

EmbarcaTech

Projeto de encerramento da Fase 1 (Relatório Final)

Autor: Jorge Wilker Mamede de Andrade

Turma: Campinas

Link do Repositório deste projeto: https://github.com/JorgeWilker/U7T_JWMA

1. Escopo do Projeto

Apresentação do Projeto:

Este projeto consiste no desenvolvimento de um **sistema embarcado de monitoramento de ruído** para ambientes onde o silêncio ou conversas em níveis moderados são essenciais. O dispositivo detecta automaticamente sons acima de um limiar pré-definido e emite alertas visuais e sonoros para conscientizar as pessoas sobre o excesso de barulho. Ele é ideal para ser utilizado em locais como hospitais, clínicas, bibliotecas, salas de espera e até em relacionamentos interpessoais. Muitas vezes, em discussões ou conversas acaloradas, as pessoas que falam alto não percebem seu próprio comportamento, o que pode gerar desentendimentos. É comum em uma discussão a parte ofendida dizer que a outra parte está “falando alto”, onde muitas vezes, essa confirmação não acontece pelo acusado, que acaba dizendo, que está falando em nível normal. Esse dispositivo vem para acabar com essa “teima” e tentar intermediar a paz nos relacionamentos através da constatação visual e sonora dos próprios envolvidos.

Título do Projeto:

Alarme de Monitoramento de Ruído para Ambientes Silenciosos e Relacionamentos

Objetivos do Projeto:

- Desenvolver um sistema que detecte níveis elevados de ruído em ambientes que exigem silêncio.
- Implementar respostas visuais (LEDs) e sonoras (buzzer) ao evento de detecção.
- Garantir que o sistema seja simples, funcional e fácil de reproduzir.

Principais Requisitos: Um dispositivo que ofereça uma medição objetiva do nível de ruído a um grau tolerado pelo contratante, ou seja, podendo ser ajustado facilmente, a depende da satisfação dele. Dispensar a necessidade de funcionários intervir diretamente para exigir silêncio, eliminando assim constrangimentos tanto para os funcionários quanto para os clientes. Espera-se que a automação do processo garanta que o ambiente permaneça tranquilo de forma discreta e eficiente, sem a necessidade de intervenção humana. O dispositivo deve ajudar a promover maior consciência e harmonia nas interações

Descrição do Funcionamento:

O sistema utiliza o ADC2 da Raspberry Pi Pico para ler sinais do microfone e calcular a amplitude do som. Quando a amplitude excede o limiar, o sistema ativa os LEDs e o buzzer por um período definido. A matriz de LEDs de 5x5 exibe um coração verde enquanto o ambiente está tranquilo, mas alterna para uma letra 'X' ao detectar ruído excessivo, acompanhada pelo som do buzzer. Após o tempo programado de alarme finalizar, o sistema retorna ao estado inicial, com o coração verde novamente visível, aguardando novas captações. Sendo assim o dispositivo opera em loop infinito.

Justificativa:

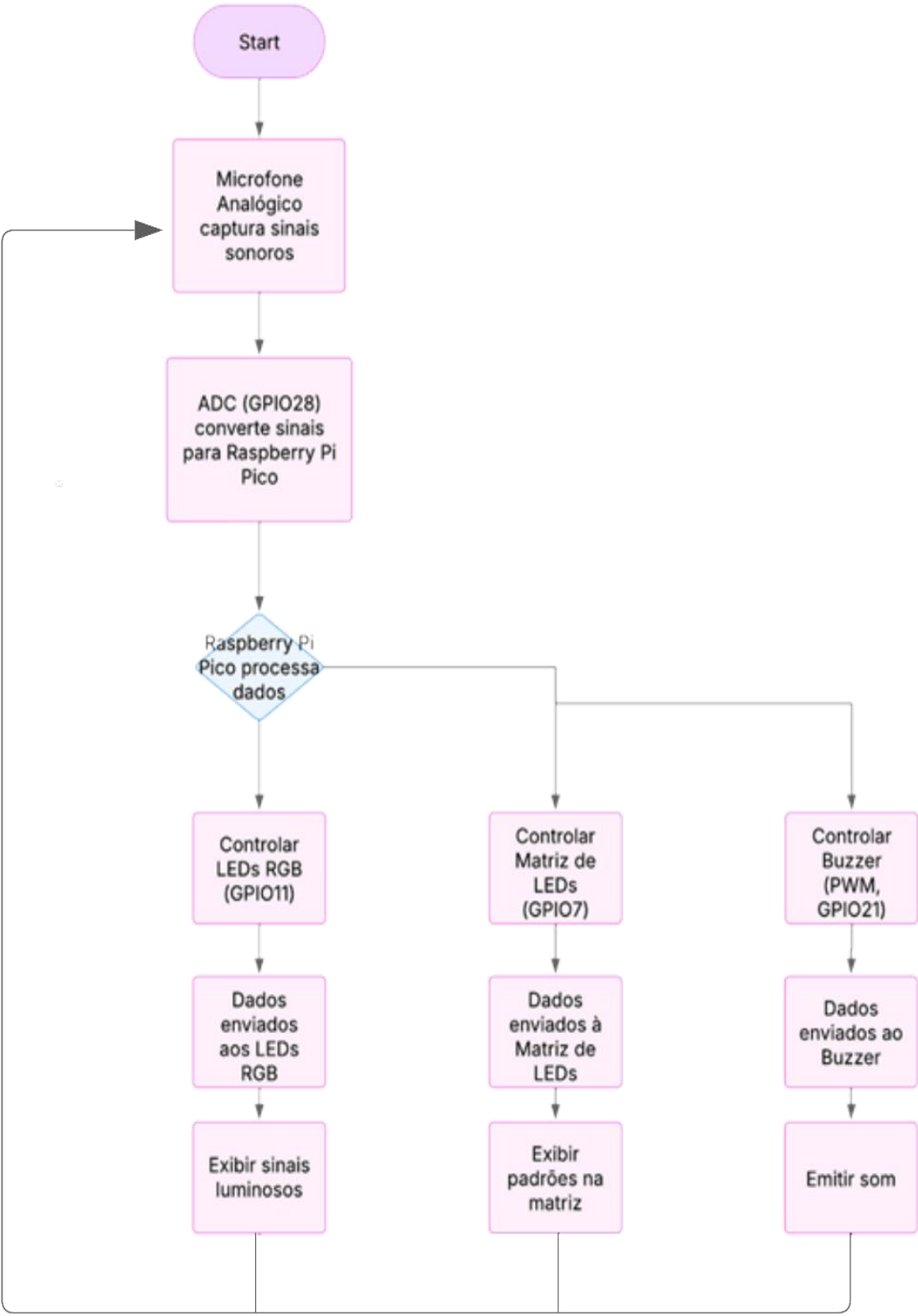
Este projeto demonstra a aplicação prática de sistemas embarcados em cenários de monitoramento sonoro. Ele combina hardware e software para resolver um problema real de forma eficiente. Além disso, ao adotar esse sistema, elimina-se a necessidade de intervenção humana para impor o silêncio, o que minimiza constrangimentos para os colaboradores e os frequentadores, contribuindo para um ambiente mais agradável e produtivo, e garantindo uma experiência mais positiva para todos.

Originalidade:

Embora existam projetos semelhantes no GitHub, como o repositório da BitDogLab , este projeto se diferencia pela integração de múltiplas funcionalidades (detecção de ruído, controle de LEDs, imagem apresentada na matriz, buzzer e adição de Pio) em um único sistema embarcado. A programação em C desse dispositivo o torna especial, pois a economia de recursos contribui para sua maior otimização. Além disso, a aplicação específica para monitoramento de ruído em ambientes como bibliotecas, hospitais e clínicas é uma **inovação** , pois soluções não são vistas nesse ambientes. A regulagem do ajuste fino de ruído é possível através de um simples comando de software que será demonstrado ao longo desse projeto. Assim ao adquirir o sistema embarcado o adquirente informa ao programador os limites de decibéis que deseja usar.

2. Hardware

Diagrama em Blocos:



Função de Cada Bloco:

- **Microfone:** Captura sinais sonoros e converte-os em sinais elétricos.
- **ADC:** Converte sinais analógicos do microfone em valores digitais.
- **Raspberry Pi Pico:** Processa os dados e controla os periféricos.
- **LEDs e Buzzer:** Fornecem feedback visual e sonoro.

Configuração de Cada Bloco:

- **Microfone:** Conectado ao pino **MICROPHONE_PIN** (ADC2, GPIO28).
- **LEDs RGB:** Controlados via PIO no pino **RGB_LED_PIN**.
- **Matriz de LEDs:** Controlada via PIO no pino **MATRIX_LED_PIN**. Exibe um coração verde quando o ambiente está tranquilo e uma letra 'X' quando o ruído é detectado.
- **Buzzer:** Controlado por PWM no pino **BUZZER_PIN** (slice PWM 4, GPIO21).

Especificações:

O sistema atende aos requisitos de baixo custo, tamanho reduzido e simplicidade de implementação como deve ser todo bom sistema embarcado. Os materiais usados na montagem são facilmente encontrados na internet e o preço é acessível não passando de R\$ 100,00. Quanto ao cliente, estará adquirindo um sistema com excelente custo-benefício. Após a configuração em seu ambiente, a manutenção será pouco necessária por um longo período, devido à robustez e à programação eficiente do sistema.

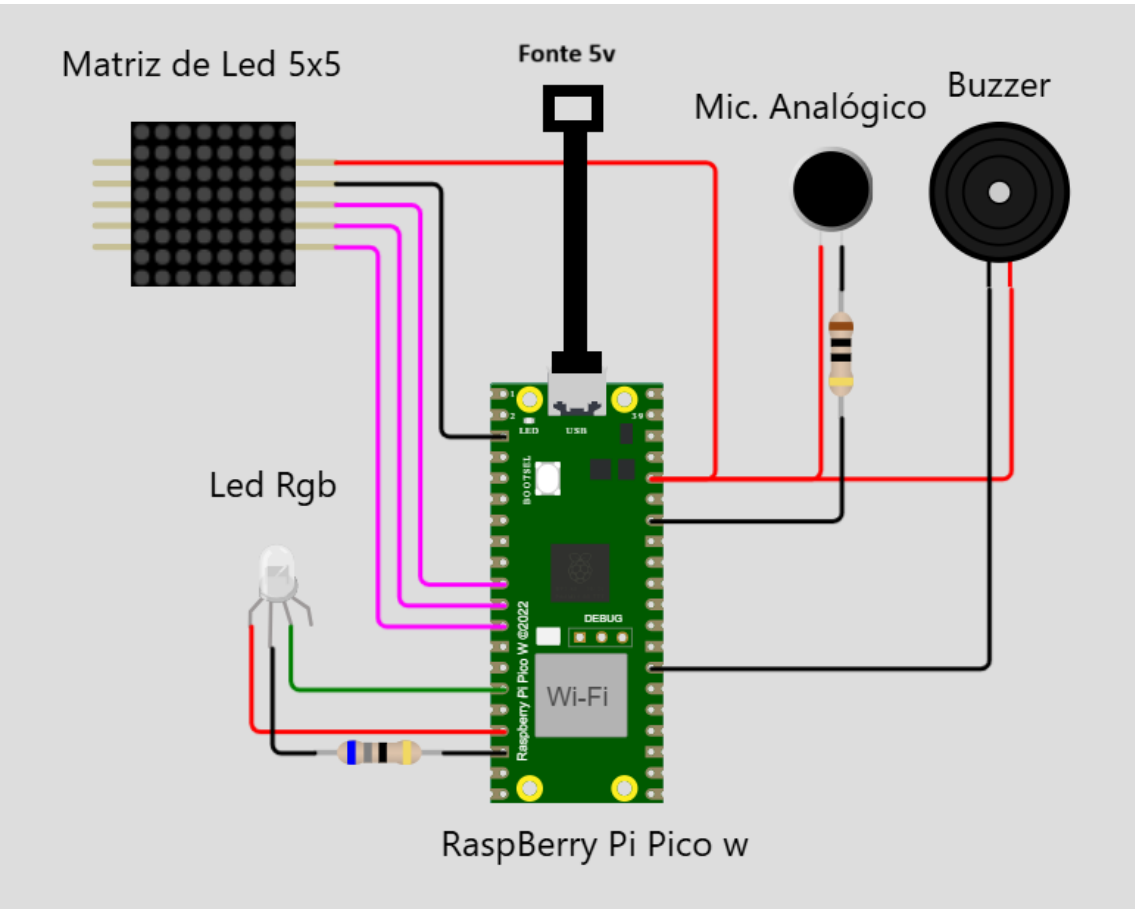
Lista de Materiais:

Componente	Quantidade	Disponibilidade	Preço em Fev/2025
Raspberry Pi Pico	1	Abundante	R\$ 40,00
Microfone Analógico	1	Abundante	R\$ 1,50
LEDs RGB	1	Abundante	R\$ 10,00
Matriz de LEDs	1	Abundante	R\$ 40,00
Buzzer	1 a 2	Abundante	R\$ 3,00

Descrição da Pinagem Usada:

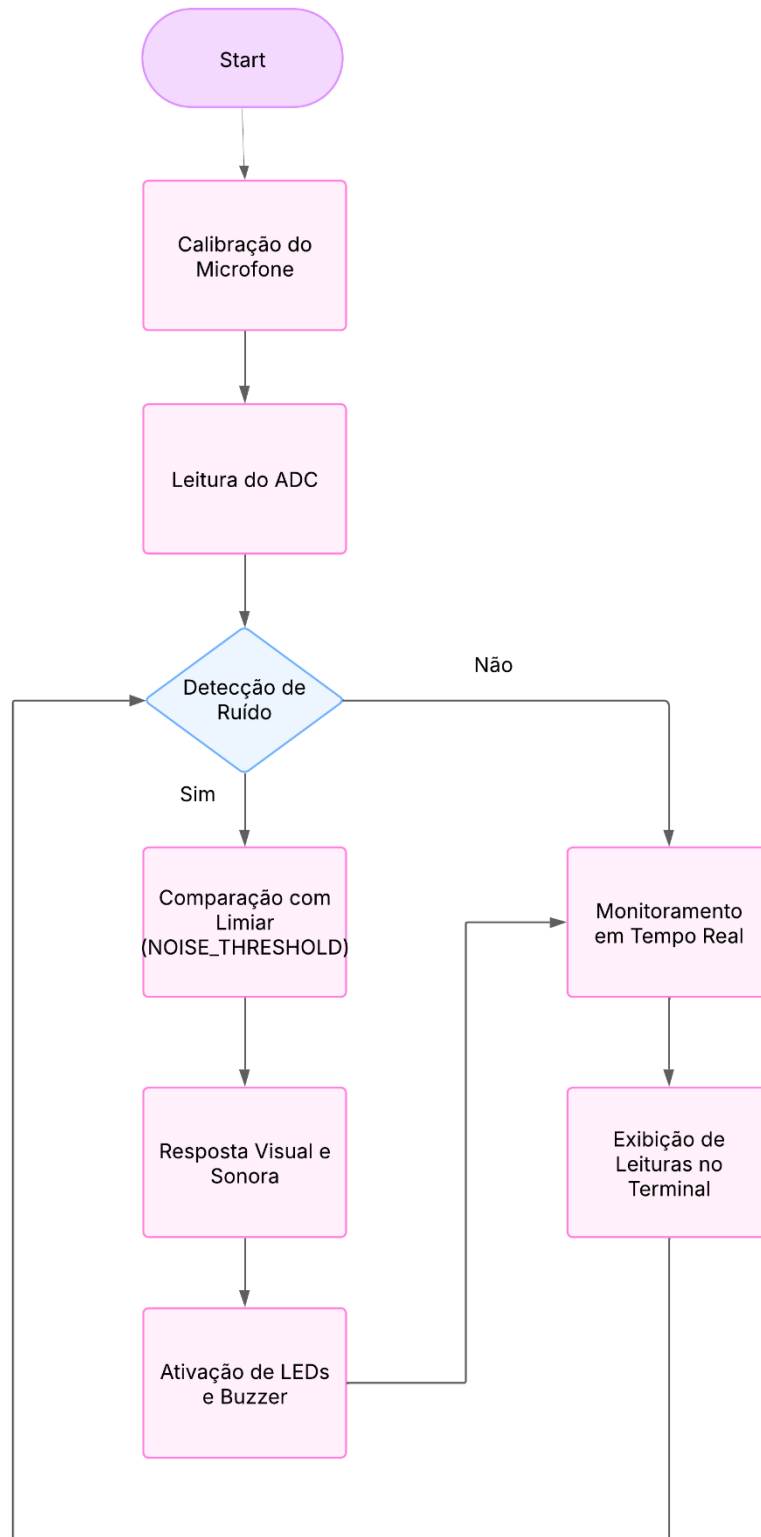
Pino GPIO	Função	Componente Conectado
MICROPHONE_PIN (28)	Entrada analógica (ADC2)	Microfone analógico
LED_PIN (13)	Saída digital	LED indicador
RGB_LED_PIN (11)	Saída digital	LED RGB
MATRIX_LED_PIN (7)	Saída controlada via PIO	Matriz de LEDs
BUZZER_PIN (21)	Saída PWM (slice PWM 4)	Buzzer

Circuito Completo do Hardware:



3. Software

Blocos Funcionais:



Descrição das Funcionalidades:

- **Calibração do Microfone:** Ajusta automaticamente o offset do ADC.
- **Deteção de Ruído:** Calcula a amplitude do som e compara com o limiar.
- **Resposta Visual e Sonora:** Ativa LEDs e buzzer quando o ruído é detectado. A matriz de LEDs exibe um coração verde em modo padrão e uma letra 'X' durante a deteção de ruído.
- **Monitoramento em Tempo Real:** O código implementa um monitoramento em tempo real via terminal, exibindo as leituras atuais do microfone. Esse recurso auxilia no ajuste do **NOISE_THRESHOLD**, permitindo que o usuário configure o limiar de deteção de ruído com base nos valores observados na saída.

Definição das Variáveis:

- **MIC_OFFSET:** Offset do ADC calculado durante a calibração.
- **NOISE_THRESHOLD:** Limiar de deteção de ruído.
- **DETECTION_DURATION:** Duração da resposta visual e sonora.

Uso do PIO:

O **PIO (Programmable Input/Output)** da Raspberry Pi Pico foi utilizado para controlar a matriz de LEDs e os LEDs RGB. O PIO permite criar máquinas de estado programáveis diretamente no hardware, liberando o processador principal para outras tarefas. No caso deste projeto, o PIO foi usado para garantir um controle preciso e otimizado desses componentes.

A implementação do PIO pode ser observada nas seguintes linhas do código:

```
// Configuração do PIO para controlar a matriz de LEDs
```

```
pio_add_program(pio, &matrix_led_program);
```

```
// Inicialização do PIO para controle dos LEDs RGB
```

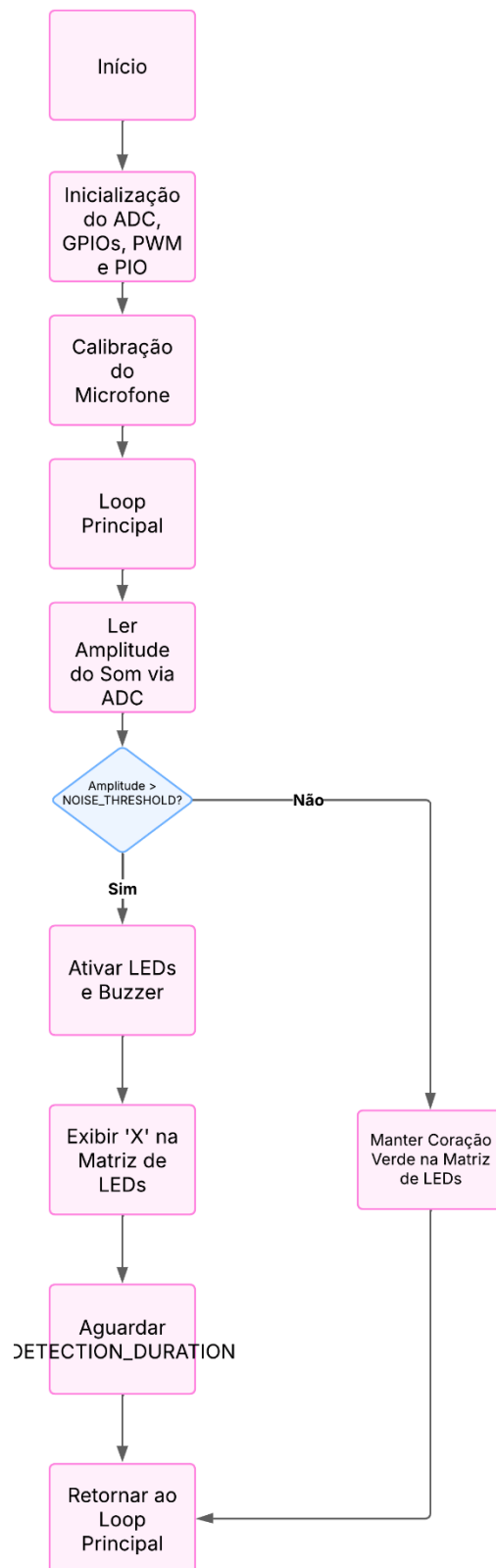
```
pio_sm_set_pins_with_mask(pio, sm, rgb_pins, pin_mask);
```

Essas linhas configuram os programas PIO para gerenciar os sinais enviados aos LEDs, permitindo exibir padrões visuais como o coração verde e a letra 'X'.

O arquivo **ws2818b.pio** contém o **código Assembly específico para o PIO, que:**

- Gerar pulsos de clock com tempos específicos.
- Codificar os bits de dados (0 ou 1) no formato exigido pelo WS2812B.
- Enviar os dados para os LEDs de forma sequencial.

Fluxograma:



Inicialização:

Configuração do ADC, GPIOs, PWM e PIO.

Configurações dos Registros:

Registradores configurados para uso do ADC, PWM e PIO.

Estrutura e Formato dos Dados:

Os dados são processados em buffers de amostras (**adc_values**).

Organização da Memória:

Memória RAM usada para armazenar variáveis temporárias.

4. Execução do Projeto**Metodologia:**

- Pesquisa inicial sobre projetos semelhantes no GitHub
- Desenvolvimento de Software eficiente e rápido, com 193 linhas fazendo uso de Pio, ADC e Pwm.
- Escolha da BitDogLab que usa o Raspberry Pi Pico para teste e validação.
- Sobre as definição das funcionalidades e implementação era desejável que o conjunto software-hardware fosse capaz de converter o sinal analógico do microfone em valores digitais e processá-los de forma eficiente. Além disso, deveria ser capaz de verificar se a amplitude do som ultrapassa o limiar estabelecido e, a partir daí, tomar decisões lógicas para acionar os periféricos adequados, como os LEDs, através do PIO, e o buzzer, utilizando o PWM. Também seria desejável que as leituras do ADC fossem exibidas no terminal do Visual Studio Code, proporcionando um meio de monitoramento contínuo e ajustes precisos no sistema, garantindo assim um funcionamento otimizado e uma interação intuitiva com o ambiente.
- Foi utilizado o ambiente de desenvolvimento de software da Raspberry Pi Pico no Visual Studio Code no Windows. Sucessivas compilações foram necessárias para corrigir deficiências em estruturas do código, até chegar na solução final que se mostrou eficiente.

Testes de Validação:

- Testes de detecção de ruído com diferentes níveis de som.
- Verificação do comportamento dos LEDs e do buzzer.
- Foram realizados testes em ambientes silenciosos durante o período antelucano, a fim de garantir que o sistema não gerasse falsos alertas. Além disso, foram feitos testes com ondas senoidais utilizando um smartphone instalado com um aplicativo específico para esse fim. Os resultados mostraram-se promissores, com uma assertividade do sistema embarcado na faixa de 99%.

Discussão dos Resultados:

O sistema funcionou conforme esperado, com detecção precisa de ruído e respostas rápidas. A matriz de LEDs exibiu o coração verde em modo padrão e a letra 'X' durante a detecção de ruído, proporcionando um feedback visual intuitivo. A aplicação específica do sistema para monitoramento de ruído em ambientes silenciosos demonstra sua **originalidade e inovação**, já

que soluções desse tipo ainda são raras no mercado.

Além disso, o sistema também dispensa a necessidade de que alguém peça silêncio, evitando situações embaraçosas entre os envolvidos. Um exemplo tecnológico que se equipara a esse sistema é uma câmera instalada em um ambiente qualquer que, como função, registra ocorridos de maneira sequencial. Os fatos por ela captados, por si só, comprovam as ocorrências, sendo que o sistema age de maneira objetiva, sem distorções. O monitoramento imparcial do sistema embarcado pode gerar resultados positivos, proporcionando um ambiente mais confortável e eficiente, promovendo uma experiência positiva para todos os envolvidos.

Vídeo Demonstrativo:

Link do Vídeo no YouTube: <https://youtu.be/eDScS83IVHq>

Referências

1. Repositório da BitDogLab no GitHub: [Pico SDK applications for the BitDogLab](#)
2. CUGNASCA, C. E. *Projetos de Sistema Embarcados* .
3. EMOÇÃO: efeitos sobre a voz e a fala na situação em público
Disponível em: <https://tede2.pucsp.br/bitstream/handle/11854/1/RenataBarbosa.pdf>
4. Documentação Raspberry pi pico Oficial
<https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html>
5. Link do Repositório deste projeto: https://github.com/JorgeWilker/U7T_JWMA