

Consumo energético en placas de desarrollo IoT al transmitir variables Float por MQTT

Gabriel Lee Alvarez Rosado^a, Veronica Quintero Rosas^b, Mario Alberto Camarillo Ramos^c, Roberto Lopez Avitia^d, Atalo Jr. Machado Carrillo^e

^a Tecnológico Nacional de México / Instituto Tecnológico de Mexicali, gabrielalvarez@itmexicali.edu.mx, Mexicali, Baja California, México.

^b Tecnológico Nacional de México / Instituto Tecnológico de Mexicali, Veronicaquintero@itmexicali.edu.mx, Mexicali, Baja California, México.

^c Tecnológico Nacional de México / Instituto Tecnológico de Mexicali, mario.camarillo@itmexicali.edu.mx, Mexicali, Baja California, México.

^d Departamento de Bioingeniería, Universidad Autónoma de Baja California, roberto.lopez.avitia@uabc.edu.mx, Mexicali, Baja California, México.

^e Tecnológico Nacional de México / Instituto Tecnológico de Mexicali, a23490422@itmexicali.edu.mx, Mexicali, Baja California, México.

Resumen

El internet de las cosas (IoT) es una tecnología que ha impactado en gran medida debido a su amplio espacio de aplicación. Estas tecnologías vienen ligadas estrechamente a la portabilidad, lo que conlleva varias dificultades, como el consumo energético y el medio de comunicación con otros dispositivos.

El consumo energético de los dispositivos IoT es algo fundamental, por lo que se realizaron pruebas al transmitir una variable tipo float por medio de WIFI con el protocolo MQTT, varios dispositivos IoT están enfocados en la captura de datos o monitoreo por lo que la precisión es algo a considerar al capturar y transmitir, por lo que en esta investigación se modificó la cantidad de dígitos de precisión para poder observar si al transmitir más dígitos de precisión se obtiene un consumo energético mayor. Se realizó la prueba en tres placas de desarrollo distintas las cuales son ESP32, ESP8266 y Arduino uno wifi rev 2. El resultado de la investigación muestra que las placas ESP32 consumen 241% más energía al transmitir ningún dígito de precisión ("1") que al transmitir 7 dígitos de precisión ("1.1234567"), esto se obtuvo dividiendo el dígito de mayor consumo (0 dígito con 27.00 mJ) entre el de menor consumo (7 dígitos con 7.91mJ), resultando en 3.41 veces más de energía consumida o un 241% más de energía consumida. La placa ESP8266 consume un 540% más de energía al transmitir 2 dígitos de precisión ("1.12") comparado con 7 dígitos de precisión ("1.1234567"). La placa de Arduino consume 2565% más energía al transmitir 6 dígitos de precisión ("1.123456"), que al transmitir 1 dígito de precisión ("1.1").

Palabras clave—IoT, Consumo, Energía, Microcontroladores, MQTT.

Abstract

The Internet of Things (IoT) is a technology that has greatly impacted due to its wide application space. These technologies are closely linked to portability, which carry several difficulties, energy consumption and the means of communication with other devices.

The energy consumption of IoT devices is fundamental, so tests were carried out by transmitting a float type variable through WIFI with the MQTT protocol, several IoT devices are focused on data

collection or monitoring so the precision is something to consider when we are capturing and transmitting, so in this research the number of precision digits is modified to be able to observe if transmitting more precision digits results in greater energy consumption. The test is carried out on three different development boards which are ESP32, ESP8266 and Arduino uno wifi rev 2. The result of the research shows that the ESP32 board consumes 241% more energy, compared to 0 precision digits ("1") to 7 precision digits ("1.1234567"). This was obtained by dividing the digit with the highest consumption (0 digit with 27.00mJ) by the digit with the lowest consumption (7 digits with 7.91mJ), resulting in 3.41 time more energy consumed or 241% more energy consumed. The ESP8266 board consumes 540% more energy transmitting 2 precision digits ("1.12") compared to 7 precision digits ("1.1234567"). And, the arduino board consumes 2565% more energy, when transmitting 6 precision digits ("1.123456"), compared to 1 precision digits ("1.1").

Keywords— IoT, Consumption, Energy, Microcontroller, MQTT.

1. INTRODUCCIÓN

El internet de las cosas (IoT) es una tecnología que ha llegado para quedarse, esta tecnología ha crecido de una gran manera, por lo que ha sido foco en la investigación de estos.

La mayoría de los dispositivos IoT están enfocados en comunicarse entre otros dispositivos (Machine-To-Machine M2M), para esto utilizan diferentes protocolos como: Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Hypertext Transfer Protocol (HTTP), entre otros más protocolos los cuales permiten entenderse entre sí.

El IoT se relaciona al desarrollo de múltiples tareas en distintos aspectos [1][2], desde crear, editar o compartir información. Algunas fuentes estiman que para 2025 existirán 9 dispositivos IoT por persona en el mundo llegando a una estimación de 75 billones [3]. Al ser tantos dispositivos conectados, el consumo energético de dispositivos IoT toma mayor relevancia en aplicaciones donde sean aplicaciones inalámbricas y una cantidad de energía limitada esté en juego por ejemplo, las baterías.

Estos dispositivos al contar con energía limitada, la cantidad de tiempo que puede estar en operación está relacionada fuertemente con la cantidad de veces que puede capturar, procesar y transmitir información, por lo que llega a ser fundamental observar cuánto consume al realizar una o varias de estas acciones.

Los protocolos de comunicación consumen energía por la cantidad de datos transmitidos [2][3][4], cada protocolo de comunicación tiene una estructura y cantidad de datos distinta, por lo que estos tienden a consumir más o menos por la cantidad de información a transmitir.

Anteriormente, en un trabajo previo [6], se desarrolló una metodología para generar un perfil energético, transmitiendo solamente un carácter "A", en donde no se colocó ninguna estructura y se evaluó con tres suministros distintos de voltaje para observar el consumo, a partir de las múltiples pruebas la placa esp8266 es la que tiene un menor consumo para realizar aplicaciones IoT utilizando Mqtt al transmitir un byte. Continuando con la investigación anterior, se realizó una

prueba en donde una variable Float es empaquetada en un archivo Json, y este es transmitido por MQTT a un servidor Mosquitto, el cual opera de manera local.

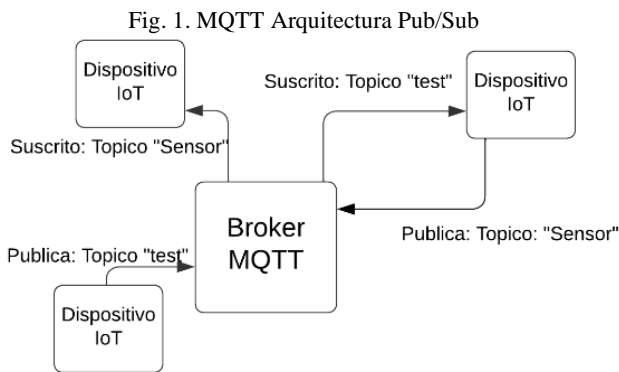
A la variable Float se le incrementó la cantidad de dígitos decimales que puede almacenar, hasta llegar a su límite, y observamos el comportamiento y consumo al transmitir tal dato, estas pruebas se realizaron en tres placas distintas, Arduino uno wifi rev 2, Esp32 y Esp8266.

2. CONTENIDO

2.1 Cómo funciona el MQTT

MQTT es un protocolo ligero de rápida comunicación diseñado para dispositivos IoT. Este permite la comunicación entre diferentes dispositivos usando el modelo Publicación/Suscripción [6][7], para esto es necesario un “bróker” el cual funciona como un servidor.

Un dispositivo IoT puede solicitar información y mandarla directamente al bróker. Cada publicación realizada cuenta con un “tópico”, y para leer esta información es necesario suscribirse al tópico correspondiente. Se observa en la Fig. 1 la arquitectura general de comunicación de los dispositivos IoT con el bróker.



Fuente: propia

MQTT tiene tres niveles de calidad (QoS):

- QoS0 (Como máximo una vez):
 - Mensaje no confirmado por el destinatario si se recibe.
- QoS1 (Al menos una vez):
 - El mensaje transmitido puede llegar una o más veces hasta que el destinatario devuelva una respuesta de confirmación.
- QoS2 (Exactamente una vez):
 - Cada mensaje transmitido se recibe solo una vez y es conformado por el receptor.

Para usar estos protocolos es necesario crear un broker ya sea de manera local o remota. Para esto existen diferentes softwares que permiten la creación de un broker de manera local como Mosquitto [7], Mosca [8] y ActiveMQ [9], para esta investigación se utilizó el software Mosquitto.

2.2 Métodos y materiales

Se eligió el protocolo MQTT ya que es uno de los más utilizados en 2023, siendo este utilizado por el 49% de los desarrolladores de aplicaciones IoT[10].

Se utilizaron 3 placas de desarrollo distintas: ESP32, ESP8266 y Arduino uno Wifi nano rev 2. Estas 3 placas tienen en común su comunicación Wifi, con la cual es posible desarrollar aplicaciones IoT.

Estas tres placas fueron alimentadas a un voltaje de 5v de operación y utilizaron un algoritmo en el que se incrementa la cantidad de dígitos a imprimir en una variable float.

Las variables float utilizadas en la IDE Arduino, se conforman de 6 a 7 dígitos de precisión [11], .

Para las pruebas de consumo energético, se realizaron mediciones de la energía al transmitir N cantidad de decimales, iniciando en 0 hasta terminar en 9 decimales, esto se muestra en la Fig. 2.

Fig. 2. Cantidad de decimales enviados

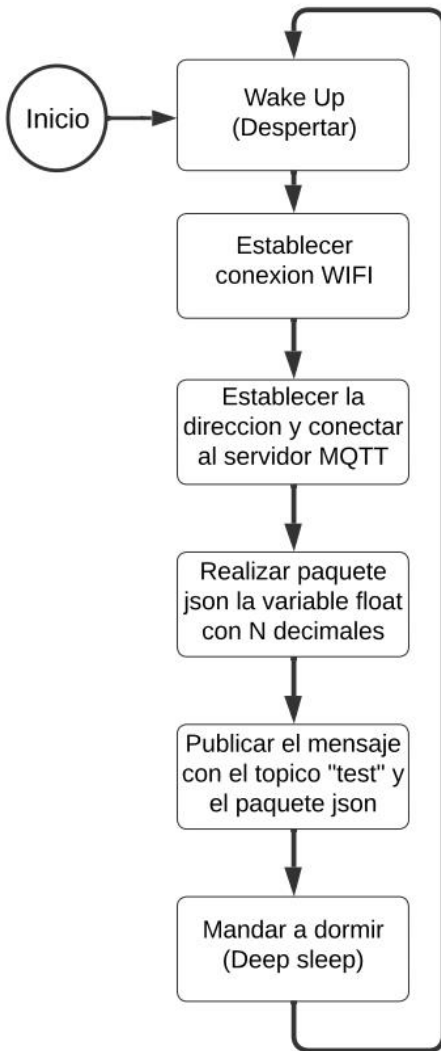


Fuente: Propia

La Fig. 3 muestra el algoritmo utilizado. Se inicia la operación inicial en la placa de desarrollo, la cual consta de configuración de puertos y osciladores. En segundo término, se realiza la conexión a la red WIFI. Al concretar la comunicación, se establece una suscripción con el servidor MQTT. Seguido se empaqueta una variable float en formato JSON. Por último, la información se transmite a través de MQTT al servidor. Al término del proceso, se procura el modo “Deep sleep”, para promover el consumo mínimo de

energía. Lo anterior se repite en cada iteración, este algoritmo puede ser utilizado en cualquier placa de desarrollo que se desee realizar pruebas, en nuestro caso se aplicó a las tres placas de desarrollo utilizando la ide de Arduino.

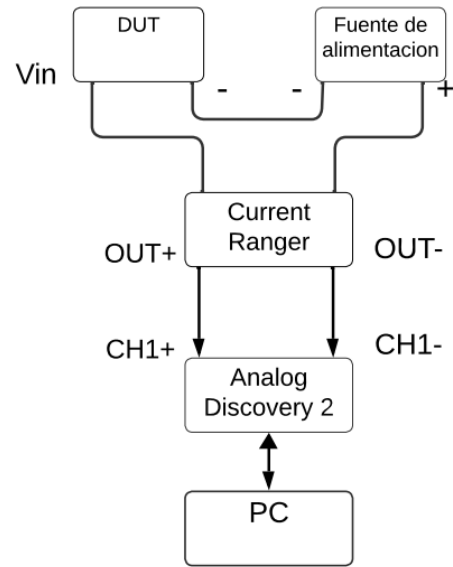
Fig. 3. Algoritmo para medir el consumo energético al transmitir información por MQTT



Fuente: Propia

El consumo energético se obtuvo capturando la corriente con un osciloscopio usb (Analog Discovery 2) y un Current Ranger, el cual es un equipo que convierte corriente a voltaje utilizando una resistencia shunt de lado bajo, todo esto acompañado de una fuente de alimentación variable y el dispositivo bajo prueba (DUT). Esta configuración se presenta en la Fig. 4.

Fig. 4. Circuito de captura de datos



Fuente: Propia

2.3 Resultados y discusiones

Al realizar las pruebas de decimales por cada placa, se obtuvieron los siguientes resultados en corriente, como se observa en la Fig. 5.

En la Fig. 5, observamos la corriente promedio consumida por las tres placas de desarrollo en cada dígito. Al analizar la corriente promedio de cada placa, las placas ESP32 y Arduino son las que más consumen corriente promedio con 137 mA y 132 mA respectivamente, y la placa ESP8266 con un promedio de 87mA al transmitir una variable float por MQTT.

Observando de manera más detallada la Fig. 6 y Fig. 7, se consume menos corriente al transmitir 7 dígitos de precisión “1.1234567”, y se consume más corriente al transmitir 4 dígitos “1.1234, pero al realizar el análisis de energía, se observó que si se transmiten variables con 1 dígitos de precisión se realiza un consumo muy pequeño de energía.

Fig. 5. Corriente consumida por las placas de desarrollo, (Rojo = ESP32, Verde = ESP8266, Azul = Arduino uno wifi rev 2).

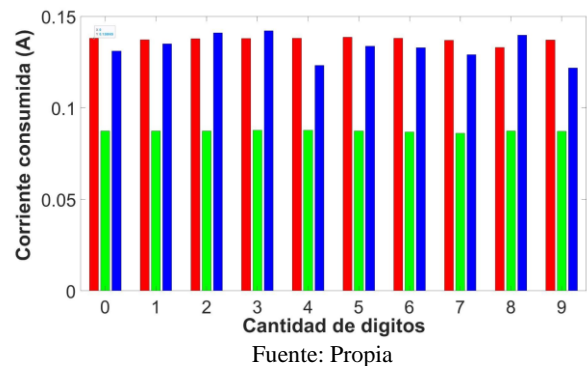
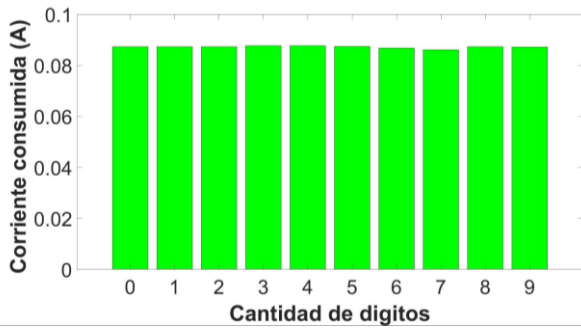
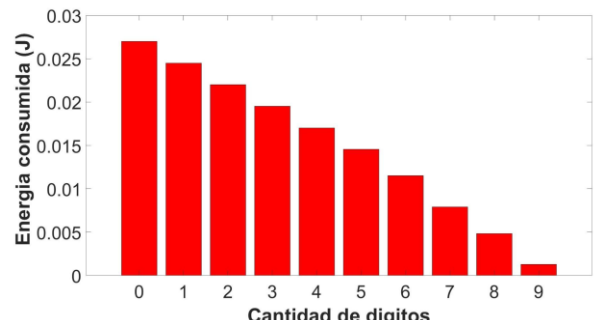


Fig. 6. Corriente promedio consumida - ESP8266

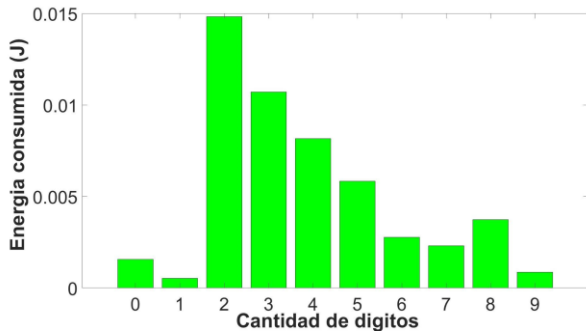


Fuente: Propia



Fuente: Propia

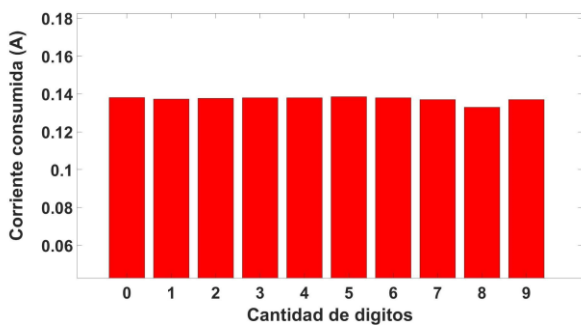
Fig. 7. Energía promedio consumida - ESP8266



Fuente: Propia

Las mediciones de energía en las placas ESP32 mostradas en las Fig. 8 y Fig. 9 sugiere al transmitir el noveno dígito un consumo menor, pero este supera los dígitos de precisión, por lo que se comparó el séptimo dígito, con la variable con ningún dígito de precisión, concluyendo que al transmitir ningún dígito de precisión se obtiene un 341% más consumo, que al enviar todos los dígitos de precisión.

Fig. 8. Corriente promedio consumida - ESP32

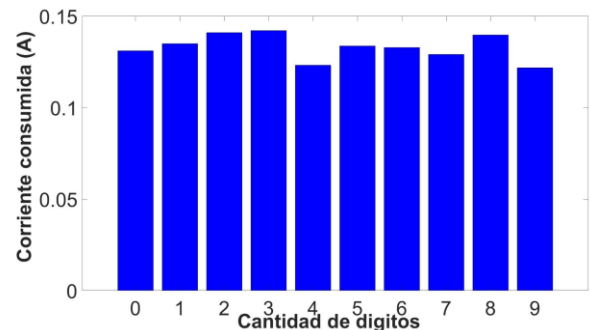


Fuente: Propia

Fig. 9. Energía promedio consumida - ESP32

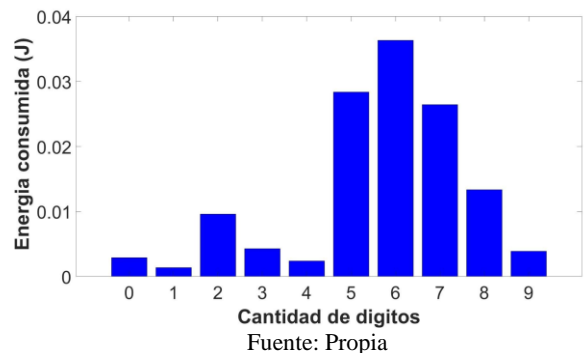
La placa de Arduino tuvo un consumo más irregular en corriente, en donde no se puede encontrar una tendencia por la cantidad de dígitos de precisión transmitidos, como se puede ver en la Fig. 10, al transmitir 1 solo dígito de precisión, aunque este tenga un consumo de 134 mA, tuvo el menor consumo de energía con un 1.36 mJ, se observa en la Fig. 11, por lo que aunque la corriente sea mayor, el tiempo en el que se transmitió fue menor, obteniendo menor energía.

Fig. 10. Corriente promedio consumida - Arduino uno WIFI rev 2



Fuente: Propia

Fig. 11. Energía promedio consumida - Arduino uno WIFI rev 2



Fuente: Propia

3. CONCLUSIONES Y RECOMENDACIONES

La cantidad de dígitos de precisión puede afectar el consumo energético. En las placas ESP32 y ESP8266 a mayor cantidad de dígitos de precisión se obtiene un menor consumo de energía. Como se muestra en la fig 8 la energía consumida por el ESP8266 en 1 si comparamos la transmisión con 7 dígitos

de precisión "1.1234567" y se compara con la transmisión de 2 dígitos "1.12", este consume 5.4 veces más energía aunque la cantidad de dígitos sea menor. Realizando el mismo análisis con la placa ESP32, esta consume 2.41 veces más al transmitir "1" que al transmitir "1.1234567". Estos resultados recomiendan transmitir utilizando la máxima cantidad de dígitos permitidos por la variable Float. La placa de desarrollo de Arduino cuenta con un consumo más irregular, por lo que se comparó la precisión de 6 dígitos "1.123456" con 1 dígito de precisión "1.1", la placa consumió 25.65 veces más al transmitir más dígitos de precisión, por lo que se recomienda transmitir una menor cantidad de dígitos de precisión en la placa Arduino uno Wifi rev 2.

Para trabajo futuro se recomienda realizar el análisis al transmitir una mayor cantidad de paquetes, bytes, en donde sea evaluado byte por byte la cantidad de energía.

4. REFERENCIAS

- [1] P. Pierleoni *et al.*, "IoT Solution based on MQTT Protocol for Real-Time Building Monitoring," *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*, Ancona, Italy, 2019, pp. 57-62, doi: 10.1109/ISCT.2019.8901024.
- [2] J. Toldinas, B. Lozinskis, E. Baranauskas and A. Dobrovolskis, "MQTT Quality of Service versus Energy Consumption," *2019 23rd International Conference Electronics*, Palanga, Lithuania, 2019, pp. 1-4, doi: 10.1109/ELECTRONICS.2019.8765692.
- [3] B. Safaei, A. M. H. Monazzah, M. B. Bafroei and A. Ejlali, "Reliability side-effects in Internet of Things application layer protocols," *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, Milan, Italy, 2017, pp. 207-212, doi: 10.1109/ICSRS.2017.8272822.
- [4] A. Viswanathan, "Analysis of power consumption of the MQTT protocol," Ph.D. dissertation, University of Pittsburgh, 2017.
- [5] Alvarez-Rosado, G., et al. "Energy consumption of an internet of things development board," in *Revista de ciencias tecnologicas*, vol. 5, no. 4, 2022.
- [6] S. Kraijak and P. Tuwanut, "A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends," *11th International Conference on Wireless Communications, Networking and [7] Mobile Computing (WiCOM 2015)*, 2015, pp. 1-6. <https://doi.org/10.1049/cp.2015.0714>.
- [22] MQTT Version 5.0. Edited by Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta. 07 March 2019. OASIS Standard. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>. Latest version: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.
- [7] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," *The Journal of Open-Source Software*, vol. 2, no. 13, May 2017, <https://doi.org/10.21105/joss.00265>.
- [8] M. Collina. (2013). Mosca. Available: <https://github.com/moscajs/mosca>. [25] APACHE. [9] ACTIVEMQ. Available: <https://activemq.apache.org>. [10] E. foundation, "IoT & Edge developer survey report December 2023," Eclipse foundation 2023. [11] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/reference/es/language/variables/data-types/float/>.