



MARCO ANTONIO DE JESUS SATURNINO

# **SISTEMA DE MONITORAMENTO DE PESSOAS ACAMADAS: UMA SOLUÇÃO INTEGRADA**

Petrolina, PE

2025

# SISTEMA DE MONITORAMENTO DE PESSOAS ACAMADAS:

## UMA SOLUÇÃO INTEGRADA

### 1. Escopo do Projeto

#### a. Apresentação do Projeto

O projeto intitulado “Sistema de Monitoramento de Pessoas Acamadas” consiste em um “Leito Inteligente” que se consolida como um sistema embarcado voltado para o monitoramento de pacientes acamados. Ele visa aprimorar o acompanhamento da saúde do paciente e fornecer informações relevantes para cuidadores e profissionais da saúde em domicílio. Para isso, o sistema contará com sensores para monitoramento contínuo do paciente, interface gráfica para visualização dos dados e mecanismos de controle para ajuste da posição do leito. A solução possui potencial para evolução em um sistema IoT, possibilitando maior conectividade e automação.

O projeto será implementado utilizando o microcontrolador **RP2040**, mais especificamente a placa **Raspberry Pi Pico W**, para garantir conectividade sem fio e eficiência no processamento.

#### b. Objetivos

- Monitorar em tempo real a frequência cardíaca, saturação de oxigênio do paciente e a temperatura;
- Monitorar a umidade da fralda usada pelo paciente;
- Fornecer um botão de pânico para chamadas de emergência e sinalização;
- Apresentar informações no display gráfico integrado;
- Estabelecer comunicação com um computador de borda via conexão serial CDC;
- Implementar LEDs RGB para indicação do status do paciente;
- Controlar a altura do leito e a posição da cabeceira via PWM;
- Viabilizar a evolução futura do sistema com integração de controle por voz e análise de dados diários.

#### c. Descrição do Funcionamento

O sistema embarcado será integrado a um leito e contará com os seguintes componentes:

- **Microcontrolador RP2040 (Raspberry Pi Pico W):** unidade de controle principal, responsável pelo processamento e comunicação;
- **Sensores de monitoramento:** dispositivos para medição da frequência cardíaca, saturação de oxigênio, temperatura e umidade;
- **Botão de pânico:** interface para o paciente ou cuidador solicitar ajuda em situações emergenciais;

- **Botão para campainha:** Toca campainha para solicitar presença do cuidador;
- **Display gráfico:** tela para exibição dos dados monitorados e informações do sistema;
- **Módulo de comunicação serial CDC:** interface para transmissão de dados a um computador de borda;
- **LEDs RGB indicativos:** iluminação para sinalização do status do paciente e do sistema;
- **Mecanismo de ajuste do leito:** controle de altura do leito e inclinação da cabeceira utilizando acionamento PWM.

#### d. Justificativa

Este projeto surge da necessidade de aprimorar o monitoramento de pacientes acamados em domicílio particular ou em ambientes que não sejam exclusivamente voltados à atenção à saúde, proporcionando maior segurança e conforto. Com o crescimento da população idosa e o aumento da prevalência de doenças crônicas, torna-se essencial o desenvolvimento de soluções tecnológicas que permitam um acompanhamento contínuo e acessível.

A solução se justifica pelos seguintes fatores:

- **Monitoramento contínuo:** possibilita a rápida identificação de alterações nos sinais vitais do paciente, permitindo intervenções precoces e reduzindo riscos à saúde;
- **Aumento da segurança:** o botão de pânico proporciona uma forma rápida e eficiente de comunicação para situações emergenciais, garantindo assistência imediata ao paciente;
- **Facilidade de operação:** a interface intuitiva do sistema permite que tanto cuidadores quanto os próprios pacientes possam utilizá-lo sem dificuldades, tornando o monitoramento acessível para diferentes perfis de usuários;
- **Automação e conectividade:** o projeto prevê a integração futura com tecnologias IoT, permitindo a análise preditiva dos dados coletados e possibilitando a implementação de comandos por voz para maior comodidade.

A necessidade de sistemas como este se intensifica à medida que a população brasileira envelhece e cresce a demanda por cuidados domiciliares eficientes e acessíveis. O desenvolvimento de soluções inovadoras voltadas para esse público pode contribuir significativamente para a melhoria da qualidade de vida dos pacientes e facilitar o trabalho dos cuidadores e familiares.

## e. Pesquisa de Anterioridade

Leitos inteligentes já são uma realidade em diversos centros de pesquisa e hospitais avançados. Esses sistemas são projetados com propósito médico, focados no monitoramento detalhado da saúde do paciente e na facilitação do trabalho de profissionais da área da saúde (G1, 2018). No entanto, grande parte dessas soluções são robustas e voltadas para ambientes hospitalares, apresentando alto custo e infraestrutura complexa.

Assim, o desenvolvimento de um sistema que permita o acompanhamento domiciliar de forma acessível já desperta interesse de pesquisadores e empresas do setor. A possibilidade de oferecer um monitoramento contínuo, aliado à facilidade de uso e à conectividade, abre caminho para novas aplicações voltadas ao bem-estar de pacientes acamados e idosos que necessitam de atenção especial.

Deste modo, a pesquisa de anterioridade revela que, embora a tecnologia de leitos inteligentes esteja em desenvolvimento e aplicação no meio hospitalar, ainda há um vasto campo aberto para soluções acessíveis e adaptáveis ao ambiente domiciliar. Dessa forma, a proposta deste projeto se destaca por viabilizar uma alternativa economicamente viável, de fácil implementação e com potencial de ampliação para integração com IoT e automação. Essa abordagem se alinha ao crescente interesse público em tecnologias assistivas e à necessidade de inovação no setor de cuidados domiciliares.

## 2. Especificações do Hardware

### a. Diagrama em Blocos

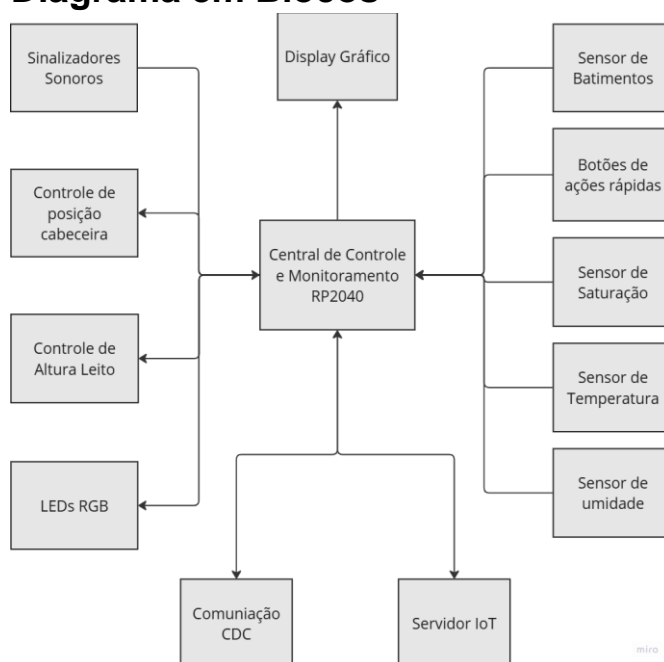


Figura 1: diagrama em blocos do sistema (fonte: autor)

O diagrama em blocos se constitui como uma ferramenta visual que facilita a visualização sistemática do projeto, permitindo um melhor entendimento e interpretação das funcionalidades implementadas. No diagrama apresentado na **Figura 1**, ao centro está a **central lógica do sistema**, responsável pelo processamento e controle das funções. Acima e à esquerda estão os **dispositivos de saída**, que fornecem informações visuais e sonoras ao usuário. À direita encontram-se os **dispositivos de entrada**, que captam dados do paciente. Na parte inferior, estão os **dispositivos bidirecionais**, que atuam tanto enviando dados ao microcontrolador quanto recebendo dados deste.

A seguir, são descritas as funcionalidades de cada bloco

- **Display Gráfico:** Responsável por exibir informações relevantes ao usuário, como sinais vitais do paciente, status do sistema e notificações. Possui uma interface intuitiva para facilitar a visualização dos dados;
- **Sensor de batimentos:** Mede a frequência cardíaca do paciente em tempo real, permitindo a detecção de possíveis alterações no ritmo cardíaco;
- **Botões de ações rápidas:** Inclui um botão de pânico para acionamento em situações emergenciais e um botão para acionamento de campainha;
- **Sensor de saturação:** Monitora o nível de oxigenação no sangue, auxiliando no acompanhamento da saúde respiratória do paciente;
- **Sensor de temperatura:** Mede a temperatura corporal do paciente, fornecendo um indicativo de febre ou hipotermia;
- **Sensor de umidade:** Captura o nível de umidade de fraldas, alertando da necessidade de substituição;
- **Servidor IoT:** Funcionalidade prevista para futura implementação, possibilitando a integração do sistema à internet para envio e armazenamento de dados em nuvem e implementação de outros recursos remotos;
- **Comunicação serial CDC:** Protocolo de comunicação utilizado para conectar o sistema a um computador de borda, permitindo troca de dados e registros de monitoramento;
- **LEDs RGB:** Indicadores visuais que sinalizam diferentes estados do sistema, como funcionamento normal, alerta ou erro;
- **Controle de altura:** Sistema de ajuste automatizado que permite elevar ou abaixar a altura do leito conforme a necessidade do paciente e/ou cuidadores;
- **Controle de posição:** Ajusta a inclinação da cabeceira do leito, proporcionando maior conforto e auxiliando em tratamentos médicos;
- **Sinalizadores sonoros:** Alarmes sonoros que alertam o usuário em caso de emergência, falhas no sistema ou necessidade de atenção. É previsto futura implementação de síntese de voz para aprimoramento da comunicação e da acessibilidade.

## b. Descrição do Hardware

O coração do sistema é o **RP2040**, um microcontrolador versátil e robusto que deve equilibrar performance e custo, viabilizando a construção de um produto capaz de ajudar famílias das mais variadas camadas sociais. Ele

possui dois núcleos ARM Cortex-M0+, operando a até 133 MHz, além de 264 KB de SRAM e suporte a interfaces como I2C, SPI, UART e PWM, além de um conversor de 12 bits.

Um dos diferenciais do RP2040 é seu **módulo PIO (Programmable I/O)**, por meio do módulo PIO, o microcontrolador fornece até 8 máquinas de estado, em dois blocos separados e com acesso a todas as GPIO, que podem funcionar em paralelo a CPU o que permite o desenvolvimento de uma gama muito grande de aplicações e protocolos sem sobrecarregar a CPU. Além disso, o Raspberry Pi Pico W, por meio da sua conectividade wireless possibilita futuras expansões para integração com IoT.

Os pinos do RP2040 serão configurados de acordo com a **Tabela 1** e distribuídos para interligação com os sensores, atuadores e demais periféricos do sistema

Para garantir um monitoramento eficiente e interatividade adequada, os seguintes componentes serão utilizados

- **Raspberry Pi Pico W:** Placa de desenvolvimento com o microcontrolador principal do sistema;
- **LEDs [WS2812](#):** LEDs RGB endereçáveis para indicação de status do sistema (usado apenas no modo pânico);
- **Display [SSD1306](#):** Display OLED para exibição de informações vitais;
- **Joystick:** Controle manual para ajustes de posição do leito, do volume dos buzzers e do brilho dos LEDs RGB (com exceção da matriz de LEDs);
- **Botões:** Inclui botão do pânico, botão para tocar campainha (alarme) e botão para alterar a função do joystick;
- **Buzzers passivos e circuito de interface:** Emissor sonoro para alertas e notificações;
- **LEDs RGB:** Indicadores visuais complementares para sinalizações específicas;
- **Sensores I2C de [temperatura](#):** Monitoramento contínuo da temperatura do paciente;
- **Sensores I2C de [saturação](#):** Monitoramento facilitado do nível de oxigenação do sangue;
- **Sensor de batimentos cardíacos [AD8232](#):** Sensor avançado para detecção e análise do ECG do paciente;
- **Circuito detector de umidade:** Sistema simples de umidade para indicar a necessidade de troca do colchão ou da roupa de cama (saída digital);
- **Resistores, capacitores, conectores:** Componentes auxiliares para a montagem do circuito eletrônico.

Além dos recursos disponíveis na **BitDogLab**, que já serão utilizados nesta versão para validação do sistema, há também componentes planejados para integração futura. Esses componentes foram projetados considerando a



disponibilidade de pinos na **BitDogLab**, garantindo compatibilidade e escalabilidade do sistema. Entre eles, destacam-se: o sensor de batimentos [AD8232](#), um sensor robusto capaz de medir a frequência cardíaca do paciente de forma confiável; o sensor [MAX30102](#) que é uma aplicação de fotopletismografia capaz de monitorar os batimentos e a saturação de oxigênio; e o sensor [MCP9808](#) que é um sensor de temperatura digital de alta precisão.

**Tabela 1: Pinout Raspberry Pi Pico W**

PINO	FUNÇÃO	PERIFÉRICO	APLICAÇÃO	DESCRIÇÃO
GP0	UART RX	UART0	Interface de comunicação externa	Comunicação CDC com computador de borda
GP1	Saída	PWM	Saída PWM	Controle de altura do leito
GP2	Saída	PWM	Saída PWM	Controle de posição da cabeceira do leito
GP3	Entrada	GPIO	Entrada Digital	Implementar um sensor de umidade para detectar estado da fralda
GP4	Entrada	GPIO	Entrada digital - botão A	Botão de pânico para emergências
GP5	Entrada	GPIO	Entrada digital - botão B	Botão para tocar campainha interna
GP6	Saída	GPIO	Saída digital - matriz de LEDs	Controle de matriz de LEDs 5x5
GP7	I2C SDA	I2C1	Comunicação I2C com oxímetro e sensor de temperatura	Módulos externos para medir a concentração de oxigênio no sangue e a temperatura
GP8	I2C SCL	I2C1	Comunicação I2C com oxímetro e sensor de temperatura	Módulos externos para medir a concentração de oxigênio no sangue e a temperatura
GP9	Saída	PWM	Saída PWM - buzzer B	Periférico para geração de sinais sonoros
GP10	Saída	GPIO	Saída digital - LED verde	Periférico para geração de sinais luminosos
GP11	Saída	GPIO	Saída digital - LED azul	Periférico para geração de sinais luminosos
GP12	Saída	GPIO	Saída digital - LED vermelho	Periférico para geração de sinais luminosos
GP13	I2C SDA	I2C1	Comunicação I2C com display	Interface de Comunicação com o usuário
GP14	I2C SCL	I2C1	Comunicação I2C com display	Interface de Comunicação com o usuário
GP15	SPI RX	SPI0	Comunicação com memória	MISO do barramento
GP16	SPI CS	SPI1	Comunicação com memória	Seleciona a memória
GP17	SPI SCK	SPI2	Comunicação com memória	Clock do barramento
GP18	SPI TX	SPI3	Comunicação com memória	MOSI do barramento
GP19	-	-	-	-
GP20	Saída	PWM	Saída PWM - buzzer A	Periférico para geração de sinais sonoros
GP21	Entrada	GPIO	Entrada digital - botão Joystick	Mudar função do Joystick
GP22	-	-	-	-
GP23	-	-	-	-
GP24	-	-	-	-
GP25	Entrada	ADC	Entrada analógica - Eixo Y joystick	Controla a altura do leito

GP26	Entrada	ADC	Entrada analógica - Eixo X joystick	Controla a posição da cabeceira
GP27	Entrada	ADC	Entrada analógica - MIC	Entrada analógica para ECG
GP28	-	-	-	-
GP29	UART RX	UART0	Interface de comunicação externa	Comunicação CDC com computador de borda

### c. Controle do Hardware

O controle do hardware será realizado pelo **microcontrolador RP2040**, utilizando o **SDK oficial do Raspberry Pi Pico**. Esse SDK fornece bibliotecas de baixo nível e também de alto nível, otimizadas para **GPIO, I2C, SPI, UART e PWM**, facilitando a implementação das funcionalidades do sistema.

As **saídas digitais** serão configuradas inicializando os pinos correspondentes e definindo sua direção como saída. Já as **entradas digitais** seguirão uma configuração semelhante, porém todos os botões serão configurados com resistores de pull-up disponíveis internamente no RP2040, visto que, na BitDogLab, os botões estão conectados ao GND da placa. Essa abordagem garante que os botões permaneçam em nível lógico alto quando não pressionados, evitando leituras instáveis.

A comunicação serial será implementada utilizando a **UART0** do RP2040, configurada para um **baud rate de 115200** bauds, garantindo comunicação eficiente com o computador de borda. Os pinos designados para a UART serão: **GPIO 0 - TX** e **GPIO 1 - RX**.

A UART será configurada com **8 bits de dados, 1 bit de parada e sem paridade (8N1)**, garantindo compatibilidade com padrões convencionais de comunicação serial.

O display **SSD1306** está conectado aos pinos **14 (SDA)** e **15 (SCL)** na placa **BitDogLab**, utilizando o periférico **I2C1**, e essa configuração será mantida no projeto. A comunicação ocorrerá a uma velocidade de **400 kHz**, garantindo compatibilidade com o display.

Para os módulos externos a **BitDogLab**, será utilizado um barramento separado, alocado nos pinos **GPIO 8 (SDA)** e **GPIO 9 (SCL)**, conectados ao periférico **I2C0**. A velocidade de operação também será **400 kHz**, garantindo um funcionamento estável e padronizado.

A tabela a seguir apresenta os endereços I2C padrão de cada dispositivo utilizado no sistema.

Tabela 2: Endereço Dispositivos I2C	
Dispositivo	Endereço I2C (Hexadecimal)
Display SSD1306	0x3C
Sensor MAX30102	0x57
Sensor MCP9808	0x18



As **saídas PWM**, serão configuradas, utilizando as bibliotecas da SDK, para operar a uma frequência ajustável visando o melhor desempenho para cada uma das aplicações previstas.

As **entradas analógicas** também foram configuradas utilizando a SDK, visando uma taxa de amostragem adequada aos sinais monitorados.

Por fim, cabe dizer que os pinos GPIO de 16 a 19 foram configurados para dar suporte a um **barramento SPI**. Futuramente, este barramento será utilizado para conexão com uma memória externa.

#### d. Circuito do Hardware

O circuito eletrônico foi desenvolvido com base na BitDogLab, proporcionando uma implementação mais rápida e simplificada. Dessa forma, a alimentação principal será fornecida via cabo USB, conectado à placa Raspberry Pi Pico W.

Para garantir a operação em caso de falha no fornecimento de energia, o sistema contará com uma bateria, permitindo o funcionamento de alguns recursos essenciais, ainda que de forma limitada.

Além dos periféricos nativos da BitDogLab, o projeto prevê a integração futura de dispositivos externos, que serão implementados após a validação da fase inicial. A BitDogLab disponibiliza, por meio do conector de expansão, além dos pinos que serão usados para tráfego de dados, pinos de alimentação 3,3V, 5V e GND, possibilitando a energização desses dispositivos de maneira eficiente (antes da consolidação dessa etapa, deve-se analisar a viabilidade por meio do estudo da capacidade de fornecimento de energia e da demanda dos dispositivos).

Por fim, cabe dizer que o diagrama elétrico da placa BitDogLab pode ser acessado clicando [aqui](#) e, visando manter-se sucinto, não será detalhado no escopo deste projeto.

### 3. Especificações do Firmware

#### a. Blocos Funcionais

Os blocos funcionais do firmware estão organizados de acordo com o modelo de camadas de sistemas embarcados, que é composto por três níveis principais: **camada de hardware**, **camada de software básico** e **camada de software de aplicação**. Essa organização modular permite uma melhor separação das responsabilidades, facilitando a manutenção e a escalabilidade do sistema.

A camada de hardware contém os componentes físicos do sistema, incluindo sensores, atuadores e microcontroladores. Neste projeto, o hardware inclui:

- Raspberry Pi Pico W;
- Sensores (MAX30102, AD8232, MCP9808);
- Displays (SSD1306, LEDs);
- Dispositivos de comunicação (CYW43439);
- Periféricos (BitDogLab);
- Módulo memória SD.

As informações relevantes relacionadas a camada de hardware foram devidamente apresentadas no tópico 2, onde todo o processo foi detalhado.

A camada de software básico está subdividida em dois blocos: *device drivers* e protocolos e serviços. O bloco “*device drivers*” é responsável por modularizar e abstrair os elementos de hardware, atuando como uma ponte entre os dispositivos físicos e as camadas superiores do sistema. Os drivers garantem que os componentes de hardware sejam acessados de maneira padronizada e eficiente, permitindo que a aplicação interaja com sensores, atuadores e periféricos sem a necessidade de lidar diretamente com os registradores e configurações de baixo nível.

Principais drivers implementados no sistema:

- **GPIO:** Controle de pinos de entrada e saída digital.
- **ADC:** Conversão analógica-digital para sensores.
- **PWM:** Geração de sinais modulares para controle de dispositivos.
- **UART:** Comunicação serial assíncrona.
- **I2C:** Comunicação com sensores e periféricos.
- **SPI:** Comunicação com memória externa.
- **Wi-Fi & Bluetooth:** Controle das interfaces sem fio.
- **TIMER:** Gestão de temporização para eventos assíncronos.
- **CLOCK:** Configuração do sistema de temporização do microcontrolador.

O bloco de protocolos e serviços gerencia a comunicação do sistema, garantindo a troca de informações entre os módulos internos e dispositivos externos. Para melhor organização, ele pode ser subdividido em dois grupos:

- **Protocolos de Transporte:** Esses protocolos são responsáveis pelo envio e recepção de dados entre dispositivos, garantindo integridade e confiabilidade na comunicação. Podemos citar:
  - **TCP/IP:** Comunicação em rede baseada em internet. Pretende-se implementar o protocolo MQTT ou HTTP.
  - **Bluetooth:** Troca de dados sem fio em curto alcance.
  - **Serial CDC:** Comunicação USB serial para interface com computadores.
- **Protocolos de Hardware:** Esses protocolos permitem a comunicação entre o microcontrolador e seus periféricos. Eles definem como os dados são organizados e interpretados no nível físico.
  - **I2C:** Protocolo serial para sensores e displays.
  - **SPI:** Protocolo serial para comunicação com memória externa.
  - **UART:** Comunicação direta com periféricos seriais.

- **PWM:** Controle de sinais analógicos simulados via modulação por largura de pulso.
- **GPIO:** Manipulação de estados lógicos de entrada e saída.
- **ADC:** Conversão de sinais analógicos em digitais.

Essa organização modular da camada de software básico permite escalabilidade, simplifica a manutenção e facilita a integração de novos dispositivos e protocolos, conforme necessário.

Por fim, a camada de aplicação implementa as funcionalidades específicas do sistema embarcado. Nesta camada estão presentes os seguintes blocos funcionais:

- **Interface Homem Máquina (IHM):** periféricos de interação com o usuário;
- **Monitoramento:** leitura dos sinais disponíveis;
- **Controle:** Implementação do sistema de controle a partir dos parâmetros monitorados/recebidos;
- **Comunicação:** estabelece a comunicação do sistema com o mundo externo.

A Figura 2 ilustra graficamente como isso está estruturado.

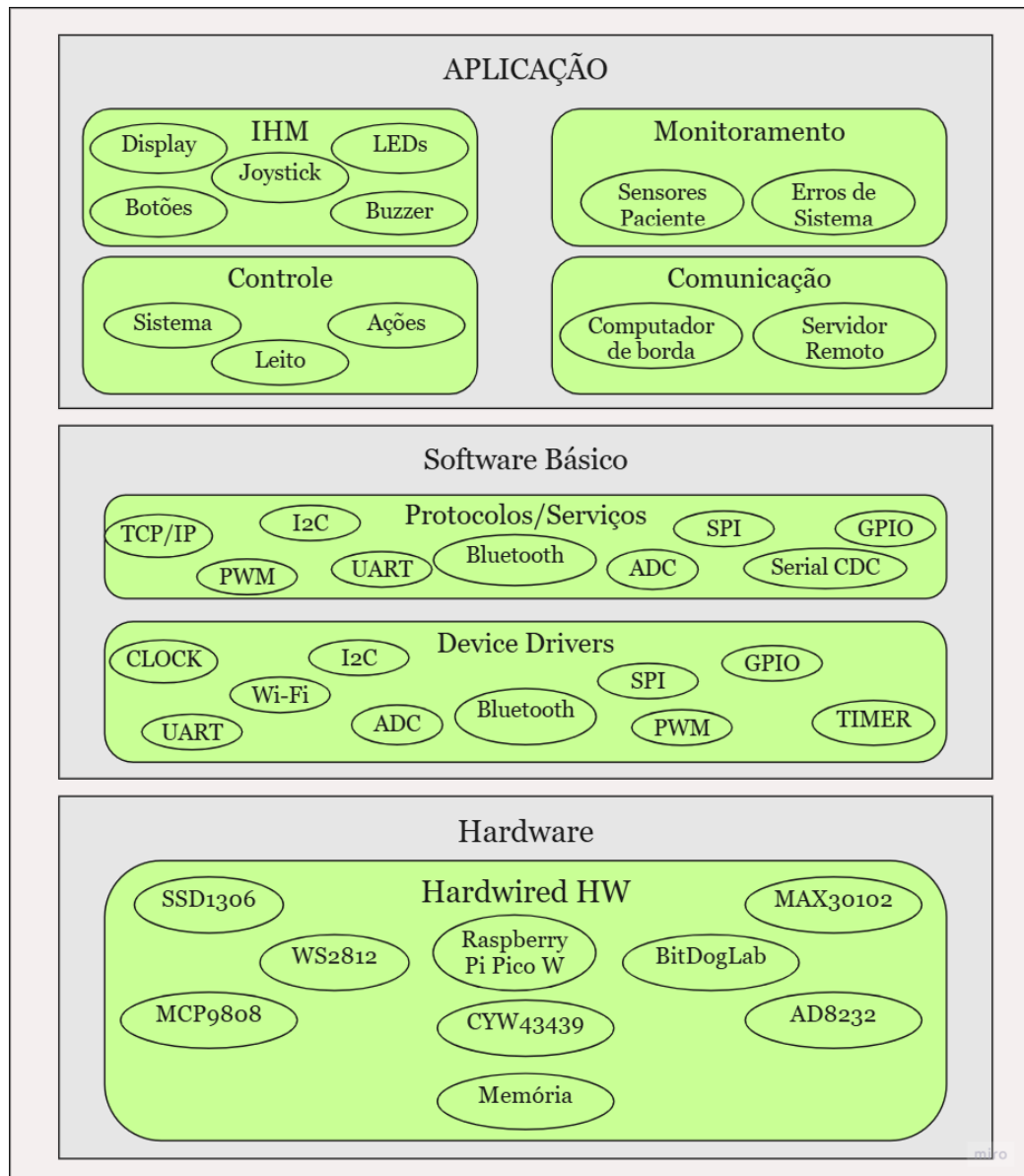


Figura 2: Diagrama em blocos do sistema embarcado (fonte: autor).

## b. Funcionalidades

As principais funcionalidades do firmware em sua versão inicial devem integrar:

- Leitura de sensores biomédicos;
- Processamento e envio de dados;
- Controle de dispositivos de saída como LEDs e buzzer;
- Geração de sinais PWM para atuadores;
- Interação com o usuário por meio da IHM.

Para versões futuras deve ser implementado suporte a comunicação, armazenamento e análise de dados remotamente. Uma memória SD deve garantir o armazenamento dos dados e configurações para o modo off-line.

### c. Definição de Variáveis

O sistema é bastante complexo e se utiliza de grande quantidade de variáveis para operar o funcionamento de forma adequada. Neste sentido, a tabela 3 abaixo apresenta as variáveis utilizadas, bem como seu tipo e uma breve descrição.

Tabela 3: Variáveis do Sistema		
Nome	Tipo	Descrição
ssd	ssd1306_t	Estrutura do display OLED
cor	bool	Cor do display
valores_ref_joy	uint16_t[6]	Vetor de calibração do joystick
buffer	char[4]	Buffer para strings
buffer_erro	char[11]	Buffer para códigos de erro
valor_adc	uint16_t	Valor do ADC
valor_adc_passado	uint16_t	Valor anterior do ADC
tempo_de_amostragem	absolute_time_t	Tempo para amostragem do microfone
amostras_por_segundo	const uint8_t	Taxa de amostragem
adc_value_x	uint16_t	Valor do ADC do eixo X
adc_value_y	uint16_t	Valor do ADC do eixo Y
passadoX	uint16_t	Valor anterior do eixo X
passadoY	uint16_t	Valor anterior do eixo Y
tempo_joy	uint64_t	Tempo do joystick
top_wrap	volatile uint16_t	Parâmetro para controle de tempo
erro_flag	volatile bool	Flag de erro
Slogan	uint8_t[1024]	Buffer para slogan
fig_principal	uint8_t[1024]	Buffer para figura principal
apagado	uint32_t[25]	Buffer para estado "apagado"
open	uint32_t[25]	Buffer para estado "aberto"
passado	volatile uint32_t	Valor anterior para debounce
flag_botoes	volatile uint8_t	Flag de eventos de interrupção
func_bot_joy	volatile uint8_t	Função executada pelo joystick
modo_panico	volatile bool	Ativa modo pânico/emergência
control	volatile bool	Variável de controle
freq	volatile uint16_t	Frequência do sistema
freq_min	volatile uint16_t	Frequência mínima
freq_max	volatile uint16_t	Frequência máxima
volume_buzA	volatile uint8_t	Volume do buzzer A
volume_buzB	volatile uint8_t	Volume do buzzer B
altura_leito	volatile uint8_t	Altura do leito
pos_cabeceira	volatile uint8_t	Posição da cabeceira
brilho_matriz	uint8_t	Controle do brilho da matriz
pio	PIO	Objeto PIO para controle periférico
offset	int	Offset usado no PIO
sm	int	State Machine do PIO

#### d. Fluxograma

A Figura 3 apresenta o fluxograma do programa principal. O sistema opera com base em uma interrupção acionada pelos botões A, B e Joy, que ativam uma flag responsável por controlar suas ações. Além disso, determinados comandos podem ser enviados a partir do computador de borda, permitindo a interação e o ajuste do funcionamento do sistema conforme necessário.

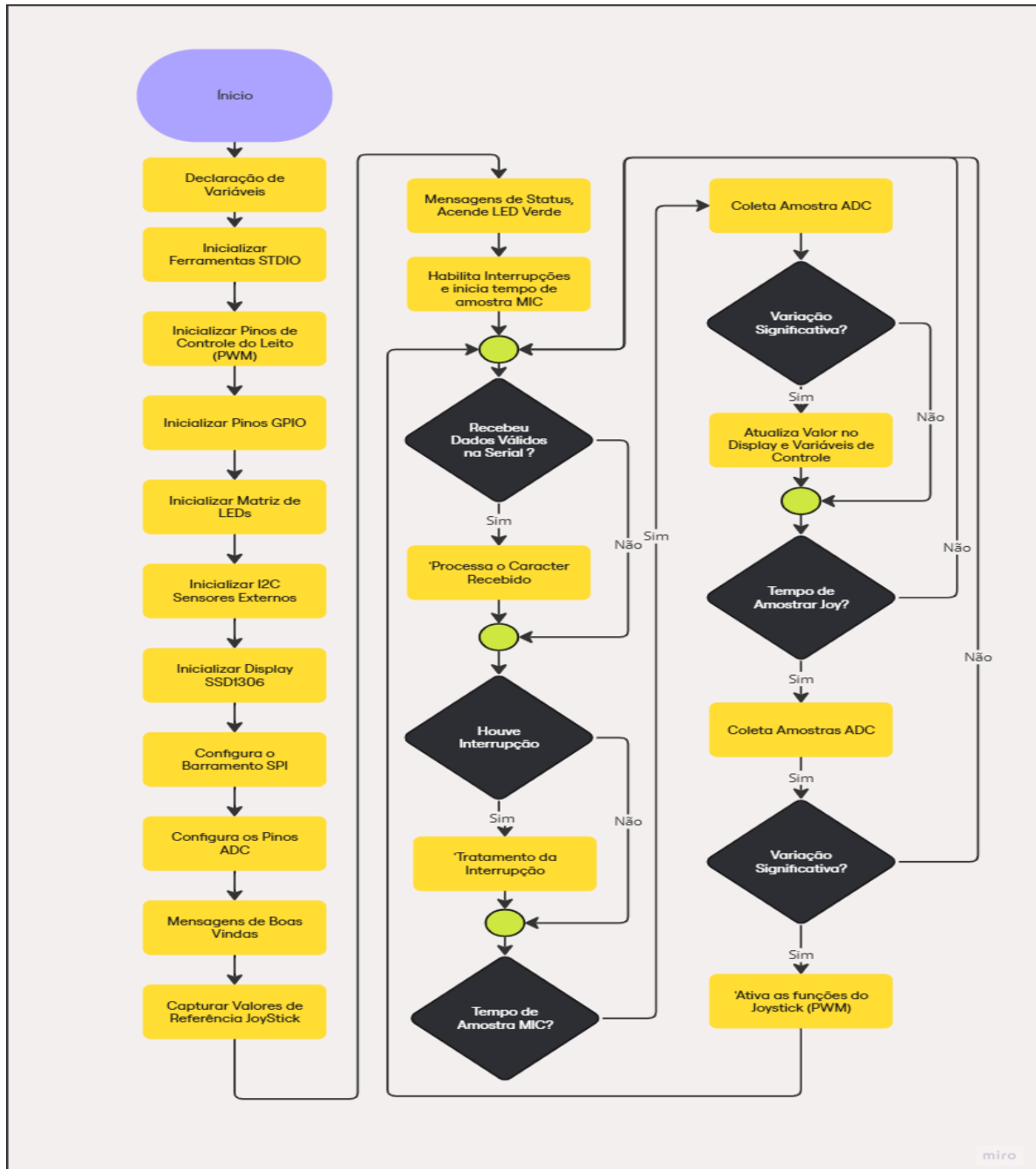


Figura 3: Fluxograma do Programa Principal (fonte: autor)



## e. Inicialização

A inicialização do sistema embarcado segue um fluxo estruturado para garantir que todos os módulos sejam configurados corretamente antes do funcionamento principal. O processo compreende as seguintes etapas:

- **Configuração do *Clock* do sistema:** Ajuste do oscilador interno do RP2040 para a frequência operacional desejada;
- **Inicialização dos Periféricos:** Configuração das interfaces UART, I2C, SPI, GPIO e PWM conforme a necessidade de cada componente;
- **Verificação dos Sensores e Dispositivos:** Checagem inicial da comunicação com os sensores de batimentos cardíacos, saturação, temperatura e umidade;
- **Inicialização do Display e Indicadores Visuais:** Configuração do display SSD1306 e dos LEDs RGB para exibição de status;
- **Configuração do Sistema de Interrupções:** Definição de rotinas para capturar eventos dos botões A, B e Joystick;
- **Estabelecimento da Comunicação Serial:** Configuração da UART para comunicação com o computador de borda;
- **Modo Operação:** Após a inicialização, o sistema entra no modo de monitoramento contínuo, aguardando interações e processamento de dados. Nesta etapa, caso os sensores estejam conectados, serão lidos monitorados continuamente.

## f. Configurações dos Registros

A configuração dos registros no RP2040 é realizada para otimizar o funcionamento dos periféricos. As principais configurações incluem:

- **GPIOs:** Configurados como entrada ou saída conforme a tabela de pinos do projeto, ativando pull-ups internos quando necessário.
- **PWM:** Frequência ajustável para controle de motores, sinalização sonora e luminosa.
- **UART:** Configuração em 115200 bauds, 8N1, garantindo comunicação eficiente com o computador de borda.
- **I2C:** Velocidade de 400 kHz para comunicação rápida e estável com sensores e display.
- **SPI:** Configurado para futura integração com memória externa.
- **Interrupções:** Configuração das flags e rotinas para captura de eventos dos botões e demais interações do usuário.

## g. Estrutura e Formato dos Dados

O sistema completo opera com um grande volume de dados, organizados em quatro categorias principais: controle, entrada, saída e monitoramento. Para garantir um desempenho eficiente, é essencial que toda

essa cadeia de informações esteja bem ajustada e se comunique de forma adequada. Quanto aos dados específicos, podemos citar:

- **Frequência Cardíaca:** Valor em BPM (Batimentos por Minuto).
- **Saturação de Oxigênio:** Percentual de SpO2.
- **Temperatura Corporal:** Medida em Celsius.
- **Estado da Fralda:** Indicação binária (1 - Troca necessária, 0 - Normal).
- **Posição do Leito:** Valores de altura e inclinação.

#### h. Protocolo de Comunicação

A comunicação entre o sistema embarcado e o computador de borda ocorre via UART, utilizando um protocolo baseado em comandos simples. Além disso, o sistema faz uso do barramento I2C para comunicação com sensores de saturação de oxigênio e temperatura, garantindo uma transmissão eficiente e confiável dos dados coletados. O barramento SPI está reservado para futura integração com uma memória externa, ampliando as possibilidades de armazenamento local.

No futuro, prevê-se a implementação de comunicação sem fio utilizando os protocolos Wi-Fi e Bluetooth, permitindo maior conectividade e monitoramento remoto do sistema. Para a integração com redes externas e aplicações IoT, pretende-se utilizar protocolos de aplicação como MQTT ou HTTP, viabilizando o envio de dados a servidores em nuvem e possibilitando o acesso remoto por profissionais da saúde ou cuidadores.

#### i. Formato de Pacotes de Dados

Os dados na comunicação com o computador de borda são organizados em pacotes compatíveis com a configuração 8N1. Para a comunicação com o display, é utilizada uma biblioteca apropriada que gerencia a interface de forma eficiente.

### 4. Execução do Projeto

#### a. Metodologia

A implementação do sistema segue um ciclo iterativo baseado em desenvolvimento, teste e otimização. A abordagem metodológica inclui:

- **Pesquisa de Anterioridade:** investigação minuciosa de soluções semelhantes;
- **Definição dos Requisitos:** Especificação dos componentes necessários, bem como do conhecimento técnico e do tempo;
- **Desenvolvimento do Firmware:** Implementação dos drivers e algoritmos de monitoramento;
- **Prototipagem:** Testes iniciais em bancada para validar funcionamento;
- **Testes de Validação:** Verificação da precisão das medições e resposta dos atuadores;

- **Ajustes e Otimizações:** Refinamento do código e calibração dos sensores;
- **Documentação e Relatório Final:** Registro detalhado do funcionamento e desempenho do sistema.

## b. Testes de Validação

Na fase de desenvolvimento da versão 1.0 do sistema, os testes são conduzidos de maneira contínua e paralela à implementação das funcionalidades. O uso do kit da BitDogLab proporciona maior agilidade nesse processo, visto que grande parte dos componentes essenciais já está integrada à placa. Dessa forma, a validação inicial se concentra nos módulos de hardware embarcados, restando apenas os sensores de saturação, temperatura e batimentos cardíacos para testes posteriores. A abordagem adotada permite identificar e corrigir rapidamente falhas no código e na configuração dos periféricos, garantindo um sistema estável e confiável antes da implementação final.

A validação do equipamento finalizado deverá ser feita como segue:

- **Teste de sensores:** Comparação dos valores medidos com equipamentos médicos;
- **Teste de Comunicação:** Verificação da integridade dos pacotes de dados transmitidos via UART;
- **Teste de Interface Gráfica:** Avaliação da exibição correta das informações no display SSD1306;
- **Teste de Controle de Leito:** Validação do funcionamento dos motores acionados por PWM;
- **Teste de Consumo Energético:** Medição do consumo da alimentação USB e da bateria reserva.

## c. Discussão dos Resultados

Os testes realizados até o momento demonstram a viabilidade do desenvolvimento do projeto e já se percebe a necessidade urgente de um dispositivo de memória não volátil para salvar as calibrações realizadas e a necessidade de aquisição dos sensores para verificar a integração destes com o sistema.

A execução do projeto representa um desafio significativo, exigindo uma abordagem estruturada e metódica para garantir a correta integração entre hardware e software. Apesar das complexidades inerentes ao desenvolvimento de sistemas embarcados, o conhecimento adquirido ao longo do curso tem sido fundamental para superar as dificuldades técnicas. O uso de plataformas de desenvolvimento bem documentadas, como o RP2040 e a BitDogLab, tem facilitado a implementação e permitido uma evolução consistente do projeto. A medida que os testes avançam e os módulos adicionais são integrados, espera-se que o sistema atinja um nível de maturidade adequado para futuras expansões, incluindo conectividade sem fio e aprimoramentos na interface de usuário.

## 5. Referências

ANALOG DEVICES. *AD8232 – Single Lead Heart Rate Monitor Front End*. Disponível em: <https://www.analog.com/en/products/ad8232.html>. Acesso em: 19 de fev. de 2025.

ANALOG DEVICES. *MAX30102 – Pulse Oximeter and Heart-Rate Sensor*. Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/max30102.pdf>. Acesso em: 19 de fev. de 2025.

FOTOPLETISMOGRAFIA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Fotopletismografia&oldid=66872504>>. Acesso em: 20 fev. 2025.

G1 São Carlos e Araraquara. Pesquisadores da USP criam leito inteligente que monitora paciente à distância. São Carlos, 19 jul. 2018. Disponível em: <https://g1.globo.com/sp/são-carlos-regiao/noticia/2018/07/19/pesquisadores-da-usp-criam-leito-inteligente-que-monitora-paciente-a-distancia.ghtml>. Acesso em: 18 fev. 2025.

MICROCHIP. *MCP9808 – High-Accuracy Digital Temperature Sensor*. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf>. Acesso em: 19 de fev. de 2025.

RIUSON. *LCD Image Converter*. Disponível em: <https://lcd-image-converter.riuson.com/en/about/>. Acesso em: 24 de fev. de 2025.

SECRETARIA DE COMUNICAÇÃO SOCIAL. Censo 2022: número de idosos na população do país cresceu 57,4% em 12 anos. Brasília, 10 out. 2023. Disponível em: <https://www.gov.br/secom/pt-br/assuntos/noticias/2023/10/cento-2022-numero-de-idosos-na-populacao-do-pais-cresceu-57-4-em-12-anos>>. Acesso em: 18 fev. 2025.

SOLOMON SYSTECH. *SSD1306 – 128 x 64 Dot Matrix OLED/PLED Segment Driver with Controller*. Disponível em: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>. Acesso em: 19 de fev. de 2025.

WORLDSEMI. *WS2812 – Intelligent control LED integrated light source*. Disponível em: <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>. Acesso em: 19 de fev. de 2025.

RASPBERRY PI. *RP2040 Datasheet*. Disponível em: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>. Acesso em: 19 de fev. de 2025. CYW. *CYW43439 – Wi-Fi and*

INFINEON TECHNOLOGIES. CYW43439 – Wi-Fi and Bluetooth Combo Chip. Disponível em: [https://www.infineon.com/dgdl/Infineon-CYW4334WKUBGT-DataSheet-v11\\_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee1f02f680f](https://www.infineon.com/dgdl/Infineon-CYW4334WKUBGT-DataSheet-v11_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee1f02f680f). Acesso em: 19 de fev. de 2025.