



Sistema de Monitoramento de Enchentes em Rios

Emyle Santana da Silva - TIC370101064

Embarcotech

Polo Ilhéus/Itabuna

Sumário

Escopo do Projeto.....	3
Apresentação.....	3
Objetivos.....	3
Descrição do Funcionamento.....	3
Justificativa.....	4
Projetos Correlatos.....	4
Especificações do Hardware.....	5
Diagrama em Blocos.....	5
Microcontrolador Raspberry Pi Pico W.....	6
Joystick (Eixo Y).....	6
Função.....	6
Configuração.....	6
Comandos e Registros Utilizados.....	6
Pinagem.....	6
Joystick (Eixo X).....	6
Função.....	6
Configuração.....	6
Comandos e Registros Utilizados.....	7
Pinagem.....	7
Botão A.....	7
Função.....	7
Configuração.....	7
Comandos e Registros Utilizados.....	7
Pinagem.....	7
Display SSD1306.....	7
Função.....	7
Configuração.....	8
Comandos e Registros Utilizados.....	8
Pinagem.....	8
USB (UART).....	8
Função.....	8
Configuração.....	8
Comandos e Registros Utilizados.....	9
Pinagem.....	9
Software Control.....	9
Circuito do Hardware.....	10
Especificações do Firmware.....	11
Blocos Funcionais - Diagrama de Camadas.....	11

Descrição das Funcionalidades.....	11
Comunicação.....	11
Processamento.....	11
Interface.....	12
Descrição das Principais Variáveis e Constantes.....	12
Macros.....	12
Variáveis.....	12
Fluxograma.....	13
Inicialização.....	14
Configurações dos Registros.....	14
Estrutura e formato dos dados.....	15
Protocolo de Comunicação.....	15
Formato do Pacote de Dados.....	16
Execução do Projeto.....	16
Metodologia.....	16
Testes de Validação.....	16
Discussão dos Resultados.....	17
Repositório no GitHub e Vídeo Demonstrativo.....	17
Referências.....	18

Escopo do Projeto

Apresentação

Este projeto consiste na implementação de um Sistema Embarcado que monitora o nível de rios, com foco no Rio Cachoeira (Ilhéus, Itabuna), para controle e alerta de possíveis enchentes e/ou inundações em locais próximos ao Rio.

Objetivos

1. Criar um sistema capaz de monitorar e medir o nível de um Rio;
2. Enviar mensagens de alerta para a população e órgãos responsáveis sempre que houver algum risco de enchente e/ou inundações;
3. Gerar relatórios quanto ao nível do rio diariamente.

Descrição do Funcionamento

O sistema utiliza um sensor ultrassônico HC-SR04 (representado pelo eixo Y do joystick na BitDogLab) para medir o nível do rio. Além disso, um sensor de chuva YL-83 (representado pelo eixo X do joystick) é empregado para verificar a existência de chuva e determinar sua intensidade. O sensor ultrassônico fornece o valor da distância entre ele e a superfície da água, permitindo verificar se houve aumento ou diminuição do nível do rio; enquanto isso, o YL-83 retorna um valor entre 0 e 1023, no qual 0 indica ausência de chuva e 1023 representa chuva intensa.

Após obter a distância medida pelo HC-SR04, o sistema realiza verificações para determinar se há risco de inundação. Caso seja identificado perigo, um alerta é enviado aos órgãos responsáveis, bem como aos moradores, a fim de que possam tomar as medidas de proteção necessárias. Os níveis de alerta variam de acordo com o nível do rio e a combinação de presença ou não de chuva.

Para a análise dos dados recebidos dos sensores, é utilizado o Raspberry Pi Pico W. Além do envio de alertas, o sistema também é capaz de enviar relatórios periódicos sobre o monitoramento; esses relatórios podem ainda ser gerados e enviados mediante solicitação do usuário (neste caso, os órgãos responsáveis). O envio dos relatórios é realizado via protocolo Wi-Fi.

No projeto, para representar o envio de mensagens, utiliza-se o display SSD1306 conectado via I2C; para o envio de relatórios, emprega-se a comunicação UART; e, para representar o recebimento de solicitações, é utilizado o botão A, acionado por meio de

interrupção. Além disso, os cálculos de nível de rio e intensidade da chuva foram adaptados para as entradas fornecidas pelo Joystick.

Justificativa

O Rio Cachoeira é um importante rio do estado da Bahia e passa por diversas cidades, entre elas Ilhéus e Itabuna. Em períodos em que ocorrem fortes chuvas, existe o perigo de inundações nas cidades de Itabuna e Ilhéus. Na cidade de Ilhéus, bairros como Banco da Vitória, Vila Cachoeira e Salobrinho (entre outros), por estarem próximos ao rio, encontram-se em grande risco em relação às enchentes provocadas pelas chuvas, assim como as áreas de Itabuna mais próximas ao rio.

Um exemplo desses acontecimentos foi o desastre ocorrido em dezembro de 2021, quando, de acordo com a prefeitura de Itabuna, 40% da zona urbana foi devastada, 7.855 residências foram atingidas, houve 16 pontos de inundação e 1.777 residências danificadas.

Dessa maneira, a existência de um sistema de monitoramento para o rio Cachoeira (assim como para outros rios) é essencial para ajudar a manter a segurança das pessoas que vivem em áreas próximas aos cursos d'água, suscetíveis a inundações. Além disso, órgãos responsáveis podem receber relatórios periódicos, de modo que soluções prévias possam ser tomadas, buscando assim diminuir os riscos à população e os prejuízos causados pelas inundações.

Projetos Correlatos

1. Monitoramento de nível de rio utilizando microcomputador, sensor ultrassônico e comunicação celular.

Autores: Marco Antonio S. Teixeira, Raul V. Rosa, Alex L. de Sousa e Mário E. Augusto.

No projeto, os autores apresentam um sistema de monitoramento de rio composto por uma estação utilizando o microcomputador Raspberry Pi, sensor ultrassônico, sensor de temperatura, Interface 3G, comunicação celular e uso de um servidor de informações.

2. Sistema de Monitoramento Remoto de Transbordo do Rio Sesmaria.

Autores: Renan de oliveira alves takeuchi, Franciole José, Eduardo Souza Motta e Juliana Gonçalves Fernandes.

O sistema de monitoramento apresentado no trabalho utiliza a plataforma Arduino Mega e alertas via uma rede de telefonia móvel GSM.

O projeto apresentado, envia alertas e também é capaz de receber uma solicitação de informação sobre o nível atual do rio, tendo como resposta o envio de um SMS com o nível apresentado no momento.

O diferencial do sistema de monitoramento de enchentes apresentado no presente trabalho, está na possibilidade de envios de relatórios sobre o nível do rio de forma periódica ou quando solicitada por órgão responsáveis, além disso, o sistema também é capaz de verificar se existe precipitação, de modo que os alertas podem ser mais eficientes quando os dois fatores (nível do rio e intensidade de chuva) são combinados na análise de risco.

O relatório inclui: Nível atual, nível apresentado no relatório passado, intensidade de chuva, e diferença percentual no nível atual com o nível do último relatório enviado, tendo como possibilidade a ampliação das informações de acordo com as demandas do local.

Especificações do Hardware

Diagrama em Blocos

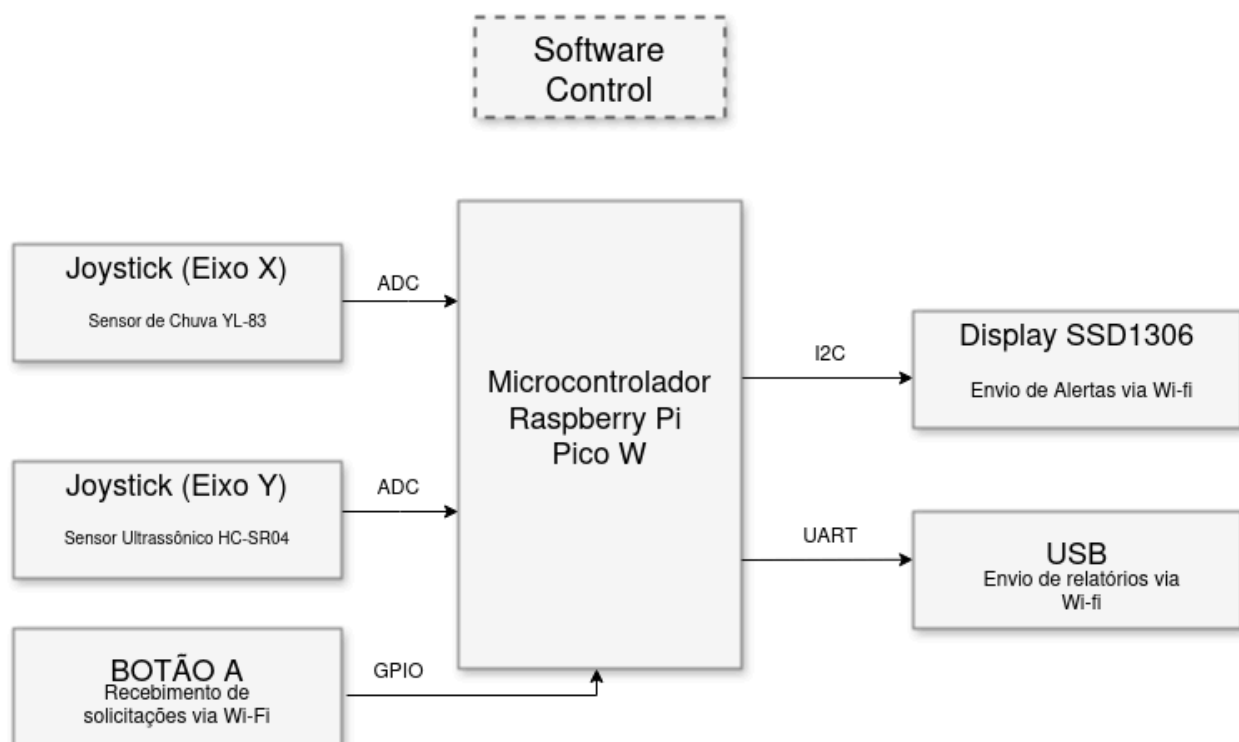




Imagem 1 - Diagrama em Blocos Hardware (Fonte: Autoral)

Microcontrolador Raspberry Pi Pico W

Responsável por ler os valores dos sensores, realizar os devidos monitoramentos, processar dados e enviar/receber informações para os outros blocos.

Joystick (Eixo Y)

Função

Simula um sensor ultrassônico HC-SR04. Responsável por medir o nível atual do rio (como se fosse a distância medida pelo sensor ultrassônico) .

Configuração

Configurado usando o ADC do Raspberry Pi Pico. Inicializado com `adc_init()`. A entrada analógica é configurada com `adc_gpio_init()` e tem o canal selecionado com `adc_select_input(1)`.

Comandos e Registros Utilizados

`adc_init()` - Habilitar o ADC

`adc_gpio_init(YL_83)` - Configura o pino usado como entrada analógica

`adc_select_input(1)` - Seleciona o canal antes da leitura

Pinagem

Pino Físico: GPIO 27

Usado como entrada analógica no canal ADC 1.

Joystick (Eixo X)

Função

Simula um sensor de chuva YL-83. Responsável por verificar se há chuva e informar a intensidade atual da precipitação.

Configuração

Configurado usando o ADC do Raspberry Pi Pico. Assim como o eixo X, é inicializado com `adc_init()`. A entrada analógica é configurada com `adc_gpio_init()` e tem o canal selecionado com `adc_select_input(0)`.

Comandos e Registros Utilizados

`adc_init()` - Habilitar o ADC

`adc_gpio_init(HC_SR04)` - Configura o pino usado como entrada analógica

`adc_select_input(0)` - Seleciona o canal

Pinagem

Pino físico: GPIO 26

Usado como entrada analógica no canal ADC 0.

Botão A

Função

Simula a solicitação do usuário por um relatório. Quando pressionado, gera uma interrupção (IRQ) que dispara a função de envio de relatório.

Configuração

Inicializado com `gpio_init(BUTTON_A)`. Depois, a direção é definida como entrada, em seguida o pull up interno é ativado e a interrupção é configurada para detectar a borda de descida.

Comandos e Registros Utilizados

`gpio_init(BUTTON_A)` - Inicializa o pino

`gpio_set_dir(BUTTON_A, GPIO_IN)` - Configura como entrada

`gpio_pull_up(BUTTON_A)` - Ativa o pull up do botão

`gpio_set_irq_enabled_with_callback(BUTTON_A, GPIO_IRQ_EDGE_FALL, true, &gpio_irq_handler)` - Configurar interrupções e a função de callback

Pinagem

Pino Físico: GPIO 5

Entrada digital com pull-up para detecção de interrupção (solicitação de relatório).

Display SSD1306

Função

Simula o receptor de dados (como celulares) dos usuários. Os dados seriam recebidos via Wi-fi. Exibe os alertas e mensagens sobre o nível do rio.

Na BitDogLab, a comunicação ocorre via interface I2C.

Configuração

I2C: Inicializado como porta I2C para 400 kHz; também foram configurados dois pinos como função I2C (SDA e SCL).

Display SSD1306: Inicializado com 128x64 pixels, endereço 0x3C e comunicação via I2C.

Os dados acima são enviados para configuração inicial e depois o display é limpo para que todos os pixels iniciem apagados.

Comandos e Registros Utilizados

i2c_init(I2C_PORT, 400*1000) - Inicializa o I2C

gpio_set_function(I2C_SDA, GPIO_FUNC_I2C) - Configura o pino SDA do display com I2C

gpio_set_function(I2C_SCL, GPIO_FUNC_I2C) - Configura o pino SCL do display com I2C

gpio_pull_up(I2C_SDA) - Aciona o pull up do pino SDA

gpio_pull_up(I2C_SCL) - Aciona o pull up do pino SCL

ssd1306_init(&ssd, WIDTH, HEIGHT, false, address, I2C_PORT) - Inicializa o display

ssd1306_config(&ssd) - Configura o display

ssd1306_send_data(&ssd) - Envia os dados para o display

ssd1306_fill(&ssd, false) - Limpa o display para que todos os pixels iniciem apagados

Pinagem

I2C_SDA: GPIO 14 - função I2C SDA

I2C_SCL: GPIO 15 - função I2C SCL

Endereço do display: 0x3C


Servem como saída de dados, onde as mensagens de alertas geradas pelo sistema são exibidas.

USB (UART)

Função

Simula o envio de dados (relatórios) para o usuário via wi-fi. No projeto, os dados são enviados via USB serial. A UART é configurada para mandar mensagens com o "printf".

Configuração



Utiliza comunicação UART. Inicializado com a uart 0 e baud rate de 115200, ademais, também é feita a inicialização do stdio para saídas de printf no USB serial.

Comandos e Registros Utilizados

`stdio_init_all()`

`uart_init(uart0, BAUD_RATE)` - Inicializa o UART.

Pinagem

UART0

Software Control

Camada lógica (firmware) que gerencia todo o sistema. É executado no microcontrolador.

Circuito do Hardware

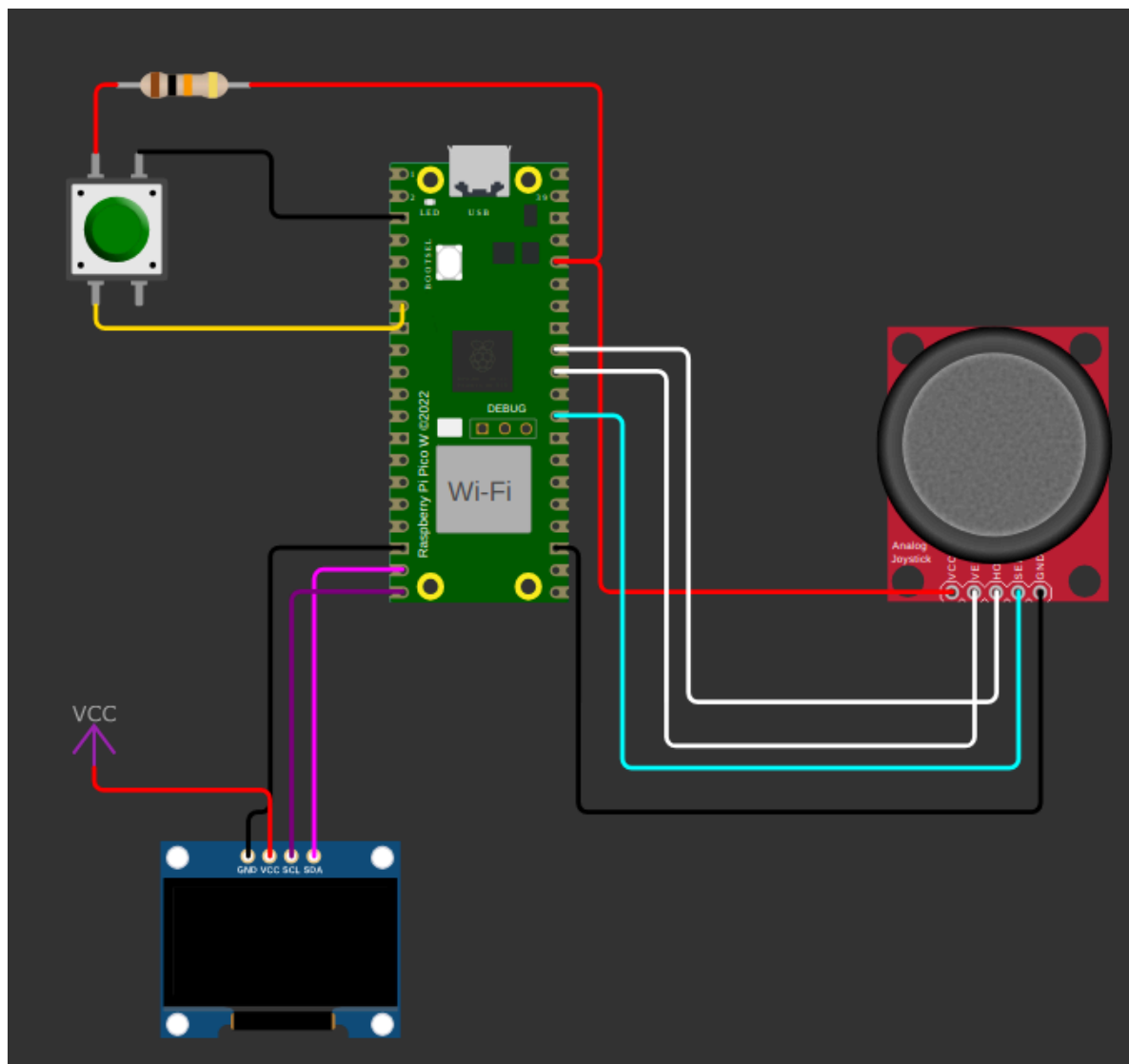


Imagem 2 - Desenho do Circuito do Hardware (Fonte: Autoral)

Especificações do Firmware

Blocos Funcionais - Diagrama de Camadas

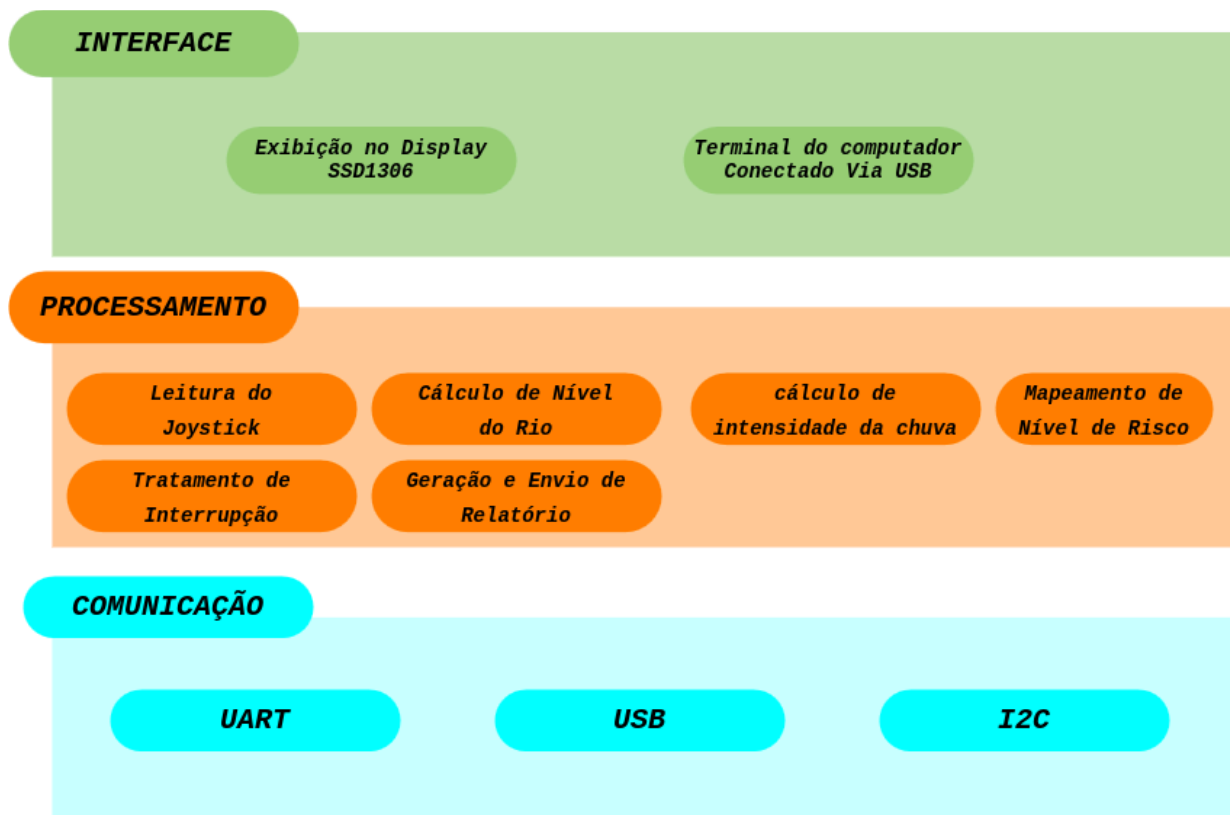


Imagem 3 - Diagrama de Blocos Funcionais (Fonte: Autoral)

Descrição das Funcionalidades

Comunicação

- UART e USB: Permite que haja comunicação entre o microcontrolador e um computador (quando a placa está conectada via USB). Os relatórios são exibidos no computador ao utilizar a UART para direcionar comandos (printf) para o USB conectado. A comunicação simula uma comunicação entre órgãos responsáveis e o sistema.
- I2C: Permite a comunicação entre o microcontrolador e o display SSD1306.

Processamento

- Faz a leitura dos dados capturados do joystick (com uso do ADC) e calcula o nível do rio (simulando o cálculo da distância lido com o sensor ultrassônico) e a intensidade da chuva (simulando o sensor de chuva). Depois o mapeamento do nível de risco é feito com os dois valores obtidos e a notificação correspondente é enviada. Além disso, também há o tratamento de interrupção quando o botão é pressionado e sempre que se passa um período especificado para envios de relatórios, em ambos os casos, um relatório é gerado e enviado para o usuário (exibido no terminal do computador ao ser enviado através do UART/USB).

Interface

Os status de risco e nível do rio são exibidos no Display SSD1306 enquanto as informações do relatório são exibidas no terminal do computador conectado com o Raspberry pi pico w.

Descrição das Principais Variáveis e Constantes

Macros

YL_83: Pino ligado ao eixo X do Joystick

HC_SR04: Pino ligado ao eixo Y do Joystick

BUTTON_A: Pino ligado ao botão A

REPORT_TIME: Define o intervalo de tempo (em ms) entre o envio dos relatórios.

Variáveis

adc_x_value e **adc_y_value:** Guardam os valores lidos do eixo X e Y do Joystick respectivamente.

current_time e **last_debounce_time:** Guardam os valores do tempo passado (o último desde o último Debounce e o atual lido, respectivamente) para realizar o tratamento de bouncing do Botão A.

river_level: Define o nível normal do rio, usado para fazer o cálculo de nível atual.

intense_rain: Define o valor máximo de intensidade de chuva para normalização (100%).

current_river_level: O novo valor do nível do rio após leitura do sensor (joystick eixo Y).

last_river_level: O valor do nível do rio enviado no último relatório.

current_rain_intensity: A intensidade atual da chuva após leitura do sensor (joystick eixo X).

level_status: Enum com os possíveis estados de alertas a serem notificados.

status: Armazena o estado atual de alerta.

Fluxograma

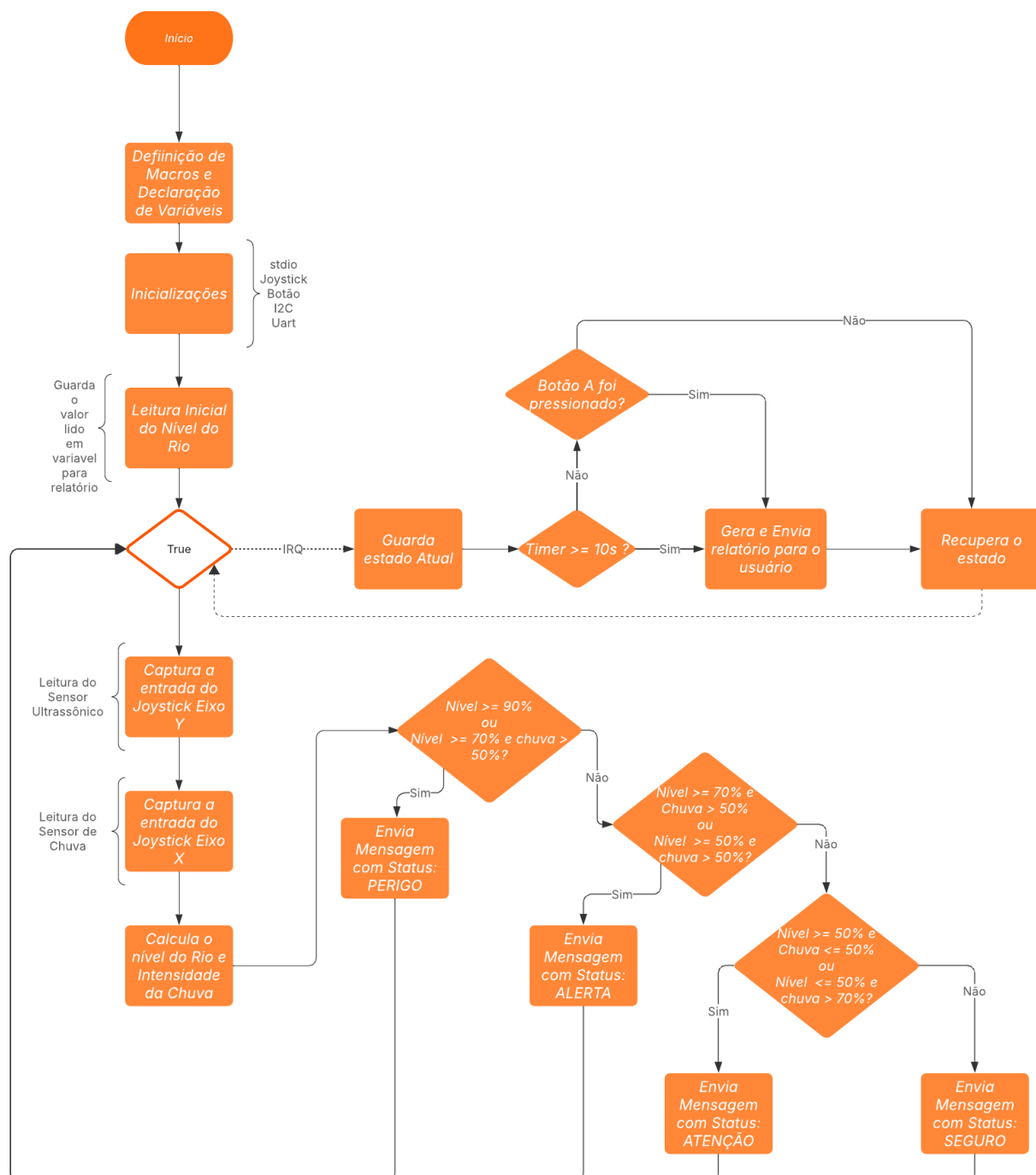


Imagem 4 - Fluxograma do Firmware (Fonte: Autoral)

Inicialização

O sistema prepara o ADC, GPIO, I2C, UART, display e Timers para iniciar o monitoramento dos sensores (joysticks) e a comunicação para envio de relatórios.

Para isso são usadas as funções:

`stdio_init_all()`: Inicializa a saída padrão (`printf`) via USB/Serial permitindo que o microcontrolador envie relatórios para o computador.

`init_joystick()`: Configura o ADC e os pinos conectados aos eixos X e Y do Joystick (que simulam os sensores HC-SR04 e YL-83).

`init_button()`: Configura o botão A como entrada digital com o pull up interno ativado.

`init_i2c_display()`: Inicializa a interface de comunicação I2C (pinos, clock de 400 kHz) e configura o display SSD1306.

`uart_init(uart0, BAUD_RATE)`: Configura a UART para 115200 bps para ser usada como saída serial.

`gpio_set_irq_enabled_with_callback(...)`: Registra a função de callback `gpio_irq_handler()` que será chamada quando houver uma interrupção gerada pelo pressionamento do botão.

`add_repeating_timer_ms(...)`: A cada **REPORT_TIME** (ms) a função `repeating_timer_callback()` é chamada, enviando automaticamente um relatório.

Por fim, antes do loop principal a primeira leitura do nível do rio é feita e a variável **last_river_level** é atualizada para que os cálculos do primeiro relatório sejam feitos corretamente.

Configurações dos Registros

Realizada pelas funções da Pico SDK (majoritariamente) para configurar o ADC, I2C, GPIO, UART e timers. O display recebe os comandos via I2C.

Principais chamadas no código:

ADC para o Joystick

`adc_init()`

`adc_gpio_init(...)`

`adc_select_input()`

`adc_read()`

GPIO para o Botão A

`gpio_init(...)`



```
gpio_set_dir(...)
```

```
gpio_pull_up(...)
```

```
gpio_set_irq_enabled_with_callback(...)
```

I2C para o Display SSD1306

```
i2c_init(...)
```

```
gpio_set_function(...)
```

ssd1306_init(...) - Envia os comandos via I2C internamente para configurar os registradores do SSD1306

UART

```
uart_init(...)
```

Timer

```
add_repeating_timer_ms(...)
```

Estrutura e formato dos dados

Os dados são armazenados em variáveis globais¹ dos tipos: uint32_t, float e int.

Além disso, também há uma enumeração de status, onde cada status define um nível diferente de risco, como enumerado abaixo:

ATTENTION: Pouco Risco

ALERT: Risco de enchente

DANGER: Risco de Inundação

SAFE: Sem Risco

Protocolo de Comunicação

O sistema utiliza dois meios de comunicação:

USB/Serial (UART)

Utiliza o printf para enviar relatórios para o computador no formato textual. Utiliza protocolo simples de texto.

I2C

Utilizado na comunicação com o display SSD1306. Os comandos são enviados por meio das funções disponíveis na biblioteca ssd1306, sem uso de protocolos customizados.

¹ Todas as principais variáveis foram apresentadas no tópico “Descrição das Principais Variáveis e Constantes”

Formato do Pacote de Dados

Pacote Serial: Conjunto de mensagens de texto separados por quebras de linha (relatórios) impressos no terminal.

Cada pacote tem o formato:

ID 1

Nível: 5.00

Chuva: 49.99%

Status: SEGURO

Dif:0.00%

Pacote I2C: Bytes de comandos e dados enviados para atualizar o display. Implementados pelas funções encontradas na biblioteca ssd1306.

Execução do Projeto

Metodologia

Primeiro, foram realizadas pesquisas para encontrar projetos correlatos. Depois, procurei por sensores para encontrar os que melhor se encaixavam ao projeto e entender o funcionamento dos mesmos, dessa maneira, a escolha para os componentes do ambiente de desenvolvimento BitDogLab que simularia os sensores poderia ser feita de maneira adequada.

As funcionalidades do software foram definidas após pesquisas sobre os projetos correlatos e pesquisas sobre ocorrências de enchentes e inundações na região para garantir que os principais pontos do problema fossem abordados pelo sistema.

Para realização da implementação do projeto foi utilizada a IDE vsCode juntamente com o uso da extensão da Raspberry Pi Pico e WokWi. A depuração foi feita através do depurador da IDE.

Testes de Validação

Os testes foram realizados de duas maneiras.

Primeiro, a simulação foi feita através do wokwi para garantir que o código funcionava da maneira esperada, sempre que uma funcionalidade era adicionada ou atualizada os testes no simulador eram refeitos para garantir que o firmware ainda cumpria o que devia.

Quando a implementação foi completamente finalizada e testada, o firmware foi gravado no microcontrolador para testes mais precisos.

Discussão dos Resultados

As enchentes e inundações provocadas pelas fortes chuvas, somadas ao aumento dos níveis dos rios na região, vêm sendo um problema recorrente. O sistema desenvolvido busca contribuir para a segurança e proteção das comunidades que vivem próximas aos rios e enfrentam esse risco constantemente.

A simulação realizada com os componentes disponíveis permitiu não apenas avaliar o funcionamento básico do sistema de monitoramento (leituras, processamento de dados e envio de relatórios), mas também possibilitou o aprendizado prático sobre o desenvolvimento de sistemas embarcados. Embora se trate de uma simulação, a arquitetura e a lógica de funcionamento podem ser adaptadas para aplicações reais, contribuindo para a redução de riscos e prevenção de desastres.

Os resultados mostram que o sistema é capaz de ler dados (simulados pelos joysticks), processá-los e fornecer relatórios para os usuários de maneira eficiente no contexto do protótipo. Entretanto, há espaço para aprimoramentos, o que tornaria a solução ainda mais robusta em um cenário real. Dessa forma, o projeto demonstra confiabilidade em sua proposta inicial, mas ressalta a possibilidade de evolução para um sistema de monitoramento plenamente operacional, oferecendo maior proteção e tranquilidade às comunidades afetadas.

Repositório no GitHub e Vídeo Demonstrativo

O repositório com a implementação do projeto pode ser acessado através do link: https://github.com/EmyleSilva/Embarcotech-U7-Projeto_Final.git

O vídeo do projeto está disponível no Youtube e pode ser acessado através do link: https://youtu.be/TJ7RmlqV_RI?si=Kxw_2LpnzQWSc2Uc

Referências

ANTONIO, M. et al. **Monitoramento de nível de rio utilizando microcomputador, sensor ultrassônico e comunicação celular**. [s.l: s.n.]. Disponível em:

<https://www.udesc.br/arquivos/ceplan/id_cpmenu/1593/6720170906VF_16681135209965_1593.pdf>. Acesso em: 20 fev. 2025.

CITISYSTEMS. **Sensor Ultrassônico Industrial - Como Funciona?** Disponível em:

<<https://www.youtube.com/watch?v=UlvuZKlSVmM>>. Acesso em: 22 fev. 2025.

ELETROGATE. **Sensor Ultrassônico HC-SR04 com Arduino - Entenda e aprenda a usar**.

Disponível em: <<https://blog.eletrogate.com/sensor-ultrassonico-hc-sr04-com-arduino/>>. Acesso em: 22 fev. 2025.

EZEQUIEL, F. et al. **SISTEMA DE MONITORAMENTO REMOTO DE TRANSBORDO DO RIO SESMARIA**. [s.l: s.n.]. Disponível em:

<<https://www.aedb.br/seget/arquivos/artigos17/20025202.pdf>>. Acesso em: 20 fev. 2025.

JULIALABS. **Sensor de Distância Ultrassônico - COMO USAR NO ARDUINO**. Disponível

em: <https://youtu.be/J_bUAs-VXA8?si=uEGPaO9qbIq1pfga>. Acesso em: 22 fev. 2025.

Prefeitura de Itabuna prossegue com a execução de ações mitigadoras das enchentes de 2021 - Prefeitura de Itabuna. Disponível em:

<<https://itabuna.ba.gov.br/2024/01/02/prefeitura-de-itabuna-prossegue-com-a-execucao-de-aco-es-mitigadoras-das-enchentes-de-2021/>>. Acesso em: 23 fev. 2025.

THOMSEN, A. **Sensor de chuva YL-83 - MakerHero**. Disponível em:

<https://www.makerhero.com/blog/sensor-de-chuva-yl-83/?srsltid=AfmBOoo-dhyLr-VjoZ_kdgi4DOhGPAkEopazR5n20Q6u4U7-racDePgQ>. Acesso em: 22 fev. 2025.