

Projeto Final
Embarcatech:

CaféScan

Detecção Precoce da Ferrugem do Cafeeiro

ALUNO

JUAN PABLO AZEVEDO SOUSA

MATRICULA

TIC370101123

REPOSITÓRIO

<https://github.com/JotaPablo/projetofinal.git>

O que é o Café Scan ?

O CaféScan é um sistema embarcado inovador que integra sensores ópticos e inteligência artificial para detectar precocemente a ferrugem do cafeeiro (*Hemileia vastatrix*), muitas vezes antes do aparecimento dos sintomas visíveis. Essa solução portátil e de baixo custo foi projetada para atender pequenos e médios produtores rurais, possibilitando intervenções rápidas, redução de perdas na produtividade e uso mais racional de insumos.

OBJETIVOS

Objetivos gerais

Desenvolver um sistema embarcado que permita a detecção precoce da ferrugem do cafeeiro, proporcionando uma ferramenta acessível para produtores que necessitam de intervenções imediatas e eficientes.

Objetivos específicos



DETECÇÃO PRECOCE

Identificar a infecção 10–15 dias antes do surgimento de manchas.



REDUÇÃO DE CUSTOS

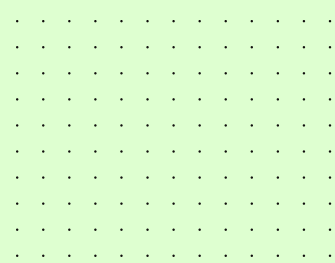
Substituir soluções caras e otimizar uso de fungicidas



DEMOCRATIZAÇÃO

Oferecer tecnologia acessível para pequenos produtores

Funcionamento



Hardware

Sensores Espectrais

Os sensores RGB e NIR capturam a reflectância das folhas, identificando alterações na clorofila (RGB) e danos celulares (NIR) que indicam a presença da ferrugem.

Processamento e Análise

Um microcontrolador processa os dados dos sensores e utiliza um modelo de inteligência artificial para detectar a infecção até 10–15 dias antes do aparecimento dos sintomas.

Feedback Imediato

O sistema comunica o estado da planta através de feedback visual (LED RGB e Display OLED) e sonoro (buzzer), permitindo ao produtor tomar decisões rápidas.

Simulação

Simulação das Leituras

Um joystick é empregado para simular as leituras de reflectância (valores NIR e RGB), representando virtualmente a condição das folhas.

Acionamento de Funções

Botões iniciam a análise e simulam a aplicação de fungicida, demonstrando a economia ao tratar apenas plantas infectadas

Exibição de Resultados

Os valores dos sensores (R, G, B, NIR, ndiv, gndiv) e um indicador RGB de infecção são exibidos no display OLED.

Justificativa

A ferrugem do cafeeiro é uma doença fúngica capaz de reduzir a produtividade em até 50%, impactando severamente a economia dos produtores. Estudos indicam que sensores NIR conseguem detectar alterações na reflectância das folhas com uma precisão de até 92%, antecipando os sintomas em cerca de 12 dias. Assim, o CaféScan não só permite um controle mais eficaz da doença, como também contribui para a redução dos custos com insumos e tratamentos emergenciais.

ORIGINALIDADE

Inspiração e Referências Acadêmicas:

O projeto foi inicialmente inspirado por um trabalho desenvolvido na UESC, que utilizava visão computacional e machine learning para avaliar a saúde de sementes de cacau. Essa iniciativa despertou a ideia de procurar técnicas para a detecção precoce de doenças em outras culturas, como o cafeeiro.

Durante a pesquisa, explorei diversas abordagens para a detecção de doenças até encontrar o artigo disponível no repositório da UFU ([Disponível aqui](#)), que demonstra o uso de sensores NIR e RGB para identificar alterações na reflectância das folhas até 12 dias antes dos sintomas visíveis de ferrugem, comprovando a viabilidade técnica da proposta

Análise das Soluções Existentes:

Os métodos atuais apresentam limitações significativas:

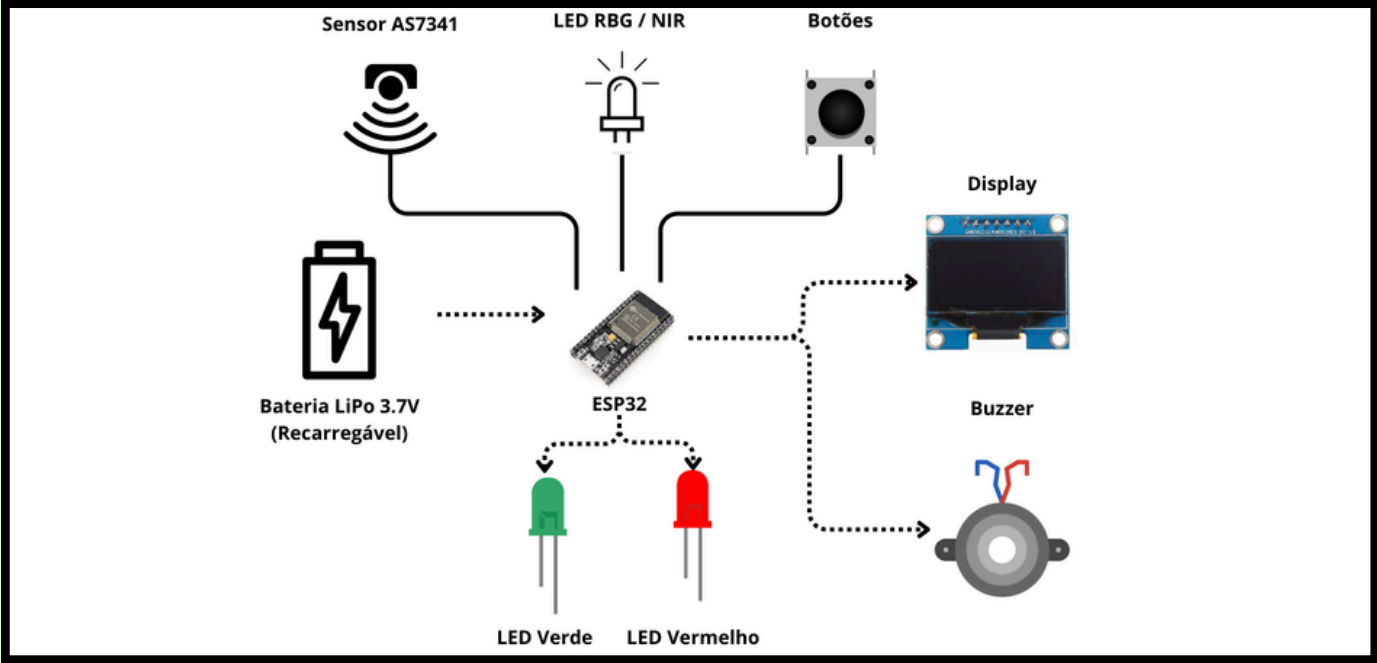
- **Inspeção Visual Manual:** Agricultores ou técnicos percorrem a plantação em busca de manchas amarelas, mas os sintomas só aparecem 15–20 dias após a infecção, além de dependerem da experiência do inspetor e demandarem cerca de 4 horas por hectare.
- **Análise Laboratorial:** Consiste na coleta de folhas para análise microscópica dos esporos. Essa técnica é cara, e demora de 3 a 5 dias para fornecer resultados e não é viável logisticamente para pequenos produtores.
- **Drones com Câmeras Multiespectrais:** Embora permitam mapear a plantação e utilizar NDVI para detectar estresse vegetal, essa solução exige um investimento elevado
- **Aplicativos de Celular (ex: Plantix):** Utilizam algoritmos na nuvem para diagnosticar doenças a partir de fotos, mas a precisão varia conforme a qualidade da imagem e os sintomas só são identificados quando já são visíveis.

Diferencial do Café Scan

Integrando uma solução portátil e de baixo custo o CaféScan se destaca ao:

- Permitir a detecção precoce da ferrugem, otimizando o uso de fungicidas ao tratar apenas as plantas infectadas.
- Superar a ineficiência da inspeção manual e os altos custos de análises laboratoriais e equipamentos multiespectrais.
- Oferecer uma tecnologia viável para pequenos e médios produtores, que necessitam de soluções de ponta a custos reduzidos.

Diagrama em Bloco



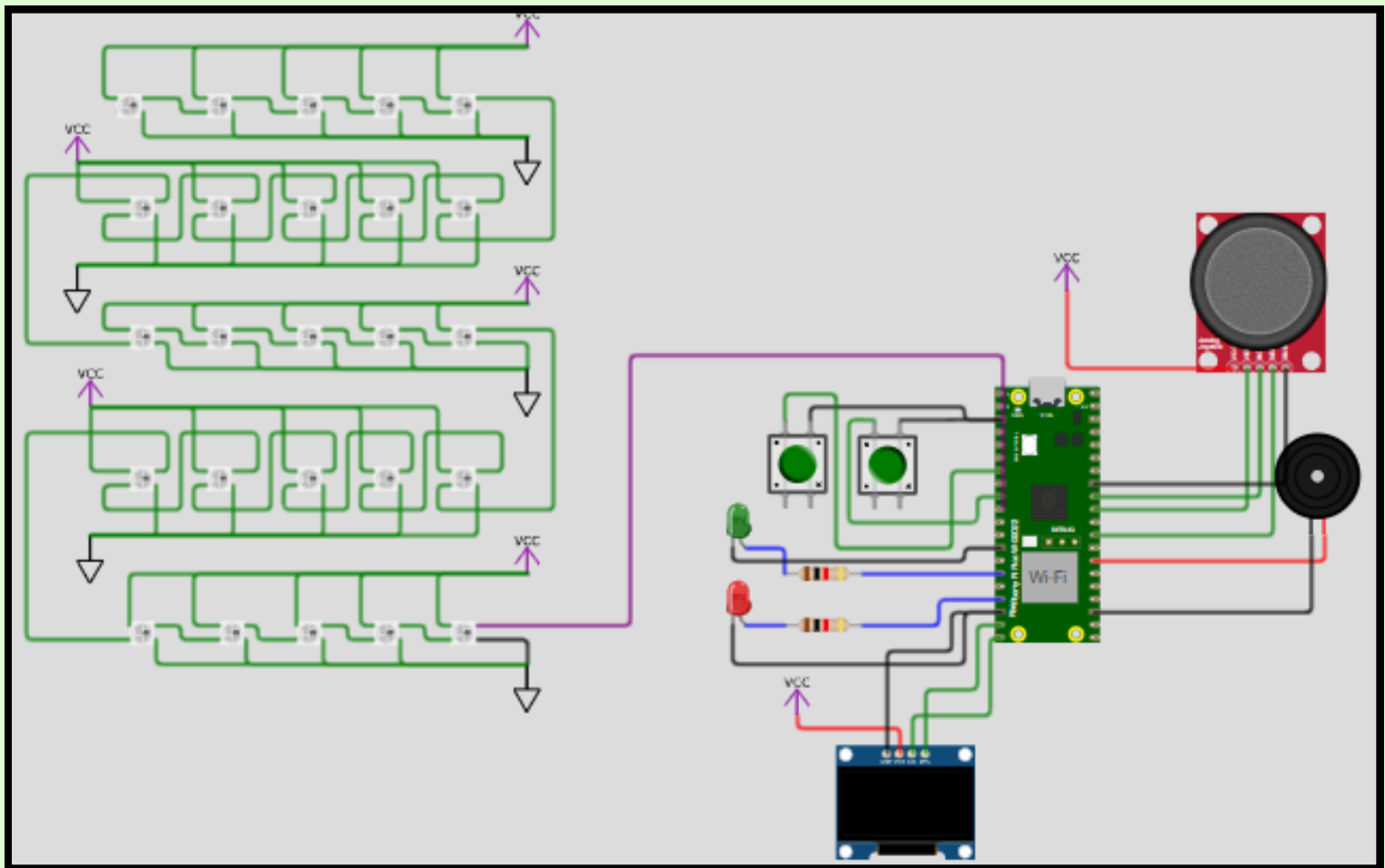
Função

Bloco	Função
ESP32	Processamento dos dados, controle dos periféricos e comunicação.
Sensor AS7341	Mede a reflectância da folha em diferentes faixas espectrais (RGB + NIR).
LEDs RBG/NIR	Iluminação ativa para calibrar medidas conforme a luz ambiente.
Display OLED	Exibe os valores capturados e o status de infecção.
LEDs Vermelho e Verde	Feedback visual: vermelho (infectado), verde (saudável).
Buzzer	Alerta sonoro baseado na análise dos dados.
Botões	Controle das funções de análise e navegação no menu.
Bateria LiPo 3,7V	Fonte de energia para o dispositivo.

Configuração de Cada Bloco(Bit Dog Lab)

Bloco	BitDogLab
Microcontrolador	RP2040
Sensor AS7341	Substituído por joystick. Leitura analógica da “reflectância” simulada. <ul style="list-style-type: none">Eixo X (GPIO 27)Eixo Y (GPIO 26)
LEDs RBG/NIR	Não aplicável. O Joystick já controla a reflectância recebida
Display OLED	<ul style="list-style-type: none">Matrix 5x5 de Leds:<ul style="list-style-type: none">GPIO 7Controlada por PIOExibe um gráfico similar da matriz oled ou exibe planta no modo “menu”Display OLED.:<ul style="list-style-type: none">SDA(GPIO 14),SCL(GPIO 15),Endereço (0x3C),Controlado por I2C.Exibe informações capturadas e instruções de manejo do dispositivo.
LEDs Vermelho e Verde	<ul style="list-style-type: none">LED Vermelho (GPIO 13). Acende quando infecção é detectada.LED Verde (GPIO 11). Acende quando a planta está saudável.
Buzzer	<ul style="list-style-type: none">Buzzer (GPIO 21) (PWM)
Botões	<ul style="list-style-type: none">Botão A (GPIO 5) (Entrada digital com pull-up).Botão B (GPIO 6) (Entrada digital com pull-up).Botão Joystick (GPIO 22) (Entrada digital com pull-up).

Diagrama de Circuito(Simulação Bit Dog Lab)



CAMADA DE SOFTWARE:

Camada de Hardware:

Contém os componentes físicos essenciais para o funcionamento do CaféScan:

- **Sensor AS7341 (Espectral):** Responsável por captar a reflectância.
- **LEDs de Iluminação (NIR/RGB):** Fornecem a iluminação adequada para a medição.
- **Botões de Controle:** Permitem a interação manual com o sistema.
- **Display OLED:** Exibe os resultados e o status do scanner.
- **Buzzer:** Emite alertas sonoros, se necessário.

Camada de Drivers:

Responsável pela comunicação direta com o hardware e pela gestão de funções críticas, incluindo:

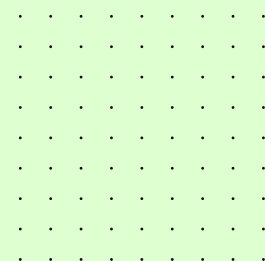
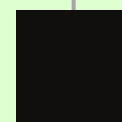
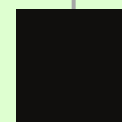
- **I2C:** Gerencia a comunicação com o sensor AS7341 e o display OLED.
- **GPIO:** Controla os botões e os LEDs, facilitando a interação física com o dispositivo.
- **PWM:** Modula o buzzer para a geração dos sinais sonoros.
- **Interrupções e Temporização do Display OLED:** Implementa o gerenciamento das interrupções geradas pelos botões ou por eventos do display, além de controlar a temporização para atualizações precisas do display.

Camada de Processamento:

Realiza o processamento dos dados capturados pelo sensor, aplicando o modelo de IA para classificar os níveis de reflectância e determinar a saúde do objeto analisado. Essa camada é o "cérebro" do sistema, onde a lógica de classificação é aplicada.

Camada de Aplicação:

Fornece a interface para o usuário e o controle geral do sistema. Nesta camada, o usuário visualiza os resultados do scanner e pode interagir com o dispositivo para ajustes simples, caso necessário.



Principais Variáveis (Bit Dog Lab)

Variável	Tipo	Descrição
valores_ajustados	Reflectancia	Estrutura com valores de reflectância (R, G, B, NIR) calibrados (0.0–1.0).
estado_escaneamento	uint8_t	Define o modo atual: MODO_ESCANEAMENTO ou ANALISE .
ndvi	float	Índices de vegetação calculados $NDVI = (NIR - R) / (NIR + R)$
gndvi	float	Índices de vegetação calculados $GNDVI = (NIR - G) / (NIR + G)$
buttonA_flag	bool	Flag indicando se o botão A foi pressionado.
buttonB_flag	bool	Flag indicando se o botão B foi pressionado.

Existem outras variaveis, mas essa são as que considero principais

Estrutura e Formato dos Dados:

1. Estutura Reflectância

```
typedef struct {  
    float R;      // Reflectância no vermelho  
    float G;      // Reflectância no verde  
    float B;      // Reflectância no azul  
    float NIR;    // Reflectância no infravermelho  
} Reflectancia;
```

2. Índices de Vegetação:

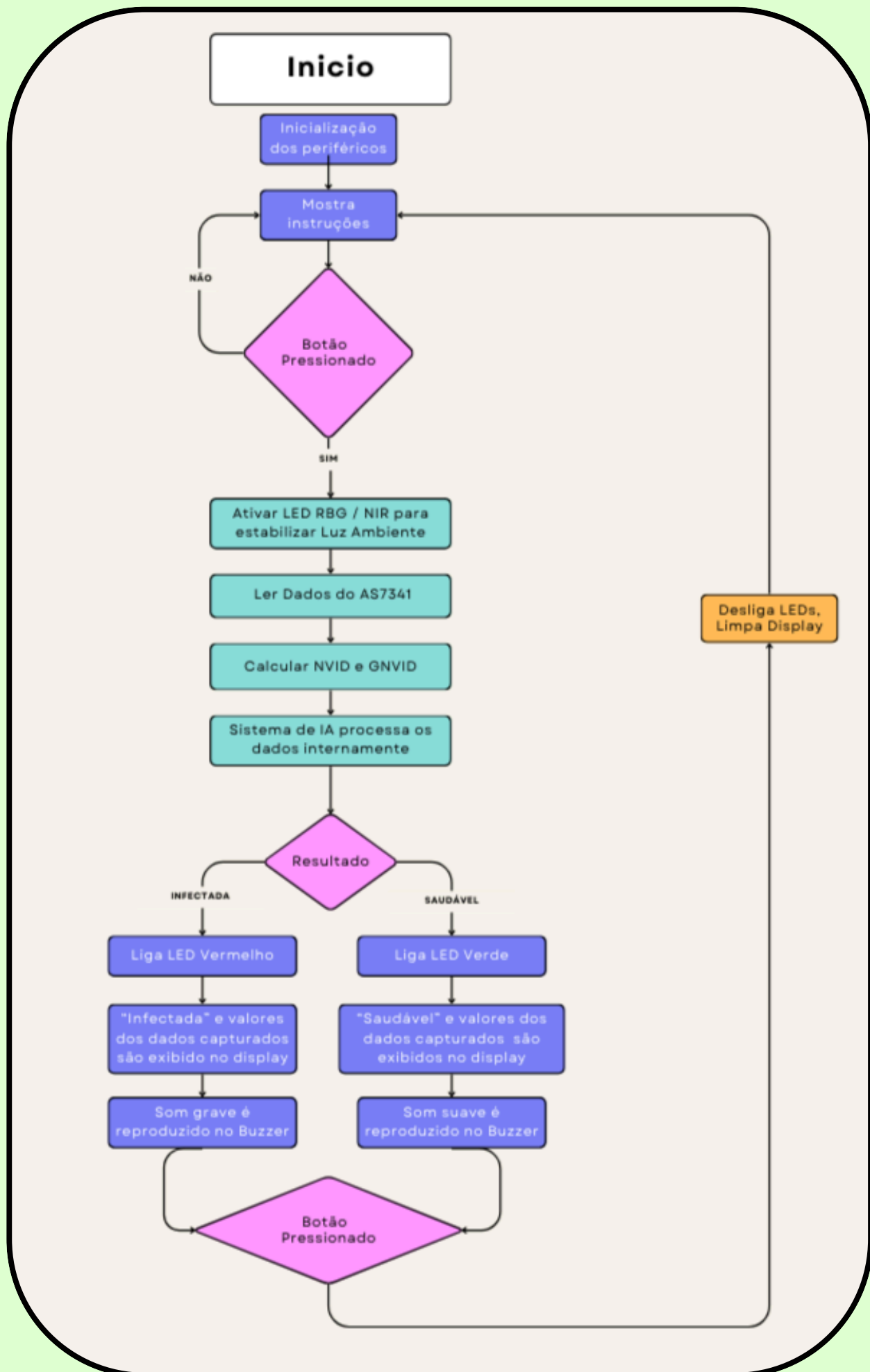
- **NDVI:** Float (-1 a 1), detecta estresse vegetativo (limiar: 0.4).
- **GNDVI:** Float (-1 a 1), mede saúde geral (limiar: 0.35).

Faixa de Valores:

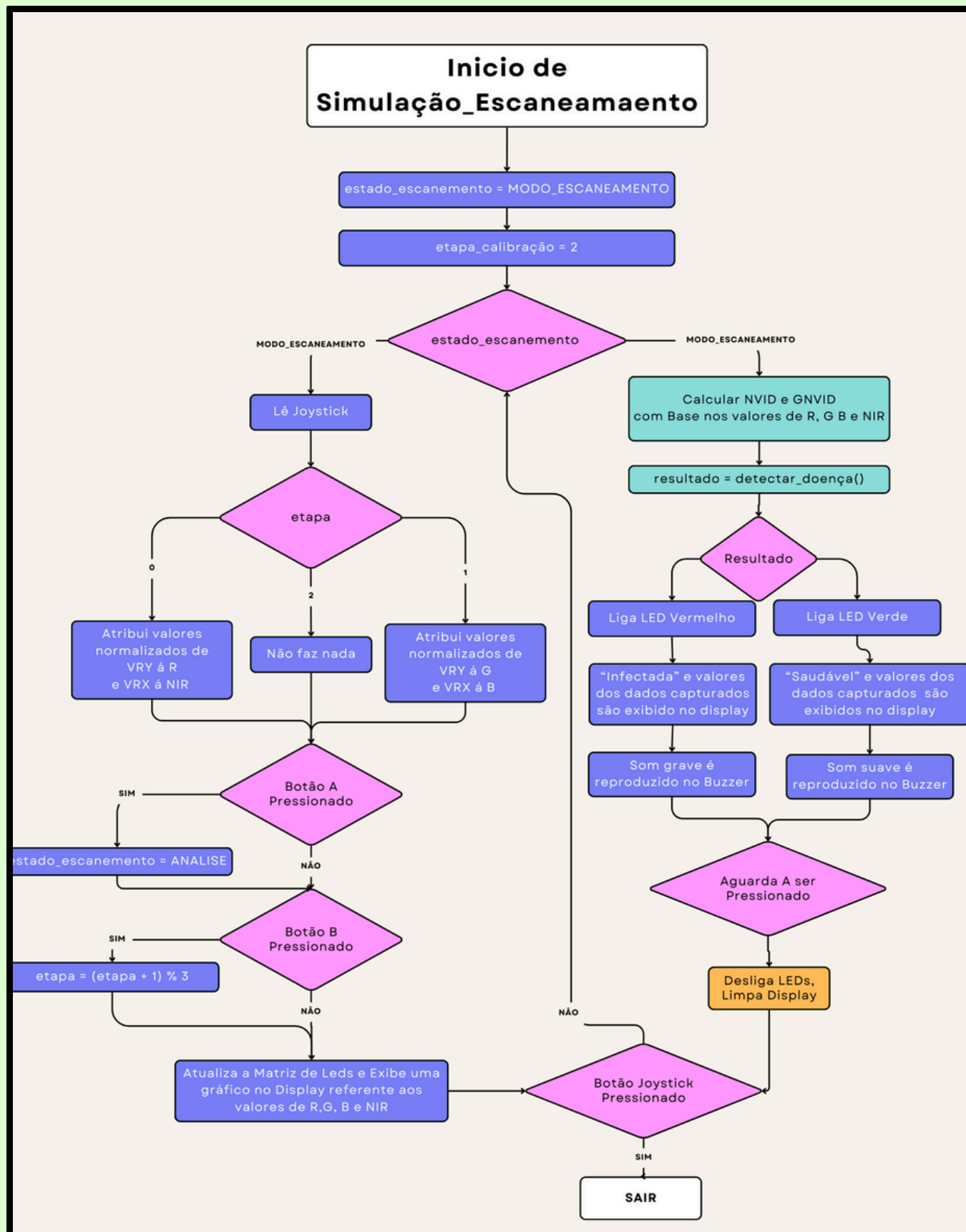
-1 a 1, mas em plantas:

- Folhas saudáveis: $GNDVI > 0.45$ (alto vigor vegetativo).
- Folhas estressadas/doentes: $GNDVI < 0.35$ (estresse metabólico ou infecção).

FLUXOGRAMA DO HARDWARE(SIMPLIFICADO)



FLUXOGRAMA DA SIMULAÇÃO_ESCANEAMENTO NA BIT DOG LAB (SIMPLIFICADO)



Descrição da Função

simular_escaneamento()

Bit Dog Lab

Esta função controla um sistema de simulação de escaneamento e para detecção de doenças em folhas de café, operando em dois modos principais: calibração e análise. Simulando a interação entre hardware (joystick, sensores, display) e lógica de processamento.

Funcionamento Geral

1. Modo de Calibração (MODO_ESCANEAMENTO)

- Ajuste de Parâmetros:
 - **Etapas 0:** Calibra os valores de reflectância do vermelho (R) e infravermelho próximo (NIR) via joystick (eixos VRX/VRX).
 - **Etapas 1:** Calibra verde (G) e azul (B).
 - **Troca de Etapas:** O botão B alterna entre etapas (etapa_calibracao cíclica: 0 → 1 → 2).
- **Início da Análise:** O botão A inicia o processamento, desabilitando interrupções externas durante a análise.
- **Feedback Visual:** Atualiza gráficos em display OLED e matriz LED em tempo real com os valores calibrados.

2. Modo de Análise (ANALISE)

- **Deteção de Doença:** Chama a função detectar_doenca() com os valores calibrados (R, G, B, NIR).
- **Cálculo de Índices:** Computa NDVI e GNDVI para validação quantitativa.
- **Feedback ao Usuário:**
 - **Sonoro:** Buzzer emite sons distintos para "saudável" ou "infectada".
 - **Visual:** LEDs e display mostram resultados e valores espectrais.
- **Retorna** ao modo de calibração após conclusão.

3. Saída do Sistema:

- O botão do joystick (**buttonJoyStick_flag**) interrompe o loop, limpa displays e restaura configurações iniciais.

Função detectar_doença:

```
bool detectar_doenca(float R, float G, float B, float NIR) {  
  
    // 1. Cálculo dos índices  
    float ndvi = (NIR - R) / (NIR + R + 0.001f);  
    float gndvi = (NIR - G) / (NIR + G + 0.001f);  
  
    // 2. Limiares científicos  
    const float NDVI_SAUDAVEL = 0.4f;  
    const float GNDVI_SAUDAVEL = 0.35f;  
    const float ERRO = 0.2f;  
  
    // 3. Critérios de detecção  
    bool criterio_ndvi = ndvi < NDVI_SAUDAVEL;  
    bool criterio_gndvi = gndvi < GNDVI_SAUDAVEL;  
    bool criterio_visivel = (R > 0.65f) && (G < 0.55f);  
  
    // 4. Lógica de decisão  
    bool resultado = (criterio_ndvi && criterio_gndvi) || criterio_visivel;  
  
    /* 5. Simulação de erro  
    if((float)rand()/RAND_MAX < ERRO) {  
        return !resultado;  
    }  
    */  
    return resultado;  
}
```

Esta função representa um protótipo simplificado para simulação (Bit Dog Lab), baseado em critérios fixos de NDVI, GNDVI e reflectância nas bandas do visível (R, G). Ela opera da seguinte forma:

Cálculo de Índices::

- **NDVI** e **GNDVI** são calculados para estimar saúde vegetal, com valores baixos (<0.4 e <0.35) indicando possível doença.
- O critério **R > 0.65** e **G < 0.55** captura alterações visuais associadas à degradação de clorofila

Lógica de Decisão:

- A doença é detectada se **ambos os índices estiverem abaixo dos limiares** *OU* se as bandas R/G atenderem ao critério visível.

Esta função ilustra um modelo simplificado para simulação de detecção de ferrugem em ambiente controlado, utilizando dados espectrais. Embora útil para validação inicial, ela não incorpora o algoritmo de aprendizado de máquina (SVM) proposto no estudo referenciado, sendo necessária a integração de IA para maior robustez em aplicações reais

Comentário sobre a Main (Bit Dog Lab)

Esta função é uma simulação para detecção de doenças em cafeeiros, integrando hardware (joystick, display OLED, buzzer) e lógica de controle de estados. Seu fluxo principal é organizado em quatro estados: menu de plantas, seleção de folhas, análise detalhada e modo de escaneamento.

Principais Características:

1. Inicialização e Configuração:

- Configura periféricos (GPIO, ADC, display) e inicializa plantas com estados pré-definidos (saudáveis, infectadas visíveis/assintomáticas).
- Utiliza estruturas de dados (Planta, EstadoFolha) para armazenar informações espectrais e status de infecção.

2. Máquina de Estados:

- **Menu Principal:** Navegação entre plantas, tratamento químico (custo em recursos) e transição para análise foliar.
- **Seleção de Folha:** Interface interativa para escolher folhas específicas, com feedback visual em uma matriz LED.
- **Análise:** Simula diagnóstico usando a função `detectar_doenca_folha()`, com feedback sonoro (buzzer) e visual (LEDs).
- **Modo Escaneamento:** Chama `simular_escaneamento()` para calibração manual e simulação de aquisição espectro-visual (R, G, B, NIR).

3. Integração Hardware-Software:

- Gerencia entradas do joystick e botões com interrupções configuráveis.
- Atualiza displays em tempo real (`exibir_grafico_display`, `exibir_grafico_matriz`) e fornece feedback multimodal (sonoro, visual).

4. Simulação:

- Simula a Redução de Custos: Direciona fungicidas apenas para plantas infectadas (detectadas pelo escaneamento), evitando desperdícios e tratamentos desnecessários.
- Simula a Identificação de infecções ocultas, permitindo tratamento precoce.

Em resumo, a função `simular_escaneamento` é a principal, atuando como o núcleo do sistema: nela, ajustamos manualmente os valores de reflectância (R, G, B, NIR) via joystick, simulando o funcionamento do dispositivo "CaféScan". Enquanto isso, a main demonstra como essa detecção avançada reduz custos operacionais, direcionando fungicidas apenas para plantas infectadas (visíveis ou assintomáticas) e evitando desperdícios.

Descrição do Processo de Inicialização e Configuração de Registros (Bit Dog Lab)

1. Inicialização do Software

A inicialização do sistema é dividida em duas etapas principais:

a. Configuração de Hardware

- **hardware_setup():**
 - Configura os periféricos essenciais (GPIO, ADC, PWM).
 - Inicializa o joystick, botões e buzzer para interação do usuário.
 - Prepara o ADC para leitura analógica dos eixos do joystick.
- **display_init(&display):**
 - Inicializa o display OLED, definindo parâmetros como tamanho de tela, protocolo de comunicação (I2C/SPI) e buffers gráficos.
 - Carrega fontes e configurações iniciais para exibição de textos e gráficos.

b. Inicialização de Dados

- **Criação de Plantas:**
 - Gera um array de Plantas (plantas[NUM_PLANTAS]), onde cada planta é inicializada com um perfil pré-definido:
 - **Saudáveis:** Sem infecção.
 - **Infectadas Visíveis:** Sintomas aparentes.
 - **Assintomáticas:** Infecção oculta, detectável apenas por análise espectral.
 - Utiliza a função gerar_planta() para definir parâmetros como reflectância (R, G, B, NIR) e índices de vegetação (NDVI, GNDVI).
- **Variáveis de Controle:**
 - Inicializa variáveis de estado (estado_atual = ESTADO_ESCANEAMENTO), índices de plantas/folhas (indice_planta, indice_folha) e custos (custo_total = 0).

Descrição do Processo de Inicialização e Configuração de Registros (Bit Dog Lab)

2. Configurações dos Registros

- **GPIO/ADC:**

- Configurados via SDK da Raspberry Pi Pico (ex: `gpio_init()`, `adc_init()`).
- Mapeamento de pinos para joystick (VRX, VRY), botões e buzzer.

- **Display OLED:**

- Registros de controle (endereço I2C, comandos SSD1306) são manipulados pela biblioteca `ssd1306.h` para atualizar pixels e textos.

b. Registros de Estado e Controle

- **Estado:**

- Controla o fluxo do sistema (**ESTADO_MENU**, **ESTADO_ESCANEAMENTO**, etc.).
- Alterado via interrupções de botões ou conclusão de tarefas.

- **Reflectancia e EstadoFolha:**

- Armazenam dados espectrais (R, G, B, NIR) e resultados de análise (NDVI, GNDVI).
- Atualizados durante o escaneamento (**simular_escaneamento()**) e análise (**detectar_doenca_folha()**).

- **Flags e Temporizadores:**

- **buttonA_flag/buttonB_flag**: Sinalizam eventos de botões.
- **atualizar_display**: Força a atualização da interface gráfica.
- **ultima_atualizacao_menu/folha**: Controla temporização de da rotação das mensagens

A inicialização prepara o hardware e dados, enquanto os registros gerenciam o estado dinâmico do sistema, garantindo interatividade e precisão na simulação.

Protocolo de Comunicação no Projeto: (Bit Dog Lab)

O sistema utiliza protocolos de comunicação internos para integrar os componentes de hardware ao microcontrolador (ex: Raspberry Pi Pico). Não há um protocolo de comunicação externo (como Wi-Fi, Bluetooth ou UART), mas os seguintes protocolos são empregados para operar os periféricos:

1. Protocolo I²C (Inter-Integrated Circuit)

- **Finalidade:** Comunicação com o display OLED (SSD1306).
- **Funcionamento:**
 - Dois fios: SDA (dados) e SCL (clock).
 - Endereço I²C do display: 0x3C (comum para displays OLED monocromáticos).
 - Biblioteca `ssd1306.h` abstrai o envio de comandos e dados (ex: inicialização, atualização de pixels).

2. Protocolo WS2812B (NeoPixel) (Não aplicável no hardware teórico)

- **Finalidade:** Controle da matriz LED NeoPixel (LEDs endereçáveis).
- **Funcionamento:**
 - Comunicação via timing de pulsos em um único fio (DATA).
 - Cada LED é endereçado individualmente, permitindo atualizações dinâmicas de cores.
 - Biblioteca `neopixel.h` gerencia o protocolo, convertendo valores RGB em sequências de pulsos.

A inclusão de um protocolo de comunicação externo (como Wi-Fi, Bluetooth ou UART) pode vir com a evolução do projeto. Permitindo atualizar o firmware ou realizar monitoramento remoto. Alguns desses protocolos já vem com o ESP32.

Execução do Projeto

Metodologia

- **Pesquisas Realizadas:**

- Revisão de literatura sobre detecção de doenças em cafeeiros utilizando imagens multiespectrais.
- Consulta a artigos científicos, como o estudo "**DETECÇÃO PRECOCE DA FERRUGEM DO CAFEIEIRO CAUSADA POR HEMILEIA VASTATRIX A PARTIR DE IMAGENS MULTIESPECTRAIS**", que fundamentou os critérios de detecção utilizados no projeto.

- **Escolha do Hardware:**

- **Sensor Espectral:** Seleção do AS7341 por ser barato e por sua capacidade de captar múltiplas bandas (R, G, B, NIR) com precisão.
- **Display e Interface:** Uso de um display OLED para exibição dos resultados e de uma LED matrix para feedback visual imediato.
- **Microcontrolador:** O ESP32 foi escolhido por sua capacidade de processamento para modelos de IA, conectividade Wi-Fi/Bluetooth (para futuras expansões) e custo acessível, integrando sensores e periféricos de forma eficiente.

- **Definição das Funcionalidades do Software:**

- **Captura e Calibração:** Leitura dos valores de reflectância dos sensores, com ajuste via joystick e botões.
- **Processamento e Análise:** Cálculo dos índices NDVI e GNDVI, e aplicação de critérios para detectar a condição da folha (saudável ou infectada).
- **Interface e Feedback:** Exibição dos gráficos e resultados tanto no display OLED quanto na LED matrix, com feedback sonoro via buzzer.

- **Instalação e Uso:**

1. Pré-requisitos

- **Clonar o repositório:**

```
git clone https://github.com/JotaPablo/projetofinal.git
cd projetofinal
```

- Instalar o Visual Studio Code com as seguintes extensões:
 - Raspberry Pi Pico SDK
 - Compilador ARM GCC

2. Compilação

- Compile o projeto no terminal:

```
mkdir build
cd build
cmake ..
make
```

- Ou utilize a extensão da Raspberry Pi Pico no VS Code.

3. Execução

- Na placa física:
 - Conecte a placa ao computador em modo BOOTSEL.
 - Copie o arquivo .uf2 gerado na pasta build para o dispositivo identificado como RPI-RP2, ou envie através da extensão da Raspberry Pi Pico no VS Code.

Execução do Projeto

Testes de Validação

- **Testes Unitários e de Integração:**

- Verificação isolada dos módulos de leitura (ADC, I2C) e dos drivers de hardware.
- Integração dos módulos para garantir que a comunicação entre sensor, display e botões ocorra sem falhas.

- **Testes Funcionais:**

- **Calibração:** Testes para confirmar que os ajustes via joystick e botões modificam corretamente os valores de reflectância.
- **Processamento:** Comparação dos índices NDVI e GNDVI calculados com valores de referência, conforme os critérios descritos no artigo científico.
- **Interface:** Verificação se os gráficos e os textos exibidos no display OLED refletem com precisão os dados captados e processados.

Foi implementada a função teste_deteccao() para testar se retorna os resultados corretos para determinados valores

Discussão dos Resultados

Os testes demonstraram que o firmware consegue detectar, de forma consistente, alterações na reflectância que indicam a presença da ferrugem, validando os critérios propostos. Entretanto, é essencial realizar testes específicos com o sensor utilizado e promover um treinamento adequado do algoritmo de IA, demandando ajustes finos nos parâmetros de detecção.

- **Confiabilidade do Sistema:**

- Os testes demonstraram que o firmware consegue detectar, de forma consistente, alterações na reflectância que indicam a presença da ferrugem, validando os critérios propostos.

- **Pontos Fortes (Simulação):**

- **Calibração Dinâmica:** Permite ajustes finos, possibilitando o teste com diferentes tipo de reflectância
- **Interface Intuitiva:** A combinação de display OLED, LED matrix e feedback sonoro facilita a interpretação dos resultados pelo usuário.

- **Desafios e Melhorias Futuras (Na Aplicação do Sistema Teórico) :**

- **Variação de Iluminação:** Ambientes com iluminação variável podem afetar as medições; técnicas de filtragem e ajuste de thresholds podem ser aprimoradas.
- **Aprimoramento do Algoritmo de IA:** O modelo atual pode ser expandido ou substituído por métodos mais robustos para reduzir falsos positivos/negativos.
- **Integração com Sistemas de Monitoramento Remoto:** Futuramente, a transmissão dos dados para uma estação central pode aumentar o controle e a análise em larga escala.

Conclusão

O projeto demonstrou alta viabilidade para a detecção precoce da ferrugem em cafeeiros, apresentando um sistema confiável e com potencial de escalabilidade. Com os aprimoramentos sugeridos – especialmente a validação específica do sensor e o treinamento do algoritmo de IA – este dispositivo pode se tornar uma ferramenta viável e eficaz para auxiliar no combate à ferrugem no cafeeiro.

Referências

- **Artigo Científico (FAZUOLI et al., 2007)**

FAZUOLI, L. C.; TOMA-BRAGHINI, M.; SILVAROLLA, M. B.; OLIVEIRA, A. C. B. A ferrugem alaranjada do cafeeiro e a obtenção de cultivares resistentes. O Agrônomo, Campinas, v. 59, n. 1, p. 48-53, 2007.

Disponível em:

<https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1138046/1/Embrapa-Cafe-Documentos-15.pdf> Acesso em: 25/02/2025.

- **Capítulo de Livro (ZAMBOLIM, 2015)**

ZAMBOLIM, L. Manejo de doenças. In: FONSECA, A. F. A.; SAKIYMA, N. S.; BOREN, A. (Ed.). Café conilon: do plantio à colheita. 1. ed. Viçosa: UFV, 2015. p. 114-137.

Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1170688/manejo-integrado-de-pragas-e-doencas-dos-cafes-conilon-e-robusta>

Acesso em: 25/02/2025.

- **Documento Técnico (OLIVEIRA et al., 2021)**

OLIVEIRA, A. C. B. et al. Cultivares de café resistentes à ferrugem: alternativa viável para a cafeicultura das Matas de Minas. Brasília, DF: Embrapa Café, 2021. 46 p. (Embrapa Café. Documentos, 15).

Disponível em: <https://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1138046>

Acesso em: 25/02/2025.

- **Publicação Técnica (CAIXETA et al., 2024)**

CAIXETA, E. T. et al. Manejo integrado de pragas e doenças dos cafés conilon e robusta. Brasília, DF: Embrapa Café, 2024.

Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1170688/manejo-integrado-de-pragas-e-doencas-dos-cafes-conilon-e-robusta>

Acesso em: 25/02/2025.

- **Dissertação (SOARES, 2022)**

SOARES, Anais da Silva. Detecção precoce da ferrugem do cafeeiro causada por Hemileia vastatrix a partir de imagens multiespectrais. 2022. 37 f. Dissertação (Mestrado em Agricultura e Informações Geoespaciais) – Universidade Federal de Uberlândia, Monte Carmelo, MG, 2022.

Disponível em: <http://doi.org/10.14393/ufu.di.2022.429>

Acesso em: 25/02/2025.

**Video de
Demonstração**



**Link do
Repositório**

