

Sistema Integrado para Leitura de ADC via ESP32 com Servidor Flask e Interface Web

Documento de Documentação Técnica

Cairo Alberto

<https://github.com/CairoAlberto123>

Sumário

1 Introdução

Este documento tem como objetivo descrever detalhadamente o funcionamento e a implementação do sistema integrado para leitura de ADC utilizando ESP32, servidor Flask e interface web. A documentação segue as normas da ABNT, apresentando referências, descrições dos componentes e manuais de funcionamento.

2 Firmware para ESP32

2.1 Descrição Geral

O firmware desenvolvido para o ESP32 realiza a leitura do ADC e envia os valores lidos via comunicação serial USB. O código foi implementado na IDE do Arduino, com estrutura modular e comentários que facilitam a manutenção e a extensão do sistema.

2.2 Conexões

- **ADC:** Conectar os sinais analógicos (+ e -) conforme a especificação do sensor utilizado. **Exemplo:** Conectar os sensores ao pino ADC1, especificamente no GPIO34, conforme utilizado no firmware.
- **USB:** O conector USB-C é utilizado para programação e comunicação serial.

2.3 Manual de Funcionamento

O firmware inicia a comunicação serial a 115200 bps, realiza leituras periódicas do ADC (utilizando o pino GPIO34) e transmite os valores lidos. Funções específicas para inicialização e leitura do ADC foram implementadas, permitindo fácil adaptação a diferentes sensores ou requisitos de projeto.

3 Servidor Flask em Python

3.1 Descrição Geral

O servidor desenvolvido em Python utiliza o framework Flask para monitorar a porta serial e coletar os dados enviados pelo ESP32. Além da leitura contínua, o servidor implementa filtros (passa-baixa e passa-alta), permite a gravação dos dados e gera um vídeo com os gráficos para posterior download.

3.2 Bibliotecas Utilizadas

As principais bibliotecas utilizadas são:

- **Flask** - Para o desenvolvimento do servidor web;
- **PySerial** - Para a comunicação serial com o ESP32;
- **NumPy** - Para o processamento numérico dos dados;
- **Matplotlib** - Para a geração de gráficos;

- **OpenCV** - Para a criação de vídeos a partir dos gráficos.

3.3 Arquivo requirements.txt

```
Flask  
pyserial  
numpy  
matplotlib  
opencv-python
```

3.4 Manual de Funcionamento

O servidor realiza a leitura contínua dos dados via porta serial, aplica os filtros configurados e envia os dados para a interface web para visualização em tempo real. A funcionalidade de gravação permite salvar os dados brutos em um arquivo de texto e gerar um vídeo com o gráfico dos dados capturados.

4 Interface Web

4.1 Descrição Geral

A interface web foi desenvolvida com HTML, CSS e JavaScript, proporcionando:

- Seleção da porta serial para conexão com o ESP32;
- Configuração dos filtros (passa-baixa e passa-alta);
- Visualização dos dados em tempo real através de um gráfico dinâmico;
- Controle de gravação dos dados, com indicadores visuais e temporizador.

4.2 Manual de Uso

O usuário pode selecionar a porta serial desejada, configurar os filtros e iniciar/parar a gravação dos dados. Os dados são atualizados dinamicamente no gráfico, e ao finalizar a gravação, os arquivos de dados e vídeo ficam disponíveis para download.

5 Conclusão

O sistema integrado apresentado permite a captura, processamento e visualização dos dados do ADC de forma modular e expansível. A documentação técnica e o código comentado facilitam a manutenção e futuras atualizações, atendendo aos requisitos propostos e seguindo as normas da ABNT.

6 Referências

Referências

- [1] Arduino. *Arduino Reference*. Disponível em: <https://www.arduino.cc/reference/en/>. Acesso em: [data de acesso].
- [2] Flask. *Flask Documentation*. Disponível em: <https://flask.palletsprojects.com/>. Acesso em: [data de acesso].
- [3] PySerial. *PySerial Documentation*. Disponível em: <https://pythonhosted.org/pyserial/>. Acesso em: [data de acesso].
- [4] Matplotlib. *Matplotlib Documentation*. Disponível em: <https://matplotlib.org/>. Acesso em: [data de acesso].
- [5] OpenCV. *OpenCV Documentation*. Disponível em: <https://docs.opencv.org/>. Acesso em: [data de acesso].