

Apresentação

Este projeto consiste em um Sistema de Treinamento Cognitivo projetado para auxiliar no desenvolvimento da memória, reflexos e coordenação motora de indivíduos, como idosos em reabilitação cognitiva ou pacientes em recuperação de lesões cerebrais. Por meio de uma interação lúdica com feedback visual e sonoro, o sistema apresenta sequências de cores e sons que devem ser reproduzidas pelo usuário, promovendo um ambiente de treino acessível e intuitivo.

Genius Terapêutico Interativo: Sistema de Treinamento Cognitivo e Reflexos

Objetivo

Desenvolver um sistema interativo que combina elementos visuais e sonoros para estimular a memória, os reflexos e a coordenação motora. A proposta é criar uma ferramenta acessível e versátil, voltada para o desenvolvimento cognitivo de usuários, como idosos ou pessoas em reabilitação. O sistema oferece partidas ajustáveis de 1 a 10 rodadas, promovendo concentração e engajamento, e proporcionando uma experiência personalizada e eficaz para atender às necessidades individuais dos usuários

Requisitos

1. **Desenvolver um sistema de treinamento cognitivo para memória e reflexos:**
O cliente pediu um sistema que ajudasse no desenvolvimento cognitivo de usuários, como idosos ou pessoas em reabilitação.
2. **Interface interativa e acessível:**
O cliente desejava que o sistema fosse intuitivo, interativo e simples de usar, mesmo para pessoas com pouca habilidade com tecnologia.
3. **Flexibilidade no número de rodadas (de 1 a 10):**
O cliente queria permitir que o usuário pudesse ajustar a dificuldade, escolhendo a quantidade de rodadas que desejasse jogar.

4. **Feedback visual e sonoro para reforço:**

O cliente solicitou um sistema que fornecesse respostas claras e imediatas, ajudando na correção e motivação.

5. **Portabilidade e autonomia:**

O cliente gostaria que o sistema pudesse ser usado em diferentes contextos, sem configurações complexas.

Funcionamento

O sistema funciona como um jogo no estilo "Genius", no qual o usuário deve observar as sequências de cores emitidas pelo LED RGB e reproduzi-las usando o joystick e os botões integrados à placa. Para cada rodada:

- Uma sequência de cores é exibida e sons associados são emitidos pelo buzzer.
- O usuário reproduz a sequência. Se acertar, avança para a próxima rodada; se errar, o jogo reinicia.
- A dificuldade aumenta conforme o número de rodadas escolhidas. Ao completar as rodadas estipuladas o sistema parabeniza o usuário.

Justificativa

Com o aumento da população idosa e a necessidade de soluções inovadoras para reabilitação cognitiva e motora, este projeto se justifica como uma ferramenta acessível e de baixo custo. Ele une entretenimento e benefícios terapêuticos, promovendo o desenvolvimento de habilidades cognitivas e motoras de forma divertida e interativa.

Originalidade

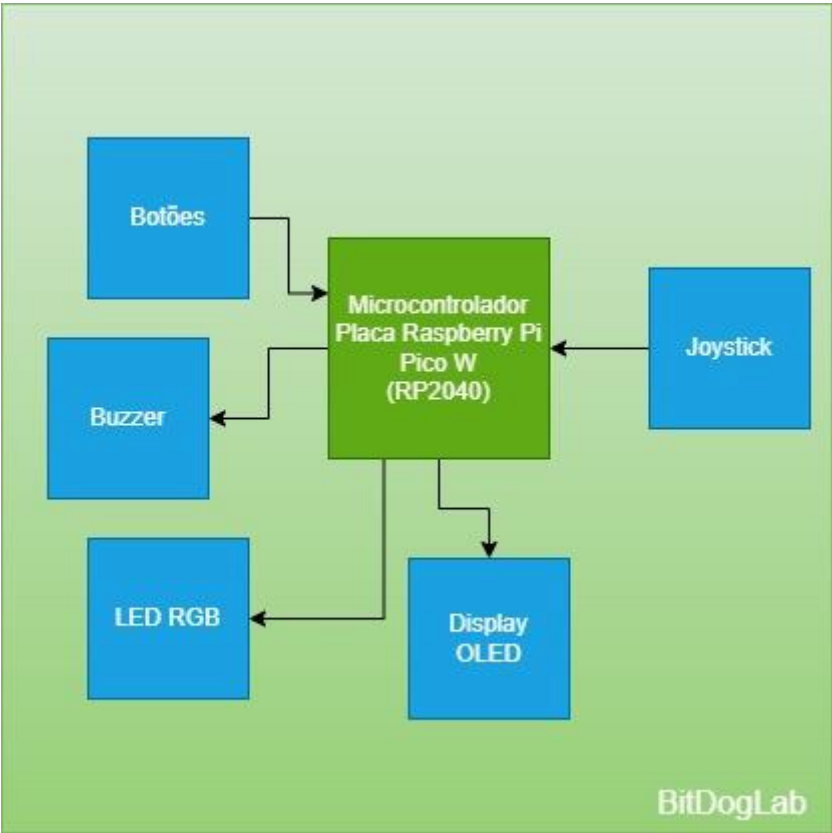
A pesquisa revelou projetos similares, como brinquedos no estilo Genius e softwares voltados para reabilitação cognitiva. No entanto, a originalidade deste projeto reside na combinação de:

1. Um sistema totalmente integrado, sem necessidade de componentes externos.
2. Capacidade de personalizar o número de rodadas, ajustando a dificuldade para diferentes perfis de usuários.
3. Uso simultâneo de feedback visual (LED RGB), sonoro (buzzer) e interação física (joystick e botões), promovendo uma experiência completa de

reabilitação cognitiva e motora. Essa abordagem diferenciada torna o projeto inovador e funcional para aplicações terapêuticas.

Hardware

Diagrama de Blocos



Bloco	Função	Configuração
Microcontrolador	Responsável por gerenciar e coordenar todas as operações do sistema. Processa os sinais de entrada dos periféricos (como joystick e botões) e controla os atuadores (LED RGB, buzzer	Configurado como a unidade de processamento central, conectando periféricos via GPIO e barramento I²C (para o display OLED).






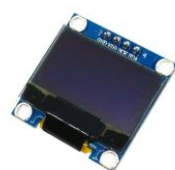

	e display) para garantir o funcionamento integrado do sistema.	
LED RGB	Exibe as cores da sequência que o usuário deve memorizar e as cores selecionadas pelo usuário durante a reprodução da sequência. Serve como feedback visual central do jogo.	Alimentado por três GPIOs do microcontrolador, com controle individual dos canais vermelho, verde e azul
Buzzer	Fornece feedback sonoro, emitindo tons específicos associados a cada cor. Também é utilizado para alertas de erro e efeitos sonoros durante animações.	Conectado ao GPIO com sinal PWM para gerar os sons correspondentes às cores e alertas.
Joystick	Permite ao usuário navegar, selecionar e confirmar as cores durante a reprodução da sequência. É o principal dispositivo de interação do usuário com o sistema	Conexão analógica para leitura dos eixos X e Y, e GPIO digital para leitura do botão.
Botões	Auxiliam na seleção de cores, inicialização do jogo, reinício e desligamento do sistema.	Conexão simples a GPIOs configurados como entrada digital.
Display OLED	Exibe informações relevantes para o usuário, como o número de rodadas, mensagens de sucesso, erro ou instruções. Melhora a experiência do usuário com feedback textual claro.	Conexão via interface I ² C (SDA e SCL) para exibição de mensagens.


Especificações

As especificações técnicas atendem aos requisitos por:

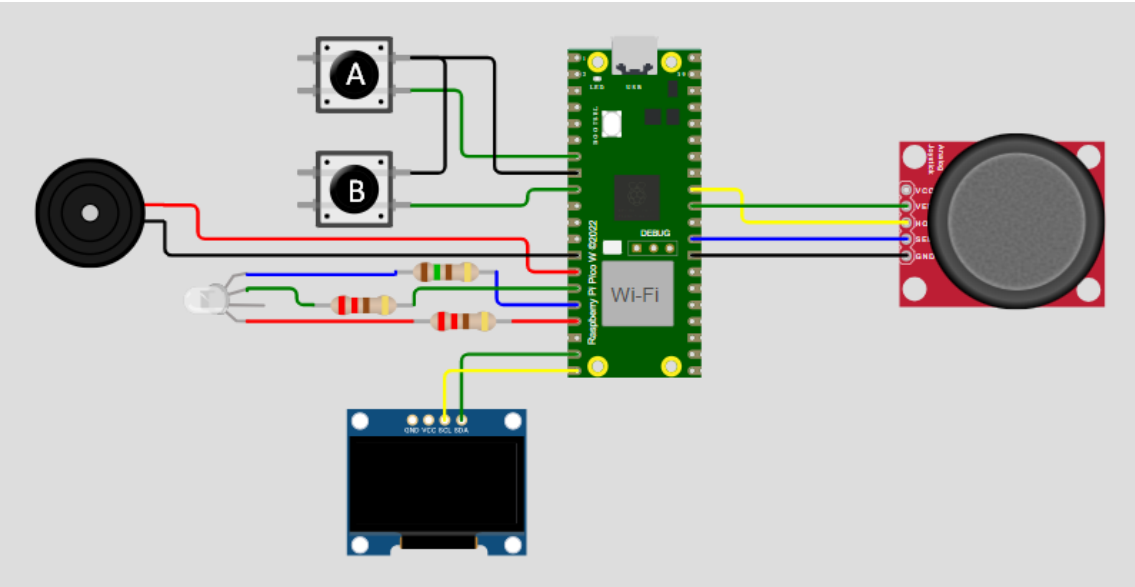
- Proporcionar um sistema acessível e de fácil interação.
- LEDs RGB para feedback visual com três cores básicas e combinação para amarelo e magenta.
- Buzzer que oferece feedback sonoro em frequências distintas para identificar cores e erros.
- Joystick e botões que permitem uma navegação intuitiva.
- Display OLED que mostra informações úteis ao usuário durante o jogo.

Lista de materiais

Componente	Imagem	Quantidade	Descrição	Pino	Função
Raspberry pi Pico W		1	Microcontrolador RP2040 com conectividade Wi-Fi.	-	Unidade central de processamento e controle do sistema
LED RGB		1	LED RGB comum (3 cores: vermelho, verde e azul).	GPIO13 (Vermelho), GPIO11 (Verde), GPIO12 (Azul)	Representação visual de cores durante o jogo.
Buzzer		1	Emissor de som para feedback sonoro.	GPIO10	Emissão de sons associados a eventos, como sequência de cores ou erros.
Joystick		1	Controle analógico com botões de direção.	GPIO26 (VRX), GPIO27 (VRX), GPIO22 (SW)	Controle adicional para navegação ou seleção durante o jogo.
Botões		2	Botões para interação do usuário (seleção de cor e confirmação).	GPIO5 (Botão A), GPIO6 (Botão B)	Controle manual para interagir com o sistema e selecionar cores.
Display OLED		1	Tela OLED para exibição de informações do jogo (pontuação, sequência de cores, etc.).	GPIO14 (SDA), GPIO15 (SCL)	Exibição de informações textuais e visuais sobre o estado do jogo.
Resistor de 220 Ω		2	Resistor para limitar corrente no LED (cores verde e vermelho).	Conectados entre GPIO11/GPIO13 e o LED RGB	Proteção do LED RGB, controlando a corrente para as cores vermelho e verde.

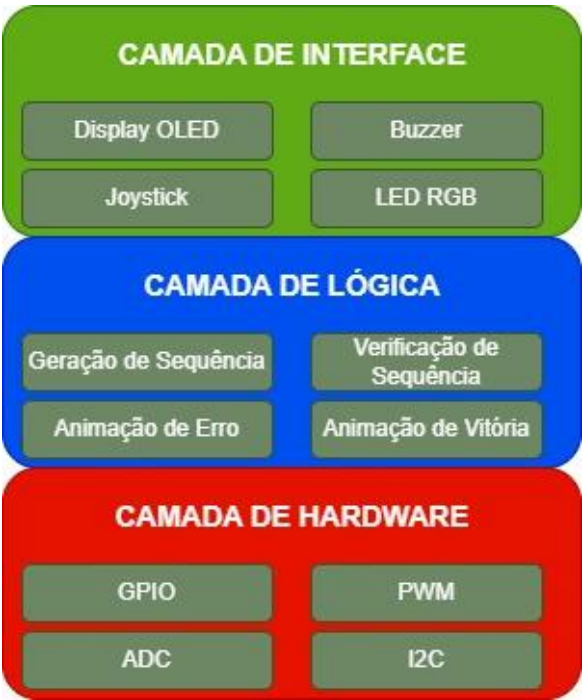
Resistor de 150 Ω		1	Resistor para limitar corrente no LED (cor azul).	Conectado entre GPIO12 e o LED RGB	Proteção do LED RGB, controlando a corrente para a cor azul.
--	---	---	---	------------------------------------	--

Circuito do hardware



Software

Blocos funcionais



Camada	Funções
Interface	<ul style="list-style-type: none"> • Display OLED: Exibe mensagens e informações do jogo. • LEDs RGB: Mostra as cores da sequência e feedback visual. • Buzzer: Fornece feedback auditivo. • Joystick: Permite ao jogador interagir com o jogo.
Lógica	<ul style="list-style-type: none"> • Geração de Sequência: Gera uma sequência aleatória de cores. • Verificação de Sequência: Compara a sequência do jogador com a sequência gerada. • Animação de Vitória: Executa uma animação e toca sons quando o jogador vence. • Animação de Erro: Executa uma animação e toca sons quando o jogador erra.
Hardware	<ul style="list-style-type: none"> • GPIO: Controla os pinos de entrada e saída para LEDs, botões e buzzer. • PWM: Gera sinais PWM para controlar o buzzer e a intensidade dos LEDs. • ADC: Lê os valores analógicos do joystick. • I2C: Comunicação com o display OLED.

Descrição das funcionalidades

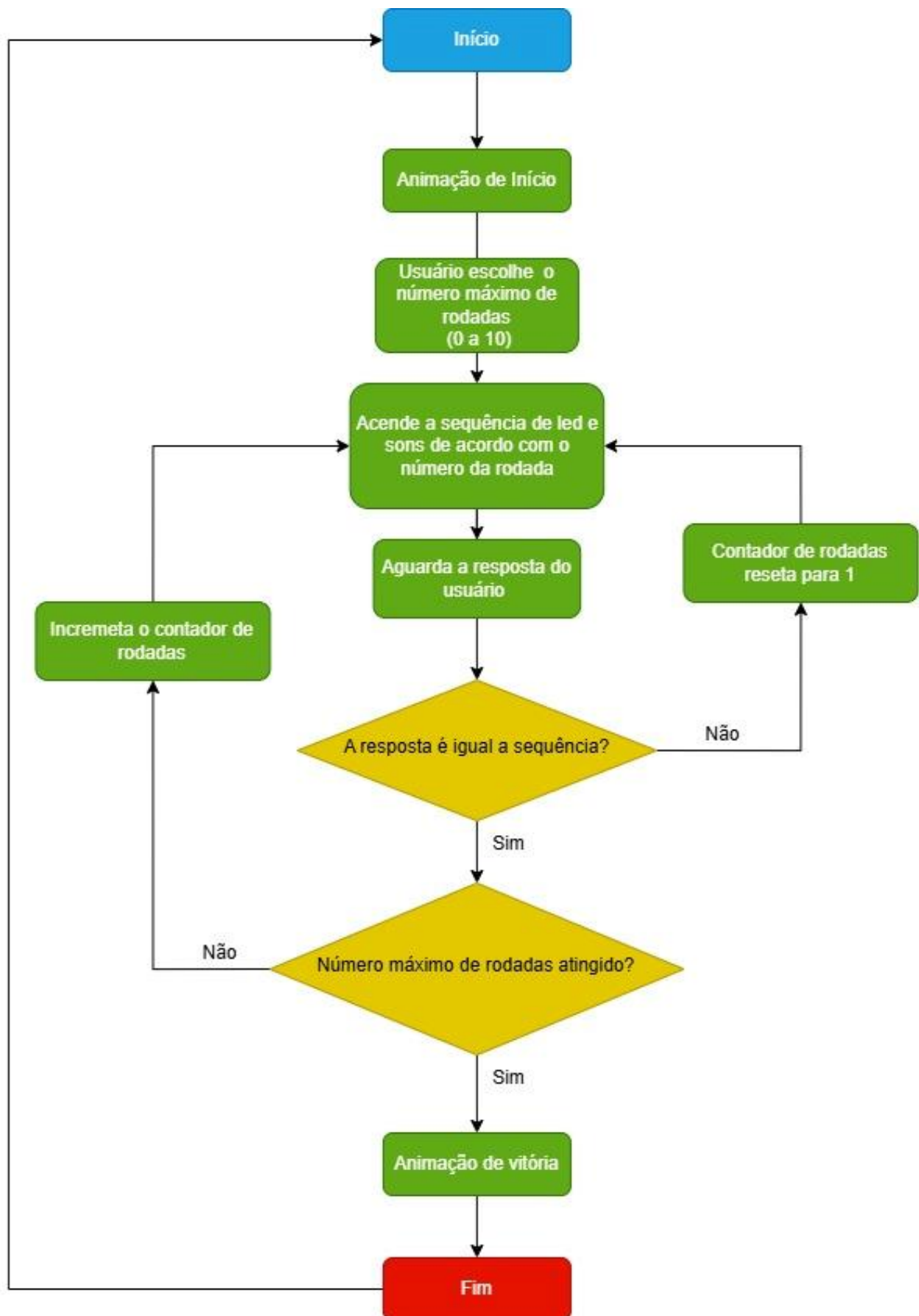
Função	Descrição
<i>set_rgb_color(ColorState color)</i>	Define a cor do LED RGB com base no estado da cor.
<i>play_tone(uint32_t frequency, uint32_t duration_ms)</i>	Toca um tom no buzzer com a frequência e duração especificadas.
<i>play_color_sound(ColorState color)</i>	Toca o som correspondente à cor.
<i>generate_sequence(ColorState sequence[], int length)</i>	Gera uma sequência incremental de cores.
<i>show_sequence(ColorState sequence[], int length)</i>	Mostra a sequência de cores e sons no LED RGB.
<i>check_sequence(ColorState sequence[], ColorState player_sequence[], int length)</i>	Verifica se a sequência do jogador está correta.
<i>victory_animation()</i>	Executa uma animação de vitória com LEDs e sons.
<i>error_animation()</i>	Executa uma animação de erro com LEDs e sons.
<i>display_message(char *message, int line)</i>	Exibe uma mensagem no display OLED.

<i>display_two_messages(char *message1, int line1, char *message2, int line2)</i>	Exibe duas mensagens no display OLED.
<i>startup_animation()</i>	Executa uma animação de inicialização com LEDs e sons.
<i>read_joystick_color()</i>	Lê a cor selecionada pelo joystick.
<i>adjust_rounds()</i>	Ajusta o número de rodadas usando o joystick.
<i>main()</i>	Função principal que inicializa o hardware e gerencia o loop do jogo.

Definição das variáveis

Variável	Descrição
<i>LED_RED_PIN</i>	Pino GPIO para o LED RGB vermelho.
<i>LED_GREEN_PIN</i>	Pino GPIO para o LED RGB verde.
<i>LED_BLUE_PIN</i>	Pino GPIO para o LED RGB azul.
<i>BUZZER_PIN</i>	Pino GPIO para o buzzer.
<i>BUTTON_A_PIN</i>	Pino GPIO para o botão A.
<i>BUTTON_B_PIN</i>	Pino GPIO para o botão B.
<i>JOYSTICK_X_PIN</i>	Pino ADC para o eixo X do joystick.
<i>JOYSTICK_Y_PIN</i>	Pino ADC para o eixo Y do joystick.
<i>I2C_SDA</i>	Pino I2C para a comunicação SDA do display OLED.
<i>I2C_SCL</i>	Pino I2C para a comunicação SCL do display OLED.
<i>slice_num</i>	Variável para controle do número do slice de PWM.
<i>channel</i>	Variável para controle do canal do PWM.
<i>total_rounds</i>	Número de rodadas escolhido pelo usuário.
<i>colorState</i>	Enumeração que define os estados das cores (MAGENTA, GREEN, BLUE, YELLOW).
<i>sequence[]</i>	Array que armazena a sequência gerada pelo sistema.
<i>player_sequence[]</i>	Array que armazena a sequência feita pelo jogador.
<i>sequence_length</i>	Tamanho atual da sequência gerada.
<i>player_index</i>	Índice atual na sequência do jogador.
<i>round</i>	Número da rodada atual.
<i>max_rounds</i>	Número máximo de rodadas completadas sem erro.
<i>game_active</i>	Estado do jogo (ativo ou inativo).

Fluxograma



Código do programa

https://github.com/nmarcal/embarcotech_git/blob/master/Genius_Terapeutico_Cognitivo/Genius_Terapeutico_Cognitivo.c

Inicialização

O processo de inicialização do software é responsável por preparar o sistema embarcado para seu funcionamento, configurando os periféricos e recursos utilizados. Ele segue os passos abaixo:

1. **Reset do Sistema:** O microcontrolador inicia em estado de reset, verificando integridade e inicializando variáveis.
2. **Configuração de GPIOs:** Os pinos utilizados para LEDs, buzzer, botões, joystick e display OLED são configurados como entrada ou saída, dependendo de sua função.
3. **Configuração do Buzzer:** Inicializa o pino como saída e define a frequência padrão do PWM.
4. **Configuração do Display OLED:** Configura a comunicação I²C nos pinos SDA e SCL para inicializar o display OLED.
5. **Inicialização da Lógica de Controle:** As estruturas de dados e variáveis para a lógica do jogo são inicializadas, como sequência de cores e contador de rodadas.
6. **Teste de Hardware:** Um autodiagnóstico verifica se os LEDs, buzzer e botões estão respondendo corretamente.
7. **Pronto para Operação:** Após a inicialização, o sistema entra no estado de espera para interação com o usuário.

Configurações dos registros

As configurações dos registros são feitas para cada periférico do sistema, garantindo que eles operem de maneira correta:

- **GPIOs:**

- Configuração de **GPIO11, GPIO12 e GPIO13** como saídas para o LED RGB.
- Configuração de **GPIO5 e GPIO6** como entradas para os botões.
- Configuração de **GPIO10 e GPIO21** como saída para o buzzer.
- **PWM:** Registro associado ao buzzer configurado para diferentes frequências, representando as notas musicais.
- **I²C:** Configuração dos registradores de comunicação para o display OLED (SDA - GPIO14 e SCL - GPIO15).
- **ADC:** Configuração para leitura do joystick em **GPIO26 (VRY)** e **GPIO27 (VRX)**.
- **SPI:** Se utilizado, configurações de registros para comunicação SPI (pinos GPIO16-GPIO19).

Estrutura e formato dos dados

Os dados manipulados no software possuem as seguintes características:

- **Sequência de Cores:** Array contendo as cores na ordem da sequência gerada para o jogo. Exemplo: `sequencia_cores = ["vermelho", "azul", "verde"]`.
- **Estado do Jogo:** Variável para rastrear o progresso do jogador, como número da rodada ou número de erros.
- **Notas Musicais:** Mapeamento das frequências PWM para sons associados às cores:
 - Vermelho: Dó (261 Hz).
 - Verde: Ré (293 Hz).
 - Azul: Mi (329 Hz).
 - Amarelo: Fá (349 Hz).
- **Interação do Usuário:** Armazena os inputs do jogador em arrays para comparação com a sequência gerada.

Organização da memória

A memória foi organizada da seguinte forma para otimizar o funcionamento do sistema:

- **RAM:**
 - Variáveis dinâmicas: Sequência de cores, progresso do jogo e inputs do jogador.
 - Buffers temporários para comunicação com o display OLED.
- **Flash:**
 - Código do programa.
 - Tabelas de mapeamento (como frequências das notas musicais).
- **Endereços de Periféricos:**
 - GPIOs: Configurados conforme o datasheet da placa BitDogLab.
 - I²C: Endereço padrão do display OLED, geralmente 0x3C.

Protocolo de comunicação

O sistema utiliza o protocolo **I²C** para comunicação com o display OLED. Os detalhes são:

- **Velocidade:** 100 kHz (Standard Mode).
- **Endereçamento:** Endereço de 7 bits (0x3C para o display OLED).
- **Funções:**
 - Escrita de comandos para inicialização e atualização do display.
 - Escrita de dados para exibição de mensagens ao jogador.

Formato de pacote de dados

Os pacotes de dados enviados pelo protocolo I²C para o display OLED seguem o seguinte formato:

1. **Comando de Controle:**
 - Primeiro byte indica se o pacote é um comando (0x00) ou dados (0x40).
2. **Dados de Configuração ou Gráficos:**

- Bytes subsequentes contêm instruções de configuração ou dados para exibição no display.

Execução

Metodologia

Pesquisas Realizadas

Para o desenvolvimento do sistema, foi realizada uma análise detalhada de projetos similares no mercado, como jogos de memória interativos e sistemas cognitivos utilizados em processos de reabilitação. A pesquisa incluiu o estudo de funcionalidades, componentes utilizados e a aplicabilidade desses sistemas em plataformas embarcadas. Além disso, foi feita uma avaliação criteriosa dos recursos disponíveis na placa BitDogLab, assegurando que o projeto fosse viável dentro das limitações dos componentes integrados.

Escolha do Hardware

A escolha da placa BitDogLab foi motivada pela integração de diversos periféricos essenciais, como LED RGB, buzzer, joystick e display OLED. Esses componentes garantem um sistema autossuficiente, eliminando a necessidade de aquisição de periféricos adicionais e simplificando a implementação do projeto.

Definição das Funcionalidades do Software

O software foi projetado para oferecer um sistema de jogo interativo, com as seguintes funcionalidades:

- Geração de sequências aleatórias de cores e sons.
- Captura das interações do usuário através de botões e joystick.
- Fornecimento de feedback visual e sonoro em tempo real.
- Escalabilidade no nível de dificuldade, com a possibilidade de até 10 rodadas.

Inicialização da IDE

A IDE escolhida para o desenvolvimento foi o Visual Studio Code, configurado para a placa Raspberry Pi Pico, conforme orientações do curso. Todas as bibliotecas necessárias foram importadas e configuradas, incluindo suporte para comunicação I²C, controle de GPIOs e emissão de sinais PWM, essenciais para a operação dos periféricos do sistema.

Programação na IDE

O código foi desenvolvido de forma modular, separando as funções principais para facilitar a organização e a depuração. As funções implementadas incluem:

- Controle do LED RGB para exibir as cores.
 - Emissão de sons utilizando o buzzer, com diferentes frequências associadas às cores.
 - Leitura das entradas do joystick e dos botões para capturar interações do usuário.
 - Exibição de informações e feedback no display OLED.
- A lógica principal do jogo foi estruturada para garantir um fluxo claro e fácil de compreender, permitindo ajustes futuros e testes de maneira eficiente.

Depuração

O processo de depuração foi realizado em etapas:

1. Cada periférico foi testado individualmente, garantindo que o LED RGB, o buzzer, o display OLED e outros componentes respondessem corretamente às configurações iniciais.
2. Após a validação dos periféricos, o sistema foi integrado e submetido a testes completos para avaliar seu funcionamento em conjunto.
3. Logs e mensagens exibidas no display OLED foram fundamentais para identificar e corrigir problemas durante a execução do código, assegurando o desempenho ideal do sistema.

Testes de validação

Testes dos Periféricos

Os periféricos integrados foram validados individualmente para garantir que todos os componentes funcionassem conforme o esperado:

- **LED RGB:** Testes de brilho e precisão na exibição das cores (vermelho, verde, azul e combinações).
- **Buzzer:** Emissão de sons com frequências diferentes, associadas a cada cor do LED RGB, para verificar clareza e resposta sonora.
- **Display OLED:** Exibição de mensagens claras e legíveis, com foco na nitidez e no contraste das informações apresentadas.
- **Joystick e Botões:** Captura precisa das entradas para garantir a detecção e resposta correta às interações do usuário.

Testes do Fluxo do Jogo

Foram realizados testes no fluxo principal do jogo para verificar a funcionalidade do sistema:

- **Geração de Sequências:** Verificação da geração correta de sequências aleatórias de cores e sons, assegurando a variabilidade nas jogadas.
- **Comparação das Entradas:** Avaliação da correspondência entre as sequências geradas e as entradas fornecidas pelo jogador, garantindo precisão na lógica do jogo.
- **Feedback Visual e Sonoro:** Verificação do uso do LED RGB e do buzzer para fornecer feedback claro para acertos e erros do jogador.

Testes de Usabilidade

A usabilidade do sistema foi avaliada considerando diferentes níveis de dificuldade e o tempo de resposta do usuário:

- **Níveis de Dificuldade:** O jogo foi testado em níveis que variam de 1 a 10 rodadas, ajustando o tempo entre a exibição das cores para garantir que o usuário tivesse tempo suficiente para acompanhar e interagir.
- **Adaptação do Fluxo:** Pequenos ajustes foram feitos para tornar a experiência mais intuitiva e fluida.

Testes de Desempenho

Por fim, a estabilidade e consistência do sistema foram verificadas por meio de testes de longa duração:

- O sistema foi executado continuamente por mais de uma hora, garantindo que todos os periféricos e a lógica do jogo mantivessem o funcionamento estável e sem falhas ao longo do tempo.

Discussão dos resultados

Confiabilidade do Sistema

Os testes realizados comprovaram que o sistema é altamente confiável e atende plenamente aos objetivos propostos.

- O jogo operou de forma consistente, gerando sequências aleatórias corretas e capturando as entradas do jogador sem apresentar erros ou falhas.
- Nenhum bug crítico foi identificado durante as fases de teste.

Aplicabilidade do Projeto

O projeto demonstrou grande versatilidade e utilidade prática em diferentes contextos, incluindo:

- **Treinamento Cognitivo e Coordenação Motora:** Adequado para diversas faixas etárias, como idosos, crianças ou pacientes em processos de reabilitação cognitiva.
- **Adaptação ao Usuário:** Nível de dificuldade ajustável, permitindo que o sistema seja adaptado às capacidades individuais dos usuários.

Pontos Positivos do Sistema

- **Uso Eficiente de Recursos:** O projeto aproveita de forma otimizada os componentes integrados à placa BitDogLab, explorando suas funcionalidades ao máximo.
- **Interface Simples e Intuitiva:** A interação com o sistema é facilitada por uma interface clara e de fácil entendimento, ideal para usuários de todas as idades.
- **Feedback Multimodal:** O uso combinado de LEDs RGB, sons emitidos pelo buzzer e mensagens no display OLED garante um feedback claro e imediato para o jogador.

Conclusão

Com base nos resultados obtidos, o projeto demonstrou um grande potencial para aplicações práticas, tanto no âmbito do entretenimento quanto no treinamento e reabilitação cognitiva. O sistema cumpre seus objetivos de forma eficaz e apresenta uma base sólida para futuras melhorias ou adaptações.

Vídeo de demonstração

<https://youtu.be/RCcmScVKlQI>

Referências

1. KUO, B. C. *Automatic control systems*. 7. ed. Upper Saddle River: Prentice Hall, 1995.
2. CUGNASCA, Carlos Eduardo. *Sistemas embarcados: aplicações e fundamentos*. São Paulo: Editora Unesp, 2015.
3. BITDOGLAB. *Datasheet e Manual do Usuário*. Disponível em: <https://www.bitdoglab.com/documentation>. Acesso em: 19 fev. 2025.
4. BITDOGLAB. *Repositório no GitHub*. Disponível em: <https://github.com/BitDogLab/BitDogLab>. Acesso em: 19 fev. 2025.
5. EAD.HARDWARE. *Curso sobre sistemas embarcados*. Disponível em: <https://ead.hardware.org.br/course/view.php?id=4§ion=6>. Acesso em: 19 fev. 2025.
6. ADAFRUIT. *Exemplos de projetos com LEDs e microcontroladores*. Disponível em: <https://learn.adafruit.com>. Acesso em: 19 fev. 2025.
7. GRAHN, Jessica A.; ROWE, James B. *Neural mechanisms in cognitive training using memory and coordination games*. *Neuropsychologia*, v. 51, n. 3, p. 489–497, 2013.
8. MICROSOFT. *Visual Studio*. Disponível em: <https://visualstudio.microsoft.com>. Acesso em: 19 fev. 2025.
9. FRITZING. *Software para criação de diagramas de circuitos*. Disponível em: <https://fritzing.org>. Acesso em: 19 fev. 2025.
10. MAKER PRO. *Exemplos de projetos cognitivos com sistemas embarcados*. Disponível em: <https://www.maker.pro>. Acesso em: 19 fev. 2025.
11. SPARKFUN. *Guia de configuração de microcontroladores e periféricos*. Disponível em: <https://learn.sparkfun.com>. Acesso em: 19 fev. 2025.