

# 1. Escopo do Projeto

Nome: Helder Lourenço de Abreu Marques

Link do projeto: [https://github.com/F1reFinger/Batalha\\_do\\_Mal](https://github.com/F1reFinger/Batalha_do_Mal)

## Apresentação do Projeto

O projeto consiste em um jogo de batalha naval baseado em uma matriz de LEDs SMD 5050 de 5x5 para exibição do campo de batalha e um monitor OLED SSD1306 para exibição da pontuação. O controle será realizado por um joystick analógico para movimentação e um botão para seleção.

## Título do Projeto

Batalha Naval em Matriz de LEDs com Pontuação OLED

## Objetivos do Projeto

- Desenvolver um jogo de batalha naval interativo utilizando a placa BitDogLab.
- Criar uma interface intuitiva com LEDs e um display OLED.
- Implementar controles responsivos com joystick e botões.

## Principais Requisitos

- Matriz de LEDs 5x5 para exibição do jogo.
- Monitor OLED SSD1306 para exibição da pontuação.
- Controle via joystick analógico e botão para seleção.
- Implementação utilizando a BitDogLab e suas conexões I2C.

## Descrição do Funcionamento

na fase de preparação o jogador recebe um prompt para definir onde os barcos estarão posicionados e com isso são colocados os barcos de tamanho 1x1

O jogador movimenta um cursor pela matriz de LEDs utilizando o joystick analógico. Ao pressionar o botão de seleção, um tiro é disparado. Se o tiro atingir um alvo, o LED correspondente pisca e a pontuação é atualizada no display OLED.

## Justificativa

Este projeto explora conceitos fundamentais de sistemas embarcados, como controle de hardware via software, comunicação I2C e interatividade com o usuário. Além disso, promove uma abordagem didática para o aprendizado de eletrônica e programação.

## Originalidade

Embora existam jogos de batalha naval eletrônicos, esta implementação específica utilizando BitDogLab, matriz de LEDs SMD 5050 e display OLED apresenta um design inovador e simplificado para sistemas embarcados.

outro ponto que este projeto foi desenvolvido para a disciplina de APC Algoritmos de Programação de Computadores ou seja uma matéria introdutória que auxilia em termos a aprender programação bem como o projeto Embarcatech.

o código de batalha naval usa somente a biblioteca <stdio.h> o que facilita e muito para programadores iniciantes aprenderem a mexer.

## 2. Hardware

### Diagrama em Blocos

[Inserir diagrama representando as conexões entre os componentes]

### Função de Cada Bloco

- **Matriz de LEDs (5x5):** Representa o campo de batalha.
- **OLED SSD1306:** Exibe a pontuação do jogador.
- **Joystick Analógico:** Move o cursor na matriz.
- **Botão B:** Confirma a seleção/disparo.
- **BitDogLab:** Processamento e controle do jogo.

### Configuração de Cada Bloco

- **Matriz de LEDs:** Controlada pelo microcontrolador, utilizando o pino de dados para manipulação das cores.
- **OLED:** Comunicação via I2C (pinos SDA e SCL).
- **Joystick:** Lê valores analógicos para determinar a direção do cursor.
- **Botão A:** Configurado como entrada digital para registrar cliques.

### Especificações

O projeto atende aos requisitos de baixo custo, tamanho compacto e baixo consumo de energia, sendo ideal para aplicações embarcadas de modo que visa ensinar a programação de embarcados.

### Lista de Materiais

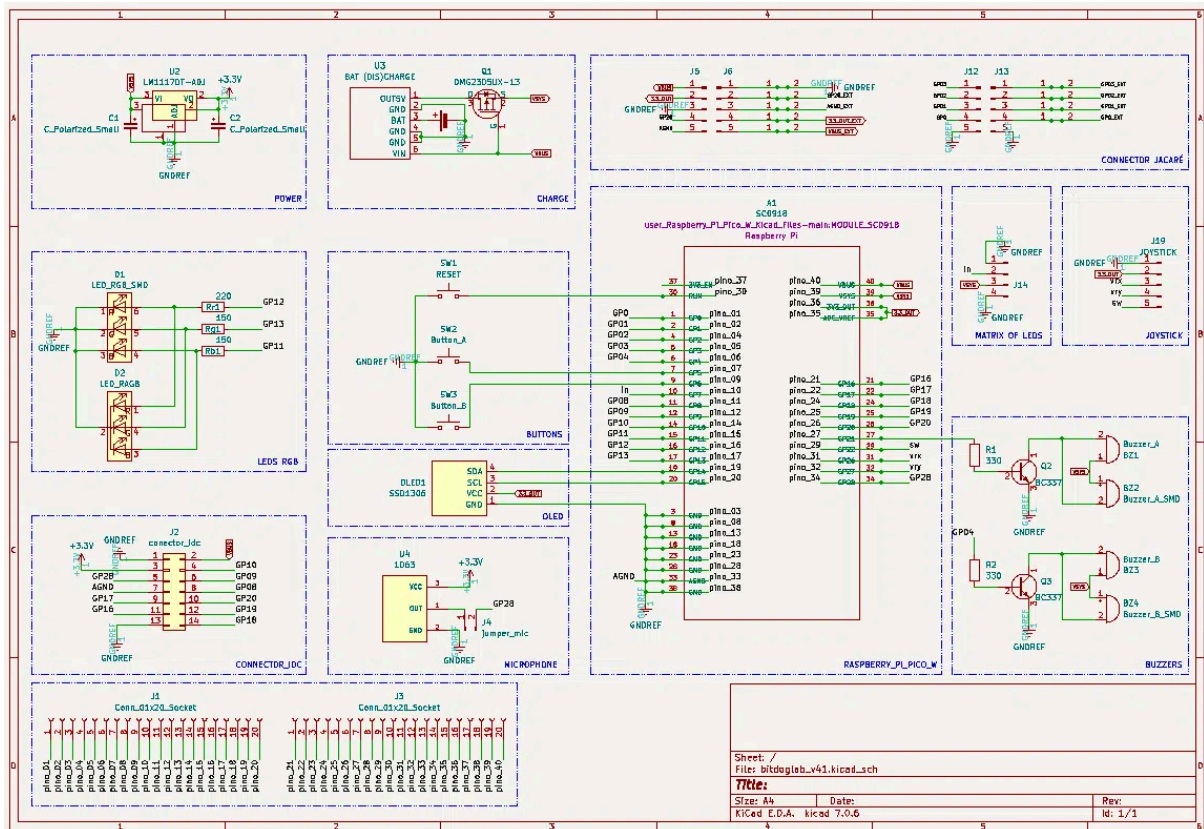
Componente	Quantidade
------------	------------

Matriz de LEDs SMD 5050 5x5	1
OLED SSD1306	1
Joystick analógico	1
Botão A	1
BitDogLab	1

## Descrição da Pinagem Usada

P	Função
7	Controle da matriz de LEDs
1	I2C SDA (OLED)
1	I2C SCL (OLED)
6	Botão B

# Circuito Completo do Hardware



## 3. Software

### Blocos Funcionais

Apresentação	<ul style="list-style-type: none"> <li>-imprime mensagem</li> <li>-imprime tabuleiro</li> <li>-imprime na tela</li> </ul>
Aplicação	<ul style="list-style-type: none"> <li>-funcMSN</li> <li>-posicionaBarco</li> <li>-atirar</li> </ul>
Domínio	<ul style="list-style-type: none"> <li>-validaEntradaLinhaColuna</li> <li>-validaPosição</li> <li>-calculaPontuação</li> </ul>
infraestrutura	<ul style="list-style-type: none"> <li>-podeAtirar</li> </ul>

Diagrama de autoria própria

## Descrição das Funcionalidades

- Inicialização do hardware.
- Controle da matriz de LEDs.
- Leitura do joystick e botões.
- Atualização do display OLED com a pontuação.
- Implementação da lógica do jogo (movimentação, disparo e pontuação).

## Definição das Variáveis

- `matrix[5][5]`: Armazena o estado da matriz de LEDs.
- `cursor_x`, `cursor_y`: Posição do cursor.
- `score`: Pontuação do jogador.

## Fluxograma

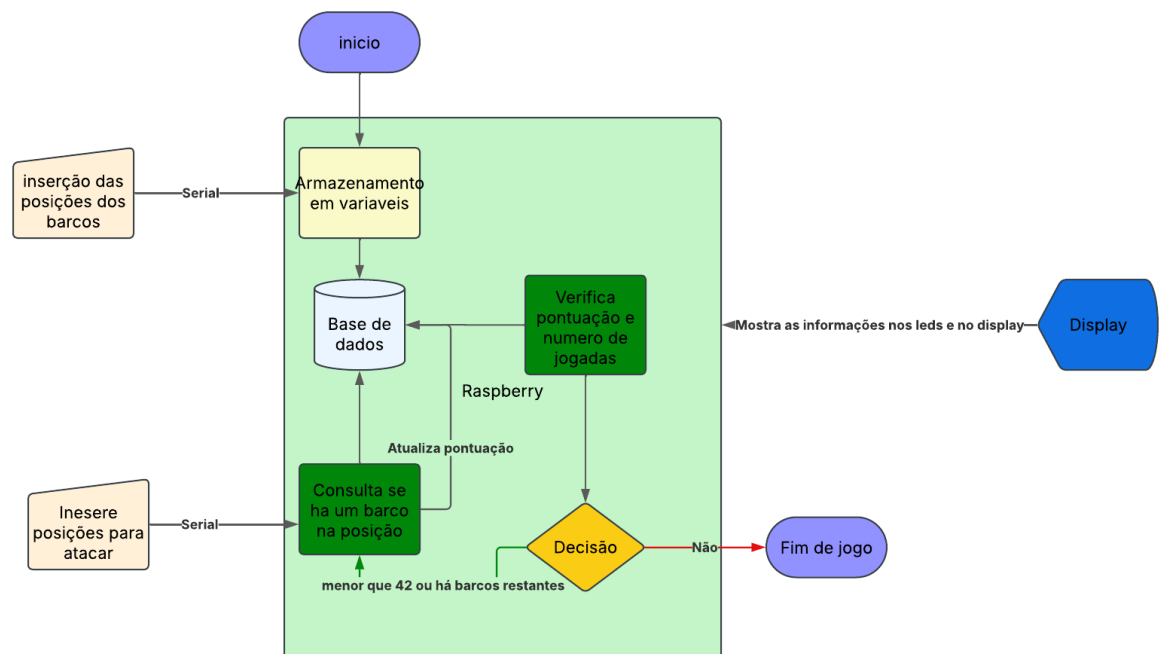


Diagrama de autoria própria

## Inicialização

- Configuração dos pinos.
- Inicialização do display OLED via I2C.
- Configuração do joystick e botões.

## Configuração dos Registros

Verificar configurações de compilação através do cmake.

## Estrutura e Formato dos Dados

Os dados são armazenados em matrizes e variáveis simples, facilitando o acesso e manipulação.

uso somente das bibliotecas contidas nos exemplos do git da bitdog lab

## **Organização da Memória**

Memória dividida entre variáveis de controle, estado da matriz e dados do display OLED.

## **Protocolo de Comunicação**

O display OLED se comunica via protocolo I2C.

## **Formato do Pacote de Dados**

Cada comando enviado ao OLED segue o protocolo I2C, incluindo endereço do dispositivo e dados da tela.

# **4. Execução do Projeto**

## **Metodologia**

1. Pesquisa sobre jogos embarcados.
2. Seleção dos componentes compatíveis com a BitDogLab.
3. Implementação do código e configuração de hardware.
4. Testes e ajustes na lógica do jogo.

## **Testes de Validação**

- Verificação da resposta do joystick.
- Teste do acionamento dos LEDs.
- Atualização correta do display OLED.
- Funcionamento da lógica de pontuação.
- algoritmo feito em C rodando no gcc:

```

*****
*
* Jogador 1 - Posicione o barco de tamanho 1 (1/6)
*
*****
  A B C D E F G H I J
  # # # # # # # # # # #
01 # . . . . . #
02 # . . . . . #
03 # . . . . . #
04 # . . . . . #
05 # . . . . . #
06 # . . . . . #
07 # . . . . . #
08 # . . . . . #
09 # . . . . . #
10 # . . . . . #
  # # # # # # # # # # #

```

- 
- algoritmo rodando direto da bitdoglab:

```

---- Opened the serial port /dev/ttyACM0 ----
---- Closed the serial port /dev/ttyACM0 ----
---- Opened the serial port /dev/ttyACM0 ----
função de atualiza tela
função de posiciona barco
função de atualiza tela
  A B C D E
  # # # # # #
01 # . . . . . #
02 # . . . . . #
03 # . . . . . #
04 # . . . . . #
05 # . . . . . #
  # # # # # #

```

- 
- código funcionando diretamente da placa não com o hardware mas rodando pela placa

## Discussão dos Resultados

O projeto demonstrou ser funcional e responsivo, oferecendo uma experiência interativa. A integração entre os componentes foi validada com sucesso.

## Vídeo de Demonstração

<https://youtube.com/shorts/z77axaXgo5c?feature=share>

video com melhor execução:

[https://youtube.com/shorts/eb-9ZTmEx\\_k?feature=share](https://youtube.com/shorts/eb-9ZTmEx_k?feature=share)



## **Conclusão**

O objetivo deste projeto é ser um projeto facilmente replicável tanto a nível de desenvolvimento de programação para sistemas operacionais mais robustos como também para sistemas embarcados.

## **Melhorias**

Apesar de inúmeros esforços muitas partes de documentação e de execução ainda tem uns pontos para serem corrigidos como por exemplo o ninja precisa de dados deixados pelo Cmake ao compilar o código pra placa e diversas vezes em sistemas operacionais unix like(manjaro linux ) ocorreram diversos problemas que causaram atrasos e geraram bugs algumas vezes inexplicáveis para o projeto, e isso fez com que o desenvolvimento sofresse algumas mudanças tanto de bibliotecas que conseguiriam ser compiladas quanto bibliotecas que poderiam agilizar o uso da placa o código de batalha naval pode ser encontrado na íntegra em :

<https://github.com/F1reFinger/batalha-naval>

O jogo foi feito pensando em sistemas operacionais e feito para ser multiplayer somente sua adaptação já funcionaria como um projeto a parte para ser jogado com 1 player e adaptar para a bitdog foi mais complexo ainda apesar de divertido.

porém os pontos principais são correções de erros e bugs pois uns 60/70% do projeto funciona.

Fontes:

Repositório dos quais foram aproveitados exemplos:

<https://github.com/BitDogLab/BitDogLab-C>

Algumas funções de ponteiros:

[https://www.tutorialspoint.com/c\\_standard\\_library/stdio\\_h.htm](https://www.tutorialspoint.com/c_standard_library/stdio_h.htm)