

Sistema De Detección De Fuego y Monitorización De Temperatura Usando IoT

J.I. Vega-Luna¹, G. Salgado-Guzmán¹, M.A. Lagos-Acosta¹, J.F. Cosme-Aceves¹, F.J. Sánchez-Rangel¹

Resumen—Se presenta un sistema de alerta de fuego y umbral de temperatura usando IoT. El objetivo fue activar un altavoz y transmitir un SMS y un mensaje de WhatsApp a un teléfono móvil al detectar fuego y/o la temperatura en un centro de datos alcance el nivel de umbral. La metodología usada para construir el sistema fue usar un detector de fuego, un sensor de temperatura, una tarjeta de desarrollo Pyboard y un módulo WiFi. El microcontrolador de la Pyboard ejecuta un programa realizado en MicroPython para transmitir periódicamente a la plataforma de IoT ThinkSpeak el valor de temperatura y estado del sensor de fuego. Cuando se activa el sensor de fuego y/o el valor de temperatura alcanza el valor de umbral configurado, ThinkSpeak solicita a la plataforma de comunicaciones Twilio el envío de los mensajes de alerta. La comunicación de la tarjeta Pyboard a la Internet se realizó usando un punto de acceso WiFi ubicado en el centro de datos. El alcance logrado en la comunicación entre el sistema construido y el punto de acceso fueron 52 metros.

Palabras claves—MicroPython, Pyboard, ThinkSpeak, Twilio, WiFi.

Abstract—This paper presents a fire warning system and temperature threshold using IoT. The goal was to activate a loudspeaker and transmit an SMS and WhatsApp message to a mobile phone when it detects fire and/or the temperature in a data center reaches the threshold level. The methodology used to build the system was to use a fire detector, a temperature sensor, a Pyboard development card and a WiFi module. The microcontroller of the Pyboard runs a program made in MicroPython to periodically transmit the temperature and state of the fire sensor to the IoT ThinkSpeak platform. When the fire sensor is activated and/or the temperature value reaches the configured threshold value, ThinkSpeak requests the Twilio communications platform to send the alert messages. The communication of the Pyboard card to the Internet was made using a WiFi access point located in the data center. The reach achieved in the communication between the built system and the access point was 52 meters.

Keywords—MicroPython, Pyboard, ThinkSpeak, Twilio, WiFi.

I. INTRODUCCIÓN

En la actualidad, casi en cualquier lugar se encuentran instalados sensores electrónicos para monitorizar alguna variable. Los sensores electrónicos se utilizan en hogares, oficinas, industrias, laboratorios, teléfonos móviles, automóviles, infraestructura de ciudades y centros de datos. Se emplean para detectar y obtener información de una variedad de procesos y dispositivos, midiendo temperatura, humedad y presión, entre otras variables. La información obtenida por los sensores es transmitida como un valor numérico o una señal eléctrica a un dispositivo central de monitorización. El dispositivo central puede estar ubicado en la Internet, permitiendo la conexión digital de objetos de uso cotidiano a la misma, esto es, conectar el mundo físico con el digital por medio de computadoras y plataformas web en la nube que procesan y almacenan la información transmitida por sensores.

Esta conexión digital de objetos cotidianos con la Internet es lo que forma el concepto de Internet de las Cosas (IoT). La IoT es una infraestructura de red inteligente que permite realizar la monitorización de variables y automatización de procesos de forma eficiente y segura. Con IoT se conecta una gran cantidad de dispositivos embebidos, o cosas, a la Internet, los cuales se comunican con personas y otras cosas proporcionando datos de sensores que se almacenan, procesan y analizan en la nube [1].

Los recientes avances tecnológicos en electrónica, sistemas digitales y comunicaciones han impulsado el desarrollo de la IoT. Con estos avances, hoy se tienen disponibles sensores más pequeños y eficientes, tarjetas digitales con microcontroladores e interfaces de conexión inalámbrica a la Internet y sistemas operativos que pueden ejecutar programas en lenguajes como Python. Este desarrollo ha sido fortalecido con la disponibilidad de proveedores de servicios de almacenamiento y análisis de información en la nube que hasta hace unos pocos años no existían. Estos servicios permiten recibir información de sensores, almacenarla y procesarla para realizar una acción sobre un proceso remotamente. Adicionalmente, pueden transmitir mensajes de voz, texto, video o de WhatsApp a un teléfono móvil [2].

¹ Universidad Autónoma Metropolitana-Azcapotzalco, Departamento de Electrónica, Área de Sistemas Digitales, Av. San Pablo 180, Colonia Reynosa, C.P. 02200, Ciudad de México, México.

* vlji@azc.uam.mx.

El trabajo aquí presentado se realizó para un centro de datos. El objetivo fue contar con sistema que active un altavoz y transmita un SMS y un mensaje de WhatsApp al teléfono móvil del administrador cuando se detecte fuego o la temperatura del centro de datos alcance un valor de umbral. Se solicitó que el sistema digital almacene en un servidor ubicado en la Internet el valor de temperatura leído cada 30 segundos y que esta información esté disponible a las entidades que auditan la operación del centro de datos. El centro de datos cuenta con un punto de acceso WiFi a la Internet localizado a 30 metros del punto más lejano del mismo.

En la actualidad, continúan realizándose numerosos trabajos e investigaciones para monitorización y control de variables como temperatura, humedad y presión. Estos trabajos surgen de una necesidad específica, cumplen su objetivo y usan los últimos avances de la tecnología como redes neuronales [3], comunicación Ethernet o inalámbrica como ZigBee [4] y sensores de fibra óptica [5] o de ondas acústicas de superficie [6]. Estas aplicaciones reportan y almacenan información en una computadora ubicada localmente en el proceso bajo supervisión y no se consideran aplicaciones de IoT.

Siendo la IoT uno de los campos de rápido crecimiento, las aplicaciones de monitorización de variables han integrado capacidades remotas. Los trabajos de este tipo llevados a cabo los últimos años se han centrado en las áreas de la salud [7]-[9], monitorización de variables en hogares [10]-[11], agricultura [12], almacenamiento de agua [13], gases [14] y granos [15] y seguimiento de flotillas de vehículos en ciudades [16], entre otras. Sin embargo, estas aplicaciones de IoT reportan la información colectada a un teléfono móvil o a una computadora ubicada en la Internet cuya instalación, mantenimiento, programación y costo es parte de la implantación. No hacen uso de proveedores de servicios en la nube que hacen más eficiente la aplicación y disminuyen costos y tiempo de desarrollo.

Tomando en cuenta lo anterior, no se han realizado hasta ahora trabajos similares al aquí presentado. A través de una tarjeta Pyboard, se monitoriza la temperatura y la presencia de fuego en el centro de datos y se transmiten periódicamente los valores leídos a la plataforma ThingSpeak. Cuando se detecta fuego y/o la temperatura alcanza el valor de umbral configurado, el programa que se ejecuta en la Pyboard transmite el SMS y el mensaje de WhatsApp usando los servicios de la plataforma Twilio.

Se usó la Pyboard porque el microcontrolador de esta tarjeta puede ejecutar programas realizados en Micropython, lo cual facilitó el desarrollo de la programación al no usar un lenguaje complejo de bajo nivel, ya que existe una gran cantidad de bibliotecas de

funciones de libre uso y código abierto en la nube para realizar una variedad de tareas.

Micropython es una implantación del lenguaje de programación Python 3 e incluye un conjunto grande de bibliotecas estándares de Python. Micropython fue creado con el objetivo de ser compatible con Python y transferir fácilmente código desarrollado en una computadora a un microcontrolador o a un sistema embebido. Ha sido optimizado para ejecutarse en microcontroladores. Existen varios tipos de tarjetas de desarrollo con microcontroladores que pueden ejecutar Micropython, una de ellas es la tarjeta Pyboard.

ThingSpeak es una plataforma de servicios de análisis de información en la nube que permite coleccionar datos de sensores instalados en dispositivos inteligentes, almacenarla, visualizarla y analizarla. En ThingSpeak se puede ejecutar código de MATLAB para analizar históricamente la información recibida o generar una acción remotamente con aplicaciones web y móviles, redes sociales como Twitter, soluciones de mensajería, VoIP, servicios en la nube como Twilio y dispositivos con hardware de código abierto como Arduino, Raspberry Pi y Pyboard.

Para aplicaciones de IoT que transmiten al servidor de la nube hasta 3 millones de mensajes por año, el servicio de ThingSpeak no tiene costo. Cada mensaje puede contener hasta 8 valores de diferentes sensores o variables. Para cantidades mayores a 3 millones de mensajes, ThingSpeak ofrece diferentes tipos de licencias de acuerdo a la cantidad a transmitir.

Twilio es una plataforma de servicios basada en API (Application Programming Interface) que permite transmitir mensajes de voz, video y texto a aplicaciones web, móviles y de escritorio. El uso gratuito de Twilio permite enviar hasta 100 mensajes al mes. Para más de 100 mensajes, Twilio ofrece diferentes planes en los que el costo promedio del mensaje es 0.00025 USD.

Los beneficios y aportaciones del sistema aquí presentado son las siguientes: 1) La programación se desarrolló en Micropython, lo cual reduce el tiempo, complejidad y costo de desarrollo del sistema, 2) Se usó una tarjeta Pyboard de última tecnología y precio accesible la cual no requiere un sistema operativo y ofrece mayor cantidad de recursos que otras de su tipo como Arduino o Raspberry, 3) Se usaron plataformas de IoT en la nube de bajo costo y reciente creación que proporcionan el servicio para almacenar la información de los sensores y 4) El administrador es notificado inmediatamente de la alarma del centro de datos.

II. PARTE TÉCNICA DEL ARTÍCULO

La metodología usada para el diseño del sistema consistió

en dividirlo en seis módulos y seleccionar los componentes adecuados, tomando en cuenta lo planteado en el objetivo, para implantar cada módulo. Los módulos que integran al sistema son los siguientes: el módulo de procesamiento, el sensor de temperatura, el detector de fuego, el módulo de activación del altavoz, la interfaz WiFi y la interfaz de usuario. En la Figura 1 se muestra el diagrama de bloques de la arquitectura del sistema y a continuación se describe el diseño y operación de cada módulo.

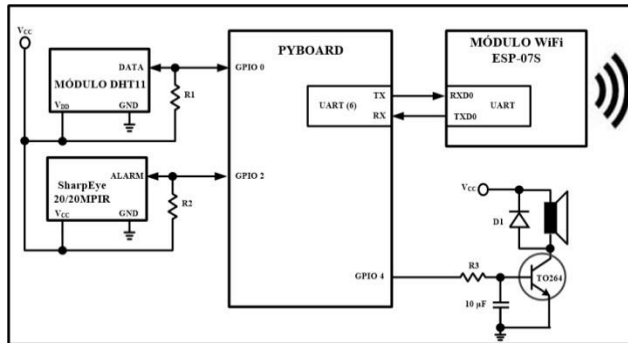


Figura 1. Arquitectura del sistema

A. El módulo de procesamiento

Este módulo está compuesto por la tarjeta Pyboard PYBV1.0. Actualmente, la tarjeta Pyboard es la más usada para ejecutar programas realizados en Mycropython. Cuenta con los siguientes recursos hardware: microcontrolador STM32F405RG con CPU Cortex M4 de 168 MHz y punto flotante, puerto USB, memoria flash ROM de 1024 KB, memoria RAM de 192 KB, ranura para tarjeta Micro SD, acelerómetro de 3 ejes (MMA7660), reloj de tiempo real, 29 terminales de entrada/salida de propósito general, 3 convertidores analógico/digital de 12 bits, 2 convertidores digital/analógico de 12 bits, 2 puertos UART, 2 puertos I²C, 4 puertos USART, programa de bootloader para actualización del firmware y se alimenta con 3.3V.

B. El sensor de temperatura

Para medir el valor de la temperatura se utilizó el módulo DHT11. Este módulo, de bajo costo y tamaño compacto, incorpora un sensor capacitivo de humedad y un termistor para medir la temperatura ambiente. Incluye un convertidor analógico-digital, un microcontrolador y una interface 1-wire para proporcionar el valor de temperatura y humedad relativa.

El voltaje de alimentación del DHT11 es de 3.3V a 5.5V, el rango de temperatura que puede medir es 0°C a 50°C, con una precisión de ±0.5°C y una resolución de 1°C. Cuenta con 4 terminales, dos son tierra, otra es para

alimentación y la última es la línea de datos serie 1-wire mediante la cual entrega el valor digital de temperatura y humedad en una palabra de 40 bits. Esta última terminal se conectó a la terminal GPIO 0 de la tarjeta Pyboard configurada como entrada. De estos 40 bits, los primeros 16 bits corresponden al valor de humedad relativa, los siguientes 16 bits son el valor de temperatura en grados Celsius, 8 bits para la parte entera y 8 bits para la parte decimal, y los últimos 8 bits indican el checksum. La señal del bus 1-wire del DHT11 tiene alcance de 20 metros.

C. El detector de fuego

El detector de fuego utilizado es el dispositivo SharpEye 20/20MPIR. Este dispositivo es un detector de flama infrarrojo compacto, ligero, de bajo consumo de energía y para uso en interiores, que se puede configurar en uno de 4 rangos de sensibilidad. Mayor sensibilidad significa mayor alcance y mayor tiempo de respuesta como se indica en la Tabla I.

TABLA I
SENSIBILIDAD Y TIEMPO DE RESPUESTA DEL DETECTOR DE FUEGO

Rango de sensibilidad (ft/m)	33/10	65/20	100/30	140/43
Tiempo de respuesta (segs)	5	8	10	10

El campo de vista del SharpEye 20/20MPIR son 100° horizontal y 90° vertical. Su ambiente de operación es entre -40 y 70°C. Se alimenta con 12V y consume 15mA. Cuenta con una salida de relevador llamada Alarm la cual indica la presencia de fuego. Esta salida se conectó, a través de un divisor de voltaje, a la terminal GPIO 2 de la tarjeta Pyboard configurada como entrada.

En este trabajo se configuró el detector de fuego en el cuarto rango de sensibilidad para lograr el mayor alcance, 140 ft/43 m, con tiempo de respuesta de 10 segundos. Este tiempo de respuesta no impacta en la confiabilidad del sistema puesto que el detector se muestrea cada 30 segundos.

D. El módulo de activación del altavoz

El altavoz de alarma empleado es de 100w de alto rendimiento. Este altavoz se conectó a la terminal GPIO 4 de la tarjeta Pyboard configurada como entrada a través de un amplificador de corriente. Cuando el programa que ejecuta el microcontrolador determina que la salida Alarm del detector de fuego se activó y/o el valor de temperatura alcanza el umbral establecido, ejecuta una rutina que genera el tono de alerta. La rutina que produce la señal de alarma trabaja de la forma siguiente: genera una onda cuadrada con una frecuencia de 440Hz y una duración de 500ms, a continuación se detiene 500ms, posteriormente genera una onda cuadrada con una frecuencia de 520Hz durante 400ms y después de 500ms repite este proceso.

E. La interfaz WiFi

La interfaz WiFi usada es el módulo TinySine WiFi Skin for Pyboard. Este módulo trabaja en base al circuito ESP-07S, el cual pertenece a la familia ESP8266. El UART del ESP-07S se conectó a uno de los UART de la tarjeta Pyboard para llevar a cabo la comunicación.

El programa que ejecuta el microcontrolador de la tarjeta Pyboard realiza las siguientes tareas: 1) Inicializa terminales de entrada/salida, temporizadores, UART, interface WiFi y establece el valor del umbral de temperatura y número de teléfono móvil del administrador del centro de datos, 2) Entra a un ciclo en el que invoca dos funciones. La primera función se ejecuta en segundo plano e implanta la interfaz de usuario. La segunda función lee periódicamente, cada 30 segundos, en la terminal GPIO 0 el valor de temperatura entregado por el DHT11 y lee en la terminal GPIO 2 el estado del detector de fuego. A continuación, convierte de binario a decimal los 40 bits que entrega el DHT11 y ejecuta la rutina que usa la REST API de escritura para enviar a ThinkSpeak, a través de un puerto USART de la tarjeta Pyboard, el mensaje con el valor de temperatura, estado (activado/desactivado) del detector de fuego y la fecha y hora. Si se activó el detector de fuego y/o el valor de temperatura alcanzó el umbral, se ejecuta la rutina que activa la señal de alarma en el altavoz y usa la REST API de Twilio para transmitir el SMS y mensaje de WhatsApp. La temporización de eventos del bus 1-wire que implanta el programa para leer la temperatura en la terminal GPIO 0 se indica en la Figura 2.

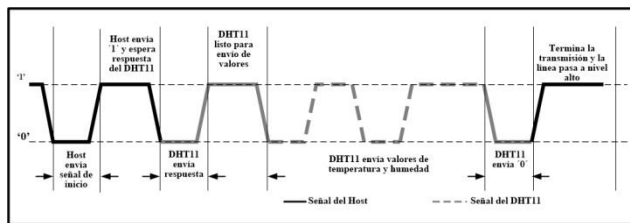


Figura 2. Temporización de señales 1-wire entre la Pyboard y el DHT11

En las aplicaciones de IoT se usan varios protocolos para el intercambio y manipulación de datos, entre los que se encuentran MQTT (Message Queuing Telemetry Transport), XMPP y REST (Representational State Transfer). Los más usados son MQTT y REST y la mayoría de proveedores de servicios de Internet como Twitter, Facebook, Youtube, ThinkSpeak y Twilio tienen disponibles MQTT API y REST API para que los usuarios puedan usarlos en la creación de aplicaciones de IoT. En el desarrollo de este trabajo se usaron REST API, debido a que trabajan como interfaz entre la programación del sistema que intercambia datos con la plataforma

ThinkSpeak utilizando el protocolo HTTP. Estos datos pueden estar como texto plano o en formato XML y JSON. Las REST API tienen la ventaja, sobre otros protocolos de IoT, que son independientes del tipo de plataforma y lenguaje de programación donde se invocan. REST es el estándar lógico, eficiente y habitual en la creación y uso de API en los servicios de Internet.

ThingSpeak usa canales para el intercambio de mensajes con las aplicaciones de IoT. Se usa una REST API para crear un canal y obtener el identificador del mismo y las claves de las REST API para escritura y lectura de datos. Una vez creado el canal, se intercambian mensajes que contienen: la ruta del REST API a usar, la clave de la REST API y hasta 8 campos que pueden tener cualquier tipo de dato, como por ejemplo el valor de una variable leída desde un sensor. Los mensajes usados en este trabajo tienen 3 campos: el valor de temperatura, el estado del detector de fuego y la fecha y hora. En la Figura 3 se muestra el código en Micropython de la rutina que usa la REST API de escritura para transmitir mensajes a ThingSpeak.

```
#!/usr/bin/python
import time
import os
import sys
import urllib
import urllib2
#
# Definicion de constantes
THINGSPEAKURL = 'https://api.thingspeak.com/update'
API-KEY = 'XXXXXXXXXXXXXXXXXXXX'
#
# Rutina de transmision de mensaje a ThinkSpeak
#
def
sendData(url,key,field1,field2,field3,temp,detector,fecha):
    valor-campos = {'api_key' : key,'field1' : temp,'field2' :
    detector,'field3' : fecha}
    postdata = urllib.urlencode(valor-campos)
    req = urllib2.Request(url, postdata)

    # Envía mensaje a Thingspeak con REST API de
    # escritura
    resp = urllib2.urlopen(req, None, 5)
    string_html = resp.read()
    resp.close()
#
# Programa Principal
#
def main():
    global temp
    global detector
    fecha = time.strftime("%d-%m-%Y,%H:%M:%S")
    sendData(THINGSPEAKURL,API-
    KEY,'field1','field2','field3',temp,detector,fecha)
```

```
if __name__ == "__main__":
    main()
```

Figura 3. Rutina de transmisión de mensajes a Twinspeak

Twilio es un servicio en la nube que proporciona, a través de un paquete, una biblioteca de funciones que pueden invocarse en un programa en Micropython para interactuar con las REST API de Twilio para la transmisión de SMS y mensajes de WhatsApp.

F. La interfaz de usuario

La interfaz de usuario está compuesta por el servidor web y la página del mismo, a través de la cual el usuario puede desplegar y descargar a un archivo de texto la información colectada por el sistema. La implantación del servidor web se basó en la biblioteca uasyncio. Esta biblioteca fue diseñada para realizar servidores web, a través de microcontroladores, usando la mínima cantidad de memoria RAM, llamados picowebs, así como para conexiones a bases de datos y colas de tareas distribuidas. A través de la página del servidor web, el usuario puede visualizar los valores, actuales e históricos, de temperatura y estado del detector de fuego. En la Figura 4 se muestra la pantalla principal de la interfaz de usuario realizada.

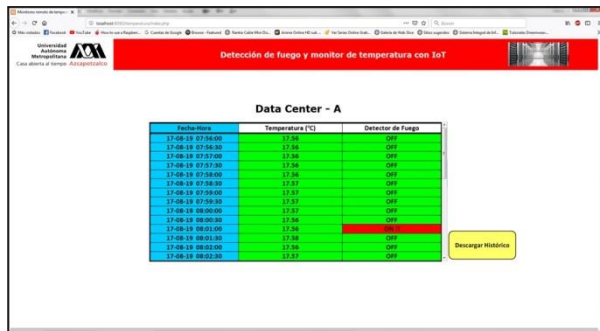


Figura 4. Pantalla principal de la interfaz de usuario

III. RESULTADOS

Se ejecutaron tres grupos de pruebas. El primer grupo tuvo como objetivo comprobar la exactitud de la temperatura medida por el sensor DHT22. Para realizar estas pruebas, se varió la temperatura del sensor artificialmente con un calentador y se midió con un termómetro analógico para compararla con la reportada en la interfaz de usuario. El fabricante indica una exactitud de $\pm 0.5^{\circ}\text{C}$, la cual se mantuvo hasta los 60°C . Después de este valor, al aumentar la temperatura, la diferencia de temperatura entre la medida con el termómetro y la reportada por el sensor aumentó proporcionalmente como se muestra en la Figura 5.

En este grupo de pruebas no se tuvo problema en la comunicación entre el sistema y el punto de acceso WiFi,

ya que el punto donde se instaló el sistema construido se ubica a 30 metros del punto de acceso. Sin embargo, se realizó un segundo grupo de pruebas para determinar el alcance del sistema.

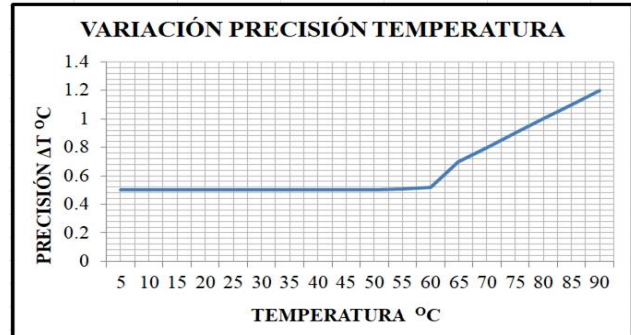


Figura 5. Precisión del sensor DHT22

Para llevar a cabo estas pruebas, se ubicó el sistema en diferentes posiciones en el centro de datos determinándose que la distancia a la cual se perdía el enlace con el punto de acceso fueron 52 metros con línea de vista.

El tercer grupo de pruebas tuvo como objetivo determinar el ángulo de cobertura del detector de fuego. Para llevar a cabo estas pruebas se fijó el detector usando el mejor ángulo de inclinación y respuesta indicado por el fabricante: 45° . A continuación, se generó fuego con un cilindro de gas en distintas ubicaciones, a la izquierda y derecha del ángulo de inclinación. Los resultados mostraron que el máximo ángulo de cobertura fue 47° a la izquierda y 48° a la derecha, un poco mayor a los especificados por el fabricante.

IV. DISCUSIÓN, CONCLUSIÓN Y RECOMENDACIONES

Se obtuvo un sistema de IoT para monitorizar remotamente presencia de fuego y temperatura, el cual reporta y almacena en una plataforma en la nube los valores de temperatura leídos y transmite SMS y mensajes de alerta a WhatsApp. El sistema realizado es fácil de instalar, usar y mantener. Es un sistema escalable al cual pueden integrarse sensores para coleccionar información de otro tipo de variables de ambiente realizando cambios mínimos en la programación del microcontrolador. El sistema puede usarse en otras instalaciones, no solo en un centro de datos como se utilizó en este trabajo. El fabricante del detector de fuego especifica que no es apto para usarse en ambientes donde exista interferencia originada por luz halógena o radiación de cuerpos negros como aquellos contruidos de carbón u hornos. Esto no impacta en la aplicación realizada en este trabajo, ya que este tipo de materiales y elementos no existen al interior de un centro de datos.

Esta aplicación se realizó usando el servicio gratuito de ThinkSpeak y Twilio los cuales permiten el envío de cierta cantidad de mensajes. Si la cantidad de mensajes a enviar es grande, deberá utilizarse una licencia de estos proveedores, lo cual incrementa un poco el costo de la aplicación.

Para aumentar el alcance de este sistema se recomienda instalar repetidores WiFi en el lugar donde se use o intercambiar la interfaz WiFi por el módulo Pycom WiPy cuyo rango de WiFi es 1Km, sin realizar cambios mayores en el hardware y programación del microcontrolador y aumentando el costo del sistema. Finalmente, existen alternativas para implantar una red de sensores conectados a la Internet y largo alcance como la tecnología LoRa. Los transceptores LoRa son de bajo costo y de varios Kilómetros, sin embargo, el costo de los puntos de acceso LoRa a la Internet es 1,000 USD.

V. REFERENCIAS

- [1] Halder, S. y Sivakumar, G. (2017). "Embedded based remote monitoring station for live streaming of temperature and humidity", in *Proceedings International Conference on Electrical, Electronics, Communication, Computer, and Optimization Technologies (ICECCOT)*, pp. 284-287.
- [2] Kadiyala, E., Meda, S. y Basani, R. (2017). "Global industrial process monitoring through IoT using Raspberry pi", in *Proceedings International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)*, pp. 260-262.
- [3] Huang, X., Li, Z. y Zhu, Y. (2018). "The System of Temperature Rise Monitoring and Temperature Prediction for Power Equipment", in *Proceedings Condition Monitoring and Diagnosis (CMD)*, pp. 1-5.
- [4] Rerktratn, A. y Kaewpoonsuk, A. (2015). "ZigBee based wireless temperature monitoring system for shrimp farm", in *Proceedings 15th International Conference on Control, Automation and Systems (ICCAS)*, pp. 428-431.
- [5] Hurezeanu, I., Nicola, C. I. y Sacerdotianu, D. (2016). "Temperature control and monitoring system for power transformer windings using fiber optic sensors", in *Proceedings International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, pp. 1-4.
- [6] Tan, H., Chen, X. y Ma, J. (2016). "A design of substation temperature online monitoring system based on SAW temperature sensor", in *Proceedings IEEE International Conference on High Voltage Engineering and Application (ICHVE)*, pp. 1-4.
- [7] Ashish, B. (2017). "Temperature monitored IoT based smart incubator", in *Proceedings International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, pp. 497-501.
- [8] Sundaravadivel, P., Mohanty, S. y Kougiianos, E. (2016). "An energy efficient sensor for thyroid monitoring through the IoT", in *Proceedings 17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems*, pp. 1-4.
- [9] Kumar, G. V., Bharadwaja, A. y Nikhil, N. (2017). "Temperature and heart beat monitoring system using IOT", in *Proceedings International Conference on Trends in Electronics and Informatics*, pp. 692-695.
- [10] Wu, F., Rüdiger, C. y Redouté, J. (2018). "WE-Safe: A wearable IoT sensor node for safety applications via LoRa", in *Proceedings IEEE 4th World Forum on Internet of Things*, pp. 144-148.
- [11] Shinde, D. y Siddiqui, S. (2018). "IOT Based Environment change Monitoring & Controlling in Greenhouse using WSN", in

Proceedings International Conference on Information, Communication, Engineering and Technology, pp. 1-5.

- [12] Nageswara, R. y Sridhar, B. (2018). "IoT based smart crop-field monitoring and automation irrigation system", in *Proceedings 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 478-483.
- [13] Raghu, K. y Harish, G. (2017). "Knowledge Based Real Time Monitoring System for Aquaculture Using IoT", in *Proceedings IEEE 7th International Advance Computing Conference (IACC)*, pp. 318-321.
- [14] Keshamoni, K. y Hemanth, S. (2017). "Smart Gas Level Monitoring, Booking & Gas Leakage Detector over IoT", in *Proceedings IEEE 7th International Advance Computing Conference (IACC)*, pp. 330-332.
- [15] Calderon, C., Gonzaga, M. y Morales, J. (2018). "Prototype industrial IoT applied to temperature monitoring in storage silos of dairy products", in *Proceedings 13th Iberian Conference on Information Systems and Technologies*, pp. 1-6.
- [16] Wang, S., Hou, Y. y Gao, F. (2016). "A novel IoT access architecture for vehicle monitoring system", in *Proceedings IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 639-642.

VI. BIOGRAFÍA



Vega-Luna José Ignacio. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1985. Maestría en Ciencias de la Computación, UAM-Azcapotzalco, Cd. de México, 1990. Labora actualmente en el área de Sistemas Digitales del Departamento de electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y sistemas operativos. El M. en C. Vega realiza investigación con redes inalámbricas de sensores y actuadores.



Salgado-Guzmán Gerardo. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1992. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y sistemas operativos. El Ing. Salgado realiza investigación con redes inalámbricas de sensores y actuadores.



Lagos-Acosta Mario Alberto. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1992. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y redes de computadoras. El Ing. Lagos realiza investigación con redes de computadoras.



Cosme-Aceves José Francisco. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1985. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Su línea de trabajo es lenguajes de descripción de hardware. El Ing. Cosme realiza investigación con sistemas embebidos y seguridad en redes de computadoras.



Sánchez-Rangel Francisco Rangel. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1987. Maestría en Ciencias de la Computación, UAM-Azcapotzalco, Cd. de México, 1999. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores.