

Proyecto Final

Curso de Sistemas Operativos y Laboratorio

Exploración de Sistemas Operativos para Dispositivos
Microcontrolados (ESP32, Raspberry Pi Nano)

Sofia Vanegas Cordoba
Samuel David Montoya Cano

Resumen

Este proyecto tiene como objetivo explorar y evaluar sistemas operativos diseñados para microcontroladores, como RIOT OS, FreeRTOS, Contiki y TinyOS, mediante el desarrollo de una aplicación IoT funcional utilizando plataformas como ESP32 y Raspberry Pi Nano. En el contexto de la rápida expansión del Internet de las Cosas (IoT), entender cómo operan estos sistemas resulta fundamental para implementar soluciones eficientes, seguras y escalables.

Introducción

En la actualidad, el IoT se consolida como un eje central en el desarrollo tecnológico, permitiendo la conexión e interacción de objetos cotidianos a través de internet. Los microcontroladores son el núcleo de estas soluciones, pues proporcionan capacidades de procesamiento y comunicación a dispositivos embebidos. Para facilitar su programación y gestión, existen sistemas operativos livianos especialmente diseñados para estas plataformas, los cuales permiten una mejor abstracción del hardware, manejo eficiente de tareas y conectividad.

¿Cuál es la necesidad y/o problema que aborda el desafío seleccionado?

El principal problema radica en la necesidad de implementar soluciones IoT confiables, flexibles y de bajo consumo energético. Esto requiere elegir sistemas operativos adecuados para microcontroladores, con capacidades multitarea, comunicación eficiente y facilidad de desarrollo.

¿Por qué es importante el desarrollo de este desafío en el contexto tecnológico actual?

Debido al crecimiento exponencial de dispositivos conectados, la selección adecuada de software para microcontroladores impacta directamente en la viabilidad y escalabilidad de aplicaciones IoT. Explorar estos sistemas operativos brinda una comprensión práctica de cómo funcionan los sistemas embebidos y su relación con la arquitectura de los sistemas operativos tradicionales.

Antecedentes o marco teórico

Para comprender este proyecto, es necesario familiarizarse con los siguientes conceptos:

- **Microcontroladores:** dispositivos con CPU, memoria y periféricos integrados en un solo chip (por ejemplo, ESP32 o RaspBerry Pi).
- **Sistemas Operativos en Tiempo Real (RTOS):** como FreeRTOS, que permiten la ejecución concurrente de tareas con restricciones de tiempo.
- **Sistemas Operativos para IoT:** como RIOT OS o Contiki, que están diseñados para dispositivos con recursos limitados y soportan conectividad (IPv6, MQTT, CoAP).
- **Tareas, planificación y concurrencia:** conceptos fundamentales en la teoría de sistemas operativos, esenciales para entender cómo se gestionan múltiples procesos en estos sistemas ligeros.

Relación con los temas del curso

Este proyecto integra conocimientos teóricos y prácticos abordados en la asignatura, como:

- Arquitectura de sistemas operativos
- Gestión de procesos e hilos
- Planificadores y multitarea
- Sistemas embebidos
- Interfaces de programación y drivers de hardware

Objetivos

Objetivo general

Desarrollar una aplicación IoT utilizando un microcontrolador (ESP32 o Raspberry Pi Nano) que permita probar las funcionalidades de un sistema operativo para microcontroladores como RIOT OS o FreeRTOS.

Objetivos específicos

- Investigar y comparar sistemas operativos adecuados para microcontroladores.
- Instalar y configurar el sistema operativo seleccionado en el dispositivo.
- Diseñar y programar una aplicación básica que involucre lectura de sensores y envío de datos.
- Evaluar el rendimiento y las características del sistema operativo implementado.

Metodología

Herramientas

- Placas de desarrollo: ESP32, Raspberry Pi Pico/Nano
- Sistemas operativos: RIOT OS, FreeRTOS, Contiki, TinyOS
- Lenguajes: C, C++, Python (según OS)
- IDEs: Arduino IDE, PlatformIO, Docker (para RIOT)
- Herramientas de red: Wireshark, MQTT broker, Node-RED

Actividades

1. **Revisión bibliográfica y técnica** sobre sistemas operativos para IoT.
2. **Selección del sistema operativo** más adecuado según los recursos disponibles.
3. **Instalación y configuración del entorno de desarrollo** (IDE, toolchains, drivers).
4. **Desarrollo de la aplicación IoT** (ej: medición de temperatura y envío por MQTT).
5. **Pruebas de funcionamiento y análisis de resultados.**
6. **Documentación del proyecto y presentación final.**

Cronograma

Diagrama de Gantt (en semanas)

Actividad	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
Investigación teórica	X					
Selección del sistema operativo	X	X				
Instalación y configuración del entorno		X	X			
Desarrollo de la aplicación			X	X		
Pruebas y ajustes				X	X	
Documentación y presentación					X	X

Referencias

- Adafruit. (s.f.). *Docker + ESP32-S3 + Linux*.
<https://learn.adafruit.com/docker-esp32-s3-linux>
- IoT Design Pro. (s.f.). *Setting up RIOT OS on Arduino*.
<https://iotdesignpro.com/projects/setting-up-riot-os-on-arduino>
- Hackster. (s.f.). *RIOT OS Projects*.
<https://www.hackster.io/riot-os/products/riot>
- Hackaday. (2022). *It's Linux on an ESP32*.
<https://hackaday.com/2022/07/14/its-linux-on-an-esp32>
- RIOT OS. (s.f.). <https://www.riot-os.org/>