

Arquitetura HIL para teste de sistemas embarcados como *vehicle interfaces* de veículos autônomos baseados no Autoware

Projeto – Etapa 1

Gabriel Toffanetto França da Rocha
g289320@dac.unicamp.br

Professor Dr. Rodrigo Moreira Bacurau

IM420X – Projeto de Sistemas Embarcados de Tempo Real
Faculdade de Engenharia Mecânica
Universidade Estadual de Campinas

8 de outubro de 2024



Schedule

- 1** Introdução
- 2** Proposta
- 3** Arquitetura
- 4** Cronograma
- 5** Referências bibliográficas



Introdução



Contextualização



Figura 1: Veículo Autônomo do LMA.

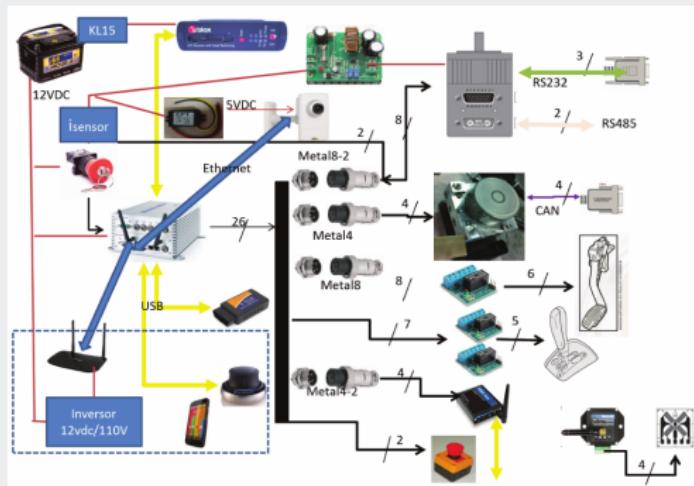
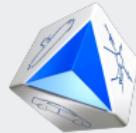


Figura 2: Diagrama de *hardware* do VILMA01 (BEDOYA, 2016).



Autoware

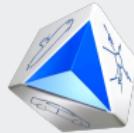
O que é?

- Projeto de software *open-source* que consiste em todas as funcionalidades requeridas para condução autônoma, em uma arquitetura modular com interfaces e APIs bem definidas;
- Primeiro "*all-in-one*" *open-source software* para veículos autônomos.

Princípios

- Projetado para suprir as necessidades de diferentes aplicações autônomas;
- Desenvolvido com as melhores práticas e padrões para alcançar alta qualidade e segurança em produtos para o mundo real.

"Autoware continuously evolves to offer more capability towards curb-to-curb Level 4 autonomous driving."



Arquitetura

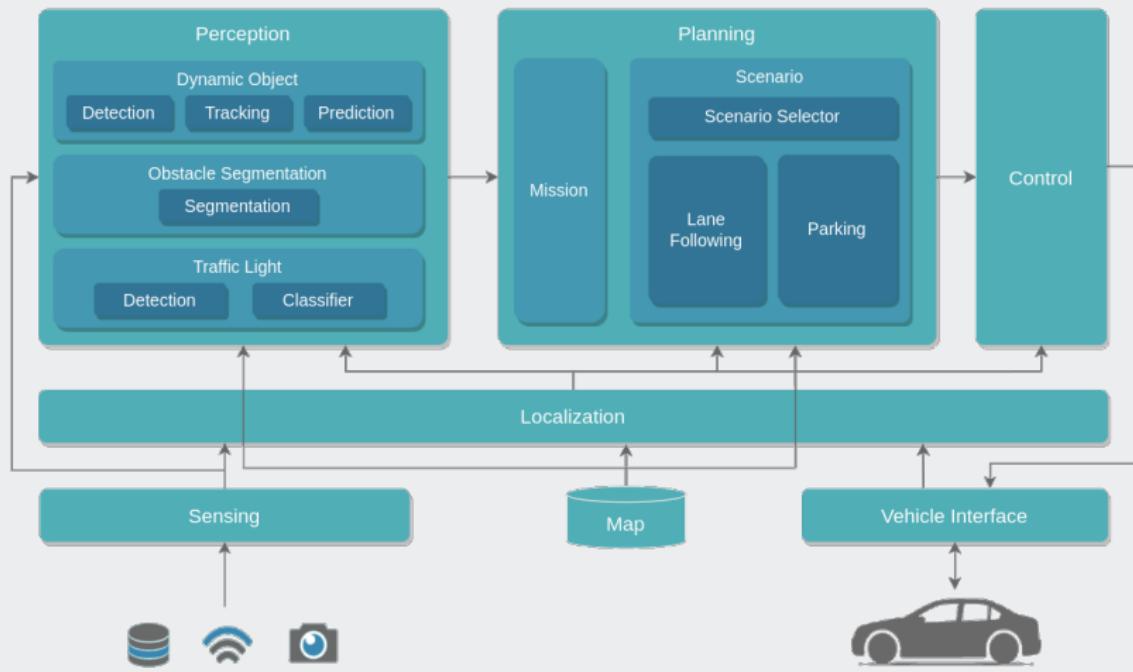
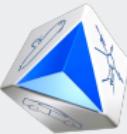


Figura 3: Arquitetura de alto nível (The Autoware Fundation, 2023).



ROS – Robot Operation System

O que é?

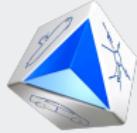
- Framework para robótica que contempla tudo que um robô precisa;
 - Ferramentas e bibliotecas;
 - Protocolos de comunicação;
 - Interfaceamento.
- Arquitetura baseada em sistema distribuído (ROS 2);
- Alta modularização com reaproveitamento de código próprio ou da comunidade;
- Portabilidade simulação/hardware.



Figura 4: Ecossistema ROS (Open Robotics, 2021).



Proposta



Proposta

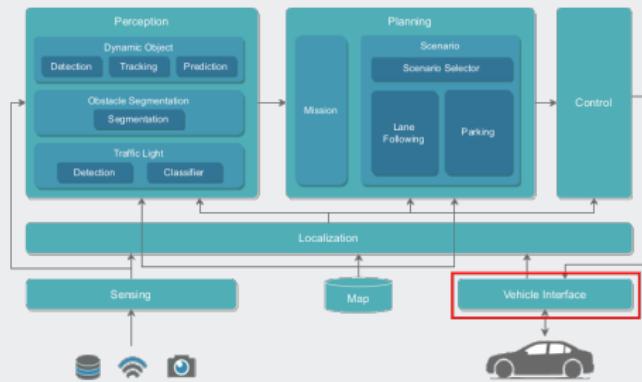


Figura 5: Escopo do projeto na arquitetura Autoware.

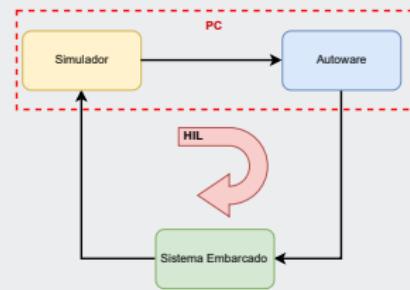
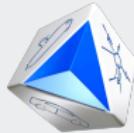


Figura 6: Arquitetura de teste do hardware.



Proposta

Objetivos

- Desenvolvimento de um sistema embarcado capaz de agir como *vehicle interface* para um veículo autônomo compatível com o Autoware utilizando o MicroROS;
- Teste do sistema embarcado por meio de *Hardware-In-the-Loop* com o simulador CARLA.

Justificativa

- O Autoware é um sistema para carros autônomos em ascensão, sendo importante que sistemas embarcados presentes nesses veículos sejam capazes de se integrar com ele. Dessa forma, a proposta da implementação de um sistema embarcado como *vehicle interface* garante a interligação entre microcontroladores STM32 ao *framework*. A validação por meio de HIL se faz interessante por substituir a necessidade de um protótipo real para testes, garantindo mais segurança, praticidade e redução de custos no desenvolvimento do projeto.



Requisitos

Requisitos funcionais

- Comunicação com o Autoware;
- Controle da aceleração, frenagem e direção do veículo;
- Controle dos faróis e luzes de sinalização (seta) do veículo;
- Teleoperação do veículo por um joystick em hardware.

Requisitos não-funcionais

- A *vehicle interface* deve ser construída na forma de um pacote portável para outros microcontroladores STM32;
- O interfaceamento com o veículo deve ser intercambiável com diferentes configurações;
- Deve-se garantir sincronização de *timestamp* entre o Autoware e o microcontrolador;
- O sistema embarcado deve abstraír o veículo como um sistema *Drive-By-Wire* (DBW) para o Autoware.



Componentes

Placa de desenvolvimento NUCLEO-H753ZI

- Microcontrolador STM32H753ZI;
- ARM Cortex-M7;
- 1 MB RAM;
- 2 MB Flash;
- 480 MHz (max) CPU;
- DMA;
- Comunicação:
 - UART/USART;
 - Ethernet;
 - USB.
- Custo: 27 dolares.

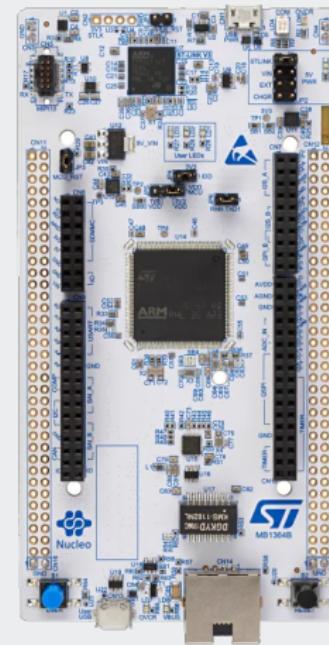
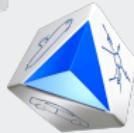


Figura 7: NUCLEO-753ZI.



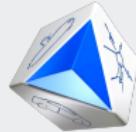
Componentes

Servidor

- Custo: 27 dolares.



Figura 8: Servidor.



Componentes

Joystick

- Custo: 27 dolares.

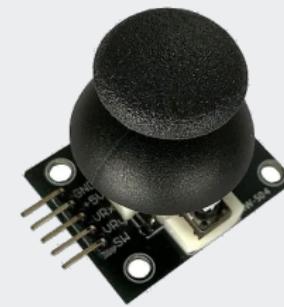


Figura 9: Joystick 2 eixos.



Arquitetura

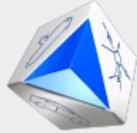


Diagrama de blocos

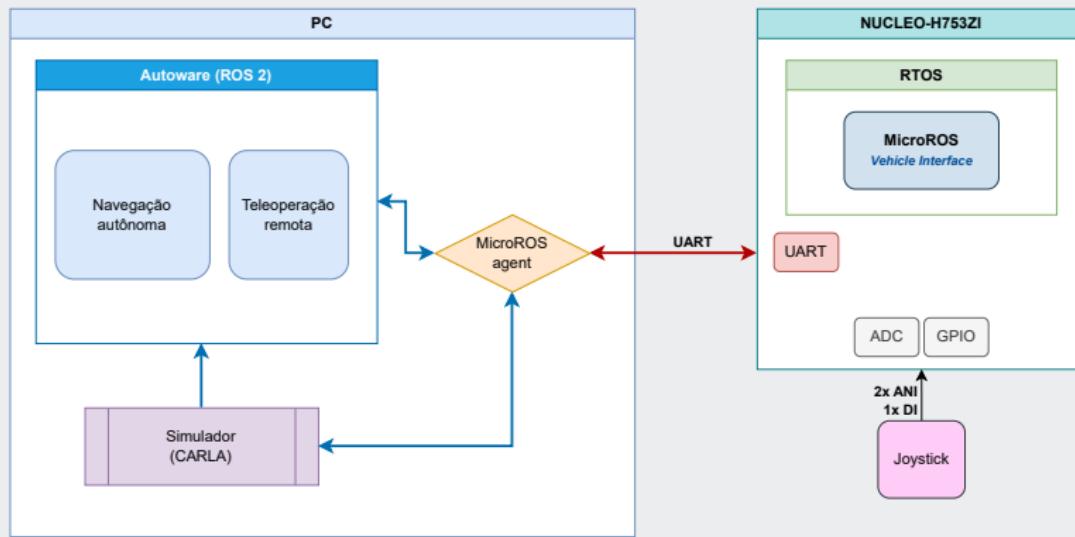


Figura 10: Diagrama de blocos da arquitetura HIL.



Esquemático

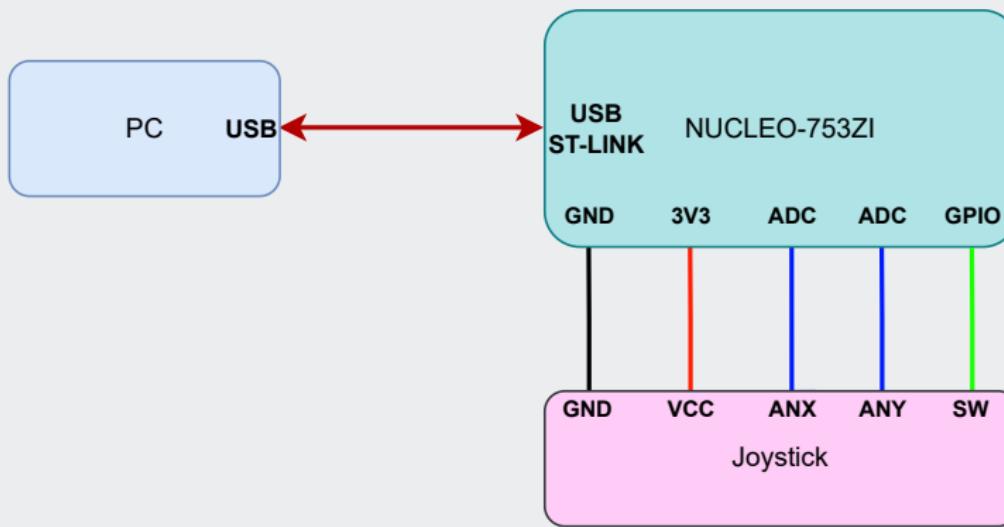


Figura 11: Esquemático de ligações elétricas.



Cronograma

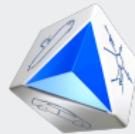


Cronograma

Atividade/Semana	1	2	3	4	5	6	7	8	9
Proposta do projeto	■								
Projeto de <i>hardware e software</i>		■	■						
Integração do STM com o MicroROS		■							
Integração do MicroROS com o Autoware			■	■					
Implementação das tarefas do sistema embarcado				■	■	■	■		
Construção do ambiente de testes					■	■	■		
Realização dos testes						■		■	■
Escrita do relatório		■	■	■	■	■			

Tabela 1: Cronograma de atividades.

- **Semana 2:** Apresentação Etapa 1
- **Semana 4:** Apresentação Etapa 2
- **Semana 7:** Apresentação Etapa 3
- **Semana 9:** Apresentação Final



Referências bibliográficas



Referências bibliográficas

-  **BEDOYA, O. G.** **Análise de risco para a cooperação entre o condutor e sistema de controle de veículos autônomos.** Tese (Doutor em Engenharia Mecânica) — Universidade Estadual de Campinas, Campinas, SP, fev. 2016. Disponível em: <<https://repositorio.unicamp.br/Busca/Download?codigoArquivo=471471>>.
-  **Open Robotics.** **The ROS Ecosystem.** 2021. Disponível em: <<https://www.ros.org/blog/ecosystem/>>.
-  **The Autoware Fundation.** **Architecture overview.** 2023. Disponível em: <<https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture/>>.

