

MONITORIZACIÓN DE PRESIÓN EN TUBERÍAS DE GAS INERTE USANDO IOT

Área técnica: Sistemas Digitales y Computación

José Ignacio Vega Luna, Mario Alberto Lagos Acosta, Francisco Javier Sánchez Rangel, José Francisco Cosme Aceves.

Área de Sistemas Digitales, Dpto. de Electrónica, Universidad Autónoma Metropolitana-Azcapotzalco.
Av. San Pablo 180, Col. Reynosa, C.P. 02200, Cd. de México.

vlji@azc.uam.mx

RESUMEN.

Se presenta un sistema de monitorización de presión en tuberías de gas inerte en un centro de datos usando una plataforma de IoT. El objetivo fue diseñar un sistema que reporte periódicamente a un servidor en la nube el valor de presión del gas en diferentes puntos de la tubería y visualizar por medio de la interfaz de usuario los valores de presión, en línea e históricamente. El sistema está compuesto por una red de cuatro nodos esclavos, que miden los valores de presión y determinan las coordenadas de su posición GPS, y un nodo maestro que colecta esta información y la transmite al servidor. Los nodos esclavos están compuestos por una tarjeta ESP32, el medidor de presión y el módulo GPS. El alcance logrado de la red WiFi de medidores de presión son 52 metros con línea de vista y la exactitud de la medida del sistema es $\pm 0.085\%$.

Palabras clave: Centro de datos, ESP32, GPS, IoT, presión de gas, WiFi.

ABSTRACT.

This paper presents a pressure monitoring system in inert gas pipes in a data center using an IoT platform. The objective was to design a system that periodically reports to a server in the cloud the gas pressure value in different points of the pipeline and visualize, through the user interface, the pressure values, online and historically. The system consists of a network of four slave nodes, which measure the pressure values and determine the coordinates of their GPS position, and a master node that collects this information and transmits it to the server. The slave nodes are composed of an ESP32 card, the pressure meter and the GPS module. The achieved reach of the WiFi network of pressure meters is 52 meters with line of sight and the accuracy of the measurement of the system is $\pm 0.0085\%$.

Keywords: Data center, ESP32, gas pressure, GPS, IoT, WiFi.

1. INTRODUCCIÓN

Los centros de datos son instalaciones que operan ininterrumpidamente las 24 horas del día. En un centro de datos se encuentra alojado equipo de procesamiento y almacenamiento de información y telecomunicaciones de empresas y organizaciones. Es importante mantener seguro este equipo ya que de él depende la productividad de clientes y usuarios.

En la construcción de los centros de datos se instalan dispositivos y mecanismos de seguridad para salvaguardar la infraestructura, los equipos e información y minimizar los riesgos a los que están expuestos. Uno de estos riesgos es el calor generado por el equipo de cómputo el cual puede generar fuego. Aunque el calor es disipado por sistemas de enfriamiento y ventilación, una de las medidas de seguridad más importantes en un centro de datos son los sistemas de extinción de incendios para combatir el fuego de manera rápida y segura ante cualquier eventualidad [1].

Existen diferentes tipos de procedimientos y sistemas de extinción de incendios para centros de datos. Estos sistemas se utilizan para extinguir el fuego lo más pronto posible con el mínimo daño posible al equipo y materiales alojados en estas instalaciones. Los procedimientos toman en cuenta tres aspectos: la detección de riesgos, la extinción del fuego y el restablecimiento de las operaciones después de la contingencia para reanudar las operaciones [2].

Las principales tecnologías usadas en los sistemas de extinción de incendios son tres: 1) Agentes líquidos, son gases en estado líquido en envases presurizados, 2) Agentes inertes, son gases naturales auto-presurizados como nitrógeno y argón y 3) Agua nebulizada, la cual consiste de gotas pequeñas usadas para atenuar fuego y calor. Los agentes líquidos y agua nebulizada usan rociadores y están diseñados para la protección de estructuras y personas. Sin embargo, el agua puede afectar el equipo de cómputo y generar más daño que el fuego [3].

La tendencia en los centros de datos construidos en años recientes, ha sido incorporar sistemas para extinción de fuego basados en gases inertes o reemplazar los sistemas anti-incendios por los que usan la tecnología de gases inertes [4]. Esta tecnología es más eficiente, limpia y segura para las personas y no afecta los equipos de cómputo. No destruye la capa de ozono y es mínimo el calentamiento global que produce. Los agentes inertes combaten la flama a nivel molecular desde la base, ya que sus componentes químicos y físicos absorben rápidamente la energía calorífica, disminuyendo la concentración de oxígeno para extinguir el fuego. La limpieza del sitio es más rápida después de la contingencia reduciendo tiempos de inactividad, inclusive el

equipo de cómputo puede continuar trabajando mientras se esparce el gas. Para el funcionamiento adecuado, la presión del gas inerte debe mantenerse estable en las tuberías a 300 bares. Los sistemas de extinción de fuego en base a gases inertes se utilizan también en: galerías de arte, museos, bibliotecas, almacenes de archivos y salas donde existe equipo de control o informático.

Los centros de datos son auditados periódica y permanentemente por instituciones y empresas que certifican, entre otras cosas, los sistemas de extinción de incendios para garantizar la calidad de servicios ofrecidos. Estos sistemas deben estar homologados bajo normas internacionales como: UL, FM, LPCB, ISO 14520, EN14520, NFPA 2001 y US EPA SNAP, entre otras [5]. Las certificaciones exigen que los centros de datos cuenten con sistemas de extinción de fuego automatizados cuyo mantenimiento y revisión sea permanente para la detección temprana e inequívoca de incendios potenciales al detectar humo y aumento de calor. Las medidas de mantenimiento incluyen el uso de válvulas electromagnéticas de apertura automática y la monitorización remota de la presión del gas inerte en las tuberías [6].

Con el desarrollo de la tecnología, en la actualidad se exige a los centros de datos la monitorización remota de variables, procesos y procedimientos desde la Internet para permitir a los responsables de la seguridad e instituciones certificadoras revisar en línea e históricamente la operación de los mismos [7]. Este requerimiento se resuelve con el desarrollo soluciones y aplicaciones que usan Internet de las Cosas (IoT).

La IoT es un concepto con el cual la información leída por sensores electrónicos usados en hogares, oficinas, industrias, laboratorios, teléfonos móviles, automóviles, infraestructura de ciudades y centros de datos, se transmite a un dispositivo central de monitorización. El dispositivo central está 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 almacenan y procesan la información transmitida por los sensores [8].

Esta conexión digital de objetos cotidianos con la Internet forma el concepto de 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 [9].

Los recientes avances tecnológicos en electrónica, sistemas digitales y comunicaciones han impulsado el desarrollo de la IoT. Con estos avances, hoy se tienen sensores más compactos y eficientes, tarjetas con microcontroladores e interfaces de conexión inalámbrica a la Internet y sistemas operativos que pueden ejecutar programas en lenguajes sencillos y poderosos como Python. Su 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 [10]. Estos servicios permiten recibir información de sensores, almacenarla y procesarla para realizar una acción sobre un proceso remotamente. Adicionalmente, existen servicios mediante los cuales se puede transmitir mensajes de alerta de voz, texto, video o de WhatsApp a un teléfono móvil para notificar un evento.

El presente trabajo es resultado del requerimiento de una empresa operadora de centros de datos. Se solicitó realizar un sistema cuyo objetivo sea transmitir periódicamente a un servidor ubicado en la nube el valor de la presión en cuatro puntos de la tubería de gas inerte de un centro de datos. El valor de la presión, las coordenadas GPS del medidor de presión, la fecha y la hora deben ser almacenados por el servidor. El sistema debe contar con una interfaz de usuario, que se pueda acceder desde la Internet, mediante la cual se muestre en línea e históricamente los valores de presión. En caso de que el valor de la presión disminuya, por alguna razón, al mínimo permitido de 280 bares se deberá enviar un mensaje de alerta SMS y de WhatsApp al teléfono móvil del administrador del sistema. El punto de acceso WiFi a la Internet está localizado a 35 metros del medidor de presión más lejano.

En la actualidad, continúan realizándose numerosos trabajos e investigaciones para monitorización y control remoto de variables y procesos. Estos trabajos surgen de una necesidad específica, cumplen su objetivo y usan los últimos avances de la tecnología para monitorizar: temperatura [11], sensores marinos [12], sistemas de cuidado de la salud [13] y líneas de alimentación de gas [14] y electricidad [15], entre otras cosas. Estas aplicaciones reportan y almacenan información en una computadora ubicada localmente en el proceso bajo supervisión y no se consideran aplicaciones de IoT.

Con el crecimiento de la IoT en los últimos años, se han integrado capacidades remotas a las aplicaciones. Se han desarrollado aplicaciones de monitorización de: sistemas de generación de energía [16], signos vitales y cuidado de la salud [17], procesos industriales [18], manejo de basura y residuos peligrosos [19], campos agrícolas [20], nivel de agua en presas [21], control de tráfico vehicular [22] y hogares, edificios y ciudades inteligentes [23], entre otras. Estas aplicaciones de IoT reportan la información colectada a una computadora ubicada en la Internet cuya instalación, mantenimiento, programación y costo es parte de la implantación de la aplicación. La mayoría no hace uso de proveedores de servicios en la nube para disminuir costos y tiempo de desarrollo y obtener una aplicación eficiente.

Tomando en cuenta lo anterior, no se han realizado hasta ahora trabajos similares al aquí presentado. A través de una tarjeta HUZAH32-ESP32, se monitoriza periódicamente el valor de la presión de gas inerte en las tuberías de un centro de datos y los valores leídos se transmiten a la plataforma ThinkSpeak. Cuando el valor de la presión disminuya al mínimo permitido, el programa que se ejecuta en la HUZAH32-ESP32 transmite

el mensaje de alerta SMS y de WhatsApp usando los servicios de la plataforma Twilio.

Se utilizó la HUZAZH32-ESP32 por dos razones: 1) El SoC (System-on-Chip) ESP32 de la tarjeta integra un microcontrolador que ejecuta programas realizados en MicroPython, lo cual facilitó el desarrollo de la programación al no usar un lenguaje complejo de bajo nivel, existiendo además una gran cantidad de bibliotecas de funciones de libre uso y código abierto en la nube para realizar una variedad de tareas y 2) El SoC incluye un transceptor WiFi y puertos UART, lo cual permitió desarrollar un sistema compacto.

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.

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

Twilio es una plataforma de servicios basada en API (Application Programming Interfaces) 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 MycroPython, lo cual redujo el tiempo, complejidad y costo de desarrollo del sistema, 2) Se usó una tarjeta HUZAZH32-ESP32 de última tecnología y precio accesible que no requiere un sistema operativo y ofrece mayor cantidad de recursos que otras de su tipo como la PyBoard, 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 colectada por el sistema y 4) El administrador es notificado inmediatamente de la alerta de presión, permitiéndole reaccionar rápidamente ante una posible contingencia.

2. DESARROLLO

La metodología usada para el desarrollo del sistema consistió dividirlo en tres módulos: los nodos esclavos, el nodo maestro y la interfaz de usuario. A continuación, cada módulo se diseñó usando los componentes adecuados tomando en cuenta las

funciones del módulo y el costo de componentes. En la Figura 1 se muestra el diagrama de bloques del sistema diseñado.

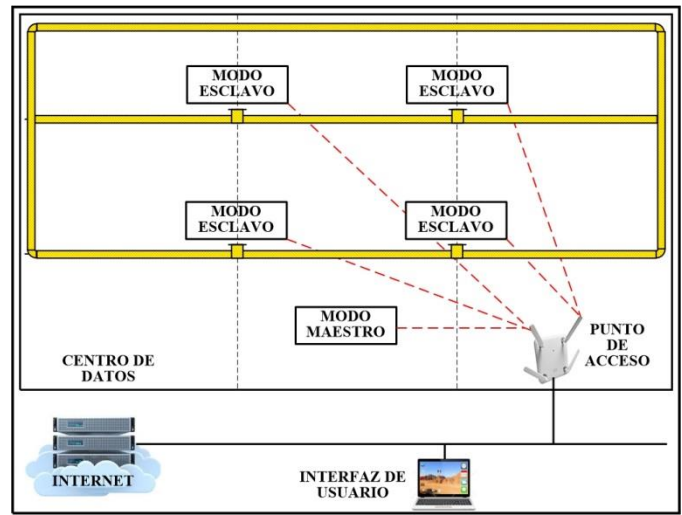


Fig. 1. Diagrama de bloques del sistema diseñado

2.1. Los nodos esclavos.

Se construyeron cuatro nodos esclavos todos con la arquitectura indicada en la Figura 2, la cual está compuesta por la tarjeta HUZAZH32-ESP32, el medidor de presión y el módulo GPS.

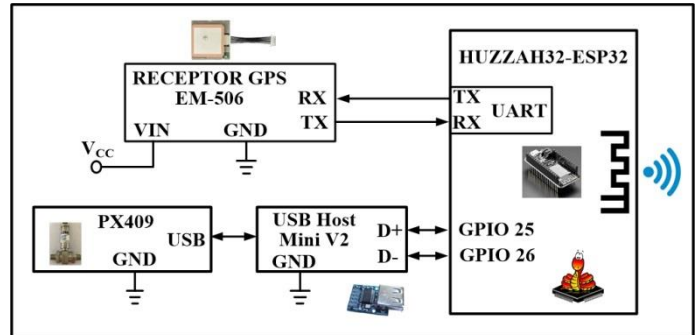


Fig. 2. Arquitectura de los nodos esclavos

La HUZAZH32-ESP32 es una tarjeta de desarrollo de ultra bajo consumo de energía usada para ejecutar programas realizados en MycroPython. Incorpora un SoC ESP32 el cual cuenta con los siguientes recursos hardware: un microcontrolador Tensilica LX6 de dos núcleos y 240 MHz, memoria SRAM de 520 KB, memoria flash de 4 MB, un transceptor HT40 WiFi 802.11b/g/n con antena integrada, un transceptor Bluetooth clásico y BLE, un conector USB, 3 puertos UART y terminales de entrada salida de propósito general o GPIO.

El medidor de presión usado es el dispositivo PX409. Este dispositivo de alta velocidad es un medidor de presión de gases construido en base a un sensor de silicio micro maquinado cuyo

rango de medida es de 25 mb a 345 bares. Realiza 1,000 lecturas de presión por segundo y proporciona el valor de la medida a través de una salida USB 2.0. La exactitud del PX409 es $\pm 0.08\%$. La razón por la que se eligió este dispositivo fue que es apto para medir la presión del gas inerte del centro de datos cuyo valor es 300 bares. Las salidas D+ y D- del puerto USB de este dispositivo se conectaron a las terminales GPIO 25 y 26 de la tarjeta HUZZAH32-ESP32, respectivamente, a través de un adaptador USB Host Mini V2. No se utilizó el puerto USB de la HUZZAH32-ESP32 para comunicarla con el medidor de presión ya que este puerto se usa para la comunicación con la computadora personal desde la cual se transfiere el programa que ejecuta la HUZZAH32-ESP32. La comunicación serie entre el medidor PX409 y la tarjeta HUZZAH32-ESP32 se realizó usando la biblioteca de funciones de código abierto *libusb-1.0-0-dev*.

Para obtener las coordenadas de ubicación de los nodos esclavos, se usó el módulo GPS NEO-6M. Este módulo cuenta con antena de cerámica integrada y un receptor GPS Ublox. Permite medir latitud, longitud, velocidad y altitud. Proporciona el valor de los parámetros medidos en forma serie por medio de un puerto UART, el cual puede trabajar a una velocidad de 4.8 a 115.2 Kbps. Se utilizó este dispositivo en los nodos esclavos, ya que sus características operacionales proporcionan una excelente opción de precisión y costo/beneficio. Se usó el puerto UART de este módulo para comunicarlo con el puerto UART de la tarjeta HUZZAH32-ESP32. La velocidad de transmisión de ambos puertos se configuró a 9.6 Kbps utilizando mensajes que siguen el protocolo NMEA. La frecuencia de actualización de datos de navegación del NEO-6M es 1 Hertz, esto significa que cada segundo proporciona al microcontrolador de la HUZZAH32-ESP32 las coordenadas GPS y la hora con una precisión de 3 metros. El protocolo NMEA es una especificación eléctrica y de intercambio de información, definida por la National Marine Electronics Association que se utiliza en la comunicación de dispositivos electrónicos de navegación marina. Cuenta con varios tipos de mensajes. El mensaje NMEA usado por el módulo NEO-6M para indicar la ubicación y la hora, se denomina GPGGA y significa Global Positioning System Fix Data. Este mensaje se compone de 18 campos, entre los cuales se encuentran la ubicación y la hora.

La programación de los nodos esclavos, realizada en MicroPython, ejecuta las siguientes tareas: 1) Configura las terminales GPIO 25 y 26 como entradas, configura los temporizadores e inicializa el puerto UART y el transceptor WiFi y 2) Entra a un ciclo en el que espera la solicitud del nodo maestro para leer el valor de presión por las terminales GPIO y las coordenadas GPS. A continuación, transmite al nodo maestro, a través del transceptor WiFi, el valor de presión, coordenadas GPS, la fecha y la hora. En la Figura 3 se muestra el diagrama de flujo usado para realizar la programación anterior.

2.2. El nodo maestro.

El nodo maestro está compuesto por la tarjeta HUZZAH32-ESP32. Las funciones de este nodo son las siguientes: 1) Solicita cada 5 minutos a los nodos esclavos realizar la lectura de presión del gas y coordenadas GPS, 2) Recibe la información transmitida por los nodos esclavos y la transmite al servidor en la nube e 3) Implanta la interfaz de usuario.

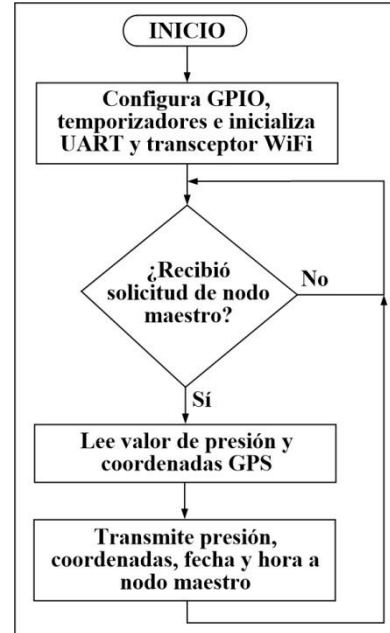


Fig. 3. Programación de los nodos esclavos

Estas funciones se realizan a través del programa que ejecuta el microcontrolador de la tarjeta HUZZAH32-ESP32 el cual lleva a cabo las siguientes tareas: A) Configura el temporizador de muestreo a 5 minutos, inicializa el transceptor WiFi y establece el valor mínimo de la presión del gas y el número de teléfono móvil del administrador del centro de datos y B) Entra a un ciclo en el que invoca la función que implanta la interfaz de usuario y espera la interrupción del temporizador.

La función que atiende la interrupción del temporizador de muestreo se encarga de solicitar y recibir la información de los nodos esclavos y transmitirla al servidor en la nube, usando la REST API de escritura para transmitir el mensaje a ThingSpeak. El mensaje contiene el identificador del nodo esclavo, el valor de presión, las coordenadas GPS del nodo, la fecha y la hora. Si el valor de la presión disminuye al mínimo permitido, se ejecuta la rutina que usa la REST API de Twilio para transmitir los mensajes de alerta SMS y de WhatsApp.

Las plataformas ThinkSpeak y Twilio tienen disponibles REST API para la creación de aplicaciones de IoT. Trabajan como interfaz para el intercambio de datos entre la plataforma y la programación del sistema utilizando el protocolo HTTP. Los datos pueden ser texto plano o estar 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 de la REST API a usar, la clave de la REST API y hasta 8 campos que pueden tener cualquier tipo de datos, como por ejemplo el valor de una variable leída desde un sensor. Los mensajes usados en este trabajo tienen 5 campos: el identificador del nodo esclavo, el valor de presión, la latitud, la longitud, la fecha y la hora.

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 proporcionadas por este proveedor para la transmisión de SMS y mensajes de WhatsApp.

2.3. La interfaz de usuario.

La interfaz de usuario está compuesta por el servidor web y la página del mismo. A través de la interfaz de usuario se visualiza en línea e históricamente la información colectada por el sistema. La implantación del servidor web se basó en la biblioteca de funciones de código abierto *uasyncio*. Esta biblioteca fue diseñada para realizar servidores web con microcontroladores usando la mínima cantidad de memoria RAM, llamados *picowebs*, así como para conexiones a bases de datos y colas de tareas distribuidas. En la Figura 4 se muestra la pantalla principal de la interfaz de usuario realizada.

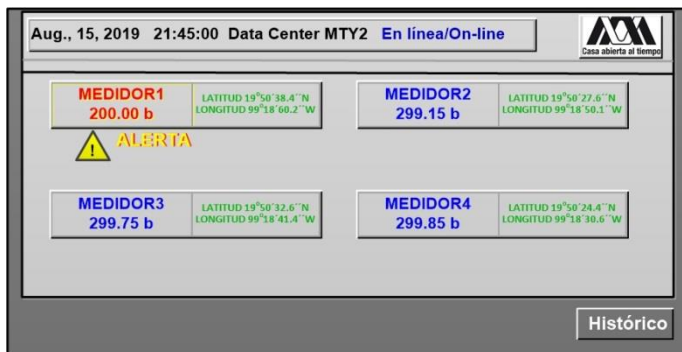


Fig. 4. Interfaz de usuario

3. RESULTADOS

Se realizaron tres grupos de pruebas. Aunque no se presentaron problemas en la comunicación entre los nodos de la red y el punto de acceso WiFi Cisco Aironet 1850 instalado a 35 metros del medidor de presión más lejano, se realizó el primer grupo de pruebas para determinar el alcance de la red. Para llevar a

cabo estas pruebas se ubicó un nodo esclavo en distintos puntos del centro de datos, inclusive más allá de los 35 metros. Los resultados mostraron que el alcance de la red fue 52 metros con línea de vista a una velocidad de 188 Mbps, menor a los 300 Mbps que pueden lograrse teóricamente usando el estándar 802.11n. A una distancia mayor a 52 metros la potencia de la señal WiFi recibida (RSSI-Received Signal Strength Indicator) en el nodo esclavo, disminuyó aceleradamente hasta perder el enlace cuando el nivel cayó a los -80 dBm como se muestra en la gráfica de la Figura 5. El nivel de RSSI se obtuvo usando la herramienta *inSSIDer* ejecutándose en una computadora portátil instalada en los puntos del nodo esclavo.

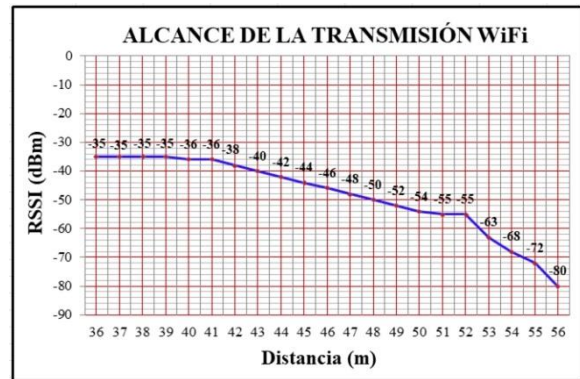


Fig. 5. Alcance de la transmisión WiFi

El segundo grupo de pruebas tuvo como objetivo determinar la exactitud de la presión medida por el sistema. Para llevar a cabo estas pruebas se instaló en la salida del tanque del gas inerte un medidor de presión del mismo tipo que el utilizado en la certificación de estos sistemas para tomar como referencia el valor de la presión reportada por este dispositivo. A continuación, se reguló la presión del gas para establecerla en 14 valores diferentes en el rango de 100 a 330 bares. En cada una de las 14 pruebas se obtuvo la media de presión de los cinco nodos esclavos mostrados en la interfaz de usuario y se determinó la exactitud usando la referencia indicada anteriormente. En cada prueba existió una diferencia pequeña entre la media de presión medida y la presión de referencia, de manera que cuando la presión de referencia fue 300 bar, la media de presión medida fue 299.915 bares, lo cual indica una exactitud de $\pm 0.085\%$. Los resultados de este grupo de pruebas mostraron también que al aumentar la presión del gas en el rango de 100 a 300 bares la exactitud aumenta ligeramente y para valores de presión mayores a 300 bares la exactitud se mantiene en $\pm 0.085\%$, como se muestra en la gráfica de la Figura 6.

El tercer grupo de pruebas tuvo como fin verificar el envío del SMS y mensaje de WhatsApp. En estas pruebas se estableció la presión del gas a 250 bares, menor al nivel mínimo permitido. Los nodos esclavos reportaron este valor al servidor,

mostrándose en la interfaz de usuario y transmitiendo el sistema los mensajes de alerta al teléfono móvil.

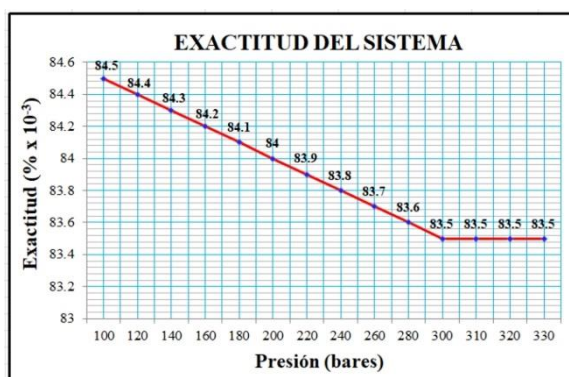


Fig. 6. Exactitud del sistema

4. CONCLUSIONES

Se obtuvo un sistema de monitorización de presión en tuberías de gas inerte en un centro de datos a través de una plataforma IoT, el cual reporta periódicamente a un servidor en la nube los valores de presión. Cuenta con una interfaz de usuario que muestra en línea e históricamente estos valores llevando a cabo una aplicación que cumple con los requisitos establecidos. Su instalación no es intrusiva, ya que usa comunicación inalámbrica WiFi cuyo alcance es 52 metros con línea de vista y no modifica el cableado del centro de datos. Se puede incrementar la cantidad de nodos esclavos, replicar el sistema en otro tipo de instalaciones o adicionar otros medidores o sensores realizando cambios mínimos en la programación de los nodos e interfaz de usuario. Para aumentar el alcance pueden utilizarse repetidores WiFi. En esta aplicación la monitorización de la presión y transmisión de mensajes a la plataforma ThinkSpeak se lleva cabo cada 5 minutos, 48 mensajes al día, lo cual no tiene costo de uso de la plataforma. Si el sistema se utiliza para monitorizar otra variable cuyo periodo de muestreo sea pequeño tal que necesiten enviarse más de 3 millones de mensajes anuales, deberá contemplarse el uso de una licencia de ThinkSpeak de mayor capacidad.

5. REFERENCIAS

[1] W. Wang, N. Lu, "Security Risk Analysis and Security Technology Research of Government Public Data Center", IEEE International Conference on Energy Internet (ICEDI), Beijing, China, 2018, pp. 185-189.
 [2] Y. Wang, R. Wang, Y. Guan, "Formal Modeling and Verification of the Safety Critical Fire-Fighting Control System", IEEE 39th Annual Computer Software and Applications Conference, Vol. 3, Taichung, Taiwan, 2015, pp. 536-541.
 [3] F. Bao, Y. Hu, "The high pressure water mist fire extinguishing system based on Anylogic modeling", Chinese Control and Decision Conference (CCDC), Yinchuan, China, 2016, pp. 6311-6315.
 [4] Z. Leiji, T. Huijiao, L. Liangliang, "A state monitoring method of fire water supply system based on hydraulic pressure mean value", 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, China, 2017, pp. 74-79.

[5] International Organization Standardization, ISO 14520-1:2015 International Standard, Switzerland, 2015.
 [6] P. A. L., V. A. Karaseva, E. P. Kunakov, "Implementation of information security and data processing center protection standards", IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT&MQ&IS), Nalchik, Russia, 2016, pp. 138-143.
 [7] H. Hejazi, H. Rajab, T. Cinkler, "Survey of platforms for massive IoT", IEEE International Conference on Future IoT Technologies (Future IoT), Eger, Hungary, 2018, pp. 1-8.
 [8] K. Routh, T. Tannistha Pal, "A survey on technological, business and societal aspects of Internet of Things by Q3, 2017", 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-4.
 [9] R. Gowtham, J. A. Daniel, S. Karthik, "Revamping Digital Land Survey using GPS and Internet of Things (IoT)", International Conference on Soft-computing and Network Security (ICSNS), Coimbatore, India, 2018, pp. 1-9.
 [10] B. Dhanalaxmi, G. A. Naidu, "A survey on design and analysis of robust IoT architecture", International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2017, pp. 375-378.
 [11] K. J. Lee, J. Y. Kim, B. R. Kim, "Real-Time 2D Microwave Differential Imaging for Temperature Monitoring", International Symposium on Antennas and Propagation (ISAP), Busan, Korea (South), 2018, pp. 1-2.
 [12] L. Dongyan, Z. Yingying, Z. Bo, "Development and calibration of on-line monitoring sensor for Marine radioactivity based on NaI(Tl) detector", Chinese Automation Congress (CAC) Xi'an, China, 2018, pp. 1973-1976.
 [13] K. Kalyan, V. K. Chugh, C. S. Anoop, "Non-invasive heart rate monitoring system using giant magneto resistance sensor", 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, USA, 2016, pp. 4873-4876.
 [14] M. Saad, E. Nyenhuis, "On-line gas monitoring for increased transformer protection", IEEE Electrical Power and Energy Conference (EPEC), Saskatoon, Canada, 2017, pp. 1-4.
 [15] L. Renforth, R. Giussani, M. T. Mendiola, "On-line partial discharge insulation condition monitoring of complete high voltage (HV) networks", Petroleum and Chemical Industry Technical Conference (PCIC), Calgary, Canada, 2017, pp. 103-112.
 [16] C. S. Choi, J. D. Jeong, I. W. Lee, "LoRa based renewable energy monitoring system with open IoT platform", International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, USA, 2018, pp. 1-2.
 [17] P. Jangra, M. Gupta, "A Design of Real-Time Multilayered Smart Healthcare Monitoring Framework Using IoT", International Conference on Intelligent and Advanced System (ICIAS), Kuala Lumpur, Malaysia, 2018, pp. 1-5.
 [18] K. S. Shinde, P. H. Bhagat, "Industrial process monitoring using IoT", International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2017, pp. 38-42.
 [19] A. Khan, A. Khachane, "Survey on IOT in Waste Management System", 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2018, pp. 27-29.
 [20] S. Heble, A. Kumar, K. V. Durga, "A low power IoT network for smart agriculture", IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, Singapore, 2018, pp. 609-614.
 [21] S. S. Siddula, P. Babu, P. C. Jain, "Water Level Monitoring and Management of Dams using IoT", 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-5.
 [22] J. Lee, G. Yoon, H. Choi, "Monitoring of IoT Data for Reducing Network Traffic", Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, 2018, pp. 395-397.
 [23] J. L. Díaz, A. E. Guerrero, M. Toledano, "IoT Based Ambient Monitoring System for Intelligent Buildings", IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA), Concepcion, Chile, 2018, pp. 1-6.