

Sistema Automatico de Riego

<<Giardelli Alan, Ojeda Ernesto, Rodriguez Bustos Mariano, Ruiz Alejandro>>
<<4272105,42592023,42194177,42375350>>
<<Lunes, Grupo 2>>

¹Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina

Resumen.

Sistema de Riego Automático basado en ESP32 con FreeRTOS

Este trabajo presenta el desarrollo de un sistema embebido para automatizar el riego de cultivos mediante una máquina de estados ejecutada sobre un microcontrolador llamado ESP32. El sistema utiliza sensores de humedad y luz para detectar condiciones ambientales y decidir el inicio o finalización del riego, ya sea de forma automática o por acción manual del usuario. Se implementa una arquitectura basada en eventos y tareas concurrentes gestionadas con FreeRTOS, eliminando el uso de retardos activos y mejorando la eficiencia y respuesta del sistema. Incluye un display LCD para indicar el estado del riego y un buzzer como alerta que indica cuando se comienza a regar. El diseño contempla modo de depuración, parámetros configurables y separación de responsabilidades.

Palabras claves: riego automático, ESP32, FreeRTOS, sensores

1 Introducción

Queridos Pinky y Cerebro:

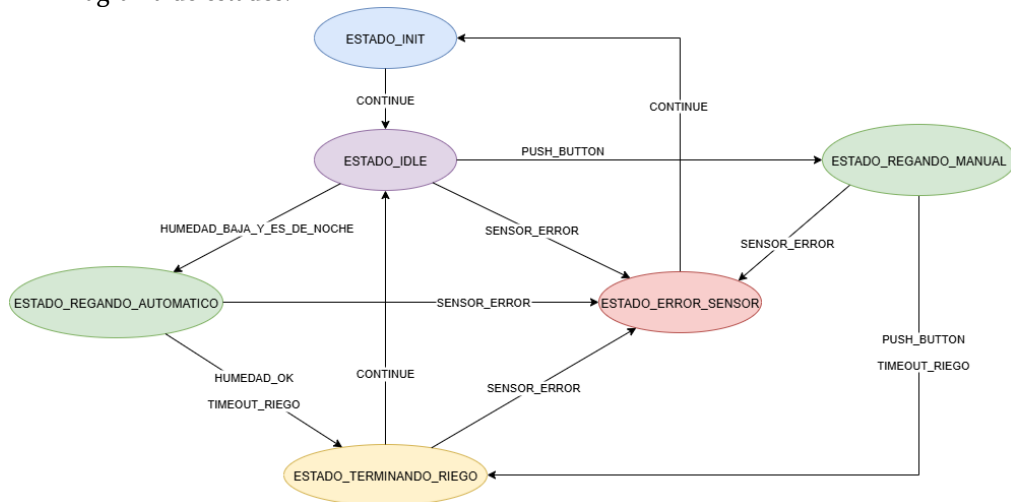
Sabemos que conquistar el mundo no es tarea sencilla, y mucho menos si tienen que perder tiempo regando sus cultivos. Por eso diseñamos este sistema embebido de riego automático, ideal para sus campos de cultivo mutantes.

Este dispositivo, basado en una ESP32 con FreeRTOS, sensores de humedad y luz, se encarga de regar sus plantas solo cuando la tierra lo necesita y el sol ya no es un obstáculo, optimizando recursos y energía. Además, cuenta con un botón de riego manual por si deciden intervenir entre planes de dominación y acelerar el proceso.

Al automatizar el cultivo de su propia comida y experimentos, podrán liberar tiempo valioso para dedicarse a lo que realmente importa: pensar, planear y ejecutar su objetivo de gobernar el mundo... una lechuga hidratada a la vez.

2 Desarrollo

- Diagrama de estados:

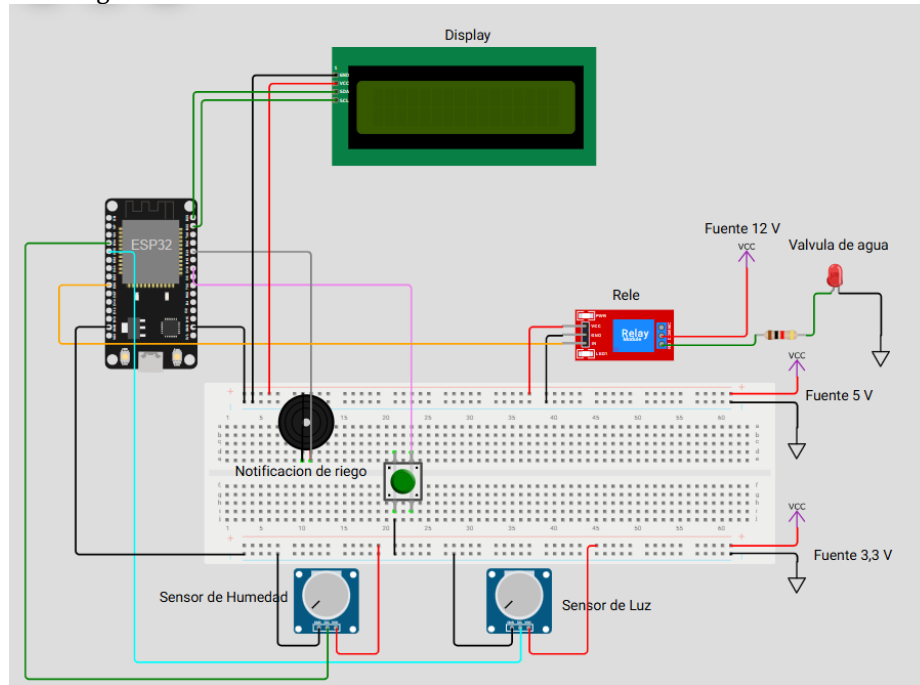


ESTADO	Descripcion
ESTADO_INIT	Inicializar el sistema
ESTADO_IDLE	Sistema a la espera de un evernto (manual o automático).
ESTADO_REGANDO_MANUAL	El riego activo de forma manual
ESTADO_REGANDO_AUTOMATICO	El riego activo de forma automatica por lectura de sensores
ESTADO_TERMINANDO_RIEGO	Se terminó el riego, esperando que finalicen tareas de cierre
ESTADO_ERROR_SENSOR	Algún sensor falla, entra en error.

EVENTO	Descripción
TIPO_EVENTO_CONTINUE	Evento vacío (mantener loop).
TIPO_EVENTO_TIMEOUT_RIEGO	Se terminó el tiempo programado de riego.
TIPO_PUSH_BUTTON	El usuario aprieta un botón (o un input) para regar manualmente.
TIPO_PUSH_BUTTON	El usuario aprieta un botón para finalizar el riego
TIPO_EVENTO_HUMEDAD_BAJA_y_ES_DE_NOCHE	Sensor detecta poca humedad y hay poca luz
TIPO_EVENTO_SENSOR_ERROR	Error de lectura en sensores de humedad o luz.
TIPO_EVENTO_HUMEDAD_OK	La humedad alcanzó el nivel deseado

Estado Actual	Evento recibido	Acción	Nuevo Estado
ESTADO_INIT	TIPO_EVENTO_CONTINUE	Setup inicial	ESTADO_IDLE
ESTADO_IDLE	TIPO_PUSH_BUTTON	Encender riego manual	ESTADO_REGANDO
ESTADO_IDLE	TIPO_EVENTO_HUMEDAD_BAJA_Y_ES_DE_NOCHE	Encender riego automático	ESTADO_REGANDO_AUTOMATICO
ESTADO_REGANDO_AUTOMATICO	TIPO_EVENTO_TIMEOUT_RIEGO	Apagar bomba (antes de tiempo)	ESTADO_TERMINANDO_RIEGO
ESTADO_REGANDO_AUTOMATICO	TIPO_EVENTO_HUMEDAD_OK	Apagar bomba (antes de tiempo)	ESTADO_TERMINANDO_RIEGO
ESTADO_REGANDO_MANUAL	TIPO_EVENTO_TIMEOUT_RIEGO	Apagar bomba	ESTADO_TERMINANDO_RIEGO
ESTADO_REGANDO_MANUAL	TIPO_PUSH_BUTTON	Apagar bomba (antes de tiempo)	ESTADO_TERMINANDO_RIEGO
ESTADO_TERMINANDO_RIEGO	TIPO_EVENTO_CONTINUE	Volver a Idle	ESTADO_IDLE
ESTADO_ERROR_SENSOR	TIPO_EVENTO_CONTINUE	Volver a Setup inicial	ESTADO_INIT
Cualquier estado (menos ESTADO_INIT)	TIPO_EVENTO_SENSOR_ERROR	Apagar bomba, si está prendida. Mostrar error	ESTADO_ERROR_SENSOR

- Diagrama de Conexiones del Circuito de Tinkercad:



- Manual de usuario del embebido simulado:

Introducción:

Este sistema implementa un control automático de riego usando un ESP32 simulado en la plataforma Wokwi. Se basa en una máquina de estados, utiliza FreeRTOS para multitarea y responde tanto a condiciones ambientales como a la interacción del usuario.

Componentes del sistema:

Componente	Descripción
ESP32	Microcontrolador que gestiona el sistema
Sensor de humedad	Simulado con un potenciómetro. Entrega valores analógicos
Sensor de luz	Simulado con un potenciómetro. Entrega valores analógicos
Relé	Simula el control de una válvula de agua

Componente	Descripción
Válvula (LED)	Led que simula la valvula, se enciende al activar el riego
Botón	Permite iniciar o finalizar riegos manuales
Buzzer	Emite sonido cuando se activa el riego
Pantalla LCD 16x2	Muestra el estado actual del sistema

Funcionamiento general

El sistema puede operar de dos maneras:

- Modo automático: el riego se activa cuando la humedad del suelo es baja y es de noche (ambas condiciones se simulan con potenciómetros).
- Modo manual: el usuario puede activar el riego presionando el botón.

Estados del Sistema:

Estado	Descripción
ESTADO_INIT	Inicio del sistema. Muestra “Iniciando...” y luego “Sistema listo”
ESTADO_IDLE	Espera sin regar. Monitorea sensores y botón
ESTADO_REGANDO_MANUAL	Riego iniciado por botón. Termina por timeout o nueva pulsación
ESTADO_REGANDO_AUTOMATICO	Riego iniciado por condiciones ambientales. Termina por timeout o humedad ok
ESTADO_TERMINANDO_RIEGO	Estado transitorio donde se apaga la válvula, y se vuelve al idle
ESTADO_ERROR_SENSOR	Error si la lectura de sensores no es válida

Condiciones de riego automatico

Condición	Umbral	Fuente
Humedad baja	< 30%	Potenciómetro A0
Oscuridad / noche	Valor analógico < 100	Potenciómetro A3
Tiempo máximo de riego	10 segundos	Constante definida

Condiciones para detener el riego:

En riego manual:

- Timeout alcanzado
- Boton presionado nuevamente

En riego automatico:

- Humedad alcanzo el nivel suficiente

- Timeout alcanzado

Modo Debug / Productivo

El sistema puede operar en dos modos:

- Modo Debug: Muestra en el monitor serie todos los eventos y transiciones de estado.
- Modo Productivo: Silencia todos los Serial.print para optimizar rendimiento.

Esto se controla con la constante:

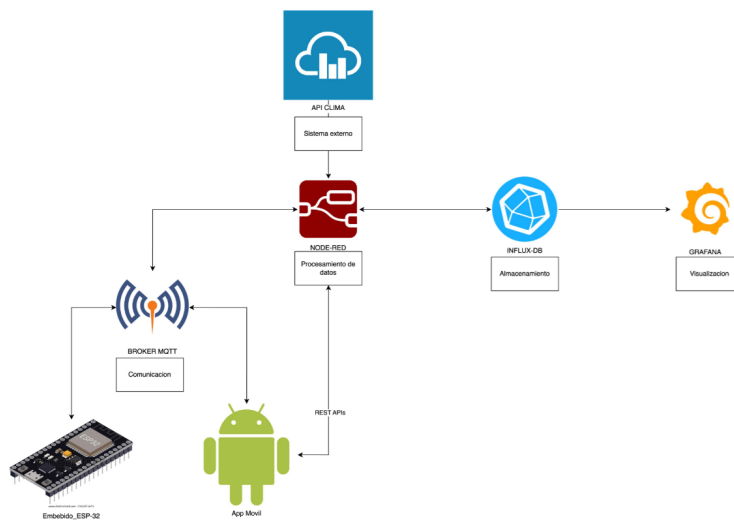
```
#define MODO_DEBUG true // cambiar a false para silenciar logs
```

Mantenimiento / Debugging

- Para simular humedad o luz, simplemente mover los potenciómetros en Wokwi.
- Para verificar riego automático, colocar humedad baja y oscuridad simulada.
- Para riego manual, presionar el botón.

- URL al proyecto: <https://wokwi.com/projects/430232767816133633>

Arquitectura de sistema



Comunicacion con Mosquitto y Broker MQTT

La arquitectura del sistema implementado incluye comunicación inalámbrica basada en el protocolo MQTT, facilitando la integración con plataformas de monitoreo y control remoto como Node-RED y aplicaciones móviles. Para ello, el ESP32 interactúa con un *broker* Mosquitto mediante la publicación y suscripción a distintos tópicos definidos con funciones específicas.

Los tópicos empleados se describen a continuación:

- **sensor/humedad:** El sistema publica en este tópico los valores de humedad del suelo en tiempo real. Esta información permite el monitoreo constante de las condiciones del entorno.
- **sensor/luz:** De forma análoga, se publican los datos del sensor de luminosidad en este tópico, proporcionando así un parámetro adicional para la toma de decisiones sobre el riego automático.

- **estado/riego:** Este tópico refleja el estado actual del sistema de riego. Se publica un valor binario, donde 1 indica que el riego está activo y 0 que está inactivo. Esta información resulta útil para aplicaciones externas que deseen visualizar el estado del sistema.
- **App/riego:** Este canal permite la activación remota del riego mediante una aplicación móvil. Cuando el usuario solicita el riego desde la interfaz Android, se publica un 1 en este tópico a través de Node-RED, el cual reenvía la orden al *broker*.
- **control/riego:** Este tópico es leído por el ESP32 y actúa como mecanismo de control directo. Cuando Node-RED publica un 1 en este canal (desencadenado por una orden previa desde App/riego), el ESP32 interpreta esta señal como una instrucción para activar el sistema de riego

Android

Como parte integral del sistema de riego inteligente, se desarrolló una aplicación Android que permite al usuario:

- Monitorear valores en tiempo real de humedad y luz.
- Visualizar el estado del sistema de riego (encendido o apagado).
- Programar riegos automáticos con lógica recurrente (diaria, cada X días o una sola vez).
- Activar manualmente el riego desde el celular, enviando comandos a través de MQTT.

La app utiliza la librería Paho MQTT para comunicarse con el broker Mosquitto instalado en un servidor cloud, publicando y suscribiéndose a tópicos específicos como sensor/humedad, sensor/luz, estado/riego y control/riego.

El enfoque está en brindar una interfaz simple, intuitiva y funcional, diseñada para usuarios sin conocimientos técnicos, pero que desean controlar su sistema de riego desde cualquier lugar dentro de la red.

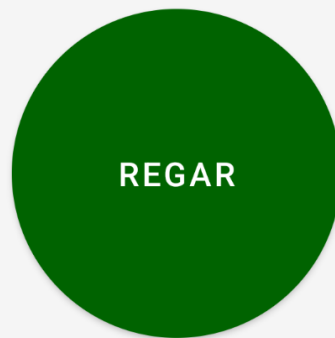
Main Activity

Es la pantalla principal de la aplicación. Brinda acceso rápido al estado actual del sistema y permite el riego manual con un solo toque.

Características:

- Muestra en tiempo real los valores de humedad y luz suscribiéndose a los tópicos sensor/humedad y sensor/luz.
- Incluye un botón central para activar el riego manual. Al presionarlo, se publica un 1 en el tópico App/riego, el cual Node-RED traduce como una orden para el ESP32.
- Refleja el estado actual del riego (activo/inactivo) suscribiéndose al tópico estado/riego.

11:41 ⚙️



🌿 Estado: En reposo

💧 Humedad: 20% ☀️ Luz: 40%

Dashboard Activity

Función:

Visualiza información histórica y en tiempo real del sistema usando gráficos integrados con los datos procesados por Node-RED y almacenados en InfluxDB.

Características:

- Muestra un gráfico de riegos por día
- Se integra con la librería MPAndroidChart para presentar los datos en forma visual y comprensible.
- En versiones futuras se puede expandir para incluir gráficos de humedad promedio o luminosidad a lo largo del tiempo.

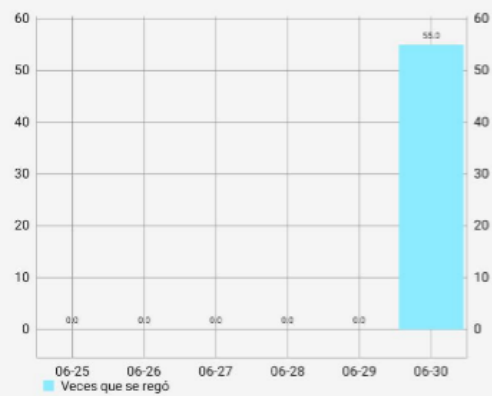
8:44



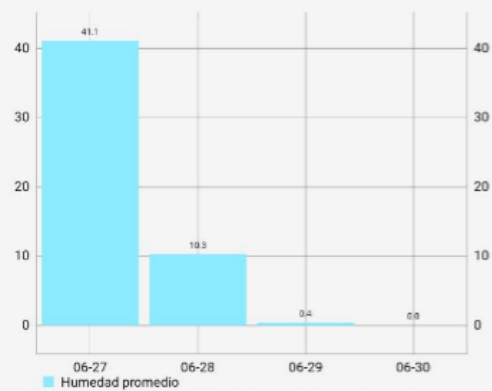
Dashboards



Historial de Riego



Humedad Promedio (5 días)



Programar Riego Activity

Función

Permite a los usuarios programar riegos automáticos sin necesidad de intervención manual.

Características:

- El usuario puede:
 - Seleccionar una fecha y hora de inicio del riego.
 - Elegir si la programación es:
 - Solo una vez.
 - Recurrente diaria.
 - Recurrente cada X días.
- Al confirmar, la app publica el objeto con la programación en formato JSON en el tópico App/programar, que Node-RED interpreta y almacena para su evaluación periódica.
- El sistema valida la coincidencia entre la hora actual y la programación guardada para activar el riego automáticamente.



July
17

Programar Riego



Seleccionar Fecha



Seleccionar Hora



Riego recurrente ☐



PROGRAMAR RIEGO

Visualización con Grafana

Para lograr una supervisión visual del sistema de riego, se integró Grafana como plataforma de dashboards en tiempo real. Esta herramienta permite monitorear valores de sensores, estados de riego y tendencias históricas, ofreciendo una interfaz clara y personalizable.

Arquitectura de integración

1. **ESP32** publica datos de humedad y luz en tópicos MQTT (sensor/humedad, sensor/luz).
2. **Node-RED**, suscripto a estos tópicos, recibe los datos y los reenvía a **InfluxDB**, una base de datos orientada a series temporales.
3. **Grafana** se conecta a InfluxDB como fuente de datos y genera dashboards que muestran:
 - a. Humedad del suelo (promedio diario y valores en tiempo real).
 - b. Luminosidad ambiente.
 - c. Estado del riego (activo/inactivo).
 - d. Historial de eventos de riego.



URL del proyecto en GitHub: https://github.com/UNLAM-SOA/2025_SOA_Q1_L2/tree/main