencia artificial.

M.I. Renán Gabriel Quijano Cetina, es egresado por parte del Instituto Tecnológico de Mérida en el cual obtuvo el grado de Ingeniero Electrónico. Maestro en Ingeniería Mecatrónica por parte de la Universidad Modelo, actualmente es profesor de tiempo completo en la Universidad Autónoma de Yucatán en el departamento de Ingeniería Mecatrónica. Entre sus áreas de interés se encuentran los sistemas embebidos, automatización y sistemas de control automático.

M.C. Juliana González Quijano es Maestra en Ciencias con especialidad en Biotecnología, por parte de la Universidad Autónoma de Yucatán. Es profesora de tiempo completo en el Instituto Tecnológico Superior Progreso. Entre sus intereses se encuentran el aprovechamiento sustentable de fuentes energéticas de energía, procesos de producción en biocombustibles y sistemas alternativos de energía.

Dr. José Daniel Ortiz López es estudiante de octavo semestre de la Licenciatura en Ingeniería en Energías Renovables, en el Instituto Tecnológico Superior Progreso.