

Código exemplo: KMI-ESP32_Format_LittleFS_flash_info.ino

Funcionalidades do Código: Gerenciamento da Memória Flash com LittleFS no ESP32

Este código demonstra, de forma interativa e educativa, como trabalhar com o sistema de arquivos **LittleFS** no microcontrolador ESP32. Cada função foi criada para ilustrar uma tarefa comum relacionada à manipulação de arquivos e análise do uso da memória flash integrada.

1. Montagem do Sistema de Arquivos (LittleFS)

- Verifica se o LittleFS foi inicializado corretamente.
- Função reutilizável (`montarLittleFS()`) para garantir que o sistema esteja pronto antes de qualquer operação.

2. Formatação Segura da Memória Flash

- Permite ao usuário **escolher se deseja apagar todo o conteúdo** do LittleFS.
- Inclui **tempo limite de 10 segundos** para decisão via Serial Monitor.
- Exibe alertas claros sobre a perda irreversível de dados.
- Usa `esp_task_wdt_reset()` para evitar travamentos durante a formatação.

3. Análise de Uso da Memória Flash

- Exibe:
 - **Espaço total** disponível no LittleFS.
 - **Espaço usado**.
 - **Porcentagem de uso** (com duas casas decimais).
- Ideal para monitoramento de recursos em projetos reais.

4. Cálculo Detalhado do Uso da Memória por Arquivos

- Percorre todos os arquivos no diretório raiz.
- Soma o tamanho de cada arquivo para calcular **espaço usado e livre**.
- Mostra resultados em bytes, com clareza.

5. Criação de Arquivo de Exemplo

- Cria um arquivo chamado Dados.txt.
 - Escreve conteúdo de exemplo com múltiplas linhas.
 - Demonstra o uso do modo de escrita ("w").
6. 📁 **Listagem de Arquivos no Sistema de Arquivos**
- Lista todos os arquivos no diretório raiz (/).
 - Mostra o **nome** e o **tamanho** (em bytes) de cada arquivo.
 - Funciona recursivamente (pode ser adaptada para subdiretórios).
7. 🔍 **Leitura e Exibição do Conteúdo de um Arquivo**
- Abre Dados.txt no modo leitura ("r").
 - Exibe todo o conteúdo linha por linha no Serial Monitor.
 - Inclui tratamento de erro para arquivos inexistentes.
8. 🗑️ **Exclusão Segura de Arquivo**
- Verifica se o arquivo existe antes de tentar excluí-lo.
 - Usa LittleFS.remove() para apagar o arquivo.
 - Confirma sucesso ou falha da operação.
9. ❌ **Teste de Arquivo Inexistente**
- Demonstra como verificar a **inexistência** de um arquivo com LittleFS.exists().
 - Útil para evitar erros em leituras ou gravações.
10. 🐛 **Função de Depuração (Debug)**
- Lê e exibe o conteúdo de **todos os arquivos** presentes no sistema.
 - Ajuda a depurar o estado atual da memória flash durante o desenvolvimento.

🔧 Fluxo Automático no setup()

O código executa automaticamente uma **sequência didática de operações**:

1. Pergunta se deseja formatar.
2. Analisa o uso da memória.
3. Cria um arquivo.

4. Lista arquivos.
5. Lê o conteúdo do arquivo.
6. Exclui o arquivo.
7. Confirma que o arquivo foi removido.
8. Exibe o estado final (sem arquivos).

Nota: O loop() permanece vazio, pois o objetivo é demonstrar funcionalidades únicas ao iniciar o ESP32.

Para que serve esse código?

- **Aprendizado:** Ideal para quem está começando com armazenamento local no ESP32.
 - **Testes:** Permite validar se o LittleFS está funcionando corretamente no seu hardware.
 - **Base para projetos:** Pode ser adaptado para salvar logs, configurações, sensores, etc.
-

Requisitos

- ESP32 com suporte a LittleFS (configurado no Arduino IDE).
- Biblioteca LittleFS.h instalada.
- Configurar o Monitor Serial para 115200 bps para interação.