JIDeTEV
Jornadas de Investigación y Desarrollo Tecnológico
Extensión, Vinculación y Muestra de la Producción

JIDeTEV- Año 2021 -ISSN 2591-4219

Preparación de Artículos para JIDeTEV – Data logger a batería e interfaz web para visualización de nivel de agua en un tanque domiciliario

Juan Pablo Pavlik Salles[a], Renzo Paolo Grilletti[b], Ricardo A. Korpys[c]

[a] *Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.*
[b] *Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.*
[c] *Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.*
e-mails: juanpipavlik@gmail.com, griren91@gmail.com, korpys@fio.unam.edu.ar

Resumen

Este proyecto se encuentra aún en desarrollo en la materia "Proyecto y Diseño Electrónico" de la carrera de Ingeniería Electrónica como proyecto final de la misma. Consiste en un sistema digital de medición y registro de nivel de agua en tanque inalámbrico y a batería basado en el protocolo MQTT. El sistema se encuentra compuesto por dos partes, el dispositivo sensor cliente MQTT conformado por un SoC ESP32 y un sensor ultrasónico que se encuentran ubicados en inmediaciones del tanque, y una Raspberry Pi trabajando como servidor encargada de registrar y permitir al usuario visualizar de manera gráfica el contenido del tanque a través de Grafana con cualquier dispositivo con navegador web.

Palabras Clave *– IOT. ESP32. Raspberry. MQTT. WiFi. Batería. Bases de datos. Grafana. Datalogger. Autonomía*

1. Introducción

En casos de corte en el suministro de agua o de sistemas aislados de la red de agua se vuelve importante tener el control de la cantidad disponible de agua en los depósitos o tanques; determinar la cantidad de agua con la que se dispone implica acceder a estos tanques que en muchas ocasiones no se encuentran fácilmente asequibles, estos se suelen encontrar ubicados en techos, torres o enterrados, en todos los casos, el acceso a estos es complicado y/o peligroso.

Este proyecto consiste en llevar a cabo un sistema inalámbrico a batería con una duración de batería de aproximadamente 3 meses a través del cual el usuario pueda, mediante una interfaz web, visualizar el contenido del tanque en todo momento de manera gráfica e intuitiva.

2. Proyecto y diseño

2.1.Esquema general de funcionamiento

El sistema basa su funcionamiento en una arquitectura de comunicación del tipo cliente subscriptor basada en el protocolo MQTT utilizando un broker de mensajería open source desarrollado por Eclipse llamado "mosquitto". Este se encuentra instalado en una Raspberry Pi 3b+ ubicada en la red local del domicilio, su función es recibir los mensajes enviados por el sistema de medición ubicado en las inmediaciones del tanque para luego almacenarlos en una base de datos SQL mariaDB de la cual Grafana extrae la información para ser visualizada en una interfaz web a través de cualquier navegador web.

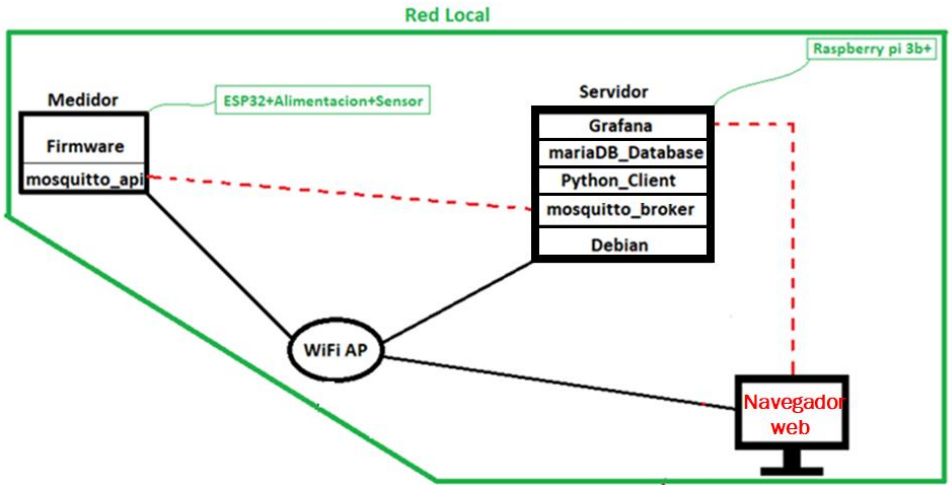

Figura ***1: Esquema general de funcionamiento del sistema de comunicación***

Todo el software utilizado en el desarrollo del proyecto es open source o elaborado por nosotros, por lo tanto es gratuito y no tiene restricciones de uso como otras posibles soluciones empleando plataformas de IOT como Thinger.io o ThingSpeak.

MQTT es un protocolo estandarizado y por lo tanto, la arquitectura presentada en la *Figura* 1 permite escalar el sistema de medición simplemente agregando clientes MQTT a este. De esta manera, lo que empezó siendo un simple sistema de medición de agua podría convertirse en un complejo sistema de registro y medición de variables o incluso el principio del desarrollo de una aplicación de domótica.

2.2. Medición de agua

La medición de volumen de agua en el tanque se obtiene mediante la adquisición de la distancia del nivel de agua respecto a la tapa del tanque utilizando un sensor ultrasónico JSN-SR04T-V3 similar a los utilizados en los vehículos en los sensores de estacionamiento.

Figura 2: Sensor JSN-SR04T-V3

Con la distancia ‘D’ y con un modelo matemático de tanque cilíndrico se puede obtener el contenido de agua en el tanque con la ecuación

$$V(D) = \pi R^2 (H - D) \quad (1)$$

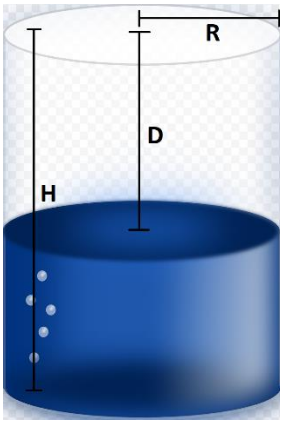

Figura 3: Magnitudes del tanque de agua

La distancia es obtenida mediante la transmisión de ondas ultrasónicas y la medición del tiempo en que estas tardan en rebotar, tanto la transmisión como el sensado de las ondas se llevan a cabo con el mismo transductor (*Figura 2*), es decir, este opera como parlante y micrófono.

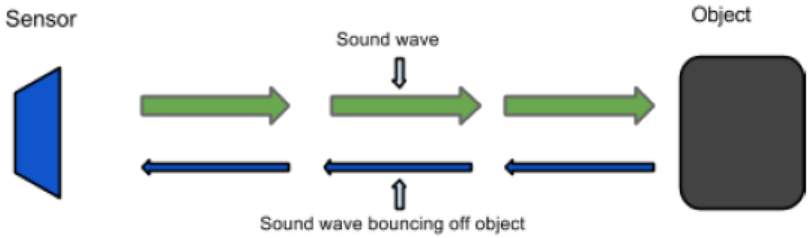

Figura 4: Medición de distancia con sensor ultrasónico

Las ondas de sonido viajan en el aire y la velocidad de esta es relativamente constante e igual a $\mathbf{0,0343}\frac{\boldsymbol{cm}}{\mu \boldsymbol{s}}$, por lo tanto, la distancia recorrida por la onda se obtiene con la Ecuación N° 2 donde $\boldsymbol{\Delta t}$ es el tiempo que tarda la onda en recorrer la distancia 'D' ida y vuelta.

$$D = 0{,}0343\frac{\Delta t}{2} \quad (2)$$

2.3. Esquema y funcionamiento del sistema de medición

El sistema de medición se basa en el SoC ESP32, el diagrama de bloques del funcionamiento de este se presenta en la *Figura 5*.

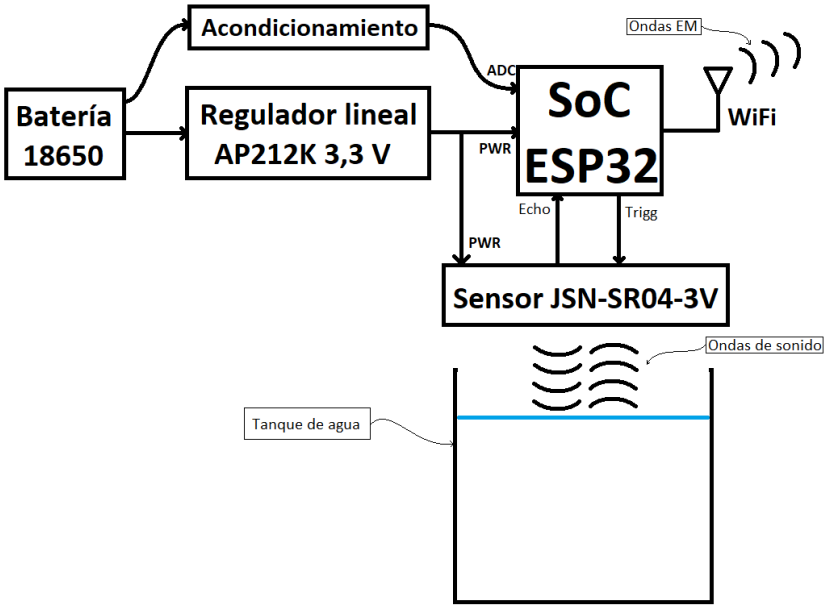

Figura 5: Diagrama de bloques del sistema de medición basado en el SoC ESP32

El SoC fue programado en el IDE de Arduino para que lleve a cabo las actividades expresadas en el siguiente diagrama de bloques.

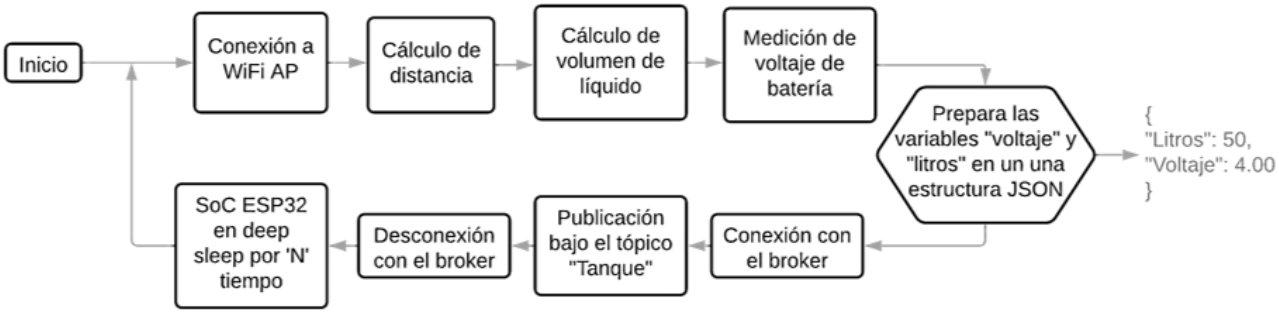

Figura 6: Diagrama de bloques del firmware que corre en el SoC

El ESP32 se inicia, lleva a cabo la conexión con el Access Point WiFi del domicilio, calcula la distancia del nivel de agua, con esta variable se calcula el volumen de agua del tanque, luego de esto se lleva a cabo la medición del voltaje de la batería y se preparan ambas variables en una estructura JSON para ser publicadas al broker bajo el tópico “tanque”, una vez hecho todo esto el ESP32 se desconecta de todas las conexiones realizadas y duerme por un período de tiempo ‘N’ definido por firmware.

2.4. Servidor

El servidor consiste en una raspberry Pi 3b+ corriendo Debian, en esta se instalaron las siguientes aplicaciones: un broker MQTT llamado “mosquitto”, un Sistema de Gestión de Bases de Datos SQL llamado “mariaDB” y una aplicación web llamada “Grafana”.

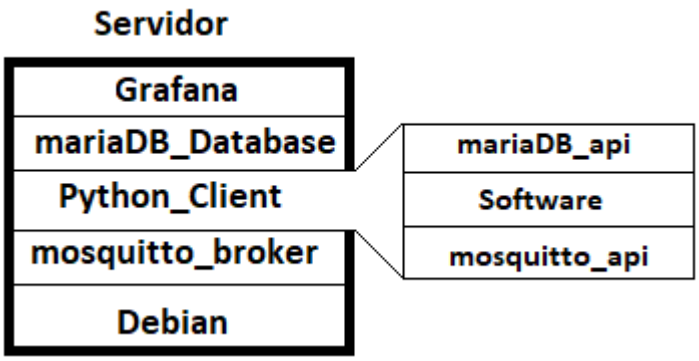

Figura 7: Aplicaciones pertinentes al proyecto que se encuentran corriendo en el servidor

El cliente Python que se encuentra en la *Figura 7* fue desarrollado por nosotros, esta aplicación se comporta como un cliente MQTT que se encuentra suscripto al tópico "tanque" bajo el que publica el sensor, de esta manera, la estructura JSON publicada por el sensor es recibida por esta aplicación, esta se encarga de decodificar la estructura JSON y genera la instrucción SQL para almacenar en la base de datos las variables de voltaje y cantidad de agua en una tabla previamente definida (*Figura 8*).

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
id	int(11)			No	*Ninguna*		AUTO_INCREMENT
TimeStamp	datetime			No	current_timestamp()		
Liquido	float(5,2)		UNSIGNED	No	*Ninguna*		
VoltajeBateria	float(3,2)		UNSIGNED	No	*Ninguna*		

Figura 8: Formato de tabla de datos

Todas las aplicaciones que se encuentran en la *Figura 7* se inician automáticamente cuando se energiza el servidor.

2.5. Estimación de duración de batería

En lo que respecta la duración de la batería, esto está definido por el consumo del sistema, que consta del SoC ESP32 y otros elementos como transistores y el sensor. El funcionamiento del sistema de medición consta de dos estados, uno activo y otro dormido en los que se tienen diferentes consumos de corriente de la batería.

En la *Figura 9* se ilustra la dinámica de la corriente consumida, alternando en dos diferentes niveles de corriente, I_{on} de alto consumo que dura un tiempo t_{on} donde el sistema realiza las mediciones, cálculos y transmisión de la información de forma inalámbrica, e I_{off} consumida en estado de reposo o hibernación durante un tiempo t_{off}.

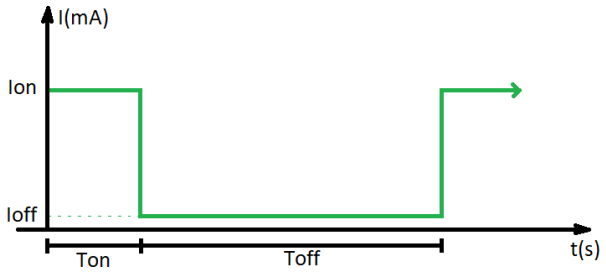

Figura 9: Consumo de corriente en los dos estados de funcionamiento del Sistema de medición

Dado que la dinámica de la corriente consumida por el sistema de medición es una señal periódica es posible obtener realizar un cálculo de una corriente media.

$$I_{MEDIA} = \frac{I_{on}T_{on} + I_{off}T_{off}}{T_{on} + T_{off}} \quad (3)$$

De esta manera, y teniendo en cuenta el parámetro de capacidad de la batería en mAh se puede determinar el tiempo de duración de la batería.

$$t_{\text{duracion de bateria}} = \frac{C_{\text{batería}}}{I_{MEDIA}} \quad (4)$$

Reemplazando la ecuación 3 en la 4 y asumiendo como constantes a T_{on}, $C_{batería}$, I_{on}, I_{off} se puede definir que la duración de la batería es función del tiempo de dormido del sistema de medición resultando una gráfica como la que se muestra en la *Figura 10*.

$$t_{\text{duracion de bateria}} = C_{\text{batería}} \frac{T_{on} + T_{off}}{I_{on}T_{on} + I_{off}T_{off}} \quad (5)$$

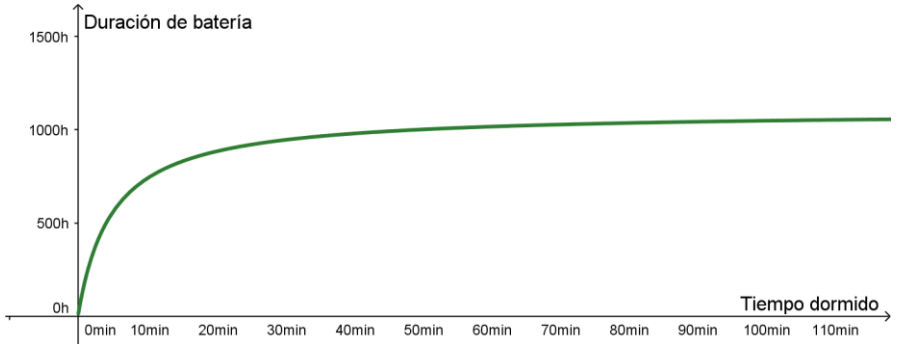

Figura 10: Gráfica de duración de batería en función de tiempo de hibernación

Teniendo en cuenta que la corriente promedio del sistema en estado activo es de 150 mA, que la corriente en reposo se espera que sea de alrededor de 84,53 µA, que la capacidad de las baterías comerciales rondan por los 2200 mAh y que el tiempo activo del sistema de medición es de alrededor de 4 segundos, entonces se define que para que el sistema dure aproximadamente 3 meses energizado se requiere de un tiempo de dormido mínimo de 10 minutos.

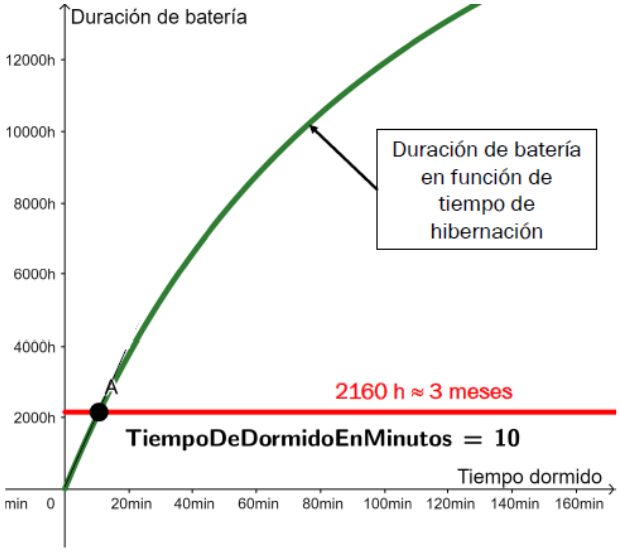

Figura 11: Cálculo de tiempo de reposo para cumplir con 3 meses de duración de batería del sistema de medición

3. Pruebas de funcionamiento

Para llevar a cabo la prueba del funcionamiento del sistema de medición en conjunto con el servidor se hizo uso de un módulo de desarrollo NODEMCU que está basado en el SoC ESP32 utilizado en este proyecto y se cargó al módulo el firmware correspondiente al diagrama de bloques de la *Figura* 6 con un tiempo de dormido 'N' de 1 minuto.

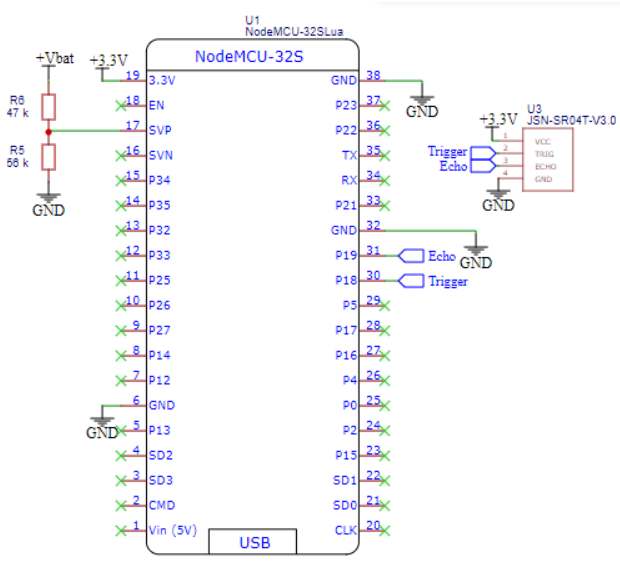

Figura 12: Conexionado para llevar a cabo las pruebas de funcionamiento del sistema mediante el módulo de desarrollo NODEMCU

Para el ensayo del funcionamiento del sistema se llevó a cabo el conexionado que se muestra en la *Figura 12* y se alimentó al módulo a través del puerto micro USB del mismo. Se ingresó la IP del servidor y puerto correspondiente a la aplicación Grafana y se pudo visualizar las variables sensadas correctamente.

Figura 13: Prueba de funcionamiento del Sistema

4. Estado actual del proyecto

Actualmente el proyecto se encuentra en la espera de componentes para la elaboración del PCB del prototipo basado en el esquemático que se presenta a continuación.

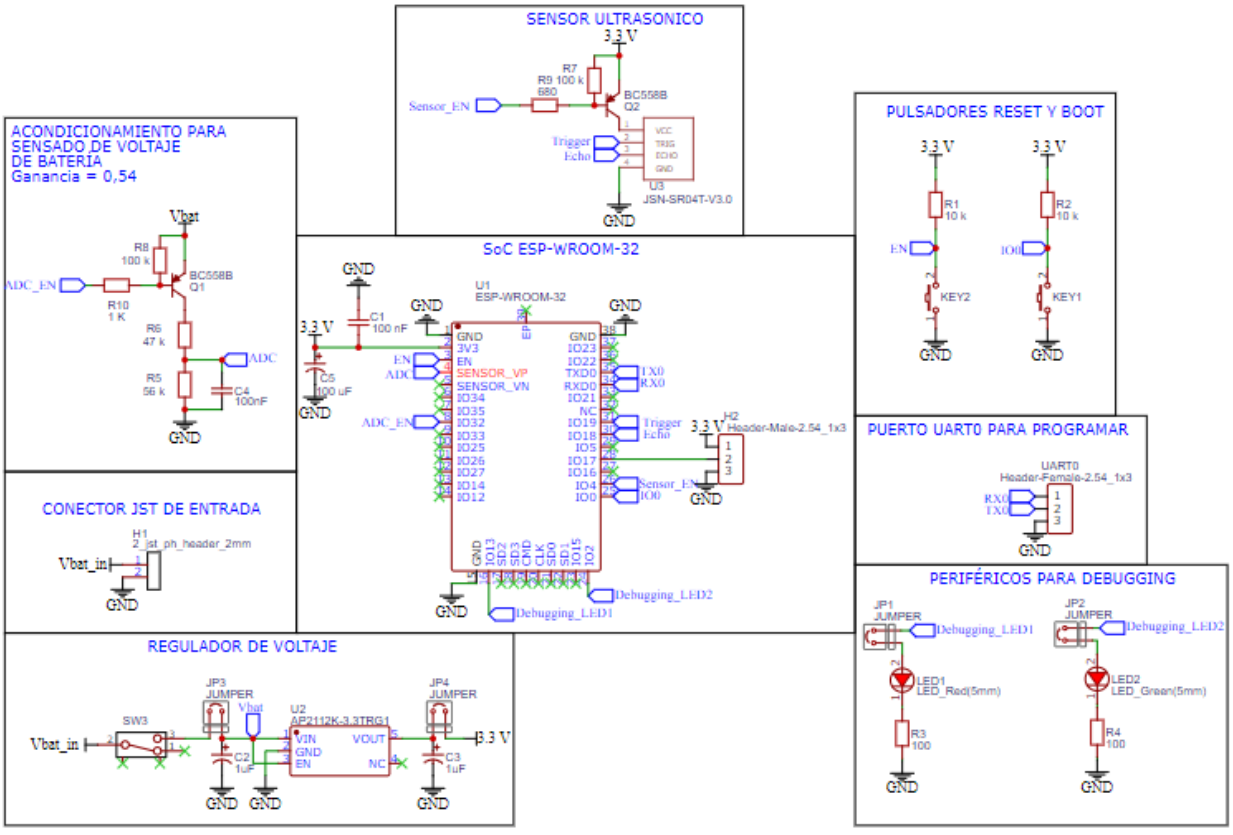

Figura 14: Esquema circuital del prototipo de PCB

5. Conclusiones

Se logró comprobar el correcto funcionamiento del sistema de medición inalámbrico en conjunto con el servidor configurado para poder visualizar en una interfaz web tanto la cantidad de contenido de agua en un tanque como el voltaje de la batería del sistema de medición.

También se pudo llegar a una expresión matemática que estima la duración de la batería del sistema de medición en función del tiempo de dormido que se configure en este.

6. Referencias

[1] Espressif, "modos de dormido del ESP32", (Accedido en junio de 2021) Disponible en: docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/sleep_modes.html.

[2] Espressif, "Hoja de referencia Técnica del SoC ESP32", (Accedido en mayo de 2021) Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.

[3] Espressif, "Hoja de datos del SoC ESP32", (Accedido en mayo de 2021) Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[4] Aprendiendoarduino, "Vinculación de Arduino con MQTT", (Accedido en mayo de 2021) Disponible en: https://aprendiendoarduino.wordpress.com/2018/11/19/mqtt/.

[5] Arduino, "Referencia de librería ArduinoMQTTClient", (Accedido en mayo de 2021) Disponible en: https://www.arduino.cc/reference/en/libraries/arduinomqttclient/.

[6] Jahankit, "Datasheet del sensor JSN-SR04-V3", (Accedido en mayo de 2021) Disponible en: https://www.jahankitshop.com/getattach.aspx?id=4635&Type=Product