

Tiempo Real en Sistemas Empotrados

Eñaut Fernandez y Unai Urgoiti

efernandez228@ikasle.ehu.eus

uurgoiti001@ikasle.ehu.eus

12/01/2025

1. Descripción del proyecto

Este sistema tiene como objetivo el uso de un sistema operativo en tiempo real para la administración de dispositivos, tareas y conexiones múltiples. Para ello se ha desarrollado un sistema pensado en una aplicación de la vida cotidiana, que podría darse en casa o de igual manera en un ámbito industrial.

Este sistema trata acerca de el control de relés desde un cliente diseñado para el usuario. En este cliente se podrá configurar los diferentes relés según el usuario decida, permitiéndole a el usuario configurar sin necesidad de estar conectados físicamente mediante un cable a el sistema del usuario.

2. Análisis Técnico

Este sistema esta conformado por varias partes. La primera de todas, los esp32, que harán uso de freeRTOS para gestionar dos tareas, la configuración de los relés y la recepción y envío de comandos para la configuración de los mismos.

El cliente operado por el usuario permitirá el envío de comandos de manera inalámbrica, siendo este gestionado mediante varias tareas en las que se incluyen el envío de datos a el servidor inalámbrico y la recogida de datos del usuario.

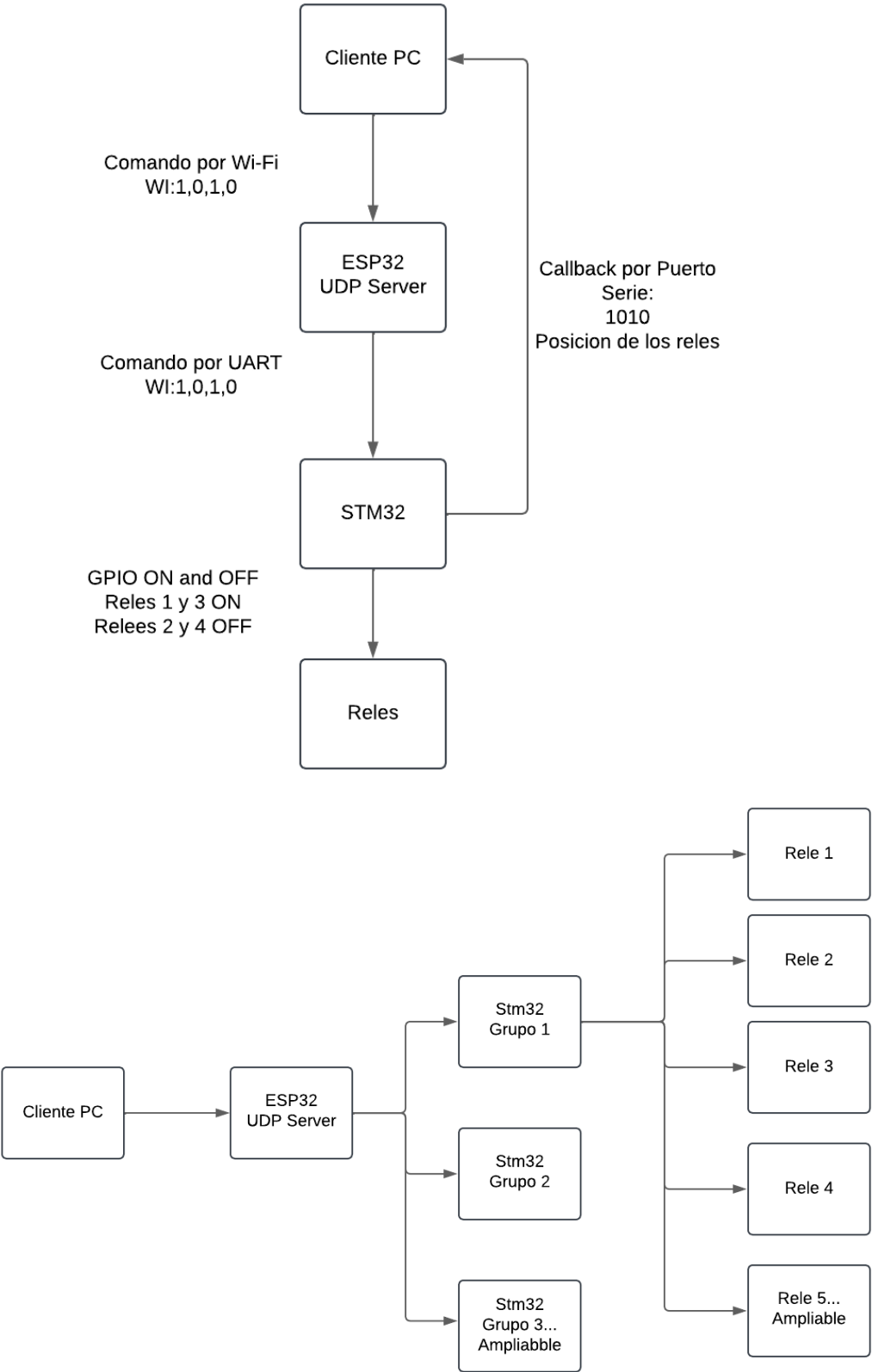
Este servidor inalámbrico estará alojado dentro de un esp32, el cual también hará uso de freeRTOS para administrar las diferentes tareas. Entre estas se incluyen, la recogida de comandos mediante Wi-Fi, el procesamiento de estos datos y el envío de estos mismos mediante diferentes tipos de conexión, específicamente siendo estos I2C, SPI e UART. Lamentablemente, debido a errores que alcanzan fuera de nuestra comprensión y diversos bugs tan solo se ha podido establecer correctamente dentro del plazo de tiempo de entrega de este proyecto la ultima conectividad, por UART.

La modularidad del sistema permite la conexión de tantos relés como se desee. Con muy pocos cambios, se podrían implementar una cantidad de relés considerables de dos maneras.

1. Añadiendo mas relés en cada stm32: Esta opción tan solo requeriría de la modificación de los pines y su asignación, junto con la modificación del string con la configuración de cada relé.

2. Añadiendo mas stm32: El envío desde el “maestro”(Técnicamente no es un maestro por que no se conecta por I2C o SPI pero actúa como tal) se puede modificar para añadir mas conexiones, tan solo haciendo falta añadir mas grupos.

3. FSM Sistema

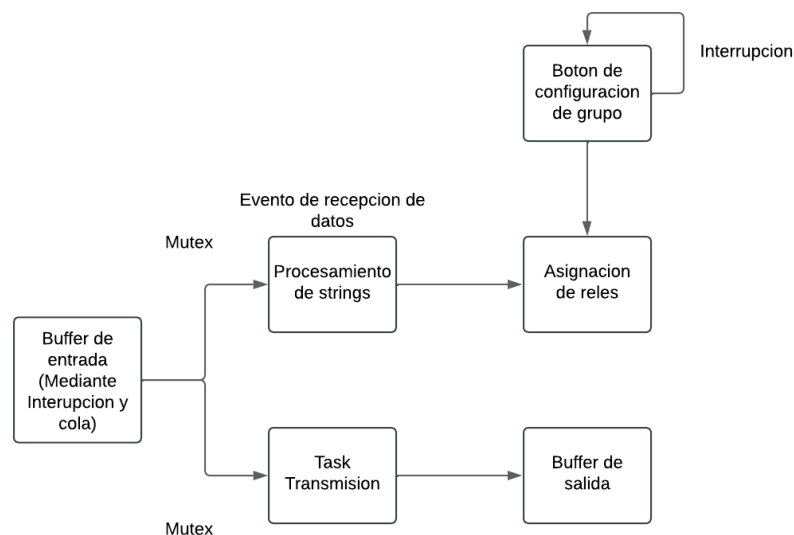


4. Descripción de las tareas, sincronización y comunicación

El firmware de la placa stm32f411REx consta de un sistema operativo, FREERTOS, en el cual hay dos tareas, una para recibir la información por UART1 y otra para mandar la información por UART2. Para la comunicación entre las dos tareas, se añade un buffer en el que se van metiendo los datos recibidos. Pero para garantizar que esos datos lleguen bien y las dos tareas tengan los mismos datos, se añade un mutex que se activará cuando una tarea tenga que coger o poner un dato en el buffer. Además, también se añade un evento para que una tarea le haga saber a la otra que están listos los datos.

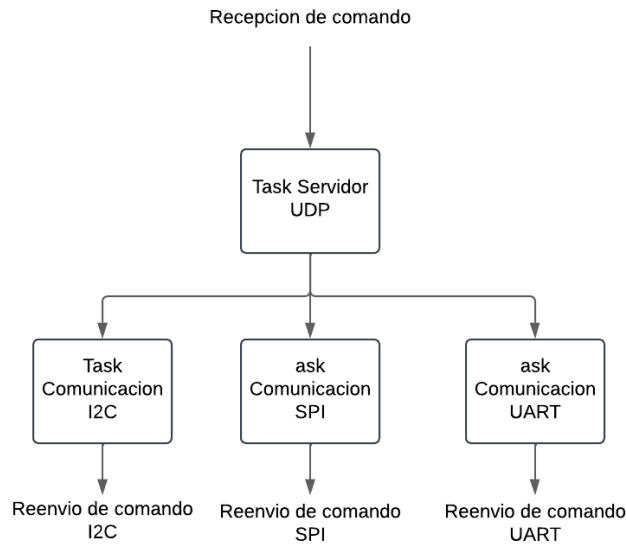
Para la comunicación por UART entre la placa y el programa principal se manda un vector de cinco caracteres en el que los primeros cuatro son un 1 o un 0 dependiendo si el relé está encendido o apagado y el ultimo carácter es un '\0' para saber donde se terminan los datos y empieza otro nuevo.

Por ejemplo: [W2:1,0,1,1] indicaría al grupo 2 que los relés 1, 3 y 4 deben activarse y el 2 desactivarse



Por otro lado, el firmware de la placa esp32 también esta formado por el sistema operativo freeRTOS, incluyendo las siguientes tareas:

- Servidor UDP: Se conecta a un punto Wi-Fi y se inicia un servidor UDP mediante sockets donde se recibirán comandos desde cualquier ordenador que especifique su IP. Se copia el buffer recibido a el buffer a enviar.
- Tarea de conexión I2C: Gestiona la comunicación y el reenvío de comandos por I2C.
- Tarea de conexión SPI: Gestiona la comunicación y el reenvío de comandos por SPI.
- Tarea de conexión UART: Gestiona la comunicación y el reenvío de comandos por UART.

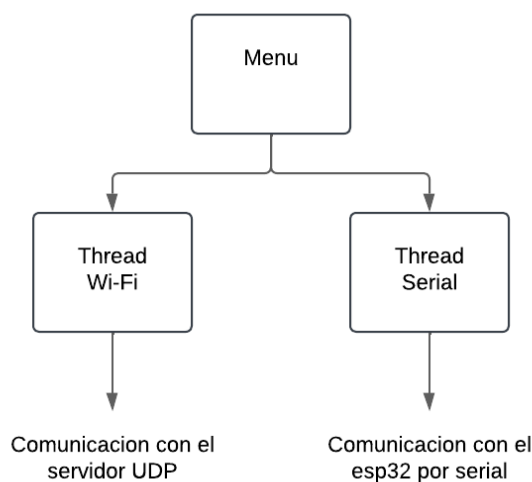


En el lado del cliente del usuario, tal como se ha explicado antes, se encuentra un menú donde el usuario podrá seleccionar la opción que desee.

1. Conexion por wifi (G1)
2. Conexion por wifi (G2)
3. Conexion por Serial
4. Salir

Los dos primeros crearan un thread cada uno donde se primero se configuraran los relés y después se gestionarán las comunicaciones con los dispositivos correspondientes.

Adicionalmente, también se crea un archivo log con los comandos utilizados en cada momento con fecha.



Link al Repositorio: <https://github.com/UnaiUrgoiti/TiempoReal>