

Monitorização GPS e Alertas usando o ESP32

Arquiteturas para Sistemas Embutidos

João Castanheira, 97512
Nuno Fahla, 97631



universidade
de aveiro

Índice

01

Ideia

02

Componentes

03

Desenvolvimento

04

Problemas
Encontrados

05

Demo

06

Trabalho Futuro

Motivação & Ideia

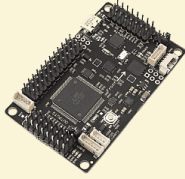
Motivação

- Mini-projeto inserido na cadeira de ASE.
- Requer utilização do ESP32, sensores e desenvolvimento de programas que utilizem as várias capacidades do ESP32.

Ideia

- Sistema de monitorização GPS para animais com alerta caso estes saiam de uma área delimitada.
- Monitorização executada através de dashboard remoto, controlo do microprocessador através de ligação por terminal.

Componentes



ESP32-DEVKITC

- Microcontrolador com comunicação UART
- Capacidades de baixo consumo
- Ligação Wi-Fi
- Sistema de ficheiros interno (Spiffs)



Módulo pMod GPS

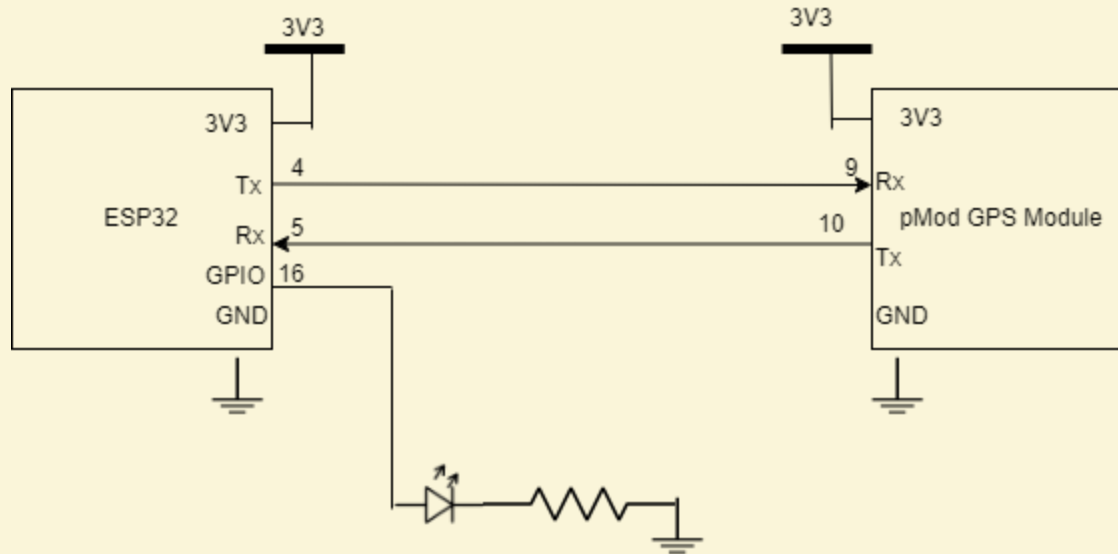
- Módulo captador de sinal GPS
- Comunicação UART
- Voltagem compatível com o ESP32



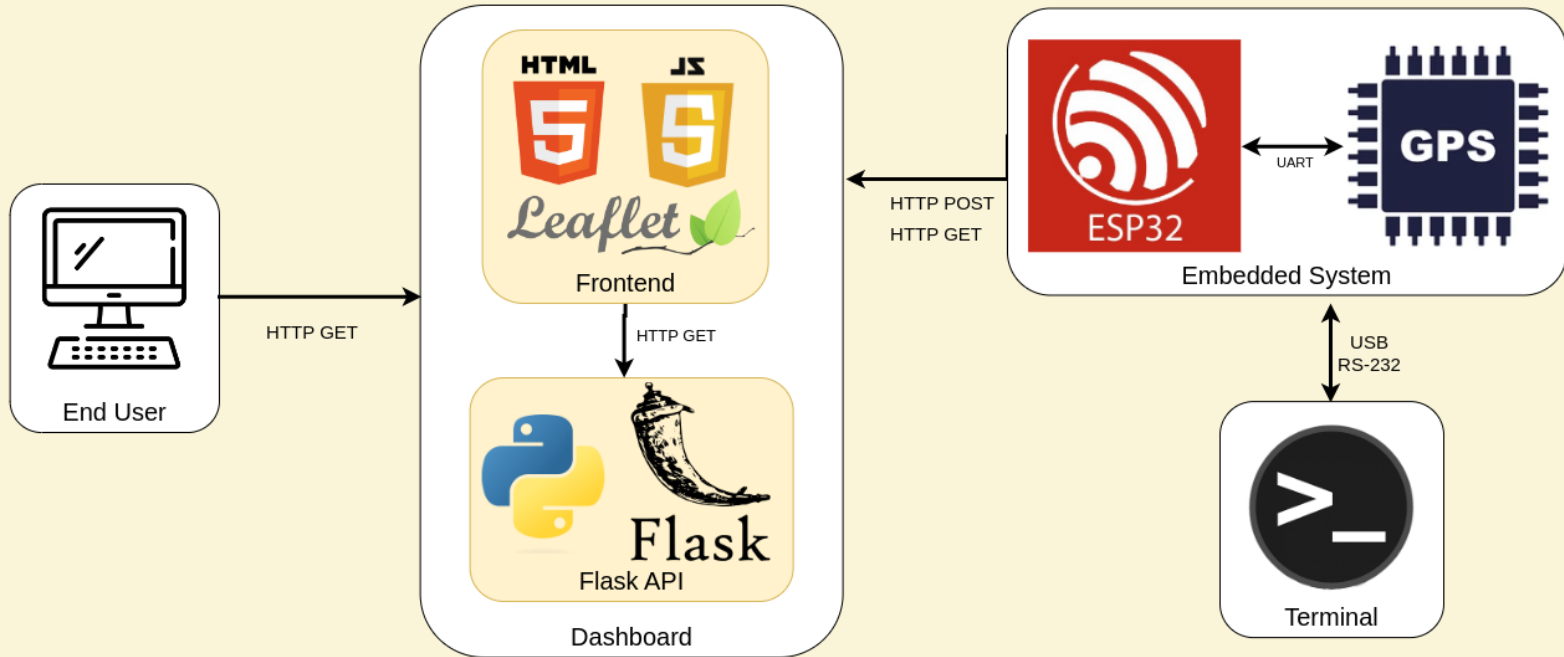
LED

- Usado como atuador para dar um display visual acerca da localização do utilizador em relação à área delimitadora

Componentes - Diagrama



Comunicações - Diagrama



Desenvolvimento

Organização do código

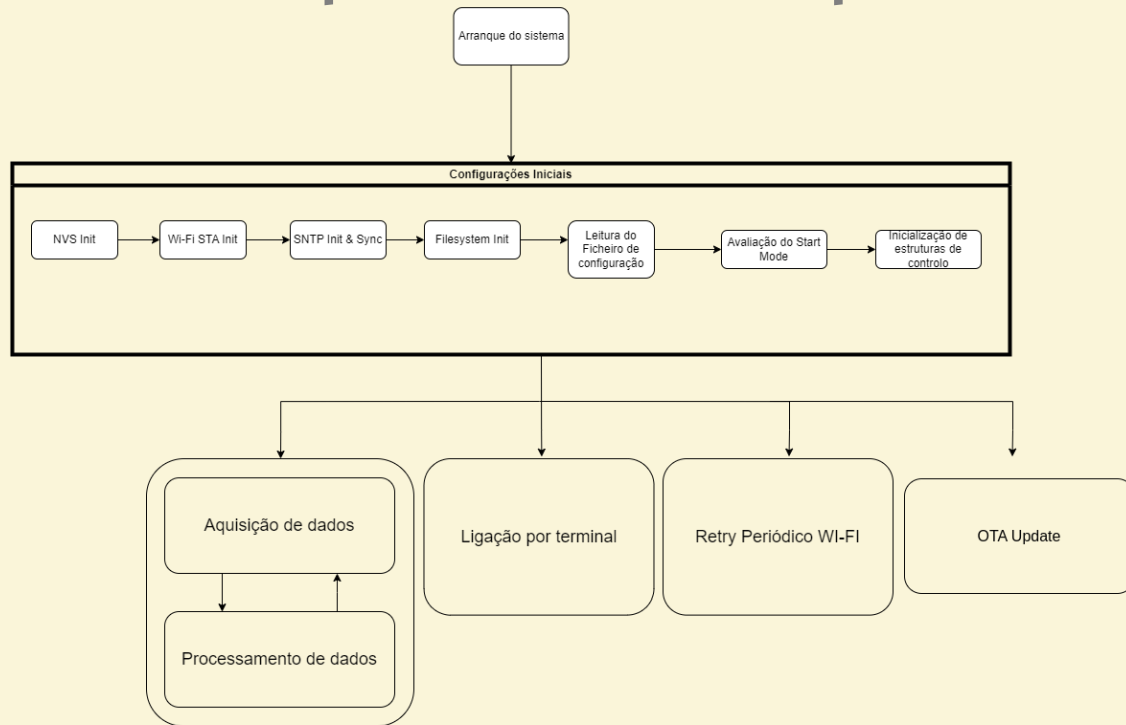
- GPS_project.c -> ficheiro principal que invoca funções e é dependente dos ficheiros seguintes
- utils.c & utils.h -> contém o código das funções e das tarefas utilizadas
- pMod.c & pMod.h -> contém o código referente à utilização do pMod

Requisitos Implementados

- Utilização do ESP32 como base do sistema
- Apresentação dos dados num dashboard remoto (conexão Wi-Fi).
- Robustez perante a perda de conexão Wi-Fi.
- Armazenamento de dados localmente através do sistema Spiffs.
- Utilização dos modos de baixo consumo energético
- Utilização de um led como atuador(controlo)

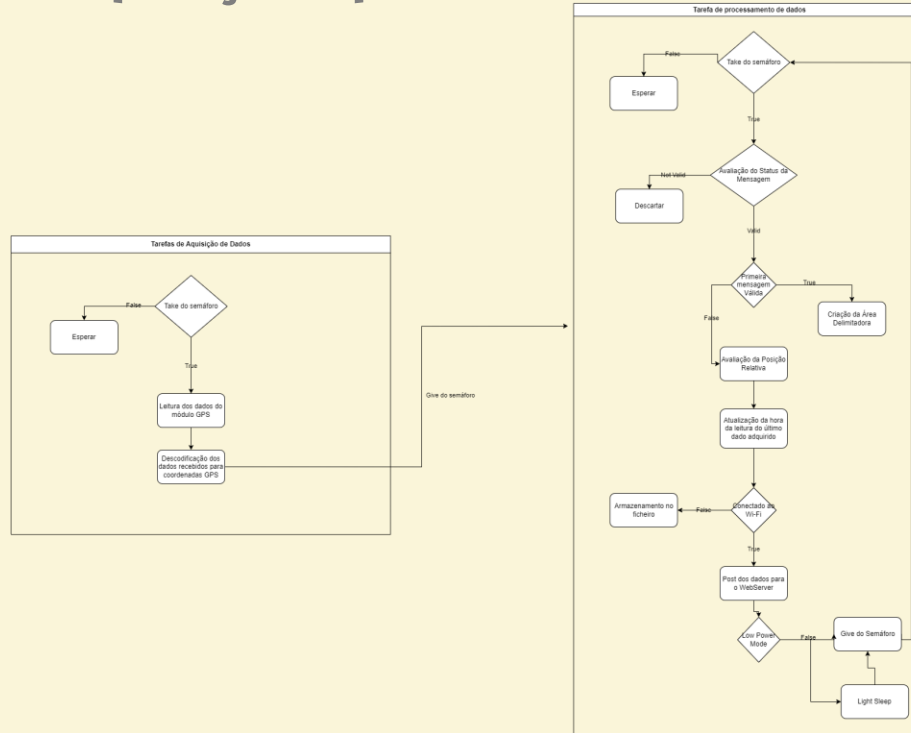
Software – Diagrama

Top Level and Main Loop



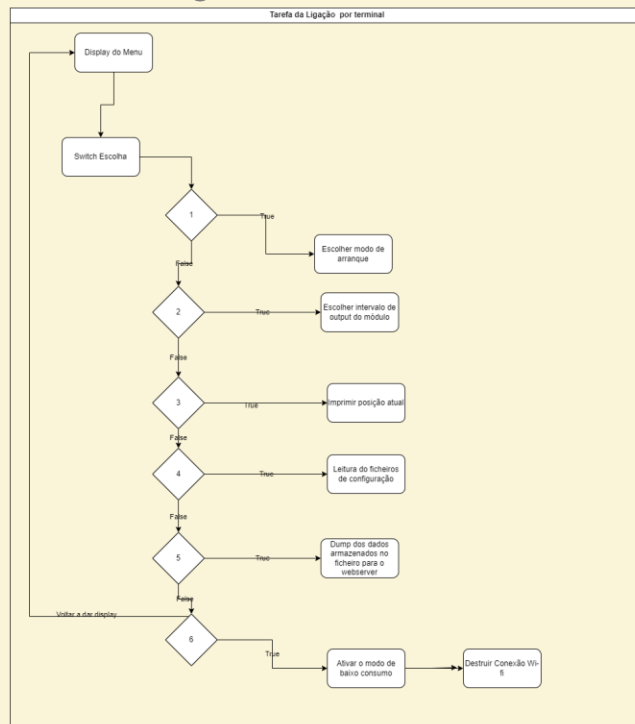
Software - Diagrama

Aquisição e processamento de dados



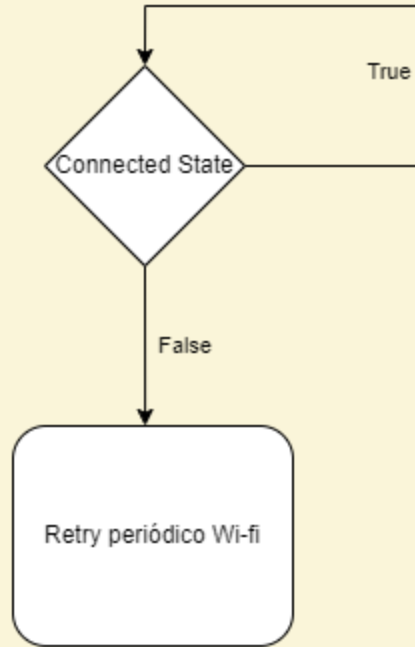
Software - Diagrama

Ligação por terminal



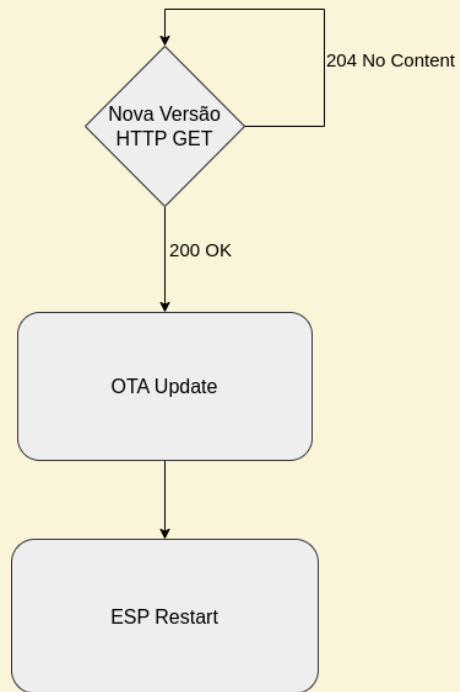
Software - Diagrama

Retry Periódico Wi-Fi



Software - Diagrama

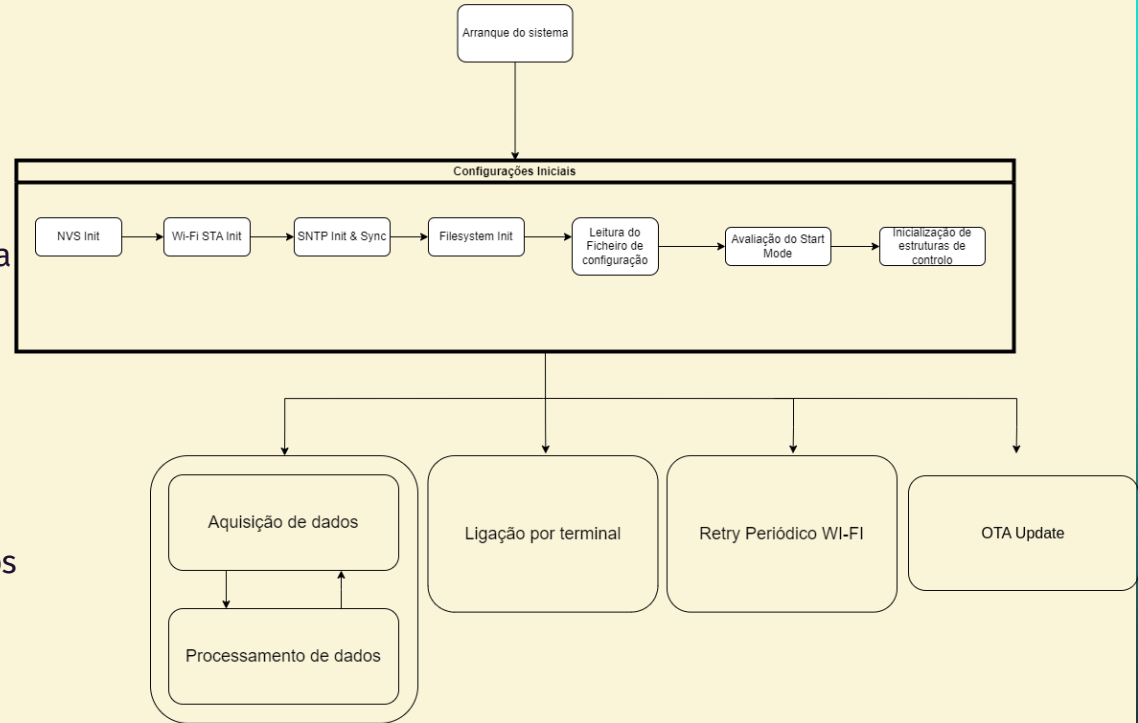
OTA Update



Desenvolvimento

Funcionamento Geral

- O ESP arranca com as configurações guardadas num ficheiro (intervalo de output do módulo, baudrate e hora da última posição registada)
- É iniciada uma tarefa que vai adquirir os dados periodicamente do módulo GPS.
- Um semáforo indica que os dados estão prontos para serem enviados
- Uma tarefa em paralelo permite o controlo do módulo e do ESP32 através do terminal



Desenvolvimento

Funcionamento de cada tarefa

Estado Inicial

- O ESP configura as UARTs para comunicação por terminal e para comunicar com o módulo GPS
- O ESP configura-se como station Wi-Fi e tenta conectar-se à rede Wi-fi definida.
- Uso do protocol SNTP para obter a hora atual e comparar com a última medida, de maneira a escolher o modo de arranque.

Tarefa de Aquisição de Dados GPS

- Verifica a existência de dados no buffer Rx do ESP.
- Verifica a receção de acknowledges.
- Dá update do ficheiro de configuração.
- Descodifica os pacotes RMC, de maneira a obter dados GPS

Tarefa de ligação por terminal

- Permite ao utilizador controlar e aceder a dados através do terminal
- Permite escolher o modo de arranque do pMod e o interval de output do mesmo
- Permite ao utilizador ver no terminal a posição atual e o output do ficheiro de configuração
- Ativar o modo de baixo consumo

Desenvolvimento

Funcionamento de cada tarefa

Tarefa de processamento de dados

- Responsável por avaliar o estado de conexão Wi-Fi
- Dependente do estado da conexão, executa requests POST para o webserver ou armazena no sistema de ficheiros.
- Cria a área delimitadora e avalia a posição relativa a essa área
- Quando em modo de baixo consumo, adormece o ESP32 até à próxima execução da tarefa de aquisição de dados GPS.

Tarefa de updates OTA

- Periodicamente faz um HTTP GET ao webserver com a versão atual do programa nos headers
- Se a versão for igual ou maior que a versão do webserver, retorna 204 No Content. Senão, retorna 200 OK, o webserver responde com o novo ficheiro Bin, o ESP lê os bytes e escreve-os na partição
- Reinicia o ESP a partir de uma das partições OTA, com o novo código
- Útil caso existam alterações no funcionamento do webserver, para alterar as configurações do sistema, complementar ou alterar os tipos de localização (RMC, GGA, etc). Tudo isto sem resgatar o hardware.

Problemas Encontrados

Familiarização com os pacotes NMEA

- Leitura de datasheets e investigação sobre os mesmos
- Aproveitamento de código fornecido pela Waveshare, para a formatação dos pacotes (Cálculo do Checksum)
- Uso constante do osciloscópio 😊

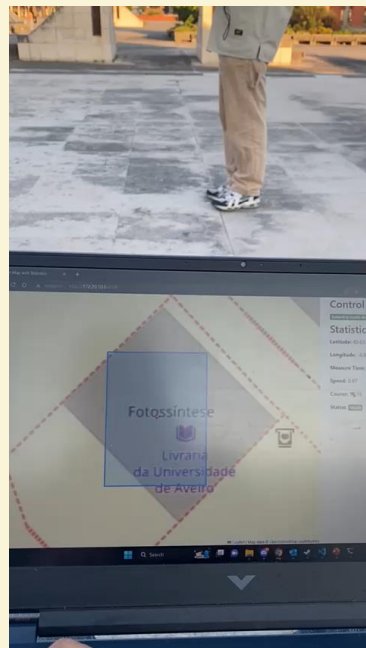
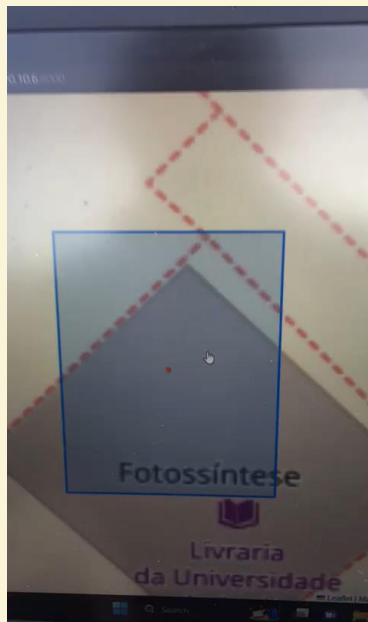
Convivência entre o modo de baixo consumo e a ligação Wi-Fi

- Modo de baixo consumo destrói a conexão Wi-Fi, de maneira a permitir o light sleep
- Ao desativar o modo de baixo consumo, reativa-se as conexões wi-fi.

Persistência das configurações

- Definição de partições
- Implementação do sistema de ficheiros spiffs.

Demo



Trabalho Futuro

Melhoria da precisão de localização

Utilização de um conjunto
de dados (GGA,RMC,etc...)
para melhorar a precisão
da localização

Checklist

Requisitos Obrigatórios

- Utilizar o ESP32DevKitC como base do embedded system; ✓
- Cada grupo deve selecionar um sensor para ligar ao kit ESP32DevKitC e propor a sua aquisição até 14/mar (Inicialmente, o módulo L76X (não funcionou), atualmente módulo pMod) ✓
- A aplicação a executar no kit ESP32DevKitC deve ser desenvolvida em C/C++ e tirar partido do FreeRTOS; ✓
- Devem ser explorados os periféricos do ESP32 que fizerem sentido no contexto do projeto, incluindo aspetos de interrupções e DMA; (não se aplicava) ✗
- Os dados recolhidos do sensor e processados no ESP32 devem ser apresentados num dashboard remoto, sendo para tal necessária conectividade de rede (WiFi / BT); ✓
- Deve ser disponibilizada uma ligação por Terminal; independente do dashboard remoto; ✓
- Devem ser exploradas as várias funcionalidades das ferramentas de desenvolvimento, incluindo debug ✓

Checklist

Requisitos Adicionais

- Podem ser explorados os modos de baixo consumo energético do ESP32; (Light Sleep) ✓
- Podem ser suportadas atualizações remotas (Over-the-Air) do sistema; ✓
- Pode ser incluído algum tipo de atuador cuja utilização faça sentido com o sensor usado (de forma a criar um loop de controlo; ou que seja controlado através do dashboard); ✓
- Pode ser suportado um sistema de ficheiros para armazenar dados localmente. ✓

Baseline

Referências

- Utilização da função de envio de comandos NMEA do exemplo do módulo L76X e utilização do datasheet seguinte: https://www.waveshare.com/w/upload/5/56/Quectel_L76_Series_GNSS_Protocol_Specification_V3.3.pdf
- Aproveitamento e inspiração dos exemplos seguintes:
 - frameworks\esp-idf-v5.0\examples\wifi\getting_started\station\main\station_example_main
 - frameworks\esp-idf-v5.0\examples\protocols\http_request\main\http_request_example_main
 - frameworks\esp-idf-v5.0\examples\storage\spiffs\main\spiffs_example_main

Contribuições

- João Castanheira – 50 %
- Nuno Fahla – 50 %

Fim