Auxiliar de locomoção para deficientes visuais

Elias de Almeida Sombra Neto[1], **Isabelly Pinheiro da Costa**[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - Campus Maracanaú
Av. Parque Central, 1315 - Distrito Industrial I, Maracanaú-CE, Brasil

elias.almeida09@aluno.ifce.edu.br[1], isabelly.pinheiro.costa08@aluno.ifce.edu.br[2]

Abstract. *Este artigo apresenta o desenvolvimento de um dispositivo eletrônico baseado no microcontrolador ESP32 com o objetivo de auxiliar a locomoção de pessoas com deficiência visual. O sistema utiliza um sensor ultrassônico para medir a distância de obstáculos à frente do usuário e, com base na proximidade detectada, fornece feedback tátil através de um motor vibratório e sonoro por meio de mensagens de áudio reproduzidas por um módulo MP3. O dispositivo é portátil, energizado por uma bateria ou power bank, e pode ser controlado remotamente por um bot do Telegram. O projeto visa promover a acessibilidade, autonomia e segurança de seus usuários.*

Resumo. *Este artigo apresenta o desenvolvimento de um dispositivo eletrônico baseado no microcontrolador ESP32 com o objetivo de auxiliar a locomoção de pessoas com deficiência visual. O sistema utiliza um sensor ultrassônico para medir a distância de obstáculos à frente do usuário e, com base na proximidade detectada, fornece feedback tátil através de um motor vibratório e sonoro por meio de mensagens de áudio reproduzidas por um módulo MP3. O dispositivo é portátil, energizado por uma bateria ou power bank, e pode ser controlado remotamente por um bot do Telegram. O projeto visa promover a acessibilidade, autonomia e segurança de seus usuários.*

1. Introdução

É comum que pessoas com deficiência visual necessitem de ajuda para se locomover entre locais, sejam eles abertos ou fechados. A ausência de pisos táteis ou até mesmo de pessoas capacitadas para desempenhar esse papel são constantemente frequentes.

Dessa forma, vimos a possibilidade de desenvolver um projeto com o microcontrolador ESP32 que executa essa função de auxiliar para o deslocamento de deficientes visuais. O intuito é medir distância de obstáculos à frente do usuário e retornar a proximidade através de áudio e de pulsos vibratórios. Assim, possibilitando uma locomoção menos dependente e tornando o ambiente mais acessível.

Sendo assim, este artigo descreve a construção de um protótipo para auxiliar na locomoção de deficientes visuais. Seu funcionamento consiste em um sensor que realiza a medição da distância de obstáculos adiante. Dependendo do nível de proximidade, um áudio de aviso é reproduzido através de um módulo MP3 que está ligado a um alto-falante. Além disso, um motor acoplado ao projeto vibra com mais intensidade quanto mais próximo o usuário se encontra de um objeto.

Os objetivos desse projeto consistem em:

- Projetar um dispositivo auxiliar de locomoção. O protótipo deve conter um botão que liga e desliga o circuito, um sensor de distância para realizar medições, um módulo MP3 ligado a um alto-falante para reproduzir áudio indicando o nível de proximidade e um motor vibratório para feedback tátil.
- Desenvolver um código na linguagem C++ que deve ser gravado no ESP32.
- Criar e configurar um bot do Telegram que recebe comandos como visualizar e limpar logs de medições do sensor, configurar voz e volume do módulo MP3 e acessar um servidor web via protocolo HTTP para realizar atualização Over-The-Air (OTA) e visualizar os registros do projeto.

2. Fundamentação teórica

Pessoas com deficiência visual enfrentam diversos desafios relacionados à mobilidade, especialmente em ambientes desconhecidos. Diversos estudos propõem soluções baseadas em sensores ultrassônicos, vibração e reconhecimento de voz para aumentar a percepção espacial desses usuários.

O sensor HC-SR04 é comumente utilizado em projetos de robótica e automação para medir distâncias com precisão razoável e baixo custo. O ESP32, por sua vez, oferece grande capacidade de processamento, conectividade Wi-Fi/Bluetooth e suporte a arquivos via sistemas como o LittleFS.

A metodologia aplicada baseia-se em um sistema embarcado reativo, onde eventos como a leitura de distância acionam imediatamente atuadores — como o motor ou o módulo MP3. O uso de tarefas paralelas no ESP32 permite que o sistema responda simultaneamente à entrada dos sensores, comunicação com o bot do Telegram e atualizações OTA.

Projetos similares descritos na literatura e em plataformas como Hackster.io e GitHub geralmente utilizam sensores ultrassônicos integrados a motores vibratórios ou módulos de áudio para fornecer feedback sobre obstáculos. No entanto, a maioria desses projetos limita-se a funcionalidades locais e não contempla recursos avançados como controle remoto via bot do Telegram, registro persistente de medições em sistema de arquivos, nem atualizações remotas via OTA. Assim, o presente trabalho se destaca ao propor uma solução mais completa, portátil e conectada, oferecendo um diferencial significativo em termos de acessibilidade e manutenção remota.

3. Metodologia

3.1. Componentes utilizados

Os materiais para a construção e acionamento do circuito incluem:

- ESP32;
- Protoboard;
- LED verde;
- Botão;
- Sensor de distância HC-SR04;
- Módulo motor vibratório;
- Módulo MP3 DFPlayer Mini;
- Alto-falante;
- Cabos jumper.

3.2. Montagem do circuito

Para realizar a montagem do circuito, é preciso seguir os seguintes passos:

- Posicionar devidamente o ESP32, o LED, o botão, o sensor de distância, o motor de vibração e o módulo MP3 na protoboard;
- GND do ESP32 ligado na protoboard;
- VIN do ESP32 ligado na protoboard;
- Pino comum do LED verde (Power) ligado no GND do ESP32;
- Pino positivo do LED verde (Power) ligado no GPIO13 do ESP32;
- Botão (Power) ligado no GND do ESP32;
- Botão (Power) ligado no GPIO12 do ESP32;
- VCC do sensor de distância ligado no VIN do ESP32;
- Trigger do sensor de distância ligado no GPIO5 do ESP32;
- Echo do sensor de distância ligado no GPIO18 do ESP32;
- GND do sensor de distância ligado no GND do ESP32;
- IN do motor de vibração ligado no GPIO25 do ESP32;
- VCC do motor de vibração ligado no VIN do ESP32;
- GND do motor de vibração ligado no GND do ESP32;
- VCC do módulo MP3 ligado no VIN do ESP32;
- RX do módulo MP3 ligado no GPIO26 do ESP32;
- TX do módulo MP3 ligado no GPIO27 do ESP32;
- GND do módulo MP3 ligado no GND do ESP32;
- Ligar os pinos do alto falante no módulo MP3.

3.3. Funcionamento

O circuito inicia desligado, com o ESP32 em modo *Deep Sleep*. Ao pressionar o botão de energia, o ESP32 é ativado, acende o LED verde, conecta-se a uma rede Wi-Fi previamente configurada e começa a realizar leituras com o sensor HC-SR04.

As medições do sensor são realizadas 101 vezes, e a mediana desses valores é calculada para garantir maior precisão na detecção da distância. Com base nessa mediana, o sistema classifica a distância detectada em uma das três categorias:

- **Próximo**: 0 a 10 cm
- **Aproximando**: 10 a 20 cm
- **Distante**: 20 a 30 cm

Essas faixas podem ser facilmente ajustadas por meio de um *bot* no Telegram.

Conforme a categoria detectada, o sistema emite um som no alto-falante e uma vibração de intensidade proporcional:

- **Distante**: vibração de intensidade 255
- **Aproximando**: vibração de intensidade 190
- **Próximo**: vibração de intensidade 130

Ao pressionar o botão novamente, o LED se apaga e o sistema retorna ao modo *Deep Sleep*.

3.4. Desenvolvimento do código

O código foi escrito com a linguagem C++ no software Visual Studio Code com uso da extensão PlatformIO. Para manter um sistema de versionamento foi utilizado o Git e para enviar as alterações para repositório remoto foi utilizado o GitHub. O link para acesso ao código completo está disponível em https://github.com/eliasdeallmeida/auxiliar-de-locomocao.

3.5. Segmentação do código

Dadas as proporções do projeto, viu-se a necessidade de segmentar o código em diretórios para facilitar o desenvolvimento e manutenção. A organização dos arquivos no repositório é descrita na Figura 2, onde as pastas são separadas em:

- /actuators: Referente aos atuadores módulo MP3 e motor vibratório.
- /config: Configurações de módulos do circuito.
- /hardware: Configurações das pinagens.
- /logs: Referente ao registro de logs em arquivo com LittleFS.
- /ota: Define o web server para atualização OTA.
- /power: Referente ao estado de Deep Sleep.
- /sensors: Funções de leitura do sensor de distância.
- /tasks: Tasks criadas para executar nos núcleos do ESP32.
- main.cpp: Arquivo princpal do projeto.

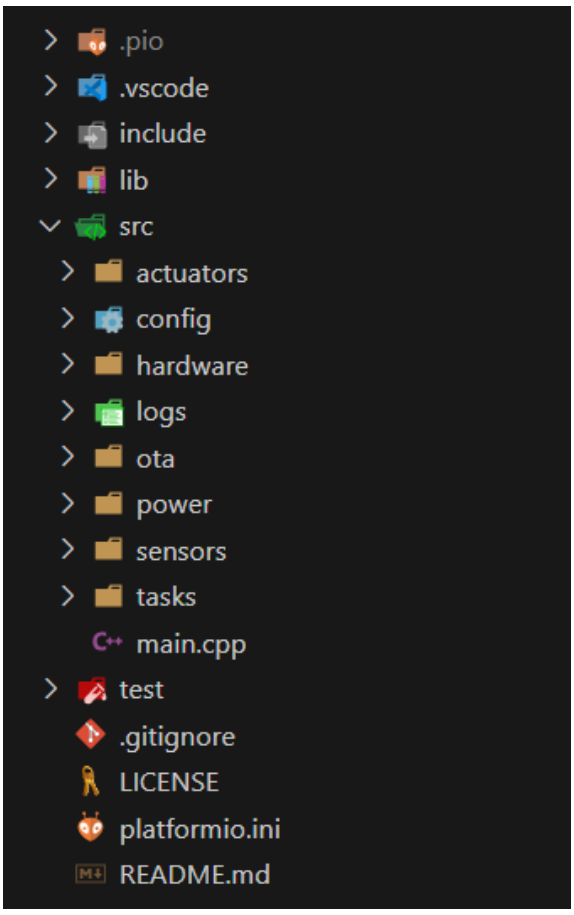

Figure 1. Estrutura de diretórios do código

3.6. Implementação de tasks

O código do sistema foi modularizado em tarefas independentes, utilizando os recursos de multitarefa (FreeRTOS) disponíveis no ESP32. Cada tarefa é responsável por um aspecto específico do funcionamento do dispositivo, favorecendo a organização do código e a execução paralela.

As principais tarefas estão divididas da seguinte forma:

- **Distance Task**: realiza leituras contínuas com o sensor ultrassônico HC-SR04. A cada nova leitura, calcula a distância, envia o valor para o sistema de logs, aciona o motor vibratório com intensidade proporcional à proximidade detectada e define o áudio a ser reproduzido.
- **Button Task**: gerencia o botão físico de acionamento do sistema. Quando o botão é pressionado, a tarefa alterna entre ligar/desligar o sistema, controlando o LED indicador e acionando ou suspendendo as demais tarefas, além de colocar o ESP32 em modo Deep Sleep.
- **Telegram Task**: estabelece conexão com a internet via Wi-Fi e monitora comandos recebidos de um bot no Telegram. Os comandos disponíveis incluem: `/log`, `/clearlogs`, `/voz`, `/volume`, `/ota` e `/otastatus`.
- **MP3 Task**: gerencia a reprodução de áudios utilizando o módulo DFPlayer Mini. Reproduz os arquivos correspondentes aos estados ”próximo”, ”aproximando” e ”distante”, com base na distância recebida da distância lida.
- **OTA Task**: inicia um servidor web OTA (Over-The-Air), permitindo que o firmware do dispositivo seja atualizado remotamente pelo navegador ao enviar o arquivo `.bin`. A task acompanha o progresso e exibe o status da atualização.

Essa arquitetura baseada em tarefas independentes permite que o sistema reaja rapidamente a estímulos do ambiente, processando leitura de sensores, atuação de motores, reprodução de áudio e comunicação em paralelo, sem travamentos ou atrasos significativos.

3.7. Bot do Telegram

Em paralelo, um bot no Telegram permite visualizar logs, apagar logs, ajustar volume do alto falante, trocar gênero da voz que faz leitura da proximidade de obstáculos, realizar atualização OTA e consultar o status da atualização OTA. Os comandos permitidos estão descritos na Figura 1.

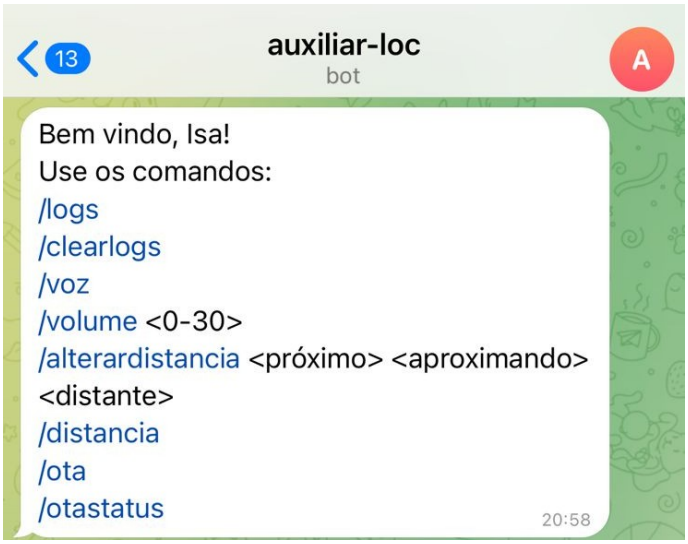

Figure 2. Comandos implementados no bot do Telegram

3.8. Servidor Web para OTA e visualização de logs

Além da comunicação via bot do Telegram, o dispositivo implementa um servidor web acessível pela rede Wi-Fi local, baseado no protocolo HTTP. Esse servidor permite que usuários realizem duas operações principais diretamente por um navegador, sem a necessidade de conexão via cabo ou uso de terminal:

- **Atualização OTA (Over-the-Air):** o servidor expõe uma página HTML simples com um formulário que permite selecionar e enviar um arquivo `.bin` contendo uma nova versão do firmware. Após o envio, o ESP32 realiza a atualização automaticamente e reinicia. O processo é assíncrono, acompanhado por mensagens de status como “atualizando”, “sucesso” ou “erro”.
- **Visualização de logs:** também é possível acessar, por meio de rota dedicada, os registros salvos no arquivo `logs.txt`. Esses registros incluem informações sobre medições de distância, comandos recebidos do bot e mudanças de estado do sistema. A visualização em formato HTML torna mais acessível a análise dos eventos ocorridos, sem depender de acesso serial.

Essa funcionalidade oferece uma interface amigável, que pode ser conferida na figuras 3 e 4, para manutenção do dispositivo, especialmente em ambientes onde não há acesso físico fácil ao microcontrolador. O recurso OTA facilita a entrega de atualizações ou correções de bugs de forma remota, enquanto a visualização dos logs auxilia no diagnóstico do comportamento do sistema ao longo do tempo.

Figure 3. Interface do web server para atualização OTA

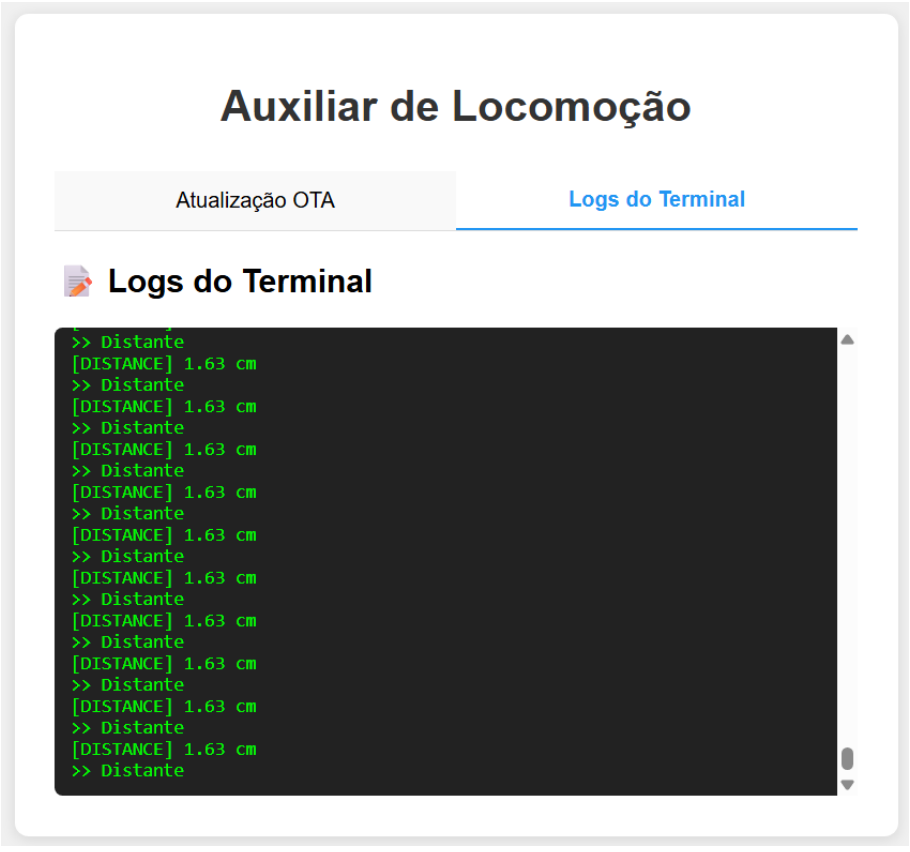

Figure 4. Interface do web server para visualizar registros

3.9. Fluxograma

Na Figura 5, podemos visualizar o fluxograma de funcionamento do projeto com todas as suas vertentes .

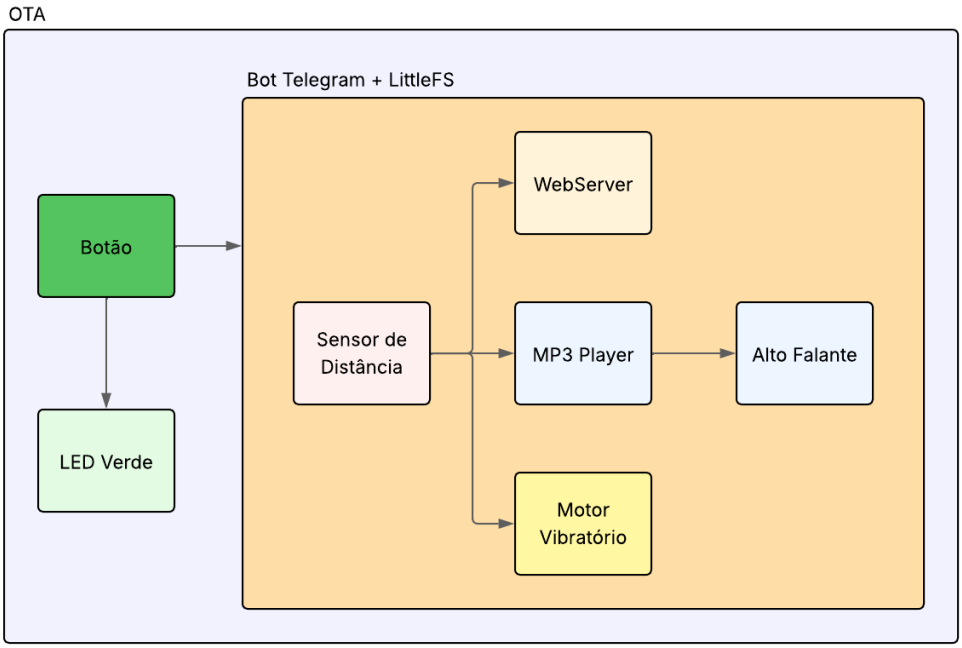

Figure 5. Fluxograma do funcionamento do projeto

4. Experimentos

Para validar o funcionamento do dispositivo proposto, foram realizados experimentos práticos com o protótipo em ambiente controlado. O objetivo principal foi verificar a eficácia do sistema na detecção de obstáculos, bem como a resposta dos atuadores (vibração e áudio) conforme a distância medida pelo sensor ultrassônico HC-SR04. A visualização da montagem do circuito está na Figura 6.

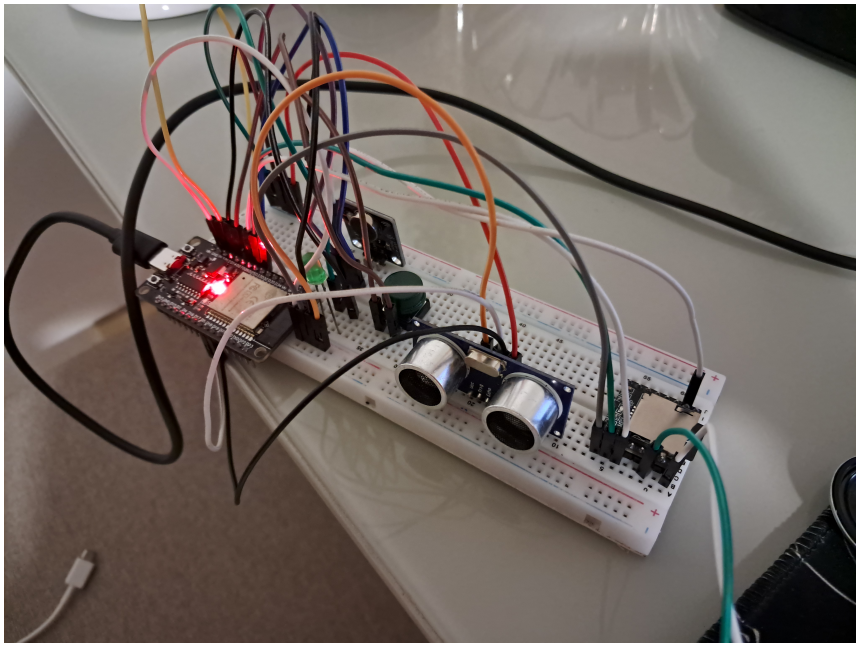

Figure 6. Montagem do circuito

4.1. Configuração do Ambiente de Teste

Os testes foram realizados em um quarto com iluminação padrão e superfícies planas, visando simular um ambiente realista para a locomoção de um usuário. Obstáculos foram posicionados a diferentes distâncias ao longo do trajeto e avaliado se ele estava detectando os objetos.

O ESP32 foi alimentado por um power bank de 5V. O bot do Telegram e os logs por meio do webserver foram acessado por smartphone conectado à mesma rede Wi-Fi do microcontrolador.

4.2. Resultados de Detecção de Obstáculos

Foram posicionados obstáculos em três faixas de distância previamente definidas:

- **Distância curta (até 10 cm):** Áudio “próximo”, vibração forte (255).
- **Distância média (de 10 cm até 20 cm):** Áudio “aproximando”, vibração moderada (190).
- **Distância longa (de 20 cm até 30 cm):** Áudio “distante”, vibração fraca (130).
- **Distância fora de alcance (acima de 30 cm):** Sem áudio e sem vibração.

A Tabela 1 apresenta uma amostragem de medições coletadas e as respostas associadas.

Table 1. Faixas de distância e respostas do sistema

Distância (cm)	Feedback tátil	Áudio reproduzido	Classificação
6.43	Forte	”Próximo”	Curta
9.85	Forte	”Próximo”	Curta
12.27	Moderado	”Aproximando”	Média
18.42	Moderado	”Aproximando”	Média
24.65	Fraco	”Distante”	Longa
28.91	Fraco	”Distante”	Longa
36.03	Nenhum	Nenhum	Fora de alcance

Os resultados indicam que o sistema responde corretamente às variações de distância, ativando os atuadores de forma coerente com a lógica programada.

4.3. Testes com Comandos do Bot do Telegram

Durante os testes, também foram avaliados os comandos enviados por meio do bot do Telegram. Todos os comandos operaram conforme esperado, como descrito na Tabela 2.

Table 2. Resultados da execução dos comandos do bot

Comando	Função	Status
/log	Visualização dos logs do sistema	Sucesso
/clearlogs	Limpeza do arquivo de logs	Sucesso
/volume 25	Ajuste do volume de reprodução para 25	Sucesso
/voz	Alteração da voz para feminina se estiver definido como masculino no momento do comando	Sucesso
/ota	Mostra informações para o acesso da atualização OTA e logs na web	Sucesso
/otastatus	Retorno do status atual da atualização OTA	Sucesso
/distancia	Retorna as distâncias atuais equivalentes ao próximo, aproximando e distante	Sucesso
/alterardistancia <x, y, z>	Define as distâncias equivalentes ao próximo, aproximando e distante, respectivamente	Sucesso

Os testes confirmaram que o sistema mantém a estabilidade da conexão Wi-Fi durante toda a execução, mesmo ao alternar entre tarefas sensoriais e de comunicação.

4.4. Atualização OTA e Visualização de Logs via Web

A funcionalidade de atualização OTA via navegador foi testada com sucesso. Após o envio de um novo arquivo `.bin` pela interface Web, o dispositivo reiniciou com a nova versão do firmware. A visualização dos registros armazenados no arquivo `logs.txt` também funcionou adequadamente, mostrando medições anteriores e comandos recebidos.

4.5. Análise Geral do Desempenho

Os principais pontos observados durante os experimentos foram:

- O sensor HC-SR04 apresentou boa precisão em superfícies planas, mas ruídos ou superfícies irregulares podem afetar a leitura, além de alguns problemas em relação a defeito que ele apresentou em certo ponto dos experimentos;
- O tempo de resposta entre a detecção e o feedback (tátil/sonoro) é bem rápido;
- A autonomia do sistema com power bank auxilia a mobilidade que o projeto necessita;
- A multitarefa no ESP32 mostrou-se eficaz em manter a estabilidade mesmo com múltiplos eventos simultâneos.

Video demonstração de um experimento simples: https://youtu.be/Mk0-chUzdMo.

5. Conclusão

O desenvolvimento deste projeto demonstrou ser uma alternativa viável e de baixo custo para auxiliar na locomoção de pessoas com deficiência visual. A integração entre sensor de distância, motor vibratório, módulo MP3 e ESP32 permitiu construir um sistema capaz de fornecer feedback tátil e sonoro, facilitando a percepção de obstáculos no ambiente.

Durante os testes, o dispositivo respondeu adequadamente às diferentes faixas de distância, mas com ressalvas sobre a sensibilidade do sensor e possibilidade de ruídos externos. O uso do modo Deep Sleep contribui para a economia de energia, fundamental para o uso com baterias. A integração com o bot do Telegram amplia as possibilidades de controle e monitoramento remoto.

Como trabalhos futuros, pretende-se:

- Desenvolver uma estrutura física mais compacta e vestível;
- Botão que permite configuração do projeto via Telegram sem ligar o sensor e outros atuadores do circuito.

6. Referências

- Bastos, L. et al. (2017). Sistema de auxílio à locomoção de deficientes visuais com sensores ultrassônicos. Revista de Iniciação Científica da FATEC.
- Silva, J. et al. (2019). Sistemas embarcados: uma abordagem prática com ESP32. Instituto Federal de Educação, Ciência e Tecnologia.
- Amaral, A. (2021). Programando o ESP32 com Arduino IDE. Novatec.
- DFPlayer Mini MP3 – https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299
- LittleFS – https://github.com/lorol/arduino-esp32littlefs
- ESP32 Documentation – https://docs.espressif.com