



Manual de Usuário

Splinters

Gerdau

Histórico de Versões

Data da modificação	Responsável	Descrição
21/06/2023	Luana	Correção dos itens 2.1 e 2.2

Índice

1. Especificações do sistema	3
1.1. Introdução do sistema	3
1.2. Componentes do sistema	3
1.3. Especificações técnicas do sistema	8
2. Manual operacional	9
2.1. Guia de montagem e inicialização do sistema	9
2.2. Instruções operacionais	10
2.3. Guia de manutenção e soluções para os problemas mais comuns	11
3. Considerações de segurança	x
3.1. Riscos operacionais	9
3.2. Adequação do local de operação do robô	10
3.3. Guia de prevenção e mitigação de acidentes	11

1. Especificações do sistema

1.1. Introdução

A proposta da solução é desenvolver, em simulação, um robô capaz de se mover em ambientes de espaço confinado, coletar dados a partir de sensores (principalmente de oxigênio e outros gases) e que utilize filmagens para apoiar na inspeção prévia da estrutura e na execução da atividade da Gerdau.

Desse modo, o protótipo do monitoramento dos ambientes industriais está sendo desenvolvido com a utilização do robô TurtleBot 3 Burguer, acompanhado de sensor de gás, umidade e temperatura, além de uma câmera.

Arquitetura do sistema

A arquitetura da solução consiste em três componentes principais: o embarcado, o backend e o frontend. No componente embarcado, o ROS2 é utilizado para controlar os movimentos do robô e se comunica com os periféricos, como câmera, sensor de gás, sensor de temperatura e LIDAR. O backend, por sua vez, recebe e envia informações usando o ROS2 e também incorpora o framework FastAPI, o banco de dados Supabase e o modelo YOLO para detecção de objetos. Já o frontend é construído com o framework React, permitindo a interação com o sistema através de chamadas HTTP para o backend. Essa arquitetura completa e integrada

oferece uma solução eficiente para atender às necessidades dos usuários.

A seguir, temos o diagrama de blocos da arquitetura da solução:

1.2. Componentes do sistema

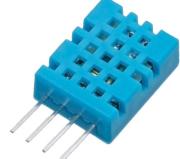
A fim de migrar o abstracionismo para conceitos mais visuais, foi desenvolvido um diagrama de arquitetura do projeto em que é mostrado cada etapa, componente e função dos elementos presentes, tais como suas ligações referente aos sensores e câmera. A partir disso, é possível visualizar de forma mais sintética toda a estrutura do projeto de forma comprehensível e organizada.

Componentes de Hardware

A seguir estão listados os componentes de hardware necessários para a montagem da solução.

Representação	Componente	Descrição	Função
	TurtleBot 3 Burguer	Robô móvel compacto e versátil, desenvolvido para pesquisa e desenvolvimento em robótica. Baseado na plataforma ROS, ele possui rodas omnidirecionais, sensores como câmera e giroscópio, e pode ser programado para realizar tarefas como navegação autônoma, mapeamento do ambiente e interação com humanos. É uma plataforma acessível e customizável, adequada para uma variedade de aplicações robóticas.	Atuador (movimentação).
	Raspberry pi 3	Computador de placa única (single-board computer) de baixo custo e tamanho reduzido, projetado para estimular o aprendizado e a prototipagem de projetos eletrônicos, ademais é alimentado por um processador ARM e possui diversos componentes integrados, como portas USB, conexão de rede, saída de vídeo e slots para cartões de memória.	Comunicação com o backend e processamento de todos os componentes.

	Sensor LIDAR 360°	<p>Dispositivo que utiliza a tecnologia LIDAR (Light Detection and Ranging) para medir distâncias e criar mapas tridimensionais do ambiente ao seu redor, emite pulsos de luz laser e mede o tempo que leva para os pulsos retornarem após atingirem objetos próximos. Com base nessas medições, o sensor LIDAR pode calcular a distância e a posição dos objetos em todas as direções ao seu redor, em um campo de visão completo de 360 graus.</p>	<p>Mapeamento de ambiente e detecção de obstáculos, com uma percepção precisa do ambiente em tempo real.</p>
	Webcam	<p>Dispositivo de captura de vídeo que pode ser conectado a um computador ou dispositivo móvel para transmitir ou gravar imagens em tempo real. Geralmente, uma webcam possui uma câmera digital integrada e um microfone para capturar tanto vídeo quanto áudio.</p>	<p>Captar as imagens do ambiente confinado.</p>
	Arduino UNO	<p>É uma placa de desenvolvimento de código aberto amplamente utilizada no campo da eletrônica e programação.</p>	<p>Leitura dos dados captados pelo sensor de gás.</p>

	Sensor de umidade e temperatura DHT11	O sensor de umidade DHT11 é um componente eletrônico capaz de medir a umidade relativa do ar e a temperatura ambiente. É amplamente utilizado em projetos de automação residencial e monitoramento de condições climáticas.	Deteta umidade relativa do ar e temperatura.
	Sensor de gás MQ135	Dispositivo de detecção de gases que determinam a qualidade do ar. Capta gases tais como dióxido de carbono, monóxido de carbono, amônia, benzeno, álcool, fumaça, entre outros.	Deteta a qualidade do ar.

Componentes de Software

A seguir estão listados os componentes externos que fazem parte da solução, além de eventuais serviços em nuvem, softwares, entre outras aplicações utilizadas.

Componente	Descrição	Tipo: entrada / saída / atuador
Interface WEB	Utilizando a interface WEB, o parceiro terá acesso a informações do espaço confinado.	Saída
Roteador wifi	É o que fará a conexão do dispositivo com a internet, dessa forma, permitindo que ele envie dados para o banco de dados.	Atuador
Computador	O computador é responsável por configurar os dispositivos e fazê-los funcionar.	Atuador

Arduino IDE	A função do Arduino IDE é simples, no computador, ele se encarrega de compilar o código em C++ e enviá-lo para o dispositivo, além de poder ser usado como debugger para acompanhar o que acontece no sensor.	Atuador
Next.js	Next.js é um framework de desenvolvimento web em JavaScript que permite criar aplicativos React com renderização do lado do servidor (SSR). Ele simplifica a construção de aplicações web rápidas, escaláveis e com ótima experiência do usuário.	Atuador
SCSS	SCSS (Sassy CSS) é uma extensão do CSS que traz recursos avançados, como variáveis, aninhamento de seletores e mixins, para facilitar e organizar a escrita de estilos em projetos web. Ele compila para CSS padrão e é amplamente utilizado para criar estilos mais flexíveis e reutilizáveis.	Atuador
Google Sheets ou Excel	Plataforma do Google usada como banco de dados, onde são guardadas as informações coletadas pelo dispositivo e distribuídas, utilizando IFTTT à interface WEB.	Atuador
JavaScript	Linguagem de programação utilizada para funções simples na interface WEB como troca de telas e para pegar informações do banco de dados.	Atuador
Supabase	Plataforma de desenvolvimento de aplicativos de código aberto que oferece uma infraestrutura completa de backend como serviço (BaaS).	Atuador
WSL	Sistema operacional de distribuição Linux.	Atuador
Ubuntu	Permite executar um ambiente Linux em um dispositivo do sistema operacional Windows.	Atuador

1.3. Especificações técnicas do sistema

A seguir estão listados os requisitos de conectividade que fazem parte da solução.

Componente	Descrição
TCP/IP	TCP/IP é um conjunto de protocolos de comunicação que permite a conexão e o intercâmbio de informações entre dispositivos em redes de computadores.
Roteador wifi	É o que fará a conexão do dispositivo com a internet, dessa forma, permitindo que ele envie dados para o banco de dados.
Computador	O computador é responsável por configurar os dispositivos e fazê-los funcionar.
WebSocket	WebSocket é um protocolo de comunicação bidirecional que permite a troca de dados em tempo real entre um cliente e um servidor, ideal para aplicações web que exigem atualizações frequentes e em tempo real.
ROS	ROS (Robot Operating System) é uma plataforma de software flexível e modular para desenvolvimento de robótica, que fornece uma estrutura de comunicação entre os componentes do sistema, facilitando o desenvolvimento, controle e integração de robôs e dispositivos relacionados.

2. Manual operacional

2.1. Guia de montagem e inicialização do sistema

A seguir estão listados os passos detalhados para montar e inicializar o sistema. Assim, é possível encontrar instruções claras sobre a configuração de cada componente do projeto, desde a montagem física do robô até a configuração do software necessário.

Montagem do TurtleBot3

A montagem do TurtleBot3 é fornecida diretamente pelo fabricante e inclui componentes essenciais, como o microcontrolador OpenCR, o sensor de gás MQ2, o Raspberry Pi 3, a webcam e o sensor Lidar 360°. Para garantir uma instalação adequada, é recomendado seguir os passos detalhados no manual fornecido pelo fabricante. Dessa forma, para ligar o robô, é preciso pressionar o botão próximo à sua bateria. Após um efeito sonoro, pode-se afirmar que ele está ligado e pronto para se conectar à rede.

Quanto à configuração do software do TurtleBot3, é necessário realizar a instalação de diversas bibliotecas e pacotes, conforme documentado no guia de configuração. Esse guia oferece instruções passo a passo para o processo de instalação, visando

garantir o funcionamento correto do TurtleBot3 em conformidade com seus componentes.

Informações adicionais dos componentes

1. Microcontrolador OpenCR: Instalação do microcontrolador OpenCR de acordo com as especificações do fabricante, ele será responsável por coordenar os movimentos do robô.
2. Sensor de gás MQ2: Conectar o sensor de gás MQ2 ao sistema do robô, ele será responsável por medir os níveis de oxigênio e outros gases no ambiente.
3. Raspberry Pi 3: Instalação do Raspberry Pi 3 no robô, tal microprocessador executa o sistema operacional Ubuntu e coordena a comunicação entre os componentes do sistema.
4. Webcam: Conectar a webcam ao sistema do robô, ela será utilizada para capturar imagens que auxiliarão na inspeção prévia da estrutura e na execução das atividades.
5. Sensor Lidar 360°: Instalação do sensor Lidar 360° no robô. Esse sensor utilizará laser para medir distâncias e criar um mapa tridimensional do ambiente.

Inicialização do sistema operacional

A inicialização do sistema operacional está relacionada ao backend e frontend.

Configuração do componente backend:

1. Configuração do ambiente ROS2:

- Instalação do ROS2 em seu sistema operacional seguindo as instruções oficiais da documentação do ROS2;
- Criação de um novo workspace do ROS2 para o projeto.

2. Implementação da arquitetura ROS2:

- Criação dos pacotes ROS2 necessários para o seu projeto, usando o comando ‘ros2 pkg create’ para criar novos pacotes;
- Definição dos tópicos, mensagens e serviços necessários para a comunicação entre o robô e o resto do sistema, incluindo a definição de mensagens personalizadas para transmitir os dados relevantes entre os diferentes componentes.

3. Integração do FastAPI com o ROS2:

- Utilização do framework FastAPI para desenvolver a API do backend;
- Criação de endpoints no FastAPI para diferentes operações e interações com o sistema, como receber comandos do frontend, enviar informações para o robô e fornecer dados do sistema;
- Para estabelecer a comunicação entre o FastAPI e o ROS2, utilizamos a biblioteca rclpy do ROS2, que permite publicar mensagens em tópicos do ROS2 a partir do backend e receber mensagens de tópicos do ROS2 no backend.

4. Configuração do banco de dados Supabase:

- Acessamos o painel de administração do Supabase em <https://app.supabase.io> utilizando as credenciais de acesso:

URL:

API_key:

5. Integração do Supabase com o backend:

- Utilização o Supabase como serviço de banco de dados para armazenar e gerenciar os dados do sistema;
- Configuração da conexão do backend com o Supabase, fornecendo as credenciais de acesso obtidas anteriormente;
- Caso necessário, é possível utilizar um ORM (Object-Relational Mapping) como o SQLAlchemy para facilitar a interação com o banco de dados, permitindo realizar operações de criação, leitura, atualização e exclusão (CRUD) no banco de dados.

6. Integração do OpenCV e YOLO:

- Integração do OpenCV ao backend para a análise de imagens capturadas pela webcam.
- Implementação do YOLO (You Only Look Once) em conjunto com o OpenCV para realizar a detecção de objetos nas imagens, no caso do projeto, as rachaduras.
- Configuração da captura de imagens da webcam utilizando o OpenCV e envio das imagens para análise pelo YOLO.

7. Comunicação HTTP:

- Utilização da comunicação HTTP (HyperText Transfer Protocol) para interagir com o frontend e outros componentes do sistema.
- O FastAPI fornece uma estrutura robusta para a criação de endpoints HTTP, permitindo a troca de dados entre os diferentes componentes do sistema de forma eficiente e segura.

8. Inicialização do backend:

Para iniciar o serviço do backend FastAPI no seu ambiente de desenvolvimento, siga as etapas abaixo:

1. Certifique-se de ter o ambiente de desenvolvimento configurado e todas as dependências do backend instaladas corretamente.
2. Abra o terminal ou prompt de comando e navegue até a pasta raiz do projeto backend.
3. Execute o comando para iniciar o serviço do backend FastAPI. O comando pode variar dependendo das configurações do seu ambiente, mas geralmente é algo como: ‘uvicorn main:app --reload’. Esse comando iniciará o servidor FastAPI e tornará o backend disponível para receber requisições.

Após iniciar o serviço do backend, é importante realizar testes para verificar se o sistema está funcionando corretamente. A seguir, estão os testes que você pode executar:

1. Verifique se a comunicação entre o frontend e o backend está ocorrendo adequadamente, enviando requisições HTTP para os endpoints definidos no FastAPI. Utilize ferramentas como o Postman, cURL ou qualquer cliente HTTP para enviar requisições e verificar as respostas recebidas. Certifique-se de que as requisições são processadas corretamente e as respostas estão de acordo com as expectativas.
2. Verifique se as mensagens estão sendo enviadas e recebidas corretamente entre o backend e os componentes do ROS2. Certifique-se de que a comunicação entre o ROS2 e o backend esteja configurada corretamente. Envie mensagens de teste para o backend e verifique se elas estão sendo recebidas e processadas corretamente pelo ROS2. Confirme também se as respostas do ROS2 estão sendo enviadas de volta para o backend conforme o esperado.
3. Verifique se os dados estão sendo persistidos e recuperados adequadamente no banco de dados do Supabase, além de ter os frames salvos em um bucket. Realize testes de inserção, atualização e consulta de dados no banco de dados do Supabase para garantir que a integração esteja funcionando corretamente. Verifique se os dados são armazenados corretamente no banco de dados e se podem ser recuperados com precisão quando necessário. Além disso, confirme se os frames capturados estão sendo salvos no bucket de armazenamento designado e podem ser acessados quando necessário.

4. Verifique se a análise de imagens com o OpenCV e o YOLO está ocorrendo conforme o esperado. Envie imagens de teste para o backend e verifique se a análise de imagens está sendo realizada corretamente. Certifique-se de que o OpenCV e o YOLO estejam corretamente integrados ao backend e capazes de detectar objetos e retornar os resultados esperados.

Ao realizar esses testes, você poderá garantir que o sistema esteja funcionando corretamente, verificando a comunicação entre o frontend e o backend, o fluxo de mensagens com o ROS2, a persistência e recuperação de dados no banco de dados do Supabase, e a análise de imagens com o OpenCV e o YOLO. Essa abordagem ajudará a identificar e solucionar possíveis problemas no sistema.

9. Inicialização do frontend:

Para iniciar o frontend do sistema, siga os passos abaixo:

1. Certifique-se de ter o Node.js instalado em seu computador. Se você não tiver o Node.js instalado, faça o download e a instalação a partir do site oficial do Node.js (<https://nodejs.org>).
2. Abra o terminal ou prompt de comando e navegue até a pasta raiz do projeto.
3. Execute o comando ‘npm install’ para instalar todas as dependências necessárias. Certifique-se de que você esteja na pasta correta do projeto antes de executar o comando.

Esse comando irá baixar e instalar todas as bibliotecas e pacotes necessários para o funcionamento do sistema frontend.

4. Após a conclusão da instalação, execute o comando ‘npm run dev’ para iniciar o servidor de desenvolvimento. Esse comando irá iniciar o servidor local e disponibilizar o frontend para visualização.
5. Após executar o comando, você poderá acessar o frontend em seu navegador, utilizando o endereço fornecido pelo servidor local. Normalmente, o endereço será “<http://localhost:3000>” ou um endereço personalizado definido durante a configuração do ambiente. Certifique-se de manter o servidor em execução enquanto estiver trabalhando no frontend, para ter acesso contínuo à aplicação.

Posto isso, nosso frontend foi desenvolvido utilizando o Next.js, uma estrutura de desenvolvimento web front-end de código aberto baseada em React, criada pela Vercel. O Next.js oferece recursos avançados, como renderização no lado do servidor e geração de sites estáticos para aplicativos web React, proporcionando uma experiência de desenvolvimento mais eficiente e rápida.

Ao iniciar o sistema, você será direcionado para um dashboard com um design intuitivo e minimalista. Nesse dashboard, você encontrará a visão computacional, onde serão exibidos os dados e informações analisados. Haverá campos de entrada para

que você possa descrever o local e o endereço, bem como adicionar observações adicionais.

Além disso, o sistema contará com botões de controle que permitirão iniciar a movimentação do robô e gerar relatórios com base nas análises realizadas. Também haverá uma aba de histórico, onde todos os relatórios anteriores serão registrados e estarão disponíveis para consulta.

O frontend fornecerá uma interface amigável e funcional para interagir com o sistema, permitindo a análise dos dados atmosféricos capturados pelo robô e a geração de relatórios com facilidade.

Para armazenar e recuperar os dados apresentados no frontend, utilizaremos o Supabase como nosso banco de dados. O Supabase é uma plataforma confiável que oferece recursos de banco de dados e autenticação para desenvolvimento de aplicativos web. No Supabase, teremos tabelas específicas para armazenar os dados atmosféricos, informações relacionadas aos relatórios gerados e dados dos usuários.

Ao seguir essas instruções, você estará pronto para iniciar o frontend do sistema e aproveitar todas as funcionalidades oferecidas.

2.2. Instruções operacionais

O TurtleBot3 é operado por meio de uma plataforma web intuitiva, que se comunica com um backend para enviar comandos

ao robô. A seguir, apresentamos as instruções operacionais detalhadas para utilizar o TurtleBot3 de forma eficiente e realizar uma variedade de tarefas.

Acesso a Interface Web

Na interface web (como já foi explicado no tópico 2.1), estão disponíveis diversas opções para controlar o TurtleBot3, como movimentar o robô, utilizar a webcam e o sensor Lidar 360°, além de monitorar os dados do sensor de gás MQ2. Basta selecionar a ação desejada para que o comando seja enviado ao backend, que, por sua vez, repassa o comando ao TurtleBot3.

Ao utilizar a interface web, o usuário tem a possibilidade de monitorar o estado do TurtleBot3 de forma abrangente. É possível verificar a posição atual do robô, visualizar imagens capturadas pela webcam e pelo sensor Lidar 360°, e obter informações relevantes dos dados coletados pelo sensor de gás MQ2.

Funcionalidades da Interface Web

A interface web oferece um conjunto de tarefas que o usuário pode executar com o TurtleBot3, proporcionando uma experiência personalizada. Algumas das tarefas disponíveis incluem:

1. **Geração de Relatórios:** Os usuários podem criar relatórios detalhados sobre as atividades do TurtleBot3. Para isso, basta acessar a seção de relatórios na interface web e seguir as instruções fornecidas.

2. **Gerenciamento de Análises:** A interface permite aos usuários visualizar, editar, excluir e cadastrar análises relacionadas ao TurtleBot3. Para gerenciar análises, basta acessar a seção correspondente na interface e seguir as orientações apresentadas.
3. **Acompanhamento em Tempo Real:** Por meio da interface web, os usuários podem acompanhar em tempo real as atividades do TurtleBot3. Ao acessar a seção de acompanhamento em tempo real, é possível obter informações atualizadas e relevantes sobre as ações e o status do robô.

Interpretação dos Dados dos Sensores

A interface web exibe os dados coletados pelos sensores do TurtleBot3, como informações do sensor de gás MQ2, além das imagens capturadas pela webcam e dados fornecidos pelo sensor Lidar 360°. Dessa forma, para interpretar esses dados de forma adequada, é essencial que o usuário compreenda a função de cada sensor e os interprete de acordo com seus parâmetros, os dados retornados no relatório fornecido na interface web.

Cuidados na operação

Ao operar o TurtleBot3, é importante prestar atenção às condições do ambiente e do robô. Abaixo estão descritas algumas dessas situações:

- Verifique regularmente a carga da bateria: evite utilizar o robô quando os níveis de energia estiverem abaixo do normal.

Nessa circunstância, o robô emitirá um aviso sonoro e um aviso na interface web para que o operador tome uma ação de interrupção do sistema.

- Assegure-se de que o TurtleBot3 está em uma área segura antes de iniciar qualquer operação, de forma a evitar danos em seus componentes.
- Utilize o TurtleBot3 em locais com substâncias perigosas ou inflamáveis somente se o robô estiver equipado com os sensores.
- Mantenha sempre vigilância sobre o estado do TurtleBot3 por meio da interface web durante a operação.
- Em caso de qualquer problema ou comportamento inesperado, interrompa a operação do robô.

2.3. Guia de manutenção e soluções para os problemas mais comuns

A fim de garantir o bom funcionamento contínuo do TurtleBot3, é importante estar ciente de possíveis problemas e saber como solucioná-los. No Guia de Manutenção, fornecemos informações valiosas sobre os problemas mais comuns enfrentados pelos usuários e as soluções recomendadas para resolvê-los.

Manutenção do TurtleBot3

A manutenção adequada do TurtleBot3 é crucial para garantir um desempenho consistente e uma vida útil prolongada. Aqui estão

algumas diretrizes essenciais para manter o TurtleBot3 em ótimas condições:

1. Limpeza e Proteção:

A limpeza regular do TurtleBot3 é fundamental para remover poeira e detritos que possam comprometer os sensores e componentes. Utilize um pano macio e seco para limpar suavemente a superfície do robô. Além disso, considere utilizar uma capa de proteção para evitar a entrada de sujeira durante o armazenamento.

2. Verificação dos Componentes:

Realize inspeções periódicas em todos os componentes do TurtleBot3, como o microcontrolador OpenCR, o sensor de gás MQ2, o Raspberry Pi 3, a webcam e o sensor Lidar 360°. Verifique se estão devidamente conectados e funcionando corretamente. Caso identifique algum dano ou mau funcionamento, substitua o componente afetado o mais rápido possível.

3. Atualização de Firmware e Software:

Mantenha o firmware do TurtleBot3 atualizado, pois as atualizações podem corrigir problemas conhecidos e melhorar o desempenho. Além disso, verifique se há atualizações de software, como bibliotecas e pacotes, para garantir compatibilidade e segurança. Siga as instruções do fabricante para realizar as atualizações de forma adequada.

4. Calibração dos Sensores:

Realize a calibração regular dos sensores do TurtleBot3, como o sensor Lidar 360°, para garantir medições precisas e confiáveis.

5. Backup de Dados e Configurações:

Faça regularmente backup dos dados e configurações importantes do TurtleBot3, como mapas ou scripts personalizados. Isso ajudará a evitar perda de informações valiosas em caso de falhas ou reinicialização do sistema.

Soluções para problemas mais comuns

1. Problemas de Conectividade com o Backend:

Caso ocorram dificuldades na conexão do TurtleBot3 com o backend, é fundamental verificar a disponibilidade e o correto funcionamento do backend. Certifique-se também de que o TurtleBot3 esteja conectado à rede correta. Se persistirem as falhas, tente reiniciar tanto o TurtleBot3 quanto o backend para restabelecer a comunicação.

2. Abordando Questões Relacionadas aos Sensores:

Caso algum sensor do TurtleBot3 apresente mau funcionamento, inicie verificando se o sensor está limpo e livre de qualquer obstrução física. Se mesmo assim o sensor continuar apresentando problemas, é possível que seja necessário substituí-lo por um componente funcional e compatível com o TurtleBot3.

3. Solucionando Problemas com a Interface Web:

Se a interface web não estiver operando corretamente, tente atualizar a página ou experimentar utilizar um navegador diferente para verificar se o problema é específico do navegador em uso. Caso persistam as dificuldades, é possível que ocorra um problema no frontend. Nesse caso, é recomendável entrar em contato com o suporte técnico para receber assistência especializada na resolução do problema.

4. Recursos de Suporte Adicionais:

Caso encontre desafios que não possa solucionar por conta própria, é altamente recomendado buscar suporte técnico junto às empresas ou desenvolvedores envolvidos no projeto. Certifique-se de fornecer o máximo de informações detalhadas possível sobre o problema, incluindo quaisquer mensagens de erro ou comportamentos anômalos que tenha observado. Isso ajudará a equipe de suporte a compreender melhor a situação e fornecer a assistência adequada para solucionar o problema de forma eficaz.

5. Expansão e Customização do TurtleBot3:

Além de resolver problemas comuns, o TurtleBot3 também oferece oportunidades de expansão e customização. Você pode explorar diferentes módulos e sensores compatíveis para aprimorar as capacidades do robô. Além disso, a comunidade de desenvolvedores do TurtleBot3 é rica em recursos e conhecimentos, oferecendo fóruns e documentação detalhada para inspirar novos projetos e compartilhar soluções inovadoras.

6. Atualizações e Melhorias Contínuas:

Manter-se atualizado com as versões mais recentes de firmware, software e bibliotecas relacionadas ao TurtleBot3 ajudará a aproveitar melhorias de desempenho, correções de bugs e novos recursos. Consulte os recursos oficiais e a documentação disponível para obter informações sobre as atualizações e como aplicá-las corretamente em seu TurtleBot3.

3. Considerações de segurança

3.1. Riscos operacionais

Ao lidar com a operação de ambientes industriais e a inspeção prévia da estrutura, apresentamos nesta seção, uma descrição abrangente dos riscos operacionais associados à utilização do sistema, a fim de conscientizar os usuários sobre os possíveis desafios e tomar as medidas adequadas para mitigar esses riscos.

1. Risco de exposição a substâncias perigosas:

O sistema é projetado para detectar gases potencialmente perigosos nos espaços confinados. No entanto, é importante ressaltar que certas substâncias tóxicas podem não ser detectadas ou podem ser detectadas com atraso devido a fatores como a concentração ou a interação com outros elementos presentes no ambiente. Portanto, os usuários devem sempre utilizar equipamentos de proteção individual apropriados e seguir os procedimentos de segurança estabelecidos pela organização para minimizar o risco de exposição a substâncias nocivas.

2. Risco de falhas técnicas:

O sistema é composto por diversos componentes, como o robô TurtleBot 3 Burger, sensores, câmeras e sistemas de comunicação. Embora tenham sido tomadas medidas para garantir a confiabilidade desses componentes, ainda há a possibilidade de falhas técnicas. Por exemplo, pode ocorrer falha de comunicação entre os dispositivos, falhas de energia ou falhas no sistema de controle do robô. Os usuários devem estar cientes desses riscos e realizar verificações regulares dos equipamentos, bem como manter backups adequados dos dados coletados para evitar perda de informações críticas.

3. Risco de segurança física:

Durante a operação do sistema, é essencial considerar a segurança física dos operadores e das instalações industriais. O robô e outros componentes móveis podem representar riscos de colisão com objetos ou pessoas, especialmente em espaços confinados. Os usuários devem garantir que a área de trabalho esteja adequadamente delimitada e que medidas de segurança, como barreiras ou sistemas de detecção de proximidade, sejam implementadas para prevenir acidentes. Além disso, é importante seguir os procedimentos operacionais recomendados e receber treinamento adequado para garantir a segurança durante a interação com o sistema.

4. Risco de segurança cibernética:

O sistema de monitoramento de ambientes industriais depende de tecnologias de comunicação e armazenamento de dados. Isso pode expor o sistema a ameaças cibernéticas, como ataques de hackers ou malwares. Os usuários devem implementar medidas de segurança cibernética, como o uso de firewalls, criptografia de dados e atualizações regulares de software, para proteger o sistema contra essas ameaças. Além disso, é importante educar os usuários sobre as práticas recomendadas de segurança cibernética, como o cuidado ao abrir e-mails ou clicar em links suspeitos, a fim de evitar a exploração de vulnerabilidades do sistema.

Ao estar ciente desses riscos operacionais e seguir as diretrizes fornecidas neste manual do usuário, os usuários poderão utilizar o sistema de monitoramento de ambientes industriais com segurança e eficiência. É fundamental que os usuários recebam treinamento adequado sobre os procedimentos de segurança e operacionais, realizem a manutenção regular do sistema e relatem quaisquer problemas ou preocupações à equipe responsável. Da mesma forma, a equipe responsável pela operação e manutenção do sistema deve estar devidamente treinada e consciente dos riscos envolvidos. A adoção de procedimentos de segurança, manutenção regular, testes de funcionalidade e monitoramento contínuo são essenciais para mitigar os riscos operacionais e garantir a operação segura e eficiente do sistema de monitoramento de ambientes industriais.

3.2. Adequações do local de operação do robô

Ao utilizar o sistema de monitoramento de ambientes confinados com o robô TurtleBot 3 Burguer (que é o caso da nossa solução), é fundamental considerar diversos aspectos para garantir a adequação do local de operação. Esses aspectos visam não apenas a segurança e eficiência do sistema, mas também o bom desempenho das atividades realizadas.

1. Infraestrutura e ambiente

- Verifique se o ambiente possui uma superfície estável, nivelada e adequada para a movimentação segura do robô.
- Certifique-se de que o local ofereça suporte para a comunicação sem fio necessária entre o robô e os componentes do sistema.
- Considere a existência de rampas, degraus ou outras características que possam exigir ajustes ou precauções adicionais durante a locomoção do robô.

2. Obstáculos e interferências

Apesar de termos um script de detecção de obstáculos integrado ao sistema, com a intenção de evitar colisões, é importante lembrar de alguns objetivos .

- Identifique objetos fixos, como paredes, pilares, maquinários ou outros elementos que possam obstruir o trajeto do robô. Remova ou contorne esses obstáculos para garantir uma movimentação livre de interferências.
- Certifique-se de que não existam objetos soltos, fios, cabos ou materiais que possam se enroscar no robô durante sua operação. Mantenha o local organizado e livre de qualquer item que possa representar riscos.

3. Condições ambientais

Além disso, considere as condições ambientais do local. Verifique a iluminação para garantir que seja adequada para as atividades de captura de imagens da câmera do robô. Ambientes com baixa luminosidade podem comprometer a qualidade das imagens e dificultar a detecção de detalhes importantes. Também é importante avaliar a presença de gases, poeira ou outros agentes que possam afetar a integridade do sistema ou a saúde dos operadores. Em casos específicos, equipamentos de proteção individual podem ser necessários para a segurança dos operadores.

4. Equipe

Por fim, é fundamental que a equipe responsável pela operação do sistema esteja devidamente treinada e familiarizada com os procedimentos de segurança e operacionais. É recomendado seguir as diretrizes e instruções fornecidas pela equipe de suporte técnico, realizar manutenções regulares do robô e relatar quaisquer problemas ou preocupações identificados durante a operação.

Ao considerar todos esses aspectos e tomar as devidas providências, você estará garantindo a adequação do local de operação do robô, proporcionando uma operação segura, eficiente e livre de incidentes.

3.3. Guia de prevenção e mitigação de acidentes

A segurança é nossa maior preocupação ao operar o sistema de monitoramento de ambientes industriais com o robô TurtleBot 3 Burguer. Neste guia, apresentamos diretrizes essenciais para prevenção e mitigação de acidentes, visando garantir uma operação segura e proteger tanto os operadores quanto o ambiente de trabalho.

1. Treinamento e Conscientização:

- Todos os usuários envolvidos na operação do sistema devem receber treinamentos abrangentes sobre as funcionalidades, procedimentos de segurança e boas práticas de operação.
- Esteja ciente dos riscos potenciais associados à operação do robô e familiarize-se com as medidas de segurança necessárias para mitigar esses riscos.
- Mantenha-se atualizado sobre as diretrizes de segurança e políticas de operação estabelecidas pela equipe responsável.

2. Inspeção e Preparação do Ambiente:

- Antes de iniciar a operação do robô, realize uma inspeção minuciosa do ambiente de trabalho, identificando e removendo obstáculos, objetos soltos, fios ou cabos que possam interferir na movimentação do robô.

- Certifique-se de que o ambiente oferece uma superfície estável e adequada para a movimentação segura do robô. Evite superfícies escorregadias, desniveladas ou instáveis.

- Avalie as condições ambientais, incluindo iluminação adequada e a presença de gases, poeira ou outros agentes que possam afetar a operação do sistema. Tome medidas para garantir a integridade do sistema e a segurança dos operadores.

3. Monitoramento e Supervisão:

- Durante a operação do robô, mantenha uma supervisão constante do sistema, observando atentamente seu desempenho e comportamento.

- Esteja preparado para intervir manualmente, se necessário, em situações de risco iminente ou quando o script de detecção de obstáculos não for capaz de evitar uma colisão.

- Monitore regularmente as condições do ambiente e verifique se não ocorreram mudanças significativas que possam comprometer a segurança do robô e dos operadores.

4. Manutenção Regular:

- Realize a manutenção regular do robô e de seus componentes de acordo com as especificações e recomendações do fabricante.

- Verifique o funcionamento adequado dos sensores, atuadores e sistemas de segurança do robô.

- Substitua imediatamente peças desgastadas ou danificadas e siga as orientações de manutenção preventiva fornecidas.

Lembramos que a prevenção de acidentes é responsabilidade conjunta de todos os envolvidos na operação do sistema. Siga todas as orientações deste guia, esteja sempre atento aos riscos e tome as medidas necessárias para garantir a segurança durante a operação do sistema de monitoramento de ambientes industriais com o robô TurtleBot 3 Burguer. Em caso de dúvidas ou incidentes, contate a equipe de suporte técnico para obter a assistência adequada. A segurança é primordial.