



Técnico superior en Telecomunicaciones

Programación

Proyecto:

Visualizador de Temperatura y Humedad

Alumnos:

- Karina Jazmin Barbero.
- Macarena Aylén Carballo.
- Nicolás Nahuel Barrionuevo.
- Raúl Antonio Jara.

Profesor:

- Lisandro Lanfranco

Índice

Resumen	2
Stack Tecnológico.....	3
Introducción.....	4
Metodología	6
Resultados	6
Conclusión.....	7

Resumen del Proyecto

Este proyecto se enfoca en el diseño e implementación de un dispositivo IoT para el monitoreo de variables ambientales, en este caso la temperatura y la humedad.

El sistema utiliza un sensor DHT11 para relevar ambas variables, las cuales son procesadas por un microcontrolador ESP32 que se comunica vía Wifi con una base de datos MySQL para almacenar los datos recogidos. Además de las lecturas de temperatura y humedad, el sistema también monitorea el nivel de batería del dispositivo para garantizar su funcionamiento continuo.

Se ha implementado un módulo LED WS2812 como actuador, que se activa cuando los valores de temperatura o humedad superan los umbrales predefinidos, proporcionando una señal visual de alerta.

El diseño eléctrico se ha desarrollado para integrar de manera eficiente el sensor, el medidor de nivel de batería, y el actuador al ESP32. La elección del DHT11 como sensor se basa en su compatibilidad con el ESP32, su precisión suficiente para aplicaciones de monitoreo ambiental, y su facilidad de uso en proyectos de IoT.

La activación del actuador está programada para que el LED cambie de color cuando exceda la temperatura o la humedad supere algún parámetro, lo que permite al usuario reaccionar ante condiciones ambientales fuera de lo deseado.

Este informe documenta la metodología de implementación y proporciona un resumen de los resultados, incluyendo las especificaciones técnicas de los componentes utilizados, así como el diseño del sistema y la justificación del criterio de accionamiento.

Las pruebas del sistema han demostrado un funcionamiento eficiente, con lecturas precisas del sensor y la activación adecuada del actuador en tiempo real.

Stack Tecnológico

Este proyecto IoT para el monitoreo de temperatura y humedad utiliza las siguientes tecnologías y herramientas:

1. Lenguajes de Programación

- **C++:** Utilizado para programar el ESP32 y controlar los sensores, actuadores, y la lógica del dispositivo.

2. Entorno de Desarrollo

- **Visual Studio Code con PlatformIO:** IDE y extensión para el desarrollo y depuración del firmware del ESP32, integrando librerías y configuraciones de hardware necesarias.

3. Microcontrolador

- **ESP32:** Microcontrolador que procesa los datos recogidos del sensor DHT11, controla el LED WS2812, y gestiona la conexión con la base de datos MySQL.

4. Base de Datos

- **MySQL:** Base de datos relacional para el almacenamiento y gestión de los datos de temperatura, humedad, y nivel de batería.
- **MySQL Workbench:** Herramienta de administración visual para modelar, diseñar y consultar la base de datos.

5. Componentes Físicos

- **Sensor DHT11:** Captura las variables de temperatura y humedad.
- **LED WS2812:** Actúa como indicador visual del estado ambiental.
- **Medidor de nivel de batería:** Monitorea el estado de la batería del dispositivo.

6. Conectividad

- **Wi-Fi:** Utilizado para la transmisión de los datos desde el ESP32 hacia la base de datos MySQL de manera inalámbrica.

7. Python:

- Utilizado para conectar la base de datos MySQL y realizar consultas a los datos almacenados para análisis.

Introducción

Visualizador de Temperatura y Humedad.

El presente informe documenta el desarrollo de un sistema IoT diseñado para el monitoreo ambiental de temperatura y humedad, utilizando un microcontrolador ESP32 y un sensor DHT11. Estos sistemas se encuentran en aplicaciones industriales, agrícolas y domésticas, permitiendo un control eficiente del entorno y facilitando la automatización de procesos en función de parámetros climáticos.

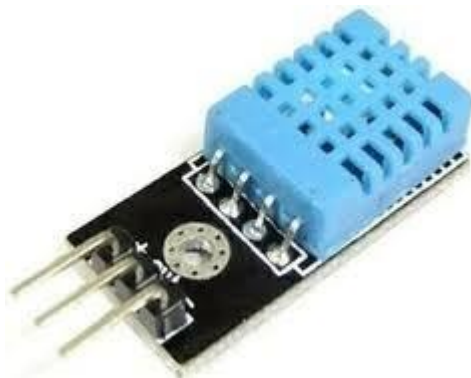
La medición y control de variables ambientales como la temperatura y la humedad son cruciales en entornos como invernaderos, almacenes de alimentos y espacios interiores donde las condiciones deben mantenerse dentro de ciertos rangos.

En trabajos previos, la implementación de sistemas de monitoreo ambiental se ha llevado a cabo utilizando el ESP32 que proporciona una solución más eficiente gracias a su capacidad de conexión WiFi integrada y su procesamiento de bajo consumo, lo que lo convierte en una opción ideal para sistemas autónomos de monitoreo. Además, la combinación con el sensor DHT11 permite obtener mediciones de temperatura y humedad a un costo reducido, haciendo que este tipo de soluciones IoT sea accesible para aplicaciones a pequeña y gran escala.

El objetivo principal de este trabajo es desarrollar un sistema capaz de medir, almacenar y analizar las variables de temperatura y humedad en tiempo real. Para ello, se utilizarán los datos obtenidos por el sensor DHT11, que serán enviados a una base de datos MySQL para su posterior análisis. Además, se implementará un sistema de alerta visual mediante un módulo LED WS2812, que cambiará de color cuando las condiciones excedan ciertos umbrales, proporcionando una señal inmediata al usuario sobre la necesidad de intervención.

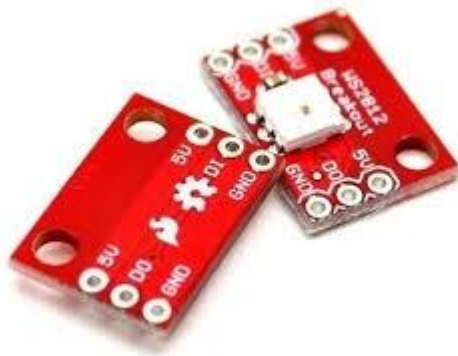
Componentes a utilizar y sus características

1) *Sensor DHT11*



Modelo	DHT11
Alimentación	de 3,5 V a 5 V
Consumo	2,5 mA
Señal de salida	Digital
Interfaz de comunicación	Interfaz de un solo hilo
Corriente de Operación	0.3 mA durante la medición, y hasta 1 mA
Frecuencia de muestreo	Nueva lectura cada segundo
Temperatura	
Rango	de 0°C a 50°C
Precisión	a 25°C \pm 2°C
Resolución	1°C (8-bit)
Humedad	
Rango	de 20% RH a 90% RH
Precisión	entre 0°C y 50°C \pm 5% RH
Resolución	1% RH

2) **Módulo LED WS2812**



Modelo	Módulo LED WS2812
Tipo de LED	RGB (Rojo, Verde, Azul)
Voltaje de operación	3.3V a 5V
Corriente de operación	18-20 mA cuando está encendido al máximo brillo
Control	Protocolo de comunicación de una sola línea
Temperatura de operación	-25°C a 80°C
Tiempo de respuesta	800 kHz
Dimensiones	cada LED individual mide 5mm x 5mm
Capacidad de Cadena	pueden conectarse en serie
Colores	16,777,216 colores diferentes
Protocolos de comunicación	pulsos de ancho modulado (PWM)

3) ESP 32



Es un microcontrolador altamente integrado con capacidades avanzadas de comunicación inalámbrica, utilizado comúnmente en proyectos de Internet de las Cosas (IoT) debido a su conectividad WiFi y Bluetooth, su versatilidad y bajo consumo energético. El modelo con 38 pines es ideal para proyectos que requieren múltiples entradas y salidas, además de comunicación con varios sensores y actuadores.

Especificaciones Principales:

- **CPU:**
 - Dual-core Tensilica LX6 a 160/240 MHz.
 - 32 bits, con un total de 520 KB de SRAM (memoria RAM estática).
 - Proporciona suficiente potencia de procesamiento para tareas complejas como la conexión WiFi y el procesamiento de datos de múltiples sensores.
- **WiFi:**
 - Conectividad WiFi 802.11 b/g/n a 2.4 GHz.
 - Permite conectar el ESP32 a una red WiFi local, lo que lo convierte en una solución ideal para proyectos IoT donde la transmisión de datos en tiempo real es esencial.
- **Bluetooth:**
 - Soporta Bluetooth 4.2 y Bluetooth Low Energy (BLE).
 - Facilita la comunicación con dispositivos móviles y otros módulos Bluetooth.
- **GPIO:**
 - 38 pines GPIO disponibles para uso general.

- Se pueden asignar a varias funciones como PWM, ADC (conversor analógico-digital), DAC (conversor digital-analógico), SPI, I2C, UART, y más.
- Algunos pines utilizados en este proyecto incluyen:
 - GPIO 15 para el sensor DHT11.
 - GPIO 4 para el módulo LED WS2812.
- **ADC** (Conversor Analógico-Digital):
 - Cuenta con 18 canales ADC de 12 bits, lo que permite medir señales analógicas con alta precisión.
 - Uno de los pines ADC se utiliza en este proyecto para medir el nivel de batería.
- **DAC** (Conversor Digital-Analógico):
 - Incluye 2 canales DAC de 8 bits, lo que permite la conversión de señales digitales a analógicas.
- **Memoria Flash:**
 - Hasta 4 MB de memoria Flash integrada, lo que permite almacenar programas más complejos y almacenar datos localmente si es necesario.
- **Tensión de operación:**
 - El ESP32 funciona con 3.3V, lo que lo hace compatible con sensores y periféricos de bajo voltaje.
 - Cuenta con múltiples modos de ahorro de energía, lo que lo hace ideal para proyectos alimentados por batería.
- **Interfaz de comunicación:**
 - I2C, SPI, UART, y PWM.
 - Estos buses de comunicación permiten la interacción con una amplia variedad de sensores, actuadores y otros dispositivos, lo que lo convierte en un microcontrolador versátil.
- **Capacidades de Red y Protocolo:**
 - Soporte para IPv4 y IPv6.
 - Soporte para TCP, UDP, HTTP/HTTPS, MQTT, entre otros, lo que permite la integración con plataformas en la nube y servicios web.
- **Consumo energético:**
 - Bajo consumo en modo de reposo, con opciones de modos de bajo consumo de energía, que permiten reducir el consumo a solo unos microamperios cuando no se están ejecutando tareas intensivas.

- **Temperatura de operación:**

- El ESP32 funciona en un rango de temperatura de -40°C a 125°C, lo que lo hace adecuado para aplicaciones en entornos extremos.

Metodología de Trabajo

El desarrollo del proyecto se llevó a cabo en dos fases principales: simulación y implementación física.

A continuación, se describe el proceso seguido en cada una de estas fases, detallando los componentes utilizados, la programación de los mismos y las pruebas realizadas para asegurar el correcto funcionamiento del sistema.

1. Simulación

Antes de proceder con la implementación física, se realizaron simulaciones para comprobar el correcto funcionamiento de los componentes y su interconexión, utilizando la plataforma Wokwi, que permite emular el comportamiento de microcontroladores y sensores en entornos de IoT.

La simulación incluyó los siguientes pasos:

- **Paso 1-** Selección del hardware virtual:

Se seleccionó un microcontrolador ESP32 de 38 pines, un sensor DHT11 para la medición de temperatura y humedad, y un módulo LED WS2812 como actuador visual.

- **Paso 2-** Configuración de la simulación:

Se configuraron las conexiones virtuales entre el ESP32, el sensor DHT11 y el módulo WS2812.

El GPIO 15 fue utilizado para conectar el DHT11, mientras que el GPIO 4 se usó para el WS2812.

- **Paso 3-** Programación en C++:

Se programó el ESP32 para recibir las lecturas de temperatura y humedad del sensor DHT11 y, en base a ciertos umbrales predefinidos (30°C para temperatura y 60% para humedad), cambiar el color del LED WS2812.

- **Paso 4-** Pruebas en simulación:

Se realizaron pruebas en el entorno de simulación para verificar que los componentes respondieran correctamente a las lecturas del sensor. Las pruebas mostraron que el sistema visual del LED se activaba cuando las condiciones excedían los umbrales, confirmando el comportamiento esperado.

2. Implementación Física

Una vez validado el diseño en simulación, se procedió con la implementación física del proyecto. Los siguientes pasos describen cómo se realizó la construcción del sistema y las pruebas físicas:

- **Montaje de los componentes:**

Se utilizaron componentes reales, incluyendo el ESP32 de 38 pines, el DHT11 y el módulo LED WS2812.

El DHT11 se conectó al pin GPIO 15 del ESP32 para recibir las lecturas de temperatura y humedad, mientras que el WS2812 se conectó al pin GPIO 4 para actuar como indicador visual.

- **Programación en PlatformIO:**

El código desarrollado en la simulación fue migrado a la plataforma VSCode con PlatformIO, donde se ajustaron los parámetros para la implementación física. Se utilizó el mismo código en C++ para controlar el sistema y enviar los datos de temperatura y humedad a una base de datos MySQL.

- **Conexión WiFi:**

Se configuró el ESP32 para conectarse a una red WiFi local, permitiendo la transmisión de datos a un servidor remoto donde se almacenan en una base de datos. El código Python se utilizó para gestionar las consultas hacia la base de datos y almacenar las mediciones de manera eficiente.

- **Medición del nivel de batería:**

Para garantizar la autonomía del dispositivo, se implementó un sistema de medición del nivel de batería utilizando un divisor de voltaje conectado a un pin ADC del ESP32. Este sistema monitorea continuamente el voltaje de la batería, permitiendo un seguimiento en tiempo real de su estado.

- **Pruebas de campo:**

Finalmente, se realizaron pruebas en entornos controlados, donde se monitorearon las condiciones ambientales reales. Se verificó el correcto funcionamiento del sistema en la toma de mediciones, el envío de datos a la base de datos, y la activación del LED cuando se alcanzaron los umbrales definidos.

Esta metodología asegura que el sistema sea replicable, ya que describe detalladamente tanto la fase de simulación como la implementación física. Cualquier persona que desee repetir este trabajo o un proyecto similar podrá seguir estos pasos y utilizar herramientas como Wokwi y PlatformIO para llevar a cabo su propio sistema de monitoreo.

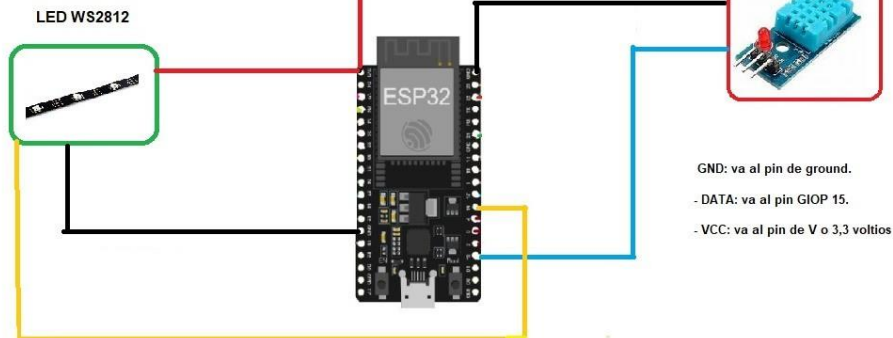
- **Implementación:**

- <https://wokwi.com/projects/409147778803738625>
- <https://github.com/ISPC-TST-MOD-PROGR-2024/Visualizador-de-Temperatura-y-Humedad>

Diagrama de Conexión

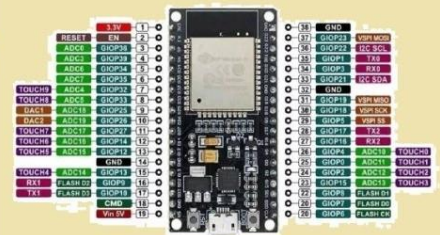
ESQUEMA ELECTRICO

- GND: va al pin de ground.
- PIN DIN : va al GPIO 4.
- VCC: va al pin de V o 3,3 voltios

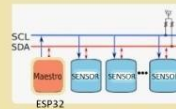


- GND: va al pin de ground.
- DATA: va al pin GPIO 15.
- VCC: va al pin de V o 3,3 voltios

Plataforma IoT para Monitoreo de Temperatura y Humedad con ESP32 y Almacenamiento en MySQL



Protocolo I2C



- (36) - GPIO22 - I2C SCL
- (33) - GPIO21 - I2C SDA

Protocolo SPI



- (37) - GPIO23 - VSPH MOSI
- (31) - GPIO19 - VSPH MISO
- (38) - GPIO18 - VSPH SCK
- (29) - GPIO5 - VSPH CS