

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:

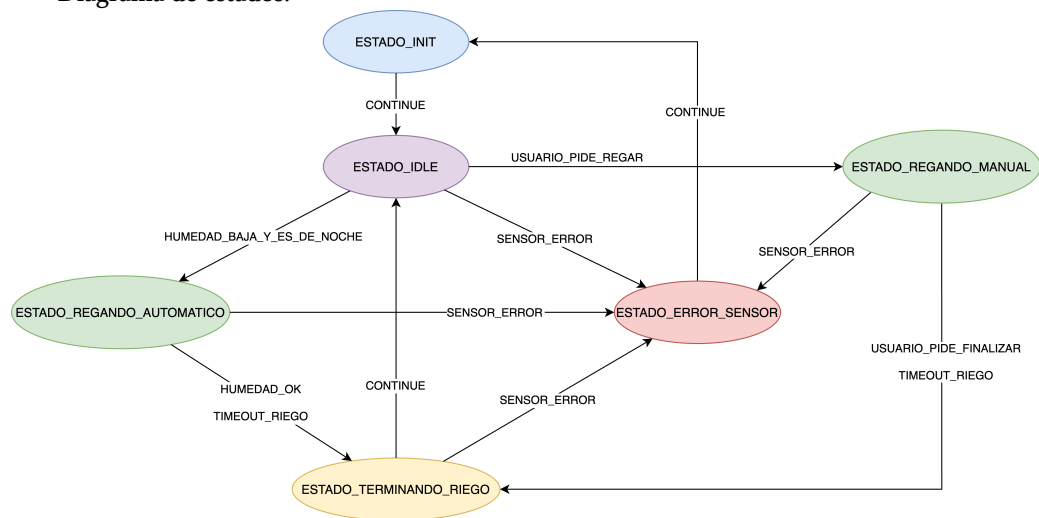
Sabemos que conquistar el mundo no es tarea sencilla, y mucho menos si tienen que perder tiempo regandosus cultivos. Por eso diseñamos este sistema embebido de riego automático, ideal para sus campos de cultivo mutantes.

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 dominacióny 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:

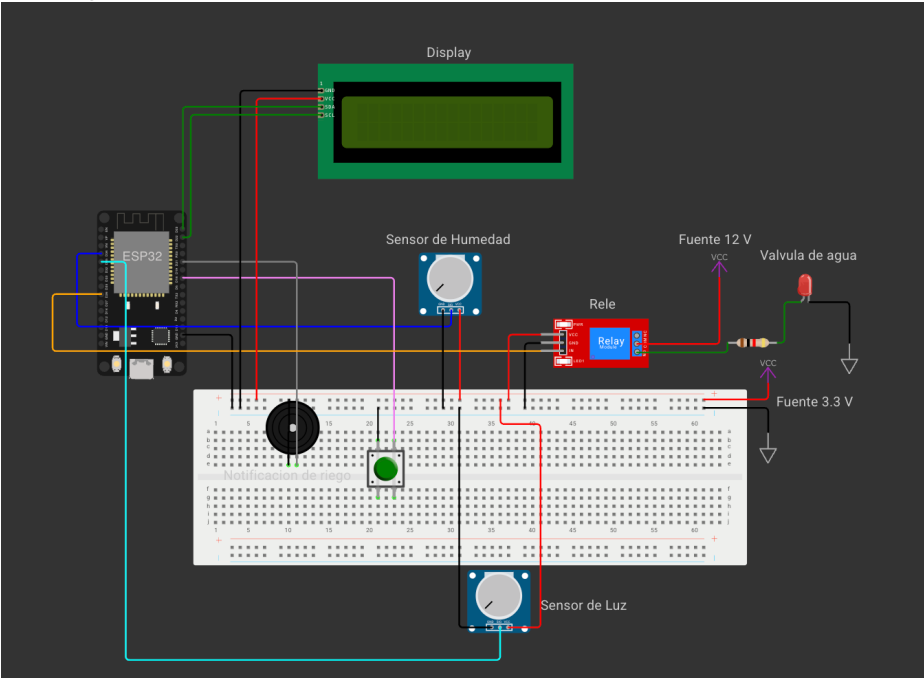


ESTADO	Descripción
ESTADO_INIT	Inicializar el sistema
ESTADO_IDLE	Sistema a la espera de un evento (manual o automático).
ESTADO_REGANDO_MANUAL	El riego activo de forma manual
ESTADO_REGANDO_AUTOMATICO	El riego activo de forma automática por lectura de sensores
ESTADO_TERMINANDO_RIEGO	Se terminó el riego,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

Estado Actual	Evento recibido	Acción	Nuevo Estado
ESTADO_INIT	TIPO_EVENTO_CONTINUE	Setup inicial	ESTADO_IDLE
ESTADO_IDLE	TIPO_EVENTO_USUARIO_PIDE_REGAR	Encender riego manual	ESTADO_REGANDO
ESTADO_IDLE	TIPO_EVENTO_HUMEDAD_BAJA_Y_ES_DE_NOCHE	Encender riego automático	ESTADO_REGANDO
ESTADO_REGANDO_AUTOMATICO	TIPO_EVENTO_TIMEOUT_RIEGO	Apagar bomba	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_EVENTO_USUARIO_PIDE_FINALIZAR	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 esta preñdida. Mostrar error	ESTADO_ERROR_SENSOR

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

- Diagrama de Conexiones del Circuito de Tinkercad:



- Manual de usuario del embebido simulado:

Introduccion:

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	de 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
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/430142015702246401>