

Sistema Embebido de Detección Pasiva de Impactos de Rayo Basado en ESP32

Carlos Elizondo A.

Escuela de Ingeniería Electrónica

Instituto Tecnológico de Costa Rica

Cartago, Costa Rica

Email: alfaquillo@estudiantec.cr

Manuel Garita B.

Escuela de Ingeniería Electrónica

Instituto Tecnológico de Costa Rica

Cartago, Costa Rica

Email: magabretec@estudiantec.cr

Rodrigo Venegas M.

Escuela de Ingeniería Electrónica

Instituto Tecnológico de Costa Rica

Cartago, Costa Rica

Email: rovenegas@estudiante.cr

Resumen—En este trabajo se presenta el diseño e implementación de un sistema embebido para el monitoreo pasivo de descargas atmosféricas asociadas a pararrayos, utilizando tecnologías IoT de bajo costo. A nivel mundial ocurren más de 1.4 mil millones de descargas por año y, en regiones como Costa Rica, se registran hasta 25 descargas por km²/año, lo cual supone un riesgo significativo para infraestructura energética, telecomunicaciones e instalaciones industriales. Los pararrayos tradicionales desvían la descarga a tierra, pero no registran la cantidad ni la ocurrencia temporal de los eventos, lo que limita el mantenimiento predictivo, el análisis de riesgo y la trazabilidad frente a aseguradoras.

Se propone un sistema no invasivo basado en un sensor inductivo pasivo y un microcontrolador ESP32 con FreeRTOS para la detección, conteo y registro de eventos, incorporando telemetría inalámbrica hacia un *dashboard* en ThingsBoard. El prototipo permite detectar impulsos equivalentes a descargas ≥ 1 kA, registrar su marca de tiempo y transmitir los datos mediante MQTT. La validación preliminar se realizó mediante simulación en Wokwi y mediante la síntesis de una imagen de FreeRTOS funcional, aunque no fue posible su verificación en hardware real debido a la ausencia del dispositivo.

Index Terms—descargas atmosféricas, sistemas embebidos, FreeRTOS, IoT industrial, ESP32, sensores pasivos, telemetría inalámbrica

I. INTRODUCCIÓN

Las descargas atmosféricas representan una amenaza relevante para infraestructura eléctrica, telecomunicaciones y sistemas industriales, ocasionando interrupciones de servicio, deterioro progresivo y fallas catastróficas. En zonas tropicales como Costa Rica, donde se registran algunas de las mayores densidades de descargas por km²/año, resulta crítico implementar mecanismos de monitoreo local y trazabilidad de eventos.

Los pararrayos tradicionales capturan y desvían la descarga a tierra, pero no reportan la cantidad ni las características temporales de las descargas. Esta ausencia de registro dificulta el mantenimiento predictivo, limita la capacidad de análisis para aseguradoras y reduce la confiabilidad operativa de sistemas industriales. Aunque existen contadores comerciales, estos suelen ser costosos, no poseen telemetría integrada y requieren instalación invasiva.

Aprovechando el avance de plataformas IoT como el ESP32 y técnicas de adquisición basadas en sensores inductivos pasivos, este proyecto desarrolla un sistema embebido capaz de

detectar el pulso electromagnético inducido por una descarga atmosférica, procesarlo digitalmente, almacenarlo y reportarlo remotamente mediante interfaces inalámbricas. El propósito de este informe es documentar la arquitectura, implementación y validación parcial del sistema, conforme a las actividades requeridas en el proyecto de Sistemas Embebidos con sistemas operativos de tiempo real.

II. METODOLOGÍA Y DESARROLLO

II-A. Análisis del problema y requerimientos

Con base en la investigación presentada en la propuesta de diseño, el sistema debe resolver cuatro desafíos principales: (i) detectar la señal electromagnética inducida por descargas reales, (ii) filtrar ruido de origen industrial o ambiental, (iii) registrar eventos y marcas de tiempo de manera confiable y (iv) transmitir los datos hacia un sistema externo para análisis.

A partir de lo anterior se definieron los siguientes requerimientos funcionales:

- RF001: Capturar impulsos electromagnéticos mediante una bobina inductiva pasiva.
- RF002: Filtrar y limitar la señal analógica evitando la saturación del ADC.
- RF003: Detectar pulsos equivalentes a descargas ≥ 1 kA.
- RF004: Registrar y contar cada evento detectado.
- RF005: Transmitir datos vía WiFi/MQTT hacia una plataforma de monitoreo (ThingsBoard).
- RF006: Registrar eventos localmente en memoria NVS.

Adicionalmente, se consideraron los siguientes requerimientos no funcionales:

- RNF001: Robustez ante ruido electromagnético y condiciones ambientales severas.
- RNF002: Consumo inferior a 200 mA en operación continua.
- RNF003: Tolerancia a impulsos indirectos superiores a 20 kV en el pararrayo.
- RNF004: Aislamiento galvánico completo entre el sensor inductivo y el microcontrolador.

II-B. Vista operacional del sistema

El flujo de operación del sistema se resume en las siguientes etapas:

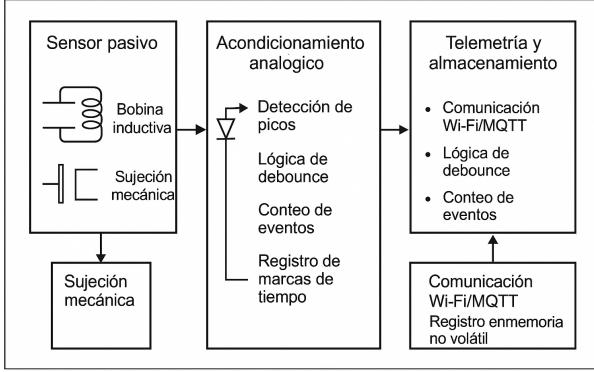


Figura 1. Arquitectura general del sistema propuesto de detección pasiva de rayos basada en ESP32.

1. Una bobina inductiva colocada alrededor del conductor del pararrayo capta el pulso electromagnético generado por la descarga.
2. El pulso se acondiciona mediante un filtro RC, rectificación y un dispositivo TVS que limita el voltaje.
3. El ESP32 detecta un cruce de umbral, ya sea mediante su ADC o mediante una entrada digital con interrupción.
4. El evento se registra con su marca de tiempo y se incrementa el contador acumulado de impactos.
5. El sistema transmite los datos a ThingsBoard mediante MQTT o los almacena localmente en NVS.

II-C. Arquitectura funcional y de hardware

La arquitectura funcional se compone de cuatro bloques principales:

- **Sensor pasivo:** bobina inductiva y sujeción mecánica alrededor del conductor del pararrayo.
- **Acondicionamiento analógico:** rectificador, filtro RC y limitador TVS para proteger la entrada del ESP32.
- **Procesamiento digital:** detección de picos, lógica de debounce, conteo de eventos y registro de marcas de tiempo.
- **Telemetría y almacenamiento:** comunicación Wi-Fi/MQTT y registro en memoria no volátil.

En cuanto a los componentes físicos, el sistema utiliza un módulo ESP32-WROOM, una bobina toroidal de entre 40 y 200 vueltas, una resistencia de 100 kΩ, un diodo rápido 1N4148, un supresor de voltaje transitorio (TVS) modelo SMAJ5.0A, capacitores de 1 nF y 100 nF, y una caja IP65 para instalación en exteriores.

El *software* se basa en el *framework* ESP-IDF, utilizando FreeRTOS, periféricos ADC e interrupciones, y una pila MQTT opcional para telemetría.

II-D. Desarrollo de software y tareas FreeRTOS

Para cumplir con los requerimientos de concurrencia y respuesta en tiempo real se definieron las siguientes tareas en FreeRTOS:

- **TaskSensor:** gestiona la detección de eventos a partir de interrupciones o lecturas periódicas del ADC.
- **TaskCounter:** aplica *debounce*, incrementa el contador de eventos y actualiza las estructuras de datos de registro.
- **TaskTelemetry:** empaqueta y envía los datos al *broker* MQTT, integrándose con el *dashboard* en ThingsBoard.
- **TaskStorage:** realiza el almacenamiento persistente de eventos en memoria NVS.

La síntesis de la imagen del sistema se realizó con el flujo estándar de ESP-IDF, configurando el objetivo con `idf.py set-target esp32` y utilizando `idf.py build` para generar el *firmware*, el *bootloader* y la tabla de particiones. Este proceso cumple con los requerimientos del proyecto en cuanto a la demostración del uso de un sistema operativo de tiempo real embebido.

II-E. Validación mediante simulación

Debido a la falta del dispositivo físico ESP32 durante el desarrollo, la validación se realizó en dos niveles:

1. **Simulación funcional en Wokwi:** se utilizó un botón para simular la llegada de un pulso equivalente a un impacto de rayo. Los datos se registraron y enviaron a ThingsBoard, donde se visualizó una tabla con la marca de tiempo, el identificador del dispositivo y el conteo acumulado de eventos.
2. **Síntesis de la imagen FreeRTOS:** se generó exitosamente la imagen del sistema basada en ESP32 y FreeRTOS, verificando la correcta compilación e integración de las tareas, aunque no se pudo realizar la instalación ni ejecución en hardware real.

III. RESULTADOS Y DISCUSIÓN

Los principales resultados obtenidos pueden resumirse de la siguiente forma:

- Se implementó una arquitectura completa de detección pasiva no invasiva basada en sensores inductivos, que permite instrumentar un pararrayo sin modificar su conductor principal.
- El código de la aplicación operó correctamente en simulación, registrando eventos y enviando telemetría hacia ThingsBoard, lo cual demuestra la viabilidad de integrar el sistema en una plataforma de monitoreo remoto.
- Se evidenció la capacidad de FreeRTOS para gestionar múltiples tareas concurrentes, incluyendo detección, conteo, telemetría y registro, cumpliendo con los objetivos del curso relacionados con sistemas operativos de tiempo real.
- La imagen del sistema operativo embebido se generó correctamente mediante ESP-IDF, integrando las dependencias de FreeRTOS y los módulos de comunicación requeridos.

La limitación principal del trabajo radica en la imposibilidad de completar las etapas de instalación, inicialización y levantamiento de la aplicación sobre hardware real, debido a la ausencia del dispositivo ESP32 durante el periodo de

desarrollo. Esto impidió validar en campo la respuesta del sensor inductivo ante descargas reales o pulsos generados en laboratorio, así como caracterizar el desempeño del acondicionamiento analógico frente a ruido y transientes.

A pesar de esta limitación, la combinación de simulación en Wokwi y la integración con un *dashboard* en ThingsBoard proporciona evidencia suficiente de la corrección lógica del diseño y del cumplimiento de los requerimientos funcionales del sistema en un entorno controlado.

IV. CONCLUSIONES

A partir del desarrollo y resultados obtenidos, se pueden extraer las siguientes conclusiones:

1. Se desarrolló un sistema embebido de detección pasiva de descargas atmosféricas que cumple con los requerimientos funcionales definidos para captura, conteo y registro de eventos asociados a pararrayos.
2. La arquitectura propuesta demuestra viabilidad técnica para habilitar esquemas de mantenimiento predictivo y monitoreo remoto de pararrayos, al proporcionar trazabilidad temporal de los impactos y un canal de telemetría hacia plataformas IoT.
3. La simulación en Wokwi, combinada con la integración en ThingsBoard, confirmó el funcionamiento lógico del sistema y la capacidad de FreeRTOS para gestionar tareas concurrentes de detección, procesamiento y comunicación.
4. La imagen de FreeRTOS sintetizada mediante ESP-IDF valida la integración de los componentes de *software* requeridos; sin embargo, la falta de pruebas en hardware real deja pendiente la caracterización cuantitativa del sistema en condiciones de operación reales.

REFERENCIAS

- [1] IEC 62305-1, “Protección contra rayos – Parte 1: Principios generales,” International Electrotechnical Commission, 2023.
- [2] IEEE Std 142, “IEEE Recommended Practice for Grounding of Industrial and Commercial Power Systems,” IEEE, 2022.
- [3] V. A. Rakov and M. A. Uman, *Lightning: Physics and Effects*. Cambridge University Press, 2020.
- [4] J. López et al., “Low-Cost Sensors for Electromagnetic Surge Detection,” *IEEE Sensors Journal*, vol. 21, no. 5, pp. 1234–1242, 2021.
- [5] Espressif Systems, “ESP32 Technical Reference Manual,” 2023. [En línea]. Disponible en: <https://www.espressif.com>
- [6] NOAA, “Global Lightning Activity,” National Oceanic and Atmospheric Administration, 2024. [En línea].
- [7] CIGEFI, “Estadísticas de descargas eléctricas en Costa Rica,” Centro de Investigaciones Geofísicas, Universidad de Costa Rica, 2023.