


Projeto nível sonoro

▼ Condição	Terminado
☰ From	 <u>Sistemas de monitoramento</u>
▼ Prioridade	Alta

Introdução

Este projeto utiliza um microfone MAX9814 conectado ao ESP32 para medir níveis de som em decibéis (dB) a partir de valores analógicos obtidos pelo ADC. Uma filtragem da amostragem de dados e interpolações foram empregadas para fornecer uma conversão precisa entre os valores do ADC e os níveis sonoros em dB com base em dados experimentais.

Componentes utilizados

O **ESP32** é um microcontrolador poderoso e versátil, amplamente utilizado em aplicações de automação, Internet das Coisas (IoT) e sistemas embarcados. Desenvolvido pela Espressif Systems, ele se destaca pela sua conectividade Wi-Fi e Bluetooth integrada, tornando-se uma escolha ideal para projetos que envolvem comunicação sem fio. Além disso, conta com um processador dual-core de alta performance, diversos pinos de entrada e saída (GPIOs), conversores analógico-digitais (ADCs) de 12 bits e suporte a múltiplos protocolos de comunicação, como I2C, SPI e UART. Graças à sua capacidade de processamento e eficiência energética, o ESP32 permite a criação de dispositivos inteligentes e sistemas embarcados sofisticados.

Para capturar e processar sinais de áudio, um dos sensores amplamente utilizados é o **módulo microfone MAX9814**. Esse módulo conta com um amplificador integrado de ganho automático (AGC), que ajusta dinamicamente o volume do sinal de entrada, garantindo uma captação uniforme do som, independentemente da intensidade da fonte sonora. Ele é particularmente útil em aplicações que envolvem reconhecimento de voz, análise de ruído ambiente ou medições de nível sonoro, como um decibelímetro. Além disso, a saída analógica do MAX9814 pode ser facilmente conectada a um pino ADC do

ESP32, permitindo a leitura e o processamento dos sinais de áudio de forma eficiente.

Para exibir informações em tempo real, um ótimo complemento para esse sistema é o **módulo OLED I2C de 0.96"**. Esse pequeno display utiliza a tecnologia OLED, o que significa que ele não requer backlight, proporcionando excelente visibilidade mesmo em ambientes escuros e garantindo baixo consumo de energia. Com uma resolução de 128×64 pixels, ele é capaz de exibir texto e gráficos com ótima definição. O uso da interface I2C simplifica a conexão com o ESP32, necessitando de apenas dois fios para comunicação, além da alimentação. Isso torna o módulo OLED ideal para exibir leituras de sensores, status do sistema ou qualquer outra informação relevante de forma clara e acessível.

A integração desses três componentes — ESP32, microfone MAX9814 e display OLED — permite o desenvolvimento de aplicações interessantes, como um monitor de nível de ruído em tempo real. O microfone capta os sons do ambiente, o ESP32 processa os dados e realiza cálculos de nível sonoro, e o display OLED exibe os valores em decibéis, fornecendo um feedback imediato ao usuário. Essa combinação de hardware pode ser utilizada em diversas áreas, desde monitoramento acústico em ambientes urbanos até sistemas de alerta para locais sensíveis ao ruído.

Conexões elétricas

Conexões entre ESP32 e Microfone MAX9814.

Microfone MAX9814	ESP32	Descrição
GND	GND	Terra comum
VIN	Vdd	Alimentação do microfone
GND	Gain	Configuração do ganho para 50dB
Out	GPIO35	Saída do microfone conectada ao ADC do ESP32

Conexões entre Módulo OLED e ESP32.

Módulo OLED (I2C)	ESP32	Descrição
VCC	3V3	Alimentação do módulo OLED
GND	GND	Terra comum

SDA	GPIO21	Linha de dados I2C
SCL	GPIO22	Linha de clock I2C

Amostragem dos dados

O som é uma **onda mecânica** que se propaga através de um meio físico, como o ar, causando variações na pressão do ambiente. Essas variações de pressão são captadas pelo **módulo microfone MAX9814**, que converte as oscilações da onda sonora em um **sinal elétrico analógico**, ou seja, uma variação contínua de tensão ao longo do tempo.

O **ESP32**, por sua vez, interpreta essa variação de tensão por meio do seu **Conversor Analógico-Digital (ADC)**. O ADC transforma o sinal analógico do microfone em valores numéricos discretos dentro de um intervalo específico. No entanto, o sinal analógico do microfone é naturalmente **oscilatório**, ou seja, apresenta valores que variam periodicamente ao longo do tempo, formando uma onda semelhante a uma **função senoidal**. Essa oscilação reflete a natureza da onda sonora, onde regiões de compressão e rarefação da pressão do ar resultam em variações na amplitude do sinal elétrico gerado pelo microfone.

Amostragem e Limitações do ADC

O ADC do ESP32 não mede o sinal continuamente, mas sim em intervalos discretos de tempo, um processo chamado **amostragem**. A taxa de amostragem do microcontrolador define quantas vezes por segundo o ADC captura um valor da onda sonora. Como essa taxa não é infinita, pode acontecer de algumas variações rápidas do sinal não serem capturadas precisamente, levando a perdas de informações entre os picos e vales da onda sonora.

Como nosso objetivo é medir a **intensidade sonora** do ambiente, a informação mais relevante para nós é a **amplitude do sinal**, pois a intensidade sonora percebida pelo ouvido humano está diretamente relacionada à amplitude da onda sonora. A amplitude representa a variação máxima da pressão sonora e, conseqüentemente, do sinal elétrico gerado pelo microfone.

Detecção de Picos e Filtragem do Sinal

Devido à oscilação do sinal captado pelo microfone, os valores fornecidos pelo ADC flutuam continuamente entre picos (máximos locais) e vales (mínimos

locais). Para evitar leituras imprecisas e obter uma estimativa confiável da intensidade do som, é essencial focar na detecção dos **picos da onda sonora**.

Se considerássemos apenas leituras aleatórias sem filtragem, poderíamos obter valores intermediários que não representariam corretamente a intensidade real do som. Para corrigir isso, podemos utilizar estratégias como:

1. Registro do valor máximo em uma janela de tempo:

- Durante um curto intervalo de tempo, o código mantém o maior valor ADC registrado. Isso garante que capturamos o pico da onda sonora sem depender diretamente da taxa de amostragem.

2. Cálculo da média dos picos ao longo do tempo:

- Como alternativa, podemos calcular uma **média móvel** dos valores máximos dentro de um período definido para suavizar variações abruptas e obter um resultado mais estável.

3. Uso de filtros digitais:

- Técnicas como **filtros passa-alta e passa-baixa** podem ser aplicadas para remover ruídos indesejados e realçar os valores de pico do sinal.

A correta interpretação do sinal do microfone pelo ESP32 requer atenção à **amplitude da onda sonora** para estimar a intensidade do som de maneira precisa. Como não podemos capturar todos os pontos da onda devido à taxa de amostragem limitada, precisamos implementar **métodos de detecção de picos** para garantir que os valores convertidos pelo ADC reflitam adequadamente o nível de som presente no ambiente.

Durante todos os testes realizados, o registro do valor máximo em uma janela de tempo foi o método utilizado para filtrar os valores ADC úteis para a análise. As outras duas abordagens podem ser usadas posteriormente para avaliar a sua eficácia.

Interpolação dos dados

A conversão do valor ADC para um nível sonoro em decibéis não possui uma relação direta expressa por uma única equação que ligue precisamente os valores de ADC aos níveis sonoros medidos. Diante disso, optamos por utilizar um método de interpolação baseado em pontos de referência previamente determinados.

A interpolação matemática é uma técnica utilizada para estimar valores intermediários de uma função com base em um conjunto de dados conhecidos. Esse método consiste em construir uma função que passe exatamente pelos **nós de interpolação** — pontos de referência conhecidos — e utilizá-la para determinar valores em posições intermediárias com um alto grau de precisão.

No nosso caso, os pontos de interpolação foram determinados por meio de testes realizados com o microfone MAX9814 e o aplicativo **Sound Meter**. Para cada nível sonoro, registramos o valor ADC fornecido pelo microfone enquanto o aplicativo media a intensidade sonora ambiente. Dessa forma, conseguimos associar essas duas informações. Os pontos identificados foram:

- **1690 (valor ADC) → 58 dB**
- **2262 (valor ADC) → 70 dB**
- **2305 (valor ADC) → 75 dB**
- **2352 (valor ADC) → 80 dB**

Considerando que a intensidade sonora segue uma escala logarítmica, inicialmente testamos um modelo de **interpolação logarítmica** para ajustar os dados. No entanto, observamos que esse método não fornecia um ajuste ideal para toda a faixa de valores.

Para melhorar a precisão da conversão, implementamos uma **aproximação polinomial** para a região superior dos valores ADC. Além disso, utilizamos uma interpolação linear para suavizar a transição entre as regiões ajustadas pelos diferentes modelos. O resultado final foi uma interpolação mista composta por três trechos:

1. **Interpolação logarítmica** para valores de ADC menores que 2050.
2. **Interpolação linear** para valores entre 2050 e 2262.
3. **Interpolação polinomial** para valores acima de 2262.

Essa abordagem híbrida permitiu obter uma conversão mais precisa e confiável entre os valores ADC e os níveis sonoros em decibéis, garantindo um comportamento mais coerente com a realidade física do fenômeno sonoro.

Interpolação de ajuste logarítmico

A interpolação logarítmica é um tipo específico de interpolação onde se assume que a relação entre x e y segue um comportamento logarítmico da

forma:

$$y = a \cdot \log(x) + b$$

Aqui:

- $\log(x)$ pode ser o logaritmo natural ($\ln(x)$) ou de outra base.
- a e b são coeficientes ajustados para se encaixar nos pontos conhecidos.

Esse tipo de interpolação é útil quando a taxa de variação de y diminui à medida que x aumenta, algo comum em fenômenos como percepção sonora (decibéis), crescimento populacional inicial, e escalas perceptuais.

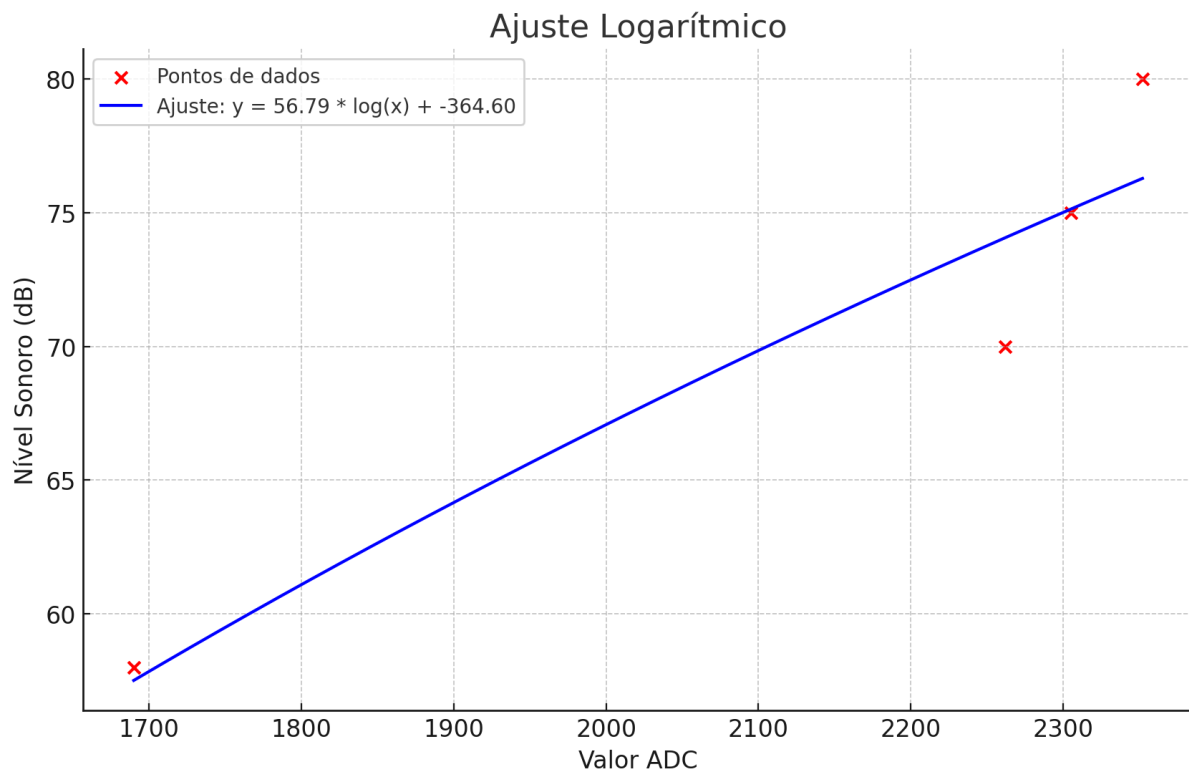
Um software de computador foi utilizado para calcular os valores a e b de acordo com os pontos avaliados anteriormente. Os coeficientes ajustados para a função logarítmica foram:

$$y = 56.79 \cdot \log(x) - 364.60$$

Onde:

- x é o valor ADC.
- y é o nível sonoro correspondente (em dB).

Sendo assim, um gráfico foi plotado para verificar como ficou o ajuste dos pontos.



Interpolação de ajuste logarítmico

Testes foram feitos com o auxílio do app Sound meter. Percebeu-se que os valores ficavam ajustados no intervalo de 1700 a 2050, porém, valores acima disso não correspondiam bem aos valores de decibéis informados pelo app. Para mitigar essa situação, um ajuste polinomial foi realizado.

Interpolação polonomial

A interpolação polinomial é um método usado para encontrar uma função polinomial que passa exatamente por um conjunto de pontos dados. Se temos um conjunto de pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, buscamos um polinômio da forma:

$$y = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

onde os coeficientes a_n, a_{n-1}, \dots, a_0 são determinados de forma que o polinômio passe exatamente pelos pontos fornecidos.

Para n pontos, podemos ajustar um polinômio de grau $n - 1$ que passe exatamente por todos os pontos.

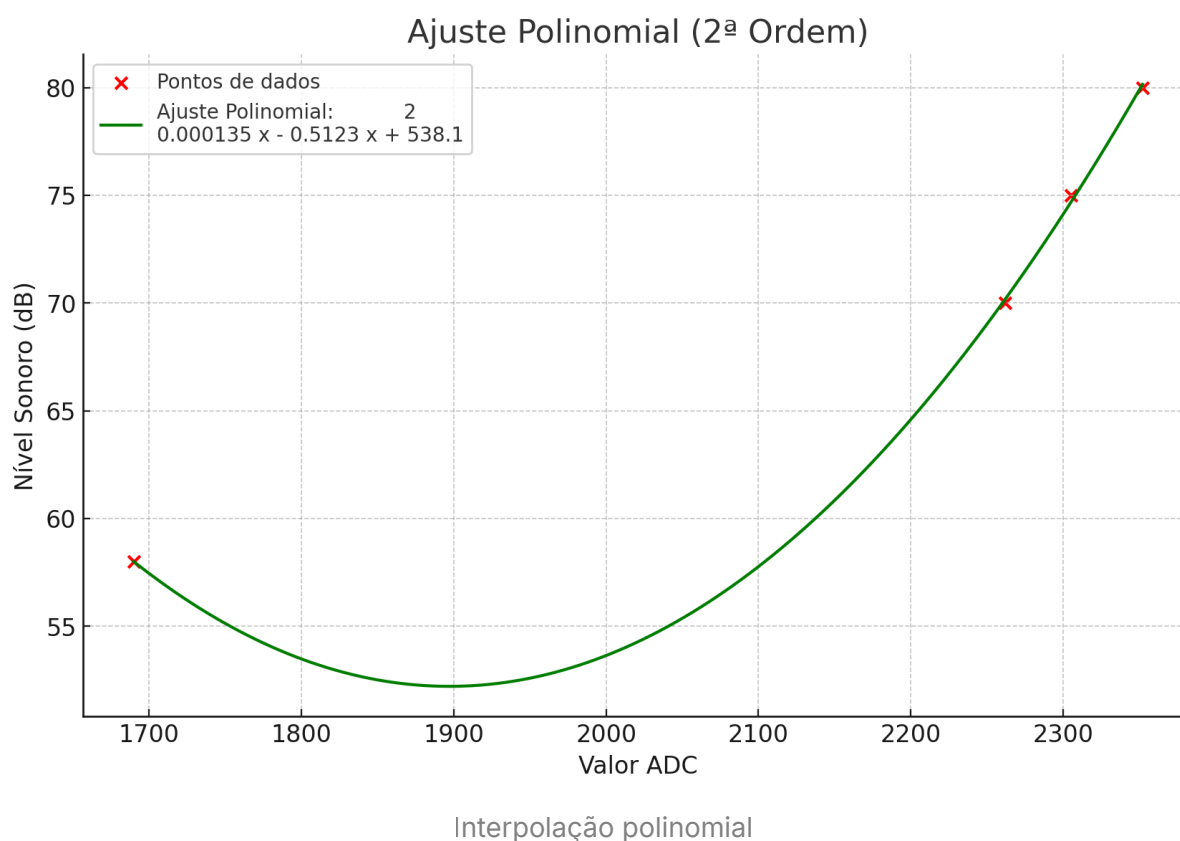
Métodos de Cálculo:

- **Interpolação de Lagrange:** Usa uma combinação de polinômios básicos para construir a função final.
- **Interpolação de Newton:** Usa diferenças divididas para encontrar os coeficientes do polinômio.
- **Ajuste por Mínimos Quadrados:** Se os pontos possuem ruído ou não seguem um padrão exato, pode-se ajustar um polinômio que minimize o erro médio.

Um software de computador foi utilizado para ajustar os parâmetros e assim criar a função polinomial que mais se aproximava de todos os pontos. A função determinada pelo software foi:

$$y = 0.000135 \cdot x^2 - 0.5123 \cdot x + 538.10$$

Sendo assim, um gráfico foi plotado para verificar como ficou o ajuste dos pontos.



Testes foram feitos e percebeu-se que para sons mais fortes, essa abordagem se saiu melhor, com valores entre 2262 e 2400. Porém valores mais baixos do que isso ficavam incorretos. Perceba por exemplo que a partir do primeiro

ponto de interpolação o gráfico é decrescente, o que não acontece experimentalmente, onde a medida que os valores ADC aumentam, os valores de intensidade sonora (db) aumentam também de algum modo. Sendo assim, na próxima etapa optou-se por uma abordagem mista.

Interpolação mista

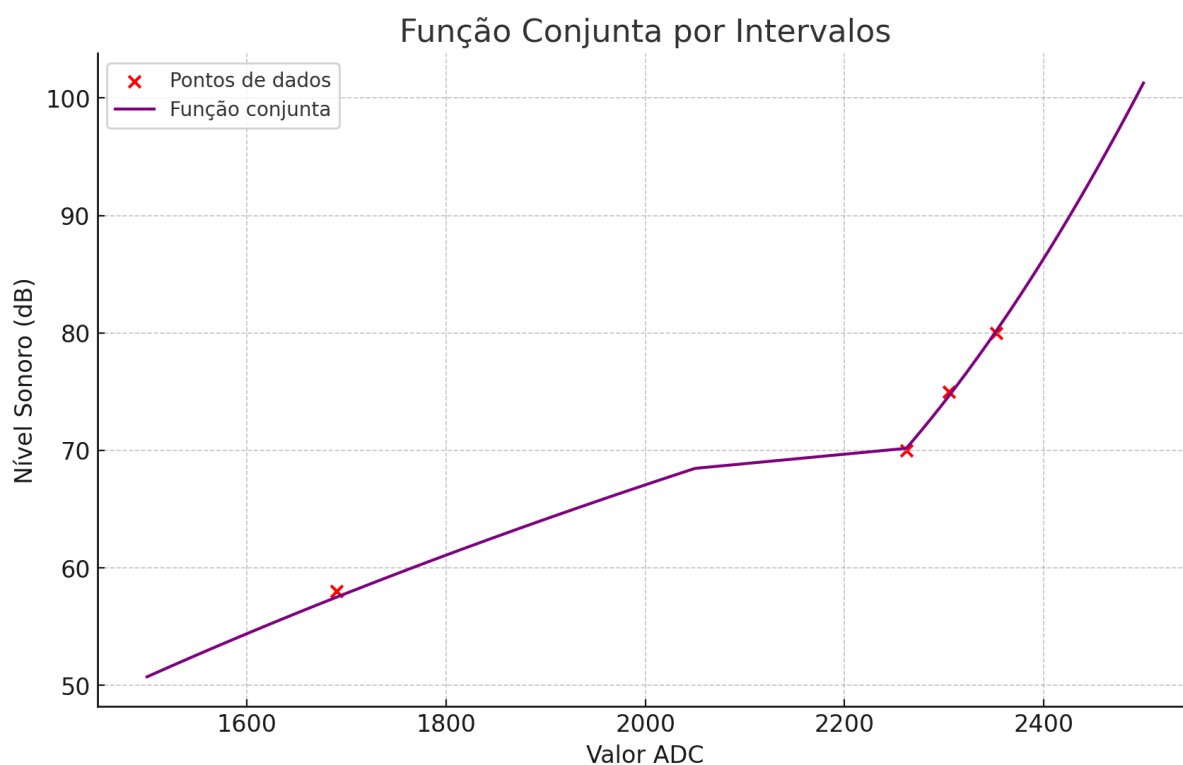
Para aproveitar os intervalos de ajuste razoavelmente precisos das interpolações anteriores, nesta etapa optou-se por criar diferentes interpolações para diferentes intervalos.

A função conjunta foi criada e avaliada nos intervalos definidos:

1. **De 0 até 2050:** Utiliza a função de ajuste logarítmico.
2. **De 2050 até 2262:** Utiliza uma função polinomial de 1ª ordem.
3. **De 2262 em diante:** Utiliza a função polinomial de 2ª ordem.

No intervalo entre 2050 e 2262, uma função polinomial de 1ª ordem foi utilizada para conectar os dois pontos do intervalo.

Sendo assim, um gráfico foi plotado para verificar como ficou o ajuste dos pontos.



Interpolação mista do tipo logarítmica, polinomial e função reta

Testes experimentais foram realizados. De acordo com os testes, os resultados do ADC e nível sonoro (em db) foram razoavelmente consistentes entre 50 e 85 decibéis de intensidade sonora.


Conclusão

Resultados

Referências

ChatGPT - Interpolação Matemática Explicada

Shared via ChatGPT

 <https://chatgpt.com/share/676ebb9e-1734-8012-a514-11b5cc59b5f2>

ChatGPT 

GitHub - Daniel-Alencar/sound-measurement-system

Contribute to Daniel-Alencar/sound-measurement-system development by creating an account on GitHub.

 <https://github.com/Daniel-Alencar/sound-measurement-system>

Daniel-Alencar/**sound-measurement-system**



 1 Contributor  0 Issues  1 Star  0 Forks

