



PROJETO FINAL: TIMER CONFIGURÁVEL COM PAUSA NA BITDOGLAB

Feira de Santana, Bahia

Fevereiro/2025



JOÃO VICTOR PAIM SANTOS ALMEIDA

PROJETO FINAL: TIMER CONFIGURÁVEL COM PAUSA NA BITDOGLAB

Atividade referente ao projeto final da
Formação Básica em Software Embarcado,
do programa Embarcatech.

Feira de Santana, Bahia

Fevereiro/2025

Sumário

a. Escopo do Projeto	4
Apresentação do Projeto	4
Título do Projeto	4
Objetivos do Projeto	4
Descrição do Funcionamento	4
Justificativa	4
Originalidade	4
b. Especificação do hardware	6
Diagrama em Bloco	6
Função de Cada Bloco	6
Configuração de Cada Bloco	6
Comandos e Registros Utilizados	7
Descrição da Pinagem Usada	7
Circuito Completo do Hardware	8
c. Especificação do firmware	9
Blocos Funcionais	9
Descrição das Funcionalidades	9
Definição das Variáveis	9
Fluxograma	10
Inicialização	10
Configurações dos Registros	10
Estrutura e Formato dos Dados	10
Protocolo de Comunicação	10
Formato do Pacote de Dados	11
d. Execução do projeto	12
Metodologia	12
Testes de Validação	12
Discussão dos Resultados	12
e. REFERÊNCIAS	13
f. Links relacionados	14
• Vídeo de demonstração	14
• Repositório no GitHub	14

a. Escopo do Projeto

Apresentação do Projeto

Este projeto apresenta o desenvolvimento de um sistema embarcado baseado na plataforma BitDogLab, utilizando o microcontrolador Raspberry Pi Pico. O sistema consiste em um timer configurável que permite ao usuário definir um tempo em minutos, iniciar a contagem regressiva, pausá-la e exibir o tempo restante, com notificações visuais e sonoras ao final da contagem.

Título do Projeto

"Timer Configurável com Pausa"

Objetivos do Projeto

- Desenvolver um sistema embarcado original para temporização configurável;
- Implementar funcionalidades de pausa e retomada da contagem;
- Utilizar interfaces de entrada (botões e joystick) e saída (display SSD1306, matriz de LEDs e buzzers) da BitDogLab;
- Garantir baixo custo e tamanho reduzido, alinhados aos conceitos de Sistemas Embarcados (SE).

Descrição do Funcionamento

O sistema permite configurar um tempo em minutos usando o eixo Y do joystick. O botão B alterna entre os modos de configuração e execução. O botão A inicia ou reinicia a contagem quando fora do modo de configuração. O botão do joystick ativa/desativa o sistema.

Durante a contagem, o tempo restante é exibido no display SSD1306 em segundos, e, nos últimos 10 segundos, a matriz de LEDs 5x5 (WS2812) mostra a contagem regressiva em números. Ao atingir zero, os dois buzzers emitem cinco bips de 200 ms cada, e "Timer encerrado" é exibido apenas no modo de execução. A contagem pode ser pausada ao entrar no modo de configuração e retomada ao sair dele.

Justificativa

Temporizadores são amplamente utilizados em diversas aplicações, como culinária, exercícios físicos e experimentos científicos. A adição de pausa e interfaces visuais/sonoras torna o sistema versátil e útil, atendendo a uma necessidade prática com uma solução de baixo custo baseada em SE. Este projeto também demonstra a aplicação dos conceitos aprendidos no curso EmbarcaTech, como programação em C e uso de microcontroladores.

Originalidade

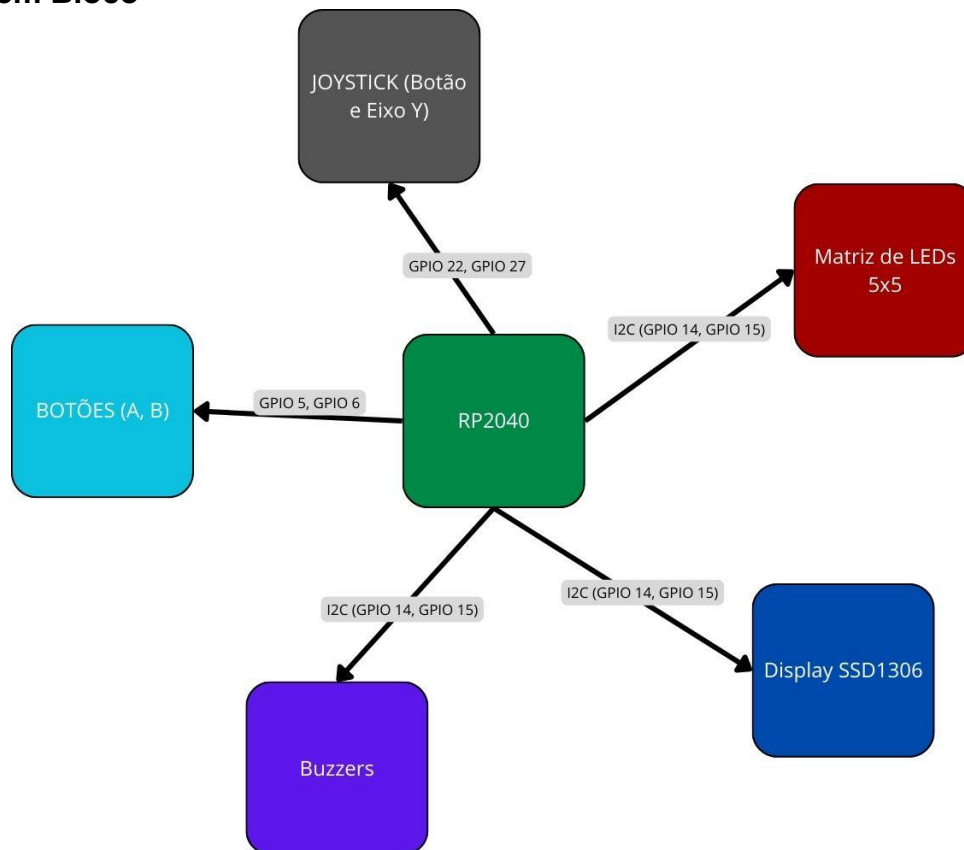
Pesquisas em plataformas como Google Acadêmico, Instructables e repositórios do GitHub revelam diversos projetos de temporizadores baseados em microcontroladores, como o Arduino e o Raspberry Pi Pico. Exemplos incluem timers simples com displays 7-segmentos ou LEDs, mas poucos integram funcionalidades específicas como pausa configurável via joystick, exibição simultânea em um display OLED (SSD1306) e uma matriz de LEDs 5x5 para contagem regressiva visual, além de alertas sonoros com buzzers.

Projetos similares, como o "Pomodoro Timer" da Adafruit, utilizam displays básicos e botões, mas não exploram a interface de joystick ou a combinação de múltiplas saídas visuais e sonoras em uma única plataforma como a BitDogLab. Esta integração, aliada à pausa dinâmica e à interface intuitiva, diferencia o projeto de soluções existentes, oferecendo

uma abordagem inovadora e funcional dentro dos conceitos de Sistemas Embarcados estudados no curso EmbarcaTech.

b. Especificação do hardware

Diagrama em Bloco



Função de Cada Bloco

- Joystick: Configura o tempo em minutos (eixo Y) e ativa/desativa o sistema (botão).
- Botões A e B: Inicia ou reinicia o temporizador (A); alterna entre modo de configuração e execução (B).
- Display SSD1306: Exibe o tempo configurado, o tempo restante e mensagens como "Timer encerrado".
- Matriz de LEDs 5x5: Mostra a contagem regressiva de 9 a 0.
- Buzzers: Emitem cinco bips ao final da contagem.

Configuração de Cada Bloco

- Joystick: Eixo Y conectado ao pino ADC (GPIO 27), botão ao GPIO 22, configurado com pull-up interno.
- Botões A e B: Conectados ao GPIO 5 (A) e GPIO 6 (B), e configurados com pull-up interno.
- Display SSD1306: Conectado via I2C (SDA ao GPIO 14, SCL ao GPIO 15), endereço 0x3C, inicializado com resolução 128x64.
- Matriz de LEDs 5x5: Conectada ao GPIO 7 via PIO, configurada com protocolo WS2812.
- Buzzers: Conectados aos GPIOs 10 e 21, configurados com PWM a 2700 Hz e 90% de duty cycle.
- LEDs RGB: Conectados aos GPIOs 11 (G), 12 (B) e 13 (R), configurados como saídas

(não utilizados).

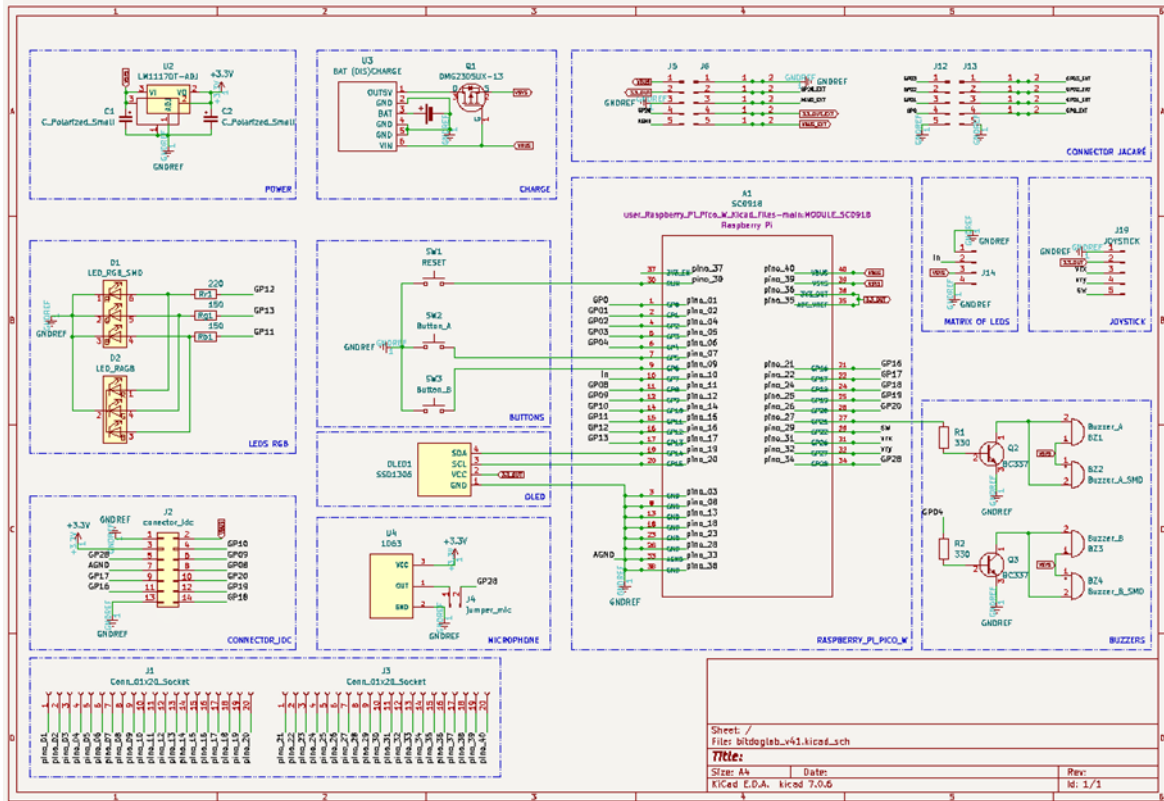
Comandos e Registros Utilizados

- Joystick: `adc_init()`, `adc_gpio_init()`, `adc_select_input()`, `adc_read()` para o eixo Y; `gpio_set_irq_enabled_with_callback()` para o botão.
- Botões: `gpio_set_irq_enabled_with_callback()` para interrupções de borda de descida.
- Display SSD1306: `i2c_init()`, `ssd1306_init()`, `ssd1306_draw_string()`, `ssd1306_send_data()`.
- Matriz de LEDs: `pio_add_program()`, `ws2812_program_init()`, `put_pixel()` via PIO.
- Buzzers: `pwm_irq_setup()`, `pwm_set_enabled()`, `pwm_set_chan_level()`.

Descrição da Pinagem Usada

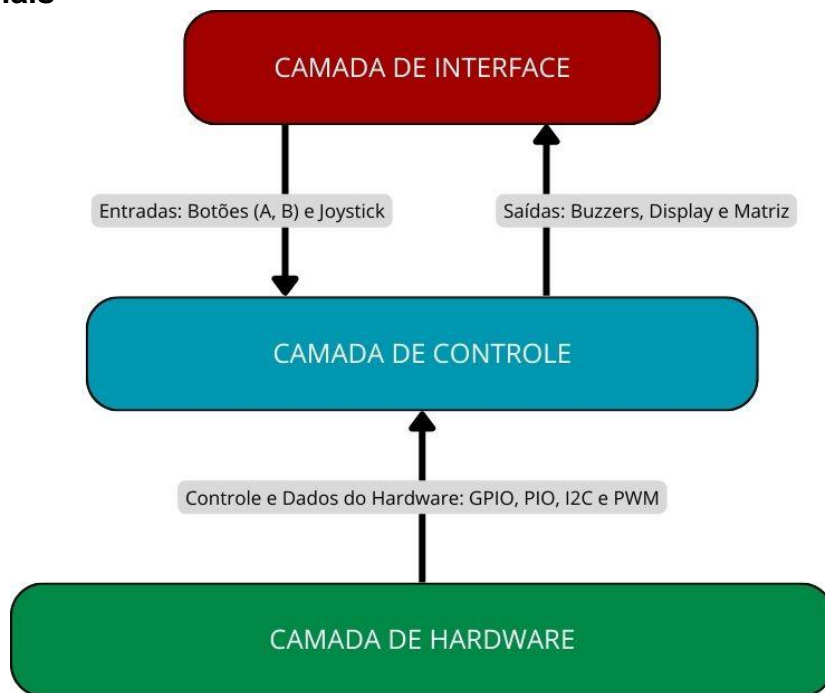
- GPIO 5: Botão A (entrada).
- GPIO 6: Botão B (entrada).
- GPIO 7: Matriz de LEDs (saída PIO).
- GPIO 10: Buzzer 1 (saída PWM).
- GPIO 11: LED RGB Verde (saída).
- GPIO 12: LED RGB Azul (saída).
- GPIO 13: LED RGB Vermelho (saída).
- GPIO 14: SDA do I2C (display SSD1306).
- GPIO 15: SCL do I2C (display SSD1306).
- GPIO 21: Buzzer 2 (saída PWM).
- GPIO 22: Botão do Joystick (entrada).
- GPIO 27: Eixo Y do Joystick (entrada ADC).

Circuito Completo do Hardware



c. Especificação do firmware

Blocos Funcionais



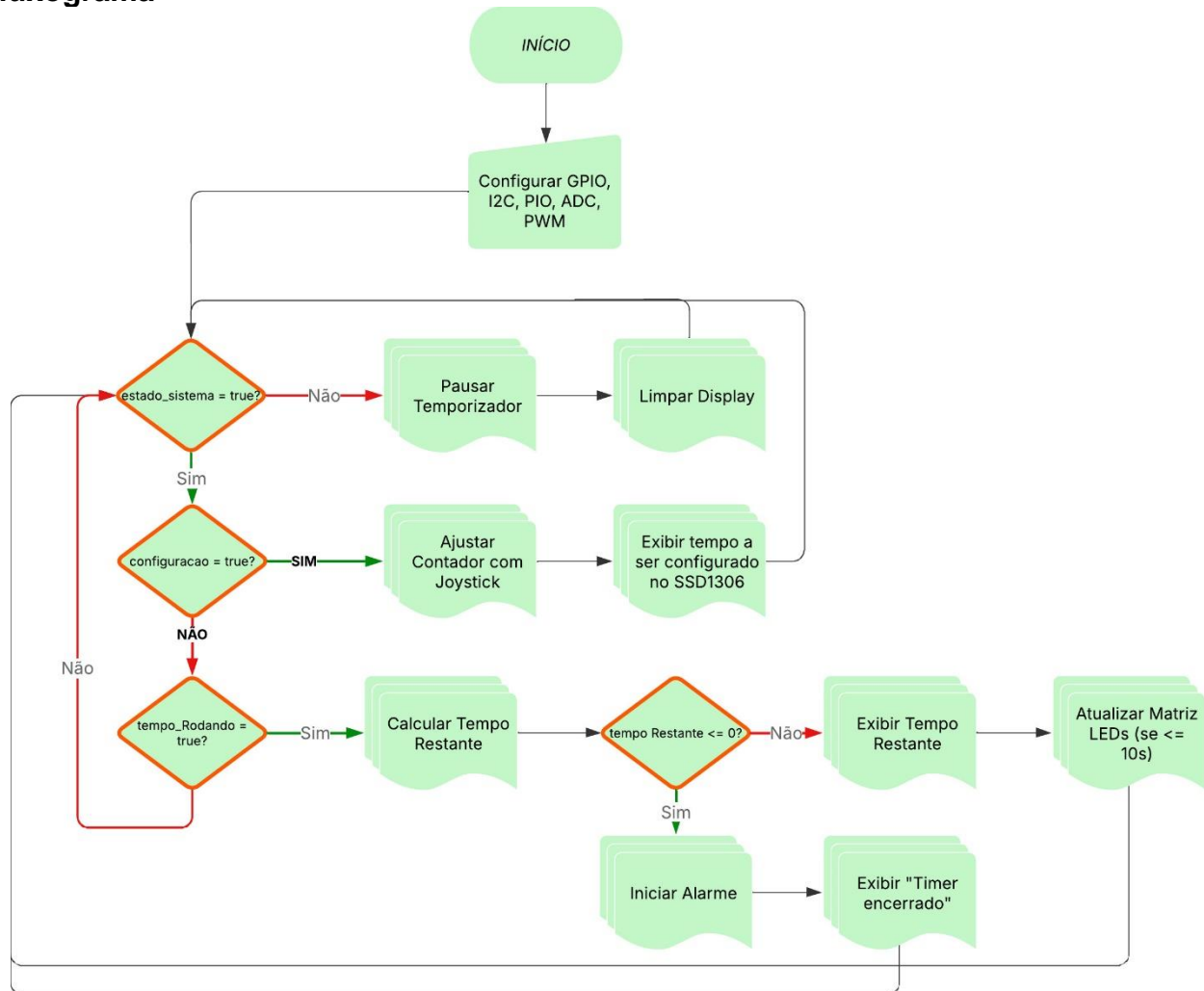
Descrição das Funcionalidades

- Inicialização: Configura GPIO, I2C, PIO, ADC e PWM.
- Interrupções: Gerencia entradas dos botões e joystick via interrupções.
- Controle do Temporizador: Calcula o tempo restante e gerencia pausa/retomada.
- Exibição no Display: Atualiza o SSD1306 com tempo configurado, restante e mensagens.
- Controle da Matriz de LEDs: Exibe contagem regressiva de 10 a 0.
- Controle dos Buzzers: Emite bips ao final da contagem.

Definição das Variáveis

- contador: Tempo configurado em minutos (int).
- tempo_inicio: Timestamp de início (uint64_t).
- tempo_rodando: Estado do temporizador (bool).
- tempo_pausado: Tempo acumulado ao pausar (uint64_t).
- configuracao: Modo do sistema (bool).
- estado_sistema: Sistema ligado/desligado (bool).
- ultimo_buffer: Último valor exibido no display (char[20]).

Fluxograma



Inicialização

O firmware inicializa o I2C (400 kHz), GPIO (entradas/saídas), PIO (WS2812 a 800 kHz), ADC (joystick) e PWM (buzzers a 2700 Hz). Define interrupções de borda de descida para os botões e configura o display SSD1306.

Configurações dos Registros

- I2C: `i2c_init(I2C_PORT, 400 * 1000)` para comunicação com o SSD1306.
- GPIO: `gpio_set_function()` para I2C e PWM, `gpio_set_irq_enabled_with_callback()` para interrupções.
- PIO: `pio_add_program()` e `ws2812_program_init()` para a matriz de LEDs.
- PWM: `pwm_config_set_wrap()` e `pwm_set_chan_level()` para buzzers.

Estrutura e Formato dos Dados

- Tempo configurado: Inteiro em minutos (contador).
- Tempo restante: Float em segundos (`tempo_restante`), exibido como string no display.
- Matriz de LEDs: Array booleano 11x25 (numeros) para padrões de 0 a 10.

Protocolo de Comunicação

- I2C para o SSD1306, sem protocolo adicional de dados.

Formato do Pacote de Dados

Não aplicável, pois não há comunicação externa.

d. Execução do projeto

Metodologia

- Pesquisa: Analisei temporizadores em plataformas como Arduino e Pico, adaptando conceitos para a BitDogLab.
- Escolha do Hardware: Selecionei a BitDogLab por sua integração de joystick, botões, display, LEDs e buzzers.
- Definição do Software: Planejei funções para configuração, contagem, pausa e exibição.
- Inicialização da IDE: Configurei o Pico SDK no VS Code.
- Programação: Implementei em C, iterando com base em testes.
- Depuração: Usei printf() para verificar contagem e interrupções.

Testes de Validação

- Configuração: Testei o ajuste do tempo com o joystick (ex.: 1 a 5 minutos).
- Contagem: Verifiquei a exibição de "60 s" a "0 s" sem piscar.
- Pausa: Confirmei que a contagem pausa ao entrar no modo de configuração e retoma corretamente.
- Finalização: Validei os cinco bips e a exibição de "Timer encerrado" apenas quando configuracao = false.

Discussão dos Resultados

O projeto atende aos requisitos de funcionalidade, baixo custo e tamanho reduzido. A pausa funciona como esperado, e o "Timer encerrado" é exibido corretamente apenas no modo de execução. Pequenos ajustes foram necessários para evitar piscar no display, mas o sistema é confiável e reproduzível.

e. REFERÊNCIAS

- Raspberry Pi Foundation. "Raspberry Pi Pico Datasheet." Disponível em: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>. Acesso em: 26 fev. 2025.
- ADAFRUIT INDUSTRIES. A NeoPixel Pomodoro Timer. [S.l.]: Adafruit Industries, [2021?]. Disponível em: <https://cdn-learn.adafruit.com/downloads/pdf/a-neopixel-pomodoro-timer.pdf>. Acesso em: 26 fev. 2025.
- INSTRUCTABLES. Raspberry Pi Pico MX7219 Eight Digits of Seven Segment. Instructables, [2022?]. Disponível em: <https://www.instructables.com/Raspberry-Pi-Pico-MX7219-Eight-Digits-of-Seven-Seg/>. Acesso em: 26 fev. 2025.
- LANZNASTER, Luiz Carlos. Temporizador digital para controle de reuniões. 2020. 68 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Regional de Blumenau, Blumenau, 2020. Disponível em: https://professorpetry.com.br/Ensino/Defesas_Pos_Graduacao/Defesa%2049_Luiz%20Carlos%20Lanznaster_Temporizador%20Digital%20Para%20Controle%20de%20Reunioes.pdf. Acesso em: 26 fev. 2025.
- LELEPG. Trabalho Final Microcontroladores. GitHub, 2023. Disponível em: <https://github.com/LelePG/Trabalho-Final-Microcontroladores/blob/main/README.md>. Acesso em: 26 fev. 2025.

f. Links relacionados

- **Vídeo de demonstração**
- **Repositório no GitHub**