

# Projeto de Subsistema de Software

[Clique Aqui](#) para baixar o documento.

Este documento descreve o projeto de um subsistema de software, destacando suas principais características, componentes e interações. O objetivo é fornecer uma visão geral das escolhas técnicas e de design que guiam o desenvolvimento do subsistema. Abaixo, a tabela resume os artefatos-chave do projeto e seus detalhes, com links que poderão ser atualizados para um acesso direto aos conteúdos completos.

**Tabela 1:** Descrição dos artefatos principais do subsistema e suas funções.

Artefato	Descrição
<a href="#">Arquitetura de Software</a>	Documento que descreve a estrutura do subsistema, seus componentes, as metas e restrições arquiteturais, além dos diagramas de arquitetura.
<a href="#">Backlog do Produto</a>	Lista priorizada de histórias de usuário, roadmap, kanban e tabelas que detalham os itens planejados para o produto, incluindo a gestão de versões.
<a href="#">Documento de Identidade Visual</a>	Descrição da identidade visual do produto, abordando a proposta de valor, logo, cores, tipografia e outros elementos gráficos.
<a href="#">Fluxos de Usuário</a>	Descrição dos diferentes fluxos de usuário, incluindo fluxogramas de fluxos felizes e de exceção, detalhando as interações com o sistema.

Fonte: Autoria Própria. Todos os direitos reservados.

## Comunicação com a Pixhawk e Raspberry

A comunicação entre o **Pixhawk** e o **Raspberry Pi**, os principais núcleos de controle do projeto, é fundamental para o funcionamento adequado do voo do drone. Ela permite que o Raspberry Pi processe dados de sensores e integre sistemas essenciais para a operação do drone, como o assistente de mira. Esta seção detalha os métodos e protocolos utilizados para essa comunicação, abordando a troca de dados e comandos entre o software e o hardware, com ênfase em aspectos como desempenho, segurança e confiabilidade.

### Métodos de Comunicação

A interação entre o Pixhawk e o Raspberry Pi ocorre por meio de interfaces de comunicação, permitindo que o Raspberry Pi atue como um hub de integração e controle adicional no sistema. Os principais métodos de comunicação incluem **MAVLink** e **Comunicação Serial (UART)**, conforme detalhado na **Tabela 2**.

**Tabela 2:** Métodos de comunicação entre o Pixhawk e o Raspberry Pi.

Método de Comunicação	Descrição	Detalhamento
<a href="#">MAVLink (Micro Air Vehicle Link)</a>	Protocolo eficiente e leve para comunicação entre o Pixhawk e o Raspberry Pi. Permite o envio de comandos e recepção de dados essenciais, como status de sensores, posição do drone, altitude, entre outros.	O MAVLink permite a troca bidirecional de dados, com baixa latência e alta eficiência. O Raspberry Pi pode enviar comandos ao Pixhawk, como ajustes de parâmetros de voo, e o Pixhawk envia dados sobre a navegação e status do drone. Projetado para sistemas embarcados, o MAVLink utiliza uma estrutura compacta de pacotes de dados para reduzir a latência.
<a href="#">Comunicação Serial (UART/USART)</a>	Conexão direta através de uma porta serial (TX/RX) entre o Raspberry Pi e o Pixhawk. Permite a troca contínua de dados e comandos.	A comunicação serial via UART é simples de implementar e altamente confiável. A comunicação é síncrona, permitindo a troca contínua de dados em tempo real, o que é crucial para o controle preciso do voo. O Pixhawk envia dados sobre o estado dos sensores e a posição, enquanto o Raspberry Pi pode enviar comandos de controle e ler sensores conectados diretamente ao Raspberry Pi.

Fonte: Autoria Própria. Todos os direitos reservados.

## Desempenho e Eficiência

A comunicação entre o Pixhawk e o Raspberry Pi deve garantir um desempenho rápido e eficiente, minimizando falhas no sistema de voo. O uso de **MAVLink** e **UART** assegura baixa latência e troca eficiente de dados cruciais, como posição do drone e estado dos sensores, conforme ilustrado na **Tabela 3**.

**Tabela 3:** Aspectos de desempenho na comunicação entre o Pixhawk e o Raspberry Pi.

Aspecto	Importância
Desempenho Rápido	Minimiza o tempo de resposta entre o envio de comandos e a resposta do sistema.
Eficiência de Comunicação	Garante a troca de dados críticos durante o voo, com alta velocidade e confiabilidade.

Fonte: Autoria Própria. Todos os direitos reservados.

## Segurança e Confiabilidade

A segurança e confiabilidade da comunicação entre o Pixhawk e o Raspberry Pi são essenciais, especialmente em sistemas autônomos, onde falhas podem comprometer a missão. Para garantir a segurança, práticas como o uso de **checksums** são empregadas para validar a integridade dos dados transmitidos, conforme ilustrado na **Tabela 4**.

**Tabela 4:** Técnicas de segurança e verificação de dados na comunicação Pixhawk-Raspberry Pi.

Técnica de Segurança	Descrição
Checksums	O MAVLink utiliza checksums para verificar a integridade dos dados transmitidos, garantindo que os pacotes não sejam corrompidos durante a comunicação. Essa técnica é fundamental, já que falhas de comunicação podem comprometer a segurança do voo.
Verificação de Integridade	Assegura que os dados recebidos são válidos e não foram alterados ou corrompidos, o que é essencial em sistemas críticos como os de controle de voo.

Fonte: Autoria Própria. Todos os direitos reservados.

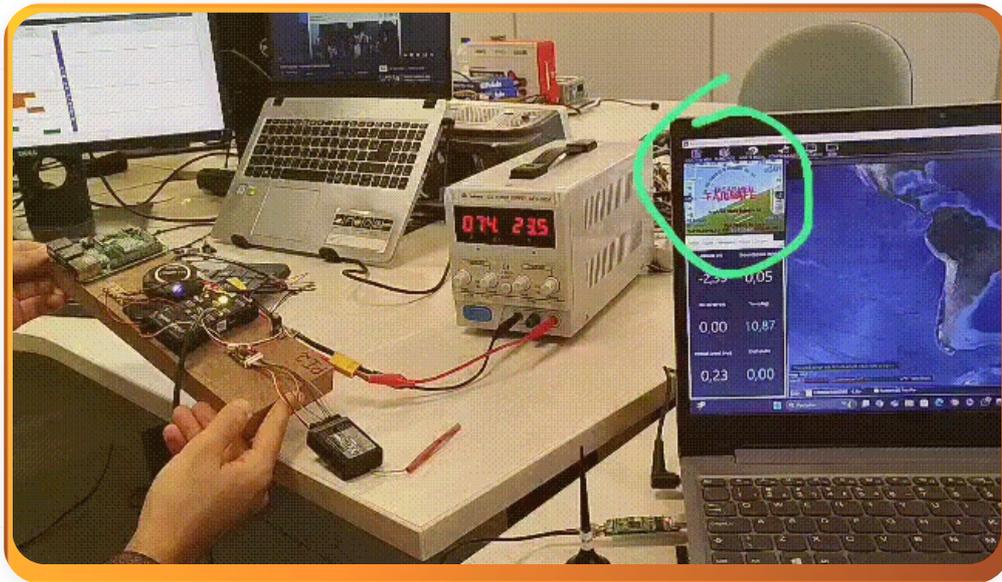
## Software Embarcado

Esta seção apresenta o software embarcado que opera na raspberry, incluindo funcionalidades, estrutura de código e integração com outros componentes do sistema. Ele é projetado para garantir precisão e robustez nas operações realizadas pelo hardware.

### Simulação do Pêndulo

Essa funcionalidade tem como objetivo controlar e visualizar uma simulação de um pêndulo em uma interface gráfica na raspberry, além de monitorar o uso da CPU. Ele utiliza a câmera para capturar imagens e sobrepõe gráficos na tela, criando uma visualização interativa do sistema conforme a **Figura 1** abaixo:

Figura 1: Exemplo do funcionamento do pêndulo.



Fonte: Autoria própria. Todos os direitos reservados.

#### ESTRUTURA DO CÓDIGO

O software segue uma estrutura modular, onde cada funcionalidade principal é implementada como uma função:

- **Monitoramento do Hardware:**

Funções como `get_cpu_temperature` e `get_cpu_usage` obtêm dados do sistema operacional, garantindo um controle contínuo dos recursos da Raspberry Pi.

- **Desenho Gráfico:**

- `draw_crosshair`: Exibe um crosshair central na tela.
- `draw_cpu_info`: Mostra temperatura e uso da CPU no canto superior direito.
- `draw_moc3_circles`: Indica o estado da variável `moc3` com círculos na parte superior esquerda.
- `draw_pendulum`: Simula o movimento de um pêndulo na tela, baseado no ângulo fornecido.

- **Captura e Renderização de Frames:**

A integração com a câmera é feita por meio da biblioteca `Picamera2`, com configuração personalizada para preview. Cada frame capturado é processado para incluir as sobreposições gráficas antes de ser exibido na tela.

#### INTEGRAÇÃO COM OUTROS COMPONENTES

Este software embarcado é projetado para interagir diretamente com o hardware da Raspberry Pi e a câmera. Além disso, pode ser estendido para integração com sensores adicionais, caso necessário.

## Referências

1. **MATHWORKS**. *CAN Communication with Pixhawk Using Raspberry Pi*. Acesso em: 15 nov. 2024.
2. **ARDUPILOT**. *Communicating with Raspberry Pi via MAVLink*. Acesso em: 15 nov. 2024.

## Histórico de Versões

Versão	Data	Descrição	Autor(es)
1.0	03/11/2024	Criação e estruturação inicial do documento	Gustavo Martins
1.1	15/11/2024	Desenvolvimento do tópico <b>Arquitetura de Software</b>	Felipe Freire
1.2	15/11/2024	Desenvolvimento do tópico <b>Comunicação com a Pixhawk e Raspberry</b>	Felipe Freire
1.3	21/11/2024	Revisão e correção geral e evolução do artefato	Gustavo Martins Ribeiro

1.4

22/11/2024

Desenvolvimento do tópico **Software Embarcado**

Felipe Freire