



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE GUAYAQUIL**  
**CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO INTERACTIVO IOT USANDO  
DISPOSITIVOS LOW-COST Y SERVICIOS WEB.**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero Electrónico

**AUTORES:** Israel Fernando Brito Martínez  
Marlon Gabriel Arellano Barreiro

**TUTOR:** Ing. Orlando Giovanni Barcia Ayala, Msc.

Guayaquil – Ecuador

2023

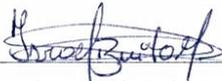
**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Nosotros, Israel Fernando Brito Martínez con documento de identificación N°0931240956 y Marlon Gabriel Arellano Barreiro con documento de identificación N°0950591875; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 14 de agosto del 2023

Atentamente,



Israel Fernando Brito Martínez

C.I: 0931240956



Marlon Gabriel Arellano Barreiro

C.I: 0950591875

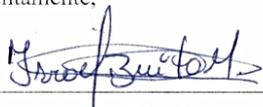
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE  
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Israel Fernando Brito Martínez con documento de identificación N°0931240956 y Marlon Gabriel Arellano Barreiro con documento de identificación N°0950591875, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO INTERACTIVO IOT USANDO DISPOSITIVOS LOW-COST Y SERVICIOS WEB”, el cual ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 14 de agosto del 2023

Atentamente,

  
Israel Fernando Brito Martínez  
C.I: 0931240956

  
Marlon Gabriel Arellano Barreiro  
C.I: 0950591875

### **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN**

Yo, Ing. Orlando Giovanni Barcia Ayala, Msc con documento de identificación N°1309445714, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO INTERACTIVO IOT USANDO DISPOSITIVOS LOW-COST Y SERVICIOS WEB”, realizado por Israel Fernando Brito Martínez con documento de identificación N°0931240956 y Marlon Gabriel Arellano Barreiro con documento de identificación N°0950591875, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 14 de agosto del año 2023

Atentamente,



---

Ing. Orlando Giovanni Barcia Ayala, Msc.

C.I: 1309445714

## **AGRADECIMIENTO**

Agradezco a Dios por guiarme y darme la fortaleza a seguir y no abandonar mis estudios y poder cumplir mis metas.

A mi familia por la paciencia y la comprensión la cual ha sido de gran impulso para no abandonar mis estudios.

A mis familiares que hoy en día no se encuentra en esta vida terrenal, pero viven en nuestros corazones y pensamientos, que de alguna manera me ha apoyado para cumplir este objetivo.

## **DEDICATORIA**

A mis padres quienes me enseñaron e impulsaron a seguir adelante en mis estudios para poder cumplir mis metas.

A mi esposa Emily y mi hijo Maximiliano que han sido el motor para seguir superándome cada día y no abandonar esta meta la cual será de gran ayuda para sacar adelante a mi familia.

A mis hermanos por ser una ayuda incondicional en este proceso y dejarle marcado el camino con mi ejemplo para que ellos también se conviertan en excelentes profesionales.

**Israel Fernando Brito Martínez**

## **AGRADECIMIENTO**

En primer lugar, agradezco a mi madre Rosa Barreiro, ya que ha sido el pilar principal de mi vida, me ha apoyado siempre desde pequeño y hasta el día de hoy lo sigue haciendo.

Agradezco también a mi tía Norma Barreiro ya que ha sido como una madre para mí y junto con mi madre me han apoyado mucho durante este proceso.

Agradezco también a mis amigos, Paúl, Byron y Katherine, los cuales siempre han estado a mi lado en los mejores y peores momentos de mi vida, sé que cuento y contaré con ellos siempre.

Agradezco también a mi compañero y amigo Israel y a mis amigos Fabricio y Melannie, los cuales han estado conmigo desde que empecé esta aventura y me han dado su apoyo incontables veces durante toda la carrera.

## **DEDICATORIA**

Dedico este logro a mis abuelos, Luis Meneses y Leonor Zárate, gracias a su apoyo económico y emocional todo esto fue posible. Me hubiera gustado que estén conmigo en este momento tan importante de mi vida, pero como siempre lo digo: “Desde el cielo sus ojos me guían”.

**Marlon Gabriel Arellano Barreiro**

## ÍNDICE GENERAL

AGRADECIMIENTO .....	IV
AGRADECIMIENTO .....	V
DEDICATORIA .....	V
ÍNDICE GENERAL .....	VI
ÍNDICE DE FIGURAS.....	IX
ÍNDICE DE TABLAS .....	X
RESUMEN .....	XI
ABSTRACT .....	XII
Introducción.....	1
<b>1. Descripción del Problema .....</b>	<b>2</b>
1.1. Descripción general del problema.....	2
1.2. Antecedentes .....	3
1.3. Importancia y Alcance.....	3
1.4. Delimitación.....	6
1.4.1. Delimitación Temporal .....	6
1.4.2. Delimitación Espacial .....	6
1.4.3. Delimitación Académica .....	6
1.5. Objetivos.....	6
1.5.1. Objetivo General.....	6
1.5.2. Objetivos Específicos.....	6
<b>2. Fundamentos Teóricos .....</b>	<b>8</b>
2.1. ESP32 .....	8
2.2. Sensores Lowcost .....	9
2.3. Plataforma IoT: Ubidots .....	12
2.4. El modelo TCP/IP .....	13
2.5. Protocolo TCP.....	13
2.6. Protocolo UDP.....	15
2.7. Protocolo HTTP .....	16
2.8. Protocolo MQTT .....	16
2.9. Comparativa entre protocolos de comunicación.....	18
2.10. Protocolos TCP, UDP y HTTP para la conexión del ESP32 a la plataforma Ubidots. 20	

2.10.1.	Envío de variables a la plataforma de Ubidots.....	20
2.10.2.	Obtener variables desde Ubidots.....	20
2.11.	Protocolo MQTT para la conexión del ESP32 a la plataforma Ubidots.....	21
2.11.1.	Publicar valores en una variable.....	21
2.11.2.	Paso 1:TOKEN.....	21
2.11.3.	Paso 2: Definir las librerías.....	21
2.11.4.	Definir las instancias y constantes:.....	21
2.11.5.	Paso 3: Envío de variables a la plataforma de Ubidots.....	21
<b>3.</b>	<b>Marco metodológico .....</b>	<b>22</b>
3.1.	Propuesta de solución.....	22
3.2.	Diseño del módulo.....	23
3.3.	Programación (IDE Arduino).....	24
3.4.	Paso 1: Cabecera del Código.....	24
3.5.	Paso 2: Variables para enviar de datos a Ubidots.....	24
3.6.	Paso 3: Función Setup.....	25
3.7.	Paso 4: Función Loop.....	25
3.8.	Configuración Ubidots.....	25
3.8.1.	Paso 1: Crear un nuevo dispositivo.....	25
3.8.2.	Paso 2: Crear variables.....	26
3.8.3.	Paso 3: Enlazar las variables con el Dashboard.....	26
3.8.3.1.	Widgets.....	27
3.9.	App Inventor.....	28
3.9.1.	Interfaz de usuario.....	28
3.9.2.	Interfaz de usuario final de la app.....	29
3.10.	Implementación del módulo.....	30
3.10.1.	Selección y Diseño de la Caja.....	30
3.10.2.	Preparación de los Huecos y Espacios.....	31
3.10.3.	Montaje de la Pantalla LCD.....	31
3.10.4.	Instalación del Selector de Dos Posiciones.....	31
3.10.5.	Montaje de la Luz Piloto (Voltímetro).....	32
3.10.6.	Integración del Botón de Paro de Emergencia.....	32
3.10.7.	Incorporación del Plafón y Fuente AC.....	33
3.10.8.	Conexión del Selector al Ventilador.....	33
3.10.9.	Prueba y conexión.....	34

<b>4. Resultados</b> .....	<b>35</b>
CRONOGRAMA.....	38
PRESUPUESTO.....	39
Conclusiones.....	40
Recomendaciones .....	41
<b>Bibliografía</b> .....	<b>42</b>
ANEXOS.....	44

## ÍNDICE DE FIGURAS

Figura 1. Sensor Mq-135 .....	9
Figura 2. Sensor HC-SR04 .....	10
Figura 3. Sensor LDR LM393. ....	11
Figura 4. Sensor DHT22.....	11
Figura 5. Plataforma Ubidots.....	12
Figura 6. Protocolo TCP/Ip.....	13
Figura 7. Protocolo TCP .....	14
Figura 8. Protocolo UDP. ....	15
Figura 9. Diseño Prototipo V1. ....	23
Figura 10. Dispositivos Ubidots.....	26
Figura 11. Variables Ubidots .....	26
Figura 12. Interfaz gráfica Ubidots .....	26
Figura 13. Widgets Metrics Ubidots. ....	27
Figura 14. Plataforma Web APPinventor. ....	28
Figura 15. Pantalla de desarrollo APPinventor.....	29
Figura 16. Interfaz APP offline .....	29

## ÍNDICE DE TABLAS

Tabla 1. Comparativa entre ESP32 y el ESP8266 .....	8
Tabla 2. Tabla Comparativa entre TCP/UDP/HTTP/MQTT .....	18
Tabla 3. Conexiones del Prototipo. ....	23

## RESUMEN

El presente proyecto tiene como objetivo principal diseñar e implementar un módulo interactivo basado en IoT utilizando el ESP32. La iniciativa busca aprovechar la tecnología de Internet de las cosas para establecer una comunicación fluida entre personas, procesos y dispositivos, mejorando la interacción entre los miembros de la comunidad académica y facilitando el control remoto de equipos y sistemas.

El módulo interactivo debe ser equipado con sensores, luces piloto y actuadores, permitiendo el monitoreo y control de diferentes sensores y actuadores. Además, se integra a la plataforma IoT de Ubidots, en la cual se puede configurar la comunicación mediante los diferentes protocolos conocidos.

La versatilidad del proyecto radica en su enfoque práctico y didáctico. Se pretende utilizar un banco de conectores plug que facilita la realización de prácticas, pruebas y diferentes aplicaciones, fomentando el aprendizaje y la capacitación en el uso de tecnologías emergentes en el campo de la automatización y la domótica.

Además, el módulo interactivo IoT se configura con un sistema de alertas y notificaciones en tiempo real, para que los usuarios reciban información relevante acerca de cualquier eventualidad o situación que requiera su atención inmediata. Esto garantiza una respuesta rápida y efectiva ante posibles problemas o emergencias.

El resultado final será la creación de un panel de control personalizado que los usuarios, como estudiantes, profesores e investigadores, pueden acceder desde cualquier dispositivo con conexión a Internet. Así, se establece un método de aprendizaje más eficiente y colaborativo.

**Palabras claves:** ESP32, IoT, panel de control, domótica, automatización, internet, sensores, actuadores.

## ABSTRACT

The main objective of this project is to design and implement an interactive module based on IoT using the ESP32. The initiative seeks to take advantage of Internet of Things technology to establish fluid communication between people, processes and devices, improve interaction between members of the academic community and facilitate remote control of equipment and systems.

The interactive module will be equipped with sensors, pilot lights and actuators, allowing the monitoring and control of different sensors and actuators. In addition, it will be integrated into the Ubidots IoT platform, in which we can configure communication through the different known protocols.

The versatility of the project lies in its practical and didactic approach. It is intended to use a bank of plug connectors that will facilitate practices, tests and different applications, promoting learning and training in the use of emerging technologies in the field of automation and home automation.

In addition, the interactive IoT module will be configured with a real-time alert and notification system, so that users receive relevant information about any eventuality or situation that requires their immediate attention. This will guarantee a quick and effective response to possible problems or emergencies.

The end result will be the creation of a personalized control panel that users, such as students, teachers and researchers, will be able to access from any device with an Internet connection. Thus, a more efficient and collaborative learning method will be established.

**Keywords:** ESP32, IoT, control panel, home automation, automation, internet, sensors, actuators.

## **Introducción**

En la era actual de la tecnología interconectada, el Internet de las Cosas (IoT) ha revolucionado la forma en que se interactúa con el mundo digital. Esta innovadora tecnología permite la conexión y comunicación fluida entre dispositivos, sistemas y personas, abriendo un mundo de posibilidades en términos de eficiencia, control y monitoreo. Sin embargo, en los laboratorios de la Universidad Politécnica Salesiana, se ha identificado un desafío: la falta de módulos domóticos con plataforma IoT accesibles para la comunidad universitaria.

El presente proyecto tiene como objetivo abordar esta problemática y brindar una solución efectiva para impulsar el desarrollo de habilidades técnicas en el área de ingeniería. Se propone el desarrollo de un módulo interactivo basado en el ESP32, un microcontrolador versátil y de bajo costo, que integra la plataforma IoT para permitir una comunicación fluida entre personas, procesos y dispositivos.

Mediante la implementación de este módulo interactivo, se busca fomentar el aprendizaje práctico, la investigación y la colaboración en el entorno universitario. Los estudiantes y docentes tener la posibilidad de realizar prácticas, experimentos y proyectos utilizando sensores, actuadores y una variedad de protocolos de comunicación, como TCP, UDP y HTTP.

Además, la plataforma IoT elegida, Ubidots, brinda la posibilidad de configurar sistemas de alertas y notificaciones en tiempo real, lo que permite a los usuarios recibir información relevante sobre el estado y funcionamiento de los dispositivos conectados al módulo ESP32. Asimismo, se abre la posibilidad de crear paneles de control personalizados para manejar los dispositivos desde cualquier dispositivo conectado a la plataforma.

Se espera que este proyecto fomente la investigación y la innovación tecnológica en la Universidad Politécnica Salesiana. La comunidad de la UPSG tiene la posibilidad de desarrollar proyectos más complejos e innovadores, explorando las posibilidades del módulo interactivo y la plataforma IoT. Estos avances podrían contribuir al desarrollo de soluciones tecnológicas aplicadas en diversas áreas, como la domótica, la monitorización ambiental, la automatización industrial y la agricultura inteligente, entre otras.

## **1. Descripción del Problema**

### **1.1. Descripción general del problema**

En los laboratorios de la Universidad Politécnica Salesiana se dispone de varios módulos domóticos, pero son pocos los que cuenta con una plataforma IoT (Internet of things) "internet de las cosas". En este proyecto, se presenta un enfoque más amplio que incluye un ESP32 y la plataforma IoT como una novedad para la comunicación fluida entre personas, procesos y dispositivos. Además, se incluye un banco de conectores plug que permite realizar prácticas, pruebas y diferentes aplicaciones enfocadas a la adquisición y monitoreo de datos.

Es un hecho cierto que las conexiones mediante plataformas IoT tal y como se puede ver en la figura 1, han existido durante un tiempo. Avances recientes en varias tecnologías las han hecho prácticas, y las siguientes son algunas de ellas:

Se ha introducido la tecnología de sensores de bajo costo y bajo consumo, la cual está dando paso a la tecnología IoT y sumando cada día más fabricantes. En cuanto a la conectividad, existen una gran cantidad de protocolos de comunicación para Internet, lo que ha facilitado la conexión de sensores a otros dispositivos para lograr un tráfico de datos más eficiente. También ha habido un gran aumento en la disponibilidad de plataformas informáticas en la nube, lo que permite a las empresas y usuarios acceder a las infraestructuras que necesitan ascender sin la necesidad de supervisión u operación.

Gracias a los avances en aprendizaje automático y analítico, junto con la accesibilidad a grandes cantidades de datos variados almacenados en la nube, los usuarios pueden almacenar toda su información de manera mucho más rápida y sencilla. Debido al surgimiento de estas tecnologías, se siguen traspasando los límites de las plataformas IoT. Además, los datos generados por la plataforma IoT también sustentan a otras tecnologías.

## 1.2. Antecedentes

Antecedentes. - En 1874 se desarrolló el primer dispositivo de información meteorológica y de profundidad de la nieve en Mont Blanc, este fue conocido como el primer experimento de telemetría de la historia. Un equipo de investigadores franceses realizó la instalación de varios de estos dispositivos. Los datos fueron transmitidos hacia París mediante un enlace de radio. (Santos, 2020)

La evolución de estas tecnologías de la comunicación permitió el uso y desarrollo de la telemetría durante el siglo XX. En 1970, el Departamento de ciencias de la computación del Carnegie Mellon realizó la conexión de una máquina de Coca-Cola a un servidor de un área específica mediante la conexión de varios microswitches.

A inicios de los años 90 John Romkey crea el primer dispositivo con conectividad IoT, realizando una tostadora conectada a un ordenador. La comunicación se realizó mediante el protocolo TCP/IP y el control se realizó mediante el Protocolo simple de administración de redes. La tostadora podía controlarse desde un ordenador conectado a la red, controlando su encendido, apagado y tiempo de “tostado”. La única intervención humana era colocar el pan en la tostadora.

## 1.3. Importancia y Alcance

El proyecto de desarrollo de un módulo interactivo con plataforma IoT basada en el ESP32 para la comunidad universitaria tiene una gran importancia en diversos aspectos:

1. **Desarrollo de habilidades técnicas:** La implementación de este módulo brinda a los estudiantes la oportunidad de adquirir habilidades prácticas en el campo de la ingeniería y la tecnología IoT. Haciendo posible la familiarización con el uso de sensores, actuadores y protocolos de comunicación, lo que fortalece su formación académica y su capacidad para enfrentar desafíos tecnológicos.

2. **Promoción de la investigación y la innovación:** El proyecto fomenta la investigación y la generación de conocimiento en el campo de la IoT. Los estudiantes y profesores pueden explorar nuevas aplicaciones, desarrollar proyectos innovadores y contribuir al avance tecnológico en áreas como la domótica, la automatización industrial, la agricultura inteligente, entre otras.
3. **Colaboración y trabajo en equipo:** El módulo interactivo facilita la colaboración entre estudiantes y profesores, promoviendo el trabajo en equipo y el intercambio de conocimientos. Los proyectos conjuntos permiten abordar desafíos más complejos y fomentar el aprendizaje colectivo, enriqueciendo la experiencia universitaria.

**Alcances:**

El proyecto tiene un amplio alcance en términos de aplicaciones y beneficios para la comunidad universitaria.

1. **Aprendizaje práctico y experiencial:** Los estudiantes pueden aplicar los conocimientos teóricos adquiridos en las aulas a través de la realización de prácticas y proyectos con el módulo interactivo. Esto les brinda una comprensión más profunda de los conceptos de IoT y les permite adquirir experiencia práctica en la implementación de soluciones tecnológicas.
2. **Generación de proyectos innovadores:** La disponibilidad de un módulo interactivo con plataforma IoT abre nuevas posibilidades para la generación de proyectos innovadores. Los estudiantes pueden desarrollar soluciones tecnológicas creativas y aplicarlas en diferentes campos, contribuyendo así al desarrollo de la comunidad universitaria y su entorno.
3. **Integración de tecnologías emergentes:** El proyecto permite la integración de tecnologías emergentes, como la IoT, la computación en la nube y la analítica de datos. Esto les permite a los estudiantes prepararse para enfrentar los desafíos tecnológicos del mundo laboral actual.

## Justificación

La tecnología de las plataformas IoT ha revolucionado la forma en que interactúa el hombre con el mundo digital, permitiendo conectar dispositivos para obtener información y mejorar la eficiencia de los procesos. En este sentido, el presente proyecto tiene como objetivo explorar la aplicación de la plataforma IoT en un módulo programable, específicamente en el ESP32.

Este módulo programable es capaz de controlar equipos a través de sus salidas y programar sus entradas para crear una interacción más fluida entre sistemas y personas. Al incorporar la plataforma IoT en este módulo, se logra una conexión a la nube que puede ser activada y configurada directamente, permitiendo crear paneles de control personalizados para manejarlos desde cualquier dispositivo.

Este enfoque más variado sobre el autómata programable y la plataforma IoT es una novedad para la comunicación fluida entre personas, procesos y sistemas de control, ya que permite la creación de prácticas y programaciones con aplicaciones externas a través del banco de conectores plug, lo que aumenta la versatilidad del proyecto.

Es importante mencionar que, según (Santos, 2020), la evolución de las plataformas IoT se ha dado gracias a la unión de tres clases de tecnologías: los microservicios de Internet, los sistemas microeléctricos y mecánicos, y la tecnología de comunicación inalámbrica.

Esta motivación permite establecer la idea de diseñar e implementar un módulo interactivo IoT con dispositivos low-cost, el mismo que consta de un Módulo WIFI ESP32 enlazado a una plataforma IoT (Ubidots) mediante protocolos de comunicación como TCP, UDP, HTTP y MQTT (la comunicación se realiza mediante este último) a los sensores necesarios para poder interactuar con el usuario, todo esto se encuentra debidamente programado mediante el IDE de Arduino. Esto permite desarrollar habilidades prácticas y aplicar la fundamentación teórica que se ha adquirido en la UPS.

## **1.4. Delimitación**

### **1.4.1. Delimitación Temporal**

El proyecto técnico se efectúa en un periodo académico a partir de la fecha de aprobación por los miembros de Consejo de la Carrera.

### **1.4.2. Delimitación Espacial**

El proyecto técnico se realiza en las instalaciones de la Universidad Politécnica Salesiana.

### **1.4.3. Delimitación Académica**

El proyecto implica en la elaboración de un módulo didáctico comenzando por el diseño, desarrollo e implementación tanto del firmware como el prototipo físico final, aplicando conocimientos de diversas materias como es la programación y diseño electrónico.

## **1.5. Objetivos**

En los siguientes apartados se mencionan el objetivo general y los objetivos específicos que se deben tener en cuenta para el desarrollo del proyecto.

### **1.5.1. Objetivo General**

Diseñar e implementar un módulo interactivo IoT usando dispositivos low-cost y servicios web.

### **1.5.2. Objetivos Específicos**

- Definir los protocolos TCP, UDP, HTTP y MQTT para la conexión del ESP32 a la plataforma Ubidots.
- Evaluar el desempeño del módulo programable con la plataforma Ubidots, además, de una aplicación básica realizando pruebas y mediciones en distintos escenarios.

- Configurar un sistema de alertas y notificaciones en tiempo real para que los usuarios reciban información relevante acerca del estado y funcionamiento de los dispositivos conectados al módulo ESP32.
- Implementar el control de entradas y salidas del ESP32 usando sensores, luces piloto y actuadores.

## 2. Fundamentos Teóricos

### 2.1. ESP32

El ESP32 es un microcontrolador de bajo costo y bajo consumo de energía, que se ha convertido en una opción popular para aplicaciones de Internet de las cosas (IoT). Cuenta con una amplia gama de características que lo hacen ideal para proyectos de IoT, como conectividad inalámbrica Wi-Fi y Bluetooth, y una amplia gama de interfaces, incluyendo SPI, I2C y UART. Además, el ESP32 cuenta con una amplia gama de opciones de memoria, lo que lo hace ideal para aplicaciones de gran tamaño.

Una de las principales características del ESP32 es su amplia conectividad inalámbrica, lo que lo hace ideal para proyectos de IoT que requieren la transferencia de datos en tiempo real. Además, cuenta con una amplia gama de opciones de memoria, lo que lo hace ideal para aplicaciones de gran tamaño. El ESP32 también cuenta con una amplia gama de interfaces, incluyendo SPI, I2C y UART, lo que permite una fácil integración con otros dispositivos. (Mendoza Mendoza & Quevedo Celi)

Según (Torres & Saltos, 2022) la estabilidad de conexión, así como la velocidad de procesamiento de datos del ESP32 supera a la versión del ESP8266:

**Tabla 1. Tabla comparativa entre ESP32 y el ESP8266**

Características	ESP32	ESP8266
CPU	Doble núcleo Xtensa LX6 a 240 MHz	Un solo núcleo Xtensa a 80 MHz
Wi-Fi	802.11 b/g/n	802.11 b/g/n
Bluetooth	Bluetooth 4.2 BR/EDR y BLE	No
RAM	520 KB	80 KB
Flash	4 MB	4 MB
Interfaces	SPI, I2C, UART, ADC, DAC	SPI, I2C, UART, ADC
Consumo de energía	Bajo	Bajo
Precio	Ligeramente más caro que el ESP8266	Más barato que el ESP32

## 2.2. Sensores Lowcost

Los sensores de bajo costo han revolucionado la manera en que se interactúa con el mundo que nos rodea. Estas pequeñas maravillas tecnológicas nos permiten medir y recopilar datos de manera accesible y asequible, abriendo un abanico de posibilidades en campos tan diversos como la ciencia, la industria, la agricultura y la vida cotidiana.

Según (Ramírez, 2014) el desarrollo de la electrónica se ha centrado en la disminución de tamaño y la optimización de valores, dando lugar a la formación de tecnologías cada vez más integradas. Esta tendencia implica la disminución considerable de sensores individuales que, trabajando juntos, cumplen funciones específicas. Los sensores low cost (bajo costo) ofrecen una solución económica para monitorear variables clave como temperatura, humedad, luz, movimiento, gas, distancia y mucho más.

A continuación, se explora cómo funcionan estos sensores, sus aplicaciones prácticas y cómo han democratizado el acceso a la recopilación de datos en una amplia gama de proyectos y aplicaciones.

**Sensor de Gas MQ-135:** El sensor de gas MQ-135 es un dispositivo que permite detectar gases nocivos en el aire, como dióxido de carbono, monóxido de carbono, amoníaco y compuestos orgánicos volátiles. Es compacto, de bajo costo y bajo consumo de energía, lo que lo hace ideal para la integración en sistemas de monitoreo ambiental y seguridad en interiores. (Soledispa Villegas, 2020)

**Figura 1. Sensor Mq-135**



La tabla 2 presenta los efectos de diferentes concentraciones de CO en el aire en relación con el tiempo de inhalación. Muestra cómo la concentración de CO afecta la salud y los efectos que se pueden experimentar en función del tiempo de exposición. (Soledispa Villegas, 2020)

**Tabla 2. Concentración de CO2 en el aire.**

Concentración de CO en el aire		Tiempo de inhalación y efectos
30 ppm	0,003%	Límite umbral de valor (máxima concentración respirable durante un lapso de 8 horas).
200 ppm 400 ppm	0.02 % 0.04 %	Sensación de malestar leve durante 2 o 3 horas. Cefalea en la región frontal que se manifiesta en 1 o 2 horas y se propaga a toda la cabeza.
800 ppm	0,08%	Sensación de vértigo, presencia de náuseas y temblores en las piernas después de 45 minutos. Desvanecimiento y pérdida de conocimiento en un plazo de 2 horas.
1600 ppm	0,16%	Cefalea, náuseas y mareos en 20 minutos. Muerte en 2 horas.
3200 ppm	0,32%	Dolor de cabeza intenso, sensación de náuseas y mareos dentro de los primeros 20 minutos. Fallecimiento ocurriendo en un lapso de 2 horas.
6400 ppm	0,64 %	Dolor de cabeza y vértigo manifestándose en 1 o 2 minutos. Fallecimiento ocurriendo en un período de 10 a 15 minutos.
12800 ppm	1,28%	Muerte en 1 o 3 minutos

**Sensor de ultrasónico HC-SR04:** El sensor de distancia HC-SR04 es un sensor que utiliza ondas ultrasónicas para medir distancias de manera precisa y confiable, Según (Huaman, 2022) trabaja en un rango de 2 a 450 cm. Su bajo consumo de energía y su costo accesible lo convierten en una opción adecuada para aplicaciones como estacionamientos inteligentes, sistemas de seguridad y navegación de robots.

**Figura 2. Sensor HC-SR04**



**Sensor LDR - LM393:** Un LDR con un fototransistor NPN basado en infrarrojos (IR), que es sensible a las fuentes de luz en el rango de 760 nm a 1100 nm,

análogas al calor emitido en situaciones de incendio. (N. Vikram, 2017) El módulo del sensor de luz se basa en una resistencia fotográfica y, por lo tanto, mide los datos de la intensidad de la luz ambiental. La salida estará en un nivel alto hasta que la intensidad de la luz exceda el umbral establecido utilizando el valor predeterminado para que el módulo pueda distinguir entre un entorno iluminado y uno oscuro.

**Figura 3. Sensor LDR LM393.**



**Sensor de Temperatura y Humedad DHT22:** El sensor DHT22 es un sensor que proporciona mediciones precisas de temperatura y humedad relativa en el entorno. Su bajo consumo de energía permite una operación prolongada sin agotar rápidamente las fuentes de energía. (Chango Tonato, 2022)

**Figura 4. Sensor DHT22**



**Sensor PIR o sensor HC- SR501:** Los dispositivos de detección de movimiento basados en sensores infrarrojos pasivos (PIR) cuentan con un componente esencial: un sensor piroeléctrico altamente sofisticado que tiene la capacidad de capturar y transformar la radiación infrarroja emitida por los cuerpos en movimiento en una señal eléctrica detectable. (Mendoza Mendoza & Quevedo Celi) Estos sensores infrarrojos pasivos, conocidos como PIR, se caracterizan por ser asequibles en términos económicos, de dimensiones compactas, eficientes en el consumo de energía y su operación resulta muy sencilla. Estas cualidades han propiciado su uso

extendido en una variedad de aplicaciones que van más allá de la simple detección de movimiento.

**Figura 5. Sensor PIR.**



### **2.3. Plataforma IoT: Ubidots**

Ubidots es una plataforma de Internet de las Cosas (IoT) que permite a las empresas y los usuarios finales recopilar, analizar y visualizar datos en tiempo real. Esta plataforma se ha vuelto cada vez más popular debido a sus características y beneficios únicos. En este ensayo, se analiza las principales características y beneficios de Ubidots.

**Figura 6. Plataforma Ubidots**



Una de las características clave de Ubidots es su facilidad de uso. La plataforma cuenta con una interfaz intuitiva y amigable que permite a los usuarios configurar y utilizar la plataforma con facilidad. Además, Ubidots proporciona una amplia variedad de bibliotecas y herramientas de desarrollo para que los usuarios puedan integrar fácilmente la plataforma con sus dispositivos IoT. (Espinosa Apolo & Orellana Guayas, 2021)

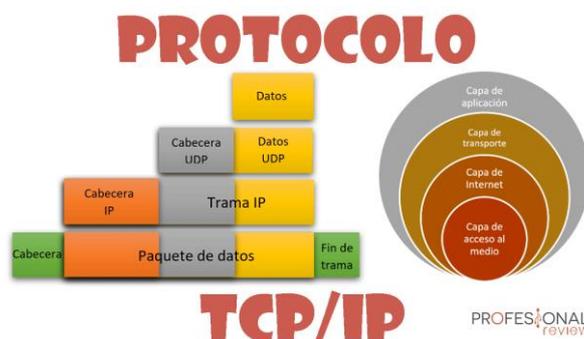
Otra característica importante de Ubidots es su capacidad para recopilar y almacenar datos en tiempo real. Esto permite a los usuarios monitorear y analizar los datos de sus dispositivos IoT en tiempo real, lo que les permite tomar decisiones más informadas y reaccionar rápidamente ante cualquier problema o situación de emergencia. Ubidots también ofrece una amplia gama de opciones de visualización de datos, lo que facilita la comprensión de los datos recopilados.

Además, Ubidots es altamente personalizable y escalable. Según (Hernandez Ruiz, 2022) La plataforma permite a los usuarios crear y personalizar sus propias aplicaciones IoT y dashboards de datos sin tener que crear una página web, lo que les permite adaptar la plataforma a sus necesidades específicas. También se puede escalar fácilmente para satisfacer las necesidades cambiantes de los usuarios a medida que crecen.

## 2.4. El modelo TCP/IP

El modelo TCP/IP, nació a finales de los sesenta como proyecto de defensa DARPA (cita), establece un estándar de comunicación en redes vigente hoy en día, unificando los protocolos IP y TCP para facilitar el intercambio de datos entre emisor-receptor de datos. Compuesto por cuatro capas fundamentales: la capa de Transporte asegura la comunicación estable a través de TCP y la rápida mediante UDP; la capa de Internet gestiona el direccionamiento y enrutamiento de datos usando el protocolo IP; la capa de Acceso a la Red regula la transmisión física con estándares como Ethernet; y la capa de Aplicación define protocolos para servicios de aplicaciones, posibilitando la comunicación y encapsulación de datos según el contexto. En sus inicios, Internet fue concebida por la DARPA para salvaguardar sistemas estratégicos y de información en medio del temor a ataques nucleares, contrastando con la actualidad donde representa una red global de computadoras y una revolución comunicativa sin precedentes en la historia. (Corona, 2004)

**Figura 7. Protocolo TCP/Ip.**



## 2.5. Protocolo TCP

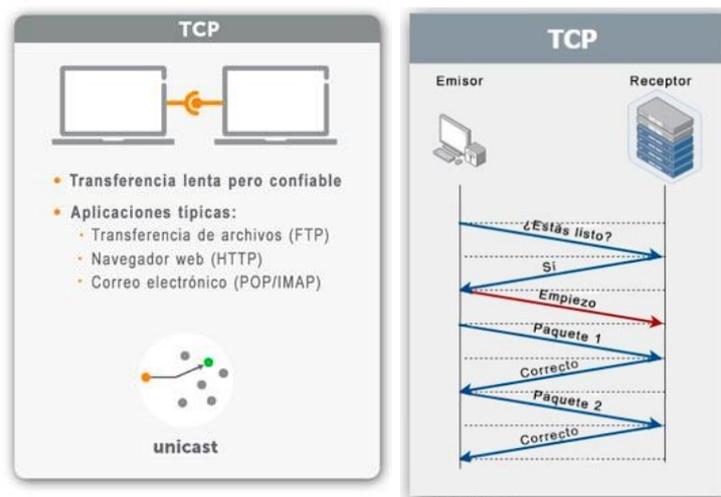
El protocolo TCP, que es el más utilizado en la navegación diaria. El TCP, o Protocolo de Control de Transmisión, se destaca por ser un protocolo de transporte

"orientado a la conexión". Esto significa que no solo se encarga de transmitir información entre dispositivos, sino que también verifica que la información se reciba correctamente. Es un protocolo diseñado para manejar conexiones de extremo a extremo. Funciona de manera complementaria con el protocolo IP, ya que los datagramas del protocolo IP no brindan un sistema de verificación mutua entre los dispositivos que intercambian información. (Castillo, 2020)

El protocolo IP, si se utilizara por sí solo, presentaría el riesgo de pérdida de datos durante la transferencia. Para evitar esto, el protocolo TCP establece una conexión entre el dispositivo emisor y el receptor, y verifica continuamente la emisión y recepción de la información entre ellos. El TCP logra esta verificación dividiendo el flujo de bytes en segmentos ordenados con un número de secuencia antes de transmitirlos a través del protocolo IP. El dispositivo receptor verifica este número de secuencia y, en caso de que falte algún segmento, el protocolo TCP solicita nuevamente su envío a través del protocolo IP hasta que el mensaje se reciba por completo en el dispositivo receptor. Gracias a su confiabilidad, el protocolo TCP brinda soporte a protocolos como HTTP, SMTP, SSH y FTP.

Analizando el protocolo TCP (Montecé Abril, 2023) afirma "TCP es un protocolo bidireccional confiable y orientado a la conexión" contribuyendo a la fiabilidad del protocolo.

**Figura 8. Protocolo TCP**



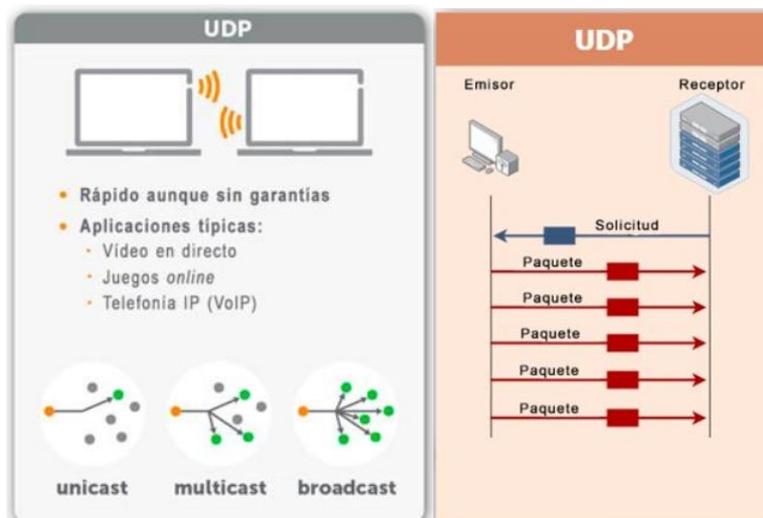
## 2.6. Protocolo UDP

El protocolo UDP o Protocolo de Datagrama de Usuario, tiene un funcionamiento similar al protocolo TCP, pero se diferencia en que no es un protocolo de transporte orientado a la conexión. A diferencia de TCP, UDP no verifica la recepción de los datos transmitidos entre dispositivos. En consecuencia, se sitúa en un nivel de capa inferior al protocolo TCP, lo que implica que el sistema de verificación de la recepción de los datos debe implementarse en las capas superiores del modelo. (Udp, n.d.)

La principal ventaja del protocolo UDP radica en su velocidad. Al no contar con un sistema de verificación bidireccional entre el emisor y el receptor, UDP permite una transferencia más rápida en comparación con TCP. Esta característica hace que UDP sea ampliamente utilizado en servicios de transmisión de voz o video en tiempo real, donde la velocidad de transmisión es crucial y se prioriza la fluidez de la transmisión sobre una posible pérdida puntual de datos.

No obstante, es importante tener en cuenta que la falta de verificación en UDP puede llevar a una mayor posibilidad de pérdida o corrupción de datos durante la transmisión. Esto implica que UDP puede no ser adecuado para aplicaciones que requieran una entrega de datos fiable y precisa. En cambio, se utiliza en situaciones donde la velocidad y la eficiencia son más prioritarias que la integridad de los datos, como en aplicaciones de transmisión de medios en tiempo real o juegos en línea.

**Figura 9. Protocolo UDP.**



## **2.7. Protocolo HTTP**

HTTP, conocido como "Hypertext Transfer Protocol" en inglés, es un protocolo utilizado para solicitar y transmitir datos y recursos en la web. Es la base fundamental de cualquier intercambio de información en Internet y sigue un modelo cliente-servidor, donde el cliente, generalmente a través de un navegador web, inicia la solicitud de datos al servidor. Esta comunicación se establece mediante una serie de mensajes entre el cliente y el servidor.

El protocolo HTTP permite obtener recursos como documentos HTML, imágenes, videos, scripts, hojas de estilo (CSS) y otros elementos necesarios para construir una página web completa. Estos recursos se unen para proporcionar una experiencia visualmente atractiva y funcional al usuario.

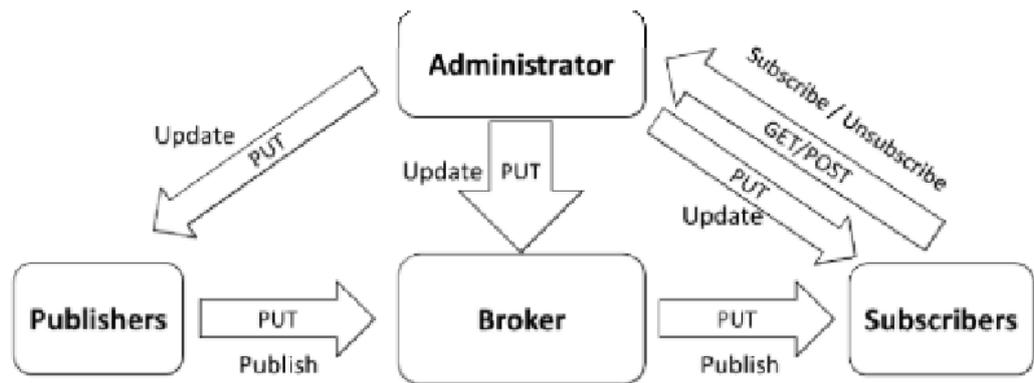
El enfoque de HTML presenta un diseño simple y eficiente al introducir un encabezado adicional denominado "State-Info" en todos los mensajes intercambiados entre el servidor y el cliente. Este encabezado está destinado a contener información relacionada con el estado y es bidireccional en su transmisión. Como resultado de esta innovación, se logra prescindir de la necesidad de que el servidor almacene información de estado en su memoria local cuando sea requerida. Cabe destacar que el contenido del encabezado "State-Info" es de naturaleza flexible y su significado es definido por las aplicaciones involucradas, adaptándose a sus necesidades y contexto específicos. (Luna del Águila)

## **2.8. Protocolo MQTT**

El protocolo MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería asíncrona y de bajo consumo diseñado para entornos de Internet de las Cosas (IoT). Fue desarrollado por IBM en la década de 1990 y se ha convertido en un estándar ampliamente adoptado en la comunicación entre dispositivos IoT.

MQTT utiliza un modelo de publicación/suscripción, donde los dispositivos se enlazan a un servidor MQTT llamado "broker" y pueden publicar variables en "temas" específicos. Otros dispositivos pueden suscribirse a esos temas y recibir los mensajes publicados en tiempo real. Esto permite una comunicación eficiente y flexible entre dispositivos, donde los mensajes se envían solo a aquellos que están interesados en ellos. (Mahedero Biot, 2020)

**Figura 10. Modelo publicación/suscripción**



Una de las principales ventajas de MQTT es su ligereza y eficiencia. Utiliza un protocolo de transporte de datos muy eficiente y tiene un tamaño de encabezado reducido, lo que minimiza la sobrecarga de la red y el consumo de energía de los dispositivos. Esto lo hace especialmente adecuado para entornos con recursos limitados, como dispositivos IoT con capacidad de procesamiento y ancho de banda limitados.

Es esencial destacar que es el cliente quien decide el nivel de QoS en todo momento. Además, conviene señalar que el nivel de Calidad de Servicio (QoS) está restringido a las comunicaciones entre el cliente y el servidor, así como entre el servidor y el cliente. Por lo tanto, cuando se publica un mensaje con nivel de QoS 1, esa calidad de servicio solo se aplica entre ese cliente y el servidor, sin tener ningún impacto entre otros clientes y servidores. (Tolia, 2018)

Otra característica importante de MQTT es su capacidad de garantizar la entrega confiable de mensajes mediante el uso de diferentes niveles de calidad de servicio (QoS). Esto permite adaptar la fiabilidad de la entrega según los requisitos de la

aplicación, desde una entrega al menos una vez (QoS 1) hasta una entrega exactamente una vez (QoS 2). (Guasch Llobera & Calleja Collado, 2019)

## 2.9. Comparativa entre protocolos de comunicación

Cuando se trata de aplicaciones de IoT, tanto TCP/IP, UDP, HTTP y MQTT son protocolos importantes con características y ventajas distintas.

El modelo TCP/IP es ampliamente utilizado en la comunicación en red y ofrece una comunicación confiable y orientada a la conexión. Es adecuado para aplicaciones que requieren una transferencia de datos segura y precisa, pero puede ser más pesado en términos de recursos y consumo de energía.

UDP, por otro lado, es un protocolo más ligero y rápido. No ofrece una garantía de entrega confiable, pero es eficiente en términos de ancho de banda y latencia. Es adecuado para aplicaciones en tiempo real donde la velocidad es prioritaria y se puede tolerar una posible pérdida ocasional de datos.

HTTP es el protocolo utilizado en la web y es muy útil para solicitar recursos y datos a través de una comunicación cliente-servidor. Es ampliamente compatible y fácil de implementar, pero puede tener una sobrecarga adicional debido a la naturaleza del protocolo.

Finalmente, MQTT es un protocolo diseñado específicamente para aplicaciones de IoT. Ofrece una comunicación ligera, eficiente y orientada a eventos utilizando un modelo de publicación/suscripción. MQTT es altamente escalable y eficiente en términos de consumo de recursos, lo que lo hace especialmente adecuado para aplicaciones de IoT con dispositivos de baja potencia y ancho de banda limitado. (Vienažindytė, 2019)

**Tabla 3. Tabla Comparativa entre TCP/UDP/HTTP/MQTT**

Protocolo	Orientado a la conexión	Garantía de entrega confiable	Eficiencia en el consumo de recursos	Comunicación en tiempo real	Escalabilidad
TCP/IP	x	X		X	x
UDP			x	x	x
HTTP	x	X			x
MQTT		Variable según QoS	x	x	x

En conclusión, el protocolo MQTT se destaca como una excelente opción para aplicaciones de IoT debido a sus numerosas ventajas. Según (Jimenez Canal, 2020) en comparación con los otros protocolos anteriormente mencionados que aportan recursos limitados de procesamiento, ancho de banda y energía en los dispositivos IoT el protocolo MQTT es mejor en las siguientes características:

**Eficiencia en el consumo de recursos:** MQTT es altamente eficiente en términos de consumo de recursos, lo que lo hace especialmente adecuado para dispositivos de IoT con limitaciones de potencia de procesamiento y ancho de banda. Su diseño ligero y la capacidad de enviar mensajes compactos hacen que sea ideal para entornos con recursos limitados.

**Comunicación en tiempo real:** MQTT ofrece una comunicación en tiempo real eficiente y de baja latencia. Esto es crucial en aplicaciones de IoT donde se requiere una respuesta rápida y una transmisión de datos casi instantánea. MQTT permite una transmisión ágil y oportuna de información, lo que resulta fundamental en casos como el monitoreo en tiempo real, la automatización y el control de dispositivos.

**Modelo de publicación/suscripción:** Se opta por la implementación de MQTT debido a su simplicidad y eficacia, que se derivan de su estructura de suscriptor/publicador. Además, es importante destacar que MQTT utiliza una cabecera de tan solo 2 bytes, lo que representa una considerable reducción en comparación con los 137 bytes requeridos por el protocolo HTTP (Jimenez Canal, 2020). Esto reduce la sobrecarga de red al enviar datos solo a los dispositivos interesados en ellos, lo que resulta en una mayor escalabilidad y eficiencia en comparación con los enfoques de solicitar/obtener utilizados en otros protocolos.

**Calidad de servicio ajustable:** MQTT ofrece diferentes niveles de calidad de servicio (QoS) que permiten adaptar la fiabilidad y la entrega de mensajes según los requisitos de la aplicación. Esto permite equilibrar la confiabilidad con el ancho de banda y la latencia, brindando flexibilidad en la gestión de la transmisión de datos.

## **2.10. Protocolos TCP, UDP y HTTP para la conexión del ESP32 a la plataforma Ubidots.**

Para los protocolos TCP, UDP y HTTP para enviar los valores de las variables se realiza el siguiente proceso. (Sepúlveda, 2023)

### **2.10.1. Envío de variables a la plataforma de Ubidots**

El proceso consiste en ingresar los datos de token, SSID y contraseña de WiFi en el ESP32. Se elige el protocolo de comunicación (HTTP, TCP o UDP) para conectarse a Ubidots. Luego, se definen las librerías necesarias y se establecen constantes clave. La comunicación con Ubidots se configura para diferentes protocolos. Después, se explica cómo enviar variables a la plataforma, asociándolas con etiquetas de dispositivo. El código verifica el envío exitoso y muestra un mensaje si se logra. Se introduce un retraso antes de repetir el proceso. (Sepúlveda, 2023) Consulta el anexo 1 para obtener el código completo.

### **2.10.2. Obtener variables desde Ubidots**

El proceso comienza la definición de las librerías necesarias en primer lugar. Luego, se establecen constantes clave como el token de Ubidots, SSID y contraseña de WiFi, junto con etiquetas de dispositivo y variable en segundo lugar. A continuación, se configura el protocolo de comunicación y se crea una instancia de Ubidots. Luego, se explica cómo obtener el último valor utilizando las etiquetas definidas previamente. Finalmente, se muestra cómo imprimir los valores obtenidos en la consola, con la condición de que el valor no sea un error. Se introduce un retraso antes de repetir el proceso. (Sepúlveda, 2023)

## **2.11. Protocolo MQTT para la conexión del ESP32 a la plataforma Ubidots.**

### **2.11.1. Publicar valores en una variable**

#### **2.11.2. Paso 1:TOKEN**

- Ingresa tu token de Ubidots, SSID y contraseña de WiFi.

#### **2.11.3. Paso 2: Definir las librerías**

```
#include "UbidotsEsp32Mqtt.h"
```

#### **2.11.4. Definir las instancias y constantes:**

Para que sea posible la comunicación se establece un token único para autenticar la comunicación entre el dispositivo y Ubidots. Luego se deben declarar constantes para ingresar las credenciales de red WiFi, permitiendo la conexión a Internet. Por último, Se definen etiquetas que identifican dispositivos y variables en la plataforma. Estas configuraciones precisas son esenciales para asegurar una conexión segura y organizada, estableciendo la base para la interacción efectiva y la creación de soluciones sólidas en el mundo del IoT. (Lopez, 2023)

#### **2.11.5. Paso 3: Envío de variables a la plataforma de Ubidots**

Para el envío de datos a Ubidots se usa el comando "ubidots.add" y luego se cierra la comunicación con el comando "ubidots.publish" (Lopez, 2023)

La estructura es la siguiente:

```
ubidots.add(nombrevariable, valor);  
ubidots.publish(etiqueta dispositivo);
```

### 3. Marco metodológico

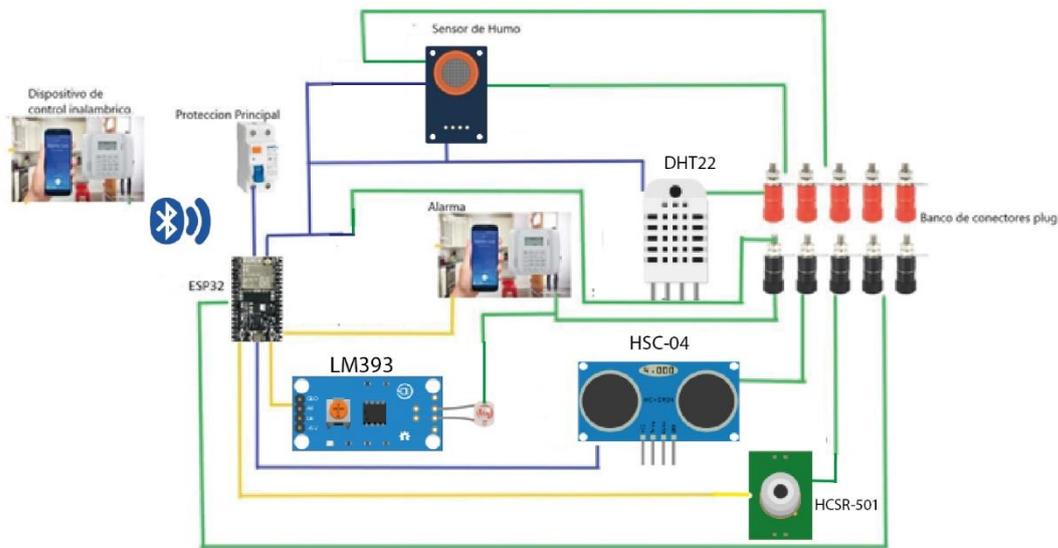
El enfoque de todo marco metodológico reside en describir el proceso que se emplea para cumplir con los objetivos establecidos. Desde la definición precisa de cada sentencia en la programación hasta la implementación de un sistema de alertas y notificaciones en tiempo real.

La idea es proporcionar una solución para facilitar el aprendizaje, con aplicaciones prácticas en diversos escenarios. A través de un análisis básico pero contundente y una implementación de cada componente, se pretende demostrar la viabilidad y utilidad de este módulo interactivo IoT.

#### 3.1. Propuesta de solución

Se parte de la idea original planteada en el anteproyecto para el desarrollo del marco metodológico. A continuación, en la figura 9 se puede observar la idea original planteada en el proyecto.

**Figura 11. Diseño proyecto.**



Fuente: Elaborado por el autor.

La metodología empleada en este proyecto es experimental y sigue un enfoque estructurado. Cada etapa se desglosa en las secciones siguientes:

Diseño del módulo

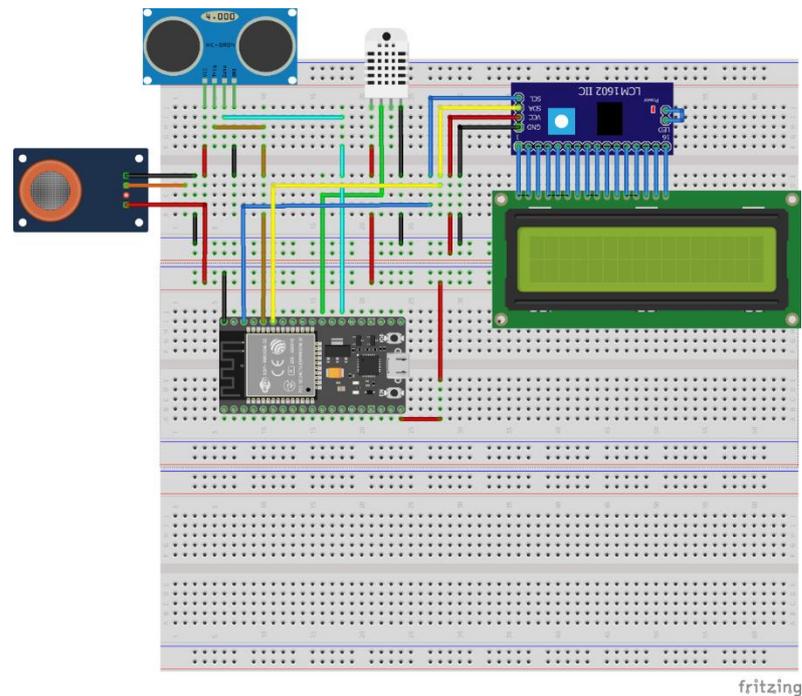
Programación del controlador

Configuración de la plataforma Ubidots

### 3.2. Diseño del módulo

En primer lugar, se desarrolla un prototipo de módulo interactivo utilizando el ESP32 como microcontrolador y tres sensores: DHT22 para medir temperatura y humedad, HCSR04 para la detección de distancias y el sensor MQ-135 para evaluar la calidad del aire. El diseño fue realizado mediante el software Fritzing, que permite visualizar y planificar el esquema de conexiones y componentes de manera intuitiva.

**Figura 12. Diseño Prototipo V1.**



En la siguiente tabla se detallan las conexiones del prototipo para el módulo interactivo.

**Tabla 4. Conexiones del Prototipo.**

Sensor	Conexión al ESP32
DHT22	Pin 17 (GPIO17)
Sensor ultrasónico	Trigger: Pin 3 (GPIO3)
Sensor ultrasónico	Echo: Pin 4 (GPIO4)
Sensor MQ-135	Pin 3 (GPIO3)
LCD	I2C: SCL -> Pin 22 (GPIO22)
LCD	I2C: SDA -> Pin 21 (GPIO21)

Fuente: Elaborado por el autor.

### **3.3. Programación (IDE Arduino)**

El desarrollo del código se realiza en el IDE de Arduino y se divide en las siguientes partes:

- Paso 1: Cabecera del código.
- Paso 2: Declaración de variables del sistema.
- Paso 3: Función SETUP.
- Paso 4: Función Loop.

### **3.4. Paso 1: Cabecera del Código.**

Se definen las librerías y variables para cada componente que se va a usar:

Se incluyen múltiples librerías para habilitar diferentes funcionalidades. La librería "Wire.h" permite la comunicación I2C, utilizada para la interacción con dispositivos como el sensor DHT22 de temperatura y humedad, controlado mediante la librería "DHTesp.h". También se emplea la librería "LiquidCrystal\_I2C.h" para controlar un LCD mediante I2C. Además, se incorpora la librería "WiFi.h" para conectar el ESP32 a una red local a través de Wi-Fi, y "UbidotsEsp32Mqtt.h" para integrar con el servicio Ubidots mediante MQTT. Se definen diversos pines GPIO para interactuar con componentes como un sensor ultrasónico, un LED y un sensor de gas MQ-135. El código configura el LCD con una dirección I2C específica y dimensiones de 20x4 para mostrar información.

### **3.5. Paso 2: Variables para enviar de datos a Ubidots**

En esta parte del código se definen las variables como "TOKEN\_UBIDOTS" para autenticación, "NOMBRE\_RED\_WIFI" y "CONTRASEÑA\_WIFI" para conectividad Wifi, y etiquetas de dispositivo y variables como "ETIQUETA\_DISPOSITIVO", " " y "ETIQUETA\_DISTANCIA", se establecen las bases para el intercambio de datos. La variable "FRECUENCIA\_PUBLICACION" define la frecuencia de actualización, mientras que "temporizador" monitorea los intervalos. Este código es esencial para establecer una comunicación segura y precisa entre el dispositivo y Ubidots, facilitando la implementación efectiva de soluciones IoT.

### **3.6. Paso 3: Función Setup**

En la función `setup()`, se realizan configuraciones esenciales para el funcionamiento del dispositivo en un entorno IoT. Se inicia la comunicación serial y se inicializa una pantalla LCD. Se conecta al sensor DHT22 para medir temperatura y humedad, se establece la conexión WiFi, se configura la plataforma Ubidots y se define el modo de varios pines, incluyendo el sensor ultrasónico de distancia y LED indicadores.

### **3.7. Paso 4: Función Loop**

La función `loop()` es el bucle principal, se encarga de la recolección y transmisión continua de datos. Tras asegurar la conectividad con Ubidots y reconectar en caso necesario, esta función emplea un temporizador y frecuencia de publicación para adquirir datos de múltiples sensores, incluyendo el sensor DHT22 para temperatura y humedad, el MQ135 para gas, el sensor ultrasónico para distancia, y añade mediciones de un sensor PIR para movimiento y un LDR para niveles de luz. Los resultados son exhibidos en pantalla y puerto serial, y luego agregados a variables predefinidas para ser publicados en Ubidots, actualizando el temporizador para el próximo ciclo.

### **3.8. Configuración Ubidots**

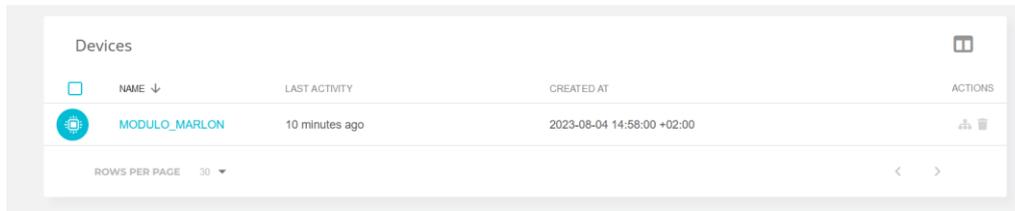
La configuración en Ubidots se debe realizar siguiendo los siguientes pasos:

- Crear un nuevo dispositivo
- Crear variables
- Enlazar las variables con el dashboard
- Añadir los widgets de métrica y control.

#### **3.8.1. Paso 1: Crear un nuevo dispositivo**

La configuración comienza creando un dispositivo en blanco, luego se debe asignar un nombre y etiqueta para el dispositivo.

**Figura 13. Dispositivos Ubidots.**

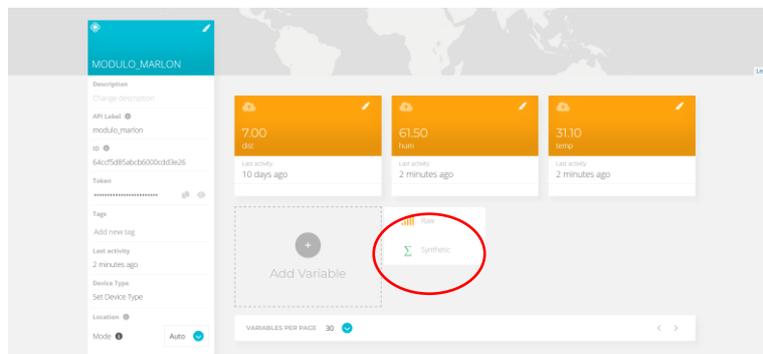


Fuente: Obtenido de Ubidots.

### 3.8.2. Paso 2: Crear variables

La creación de variables puede ser desde la plataforma web como desde la codificación de la placa ESP32. Para crear las variables desde Ubidots se debe presionar el botón add variable, luego se presiona Raw y se cambia el nombre de la variable.

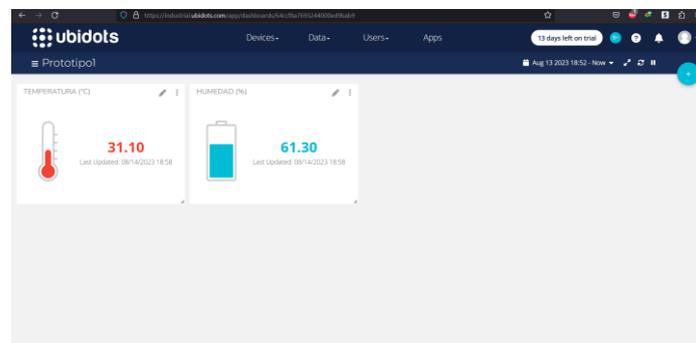
**Figura 14. Variables Ubidots**



### 3.8.3. Paso 3: Enlazar las variables con el Dashboard

Para crear un dashboard o interfaz de usuario se debe ir a “DATA” → “Dashboard” y luego se da clic al botón “+” tal como se ve la figura 12.

**Figura 15. Interfaz gráfica Ubidots**



Fuente: Obtenida de Ubidots.

### 3.8.3.1. Widgets

Para visualizar las variables de forma más intuitiva para el usuario final se debe usar los widgets

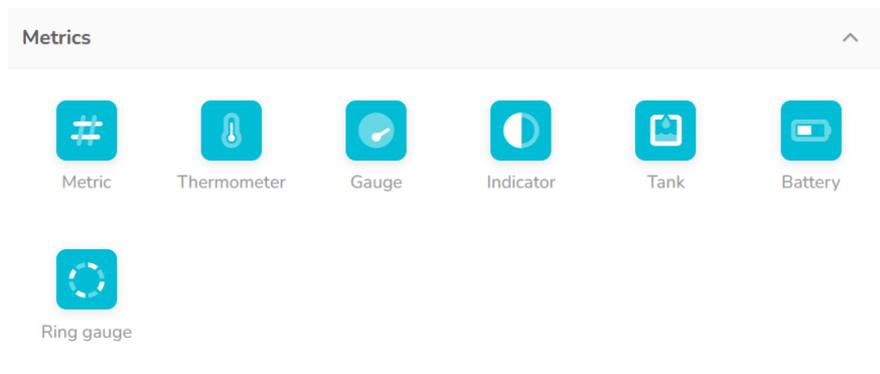
En este caso, se usan widgets Metrics que pueden ser para variables numéricas y booleanas.

Para el sensor de distancia se usa Metric, un widget que se muestra como un número que cambia según el dato recibido.

Para el sensor de temperatura se usa Thermometer, que muestra la variable con una representación visual de un termómetro.

Para el sensor de humedad, se usa Battery, que muestra el porcentaje de nivel simulando una batería.

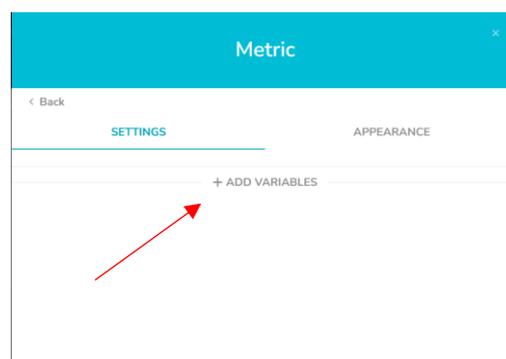
**Figura 16. Widgets Metrics Ubidots.**



Fuente: Obtenido de Ubidots.

Ahora se debe seleccionar Metric para enlazar la variable “dist” con el widget.

**Figura 17. Metric**



Se debe seleccionar una variable para poder visualizarla en la dashboard.

Por último, debe salir el widget con el valor de la variable tal como se ve en la figura 18.

**Figura 18. Widget metric.**



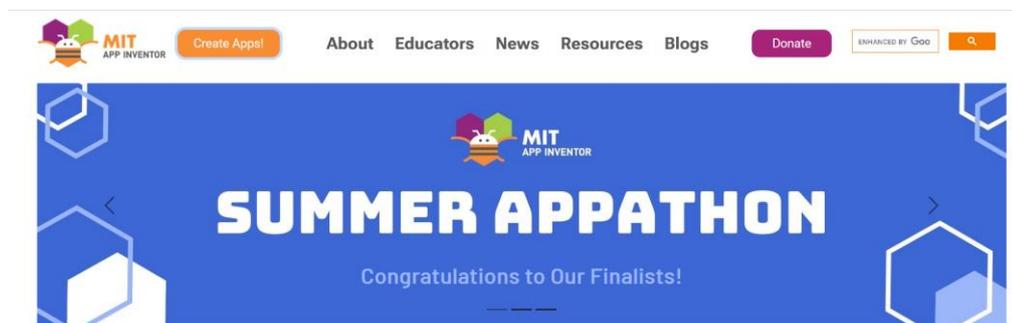
Fuente: Obtenido de Ubidots.

### **3.9. App Inventor**

La plataforma APP Inventor es la forma más práctica y versátil para desarrollar una aplicación enfocada al internet de las cosas (IoT).

Entonces, para crear la aplicación se debe entrar a la web: <https://appinventor.mit.edu/> donde se presiona el botón "Create Apps!".

**Figura 19. Plataforma Web APPinventor.**



Fuente: Obtenido de appinventor.com

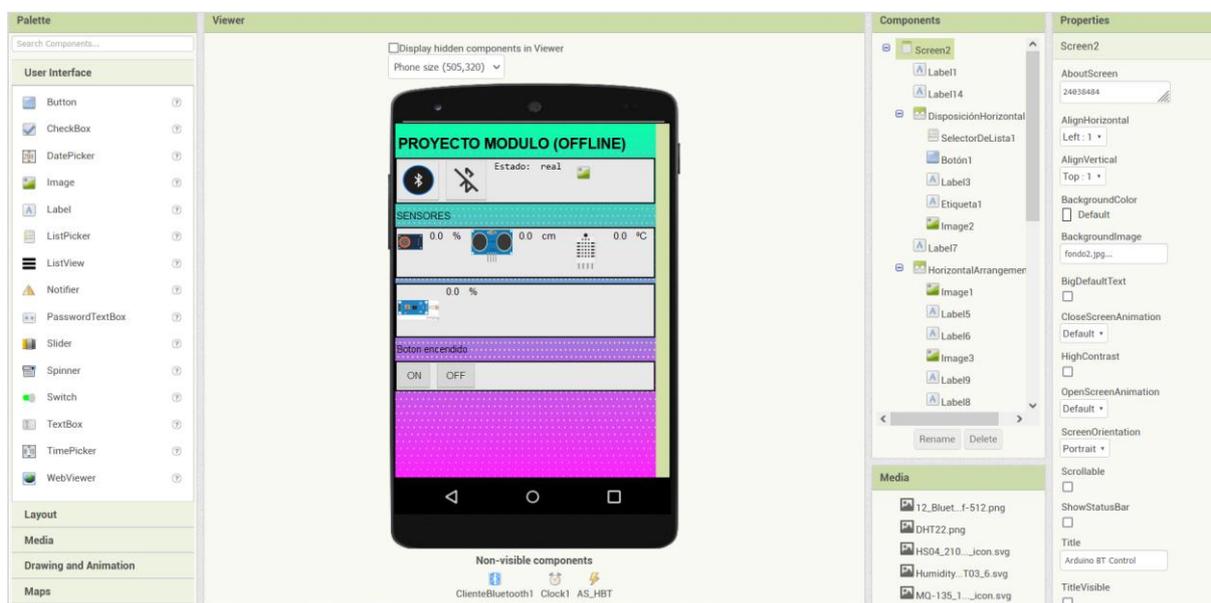
#### **3.9.1. Interfaz de usuario**

Al presionar el botón aparece la pantalla principal de desarrollo de la aplicación en donde se muestra varias herramientas para realizar el diseño de y programación de la aplicación.

En la figura 20 podemos observar las siguientes herramientas:

- **Palette:** Muestra una serie de elementos para poder añadir al área de trabajo.
- **Viewer:** Se puede visualizar los diferentes elementos que se usan para el desarrollo de la aplicación.
- **Components:** Brinda un despliegue de elementos que representan los enlaces con cada componente de la aplicación.
- **Properties:** Permite cambiar las propiedades de cada componente de la aplicación.

**Figura 20. Pantalla de desarrollo APPinventor.**



Fuente: Obtenido de Appinventor.

### 3.9.2. Interfaz de usuario final de la app

En la figura 21 se muestra el diseño de la aplicación desarrollada en Appinventor.

**Figura 21. Interfaz APP offline**



### **3.10. Implementación del módulo.**

La implementación física del módulo interactivo se llevó a cabo mediante una caja de PVC, la cual proporciona una base sólida y segura para alojar los diferentes componentes y permitir su funcionamiento eficiente.

A continuación, se describe el proceso de implementación en detalle:

#### **3.10.1. Selección y Diseño de la Caja**

Se eligió una caja de PVC de dimensiones 40x30x17 centímetros para albergar los componentes del módulo. La elección del material y tamaño proporciona durabilidad y suficiente espacio para la instalación ordenada de los elementos.



### 3.10.2. Preparación de los Huecos y Espacios

Se realizaron perforaciones y cortes precisos en la caja para acomodar cada componente de manera organizada. Se crearon huecos para la pantalla LCD y el selector de dos posiciones, asegurando un ajuste adecuado.



### 3.10.3. Montaje de la Pantalla LCD

La pantalla LCD se instaló en uno de los huecos diseñados previamente. Esto permite la visualización de información relevante sobre el estado y funcionamiento del módulo y sus componentes.



### 3.10.4. Instalación del Selector de Dos Posiciones

El selector de dos posiciones se encajó en su respectivo hueco. Este componente controla el encendido y apagado de varios elementos dentro del módulo, proporcionando una forma intuitiva de control.



### 3.10.5. Montaje de la Luz Piloto (Voltímetro)

La luz piloto, con apariencia de voltímetro, se ubicó en una posición visible y de fácil acceso. Esta luz piloto actúa como indicador visual del estado de funcionamiento del módulo.



### 3.10.6. Integración del Botón de Paro de Emergencia

El botón de paro de emergencia se conectó directamente a la fuente de alimentación, lo que permite una desconexión rápida y segura en caso de situaciones críticas.



### 3.10.7. Incorporación del Plafón y Fuente AC

El plafón se montó estratégicamente para proporcionar una iluminación adecuada en el área circundante. La fuente de alimentación AC 110V se conectó a un breaker, asegurando un suministro eléctrico confiable y seguro para el módulo.



### 3.10.8. Conexión del Selector al Ventilador

El selector de dos posiciones se conectó al ventilador, permitiendo controlar su funcionamiento. Esto garantiza que el módulo pueda regular la temperatura y el ambiente interno según sea necesario.



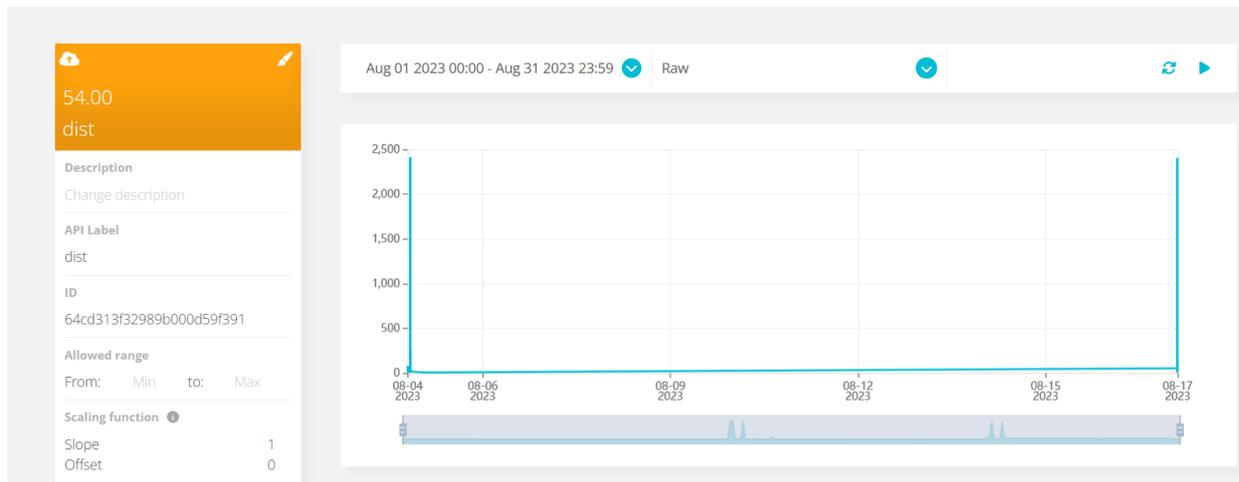
3.10.9. Prueba y conexión



## 4. Resultados

La variable "dist", que representa la distancia medida por el sensor ultrasónico del módulo interactivo en tiempo real, En la gráfica 1 podemos observar cómo esta variable evoluciona a lo largo del tiempo, es decir, esta gráfica captura los cambios en la distancia con respecto al tiempo, lo que permite identificar patrones y tendencias de manera intuitiva. En el eje x se encuentra el tiempo, mientras que en el eje y se representa el valor de la distancia. Esta representación gráfica simplifica la interpretación de los datos y proporciona información en tiempo real para registrar y tomar decisiones en base a estos datos.

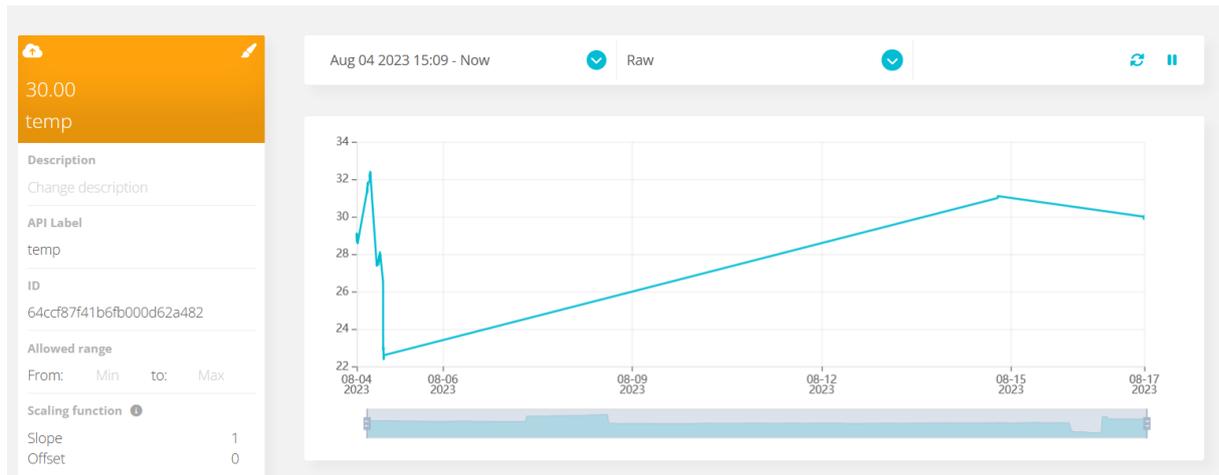
**Gráfico 1. Historial del sensor de distancia.**



El monitoreo de la variable "dist" cabe resaltar en aplicaciones de seguridad, ya que un cambio brusco en la distancia podría indicar la presencia de obstáculos no previstos o potenciales situaciones de peligro. En el ámbito industrial, el seguimiento de la distancia puede indicar el nivel de llenado de contenedores o detectar posibles obstrucciones en líneas de producción. Por otro lado, en sistemas de transporte autónomo, la medición precisa de la distancia es esencial para evitar colisiones y mantener la seguridad.

En el caso de la variable "temp", que refleja la temperatura capturada por el sensor DHT22 en del proyecto, ofrece una visualización dinámica y en tiempo real de cómo varía la temperatura a lo largo del tiempo. El gráfico 2 muestra de manera intuitiva la relación entre el tiempo y la temperatura, con el eje horizontal representando el tiempo y el eje vertical indicando los valores numéricos de la temperatura.

**Gráfico 2. Historial variable temperatura.**



Fuente: Obtenido de Ubidots.com.

El monitoreo de la variable "temp" es de gran importancia debido a su relevancia en entornos científicos, una medición precisa de la temperatura puede arrojar luz sobre procesos termodinámicos. En la industria, como en el sector farmacéutico y alimentario, mantener un seguimiento estricto de la temperatura garantiza la calidad y seguridad de los productos. Además, en aplicaciones de automatización y gestión de edificios, el monitoreo de la temperatura puede contribuir a una mayor eficiencia energética.



## CRONOGRAMA

N e		Cronograma																											
		ACTIVIDADES				MES1				MES2				MES3				MES4				MES5				MES6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1	Elegir el tema																												
2	Elaboración del anteproyecto																												
3	Revisión bibliográfica sobre ESP32																												
4	Revisión bibliográfica sobre plataformas IoT y sensores																												
5	Documentación Protocolos de comunicación																												
6	Selección y adquisición de los componentes necesarios																												
7	Desarrollar la programación del Módulo Interactivo																												
8	Desarrollar la integración con la plataforma IoT																												
9	Implementar el módulo y configurar sistema de alertas y notificaciones																												
10	Evaluar el desempeño del módulo con la plataforma Ubidots																												
11	Corregir errores pendientes según datos obtenidos																												
12	Documentar cada paso detalladamente en el borrador																												
13	Documentación y presentación del trabajo																												

## PRESUPUESTO

No	Descripción	Cantidad	Unidad	Costo Unitario (\$)	Costo Total (\$)
<b>RECURSOS MATERIALES</b>					
1	Módulo NodeMCU Bluetooth/WiFi ESP32 WROOM32	1	U	\$24,50	\$24,50
2	Sensor de temperatura y Humedad	1	U	\$3,60	\$3,60
3	Modulo Buzzer	1	U	\$1,85	\$1,85
4	Sensor LDR Fotorresistencia LM393	1	U	\$2,50	\$2,50
5	Sensor de calidad de aire y humo	1	U	\$3,50	\$3,50
6	Breaker de riel un polo 110V / 6A	1	U	\$5,50	\$5,50
7	Pantalla LCD 20x4 con módulo I2C	1	U	\$18,50	\$18,50
8	Fuente de alimentación de 5V y 3.3 V	1	U	\$3,50	\$3,50
9	Base ESP32 tipo SHIELD con borneras	1	U	\$15	\$15
10	Cautín 30 W	1	U	\$8,00	\$8,00
11	Sensor distancia	1	U	\$5,00	\$5,00
12	Sensor de movimiento	1	U	\$12,00	\$12,00
13	Selector de 2 posiciones	1	U	\$4,50	\$4,50
14	Estaño	1	U	\$3,60	\$3,60
15	Cable flexible #22	2	U	\$3,75	\$7,50
16	Cable flexible #26	4	U	\$2,37	\$9,50
17	Pulsador tipo hongo paro de emergencia	1	U	\$4,90	\$4,90
18	Gabinete plástico PVC 40x30x17	1	U	\$45,40	\$45,40
19	Conectores tipo borneras x 5	5		\$0,70	\$3,50
20	Plug Banana macho y hembra	12	U	\$2,37	\$28,44
21	Indicador piloto de voltaje y amperaje	1	U	\$9,50	\$9,50
				Sub-Total	\$256,85
				IVA 12%	\$30,82
				TOTAL	\$287,67

## Conclusiones

En base a la investigación y prueba de los protocolos definidos TCP, UDP, HTTP y MQTT, y luego de analizar la comunicación entre el ESP32 y la plataforma Ubidots. Se procede a desarrollar una tabla comparativa, en la cual se validó que el protocolo MQTT ofrece un rendimiento superior para aplicaciones IoT en términos de ancho de banda y latencia en comparación con otros protocolos. Esta conexión sólida y eficaz sentó las bases fundamentales para la interacción sin problemas de los dispositivos IoT con el entorno web, permitiendo una transferencia fluida y confiable de datos.

El desempeño del módulo interactivo en una amplia variedad de condiciones y la realización de pruebas en múltiples escenarios confirmaron su correcto funcionamiento y versatilidad en la adquisición de datos. A través de datos técnicos que demostraron la precisión y la estabilidad en la recopilación de información, queda comprobado que el módulo está acorde con los lineamientos presentados por diferentes aplicaciones.

La implementación exitosa del sistema de alertas y notificaciones en tiempo real mejora la experiencia del usuario al permitir mantenerse informados sobre las condiciones de los dispositivos conectados al módulo ESP32. Los datos técnicos demostraron que las notificaciones se entregaron con una baja latencia lo que mejoró la capacidad de respuesta y la toma de decisiones en función de los datos recopilados en tiempo real. Esto aseguró que los usuarios estuvieran al tanto de los cambios en los dispositivos, lo que a su vez mejoró la eficiencia y la eficacia de las operaciones.

La implementación del control de entradas y salidas, junto con la interacción con sensores y actuadores, validó la capacidad del módulo IoT para manipular dispositivos físicos de manera efectiva. La manipulación del módulo refleja la precisión y coherencia del control ejercido sobre los dispositivos físicos, lo que demostró que el módulo no solo adquiere datos, sino que también es capaz de cambiar aspectos físicos en tiempo real.

## **Recomendaciones**

Para la codificación se debe consultar siempre el datasheet de la placa de desarrollo ESP32 ya que no todos los pines GPIO se pueden usar para los sensores analógicos.

En cuanto a Ubidots, se debe tener en cuenta que para una interacción prolongada con esta plataforma hay que adquirir una suscripción de los planes de pago para uso indefinido.

Para la implementación física se debe considerar que el cableado tiene que estar bien ajustado para evitar problemas de conexión o fallo eléctrico.

## Bibliografía

- Caballar, J. A. (s.f.). *Qué es TCP y UDP*. Obtenido de <https://carballar.com/que-es-tcp-y-udp>
- Castillo, J. A. (21 de 03 de 2020). *Protocolo TCP/IP – Qué es y cómo funciona* . Obtenido de <https://www.profesionalreview.com/2020/03/21/protocolo-tcp-ip/>
- Chango Tonato, J. G. (03 de 2022). *Análisis comparativo de la privacidad de los datos exclusivamente en la transmisión hacia las plataformas de IoT más utilizadas*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/22172>
- contributors, M. (26 de 11 de 2022 ). *Generalidades del protocolo HTTP*. Obtenido de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- Corona, A. E. (10 de 09 de 2004). Protocolos TCP/IP de internet. pág. 7. Obtenido de [https://www.ru.tic.unam.mx/bitstream/handle/123456789/791/sep\\_art51.pdf?sequence=](https://www.ru.tic.unam.mx/bitstream/handle/123456789/791/sep_art51.pdf?sequence=)  
=
- Espinosa Apolo, B. A., & Orellana Guayas, M. E. (02 de 2021). *Desarrollo de aplicaciones de monitoreo y control basadas en IoT a través de la plataforma Ubidots. Aplicaciones a sistemas de automatización bajo entornos de simulación*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/20298>
- Guasch Llobera, J., & Calleja Collado, M. (09 de 05 de 2019). *Monitorización de sensores con arduino utilizando el protocolo MQTT*. Obtenido de <https://upcommons.upc.edu/handle/2117/134193>
- Hernandez Ruiz, J. A. (2022). *Sistema de comunicación y dashboard de resultados para un arreglo de dispositivos que detectan fallas en paneles solares con cámara RGB y redes convolucionales*. Obtenido de <http://hdl.handle.net/1992/55588>
- Huaman, D. P. (2022). Implementación de un prototipo de monitoreo del nivel del agua en el reservorio de Viñas de la ciudad de Pampas de la provincia de Tayacaja. *Llamkasun*, 3(1), 104-111., 105-106. doi:<https://doi.org/10.47797/llamkasun.v3i1.89>
- Jimenez Canal, V. R. (16 de 12 de 2020). *Evaluación y diseño de una Plataforma IoT con soporte de protocolo MQTT para un entorno de red empresarial*. Obtenido de <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/17707>
- Lopez, I. (01 de 07 de 2023). *Connect an ESP32-DevKitC to Ubidots over MQTT*. Obtenido de <https://help.ubidots.com/en/articles/748067-connect-an-esp32-devkitc-to-ubidots-over-mqtt>
- Luna del Águila, F. &. (s.f.). *Implementación del Protocolo HTTP Paralelizado en Cliente y Servidor*. Obtenido de <http://hdl.handle.net/11285/628362>
- Mahedero Biot, F. (2 de 6 de 2020). *Desarrollo de una aplicación IoT para el envío de imágenes mediante el protocolo MQTT (Doctoral dissertation)*. Obtenido de <https://riunet.upv.es/handle/10251/152408>

- Mendoza Mendoza, J. E., & Quevedo Celi, R. A. (s.f.). *Diseño e Implementación de un banco de pruebas empleando el NodeMCU (ESP8266) para aplicaciones domóticas con IoT y APPLE HOMEKIT*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/24050>
- Montecé Abril, F. A. (2023). *Diseño e implementación de un sistema SCADA para planta de nano filtración de ups controlado por dispositivo móvil*. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/24126>
- N. Vikram, K. S. (2017). A Low Cost Home Automation System Using Wi-Fi Based Wireless Sensor Network Incorporating Internet of Things (IoT). *IEEE 7th International Advance Computing Conference (IACC)*. doi: 10.1109/IACC.2017.0048.
- neo.lcc.uma.es. (s.f.). *El protocolo UDP*. Obtenido de <https://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/udp.html>
- Ramírez, L. G. (2014). *Sensores y actuadores*. Patria.
- Santos, P. R. (22 de 09 de 2020). *Telefónica Tech*. Obtenido de <https://empresas.blogthinkbig.com/>
- Sepúlveda, D. (01 de 07 de 2023). *Connect your ESP32 to Ubidots over HTTP, TCP or UDP*. Obtenido de <https://help.ubidots.com/en/articles/4855281-connect-your-esp32-to-ubidots-over-http-tcp-or-udp>
- Soledispa Villegas, J. T. (21 de 10 de 2020). *Análisis de niveles muy altos de CO2 en las aulas de clases mediante el sensor MQ-135*. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/58211>
- Team, K. (30 de 03 de 2023). *¿Qué es el protocolo HTTP?* . Obtenido de <https://keepcoding.io/blog/que-es-el-protocolo-http/>
- Tolia, M. (17 de 09 de 2018). *MQTT QoS Levels (Explained with Examples & When to use Them)*. Obtenido de <https://mntolia.com/mqtt-qos-levels-explained/>
- Torres, A. P., & Saltos, H. X. (20 de 07 de 2022). *Desarrollo de un banco de pruebas utilizando el chip ESP-32 y el asistente virtual Alexa para el laboratorio de domótica*. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/24026>
- Vienažindytė, I. (13 de 11 de 2019). Obtenido de <https://nordvpn.com/es/blog/protocolo-tcp-udp/>

# ANEXOS



炜盛科技 Zhengzhou Winsen Electronics Technology Co., Ltd

[www.winsen-sensor.com](http://www.winsen-sensor.com)

## MQ135 Semiconductor Sensor for Air Quality

### Profile

Sensitive material of MQ135 gas sensor is  $\text{SnO}_2$ .

which with lower conductivity in clean air. When target pollution gas exists, the sensor's conductivity gets higher along with the gas concentration rising. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit.

MQ135 gas sensor has high sensitivity to ammonia gas, sulfide, benzene series steam, also can monitor smoke and other toxic gases well. It can detect kinds of toxic gases and is a kind of low-cost sensor for kinds of applications.



### Features

It has good sensitivity to toxic gas in wide range, and has advantages such as long lifespan, low cost and simple drive circuit & etc.

### Main Applications

It is widely used in domestic gas alarm, industrial gas alarm and portable gas detector.

### Technical Parameters Stable.1

Model		MQ135	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite, Metal cap	
Target Gas		ammonia gas, sulfide, benzene series steam	
Detection range		10 - 1000ppm( ammonia gas, toluene, hydrogen, smoke)	
Standard Circuit Conditions	Loop Voltage	$V_L$	$\leq 24V$ DC
	Heater Voltage	$V_H$	5.0V±0.1V AC or DC
	Load Resistance	$R_L$	Adjustable
Sensor character under standard test conditions	Heater Resistance	$R_H$	29Ω±3Ω (room tem.)
	Heater consumption	$P_H$	$\leq 950mW$
	Sensitivity	S	$R_S(\text{in air})/R_S(\text{in } 400\text{ppm H}_2) \geq 5$
	Output Voltage	$V_S$	2.0V - 4.0V (in 400ppm H <sub>2</sub> )
Standard test conditions	Concentration Slope	$\alpha$	$\leq 0.4 (R_{100\text{ppm}}/R_{10\text{ppm}} \text{ H}_2)$
	Tem. Humidity	20°C±2°C; 55%±5%RH	
	Standard test circuit	$V_L: 5.0V \pm 0.1V$ $V_H: 5.0V \pm 0.1V$	
Preheat time		Over 48 hours	

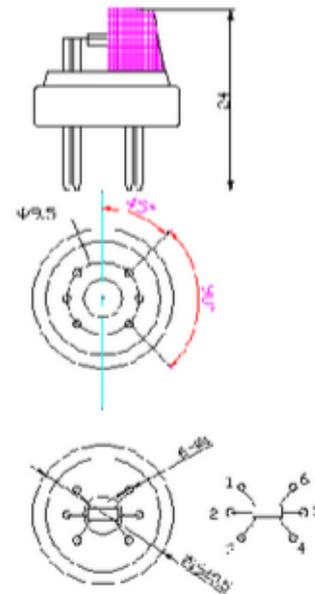


Fig1. Sensor Structure

Unit: mm

NOTE: Output voltage ( $V_S$ ) is  $V_{RH}$  in test environment.

## Ultrasonic Ranging Module HC - SR04

### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

### Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

## LMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators

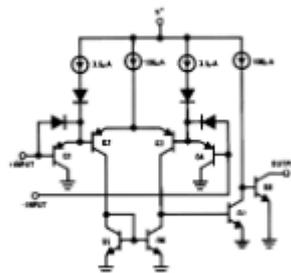
### 1 Features

- Wide Supply
  - Voltage Range: 2.0 V to 36 V
  - Single or Dual Supplies:  $\pm 1.0$  V to  $\pm 18$  V
- Very Low Supply Current Drain (0.4 mA) — Independent of Supply Voltage
- Low Input Biasing Current: 25 nA
- Low Input Offset Current:  $\pm 5$  nA
- Maximum Offset voltage:  $\pm 3$  mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Low Output Saturation Voltage: 250 mV at 4 mA
- Output Voltage Compatible with TTL, DTL, ECL, MOS and CMOS logic systems
- Available in the 8-Bump (12 mil) DSBGA Package
- See AN-1112 (SNVA009) for DSBGA Considerations
- Advantages
  - High Precision Comparators
  - Reduced  $V_{OS}$  Drift Over Temperature
  - Eliminates Need for Dual Supplies
  - Allows Sensing Near Ground
  - Compatible with All Forms of Logic
  - Power Drain Suitable for Battery Operation

### 2 Applications

- Battery powered applications
- Industrial applications

### 4 Simplified Schematic



### 3 Description

The LM193-N series consists of two independent precision voltage comparators with an offset voltage specification as low as 2.0 mV max for two comparators which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multivibrators and high voltage digital logic gates. The LM193-N series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM19-N series will directly interface with MOS logic where their low power drain is a distinct advantage over standard comparators.

The LM393 and LM2903 parts are available in TI's innovative thin DSBGA package with 8 (12 mil) large bumps.

Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM193-N	TO-99 (8)	9.08 mm x 9.08 mm
LM293-N		
LM393-N	SOIC (8)	4.90 mm x 3.91 mm
LM2903-N		

(1) For all available packages, see the orderable addendum at the end of the datasheet.

# Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

## 1. Feature & Application:

- \* Full range temperature compensated
- \* Relative humidity and temperature measurement
- \* Calibrated digital signal
- \* Outstanding long-term stability
- \* Extra components not needed
- \* Long transmission distance
- \* Low power consumption
- \* 4 pins packaged and fully interchangeable

## 2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

## 3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

## 4. Dimensions: (unit----mm)

### 1) Small size dimensions: (unit----mm)

2

Thomas Liu (Business Manager)

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn)

# HC-SR501 PIR MOTION DETECTOR

## Product Description

HC-SR501 is based on infrared technology, automatic control module, using Germany imported LHI778 probe design, high sensitivity, high reliability, ultra-low-voltage operating mode, widely used in various auto-sensing electrical equipment, especially for battery-powered automatic controlled products.

### Specification:

- Voltage: 5V ~ 20V
- Power Consumption: 65mA
- TTL output: 3.3V, 0V
- **Delay time: Adjustable (3 ~ 5min)**
- Lock time: 0.2 sec
- Trigger methods: L – disable repeat trigger, H enable repeat trigger
- Sensing range: less than 120 degree, within 7 meters
- Temperature: -15 ~ +70
- Dimension: 32\*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

### Application:

Automatically sensing light for Floor, bathroom, basement, porch, warehouse, Garage, etc, ventilator, alarm, etc.

### Features:

- Automatic induction: to enter the sensing range of the output is high, the person leaves the sensing range of the automatic delay off high, output low.
- Photosensitive control (optional, not factory-set) can be set photosensitive control, day or light intensity without induction.
- Temperature compensation (optional, factory reset): In the summer when the ambient temperature rises to 30 ° C to 32 ° C, the detection distance is slightly shorter, temperature compensation can be used for performance compensation.
- Triggered in two ways: (Jumper selectable)
  - non-repeatable trigger: the sensor output high, the delay time is over, the output is automatically changed from high level to low level;
  - repeatable trigger: the sensor output high, the delay period, if there is human activity in its sensing range, the output will always remain high until the people left after the delay will be high level goes low (sensor module detects a time delay period will be automatically extended every human activity, and the starting point for the delay time to the last event of the time).
- With Induction blocking time (the default setting: 2.5s blocked time): sensor module after each sensor output (high into low), followed by a blockade set period of time, during this time period sensor does not accept any sensor signal. This feature can be achieved sensor output time "and" blocking time "interval between the work can be applied to interval detection products; This function can inhibit a variety of interference in the process of load switching. (This time can be set at zero seconds – a few tens of seconds).
- Wide operating voltage range: default voltage DC4.5V-20V.
- Micropower consumption: static current <50 microamps, particularly suitable for battery-powered automatic control products.
- Output high signal: easy to achieve docking with the various types of circuit.

### Adjustment:

- Adjust the distance potentiometer clockwise rotation, increased sensing distance (about 7 meters), on the contrary, the sensing distance decreases (about 3 meters).
- Adjust the delay potentiometer clockwise rotation sensor the delay lengthened (300S), on the contrary, shorten the induction delay (5S).

### Instructions for use:

- Sensor module is powered up after a minute, in this initialization time intervals during this module will output 0-3 times, a minute later enters the standby state.
- Should try to avoid the lights and other sources of interference close direct module surface of the lens, in order to avoid the introduction of interference signal malfunction; environment should avoid the wind flow, the wind will cause interference on the sensor.
- Sensor module with dual probe, the probe window is rectangular, dual (A B) in both ends of the longitudinal direction
  - so when the human body from left to right or right to left through the infrared spectrum to reach dual time, distance difference, the greater the difference, the more sensitive the sensor,
  - when the human body from the front to the probe or from top to bottom or from bottom to top on the direction traveled, double detects changes in the distance of less than infrared spectroscopy, no difference value the sensor insensitive or does not work;
- The dual direction of sensor should be installed parallel as far as possible in line with human movement. In order to increase the sensor angle range, the module using a circular lens also makes the probe surrounded induction, but the left and right sides still up and down in both directions sensing range, sensitivity, still need to try to install the above requirements.

```

#include <Wire.h> // Incluye la librería Wire para la comunicación I2C
#include <LiquidCrystal_I2C.h> // Incluye la librería LiquidCrystal_I2C para
controlar el LCD
#include "DHTesp.h" // Librería para el sensor DHT22 de temperatura y humedad
#include <WiFi.h> // Librería WiFi para conectar a la red local por WiFi
#include "UbidotsEsp32Mqtt.h"
int pinDHT = 17; // Declara el pin del puerto GPIO del ESP32
#define trigger 3 // Define el pin del trigger del sensor ultrasonico
#define echo 4 // Define el pin del eco del sensor ultrasonico
#define pin_led2 19
#define MQ_PIN 36
// Variables para el LCD
LiquidCrystal_I2C lcd(0x27, 20, 4); // Dirección I2C y dimensiones del LCD

const char *UBIDOTS_TOKEN = "BBFF-MkVdksOd7HKWXPg19sZQ86lktIiALt";
// Put here your Ubidots TOKEN
const char *WIFI_SSID = "Redmi Note 10 Pro"; // Put here your Wi-Fi SSID
const char *WIFI_PASS = "099115926"; // Put here your Wi-Fi password
const char *DEVICE_LABEL = "modulo_marlon"; // Put here your Device label to
which data will be published
const char *TEMP_VARIABLE_LABEL = "temp"; // Etiqueta de la variable de
temperatura en Ubidots
const char *HUMIDITY_VARIABLE_LABEL = "hum"; // Etiqueta de la variable de
humedad en Ubidots
const char *DISTANCIA_VARIABLE_LABEL = "dist"; // Etiqueta de la variable de
humedad en Ubidots
const int PUBLISH_FREQUENCY = 5000; // Frecuencia de actualización en
milisegundos
unsigned long timer;

void setup() {
  Serial.begin(115200); // Inicializa el puerto serial
  lcd.begin(); // Inicializa el LCD
  lcd.backlight(); // Enciende la retroiluminación del LCD
  dht.setup(pinDHT, DHTesp::DHT22); // Inicializa el sensor DHT22
  ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
  ubidots.setCallback(callback);
  ubidots.setup();
  ubidots.reconnect();
  pinMode(trigger, OUTPUT); // Configura el pin del trigger como salida
  pinMode(echo, INPUT); // Configura el pin del echo como entrada
  pinMode(pin_led2, OUTPUT); // Configura el pin para los indicadores LED
  pinMode(MQ_PIN, INPUT);
}

```

