



UNICAMP

Universidade Estadual de Campinas

Faculdade de Engenharia Mecânica

IM420X – Projeto de Sistemas Embarcados de Tempo Real

Novembro de 2024

Docente: Dr. Rodrigo Moreira Bacurau

Discentes:

– Gabriel Toffanetto França da Rocha – 289320

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

Sumário

1	Resumo	2
2	Motivação	3
3	Documentação	5
3.1	Requisitos	6
3.2	Componentes	7
3.3	Arquitetura	8
3.4	Método de desenvolvimento	9
3.5	Projeto de <i>software</i>	10
4	Manual de utilização	15
5	Problemas identificados e não resolvidos	16
6	Códigos da comunidade	17
	Referências	18
	Apêndices	19

1 Resumo

2 Motivação



Figura 1: Veículo Autônomo do LMA.

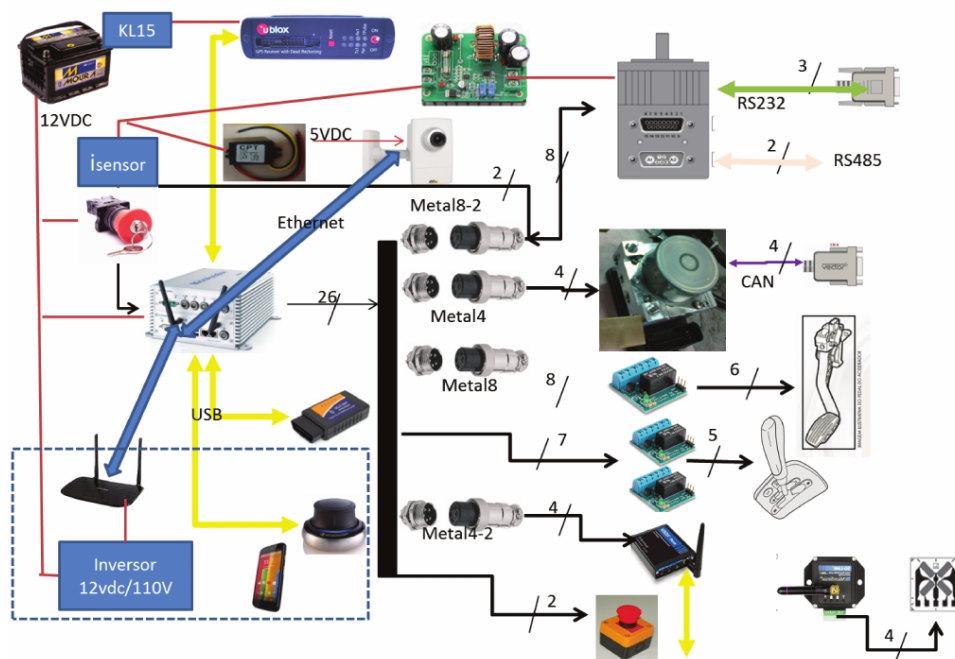


Figura 2: Diagrama de *hardware* do VILMA01 (BEDOYA, 2016).

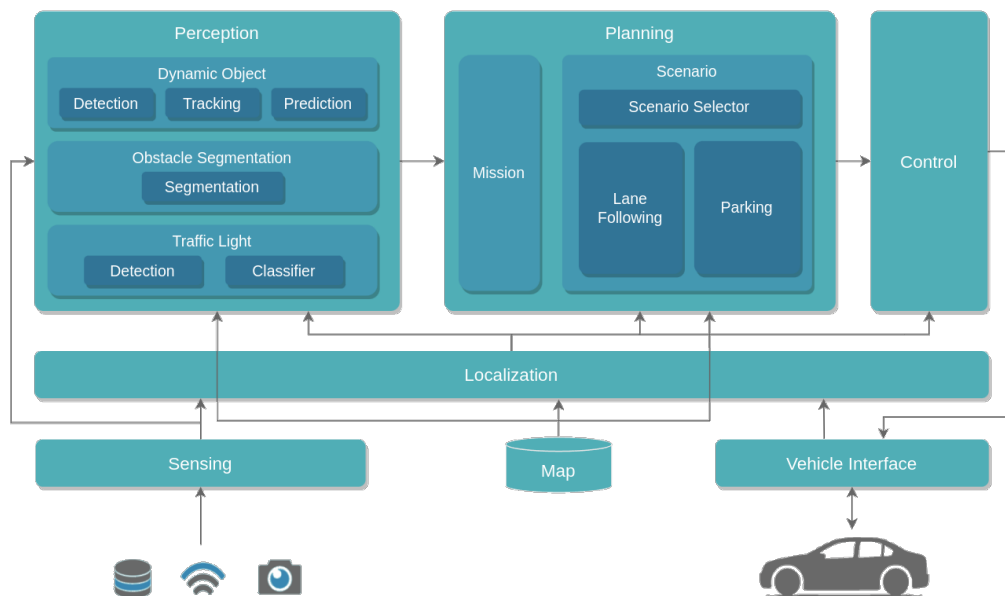


Figura 3: Arquitetura de alto nível (The Autoware Foundation, 2023).

3 Documentação

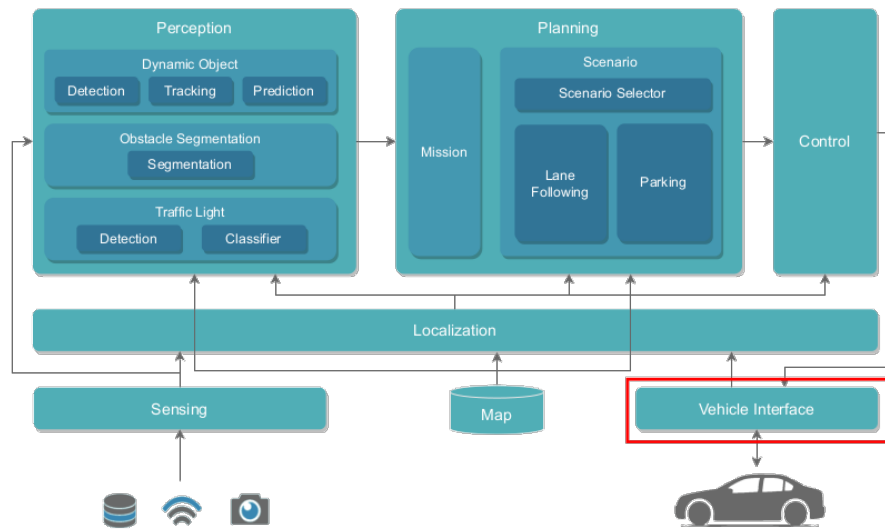


Figura 4: Escopo do projeto na arquitetura Autoware.

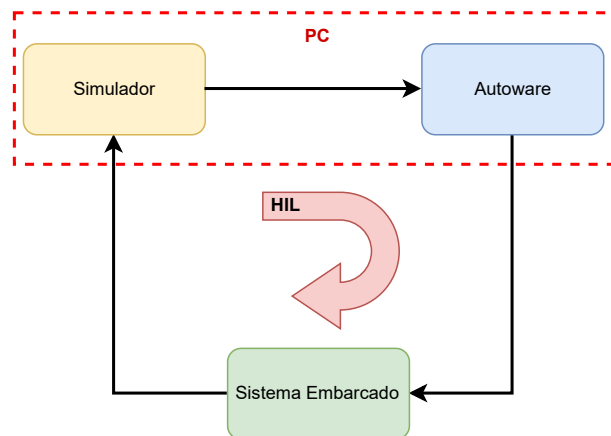


Figura 5: Arquitetura de teste do *hardware*.

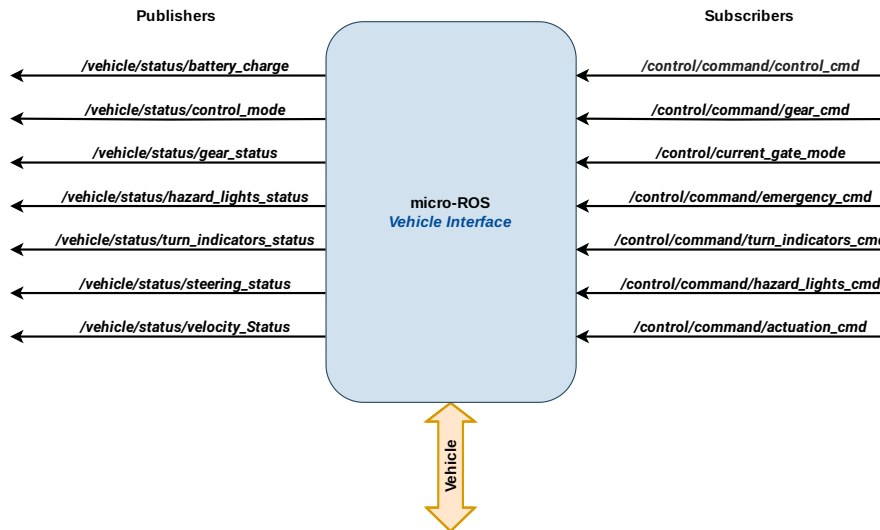


Figura 6: Diagrama de tópicos da *vehicle interface*.

3.1 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*;
- Troca do modo de operação por meio da *switch* do *joystick*;
- Subscrição por meio do micro-ROS em todos os tópicos necessários do Autoware;
- Publicação a partir micro-ROS em todos os tópicos necessários do Autoware.

Requisitos não funcionais

- A *vehicle interface* deve ser construída na forma de um pacote portátil 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 abstrair o veículo como um sistema *Drive-By-Wire* (DBW) para o Autoware.

3.2 Componentes

Placa de desenvolvimento NUCLEO-H753ZI

- Microcontrolador STM32H753ZI;
- ARM Cortex-M7;
- 1 MB RAM;
- 2 MB Flash;
- *Clock* máximo de 480 MHz;
- DMA;
- Comunicação:
 - UART/USART;
 - Ethernet;
 - USB.
- Custo: US\$ 27,00.

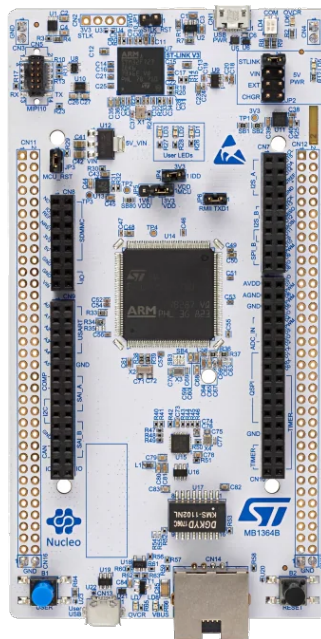


Figura 7: NUCLEO-753ZI.

Joystick

- Tensão de operação: 3V3 – 5V;
- Saída analógica referente ao eixo x ;

- Saída analógica referente ao eixo y ;
- Saída digital referente ao eixo z ;
- Custo: R\$ 10,00.



Figura 8: Joystick 2 eixos.

3.3 Arquitetura

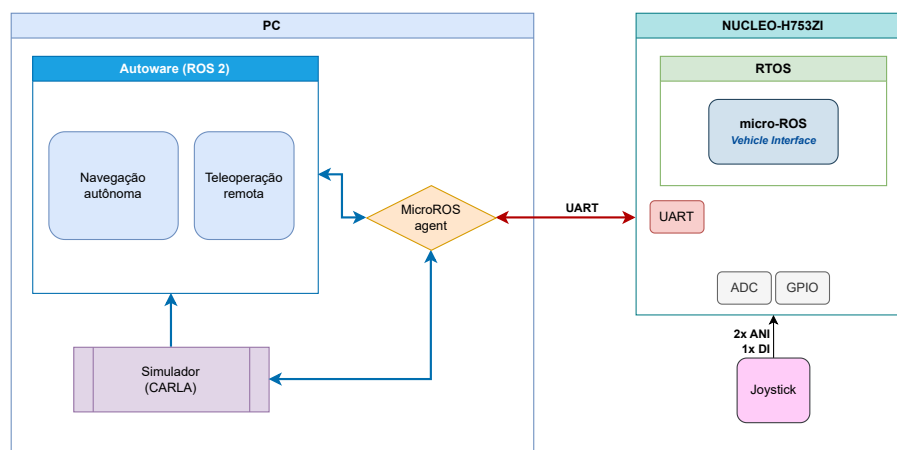


Figura 9: Diagrama de blocos da arquitetura HIL.

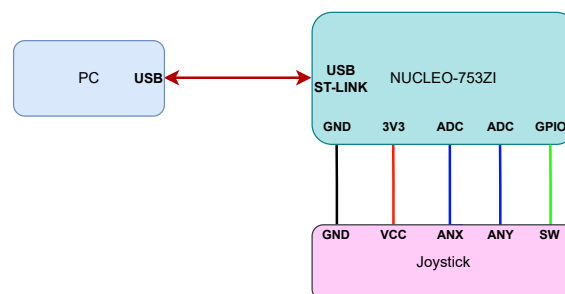


Figura 10: Esquemático de ligações elétricas.

3.4 Método de desenvolvimento

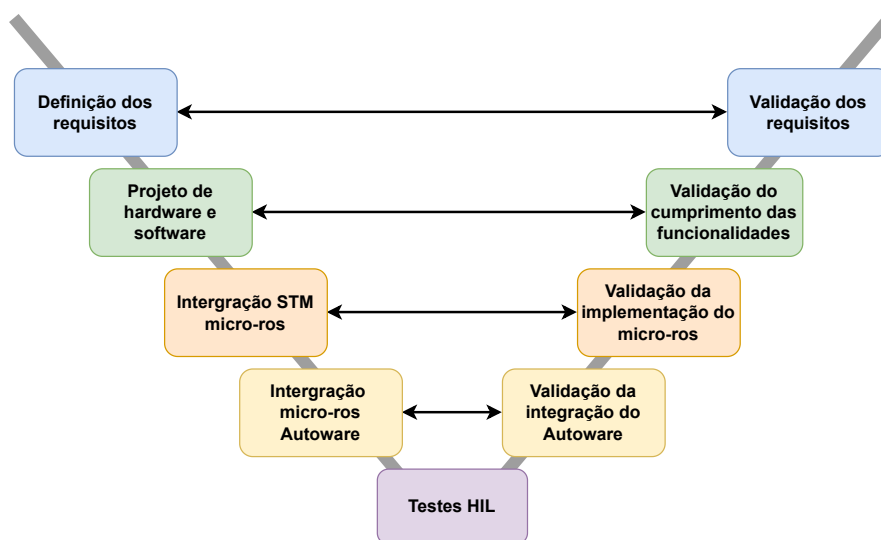


Figura 11: Modelo de execução das atividades do projeto.

Atividade/Semana	1	2	3	4	5	6	7	8	9
Proposta do projeto									
Projeto de <i>hardware</i> e <i>software</i>									
Integração do STM com o micro-ROS									
Integração do micro-ROS 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

3.5 Projeto de *software*

Estados do sistema

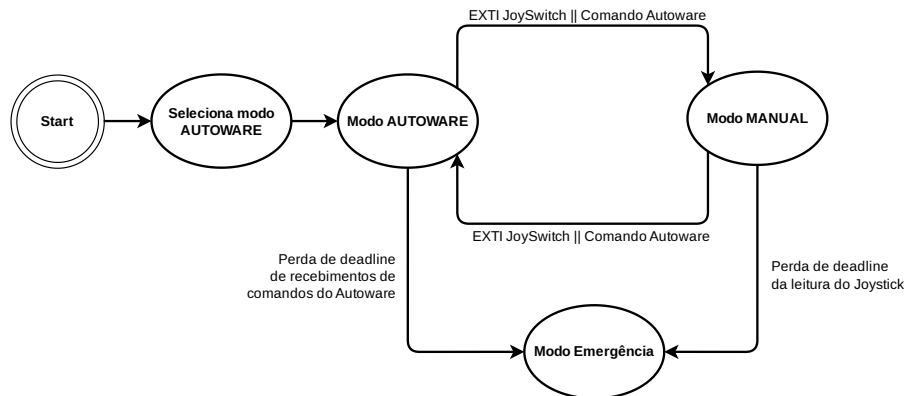


Figura 12: Máquina de estados do sistema.

Tarefas

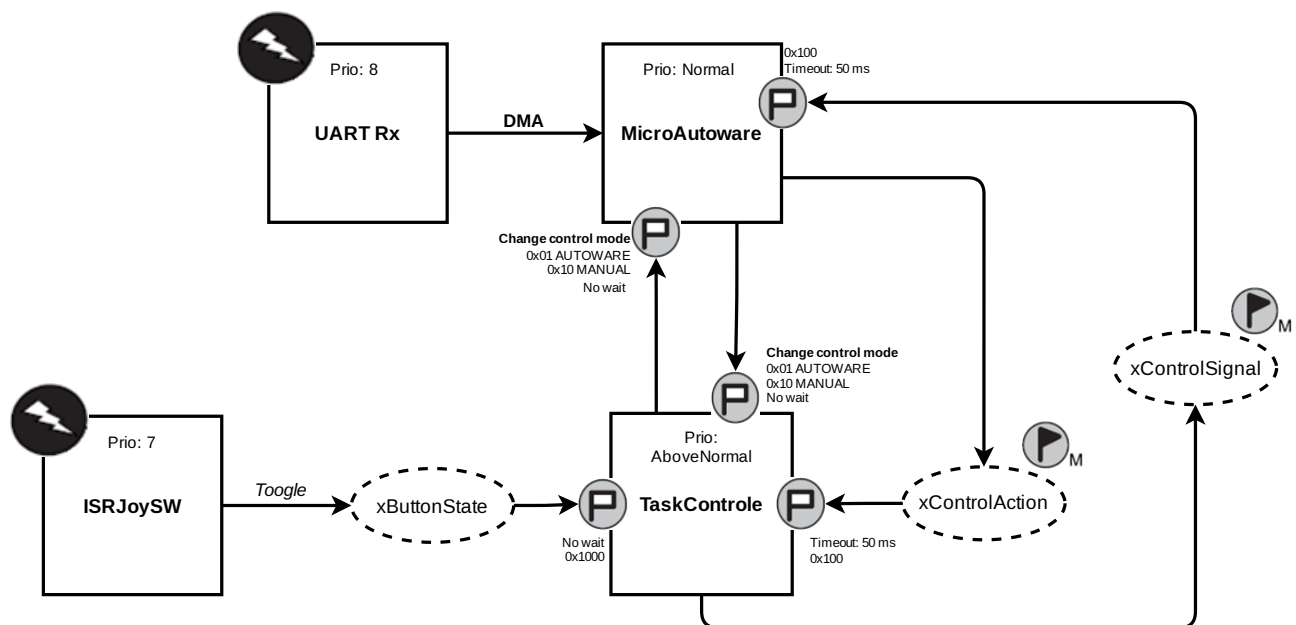


Figura 13: Diagrama do sistema.

Nome	MicroAutoware
Prioridade	Normal
Tamanho da stack	3500 kB
Detalhes	Leitura dos <i>subscribers</i> Autoware, leitura dos <i>subscribers</i> CARLA, envio das informações de controle e modo de operação para a TaskControle, recebimentos das informações de controle da TaskControle, escrita dos <i>publishers</i> Autoware, escrita dos <i>publishers</i> CARLA.

Tabela 2: Especificação da tarefa MicroAutoware.

Nome	TaskControle
Prioridade	AboveNormal
Tamanho da stack	500 kB
Detalhes	Realiza o controle do veículo utilizando a referência dada pelo <i>joystick</i> ou pelo Autoware, dado o modo de operação, podendo ser MANUAL ou AUTOWARE, respectivamente. A alteração do modo é feita por <i>ThreadFlag</i> , gerada por ISR ou pelo Autoware. Em caso do modo de operação AUTOWARE, os sinais de controle são recebidos por variável global e sincronizados por <i>ThreadFlag</i> , com tempo de 30 ms, onde caso não receba, entra em algum modo de segurança. Em caso de operação MANUAL, o <i>joystick</i> é lido por DMA, convertendo os valores analógicos em sinais de controle, onde também caso haja algum erro, o modo de emergência é acionado. O sinal de controle é enviado para o carro por meio de uma variável global, que é enviada para o MicroAutoware e sincronizado por <i>ThreadFlag</i> .

Tabela 3: Especificação da tarefa TaskControle.

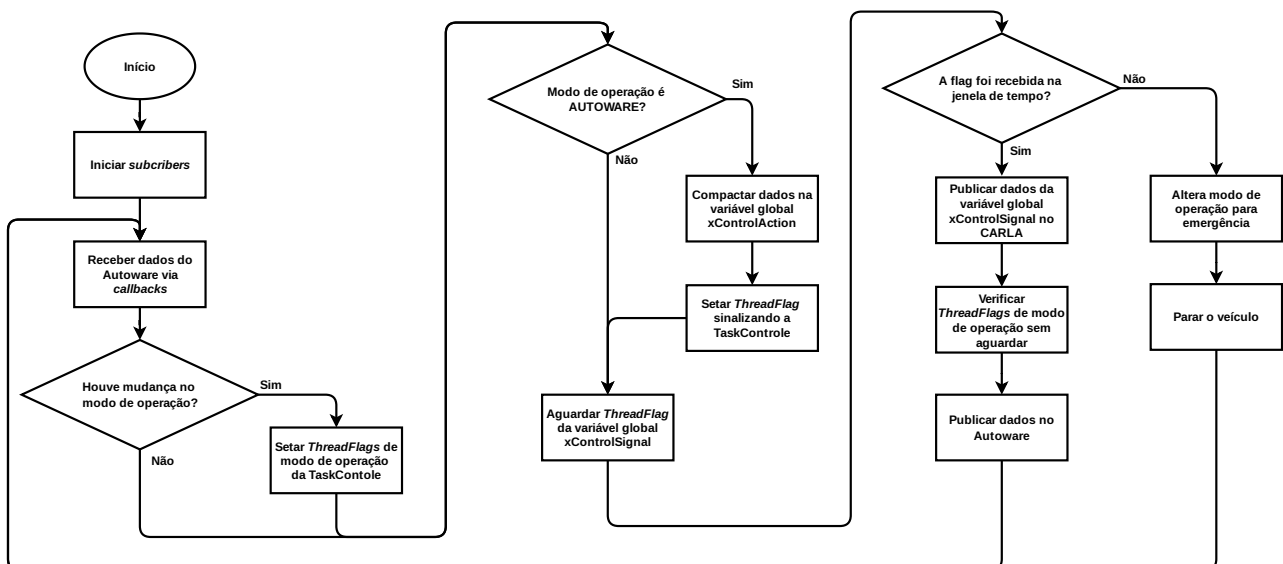


Figura 14: Fluxograma da tarefa MicroAutoware.

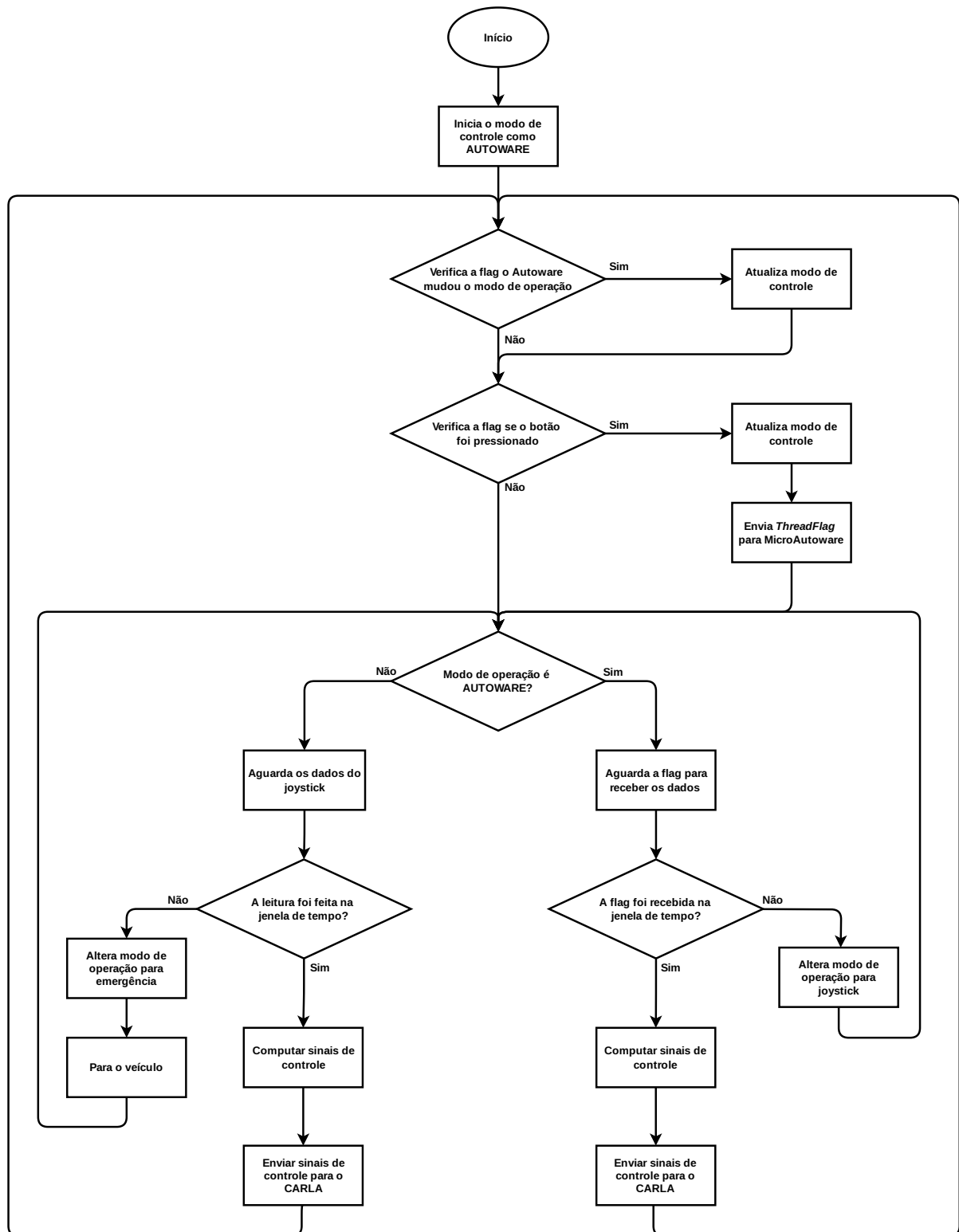


Figura 15: Fluxograma da tarefa TaskControle.

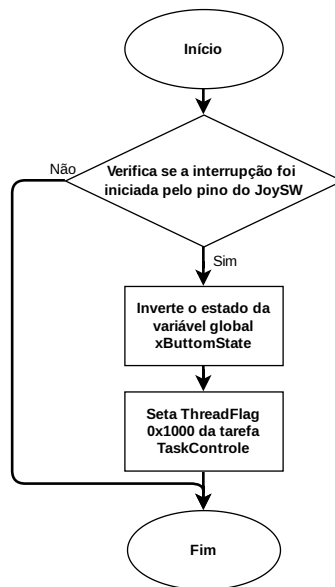


Figura 16: Fluxograma da ISR JoySW.

Sinalização xButtonState

- **Objeto:** *ThreadFlag*
- **Flag:** 0x1000
- **Modo:** *No wait*
- **Descrição:** Sinaliza ocorrência da interrupção do botão JoySW.

Sinalização xControlAction

- **Objeto:** *ThreadFlag*
- **Flag:** 0x0100
- **Modo:** *Timeout 30 ms*
- **Descrição:** Sinaliza o recebimento de dados pela variável global xControlAction.

Sinalização xControlSignal

- **Objeto:** *ThreadFlag*
- **Flag:** 0x0100
- **Modo:** *Timeout 30 ms*
- **Descrição:** Sinaliza o recebimento de dados pela variável global xControlSignal.

Alteração do modo de condução por interrupção JoySW

- **Objeto:** *ThreadFlag*
- **Flags:**
 - Modo de controle alterado para AUTOWARE: 0x01
 - Modo de controle alterado para MANUAL: 0x10
- **Modo:** *No wait*
- **Descrição:** Realiza a sincronização do modo de operação da tarefa TaskControle para a MicroAutoware.

Alteração do modo de condução pelo Autoware

- **Objeto:** *ThreadFlag*
- **Flags:**
 - Modo de controle alterado para AUTOWARE: 0x01
 - Modo de controle alterado para MANUAL: 0x10
- **Modo:** *No wait*
- **Descrição:** Realiza a sincronização do modo de operação da tarefa MicroAutoware para a TaskControle.

Proteção de recursos

Variável global `xControlSignal`

- Protegida por MUTEX.
 - `MutexControlSignal`

Variável global `xControlAction`

- Protegida por MUTEX.
 - `MutexControlAction`

Padronização de código

Padronização de código ROS

- Subscriber: `nome_subscriber_sub_`
- Publisher: `nome_subscriber_pub_`
- Mensagem: `nome_mensagem_msg_`
- Node: `nome_do_node`
- Callback: `nome_do_topico_callback`

4 Manual de utilização

5 Problemas identificados e não resolvidos

6 Códigos da comunidade

Referências

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>>.

The Autoware Foundation. *Architecture overview*. 2023. Disponível em: <<https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture/>>.

Apêndices