

SoundMonitor: Sistema Embarcado para Monitoramento Sonoro Inteligente

Eric Natan Batista Torres

Área temática: automação e controle sonoro em ambientes de grande porte

RESUMO

O presente projeto apresenta o **SoundMonitor**, um sistema embarcado voltado para a monitoração sonora em tempo real. Seu objetivo central é auxiliar operadores de som em ambientes como igrejas, teatros e eventos ao vivo, permitindo que tenham um retorno preciso sobre os níveis sonoros percebidos pelo público. Para isso, o sistema utiliza um microfone para captar o som ambiente, exibindo os níveis em uma matriz de LEDs e um display OLED. Além disso, os dados são transmitidos via Wi-Fi para o ThingSpeak, onde são armazenados e transformados em gráficos para análise histórica. Os testes demonstraram a eficácia do sistema na identificação e sinalização de diferentes níveis de volume, proporcionando uma solução acessível e prática para o controle sonoro em grandes espaços.

Palavras-chave: Monitoramento sonoro. IoT. Sistemas embarcados. Acústica. Automação.

INTRODUÇÃO

Em ambientes como igrejas, teatros e eventos ao vivo, a qualidade do som desempenha um papel fundamental na experiência do público. No entanto, um dos principais desafios enfrentados pelos operadores de som é a dificuldade de avaliar com precisão o volume percebido pelos ouvintes, uma vez que a mesa de som geralmente está localizada em uma posição distante da plateia. Essa limitação pode resultar em ajustes inadequados de volume, comprometendo a inteligibilidade do som e o conforto dos ouvintes.

Para solucionar esse problema, este projeto propõe o **SoundMonitor**, um sistema embarcado capaz de captar o som ambiente e fornecer um retorno visual ao operador de áudio. O sistema utiliza um microfone para medir os níveis sonoros, representando essas informações por meio de uma matriz de LEDs e um display OLED. Além disso, os dados são transmitidos via Wi-Fi para a plataforma ThingSpeak, permitindo a geração de gráficos que possibilitam uma análise histórica do comportamento sonoro no ambiente.

O objetivo principal do projeto é proporcionar aos operadores de som uma ferramenta confiável para monitoramento em tempo real, permitindo ajustes mais precisos e melhorando a experiência auditiva do público. Além disso, o sistema pode ser aplicado em diferentes contextos onde o responsável pelo controle do som está distante da fonte sonora, como em shows, conferências e auditórios.

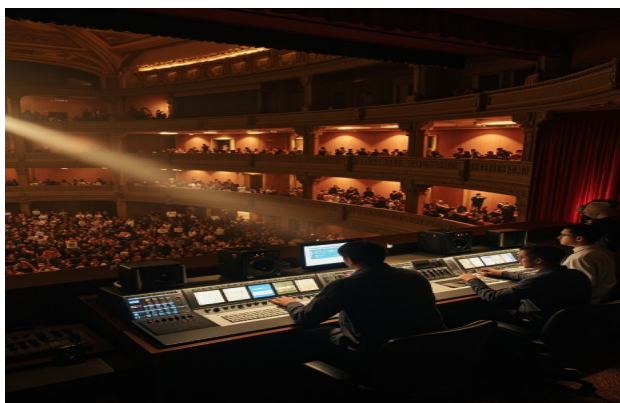


Figura 1 – Representação gerada por inteligência artificial de um teatro com cabine de monitoramento sonoro.

METODOLOGIA

O desenvolvimento do **SoundMonitor** envolveu a utilização da placa **BitDogLab**, que integra os principais componentes necessários para o processamento do sinal sonoro e comunicação com a nuvem. Os principais elementos empregados no projeto incluem:

- **Botão A:** Liga o sistema ao ser pressionado.
- **Botão B:** Desliga o sistema ao ser segurado.
- **Microfone:** Responsável pela captação do som ambiente e envio do sinal para conversão e análise.
- **Conversor Analógico-Digital (ADC):** Utilizado para transformar o sinal analógico do microfone em valores digitais.
- **Matriz de LEDs:** Empregada para fornecer um retorno visual rápido sobre o nível sonoro, alterando as cores conforme a intensidade do som, sendo azul para volumes baixos e muito baixos, verde para moderado e vermelho para alto e muito alto.

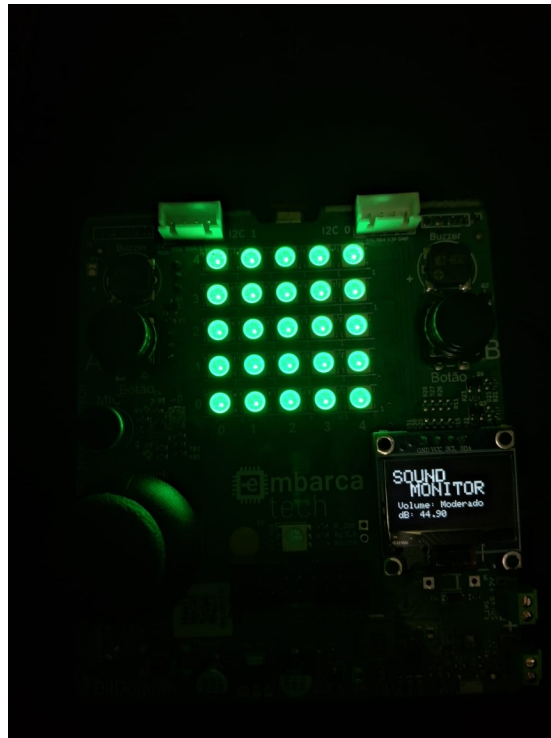


Figura 2 – Demonstração da matriz de LEDs em funcionamento para indicação de nível sonoro.

- **Display OLED:** Exibe mensagens de inicialização e desligamento e informações detalhadas, como o nível de volume e o valor medido em decibéis (dB).



Figura 3 – Exibição da mensagem de inicialização no display OLED (Parte 1).

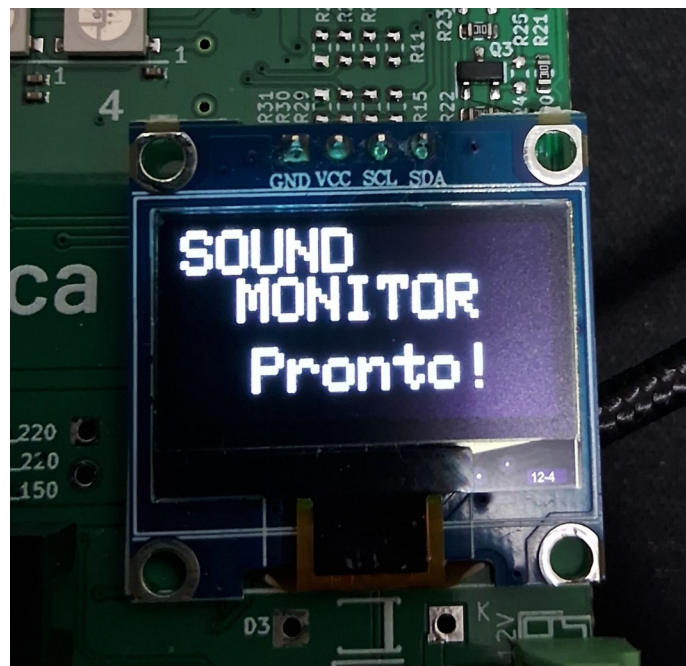


Figura 4 – Exibição da mensagem de inicialização no display OLED (Parte 2).



Figura 5 – Exibição da mensagem de monitoramento sonoro no display OLED.



Figura 6 – Exibição da mensagem de desligamento no display OLED.

- **Conectividade Wi-Fi:** Permite a transmissão dos dados para o **ThingSpeak** via protocolo HTTP.

```
[INFO] Endereço IP do ThingSpeak resolvido: 52.1.63.150
[INFO] Conectado ao servidor!
[DADOS] Dados enviados com sucesso!
[DADOS] dB: 42.72, Volume: Baixo

[INFO] Endereço IP do ThingSpeak resolvido: 52.1.63.150
[INFO] Conectado ao servidor!
[DADOS] Dados enviados com sucesso!
[DADOS] dB: 44.97, Volume: Moderado
```

Figura 7 – Captura de tela da mensagem de depuração da conexão Wi-Fi e envio de dados para o ThingSpeak.

Para melhorar a organização do projeto, o código foi modularizado em várias classes, sendo elas **neopixel.c**, **microfone.c**, **main.c**, **display_oled.c** e **wifi.c**, garantindo uma estrutura organizada e facilitando a manutenção do sistema.

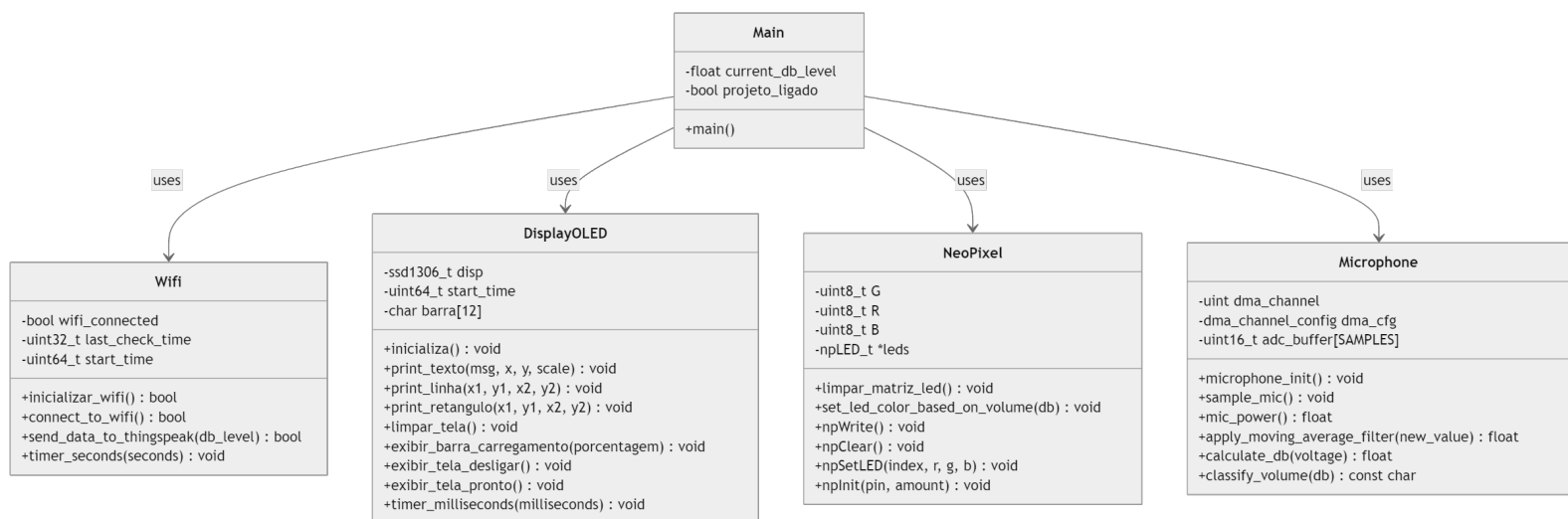


Figura 8 – Diagrama de classe do projeto.

DESENVOLVIMENTO

O desenvolvimento deste projeto foi estruturado em diversas etapas, abrangendo desde a criação de diagramas e a captação do som até a transmissão segura dos dados para a nuvem. Cada fase foi planejada com rigor para assegurar precisão, eficiência e confiabilidade no monitoramento sonoro.

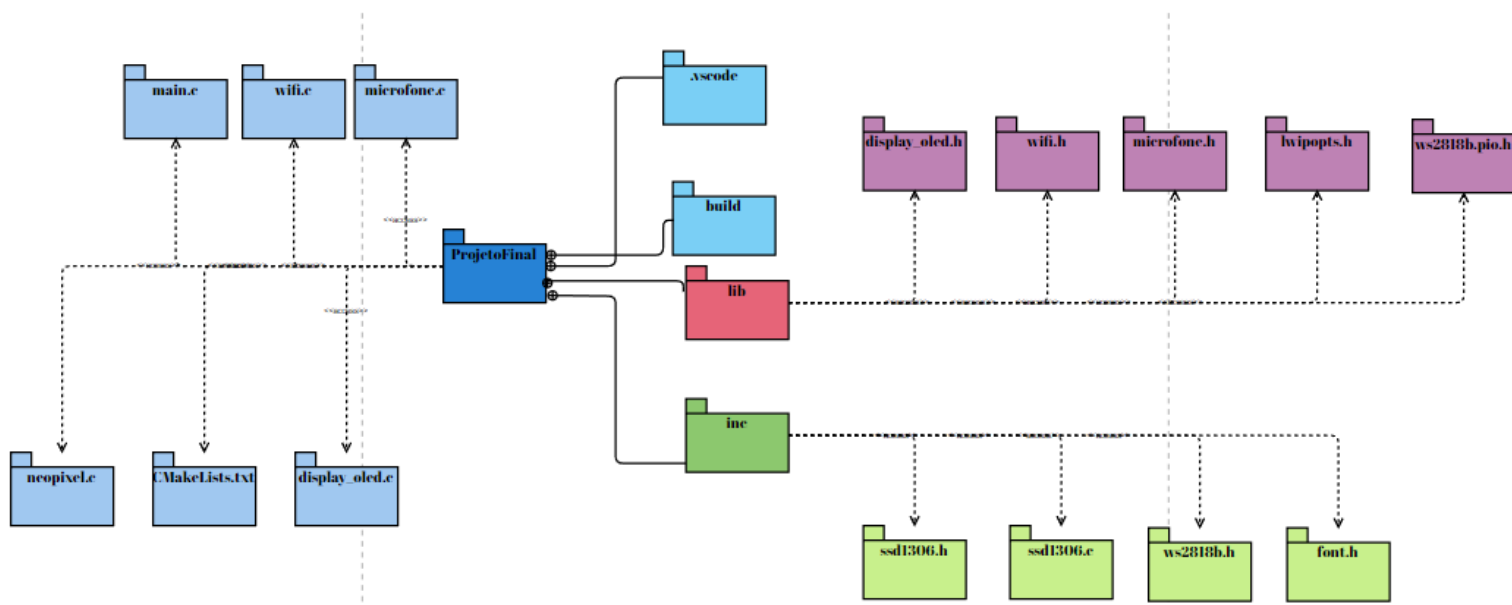


Figura 9 – Estrutura hierárquica do diretório do projeto.

Fluxo de funcionamento:

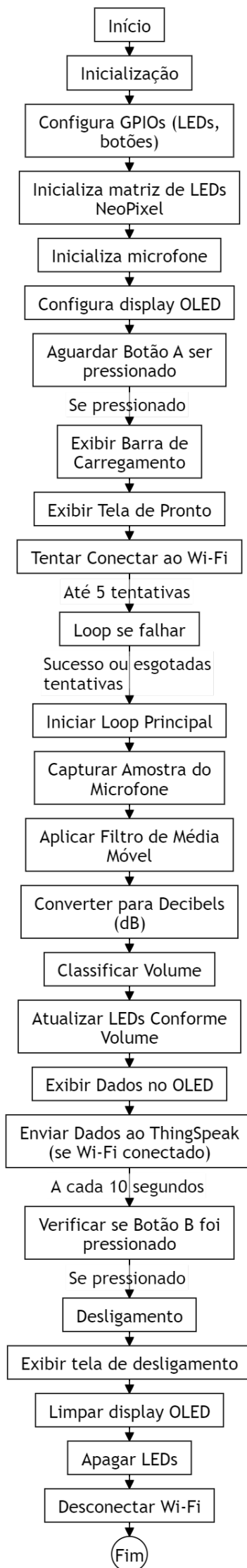


Figura 10 – Fluxograma de atividade.

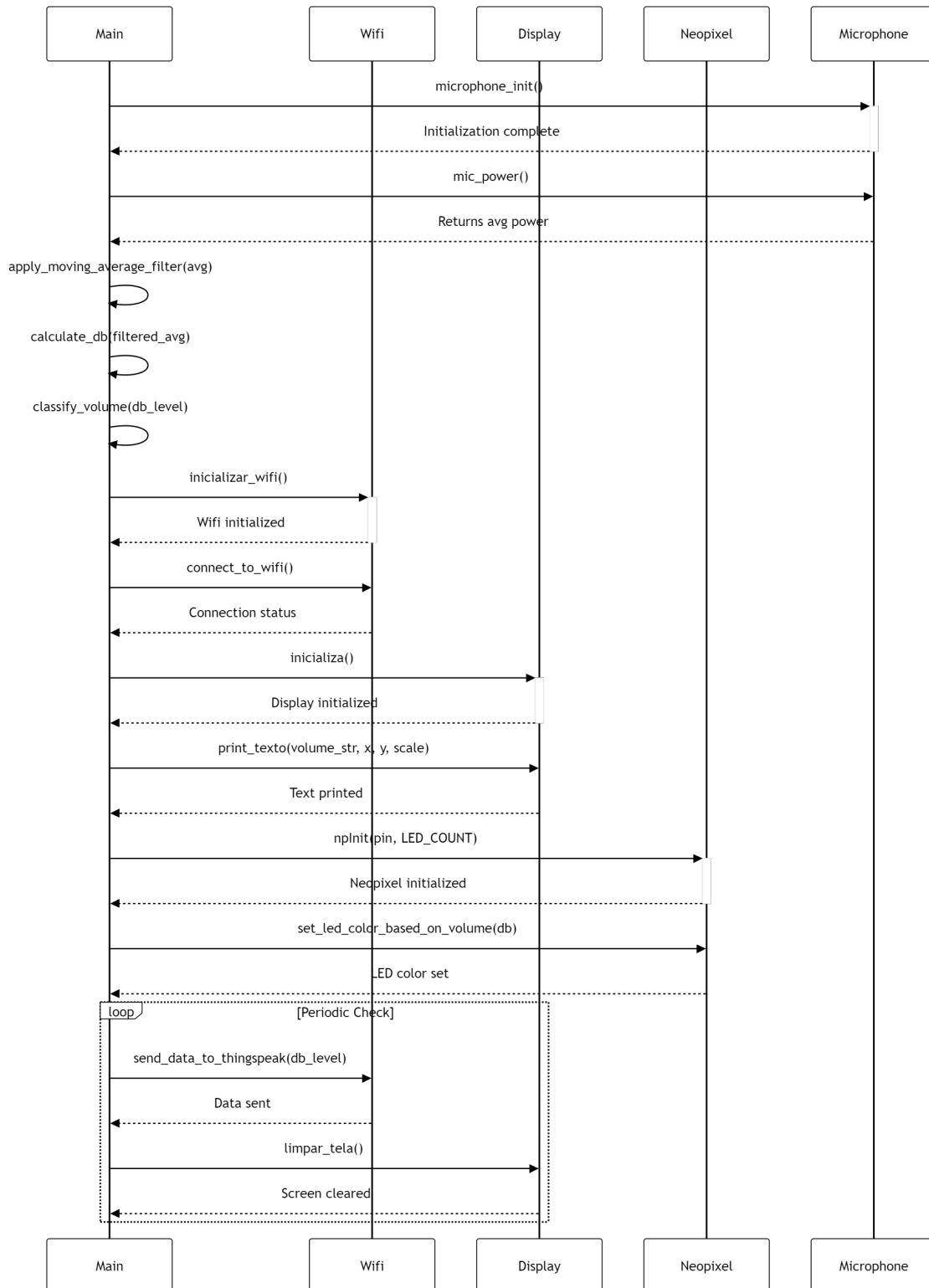


Figura 11 – Diagrama de atividade do projeto.

1. Captação e Processamento do Som

A captação do som ambiente é realizada por um microfone, que converte as ondas sonoras em sinais elétricos. Esses sinais são então transmitidos para um conversor ADC (*Analog-to-Digital Converter*), responsável por transformar os valores analógicos em dados digitais, permitindo a manipulação pelo microcontrolador.

Para otimizar o processamento dos dados e reduzir a carga sobre a CPU, o sistema utiliza DMA (*Direct Memory Access*), permitindo que as transferências de dados do ADC para a memória ocorram sem a intervenção direta do processador. Isso garante uma amostragem precisa e contínua dos níveis sonoros, reduzindo latências e minimizando ruídos indesejados. Dessa forma, o microcontrolador pode realizar outras tarefas simultaneamente, aumentando a eficiência e a confiabilidade do sistema.

2. Classificação dos Níveis Sonoros

O microcontrolador recebe os valores digitalizados e processa-os por meio de um algoritmo de categorização de som. Esse algoritmo compara os valores captados com um conjunto de faixas predefinidas (definidas de acordo com os testes realizados no microfone da placa):

- **Baixo:** Até 43 dB
- **Moderado:** Entre 44 dB e 55 dB
- **Alto:** Entre 56 dB e 60 dB
- **Muito Alto:** Acima de 60 dB

Essa classificação é essencial para determinar as reações do sistema e garantir que os alertas visuais e a transmissão de dados ocorram de forma correta.

3. Exibição das Informações

Com base na classificação do nível sonoro, o sistema utiliza diferentes meios para exibir as informações:

- **Matriz de LEDs:** Altera sua cor conforme o nível sonoro detectado:
 - Azul: Som baixo
 - Verde: Som moderado
 - Vermelho: Som alto

- Vermelho: Som muito alto
- **Display OLED:** As informações são exibidas em um display OLED, que utiliza o protocolo I2C para comunicação, permitindo uma integração eficiente com o sistema. Esse display apresenta a leitura exata em decibéis, proporcionando ao usuário uma visualização clara e imediata dos níveis sonoros captados.

Essa abordagem combinada de feedback visual e numérico assegura maior precisão na interpretação dos níveis sonoros.

4. Transmissão para a Nuvem

Os dados coletados são enviados para a plataforma ThingSpeak a cada 15 segundos, respeitando a limitação imposta pela versão gratuita do serviço. A transmissão ocorre via Wi-Fi, utilizando o protocolo HTTP para garantir a comunicação eficiente com a nuvem.

Os valores armazenados na plataforma são utilizados para:

- Gerar gráficos históricos dos níveis sonoros.
- Simular remotamente o comportamento dos LEDs, proporcionando um acompanhamento visual em tempo real.
- Permitir a análise de padrões e tendências sonoras ao longo do tempo.

A integração com o ThingSpeak amplia a utilidade do sistema, possibilitando o acesso e a análise dos dados de qualquer local, facilitando o monitoramento remoto.

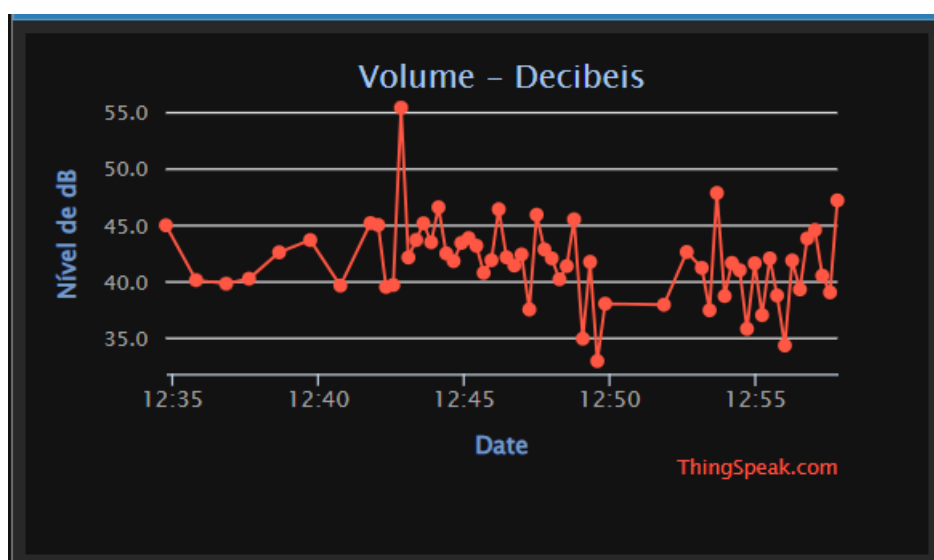


Figura 12 – Captura de tela do gráfico histórico gerado no ThingSpeak com os dados recebidos da placa.

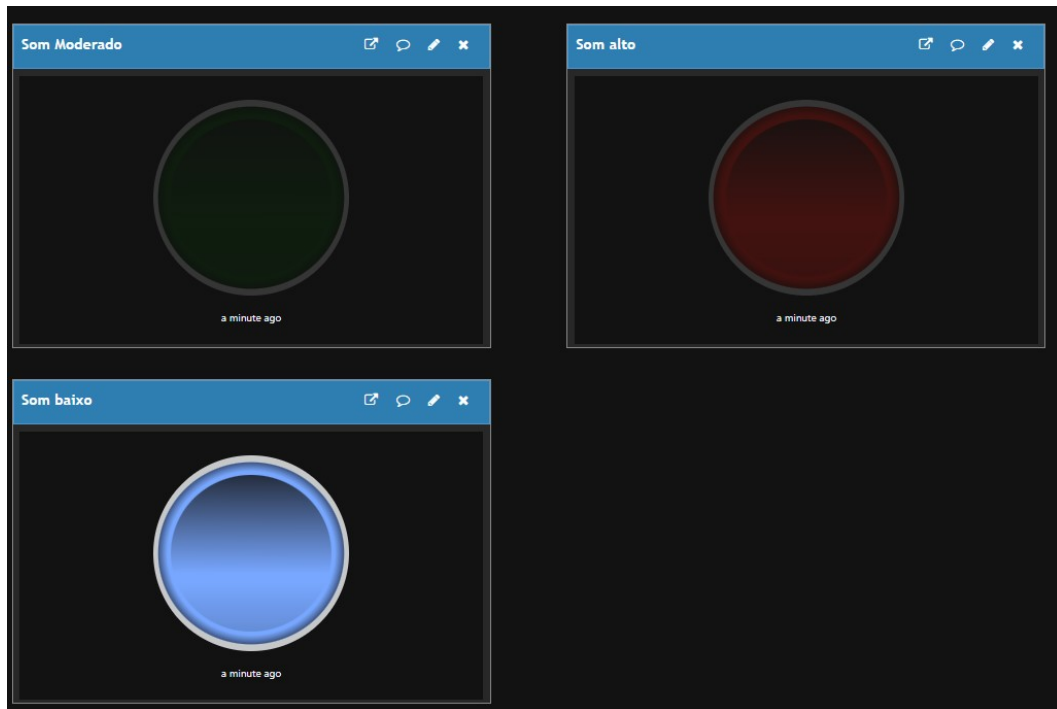


Figura 13 – Captura de tela do ThingSpeak simulando o acendimento dos LEDs de acordo com a placa.

5. Uso de Temporizadores

Durante o funcionamento do sistema, temporizadores são utilizados para controlar diferentes processos, garantindo eficiência e sincronização nas operações. Eles são empregados para:

- Pausar a execução em intervalos específicos, evitando sobrecarga no processamento.
- Realizar verificações periódicas dos níveis sonoros.
- Controlar loops de atualização do display e da matriz de LEDs.
- Gerenciar o envio de dados para a nuvem dentro do intervalo permitido pela plataforma ThingSpeak.

O uso de temporizadores permite que o sistema funcione de maneira otimizada, garantindo a precisão das medições e a fluidez na exibição e transmissão das informações.

```
void timer_milliseconds(int milliseconds) {
    uint64_t start_time = time_us_64(); // Obtém o tempo atual em microssegundos
    uint64_t delay = milliseconds * 1000; // Converte milissegundos para microssegundos

    // Aguarda até que o tempo especificado tenha passado
    while (time_us_64() - start_time < delay) {
        // Loop vazio (busy-wait)
    }
}
```

Figura 14 – Captura de tela do Visual Studio Code exibindo a implementação de um temporizador em milissegundos.

```
// Função de temporizador em segundos
void timer_seconds(int seconds) {
    uint64_t start_time = time_us_64(); // Obtém o tempo atual em microssegundos
    uint64_t delay = seconds * 1000000; // Converte segundos para microssegundos

    // Aguarda até que o tempo especificado tenha passado
    while (time_us_64() - start_time < delay) {
        // Loop vazio (busy-wait)
    }
}
```

Figura 15 – Captura de tela do Visual Studio Code exibindo a implementação de um temporizador em segundos.

RESULTADOS ALCANÇADOS E DISCUSSÕES

O desenvolvimento do **SoundMonitor** seguiu uma abordagem estruturada, garantindo a implementação correta de cada funcionalidade. A interação entre os componentes foi realizada por meio de um firmware desenvolvido para gerenciar a captação do som, a exibição dos níveis sonoros e a transmissão dos dados para a nuvem.

Para garantir o correto funcionamento do sistema, diversos testes foram realizados em cada etapa do projeto. Os principais testes incluíram:

- **Captação Sonora:** Comparação entre os valores captados pelo microfone e um medidor de decibéis externo para avaliar a precisão da conversão ADC.
- **Classificação e Resposta:** Simulação de diferentes fontes sonoras para validar a mudança de cor dos LEDs e a exibição correta dos valores no display OLED.
- **Conectividade e Transmissão:** Testes de estabilidade da conexão Wi-Fi e tempo de resposta na transmissão para o ThingSpeak.

- **Análise dos Dados na Nuvem:** Verificação da fidelidade dos gráficos gerados no ThingSpeak em relação aos dados capturados localmente.

Durante os testes, verificou-se que a matriz de LEDs apresentou uma resposta rápida e eficiente na indicação dos níveis sonoros. O display OLED forneceu informações detalhadas em tempo real, enquanto a transmissão para o ThingSpeak permitiu a criação de gráficos para análise posterior. Apesar das limitações do microfone utilizado, o sistema demonstrou ser funcional dentro das condições de teste estabelecidas. A latência na atualização dos dados na nuvem foi um pouco alta (aproximadamente 15 segundos) mas não comprometeu a usabilidade do sistema.

A evidência da confiabilidade do sistema é demonstrada pela coleta de **3250 registros de dados** no **ThingSpeak**, confirmando a robustez e a continuidade da transmissão das medições sonoras ao longo do tempo.

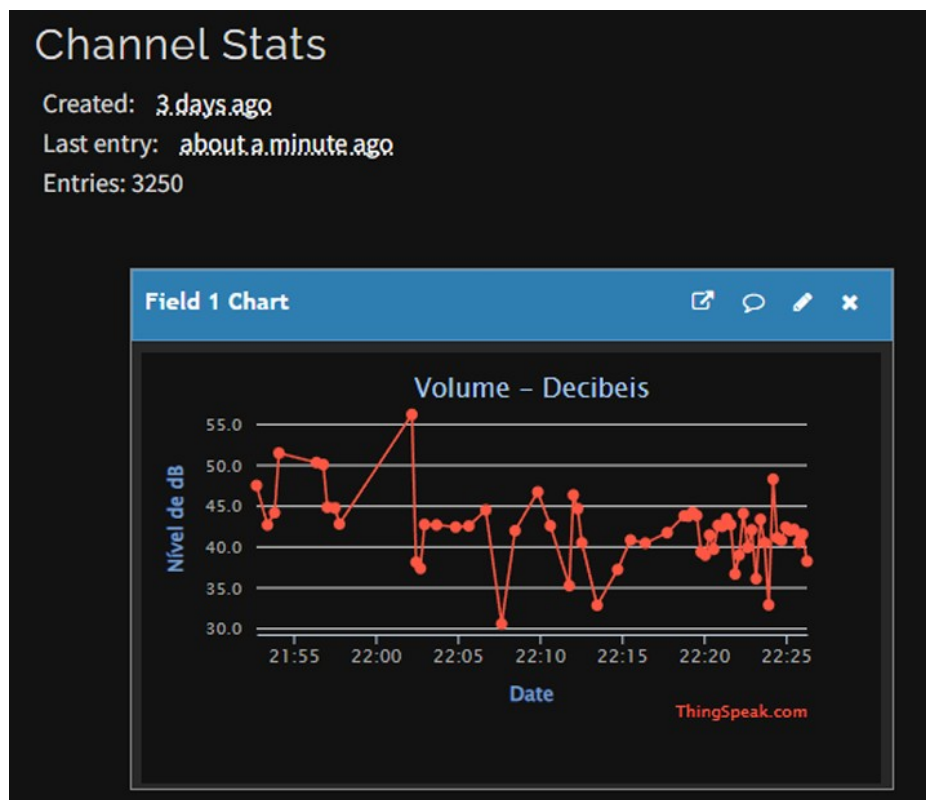


Figura 16 – Captura de tela do ThingSpeak mostrando quantidade de dados recebidos.

Em termos de aplicabilidade, o **SoundMonitor** mostrou-se uma solução viável para auxiliar operadores de som na calibração de volumes em ambientes amplos. Os resultados obtidos indicam que o sistema pode ser aprimorado com a inclusão de microfones de melhor

qualidade, algoritmos de processamento de áudio mais avançados e utilização de um serviço de nuvem melhor, ampliando sua precisão e confiabilidade.

Com base nos testes realizados, conclui-se que o **SoundMonitor** atende ao objetivo proposto de fornecer um monitoramento sonoro prático e acessível. A integração com o **ThingSpeak** amplia suas possibilidades de uso, permitindo uma análise detalhada dos padrões sonoros em diferentes eventos e ambientes. Como desenvolvimento futuro, pretende-se otimizar a captação do áudio e explorar novas funcionalidades para tornar o sistema ainda mais robusto e eficiente.

CONCLUSÃO

O projeto SoundMonitor atingiu seus objetivos ao oferecer um sistema de monitoramento sonoro acessível e funcional para operadores de som em ambientes amplos. O sistema demonstrou ser eficiente na captação e exibição dos níveis sonoros, contribuindo para um melhor ajuste do volume e aprimoramento da experiência auditiva do público.

Os principais desafios enfrentados envolveram as limitações do microfone utilizado, que impactaram a precisão da captação sonora, além das restrições do banco de dados da nuvem utilizado, o ThingSpeak na versão gratuita, que limitou a quantidade de dados armazenados e a frequência de atualizações. No entanto, o sistema se mostrou promissor, validando sua proposta inicial.

Como melhorias futuras, pretende-se substituir o microfone por um modelo de maior qualidade, aprimorar os algoritmos de processamento de áudio e explorar a integração com outros dispositivos para expandir suas funcionalidades. Também está prevista a migração para uma versão paga do ThingSpeak ou a utilização de outras soluções de banco de dados em nuvem, visando melhorar o armazenamento e a escalabilidade.

Em conclusão, o SoundMonitor representa uma solução inovadora e viável para a monitoração sonora em tempo real, sendo uma ferramenta útil para profissionais da área de áudio e automação.

LINKS DO PROJETO

Vídeo de apresentação:

<https://youtu.be/ffg8oL80Uyo>

ThingSpeak:

<https://thingspeak.mathworks.com/channels/2836790>

GitHub:

[HTTPS://GITHUB.COM/ERICNATANBT/SOUNDMONITOR](https://github.com/ERICNATANBT/SOUNDMONITOR)

REFERÊNCIAS

Monitoramento de som ambiente:

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 10151: Acústica – Medição e avaliação de níveis de pressão sonora em áreas habitadas – Aplicação de uso geral. Rio de Janeiro, 2019. Disponível em:
<http://www2.uesb.br/biblioteca/wp-content/uploads/2022/03/ABNT-NBR10151-ACÚSTICA-MEDICÃO-E-AVALIAÇÃO-DE-NÍVEL-SONORO-EM-ÁREA-HABITADAS.pdf> . Acesso em: 14 fev. 2025.

Volume em dB:

AUDIONOVA. Proteção auditiva: volume e decibéis. Disponível em:
<https://www.audionova.com.br/blog/protecao-auditiva/volume-e-decibéis/>. Acesso em: 14 fev. 2025.

Protocolos DMA:

TECNOBLOG. O que é DMA (Acesso Direto à Memória)? Disponível em:
<https://tecnoblog.net/responde/o-que-e-dma-acesso-direto-a-memoria/>. Acesso em: 14 fev. 2025.

Protocolos I2C:

EMBARCADOS. Comunicação I2C. Disponível em:
<https://embarcados.com.br/comunicacao-i2c/>. Acesso em: 14 fev. 2025.

Protocolos ADC:

DEWESOFT. O que é Conversor ADC? Disponível em: <https://dewesoft.com/pt/blog/o-que-e-conversor-adc>. Acesso em: 14 fev. 2025.

Protocolo HTTP:

MOZILLA DEVELOPER NETWORK. Protocolo HTTP. Disponível em:

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>. Acesso em: 14 fev. 2025.

Uso do ThingSpeak:

MATHWORKS. ThingSpeak™ Documentation. Disponível em: . Acesso em: 14 fev. 2025.

Placa BitDogLab:

EMBARCADOS. BitDogLab: uma jornada educativa com eletrônica embarcados e IA.

Disponível em: <https://embarcados.com.br/bitdoglab-uma-jornada-educativa-com-eletronica-embarcados-e-ia/>. Acesso em: 14 fev. 2025.