

EVIDENCIA 1: DESARROLLO DE PROTOTIPOS IoT CON ESP32

Aproximación al mundo del trabajo

Integrantes:

Jimena Galleguillo

Franco Gonzalo Arce

Magali Maylen Bechis

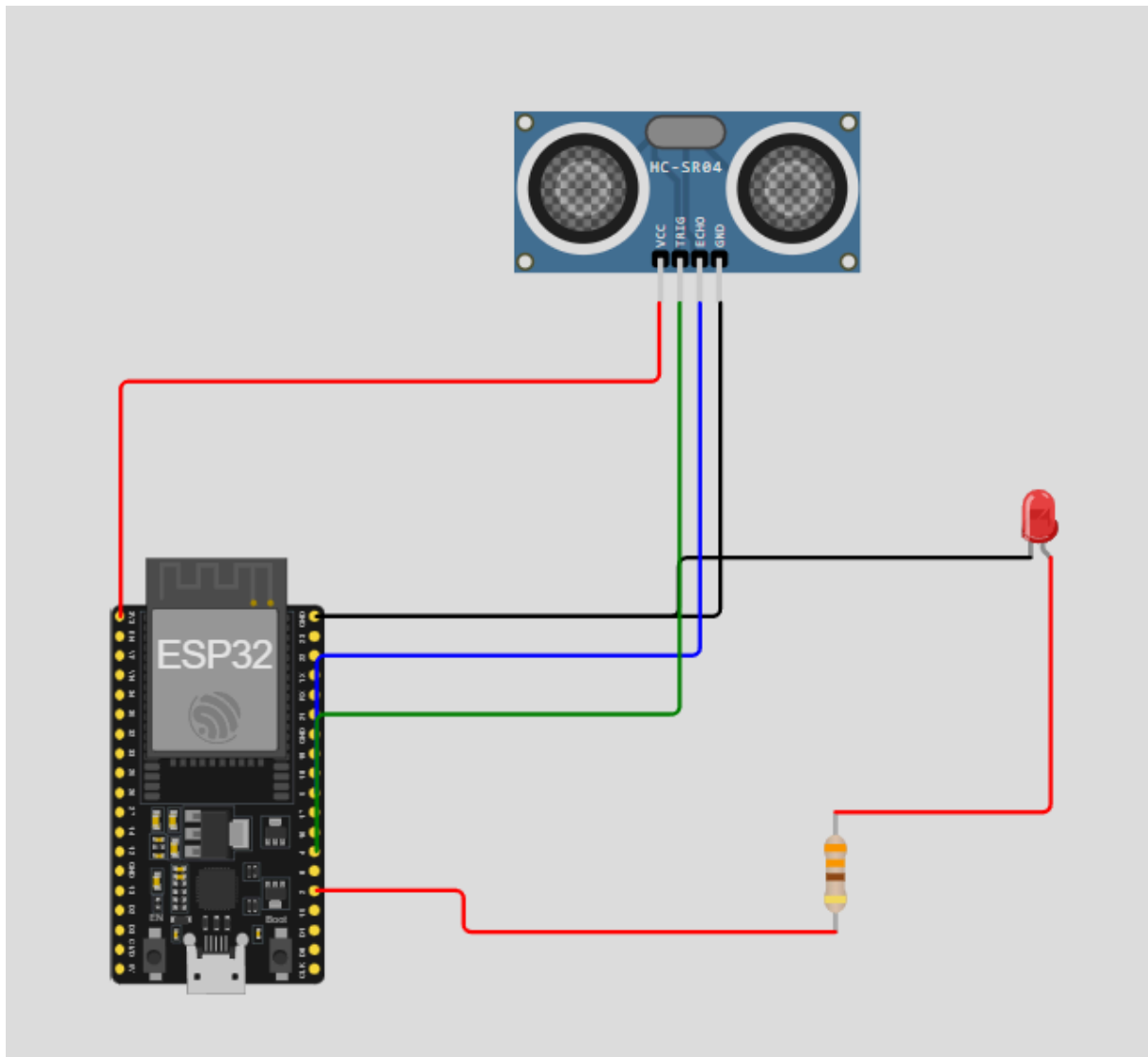
Proyecto: Medidor de nivel de agua.

En este proyecto nos centramos en el control y monitoreo del nivel de agua en tanques, mide la distancia desde la parte superior del tanque hasta la superficie del agua, lo que hace que se pueda calcular el nivel de agua en el tanque.

Aparte de medir el agua también realiza alertas a través de un led cuando el nivel de agua es bajo.

Los componentes utilizados son los siguientes:

- Microcontrolador ESP32.
- Sensor Ultrasónico HC SR04.
- LED.
- Resistor.



Como se muestra en la imagen estas serían las conexiones del proyecto.
Conexiones del HCSR04:

1. VCC (pin 1 del HCSR04) → Al pin de 3v3 de la microcontroladora.
2. Trig (pin 2 del HCSR04) → Al pin 4
3. Echo (pin 3 del HCSR04) → Al pin 21
4. GND (pin 4 del HCSR04) → Al pin de GND de la microcontroladora.

Conexiones del LED:

1. Ánodo (largo, positivo) del LED → Al GND de la microcontroladora.
2. Cátodo (corto, negativo) del LED → Al resistor
3. Otro extremo del resistor → Al pin 2

Código fuente detallado en Micropython: [Código](#)

Link al proyecto en Wokwi: [Medidor de nivel de agua Wokwi](#)

Explicación del Funcionamiento del Código y la Lógica de Programación

El código que implementa el sistema de monitoreo del nivel de agua se basa en el uso del sensor ultrasónico HCSR04 y el microcontrolador ESP32.

Inicialización:

Se importan las librerías necesarias, se inicializa el sensor HCSR04 con los pines asignados para el Trigger y Echo, y se configura un pin para controlar el LED.

Se definen constantes como la altura total del tanque, el umbral de nivel adecuado, el tiempo de parpadeo del LED, y el tiempo entre mediciones.

Bucle Principal:

- El bucle se ejecuta continuamente para medir la distancia y calcular el nivel de agua
- Medición de Distancia: Se utiliza el método `distance_cm()` del sensor para medir la distancia desde el sensor hasta la superficie del agua.
- Cálculo del Nivel de Agua: La distancia medida se usa para calcular el `nivel_agua` restando la distancia de la altura total del tanque.
- Condición de Alerta: Se compara el nivel de agua calculado con el umbral. Si el nivel es inferior al umbral predefinido, el LED parpadea cinco veces y se imprime un mensaje de alerta en la consola.
- Si el nivel de agua es normal, el LED permanece apagado.
- Manejo de Errores: Se incluye un bloque `try-except` para manejar y reportar cualquier error en la medición de la distancia, garantizando que el sistema sea más robusto y no se detenga ante fallos.

Resultados obtenidos y pruebas realizadas

- Pruebas de Medición: Llevamos a cabo pruebas durante las cuales se colocó el sensor a distintas distancias de la superficie del agua para verificar la precisión de la medida. Esto mostró que el sensor es capaz de ofrecer resultados consistentes dentro de su rango de operación.
- Funcionamiento del LED: Al ajustar el nivel del agua por debajo del umbral, observamos que el LED parpadea correctamente, indicando el estado bajo del nivel del agua. A su vez, cuando el nivel era adecuado, el LED permanece apagado.

- Comprobación de Errores: Simulamos condiciones de error (como desconectar temporalmente el sensor) para confirmar que el sistema maneja adecuadamente las excepciones y sigue funcionando sin interrupciones.

Posibles Mejoras Futuras

Notificaciones Remotas: Integrar funcionalidades de Wi-Fi para enviar notificaciones a un teléfono móvil o registrar datos en la nube.

Conclusiones

El proyecto medidor de nivel de agua, demuestra una implementación efectiva y práctica de los conocimientos adquiridos en la materia, tiene la capacidad de medir niveles de agua de manera precisa y proporcionar alertas a través de un LED.

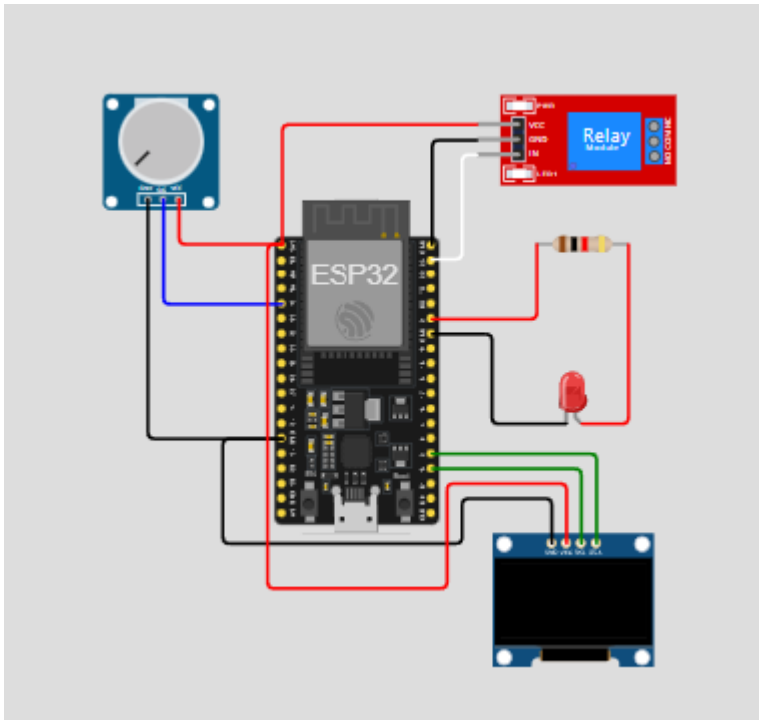
Los resultados que obtuvimos mediante las pruebas fueron satisfactorios, lo que demuestra que se puede usar correctamente, gracias a este proyecto ampliamos nuestros conocimientos, por medio de investigaciones y búsquedas para realizar el trabajo solicitado.

Proyecto: Controlador de riego automático

Este programa está diseñado para monitorear la humedad del suelo y activar o desactivar automáticamente un sistema de riego basado en los datos obtenidos. El objetivo es optimizar el consumo de agua y mantener el suelo en las condiciones de humedad ideales.

Los componentes utilizados son los siguientes:

- Microcontrolador ESP32.
- Sensor de humedad Potenciómetro.
- Módulo de relé.
- LED.
- Resistor.



Como se muestra en la imagen estas serían las conexiones del proyecto.

Código fuente detallado en Micropython: [Código fuente](#)

Link al proyecto en Wokwi: [Controlador de riego automático](#)

Explicación del Funcionamiento del Código y la Lógica de Programación

El código que implementa el sistema controlador de riego automático se basa en el uso del sensor de humedad Potenciómetro y el microcontrolador ESP32.

Inicialización:

Se importan las librerías necesarias para trabajar con los pines de entrada/salida, el conversor analógico-digital (ADC) y el protocolo de comunicación I2C.

Se definen constantes como el umbral de humedad, tiempo de espera entre mediciones, y los mensajes a mostrar.

Bucle Principal:

El bucle se ejecuta continuamente para controlar el riego.

controlar_riego(): Dentro del bucle, se llama a la función `controlar_riego()`. Esta función, como su nombre indica, se encarga de todo el proceso de control del riego:

- Lee el valor de humedad del sensor.
- Compara este valor con el umbral predefinido.
- Decide si activar o desactivar la bomba y el LED en función de la comparación.
- Actualiza la pantalla OLED con la información más reciente.

sleep(1): Después de llamar a la función `controlar_riego()`, el programa se pausa durante 1 segundo. Esto significa que el sistema tomará una decisión sobre el riego cada segundo.

Resultados obtenidos y pruebas realizadas

Pruebas del Sensor de Humedad

Realizamos pruebas simulando diferentes niveles de humedad mediante el uso de un potenciómetro, representando el sensor de humedad. Ajustamos el potenciómetro a diferentes valores y verificamos cómo el sistema responde (al ejecutar el código se debe mover la aguja del potenciómetro). Los resultados fueron consistentes: cuando el valor de humedad estaba por debajo del umbral predefinido, el sistema activaba la bomba de agua y encendía el LED. Cuando el valor de humedad superaba el umbral, el sistema apagaba la bomba y el LED.

Funcionamiento del LED

Probamos el comportamiento del LED, el cual simula el funcionamiento de la bomba de agua, en relación al valor de humedad detectado. Observamos que el LED se enciende cuando la humedad es baja y se apaga cuando la humedad es adecuada, confirmando que el sistema está funcionando según lo esperado. Además, el LED proporciona una indicación visual clara del estado del sistema, lo que facilita su monitoreo.

Comprobación del Módulo de Relé

El módulo de relé activó y desactivó la bomba de riego según lo programado. Se realizaron múltiples pruebas para verificar que el sistema de control de riego respondiera correctamente a los cambios en la humedad, y en todas las ocasiones, el relé funciona correctamente, activando o desactivando la bomba según los niveles de humedad detectados.

Manejo de Errores

Simulamos posibles errores, como desconectar temporalmente el sensor o el potenciómetro, y verificamos que el sistema continuó operando sin interrupciones críticas. Los errores fueron manejados correctamente por el programa, mostrando mensajes adecuados y asegurando que el sistema no se bloquee.

Conclusiones

El controlador de riego automático demostró ser una herramienta útil para mantener la humedad del suelo en niveles óptimos de manera automática. Gracias a este proyecto, optimizamos el consumo de agua y mejoramos la eficiencia del riego. Las pruebas realizadas validaron la precisión y confiabilidad del sistema, que responde adecuadamente a los cambios en los niveles de humedad. Este prototipo tiene el potencial de ser ampliado con características adicionales, como la integración de la posibilidad de recibir notificaciones en dispositivos móviles.