

UNIVERSIDAD CATÓLICA BOLIVIANA "SAN PABLO"
UNIDAD ACADÉMICA REGIONAL COCHABAMBA
Departamento de Ingeniería y Ciencias Exactas
Internet de las Cosas



Integración de un dispositivo inteligente con una plataforma en la nube (AWS) con la ayuda del protocolo de aplicación MQTT

Integrantes: Denilson Fiesta Inocente
Jonathan Moruno Teran
Docente: Eduardo Enrique Marin Garcia

Cochabamba - Bolivia
15 de Mayo de 2024

I. INTRODUCCIÓN

En esta era de la digitalización, la tecnología se ha hecho cada vez mas presente en nuestra vida cotidiana, desde los dispositivos inteligentes en nuestros hogares hasta las ciudades inteligentes. El internet de las cosas (IoT) ha sido crucial para esta transformación, revolucionando nuestra interacción con el entorno que nos rodea.

El IoT se refiere a la transformación de los objetos físicos para dotarlos de inteligencia y la capacidad de conectarse a internet. Un objeto inteligente es aquel que cuenta con tecnología integrada, capaz de percibir e interactuar con su entorno. Estos objetos pueden comunicarse entre sí a través de internet, lo que les permite recopilar y compartir datos para automatizar tareas, mejorando la eficiencia y la calidad de vida.

Con la creciente presencia de estas tecnologías en nuestra sociedad, es esencial comprenderlas adecuadamente. Por esta razón, el objetivo del presente informe es diseñar e implementar un circuito funcional para un dispensador de agua automático, utilizando un sensor ultrasónico, un puente H, una mini bomba de agua sumergible y una placa ESP32. Se detallara el proceso de desarrollo del sistema, desde el diseño inicial hasta la implementación y las pruebas realizadas para verificar su funcionamiento

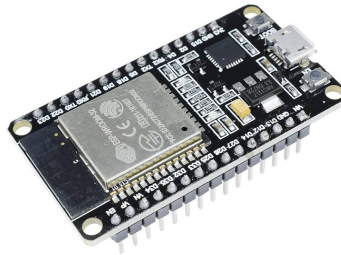
II. MARCO TEÓRICO

A continuación, se detallan los objetos inteligentes utilizados en la construcción del circuito y las herramientas de software que manejan la lógica funcional del proyecto:

■ MÓDULO ESP32

El modulo ESP32 es un microcontrolador con un procesador de doble núcleo y conectividad Wi-Fi y Bluetooth integrada. Actúa como el controlador central en varios sistemas, coordinando la comunicación entre actuadores y sensores. Se manipula mediante un cable micro USB conectado a una computadora y un IDE apropiado (TEJA, 2024).

Figura 1
MÓDULO ESP-WROOM-32



■ **SENSOR ULTRASÓNICO HC-SR04**

El sensor ultrasónico es un dispositivo que utiliza ondas de sonido de alta frecuencia para medir distancias. Este emite pulsos de ultrasonido y mide el tiempo que tarda en recibir los ecos reflejados, lo que le permite determinar la distancia a un objeto. Este sensor se compone de cuatro pines principales: VCC (Alimentación), Trig (Disparador), Echo (Eco) y GND (Tierra) siendo los pines VCC y GND los que proveen de energía a este sensor. Sin embargo presenta limitaciones, como la interferencia causada por objetos que absorben o reflejan el ultrasonido de forma irregular (ROBOCRAZE, 2022).

Figura 2
SENSOR ULTRASÓNICO HC-SR04

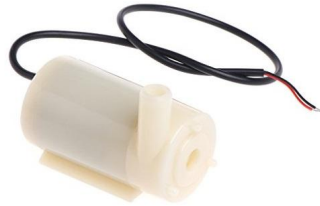


■ **MINI BOMBA DE AGUA SUMERGIBLE**

Una mini bomba de agua sumergible es un dispositivo compacto diseñado para el bombeo eficiente de agua en diversas aplicaciones. Dicho dispositivo permite impulsar el agua que recibe en la entrada, obteniendo una salida de aún mayor

presión. Cuenta además con un caudal de bombeo entre 80 y 120 litros por hora (UNIT ELECTRONICS, 2018).

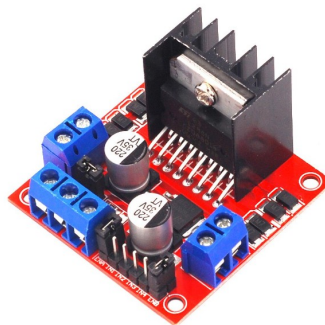
Figura 3
MINI BOMBA DE AGUA SUMERGIBLE



■ DRIVER PUENTE H L298N

El driver puente H L298N es el módulo más usado para el control de motores de corriente directa. Este nos permite mediante señales digitales el control de la velocidad y el sentido de giro de motores, que claramente están integrados en bombas de agua. Además, su diseño robusto y su capacidad para manejar corrientes de hasta 2 amperios lo hacen ideal para una amplia gama de aplicaciones en robótica y automatización (NAYLAMP MECHATRONICS, 2023).

Figura 4
DRIVER PUENTE H L298N

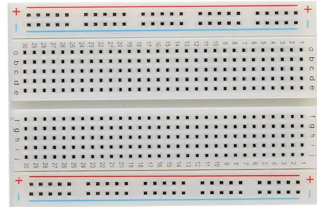


■ PROTOBOARD

Una protoboard es una placa de circuito impreso que se utiliza para realizar conexio-

nes entre componentes electrónicos. Esta se divide en filas y columnas de agujeros interconectados eléctricamente, lo que facilita la conexión de componentes sin la necesidad de soldadura (Portal Académico del CCH, 2021).

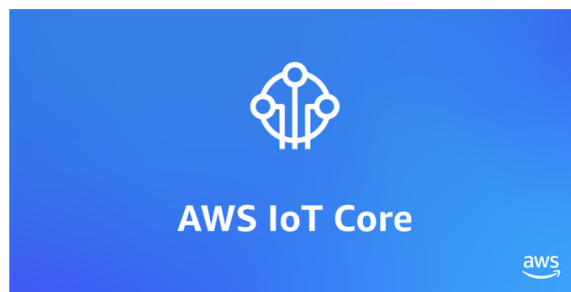
Figura 5
PROTOBOARD DE 400 PINES



■ AWS IoT CORE

AWS IoT Core es un servicio en la nube proporcionado por Amazon Web Services (AWS) que permite la conexión de dispositivos IoT a los servicios de AWS. AWS IoT Core también dirige billones de mensajes a los servicios de AWS. Podemos realizar todo esto sin necesidad de administrar la infraestructura de este servicio (AMAZON WEB SERVICES, 2023).

Figura 6
Logo AWS IoT CORE

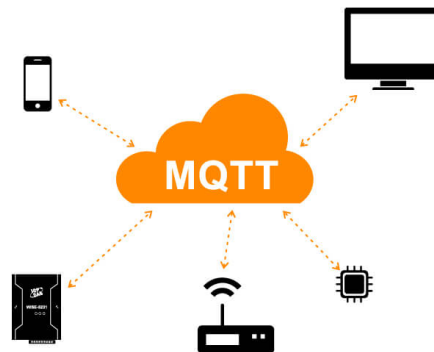


■ MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería liviano. AWS IoT Core cuenta con alta compatibilidad con este protocolo, pudiendo admitir conexiones de dispositivos que utilizan el mismo no sin antes identificarse

mediante un ID de cliente. MQTT sigue un modelo de comunicación de publicación-suscripción, donde los dispositivos publican mensajes en tópicos o temas y los suscriptores interesados reciben esos mensajes (MQTT, s.f.).

Figura 7
PROTOCOLO MQTT



■ **AWS IoT CORE SHADOWS**

Las sombras son un reflejo virtual del estado de un dispositivo. Algunas operaciones que los clientes web, otros dispositivos y servicios pueden realizar mediante el protocolo MQTT con sombras son: Su creación, actualización y eliminación mediante la consola de AWS IoT. Los objetos de AWS IoT no tienen sombras cuando se crean sin embargo pueden llegar a tener varias (AMAZON WEB SERVICES, 2024).

■ **AWS IoT CORE RULES**

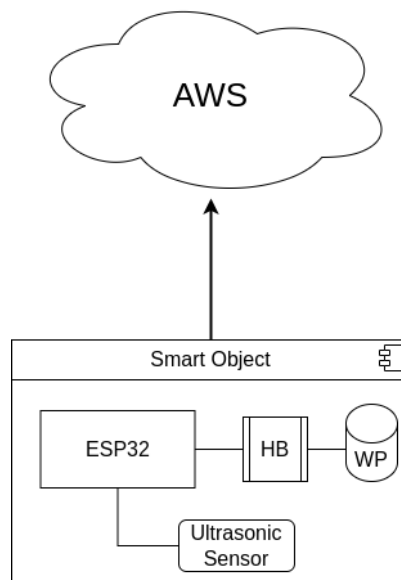
Las reglas otorgan a los dispositivos de AWS IoT Core con los que están relacionadas la capacidad de interactuar con los servicios de AWS. Estas usan mensajes MQTT que pasan por los protocolos de publicación o suscripción que sean compatibles con los protocolos de comunicación del dispositivo. Al comprender y aprovechar adecuadamente las reglas, se usa el potencial completo de AWS IoT Core para crear soluciones IoT innovadoras que resuelven problemas reales (AMAZON WEB SERVICES, 2024).

III. ANÁLISIS Y DISEÑO

Se ha diseñado un sistema de integración que asegura su correcto funcionamiento y cumple con los objetivos del informe. El sistema propuesto está compuesto por un módulos ESP32, un sensor ultrasónico, un puente H L298N, una mini bomba de agua, un protoboard de 400 pines y cables jumper.

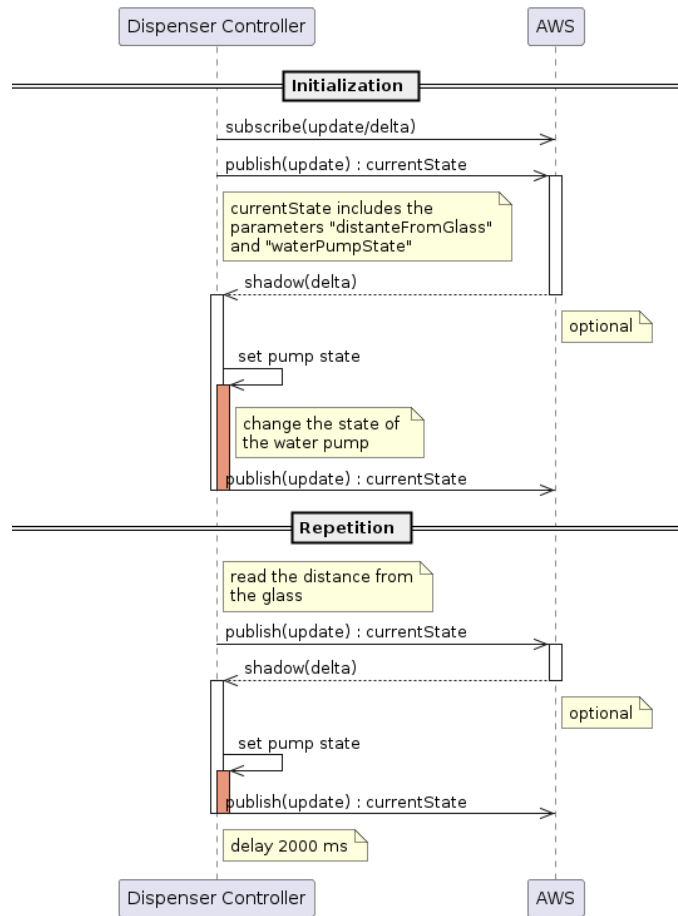
Inicialmente, se diseñó la estructura estática del sistema, que incluye los siguientes componentes: La plataforma AWS, encargado de gestionar el manejo de datos del dispositivo inteligente; el Dispositivo inteligente, compuesto de un dispositivo actuador y un dispositivo sensor. Esta disposición puede ser visualizada en la **Figura 8**.

Figura 8
DIAGRAMA DE ESTRUCTURA DEL SISTEMA



Posteriormente, se llevó a cabo un análisis y diseño de la secuencia de interacciones entre los componentes esenciales del sistema, tales como el dispositivo inteligente y la plataforma AWS. Este modelo se encuentra representado de manera visual en la **Figura 9**.

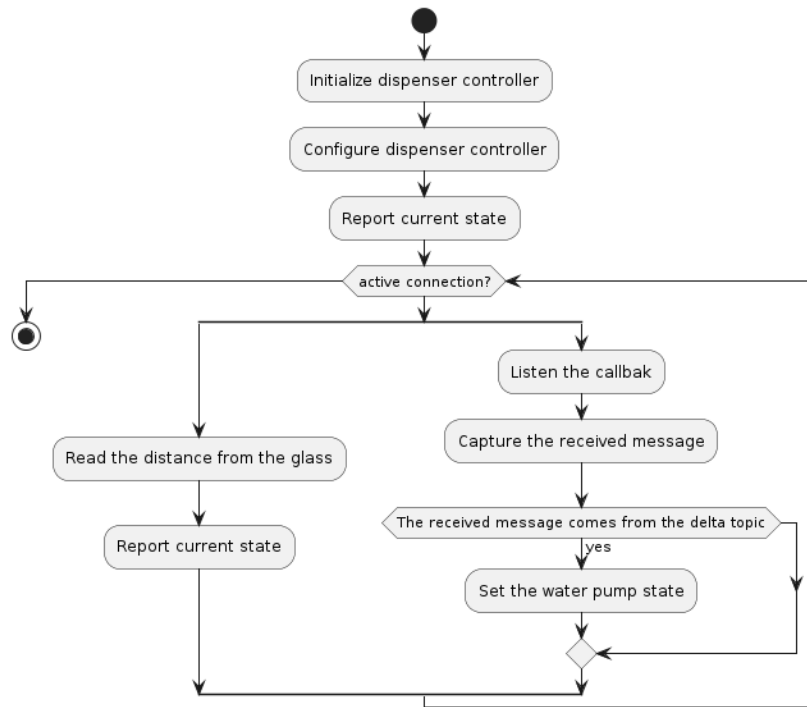
Figura 9
DIAGRAMA DE SECUENCIA DEL SISTEMA



Para una mejor comprensión de los procesos llevados a cabo por el dispositivo inteligente y la plataforma AWS, se han elaborado los siguientes diagramas de flujo, presentados en las **Figuras 10, 11 y 12**.

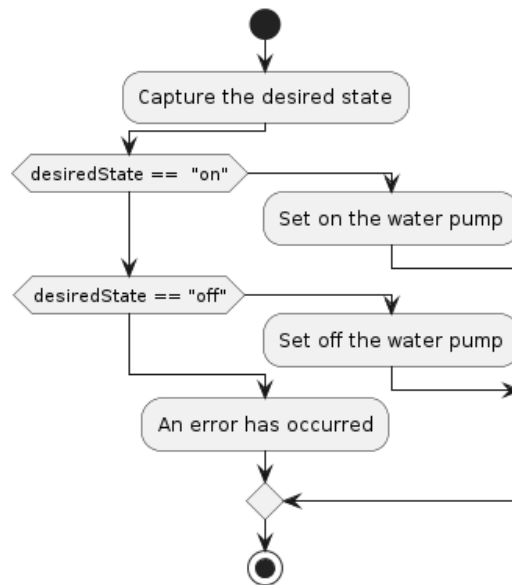
En la **Figura 10**, se detallan los procedimientos involucrados en el funcionamiento general del dispositivo inteligente.

Figura 10
DIAGRAMA DE FLUJO DEL DISPOSITIVO INTELIGENTE



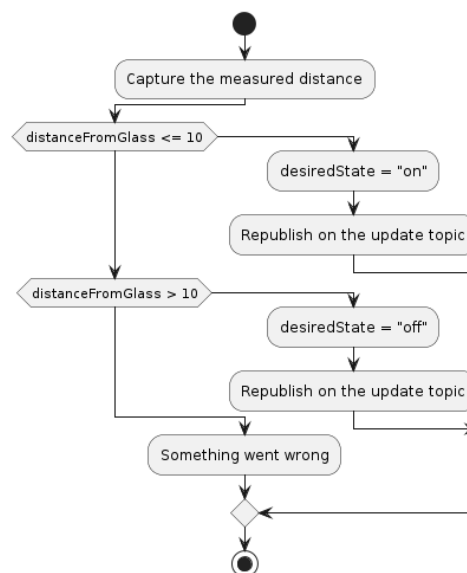
La **Figura 11** describe el proceso que sigue el objeto inteligente una vez que este captura un mensaje en el t3pico "update/delta", significando que existe un cambio de estados.

Figura 11
DIAGRAMA DE FLUJO DEL MANEJO DEL ESTADO DEL DISPENSADOR



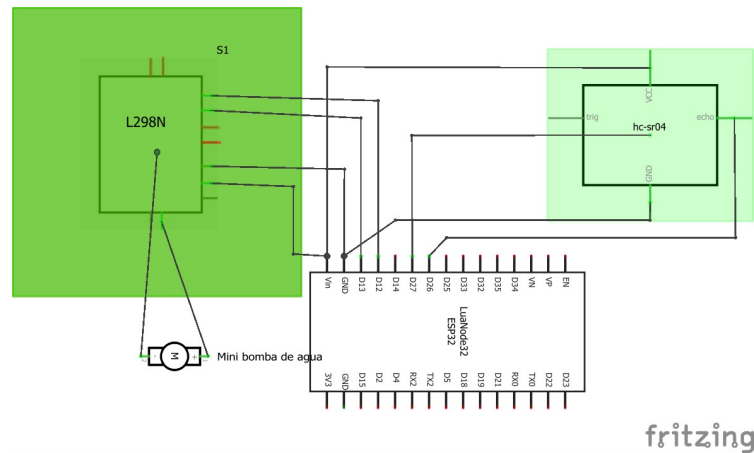
En la **Figura 12**, se muestra el proceso específico que la plataforma sigue cuando existen publicaciones en el tópico "update", esto realizándolo a través de las reglas de AWS IoT Core. Aclarar que se unieron la reglas bajo un mismo diagrama de flujo.

Figura 12
DIAGRAMA DE FLUJO DEL MANEJO DE REGLAS EN AWS



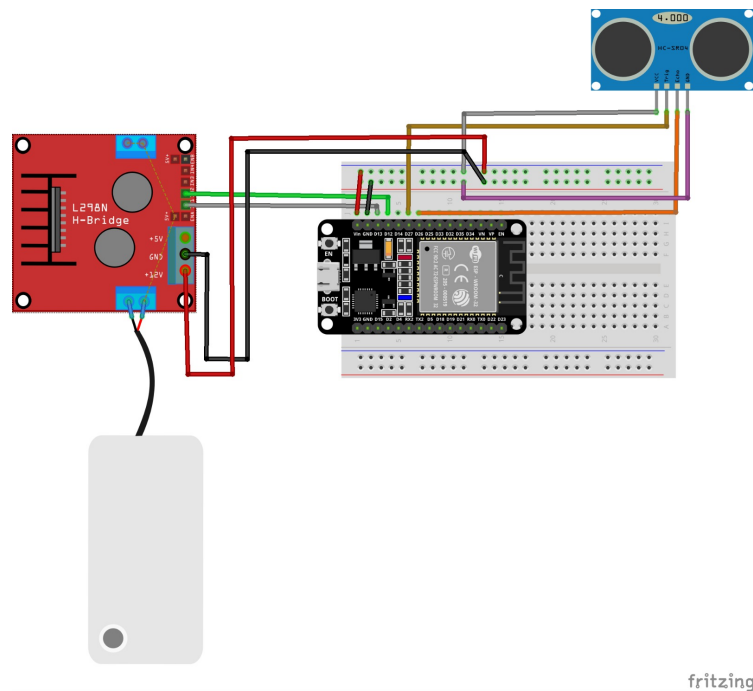
Posteriormente, se elaboraron diagramas de circuito para el dispositivo inteligente. El diagrama representa la disposición física y la conexión de los componentes electrónicos en el sistema. Este diagrama puede ser visualizado en la **Figura 13**.

Figura 13
DIAGRAMA DE CIRCUITO DEL DISPOSITIVO INTELIGENTE



Finalmente se muestra un prototipo de los componentes conectados. Este prototipo puede ser visualizado en la **Figura 14**.

Figura 14
PROTOTIPO DEL DISPOSITIVO INTELIGENTE



IV. DESARROLLO E IMPLEMENTACIÓN

Para lograr una funcionalidad completa del sistema, se programó el servidor TCP, la placa ESP32 del dispositivos actuador y sensor, con el objetivo de controlar la comunicación entre estos componentes a través de protocolos establecidos.

El código fuente utilizado en el presente informe se encuentra disponible en el siguiente enlace: <https://github.com/Al1bi/Practica3-IoT/tree/main>.

El código fuente se desarrolló utilizando el paradigma de programación orientada a objetos (POO). Además, se siguieron convenciones de código, como la calistenia de objetos, con el objetivo de hacer el código fuente más legible y escalable.

A continuación se da una explicación breve de la siguientes clases:

- **AWSDispenserController.ino**: Controla el dispensador de agua, coordinando las

acciones de dispensación según las instrucciones recibidas del AWS.

- **Dispenser.cpp:** Implementa la funcionalidad principal del dispensador, incluyendo la activación y desactivación del dispositivo. Trabaja en conjunto con el UltrasonicSensor y WaterPump.
- **UltrasonicSensor.cpp:** Controla el sensor ultrasónico utilizado para medir la distancia y detectar la presencia de objetos, proporcionando información para el control del dispensador.
- **WaterPump.cpp:** Controla la bomba de agua del dispensador, permitiendo su activación y desactivación según sea necesario.
- **MQTTManager.cpp:** Gestiona la comunicación mediante el protocolo MQTT, enviando y recibiendo mensajes entre el dispositivo y AWS.
- **MQTTConfig.cpp:** Responsable de configurar la conexión y los parámetros de comunicación para el protocolo MQTT utilizado en la interacción con el servidor.
- **AmazonCredentials.cpp** Se encarga de gestionar las credenciales necesarias para la autenticación y autorización en los servicios de Amazon Web Services (AWS).
- **WiFiManager.cpp:** Encargado de manejar la conexión WiFi del dispositivo, asegurando una conexión estable y confiable con el servidor TCP.

V. PRUEBAS Y VALIDACIONES

■ PRECISIÓN Y EXACTITUD DEL SENSOR DE PROXIMIDAD

Para obtener la precisión y exactitud del sensor ultrasónico se requirió de dos herramientas: El Serial Monitor provisto por el programa Arduino IDE y un flexómetro. Ambas herramientas nos dan la distancia que existe hacia un objeto, siendo el Serial Monitor el que provee valores con errores y el flexómetro quien otorga valores “reales” sin considerar el error humano. Tanto los valores con error y lo que “no tienen” fueron comparados, obteniéndose la siguiente tabla con promedios y desviaciones estándar de 50 datos por medida y un promedio para el error relativo porcentual. Para más detalles respecto a como se obtuvieron los siguientes datos, véase **Figura 17** en **ANEXOS**.

Figura 15

Resultados obtenidos de medias, desviaciones y errores de las lecturas del sensor a diferentes distancias

Distancia	Media del sensor [cm]	Desviacion estandar del sensor	Error relativo porcentual por medida del sensor [%]
1cm	4.4	0.494871659	340
2cm	2.04	0.197948664	2
10cm	9.94	0.239897937	0.6
20cm	19.92	0.274047516	0.4
30cm	30.1	0.303045763	0.333333333
40cm	39.98	0.141421356	0.05
50cm	48.92	0.274047516	2.16
200cm	199.04	0.197948664	0.48
Promedio error relativo porcentual sin considerar el primer registro:			0.86047619

■ **TIEMPO Y CAUDAL DE LLENADO DE UN VASO DE 130 ML CON MINI BOMBA DE AGUA**

Para obtener el caudal y el tiempo en el que se llena un vaso de 130 mililitros con una mini bomba de agua se requirió de un cronometro. Esta herramienta nos brinda un tiempo en segundos con una precisión de dos decimales. Se registro un total de 30 datos obteniéndose la siguiente tabla, con un registro de tiempo en segundos por iteración, un promedio de tiempo de llenado y el caudal de la mini bomba de agua en litros por hora. Para más detalles respecto a cómo se obtuvieron los siguientes datos, véase este mismo punto en **ANEXOS**.

Figura 16
Resultados obtenidos de tiempos de llenado de la mini bomba de agua

Nro	t [s]
1	5.75
2	5.7
3	5.9
4	5.69
5	5.89
6	5.76
7	5.85
8	5.89
9	5.69
10	5.94
11	5.72
12	5.75
13	5.86
14	5.68
15	5.77
16	5.84
17	5.71
18	5.86
19	5.67
20	5.82
21	5.76
22	5.9
23	5.68
24	5.92
25	5.74
26	5.76
27	5.83
28	5.72
29	5.94
30	5.7
Tiempo de llenado promedio	5.789666667
Caudal [L/H]	80.83366918

VI. RESULTADOS Y CONCLUSIONES

- El sensor de proximidad presenta una alta exactitud al menos en el intervalo de medida para el cual está diseñado: 2 cm a 200cm, ya que se determinó un error relativo porcentual promedio de 0,86 %, este error claramente se considera bajo.
- El caudal que se encontró para la mini bomba de agua sumergible está dentro del

rango especificado para este, lo cual sugiere que está funcionando correctamente y dentro de los parámetros esperados para su operación.

VII. BIBLIOGRAFÍA

- AMAZON WEB SERVICES. (2023). “AWS IoT Core”. Recuperado de <https://aws.amazon.com/es/iot-core/?nc=sn&loc=0> (último acceso: 10/05/2024).
- AMAZON WEB SERVICES. (2024). “AWS IoT Device Shadow service”. Recuperado de <https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html> (último acceso: 10/05/2024).
- AMAZON WEB SERVICES. (2024). “Rules for AWS IoT”. Recuperado de <https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html> (último acceso: 10/05/2024).
- IBM (s.f.). “What is the Internet of Things (IoT)?”. En: <https://www.ibm.com/topics/internet-of-things>, (fecha de consulta 10/05/2024).
- MQTT (s.f.). “The Standard for IoT Messaging”. <https://mqtt.org/>
- NAYLAMP MECHATRONICS (2023). “DRIVER PUENTE H L290N 2A DESCRIPCIÓN INFO”. En: <https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html>, (fecha de consulta 10/05/2024).
- ROBOCRAZE (2022). “What is Ultrasonic Sensor: Working Principle & Applications”. En: https://robocraze.com/blogs/post/what-is-ultrasonic-sensor?_pos=1&_psq=what+is+ultr&_ss=e&_v=1.0, (fecha de consulta 10/05/2024).
- TEJA, Ravi (2024). “Getting Started with ESP32 | Introduction to ESP32”. En: <https://www.electronicshub.org/getting-started-with-esp32/>, (fecha de consulta 10/05/2024).
- UNIT ELECTRONICS (2018). “Hoja de datos de Bomba de Agua sumergible Arduino”. En: <https://uelectronics.com/wp-content/uploads/2018/02/Hoja-de-datos-Bomba-de-Agua-sum.pdf>, (fecha de consulta 10/05/2024).
- Portal Académico del CCH (2021). “Tableta protoboard”. Recuperado de <https://portalacademico.cch.unam.mx/cibernetica1/implementacion-de-circuitos-logicos/tableta-protoboard>.

VIII. ANEXOS

■ PRECISIÓN Y EXACTITUD DEL SENSOR DE PROXIMIDAD

Figura 17
Resultados obtenidos del sensor de proximidad a diferentes distancias

Numero de Iteración	1cm	2cm	10cm	20cm	30cm	40cm	50cm	200cm
1	5	3	10	19	31	39	49	199
2	5	3	10	20	31	40	48	199
3	5	2	9	20	31	40	48	199
4	5	2	10	20	31	40	48	199
5	5	2	10	20	31	40	49	199
6	5	2	9	20	30	40	49	199
7	5	2	10	20	30	40	49	199
8	5	2	10	20	30	40	49	199
9	5	2	10	20	20	40	49	199
10	5	2	10	20	20	40	49	199
11	5	2	10	20	30	40	49	199
12	5	2	10	20	30	40	49	199
13	4	2	10	20	30	40	49	199
14	4	2	10	20	30	40	49	199
15	4	2	10	20	30	40	49	199
16	4	2	10	20	30	40	49	199
17	4	2	10	20	30	40	49	199
18	4	2	10	20	30	40	49	199
19	5	2	10	20	30	40	49	199
20	5	2	10	20	30	40	49	199
21	5	2	10	20	30	40	49	199
22	5	2	10	20	30	40	49	199
23	5	2	10	20	30	40	49	199
24	5	2	10	20	30	40	49	199
25	5	2	10	20	30	40	49	199
26	5	2	10	20	30	40	49	199
27	4	2	10	20	30	40	49	199
28	4	2	10	20	30	40	49	199
29	4	2	10	20	30	40	49	199
30	4	2	10	20	30	40	49	199
31	4	2	10	20	30	40	49	199
32	4	2	10	20	30	40	49	199
33	4	2	10	20	30	40	49	199
34	4	2	10	20	30	40	49	199
35	4	2	10	20	30	40	49	199
36	4	2	10	20	30	40	49	199
37	4	2	10	20	30	40	49	199
38	4	2	10	20	30	40	49	199
39	4	2	10	20	30	40	49	199
40	4	2	10	20	30	40	49	199
41	4	2	10	20	30	40	49	199
42	4	2	10	20	30	40	49	199
43	4	2	10	20	30	40	49	199
44	4	2	10	20	30	40	49	199
45	4	2	10	20	30	40	49	199
46	4	2	10	20	30	40	49	199
47	4	2	10	20	30	40	49	199
48	4	2	10	19	30	40	49	199
49	4	2	10	19	30	40	49	200
50	4	2	10	19	30	40	49	200
Promedio	4.4	2.04	9.94	19.92	29.7	39.98	48.92	199.04
Desviación estándar	0.494871659	0.197948664	0.239897937	0.274047516	2.022826694	0.141421356	0.274047516	0.197948664

El error porcentual relativo a diferentes distancias se determinó con la siguiente formula:

$$E \% = \frac{|DistanciaMedida - DistanciaReal|}{DistanciaReal} * 100$$

A continuación se detalla el procedimiento realizado para obtener los errores relativos porcentuales de las diferentes distancias.

- **Distancia de 1cm:**

$$\frac{|4,4-1|}{1} * 100 = 340$$

- **Distancia de 2cm:**

$$\frac{|2,04-2|}{2} * 100 = 2$$

- **Distancia de 10cm:**

$$\frac{|9,94-10|}{10} * 100 = 0,6$$

- **Distancia de 20cm:**

$$\frac{|19,92-20|}{20} * 100 = 0,4$$

- **Distancia de 30cm:**

$$\frac{|30,1-30|}{30} * 100 = 0,33$$

- **Distancia de 40cm:**

$$\frac{|39,98-40|}{40} * 100 = 0,05$$

- **Distancia de 50cm:**

$$\frac{|48,92-50|}{50} * 100 = 2,16$$

- **Distancia de 200cm:**

$$\frac{|199,04-200|}{200} * 100 = 0,48$$

El error relativo porcentual promedio se calculo usando la siguiente formula:

$$\overline{Errorrelativoporcentual} = \frac{\sum_{i=1}^n Errorrelativoporcentual}{n}$$

Donde n es la cantidad de errores relativos porcentuales.

$$\overline{Errorrelativoporcentual} = 0,86047619$$

■ **TIEMPO Y CAUDAL DE LLENADO DE UN VASO DE 130 ML CON MINI BOMBA DE AGUA**

El tiempo promedio de llenado de un vaso de 130 ml con nuestra mini bomba de agua se calculo con la siguiente formula:

$$\overline{Tiempodellenado} = \frac{\sum_{i=1}^n Tiempodellenado}{n}$$

Donde n es la cantidad de tiempos registrados.

Con el tiempo promedio de llenado se procede a calcular el caudal en litros por hora con la siguiente formula:

$$Caudal = \frac{\frac{Volumen}{\overline{Tiempodellenado}} * 1}{\frac{1000}{1}} * 3600$$

$$Caudal = 80,83366918[L/H]$$