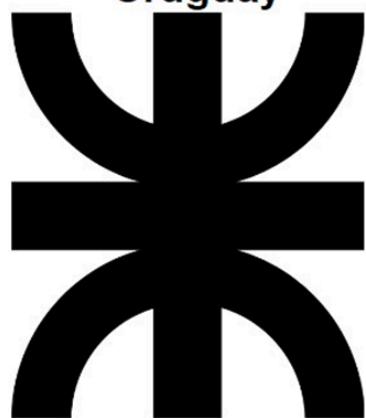


**UNIVERSIDAD TECNOLÓGICA
NACIONAL**
**Facultad Regional Concepción del
Uruguay**



Ingeniería en Sistemas de Información

Tecnologías para la automatización

Trabajo Práctico N°3: Microcontroladores

Sistema de monitoreo de planta

INTEGRANTES:

- Emmanuel Davezac
- Esteban Gay
- Felipe Palazzi
- Mauricio Nahuel Salto.

DOCENTES:

- Mauro Sander
- Jaime Piperno

1) Selección del Proyecto:

a) Justificación de la Utilidad Práctica

El proyecto de monitoreo de plantas fue seleccionado por su utilidad práctica en la agricultura y en la gestión de cultivos.

El sistema resuelve el desafío de mantener condiciones ambientales óptimas para el crecimiento de especies vegetales sensibles. En lugar de la supervisión manual constante, el sistema provee una vigilancia automatizada y alertas en tiempo real.

Permite a los usuarios actuar preventivamente, minimizando el riesgo de daño o pérdida de la planta debido a condiciones adversas.

La integración con Discord transforma el sistema en una solución IoT, permitiendo al usuario recibir alertas instantáneas sin importar su ubicación.

b) Interacción con el Entorno

El sistema interactúa con el entorno mediante:

Sensores

- **Sensor DHT11:** Mide la temperatura y la humedad del aire.
- **Sensor LDR:** Mide la intensidad de la luz ambiental.

Actuadores

- **LEDs Rojos y Verdes:** Un par de LEDs muestran el estado de la temperatura y otro par de LEDs muestra la humedad ambiental localmente. Indicando
 - Rojo = Alerta
 - Verde = OK.
- **Integración con Discord:** Actúa como un actuador virtual al enviar una notificación de alerta al canal de Discord.

2) Trabajo a Presentar

b) Explicación de los Componentes Utilizados

| Componente | Función Principal | Características Clave |
|--------------|---|---|
| Placa ESP32 | Ejecuta el código, procesa datos y gestiona la conectividad Wi-Fi. | Voltaje de alimentación: 5V (usb) Voltaje de Trabajo: 3.3V. Conectividad: Wi-Fi integrado. Entradas y salidas analógicas y digitales. |
| Sensor DHT11 | Mide la temperatura y la humedad del aire. | Voltaje de Trabajo: 3.3V. Modo de Trabajo: Digital. Rango de Temp: -40 a 80°C. Rango de Humedad: 0 a 100%. |
| Sensor LDR | Mide la intensidad de la luz | Voltaje de trabajo 3.3V Modo de Trabajo: Analógico (resistencia variable). Utiliza el ADC del ESP32. El valor de la resistencia es inversamente proporcional a la intensidad de luz. |
| LEDs | Indicadores visuales del estado del entorno. Usamos 2 Rojos y 2 Verdes | Voltaje Típico: 2-3V. Controlados mediante pines digitales del ESP32. |
| Resistencias | Regulan la corriente que circula. | |

| | | |
|--|---|--|
| | Usamos 4 resistencias de 220Ω y 1 de $10k\Omega$ | |
|--|---|--|

c) Explicación del Código Fuente

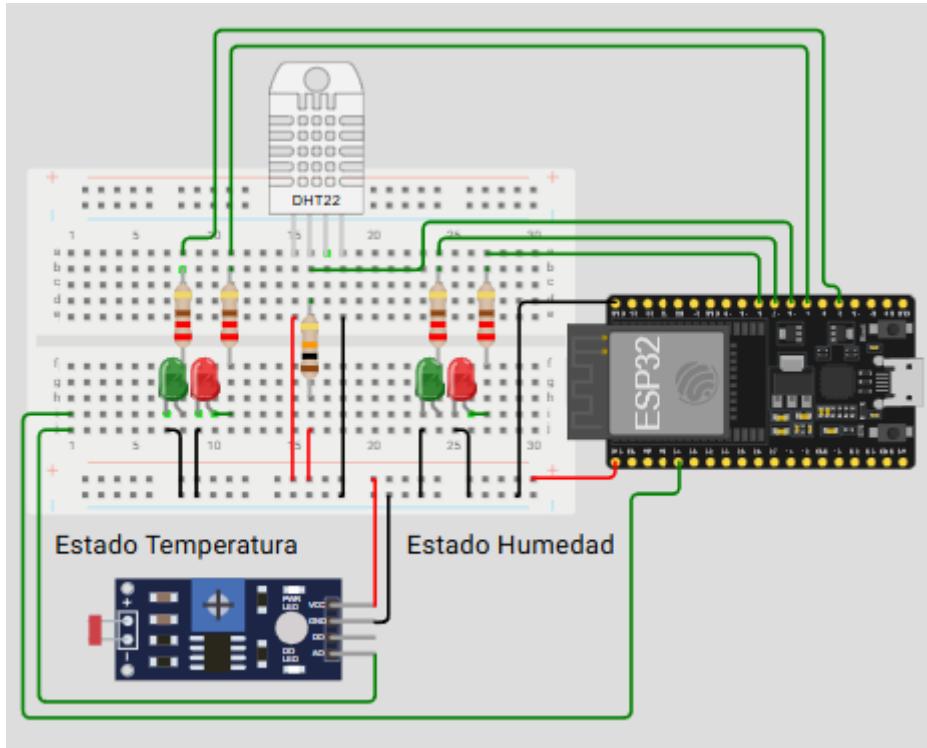
Resumen

- **Configuración:** Se definen los pines, el SSID/Contraseña, la URL del Webhook y los límites críticos (TEMP_CRITICA_ALTA, HUMEDAD_CRITICA_BAJA, VALOR_MAXIMO_LUZ_PERMITIDO).
- **enviarMensajeDiscord():** Función encargada de establecer una conexión HTTPS segura, construir la carga útil (payload) en formato JSON, y enviar la alerta mediante una petición POST.
- **setup():** Inicializa la comunicación serial, configura los pines (Entrada/Salida), inicializa el sensor DHT, y establece la conexión Wi-Fi.
- **loop():** Es el ciclo principal. Lee los sensores, compara las lecturas con los límites críticos, acumula los mensajes de alerta en la variable mensajeAlerta, controla el estado de los LEDs y, si mensajeAlerta no está vacío, envía la notificación a Discord.
 - Utiliza delay(INTERVALO_LECTURA_M) para espaciar las lecturas.
 - Utiliza COOLDOWN_ALERTA_M para espaciar alertas.

Código fuente

<https://github.com/EmmaDavezac/Automatizacion-Sistema-de-monitoreo-de-planta>

d) Diagrama de Conexión



Circuito en el simulador

<https://wokwi.com/projects/446632239126116353>

e) Conceptos de la Materia Ejemplificados

1. Realimentación:

El sistema utiliza la lectura de los sensores como realimentación del estado del entorno. Esta información se compara con los límites críticos definidos. Si hay una desviación, el sistema genera una

acción de salida (alerta por Discord y LEDs rojos) para que el usuario tome medidas (por ejemplo regar la planta o cubrirla) cerrando el ciclo.

2. Sistema Discreto:

El sistema es un ejemplo perfecto de un sistema discreto, ya que toma un entorno físico continuo (temperatura, humedad y luz) y lo convierte en una secuencia de pasos digitales.

Un sistema no monitorea el entorno constantemente, sino en instantes de tiempo específicos.

Aunque los fenómenos físicos son continuos, el microcontrolador los debe convertir en números enteros discretos.

El sistema opera con valores discretos y las salidas también son discretas.

3. Sistema Abierto :

El sistema no tiene la capacidad de auto-corregirse. Simplemente mide, compara y alerta. La corrección del error siempre depende de una intervención externa.

f) Complicaciones y Soluciones

| Complicación | Solución |
|---|---|
| El simulador no tiene el Sensor DHT11 | En el simulador usamos el sensor DHT22 que cumple la misma función y en la práctica modificamos la conexión de los cables ya que no tienen el mismo orden de pines. |
| El simulador no tiene el mismo LDR | En el simulador usamos el LDR disponible y usamos la salida analógica del sensor. |
| Lectura Inversa del LDR. El sensor LDR reduce su resistencia cuando hay más luz. | Invertimos la lógica de comparación. |
| Discord a veces rechazaba las peticiones para enviar el mensaje | Los Webhooks de Discord requieren HTTPS. Se incluyó la librería WiFiClientSecure.h y se utilizó el método client.setInsecure() |
| Mostrar múltiples alertas en un mensaje. Las alertas se solapaban entre sí. | Se utilizó una variable para acumular todas las alertas detectadas en cada ciclo, enviando un único mensaje completo. |