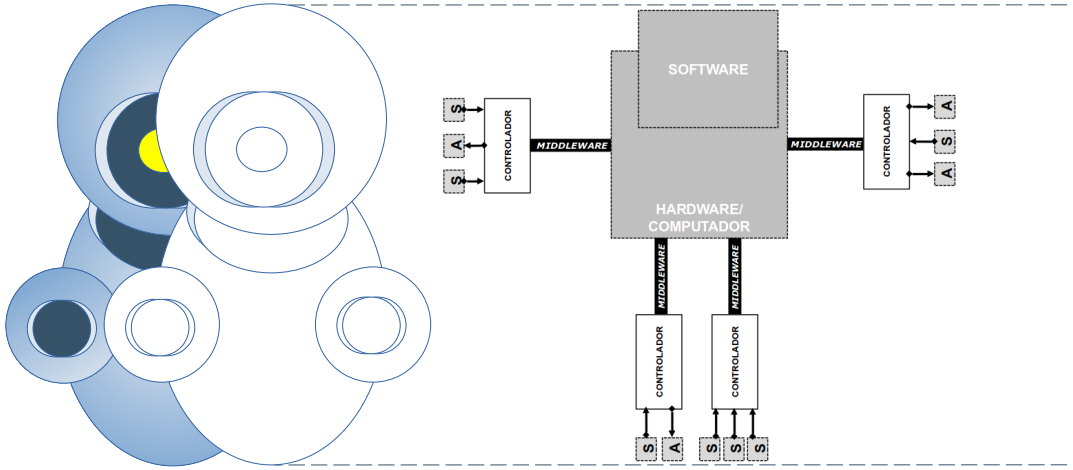
Introduction to Distributed and Embedded Multi-agent Systems

Bruno Policarpo Toledo Freitas
Carlos Eduardo Pantoja[1]
Nilson Mori Lazarin[1,2]

1. Centro Federal de Educação Tecnológica (CEFET/RJ) - 2. Universidade Federal Fluminense (UFF), Brasil

02/2024

EMBEDDED MULTI-AGENT



Robot

É um agente físico que possui:

Robot

É um agente físico que possui:

- **Componentes:** sensores e atuadores;

Robot

É um agente físico que possui:

- **Componentes:** sensores e atuadores;
- **Hardware:** controladores, plataformas e placas;

Robot

É um agente físico que possui:

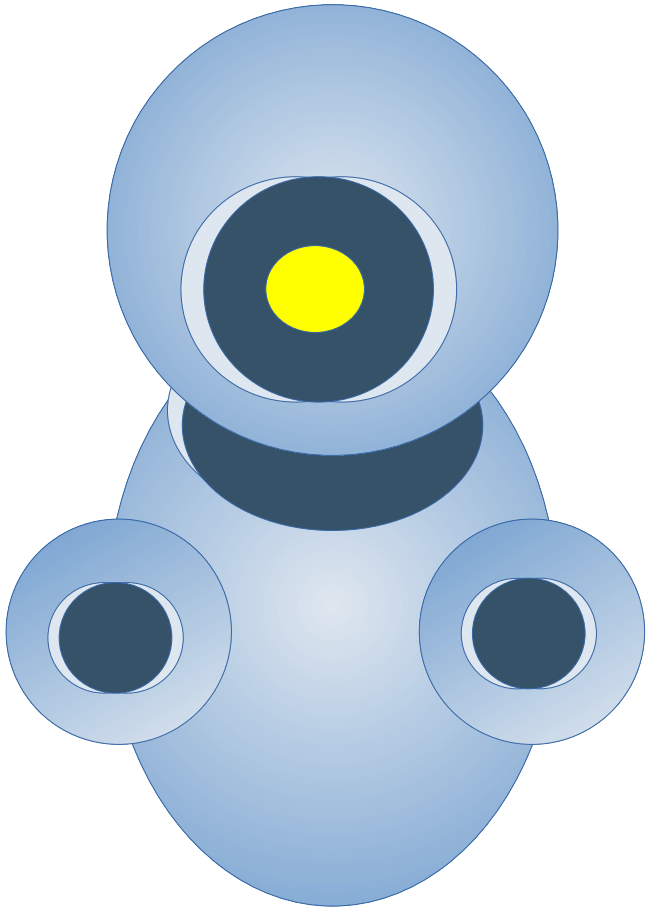
- **Componentes:** sensores e atuadores;
- **Hardware:** controladores, plataformas e placas;
- **Middleware:** para comunicação e controle de hardware;

Robot

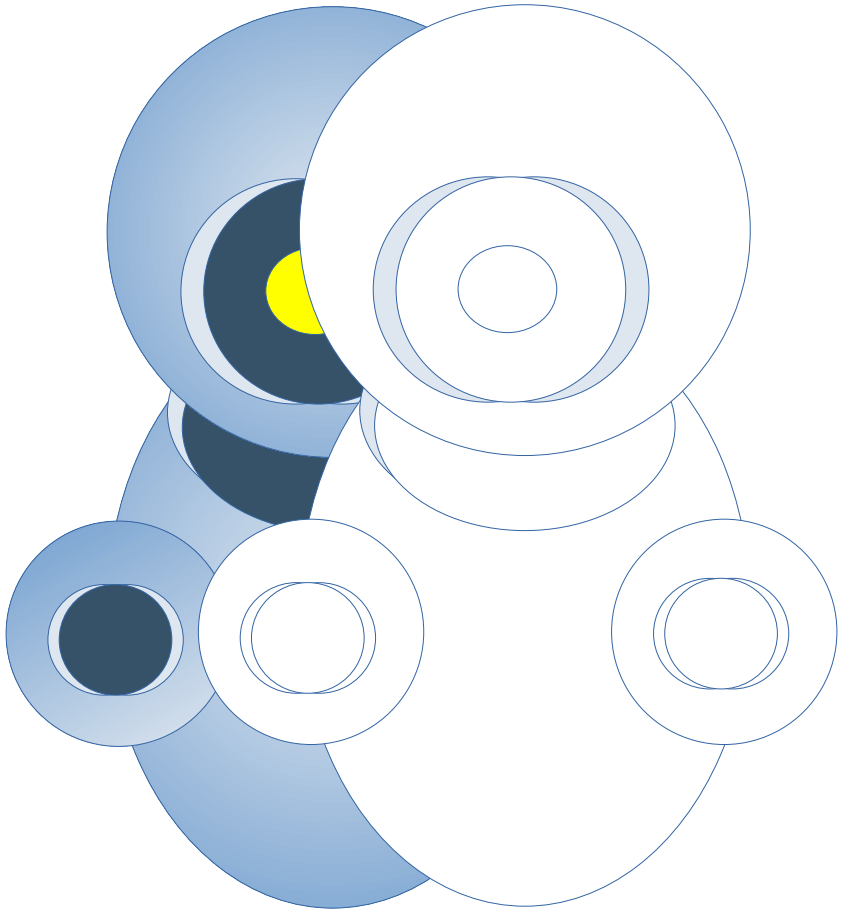
É um agente físico que possui:

- **Componentes:** sensores e atuadores;
- **Hardware:** controladores, plataformas e placas;
- **Middleware:** para comunicação e controle de hardware;
- **Software:** um sistema que realiza o raciocínio.

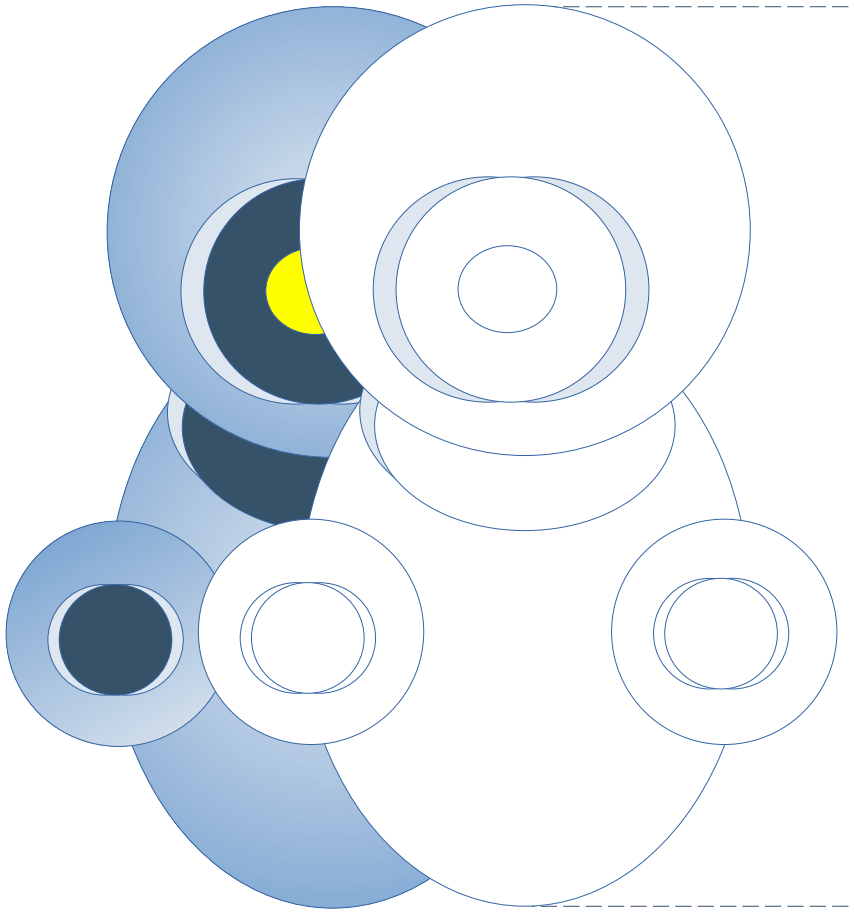
Robot

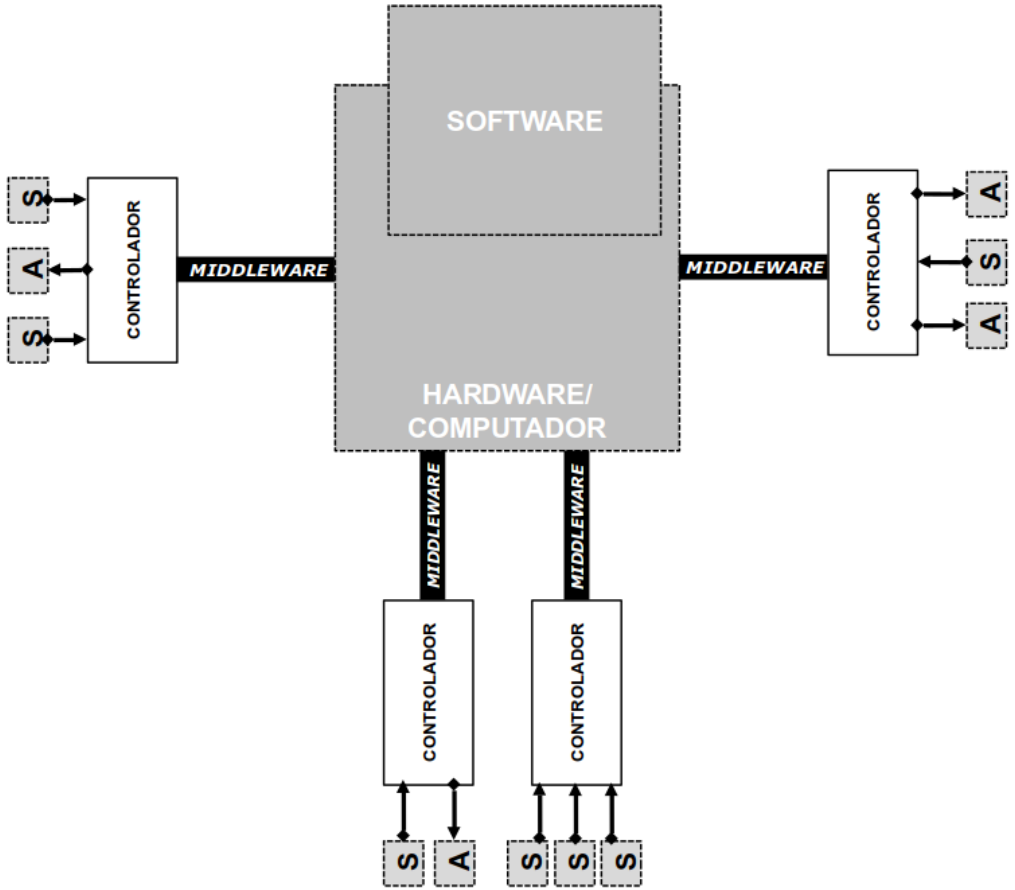


Robot



Robot




SOFTWARE
HARDWARE/
COMPUTADOR
MIDDLEWARE
MIDDLEWARE
MIDDLEWARE
MIDDLEWARE
CONTROLADOR
CONTROLADOR
CONTROLADOR
CONTROLADOR
S
A
S
A
S
S
A
S
S
S

Agente Incorporado e Embarcado

- **Agente Incorporado**. É um agente único embarcado em um dispositivo físico, que possui controle sobre um corpo físico bem definido (Rickel e Johnson, 2000);

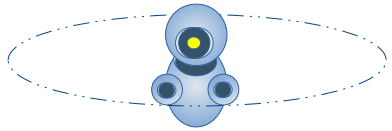
Agente Incorporado e Embarcado

- **Agente Incorporado**. É um agente único embarcado em um dispositivo físico, que possui controle sobre um corpo físico bem definido (Rickel e Johnson, 2000);

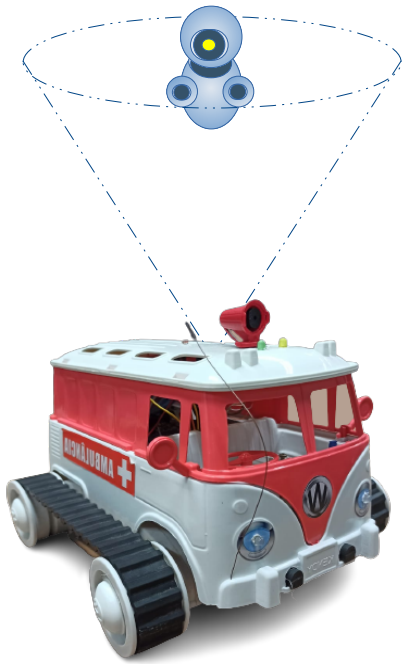
- **Agente Embarcado**. Um agente embarcado executando em dispositivos eletrônicos

Agente Incorporado e Embarcado

Agente Incorporado e Embarcado

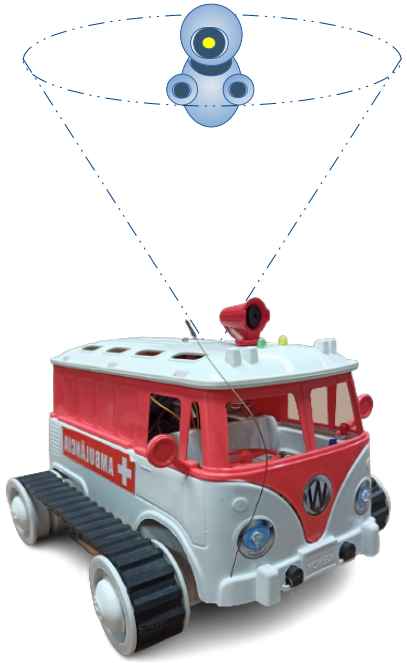


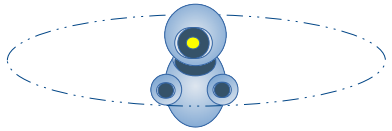
Agente Incorporado e Embarcado



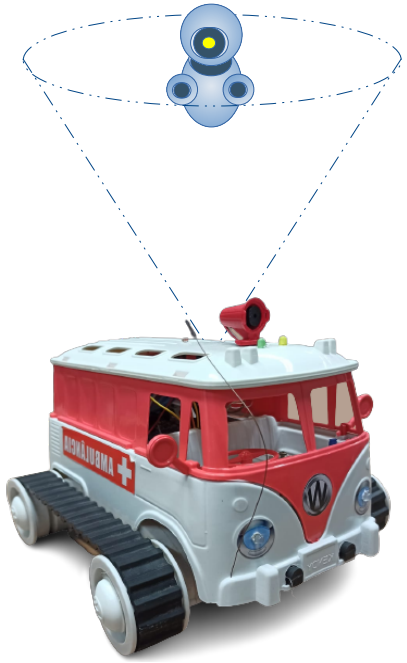
Agente Incorporado e Embarcado

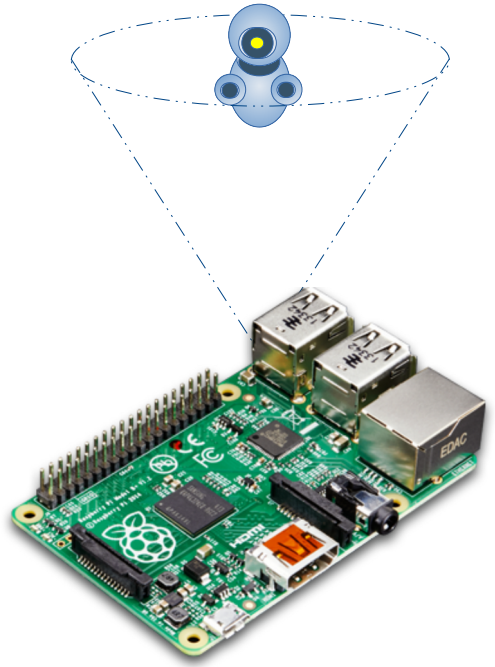
Agente Incorporado e Embarcado



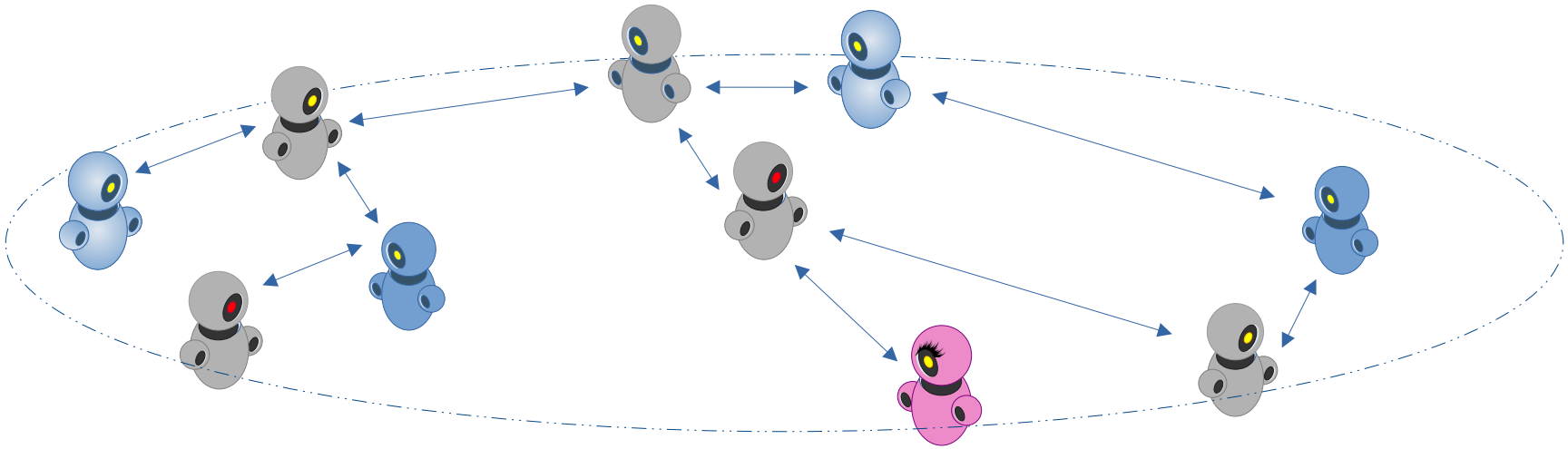


Agente Incorporado e Embarcado

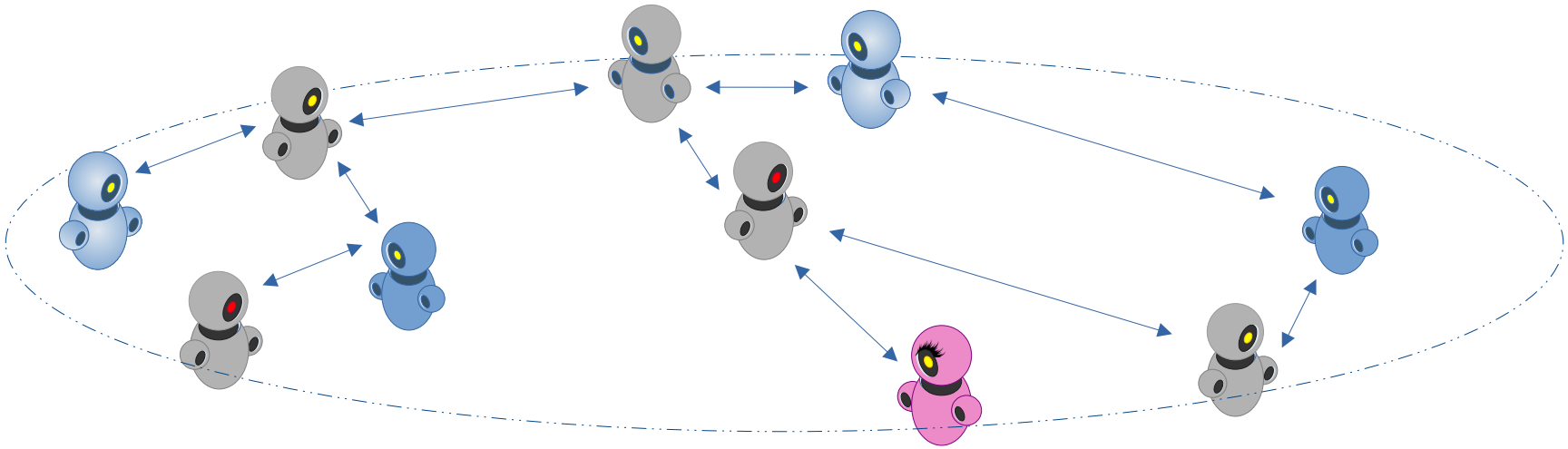




Abordagem Centralizada

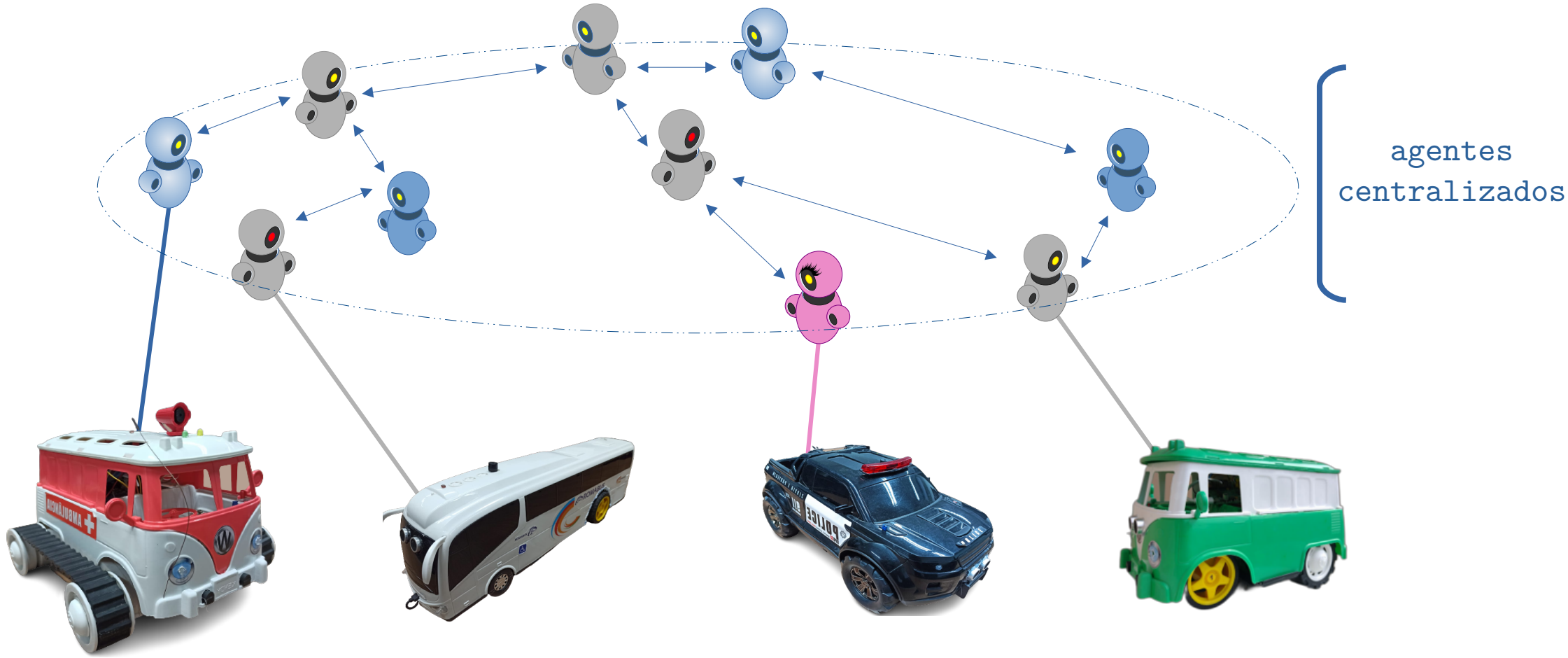


Abordagem Centralizada

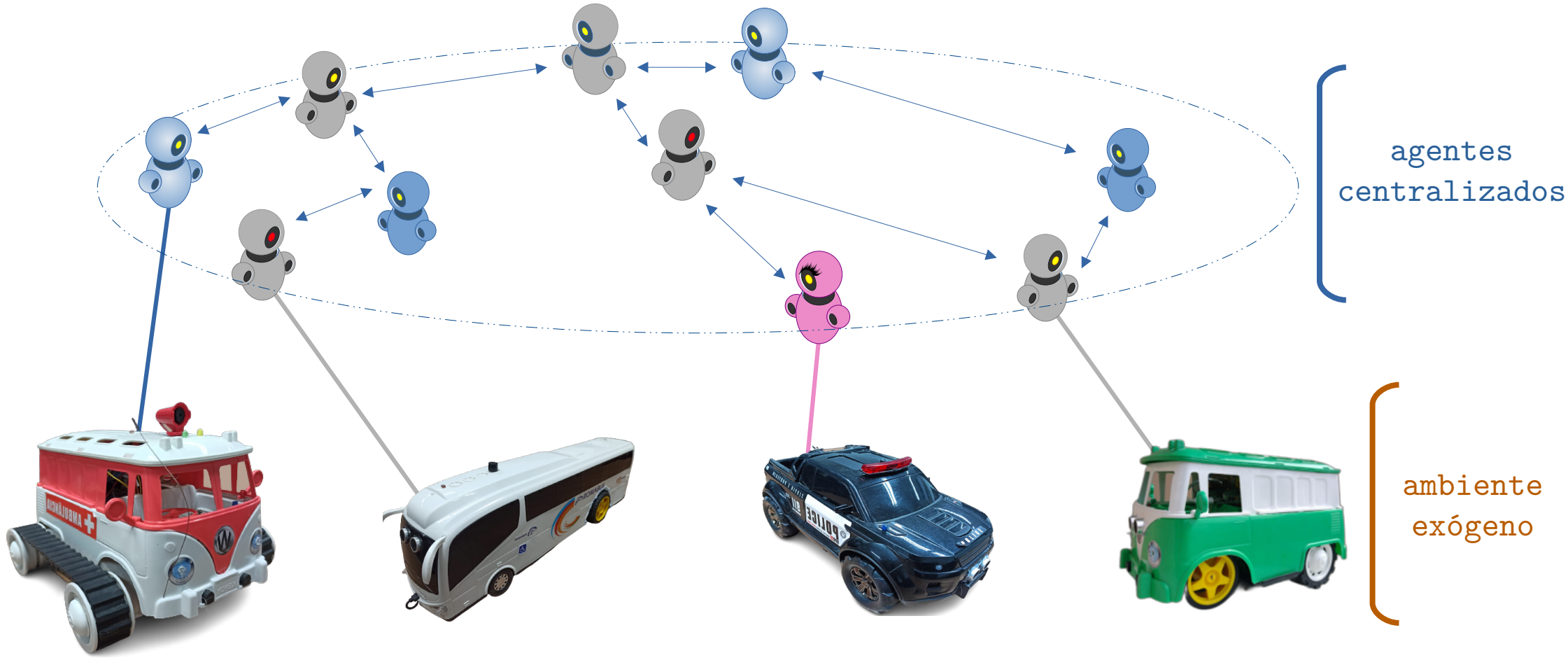


agentes centralizados

Abordagem Centralizada



Abordagem Centralizada



Embodied or embedded agents: Conventional approaches

T. LEPPÄNEM et al., Mobile Agents for Integration of Internet of Things and Wireless Sensor Networks. 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 2013, pp. 14-2.
C. SAVAGLIO, G. FORTINO and M. ZHOU, Towards interoperable, cognitive and autonomic IoT systems: An agent-based approach. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA.
M. E. PÉREZ HERNÁNDEZ and S. REIFF-MARGANIEC, Towards a Software Framework for the Autonomous Internet of Things. 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, Austria, 2016, pp. 220-227
HERINGER, V. H.; BARROS, R. S.; PANTOJA, C. E.; MACHADO, L.; LAZARIN, N. M. An Agent-oriented Ground Vehicle's Automation using Jason Framework. In : International Conference on Agents and Artificial Intelligence, 2014, ESEO. Proceedings of the 6th International Conference on Agents and Artificial Intelligence. p. 261-266.

Embodied or embedded agents: Conventional approaches

Centralised Solution

Embodied or embedded agents: Conventional approaches

Centralised Solution

- one agent p. device

Embodied or embedded agents: Conventional approaches

Centralised Solution

- one agent p. device
  - performance issues
    - only one agent into the device;
    - a lot of sensors.

Embodied or embedded agents: Conventional approaches

Centralised Solution

- one agent p. device
  - performance issues
    - only one agent into the device;
    - a lot of sensors.
- reactive artifact

Embodied or embedded agents: Conventional approaches

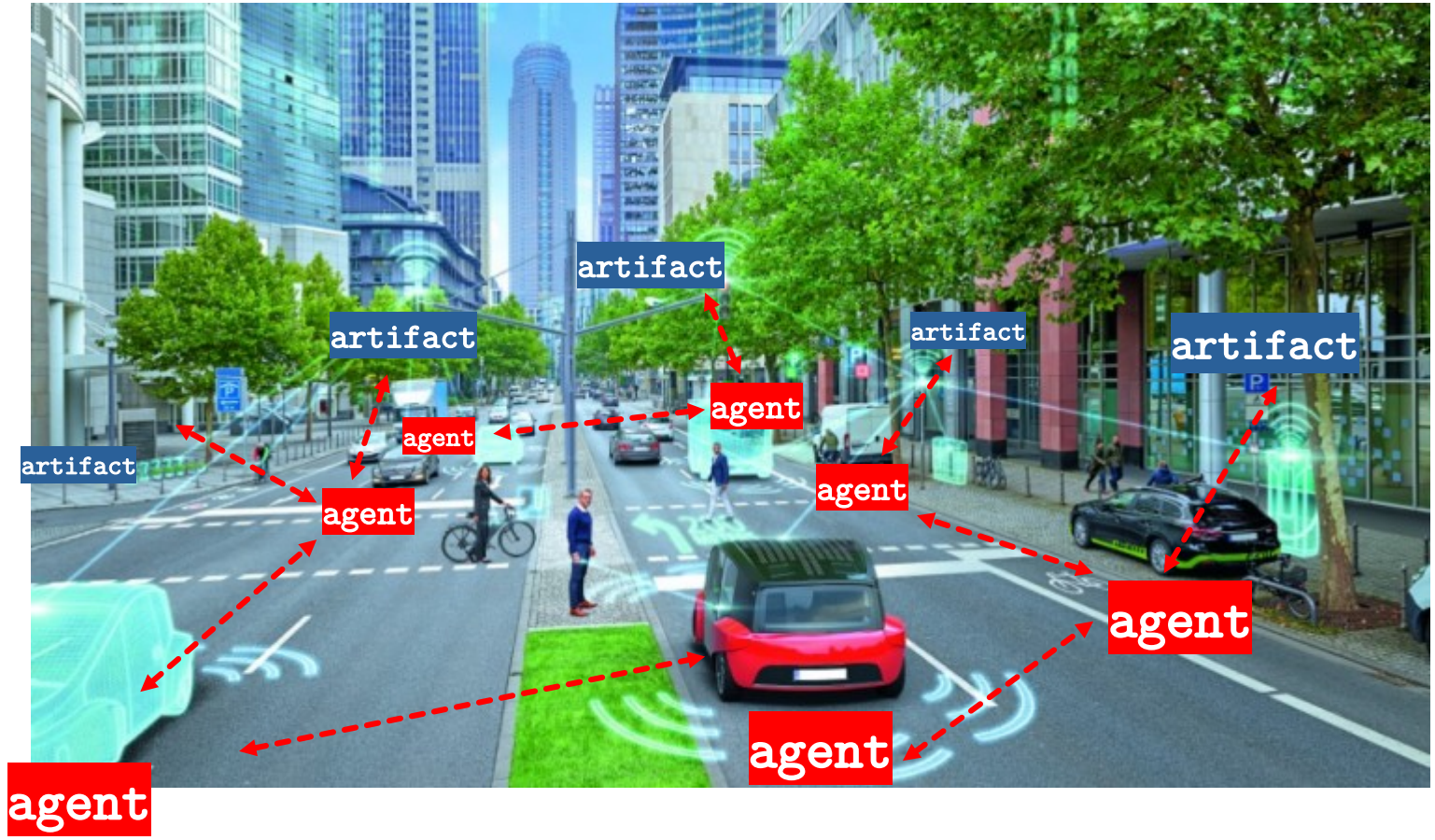
Centralised Solution

- one agent p. device
  - performance issues
    - only one agent into the device;
    - a lot of sensors.
- reactive artifact
  - are data oriented
  - don't have decision-making

Embodied or embedded agents: Conventional approaches

Centralised Solution

- one agent p. device
  - performance issues
    - only one agent into the device;
    - a lot of sensors.
- reactive artifact
  - are data oriented
  - don't have decision-making
- depends on a server



Sistemas Multiagentes Embarcados

Um SMA Embarcado é um sistema aberto de agentes móveis embarcados em um dispositivo responsável pela autonomia, pró-atividade, sociabilidade do dispositivo através do controle de atuadores, sensores e infraestruturas de comunicação.

BRANDÃO, FABIAN CESAR; LIMA, MARIA ALICE TRINTA; PANTOJA, CARLOS EDUARDO; ZAHN, JEAN; VITERBO, JOSÉ. Engineering Approaches for Programming Agent-Based IoT Objects Using the Resource Management Architecture. SENSORS, v. 21, p. 8110. Disponível em: https://doi.org/10.3390/s21238110.

Sistemas Multiagentes Embarcados

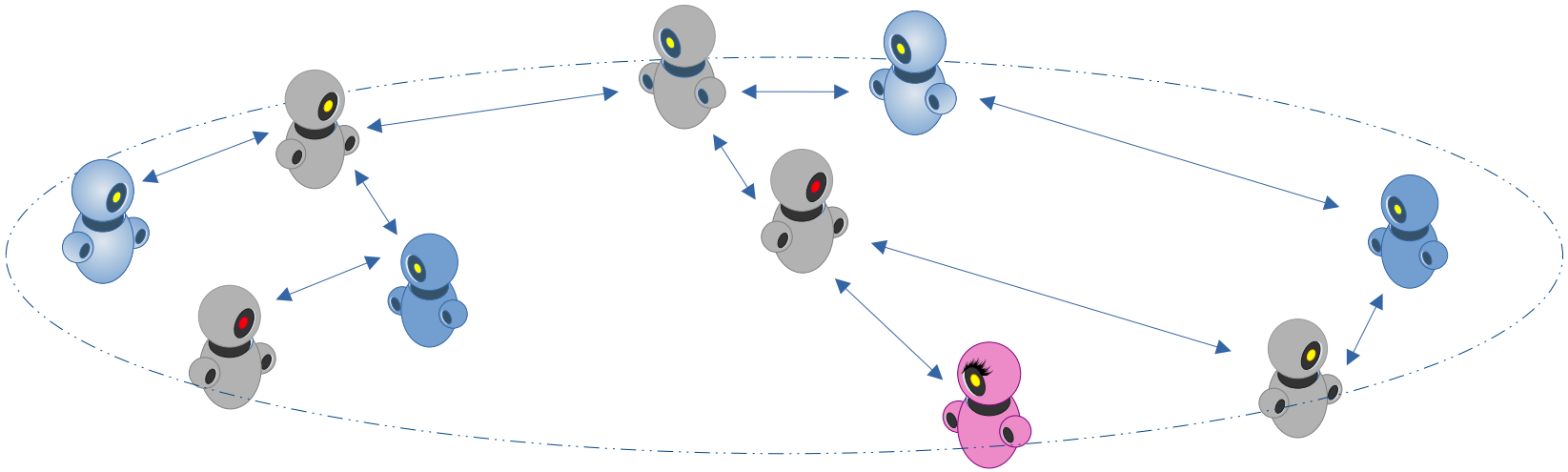
Um SMA Embarcado é um sistema aberto de agentes móveis embarcados em um dispositivo responsável pela autonomia, pró-atividade, sociabilidade do dispositivo através do controle de atuadores, sensores e infraestruturas de comunicação.

Não há controle remoto ou processamento externo.

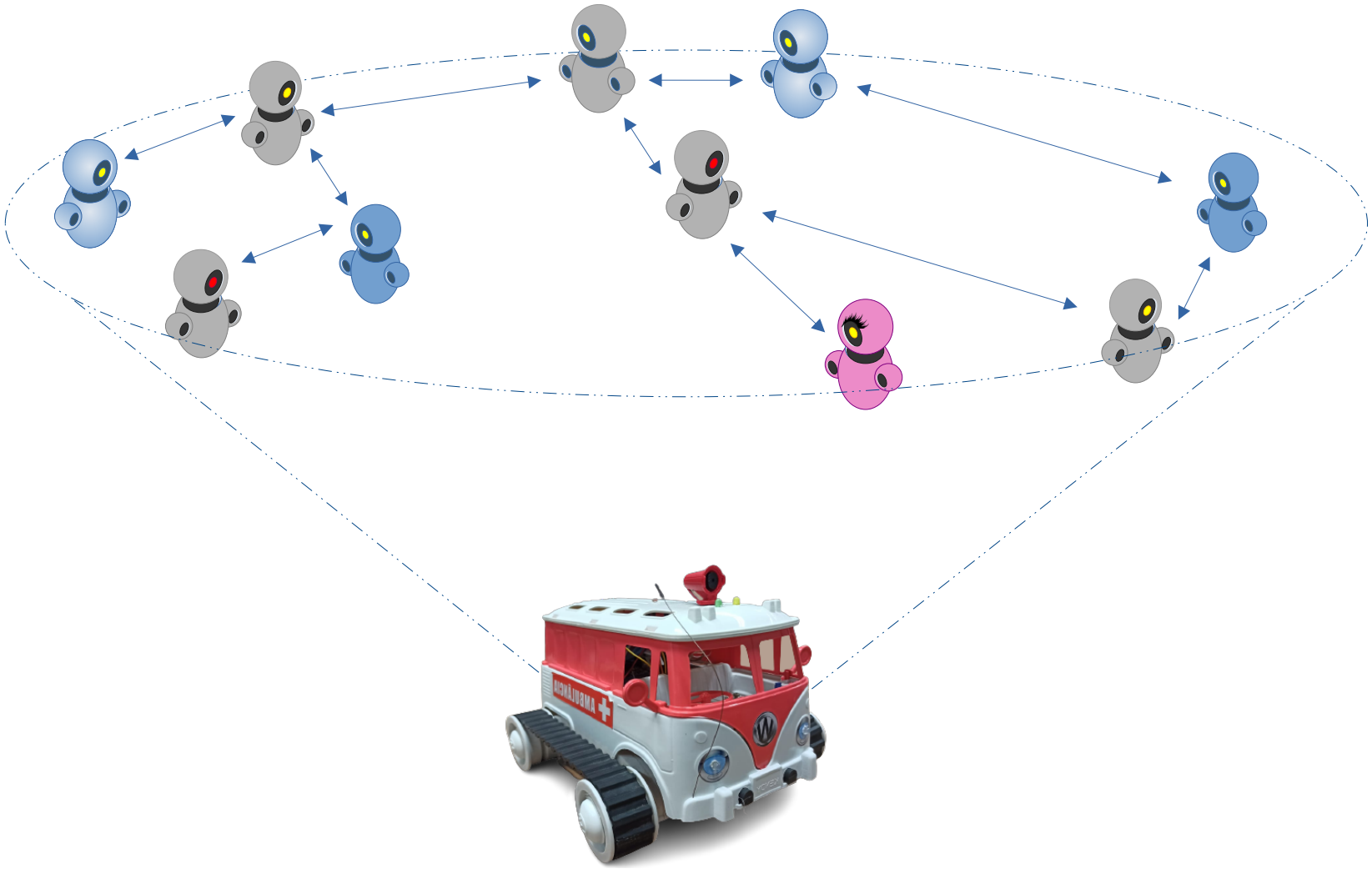
BRANDÃO, FABIAN CESAR; LIMA, MARIA ALICE TRINTA; PANTOJA, CARLOS EDUARDO; ZAHN, JEAN; VITERBO, JOSÉ. Engineering Approaches for Programming Agent-Based IoT Objects Using the Resource Management Architecture. SENSORS, v. 21, p. 8110. Disponível em: https://doi.org/10.3390/s21238110.

SMA Embarcado – Abordagem Distribuída

SMA Embarcado – Abordagem Distribuída



SMA Embarcado – Abordagem Distribuída



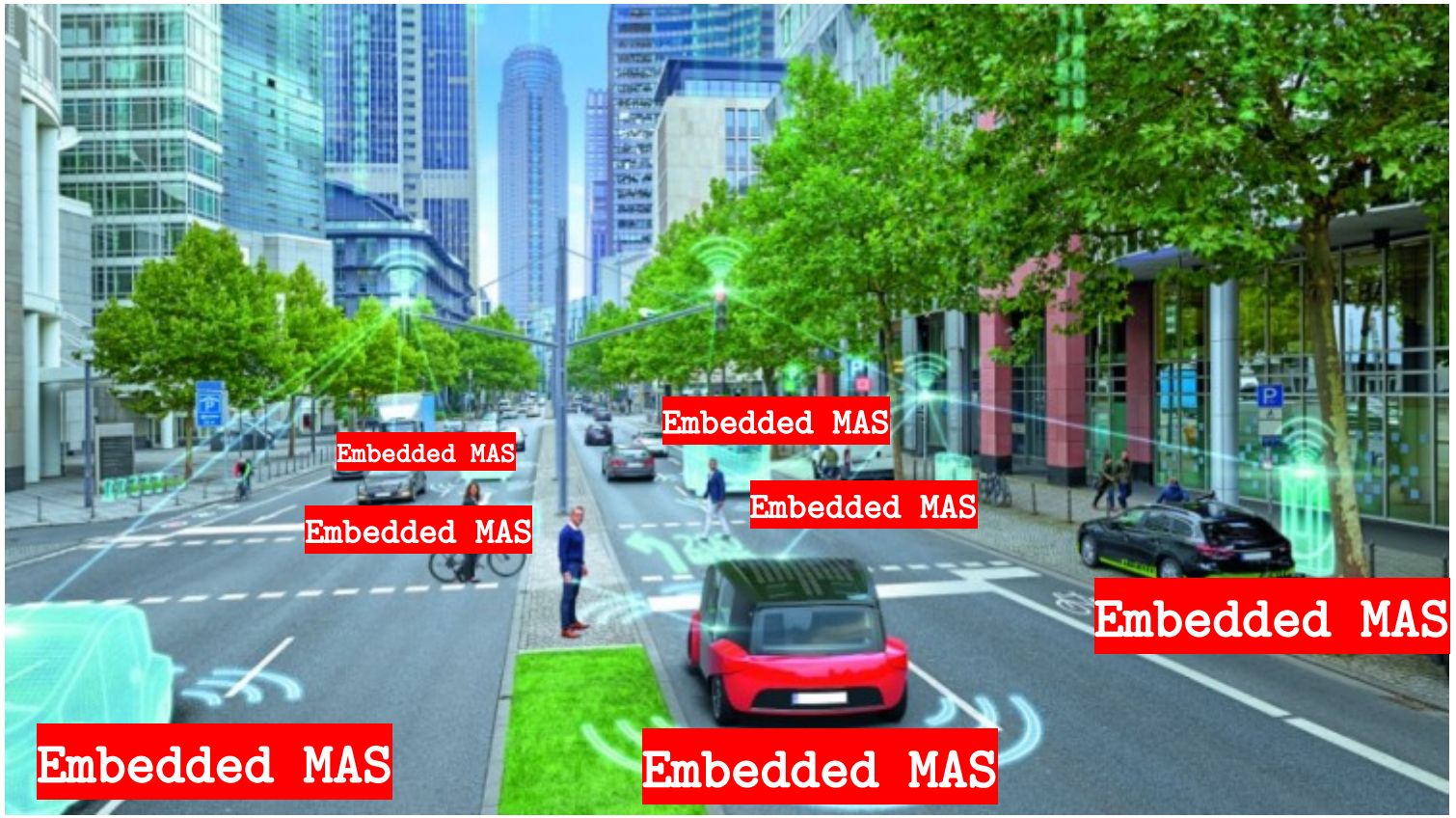
Our approach

Distributed Solution

Our approach

Distributed Solution

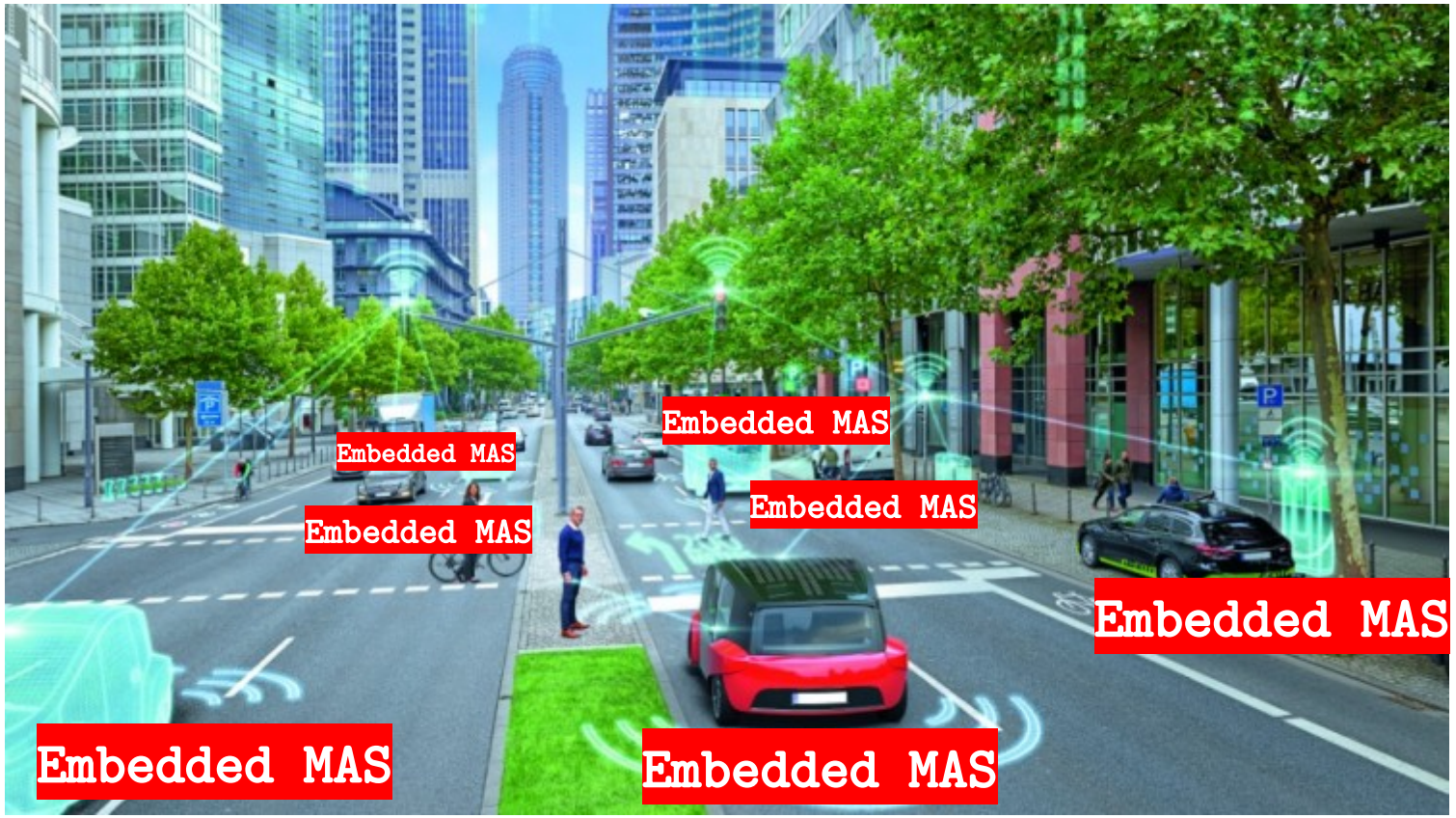
- one MAS p. device



Our approach

Distributed Solution

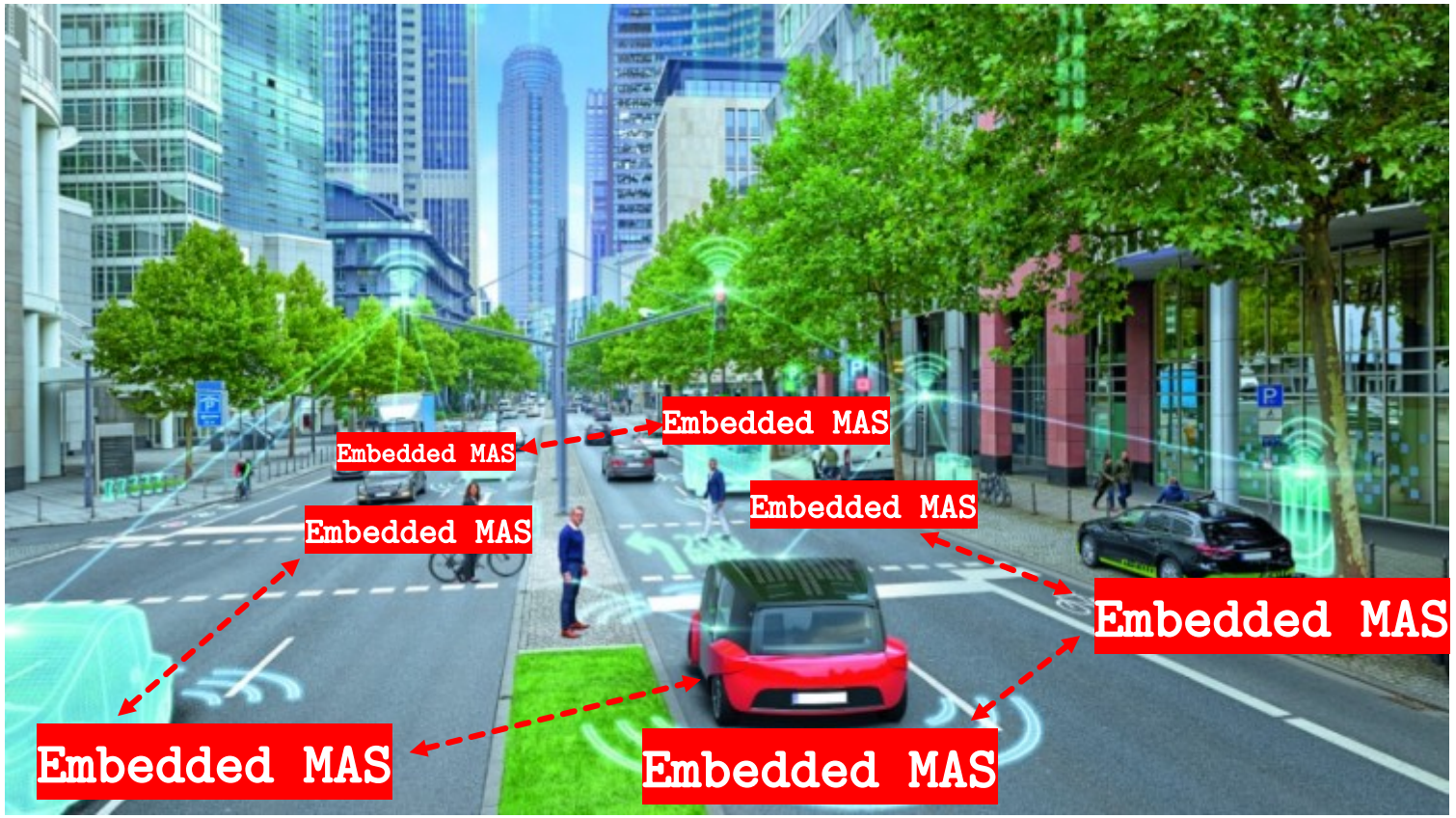
- one MAS p. device
- truly autonomy



Our approach

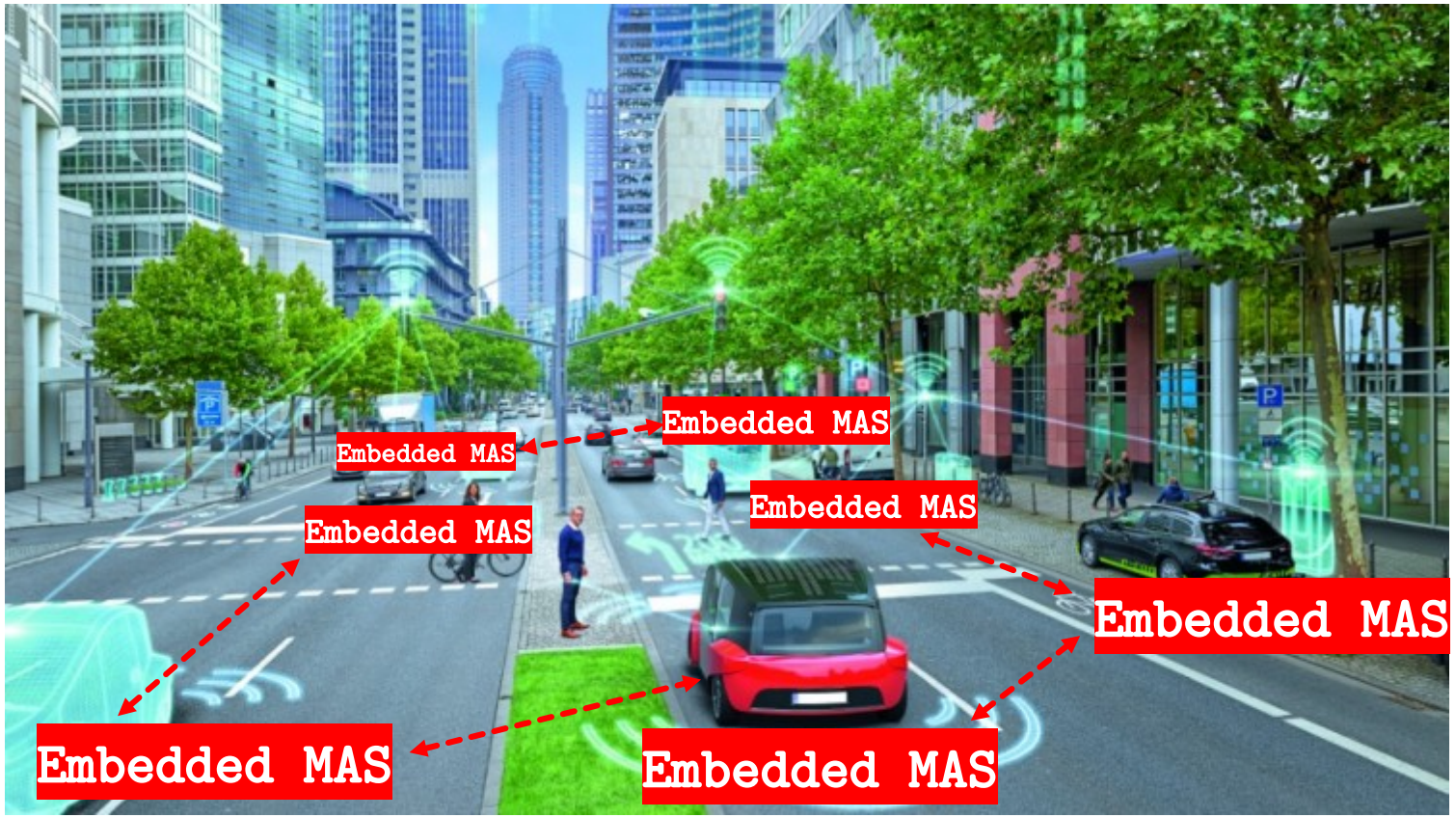
Distributed Solution

- one MAS p. device
- truly autonomy
  - communicability dependency only for external communication



Our approach

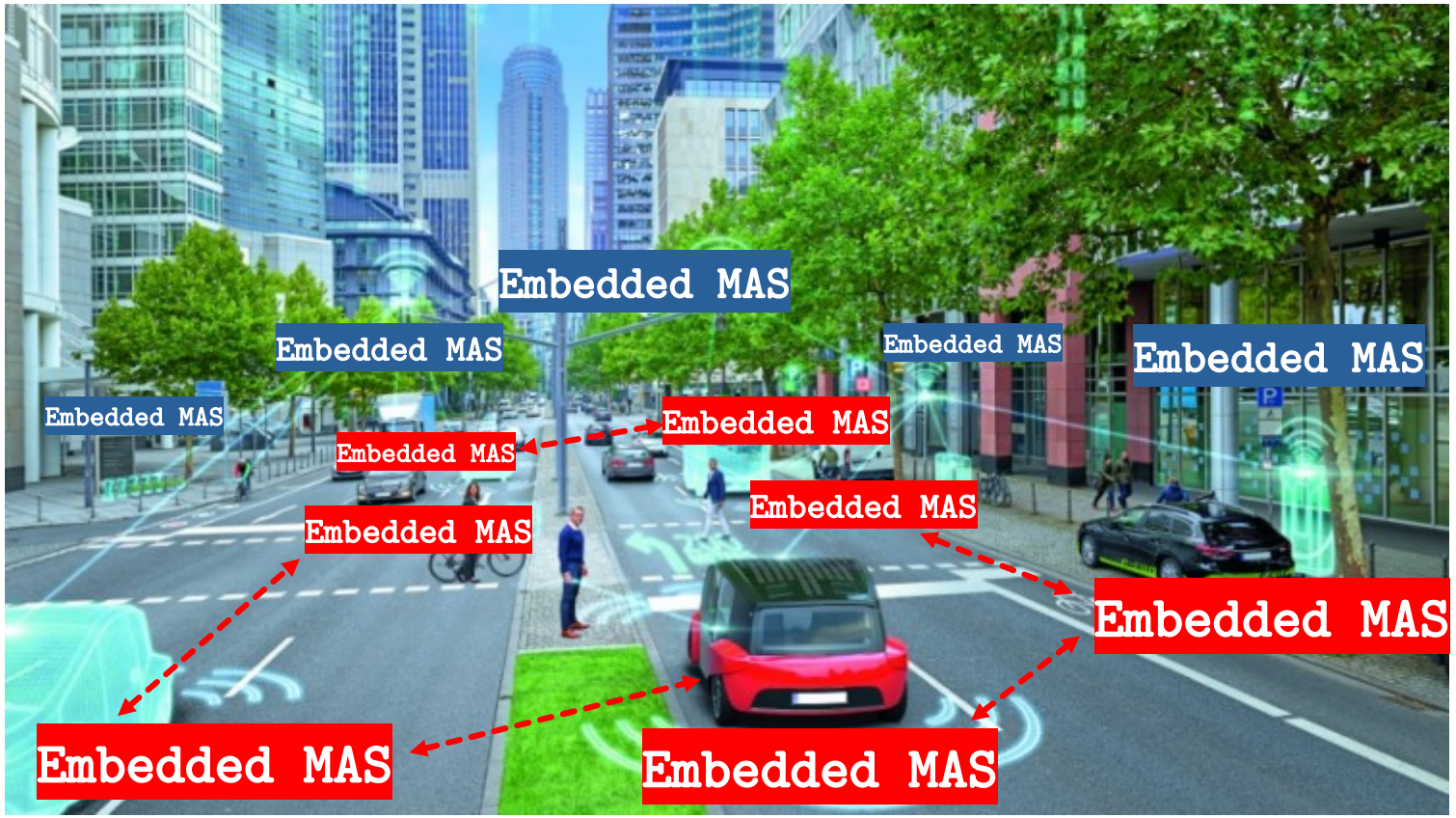
Edge Intelligence



Our approach

Edge Intelligence

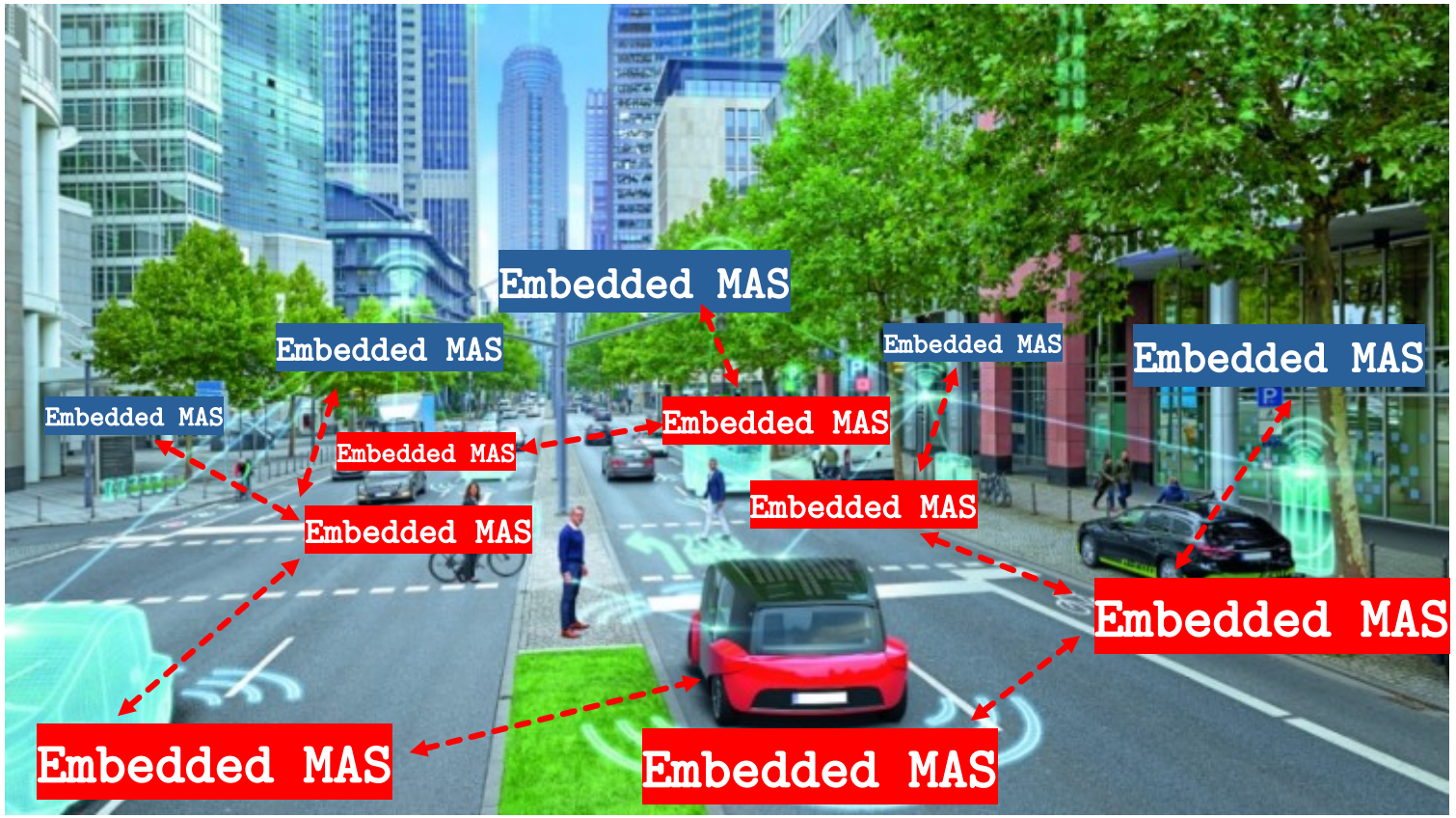
- one MAS p. artifact



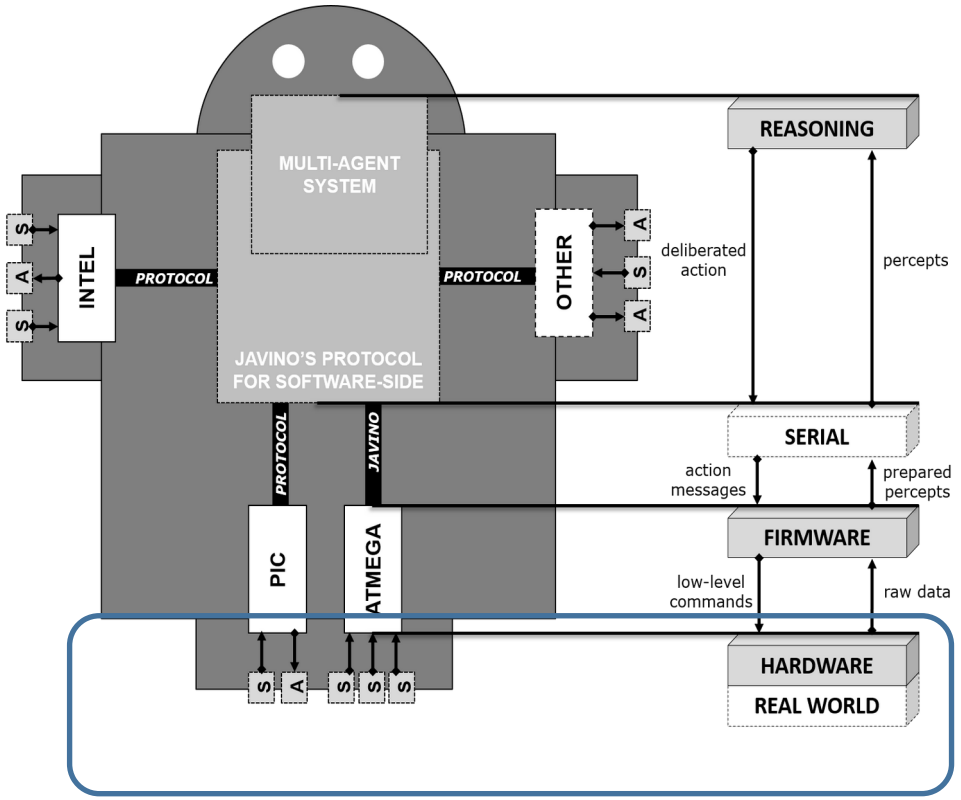
Our approach

Edge Intelligence

- one MAS p. artifact
  - pro-active artifact
  - decision-making in the edge



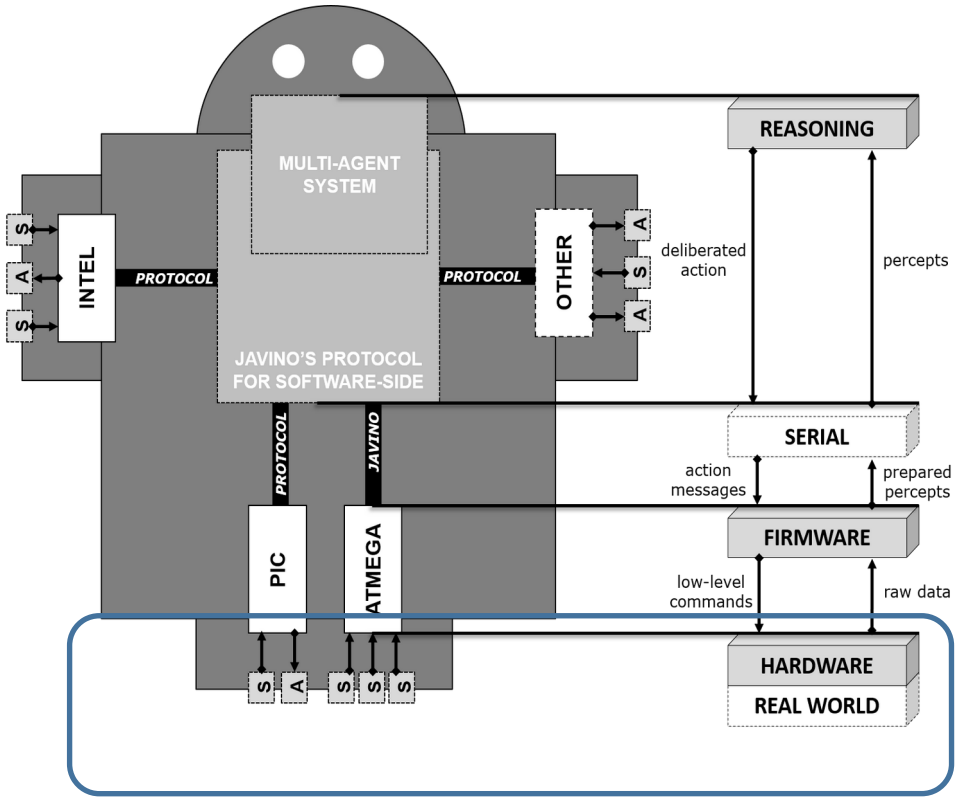
Arquitetura de 4 Camadas



PANTOJA, Carlos Eduardo; STABILE, Márcio Fernando; LAZARIN, Nilson Mori; SICHMAN, Jaime Simão. ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming. In: BALDONI, Matteo; MÜLLER, Jörg P.; NUNES, Ingrid; ZALILA-WENKSTERN, Rym (orgs.). Engineering Multi-Agent Systems. Lecture Notes in Computer Science. 10093. ed. Cham: Springer International Publishing, 2016. p. 136–155. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-50983-9_8.

Arquitetura de 4 Camadas

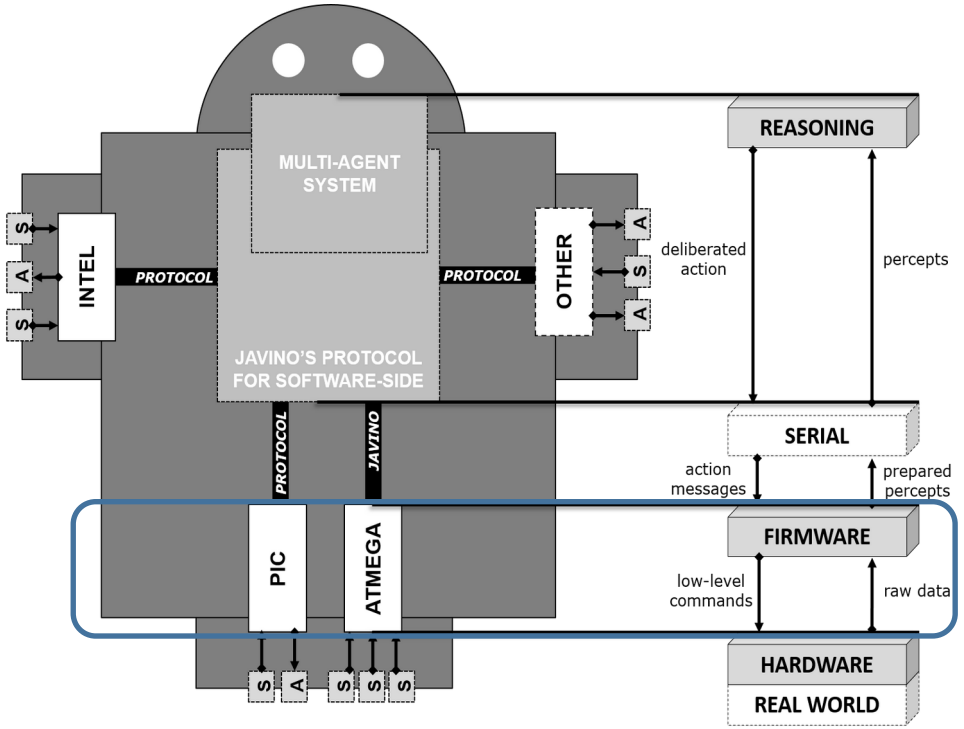
- **Hardware.** Conjunto de recursos que representam o ambiente do agente no mundo físico;



PANTOJA, Carlos Eduardo; STABILE, Márcio Fernando; LAZARIN, Nilson Mori; SICHMAN, Jaime Simão. ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming. In: BALDONI, Matteo; MÜLLER, Jörg P.; NUNES, Ingrid; ZALILA-WENKSTERN, Rym (orgs.). Engineering Multi-Agent Systems. Lecture Notes in Computer Science. 10093. ed. Cham: Springer International Publishing, 2016. p. 136–155. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-50983-9_8.

Arquitetura de 4 Camadas

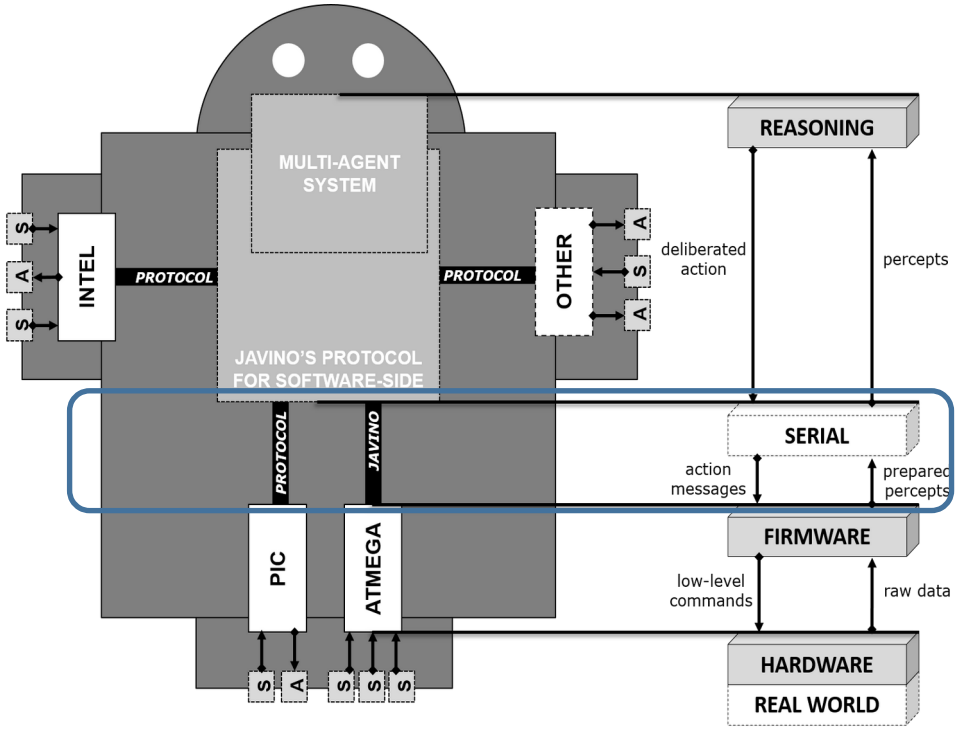
- **Hardware.** Conjunto de recursos que representam o ambiente do agente no mundo físico;
- **Firmware.** Hospedada em um ou mais microcontroladores que manipulam a camada de hardware, conforme as deliberações do agente;



PANTOJA, Carlos Eduardo; STABILE, Márcio Fernando; LAZARIN, Nilson Mori; SICHMAN, Jaime Simão. ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming. In: BALDONI, Matteo; MÜLLER, Jörg P.; NUNES, Ingrid; ZALILA-WENKSTERN, Rym (orgs.). Engineering Multi-Agent Systems. Lecture Notes in Computer Science. 10093. ed. Cham: Springer International Publishing, 2016. p. 136–155. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-50983-9_8.

Arquitetura de 4 Camadas

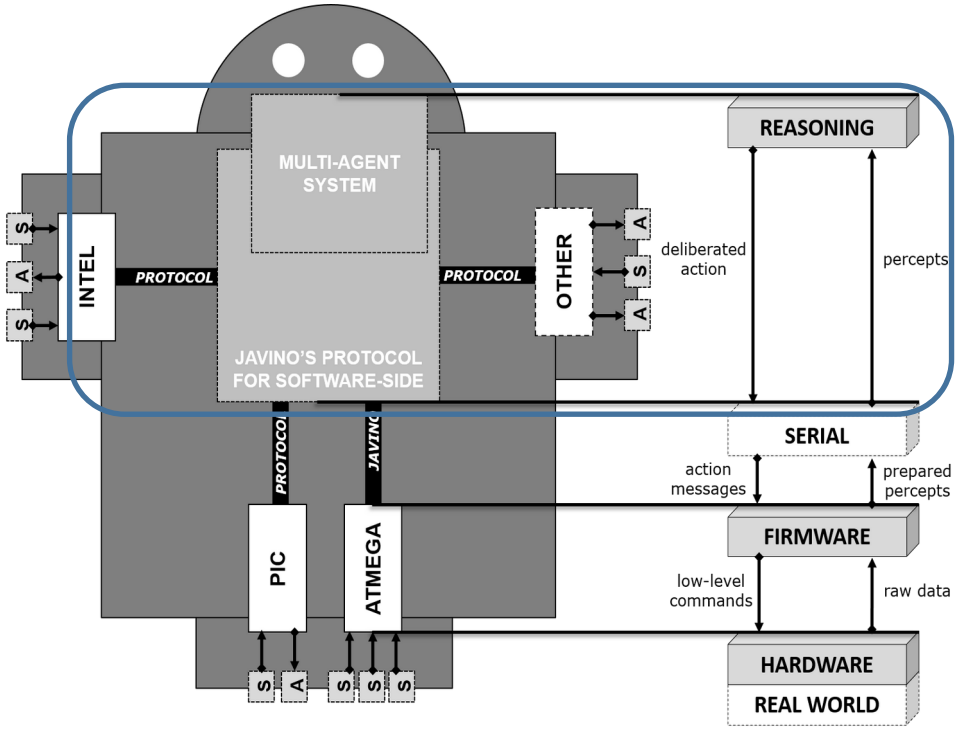
- **Hardware.** Conjunto de recursos que representam o ambiente do agente no mundo físico;
- **Firmware.** Hospedada em um ou mais microcontroladores que manipulam a camada de hardware, conforme as deliberações do agente;
- **Interfaceamento.** Permite a comunicação do agente com o microcontrolador;



PANTOJA, Carlos Eduardo; STABILE, Márcio Fernando; LAZARIN, Nilson Mori; SICHMAN, Jaime Simão. ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming. In: BALDONI, Matteo; MÜLLER, Jörg P.; NUNES, Ingrid; ZALILA-WENKSTERN, Rym (orgs.). Engineering Multi-Agent Systems. Lecture Notes in Computer Science. 10093. ed. Cham: Springer International Publishing, 2016. p. 136–155. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-50983-9_8.

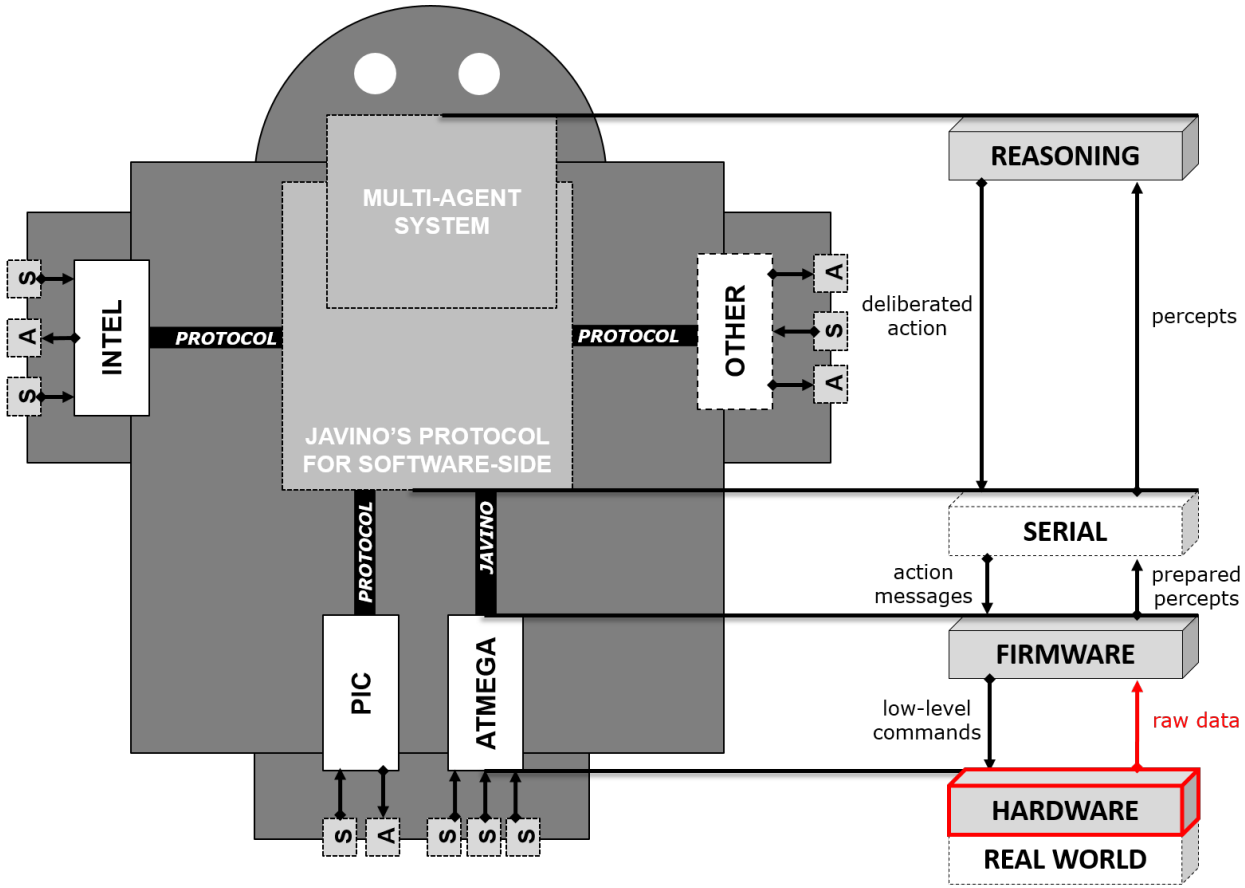
Arquitetura de 4 Camadas

- **Raciocínio.** É um SMA hospedado em um computador que executa o controle do dispositivo onde estiver embarcado.

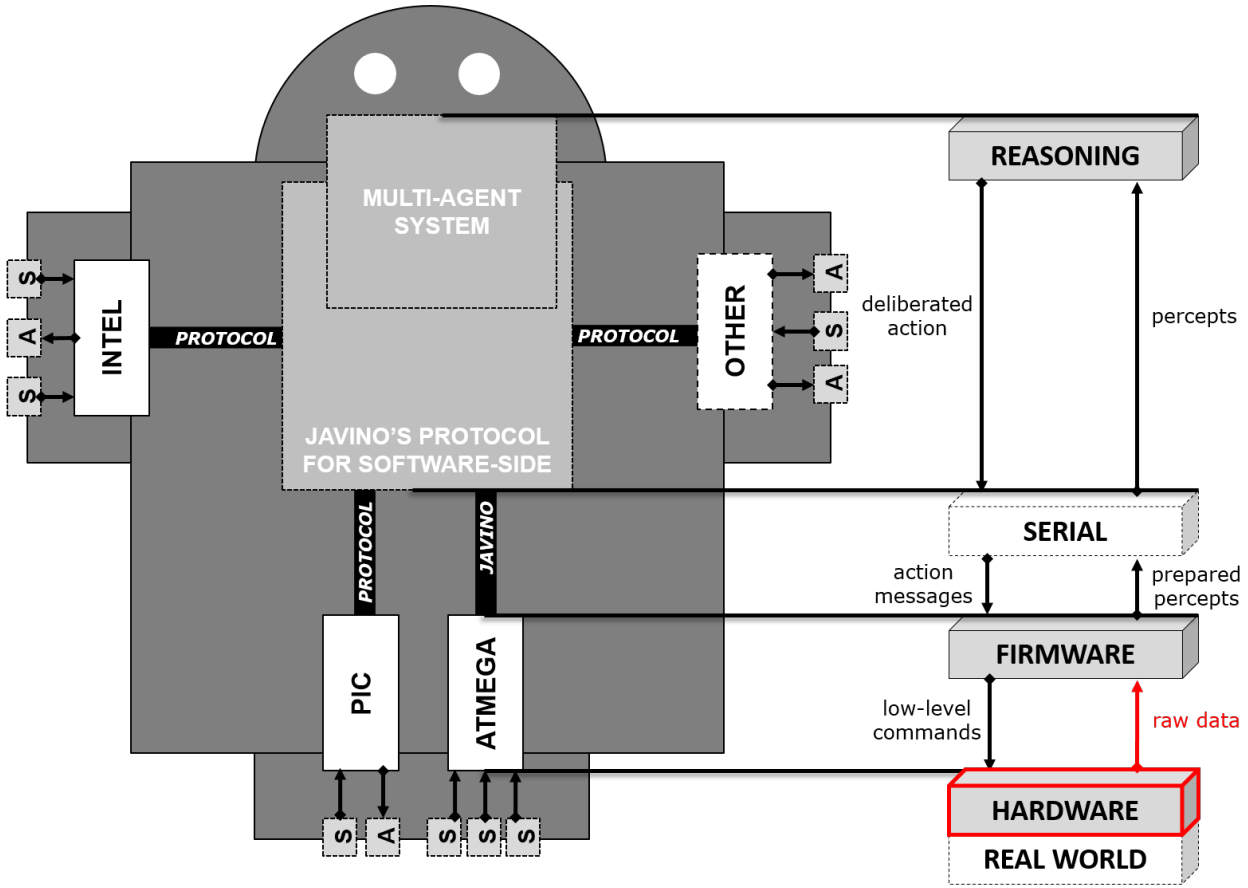


PANTOJA, Carlos Eduardo; STABILE, Márcio Fernando; LAZARIN, Nilson Mori; SICHMAN, Jaime Simão. ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming. In: BALDONI, Matteo; MÜLLER, Jörg P.; NUNES, Ingrid; ZALILA-WENKSTERN, Rym (orgs.). Engineering Multi-Agent Systems. Lecture Notes in Computer Science. 10093. ed. Cham: Springer International Publishing, 2016. p. 136–155. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-50983-9_8.

Arquitetura Física e Lógica

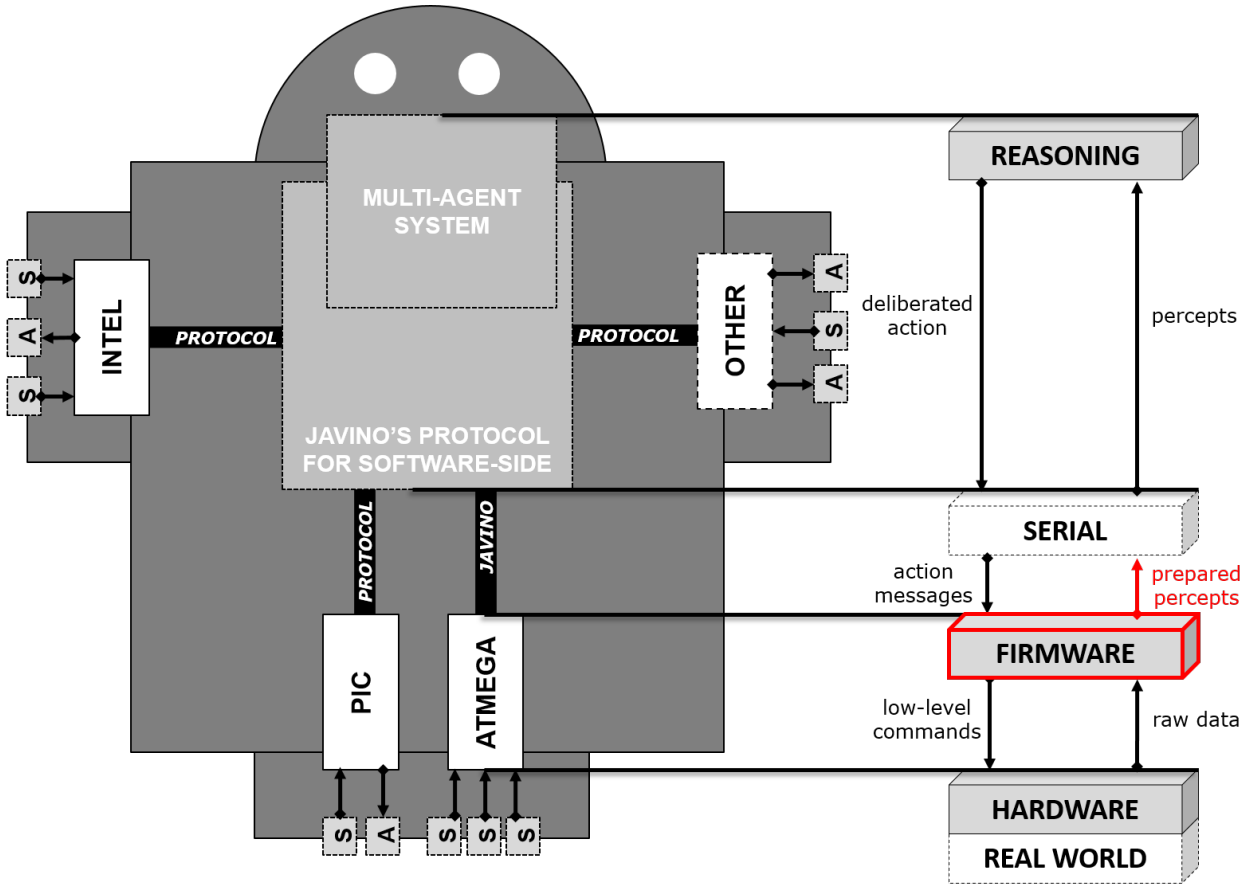


Arquitetura Física e Lógica



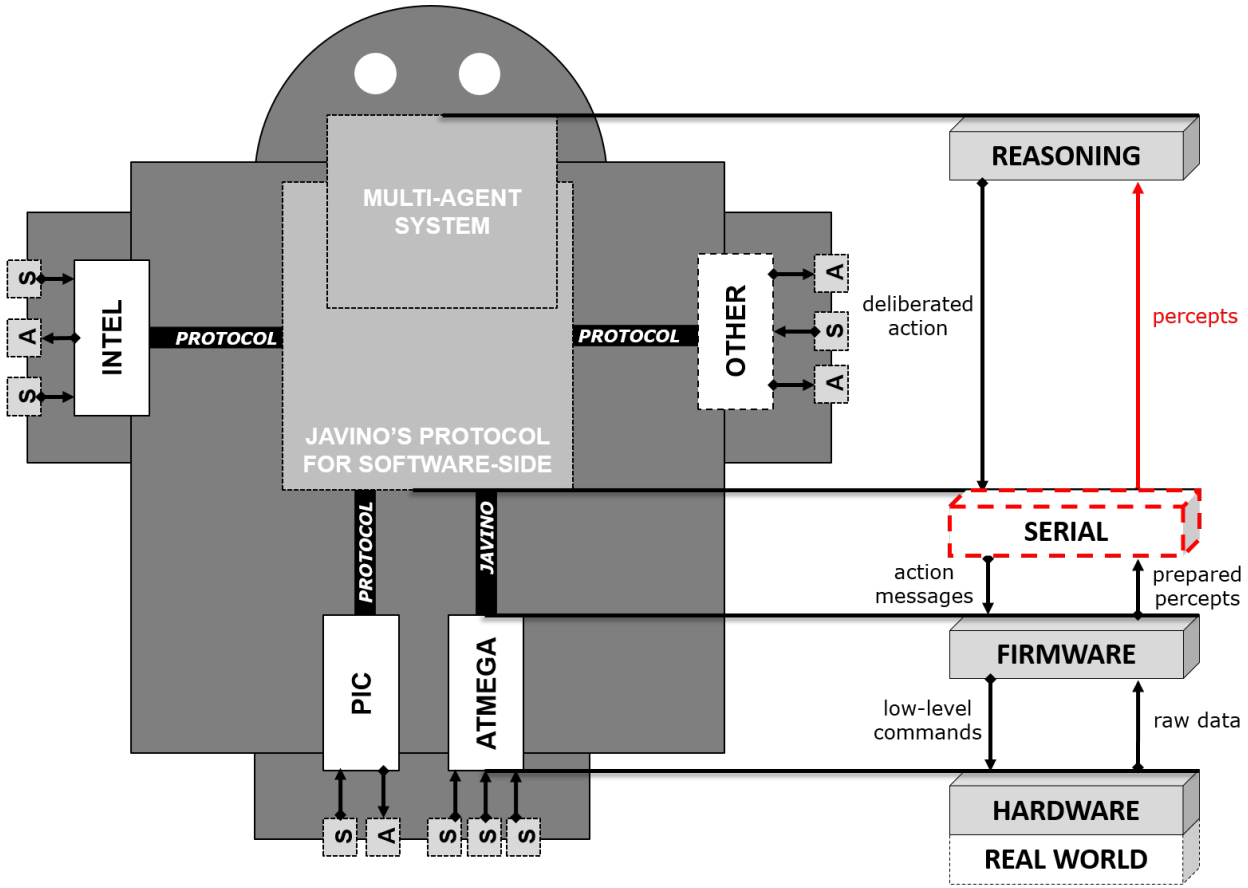
Sensores capturam dados brutos do mundo real e os enviam para um dos microcontroladores escolhido para o projeto.

Arquitetura Física e Lógica



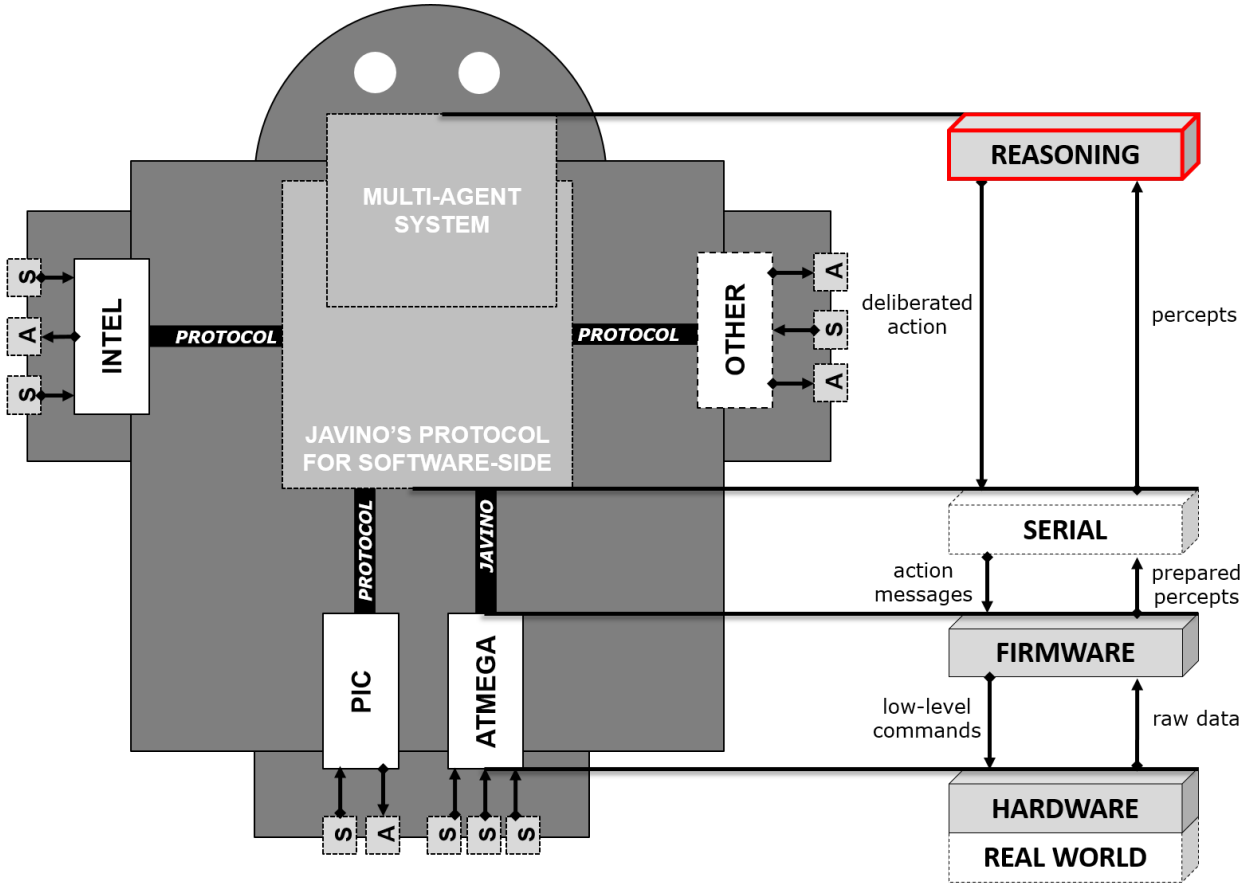
Na programação do microcontrolador, os dados brutos são transformados em percepções baseado na linguagem de programação orientada a agentes escolhida.

Arquitetura Física e Lógica



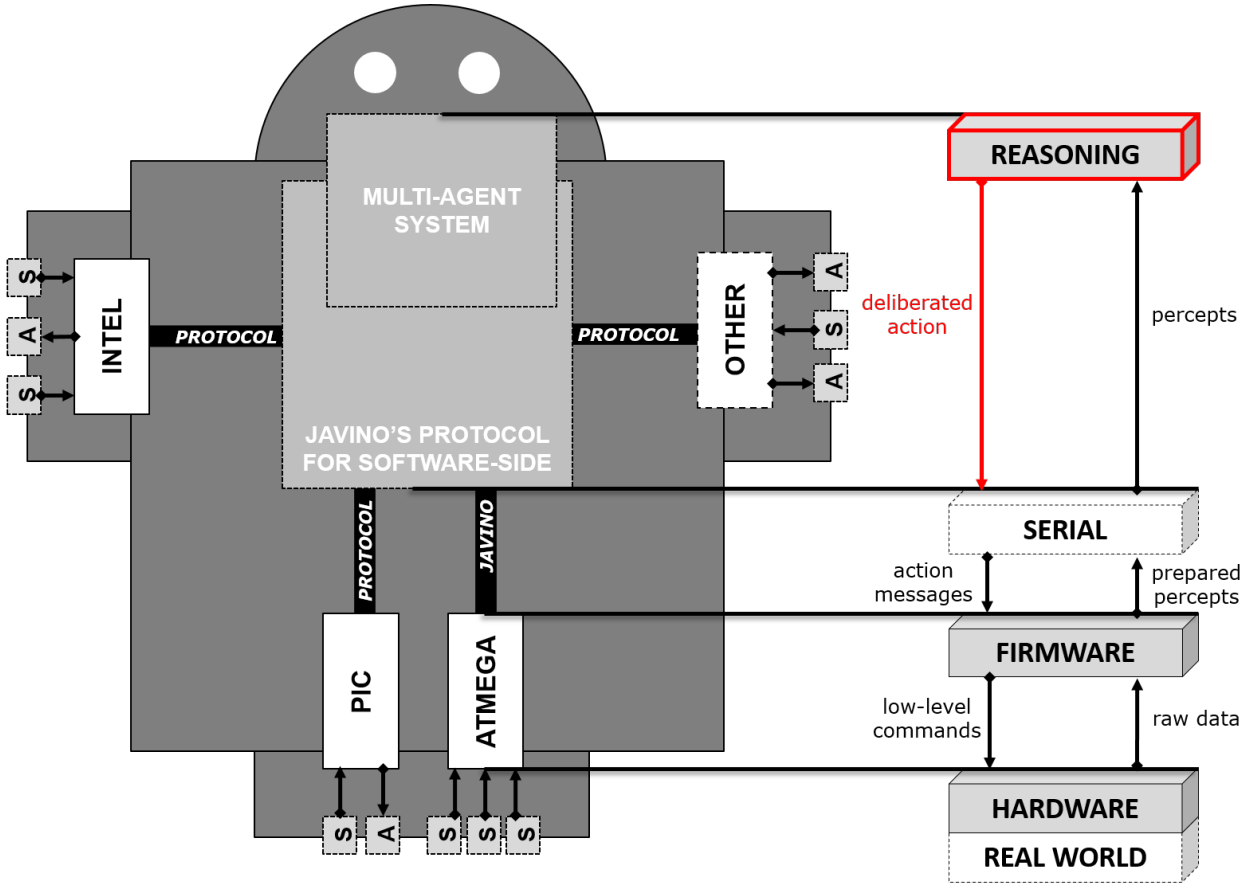
É responsável por enviar as percepções para a camada de raciocínio usando a comunicação serial.

Arquitetura Física e Lógica



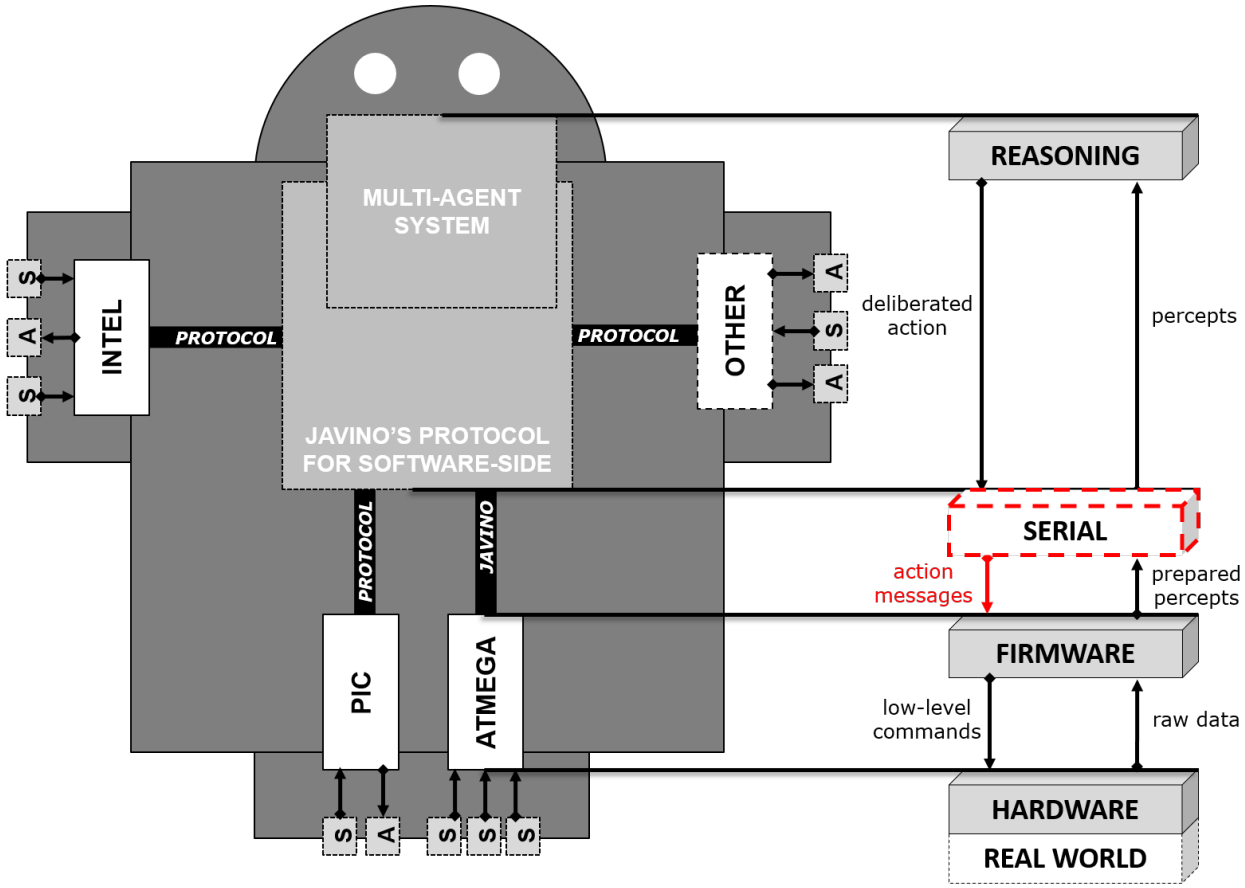
O agente é capaz de raciocinar com as percepções que vem diretamente do mundo real.

Arquitetura Física e Lógica



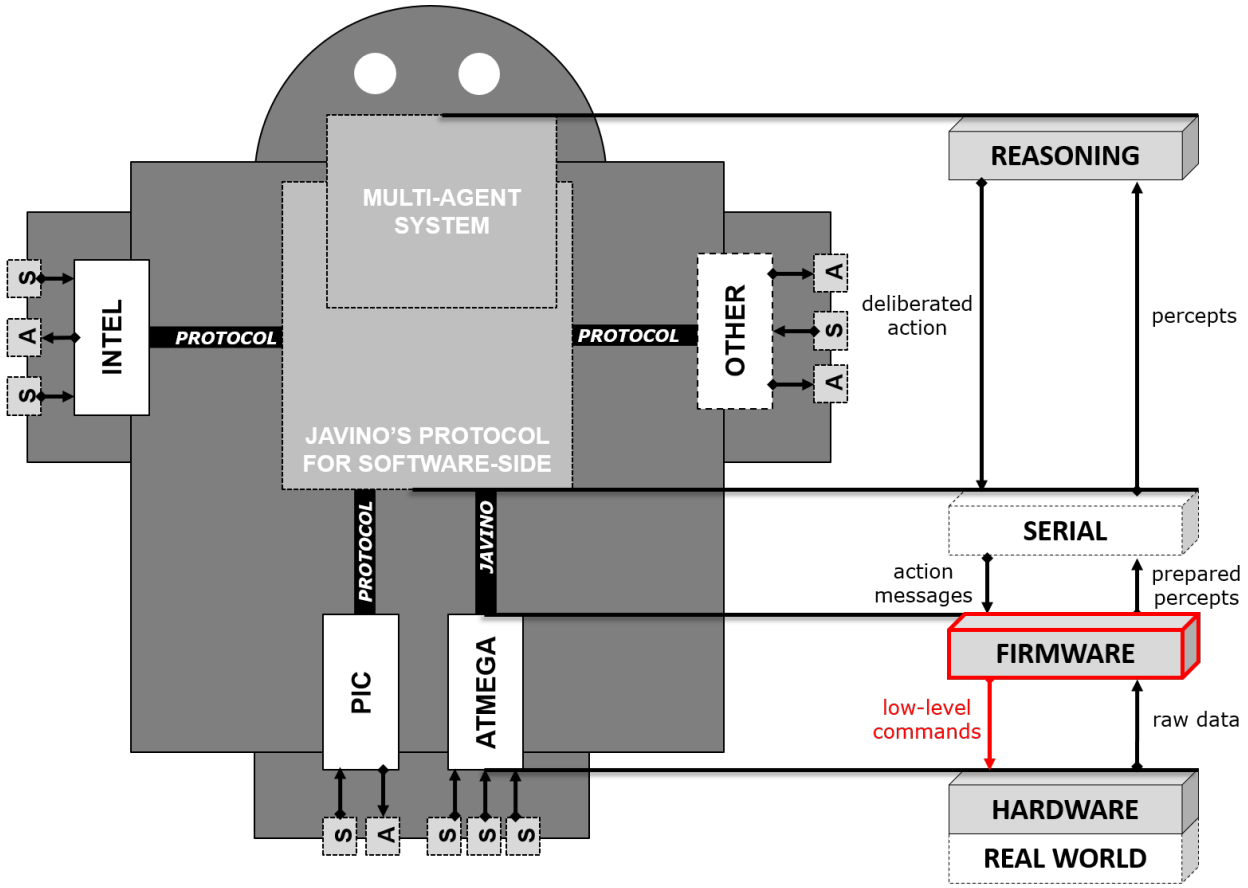
Então, o agente delibera e se alguma ação precisar ser executada. Neste caso, uma mensagem é enviada a camada serial.

Arquitetura Física e Lógica



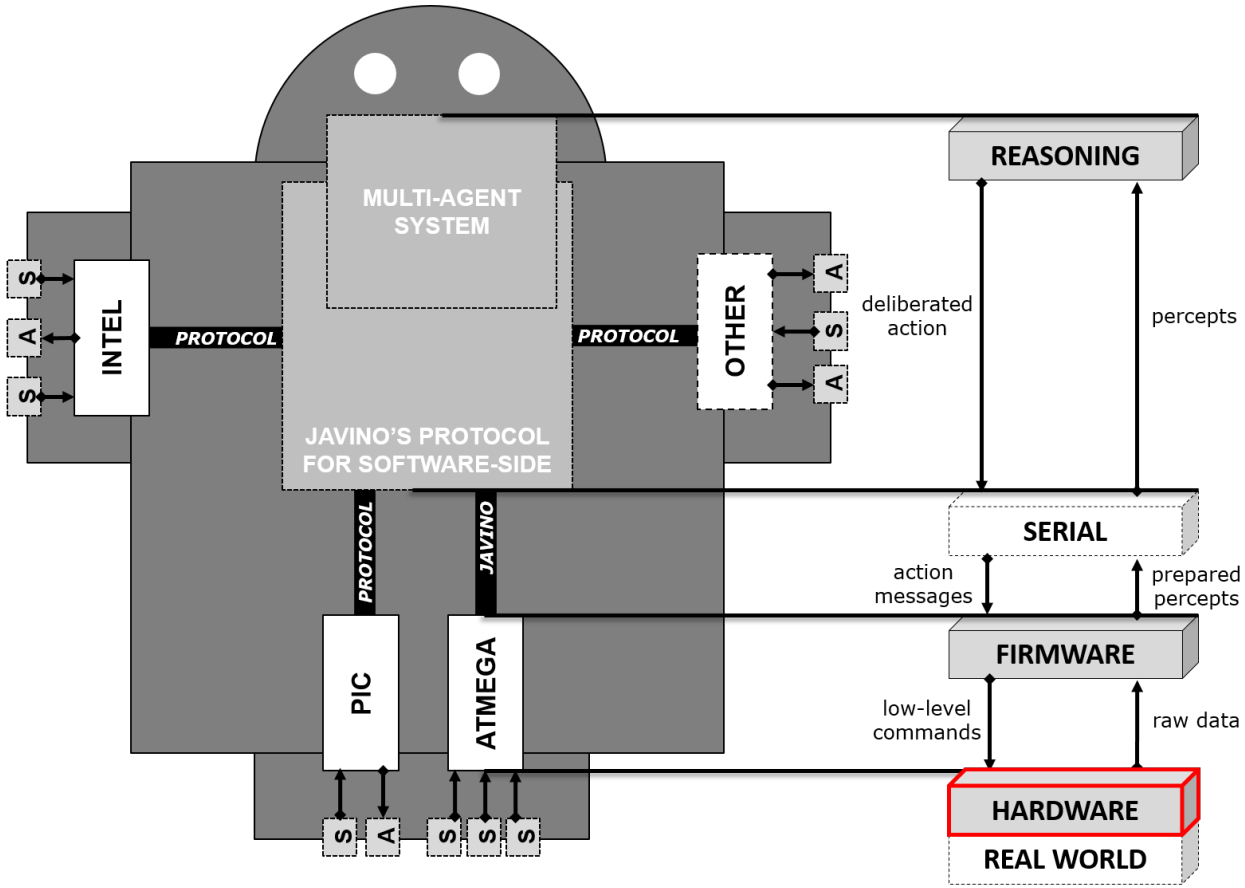
Redireciona as mensagens de ações para o microcontrolador que está conectado na porta USB identificado na mensagem.

Arquitetura Física e Lógica



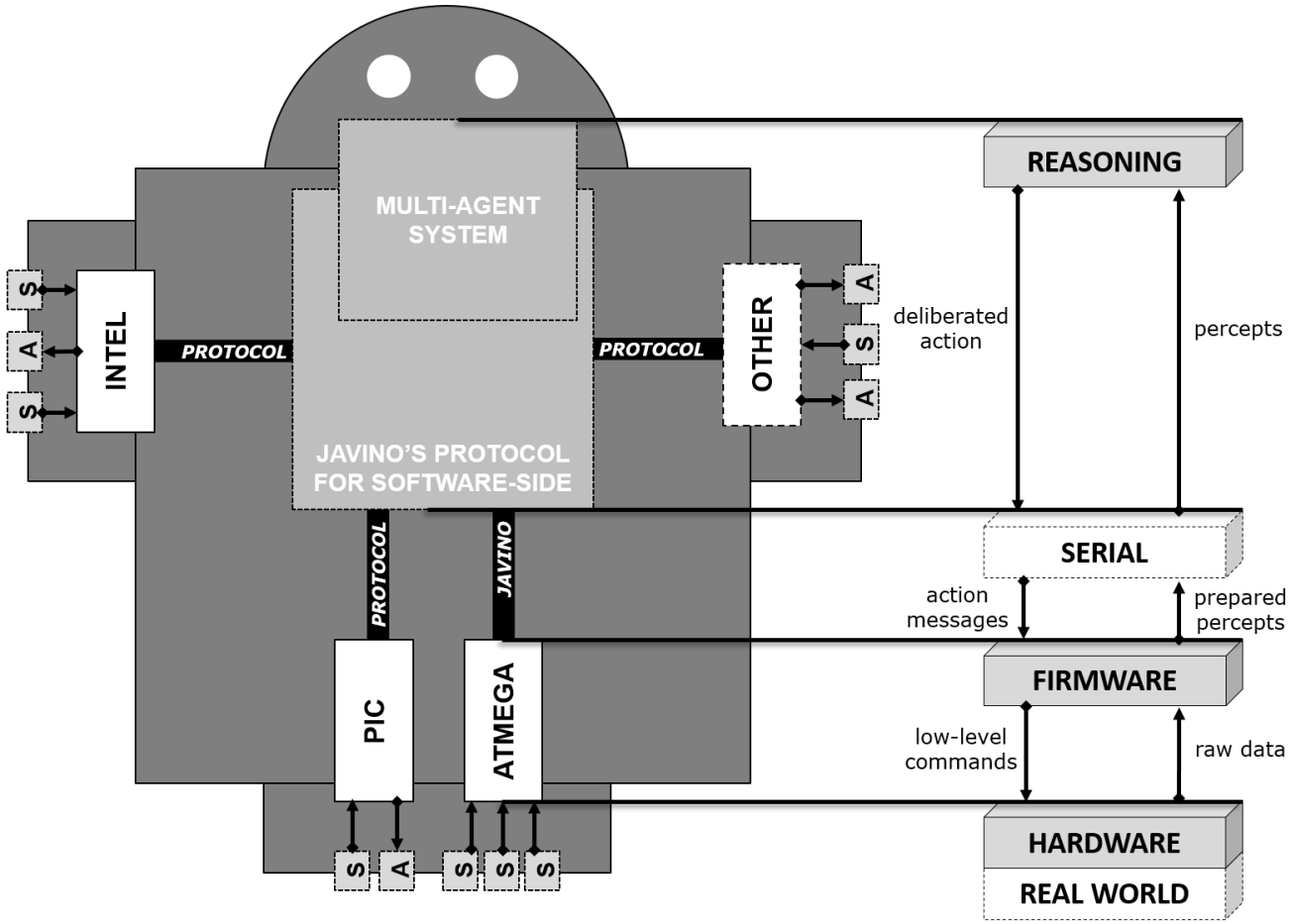
Todas as funções possíveis dos atuadores são programadas para serem executadas em resposta às mensagens vinda da porta serial.

Arquitetura Física e Lógica

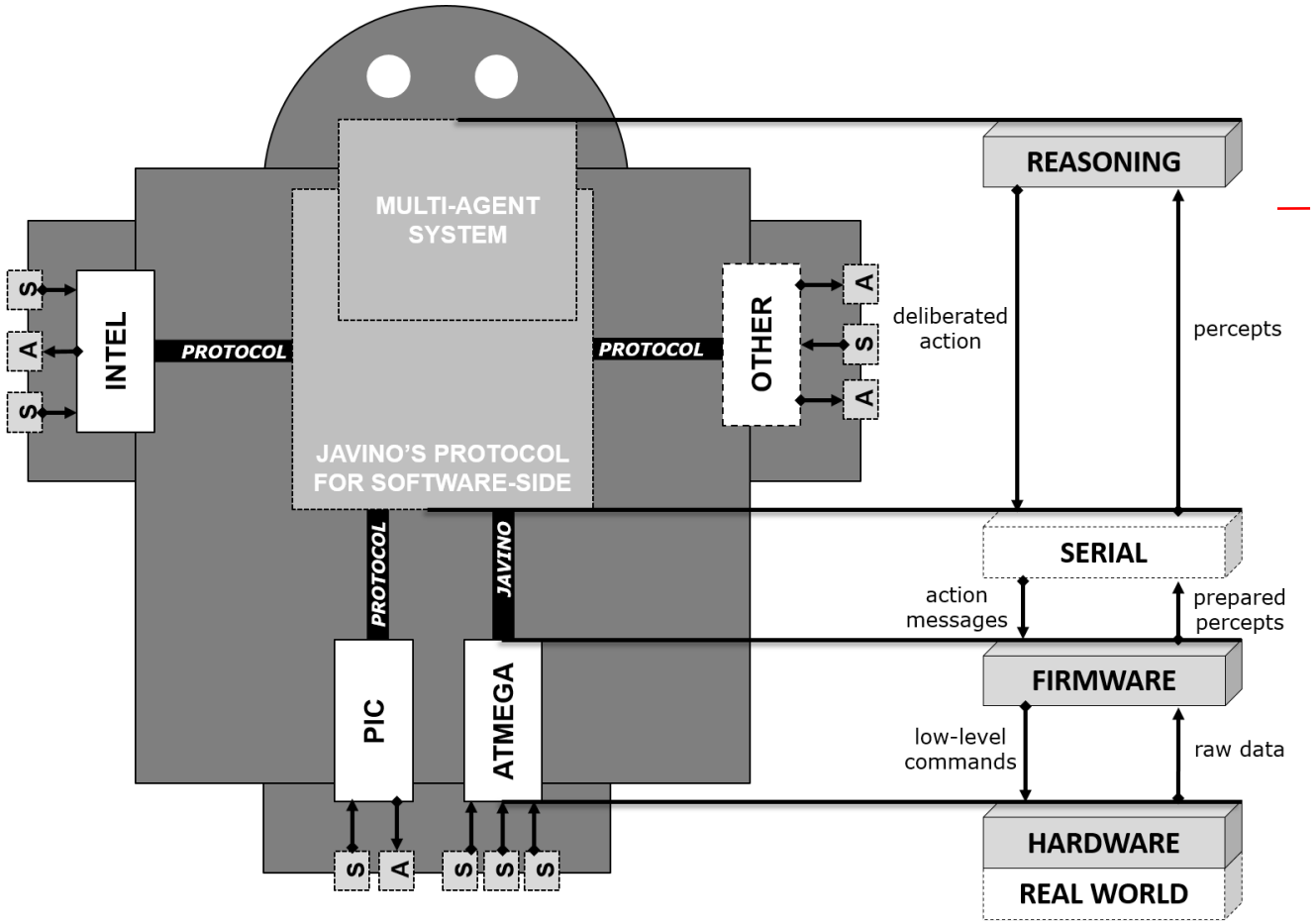


O efetuador é ativado.

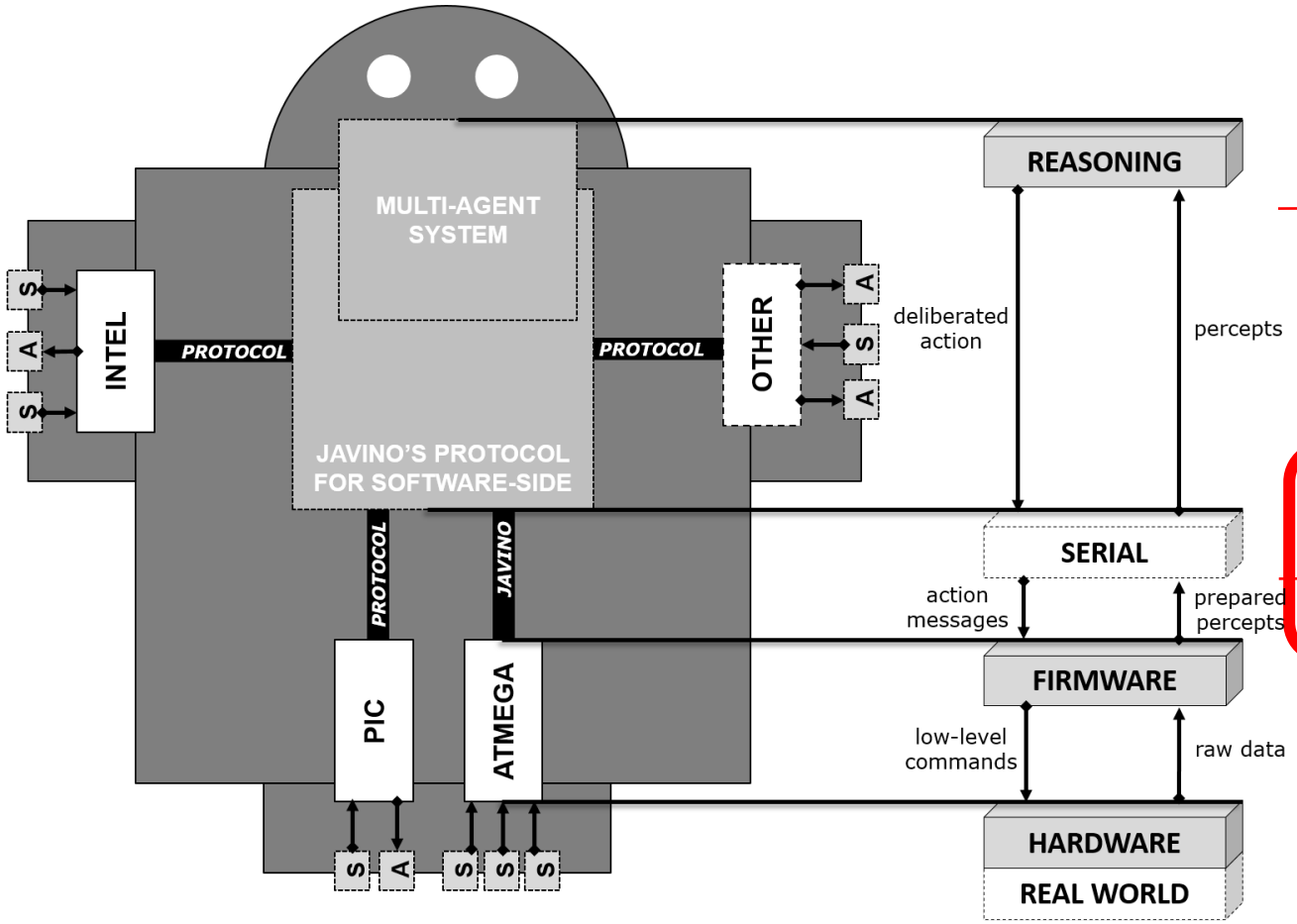
Toolkit to Facilitate Embedded MAS Development



Toolkit to Facilitate Embedded MAS Development



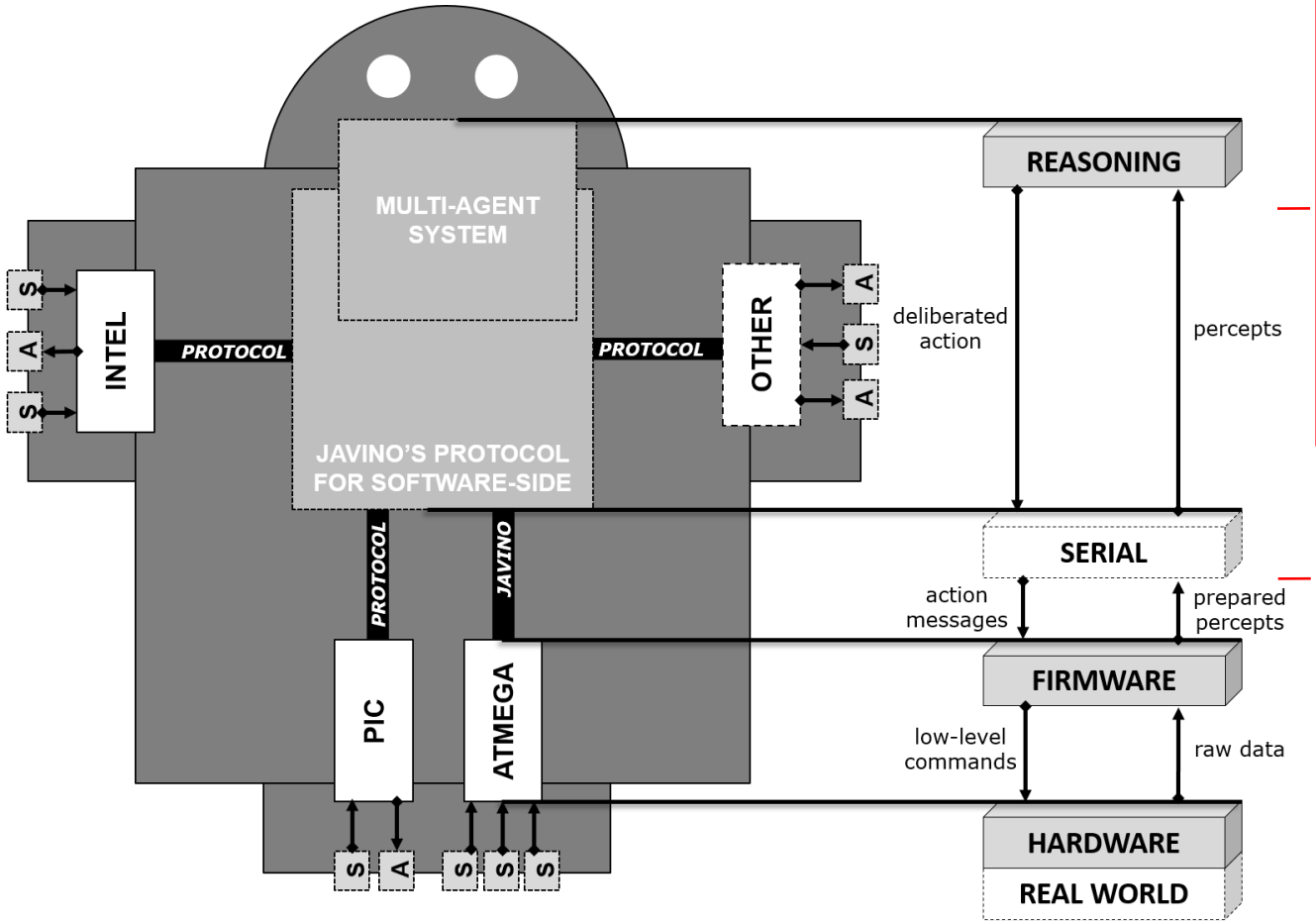
Javino

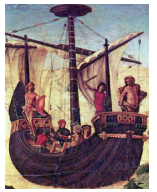


JavINO

LAZARIN, Nilson Mori; PANTOJA, Carlos Eduardo. **A robotic-agent platform for embedding software agents using raspberry pi and arduino boards**. In: 9TH SOFTWARE AGENTS, ENVIRONMENTS AND APPLICATIONS SCHOOL, 2015. Proceedings WESAAC 2015 [...]. Niteroi: UFF, 2015. p. 13–20. Disponível em: http://www2.ic.uff.br/~wesaac2015/Proceedings-WESAAC-2015.pdf.

Argo Agents



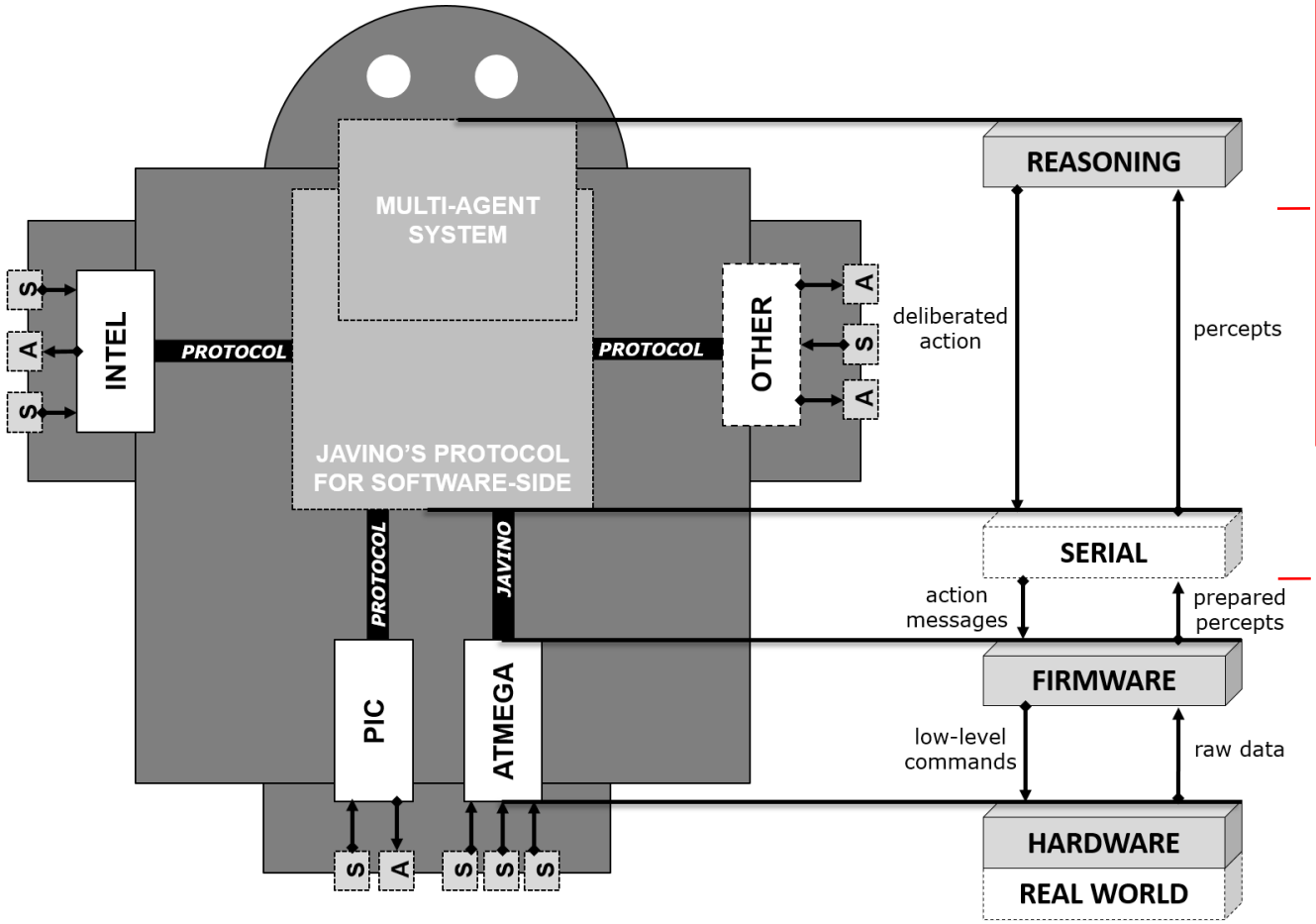


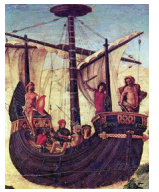
ARGO

JaVINO

Pantoja, C.E., Stabile, M.F., Lazarin, N.M., Sichman, J.S. (2016). **ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming**. In: Baldoni, M., Müller, J., Nunes, I., Zalila-Wenkstern, R. (eds) Engineering Multi-Agent Systems. EMAS 2016. Lecture Notes in Computer Science(), vol 10093. Springer, Cham. https://doi.org/10.1007/978-3-319-50983-9_8

Communicator Agents





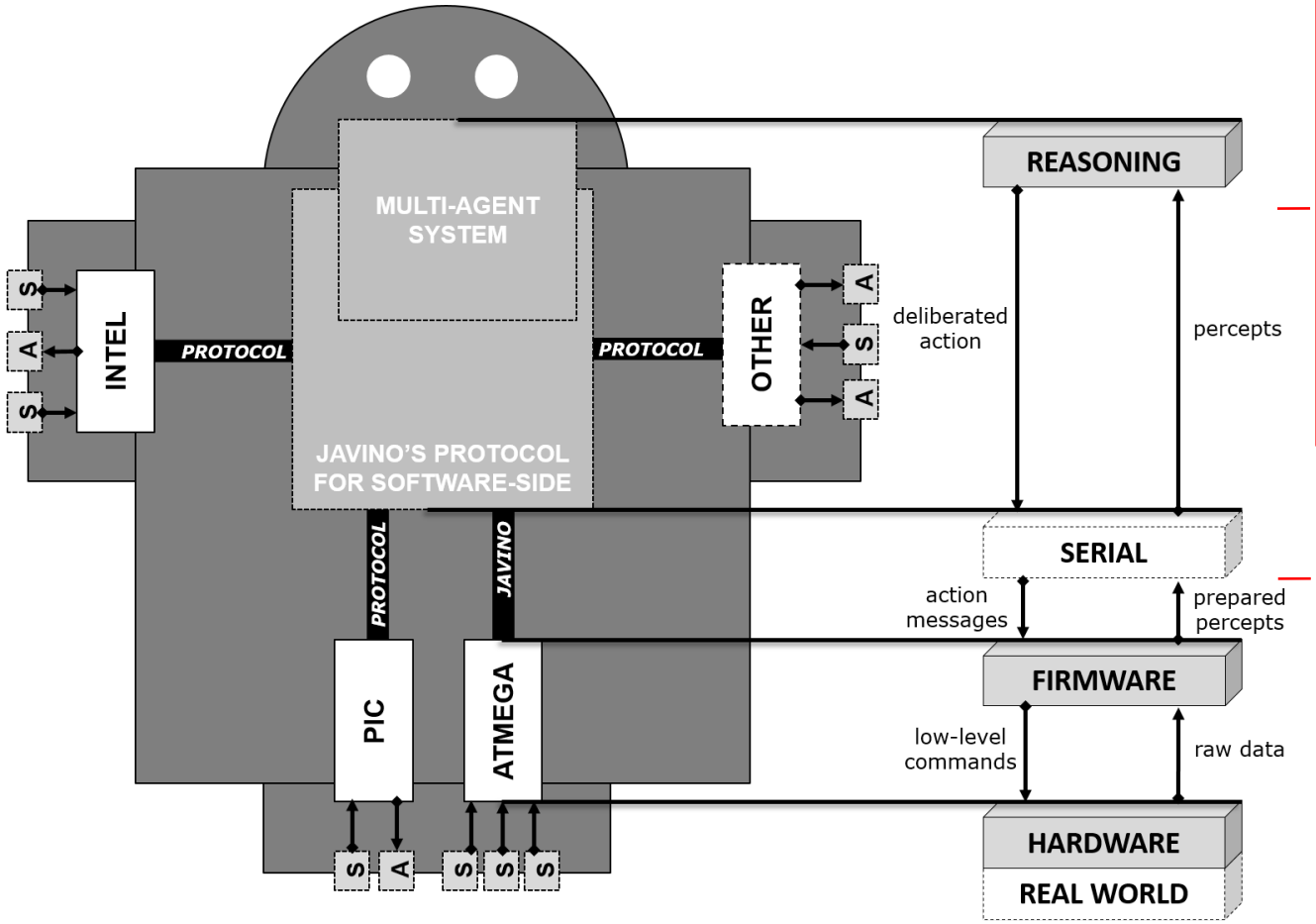
ARGO

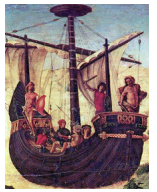
Communicators

JaVINO

PANTOJA, Carlos Eduardo; SOARES, Heder Dorneles; VITERBO, José; et al. **Exposing IoT Objects in the Internet Using the Resource Management Architecture**. International Journal of Software Engineering and Knowledge Engineering, v. 29, n. 11n12, p. 1703–1725, 2019.

Bio-Inspired Protocols





ARGO

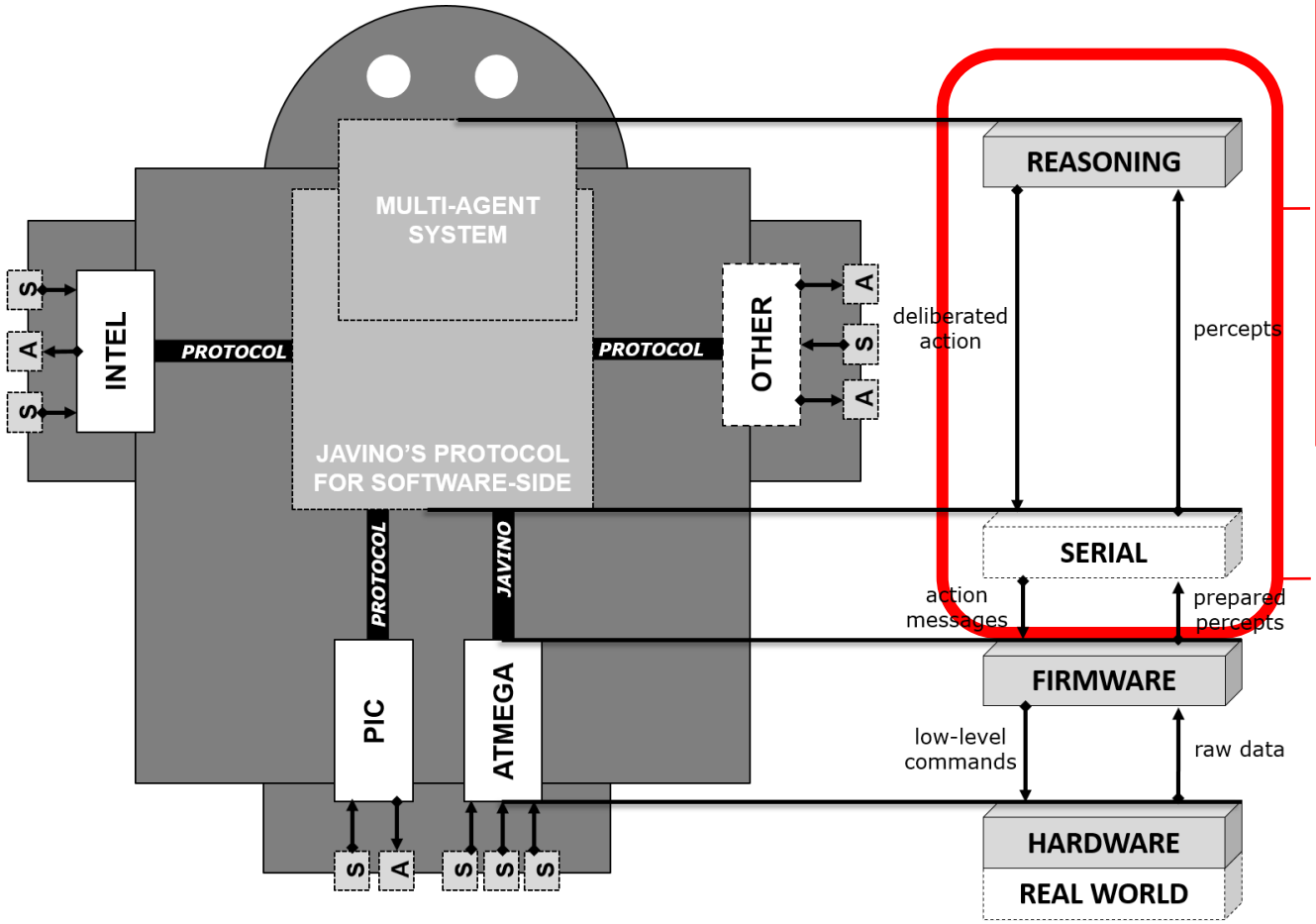
Bio-Inspired Protocols

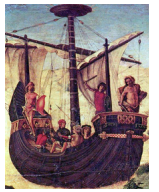
Communicators

JavINO

Souza de Jesus V., Pantoja C., Manoel F., Alves G., Viterbo J. and Bezerra E. (2021). **Bio-Inspired Protocols for Embodied Multi-Agent Systems**.In Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART, ISBN 978-989-758-484-8, pages 312-320. DOI: 10.5220/0010257803120320

Swapping Physical Resources at Runtime



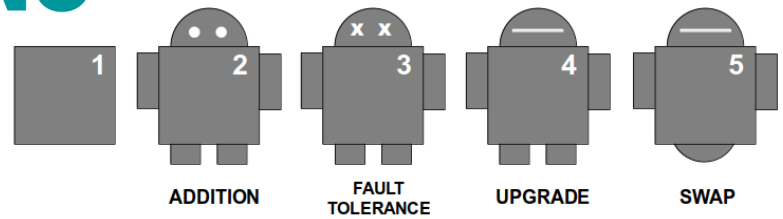


ARGO

Bio-Inspired Protocols

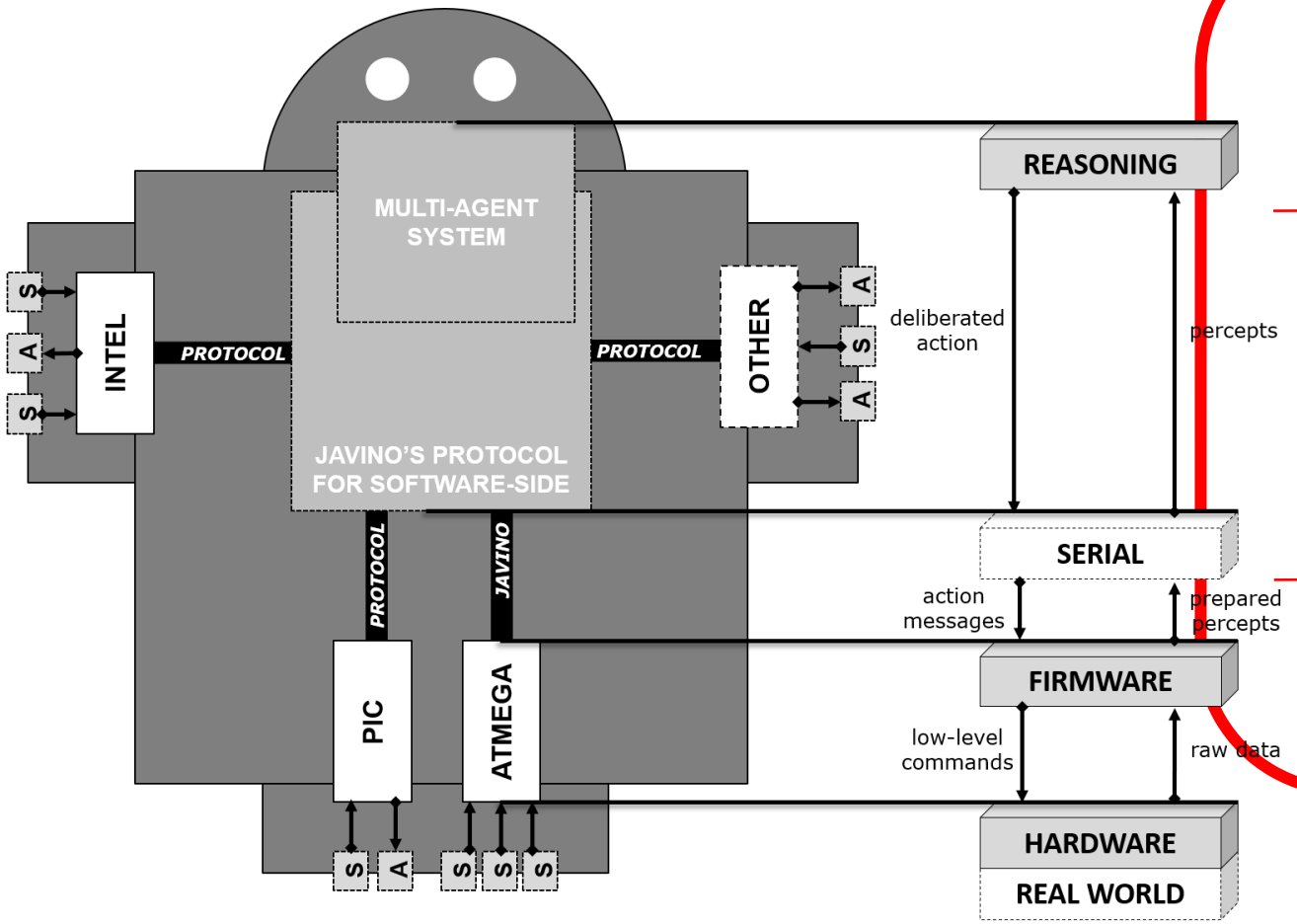
Communicators

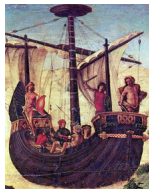
JavINO



Lazarin, N.; Pantoja, C. and Viterbo, J. (2023). **Swapping Physical Resources at Runtime in Embedded MultiAgent Systems**. In Proceedings of the 15th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART; ISBN 978-989-758-623-1; ISSN 2184-433X, SciTePress, pages 93-104. DOI: 10.5220/0011750700003393

Jason Embedded and Packages



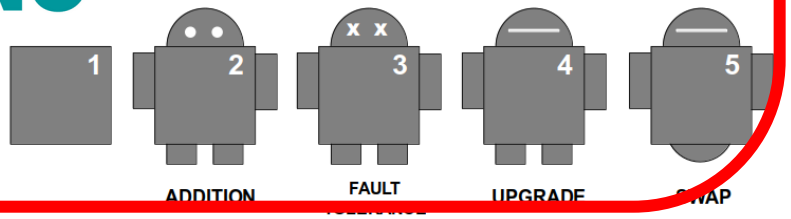


ARGO

Bio-Inspired Protocols

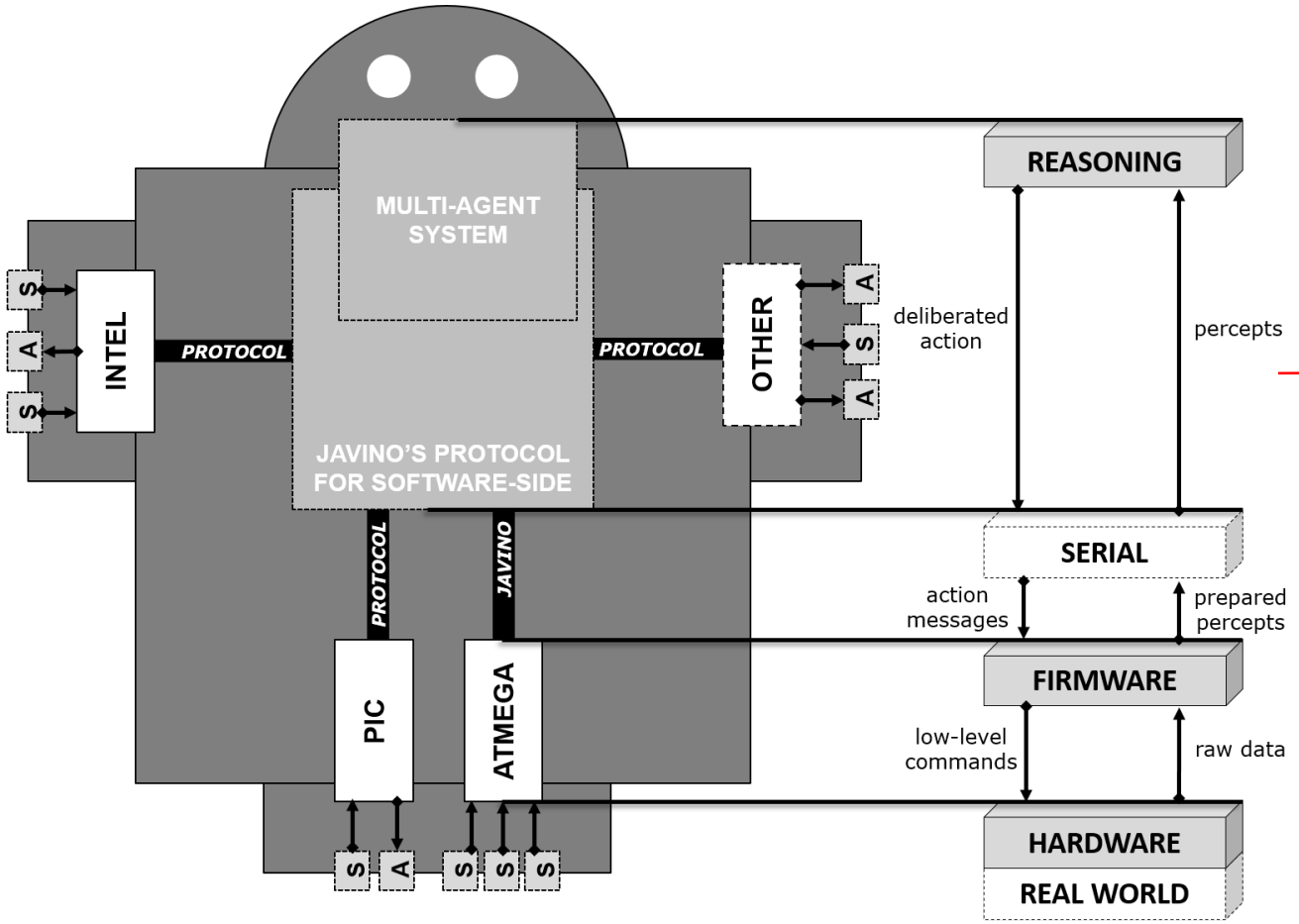
Communicators

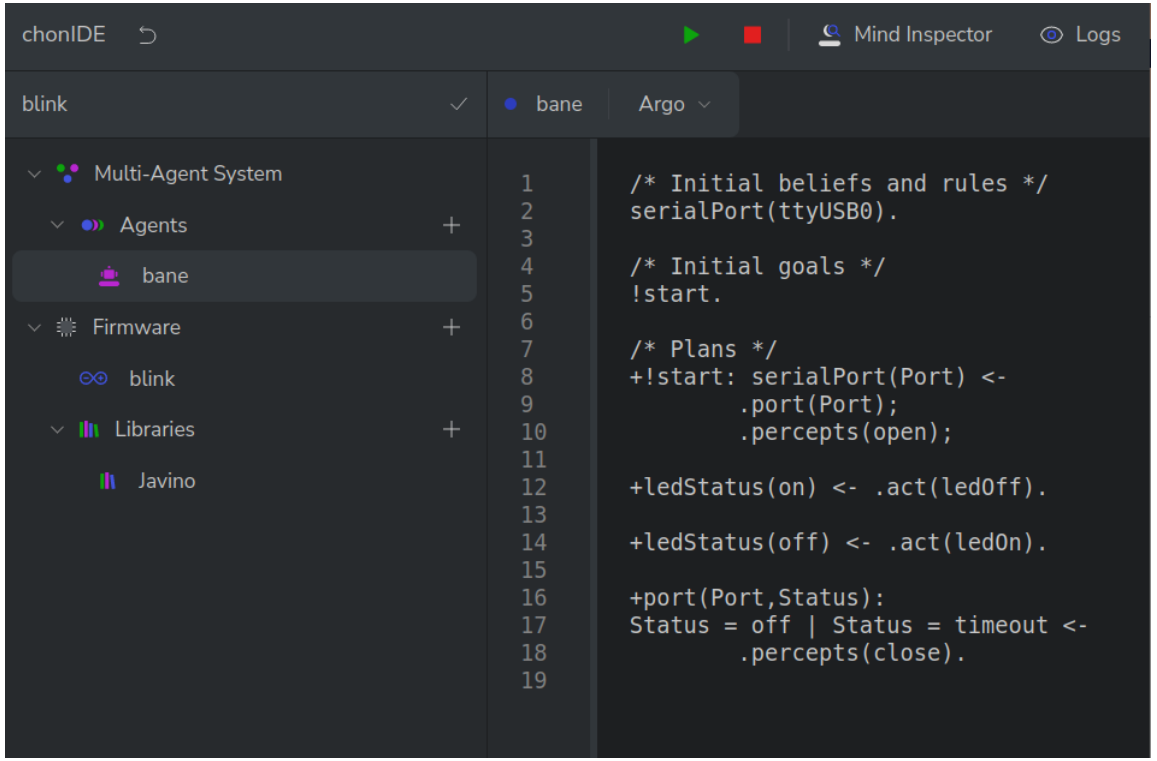
JavINO



PANTOJA, CARLOS EDUARDO; JESUS, VINICIUS SOUZA; LAZARIN, NILSON MORI; VITERBO, JOSÉ. ***A Spin-off Version of Jason for IoT and Embedded Multiagent Systems***. In : Proceedings of the 12nd Brazilian Conference on Intelligent Systems, BRACIS 2023, Belo Horizonte, Brazil, 2023

ChonIDE



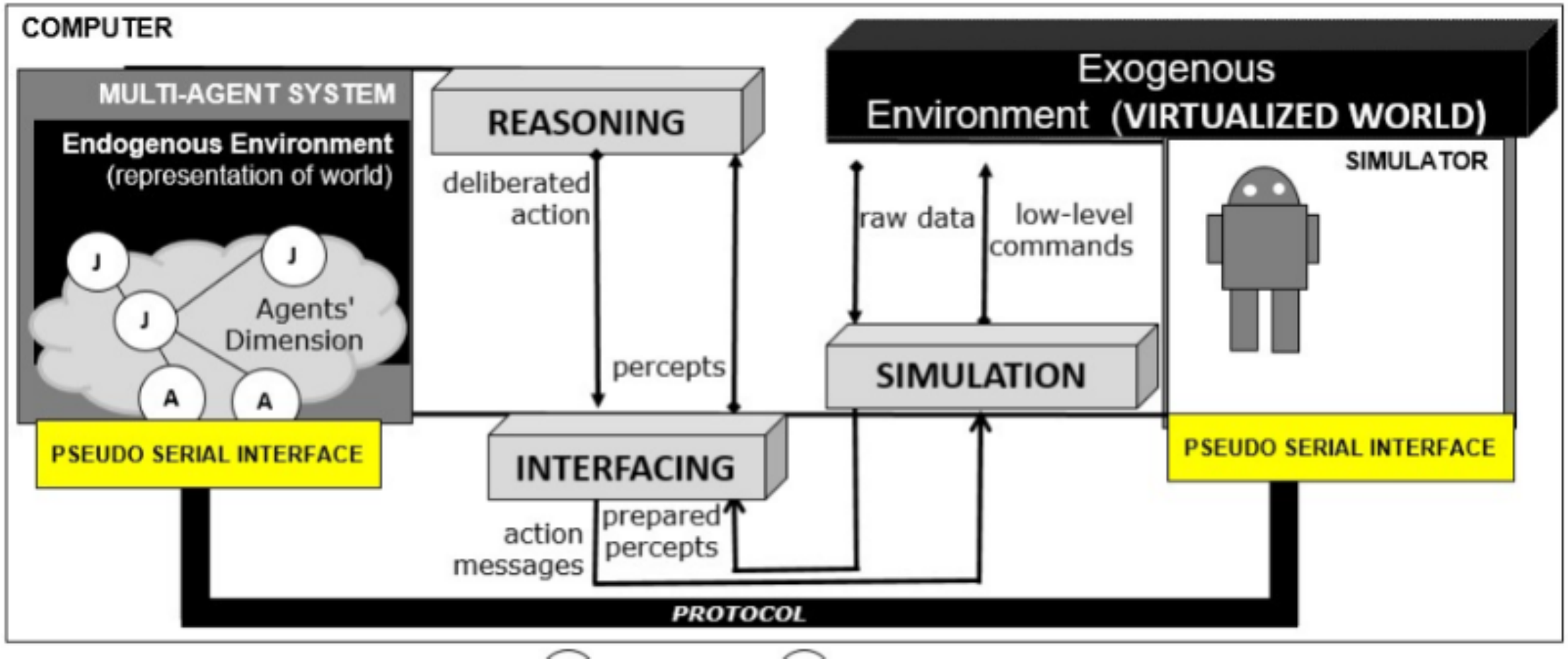


Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). **An IDE to Support the Development of Embedded Multi-Agent Systems**. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection. PAAMS 2023. Lecture Notes in Computer Science(), vol 13955. Springer, Cham. https://doi.org/10.1007/978-3-031-37616-0_29

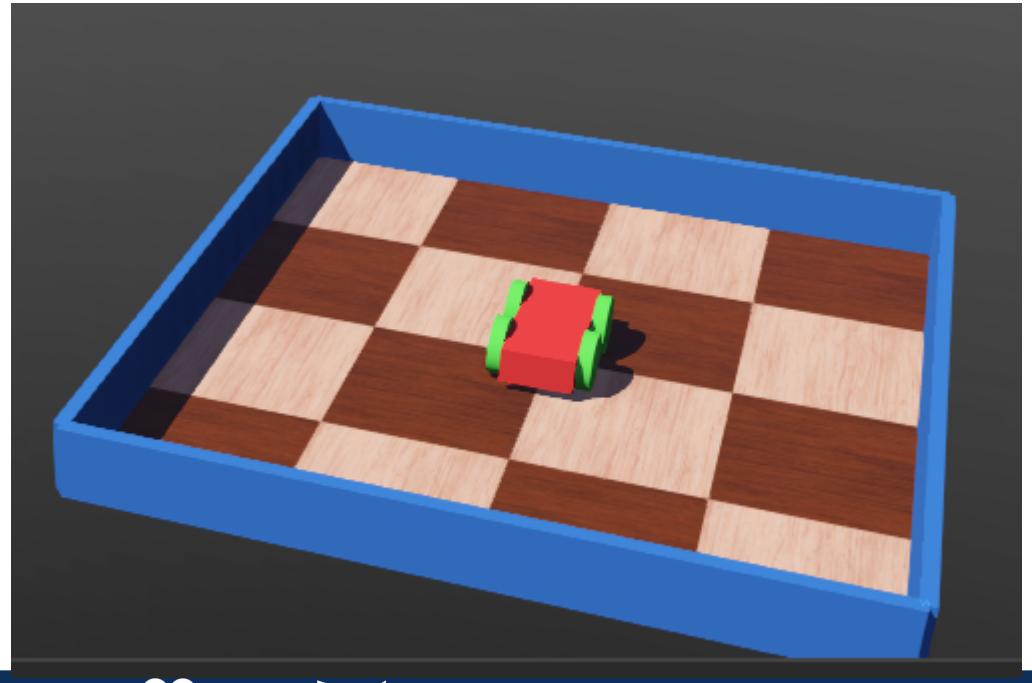
Simulação

- Criar protótipos no mundo real de sistemas embarcados ou robóticos é uma tarefa custosa e potencialmente complexa
- Validar ideias diretamente no mundo real é perigoso pois erros podem danificar o hardware
- O uso de simuladores permite validar ideias sem construir os protótipos
- Adotamos uma metodologia de simulação baseado na **emulação do canal de comunicação serial**
- Dessa maneira, ao invés de criar simuladores, integramos diversos simuladores já existentes

Simulação


COMPUTER
MULTI-AGENT SYSTEM
Endogenous Environment
(representation of world)
J
J
J
Agents'
Dimension
A
A
PSEUDO SERIAL INTERFACE
REASONING
deliberated
action
percepts
INTERFACING
action
messages
prepared
percepts
Exogenous
Environment (VIRTUALIZED WORLD)
raw data
low-level
commands
SIMULATION
SIMULATOR
PSEUDO SERIAL INTERFACE
PROTOCOL

Exemplo: protótipo ChonBot



Sensores	Ações
dLeft (d)	goAhead
dRight (d)	goLeft
battery(b)	goRight
gps(x, y , z)	goBack

Exemplo: protótipo ChonBot

git clone https://github.com/bptfreitas/WESAAC2025_CenariosWebots.git

```
bruno@bruno-laptop:~$ git clone https://github.com/bptfreitas/WESAAC2025_CenariosWebots.git
Cloning into 'WESAAC2025_CenariosWebots'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 22 (delta 3), reused 18 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (22/22), 8.26 KiB | 1.65 MiB/s, done.
Resolving deltas: 100% (3/3), done.
bruno@bruno-laptop:~$
```

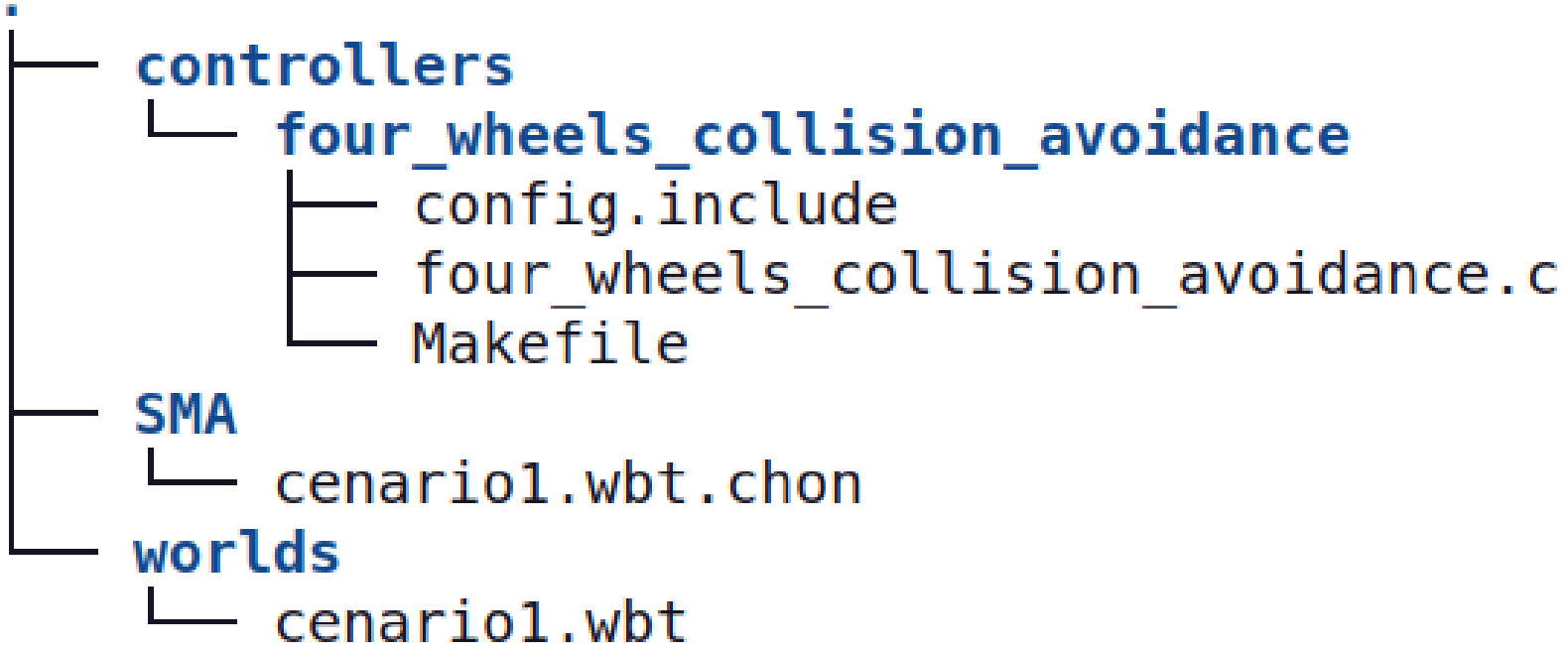
Exemplo: protótipo ChonBot

cd WESAAC2025_CenariosWebots/cenario1/

```
bruno@bruno-laptop:~$ cd WESAAC2025_CenariosWebots/cenario1/
bruno@bruno-laptop:~/WESAAC2025_CenariosWebots/cenario1$
```

Exemplo: protótipo ChonBot

Estrutura da pasta:



Exemplo: protótipo ChonBot

cd controllers/four_wheels_collision_avoidance/

make WEBOTS_HOME=/usr/local/webots all

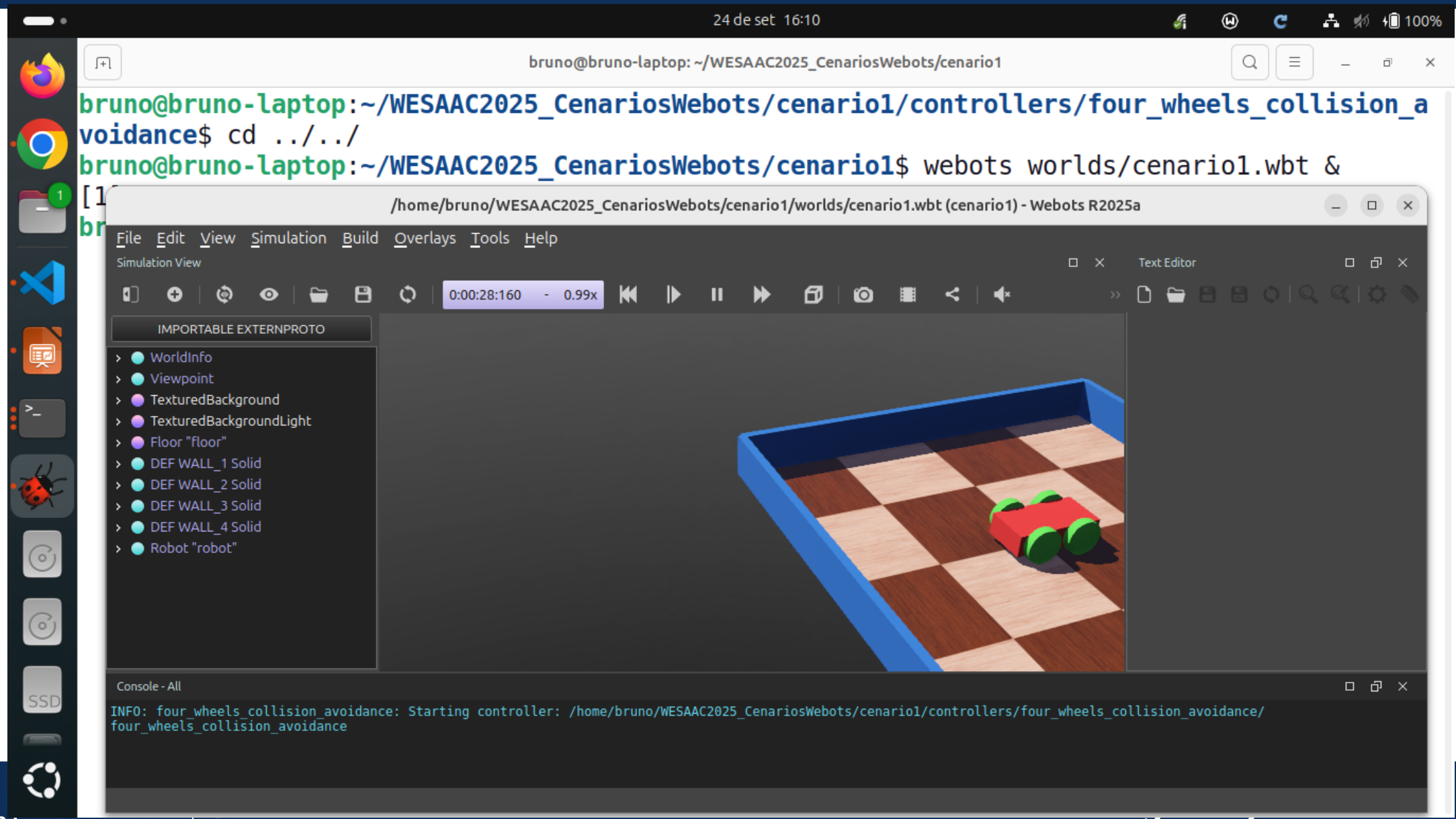
```
bruno@bruno-laptop:~/WESAAC2025_CenariosWebots/cenario1$ cd controllers/four_wheels_collisi
on_avoidance/
bruno@bruno-laptop:~/WESAAC2025_CenariosWebots/cenario1/controllers/four_wheels_collision_a
voidance$ make WEBOTS_HOME=/usr/local/webots all
# updating four_wheels_collision_avoidance.d
git clone https://github.com/chon-group/JavinoInC.git
Cloning into 'JavinoInC'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (89/89), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 89 (delta 35), reused 74 (delta 24), pack-reused 0 (from 0)
Receiving objects: 100% (89/89), 25.64 KiB | 4.27 MiB/s, done.
Resolving deltas: 100% (35/35), done.
# updating javino.d
# compiling javino.c
# compiling four_wheels_collision_avoidance.c
# linking four_wheels_collision_avoidance
# copying to four_wheels_collision_avoidance
```

Exemplo: protótipo ChonBot

cd ../..

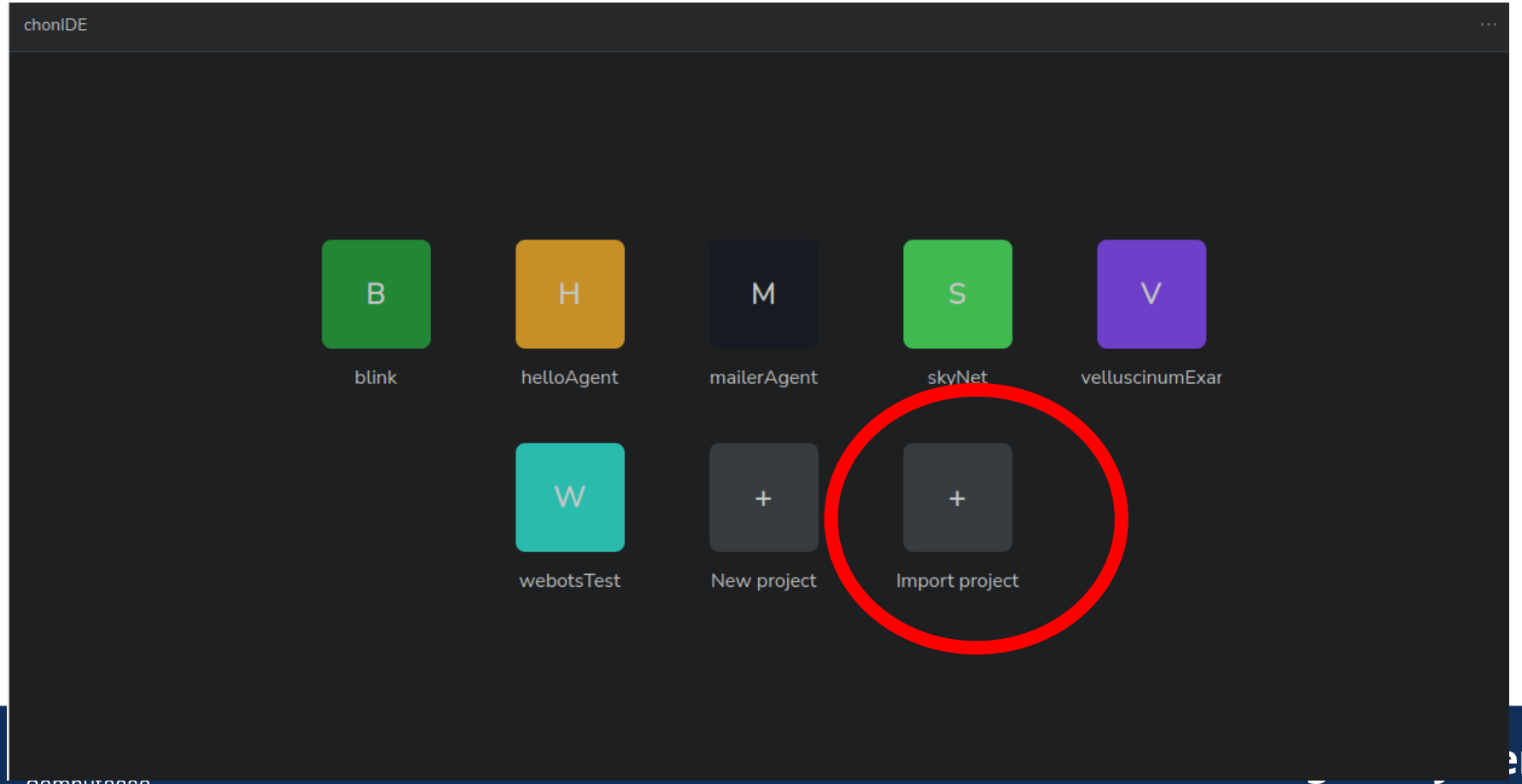
webots worlds/cenario1.wbt &

Exemplo: protótipo ChonBot



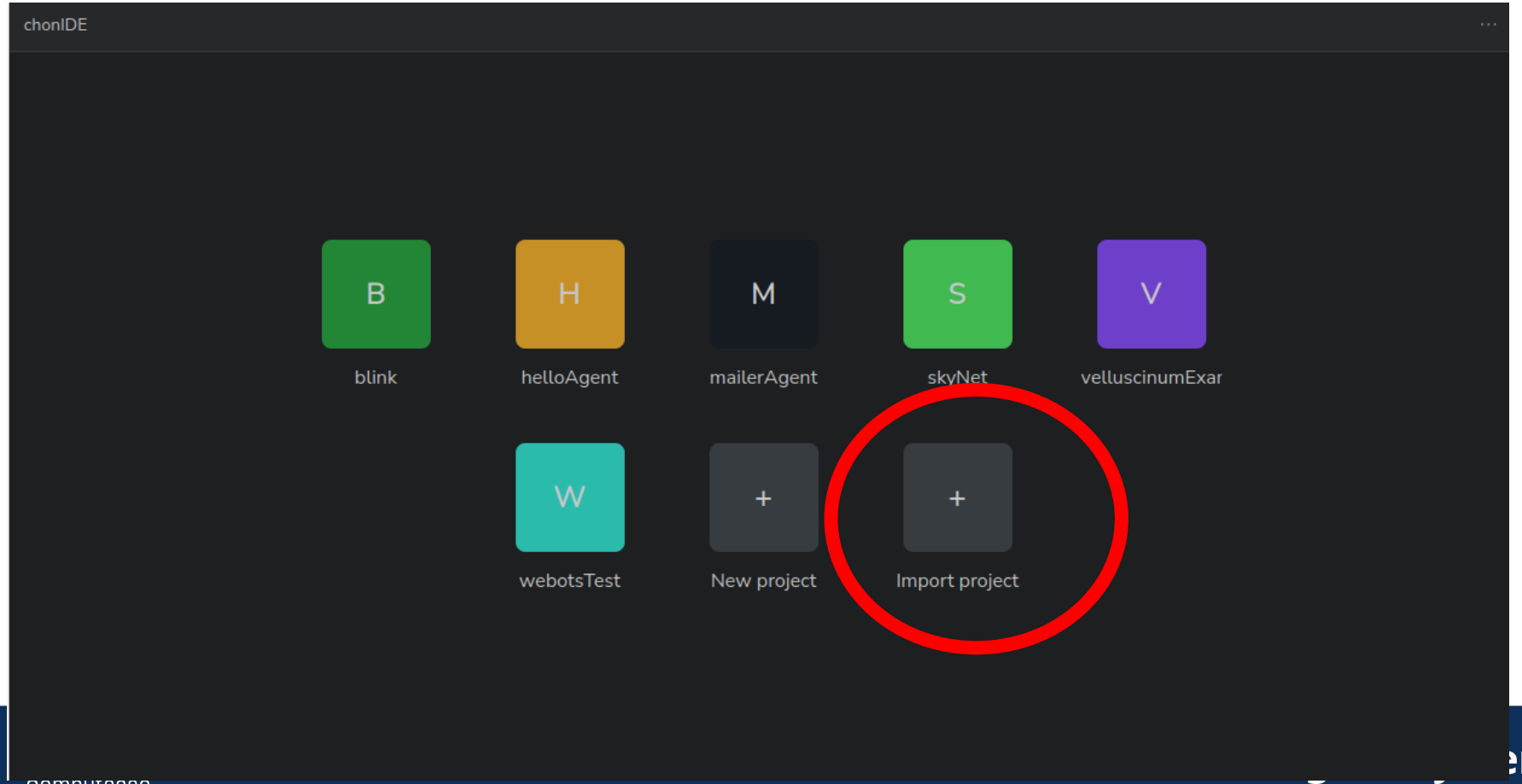
Exemplo: protótipo ChonBot

- Abra a- ChonIDE, faça login
- Selecione “Import Project”



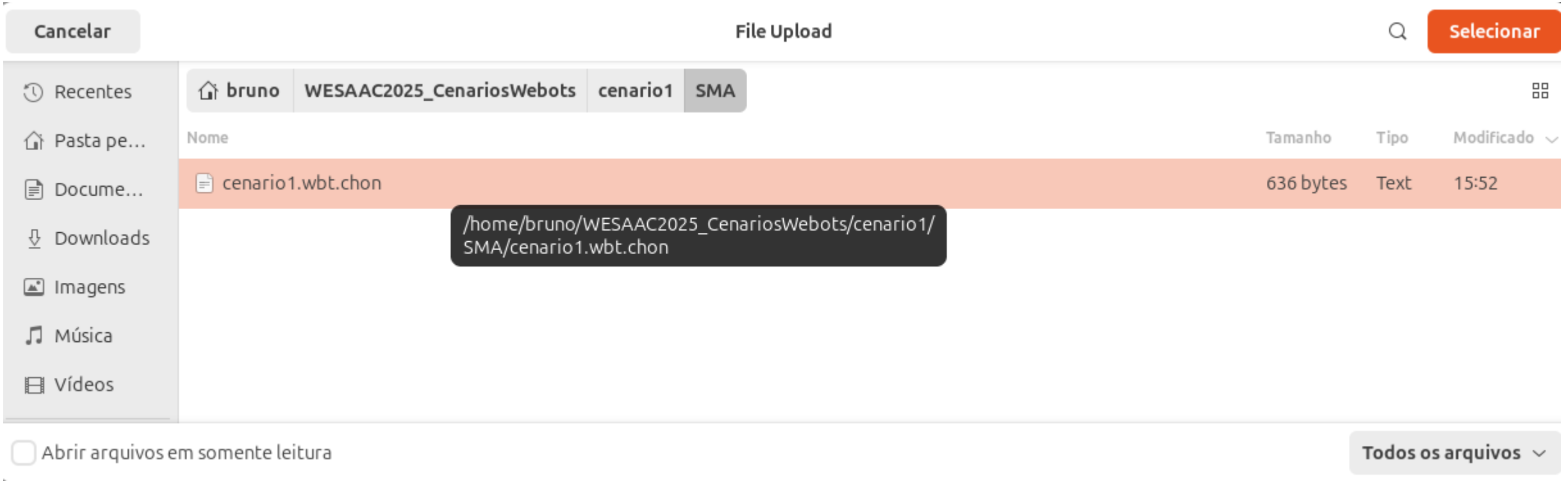
Exemplo: protótipo ChonBot

- Entre na pasta “SMA” e selecione o projeto “cenario1.”



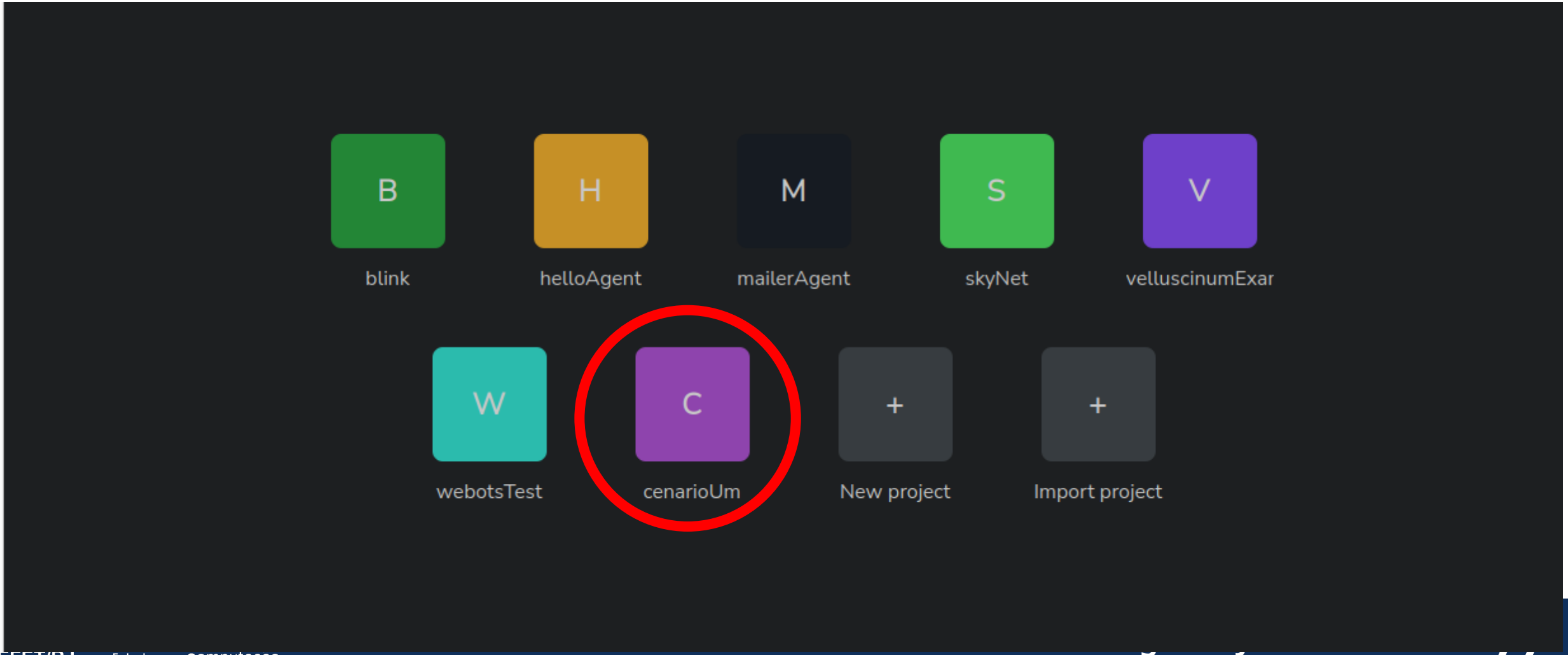
Exemplo: protótipo ChonBot

- Entre na pasta “SMA” e selecione o projeto “cenario1.wbt.chon”

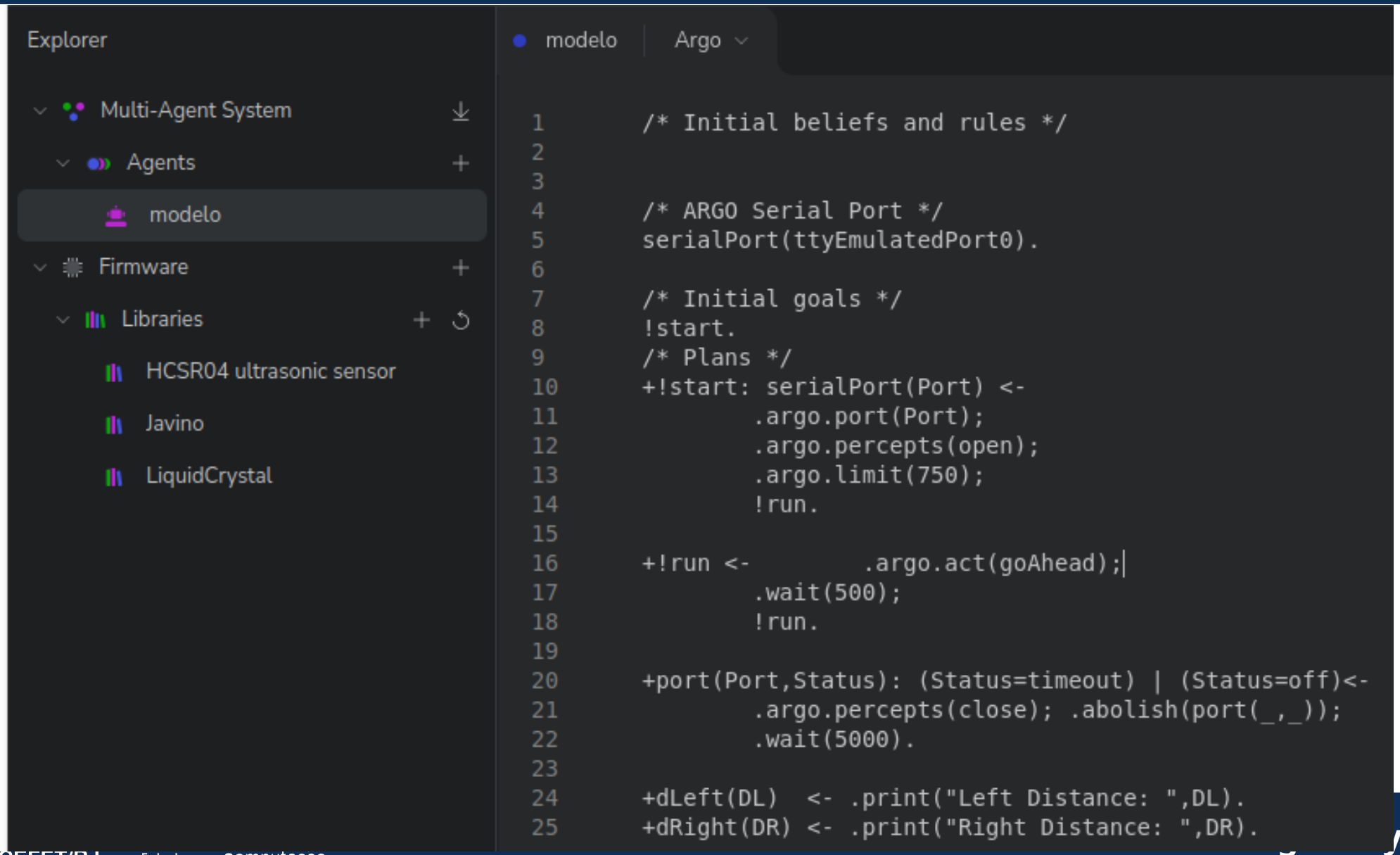


Exemplo: protótipo ChonBot

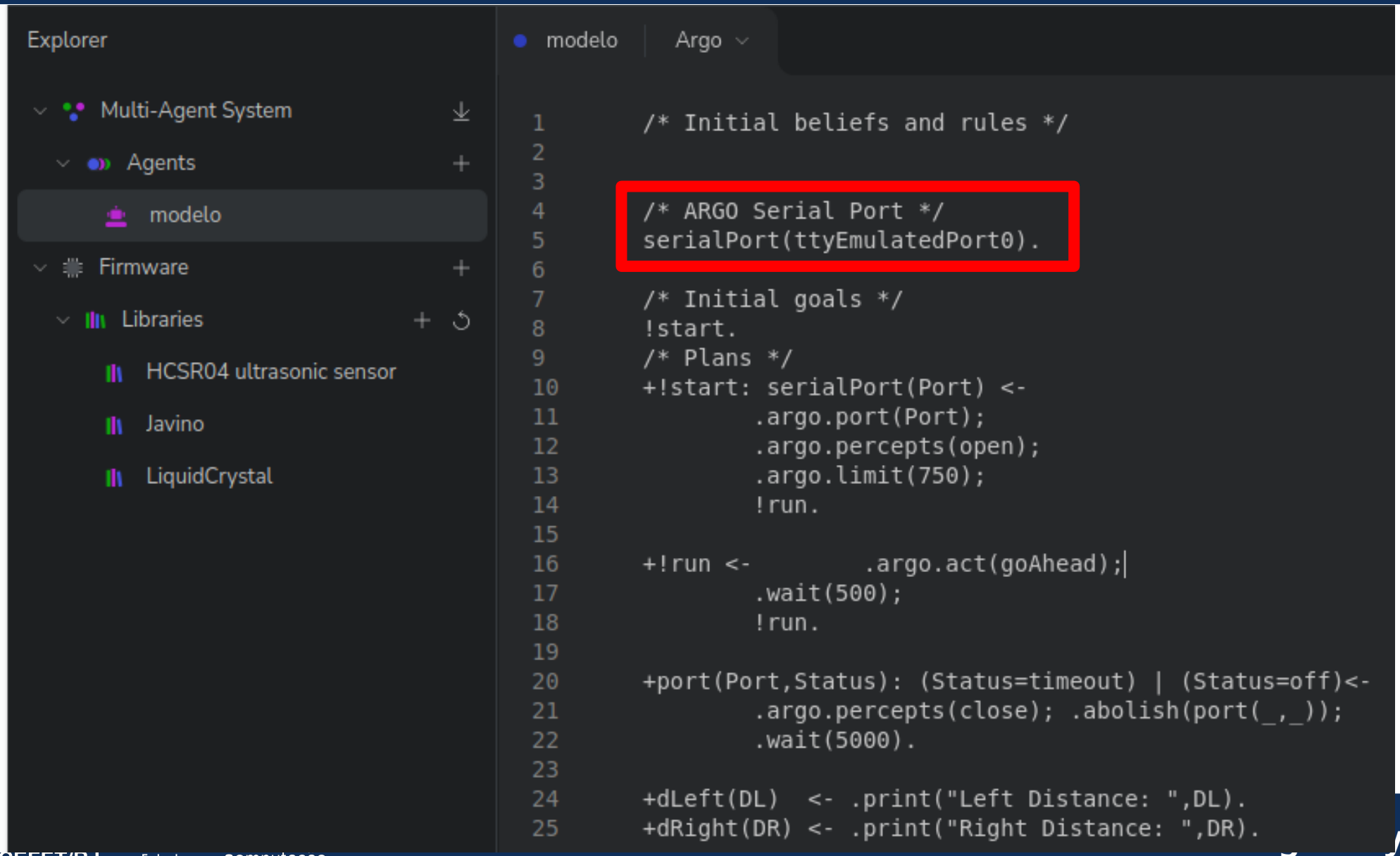
- Selecione o projeto recém-criado



Exemplo: protótipo ChonBot

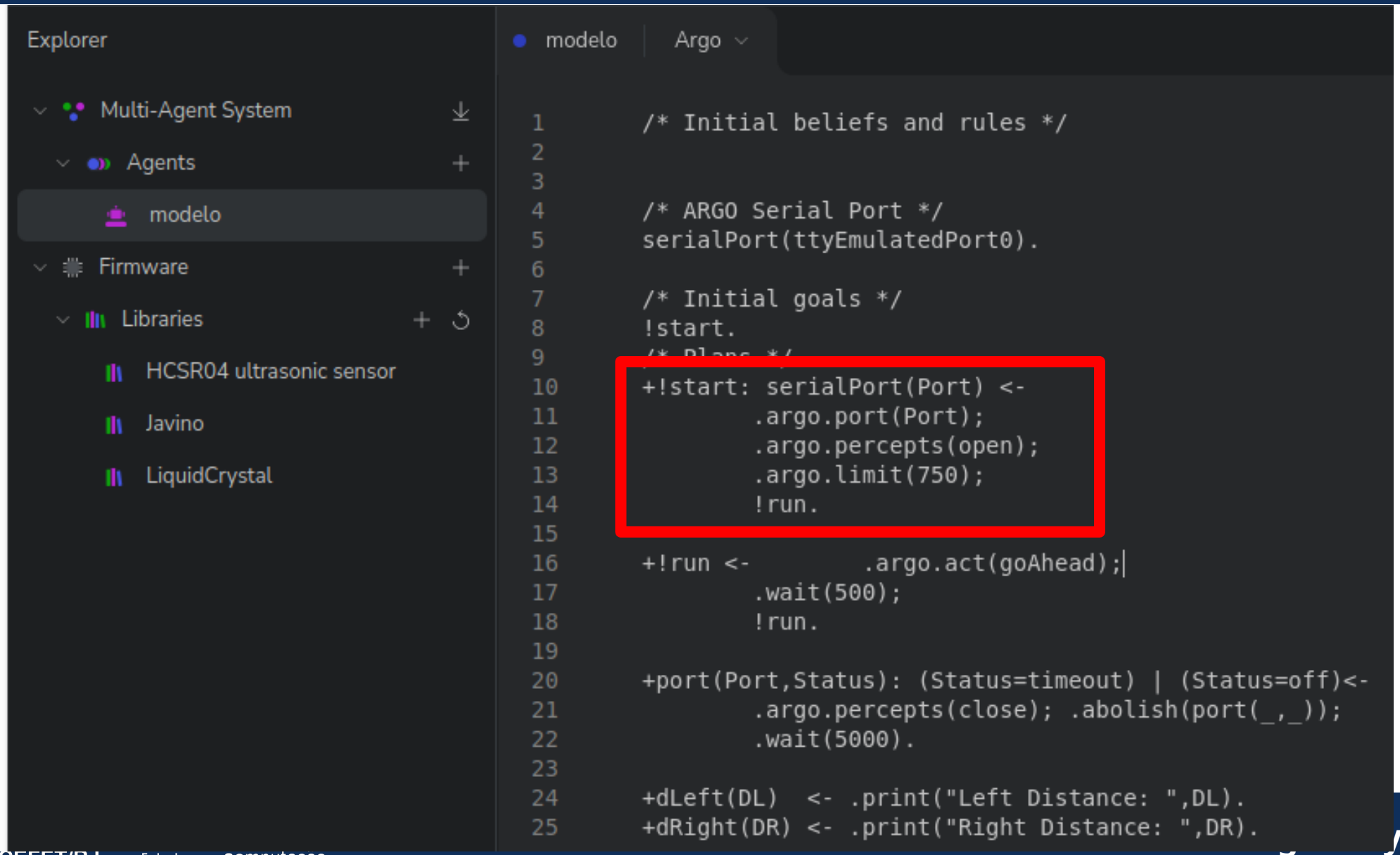


Exemplo: protótipo ChonBot



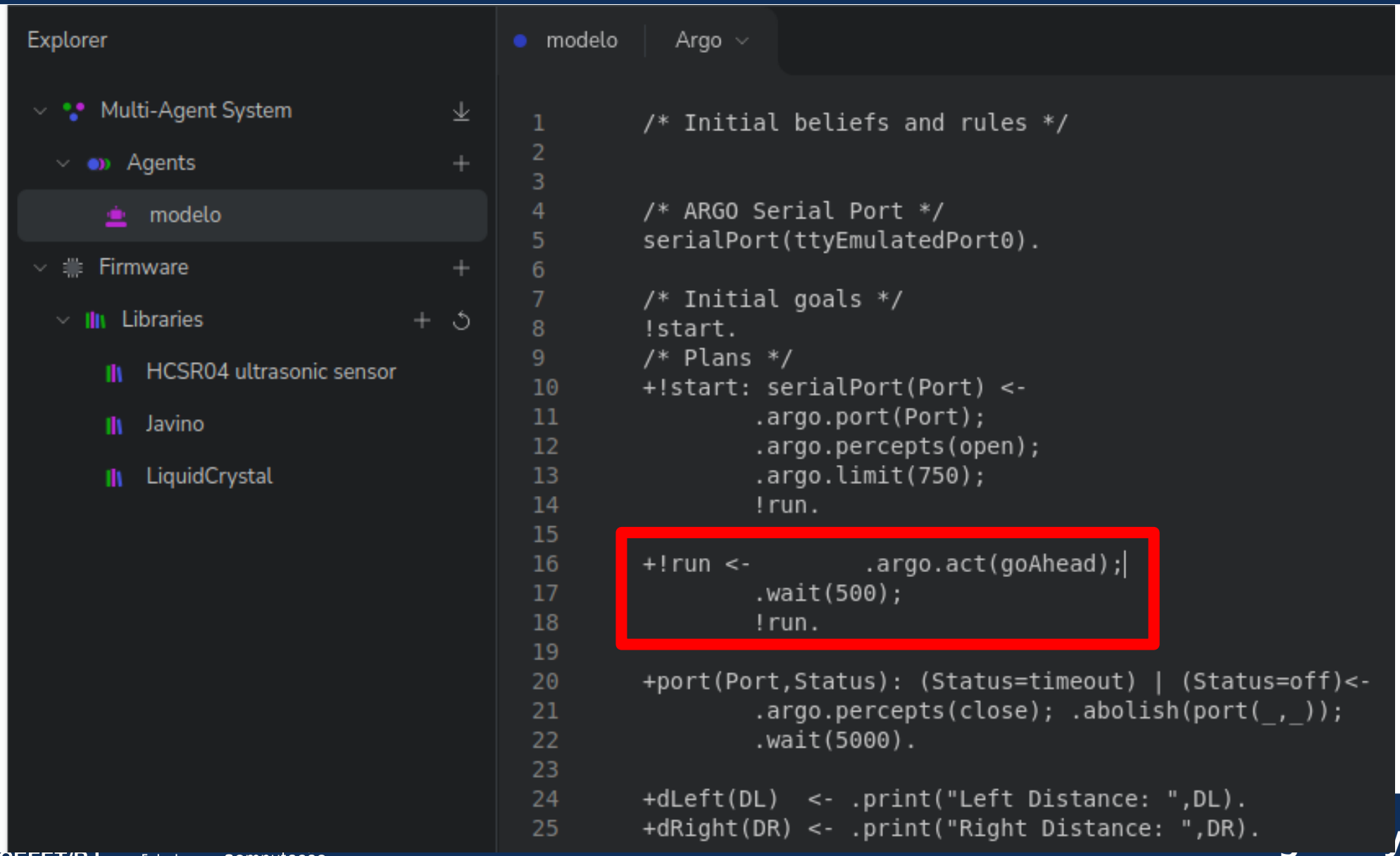
Porta serial emulada

Exemplo: protótipo ChonBot



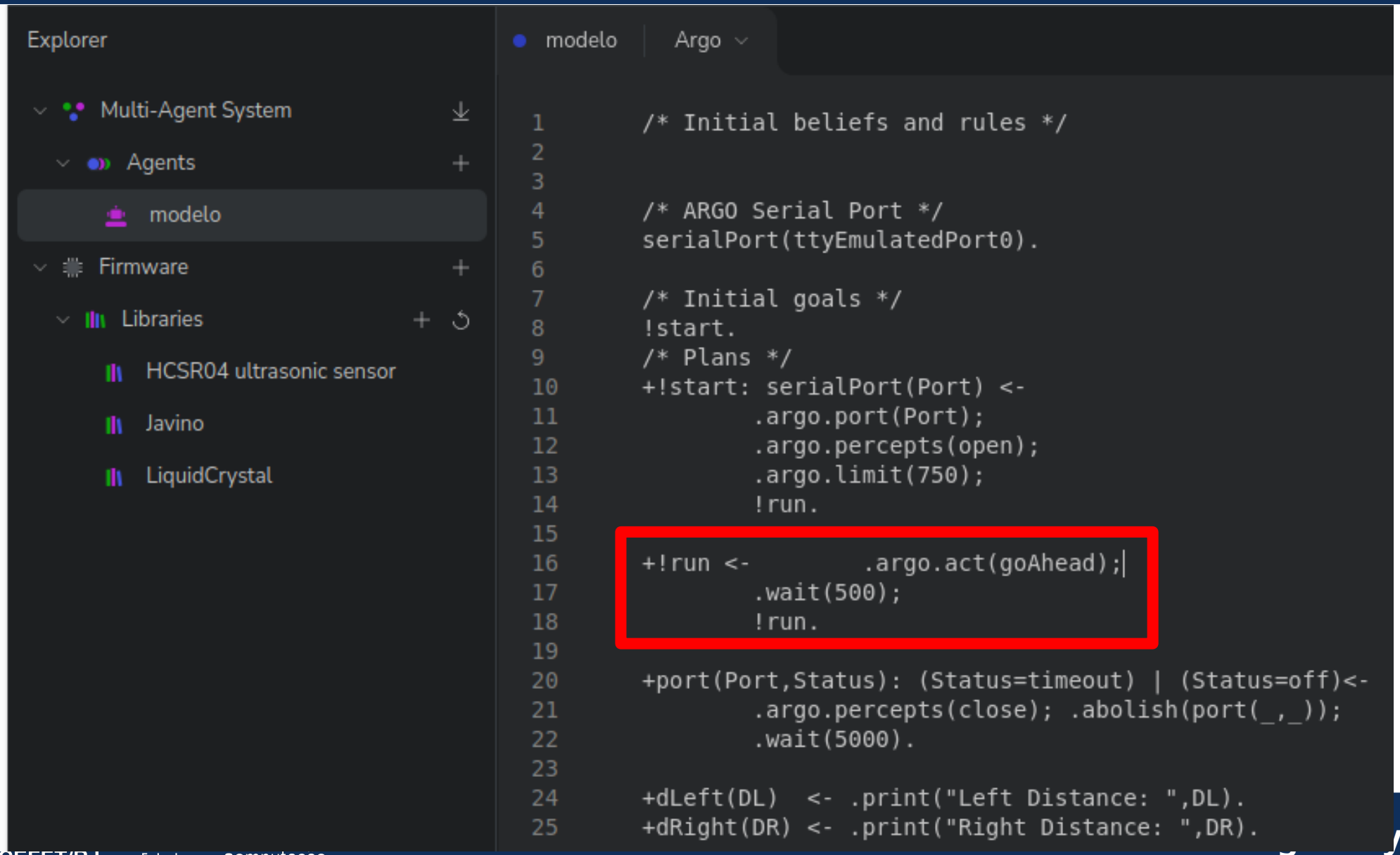
Plano *start* para iniciar captura de percepções

Exemplo: protótipo ChonBot



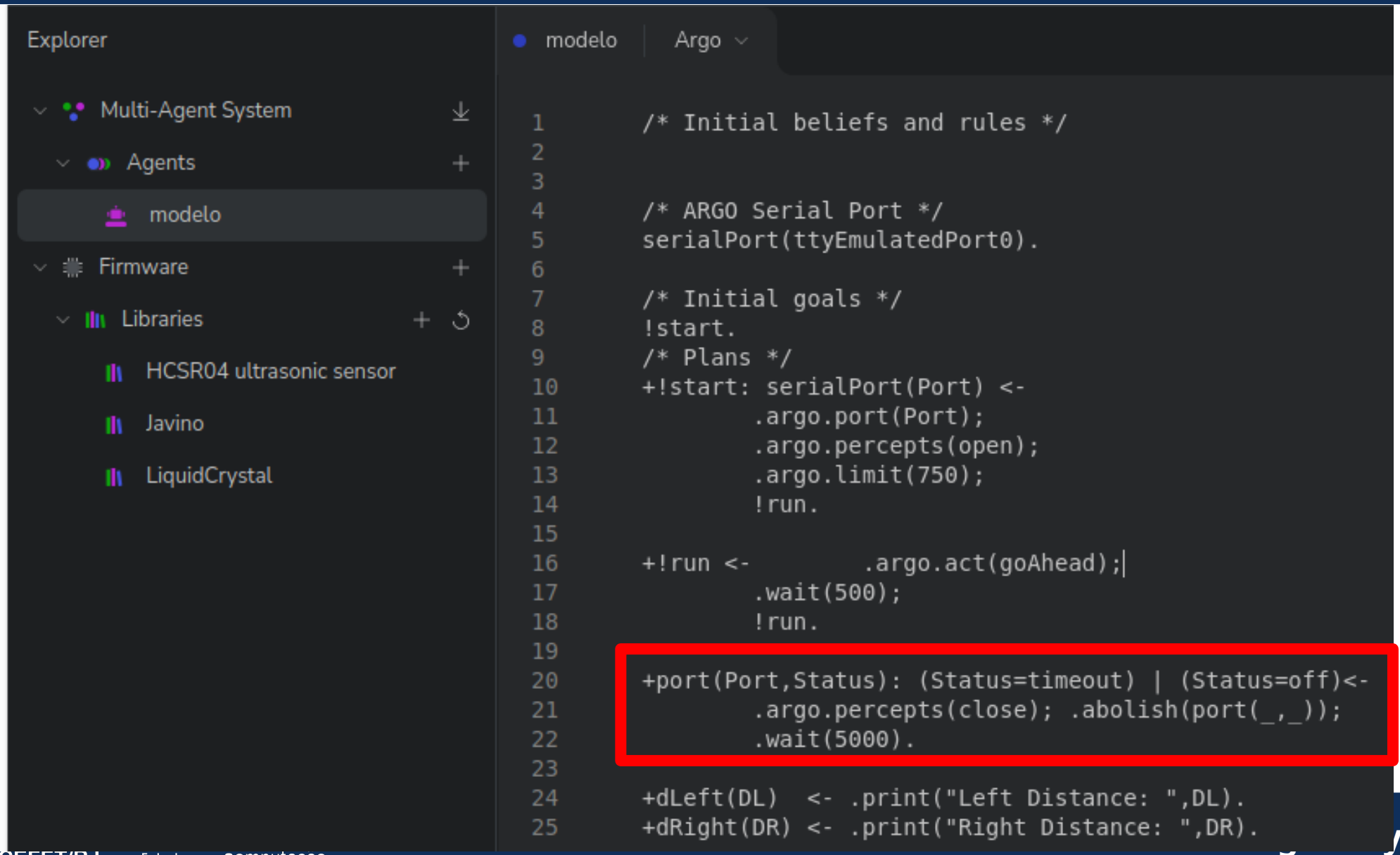
Plano *start* para iniciar captura de percepções

Exemplo: protótipo ChonBot



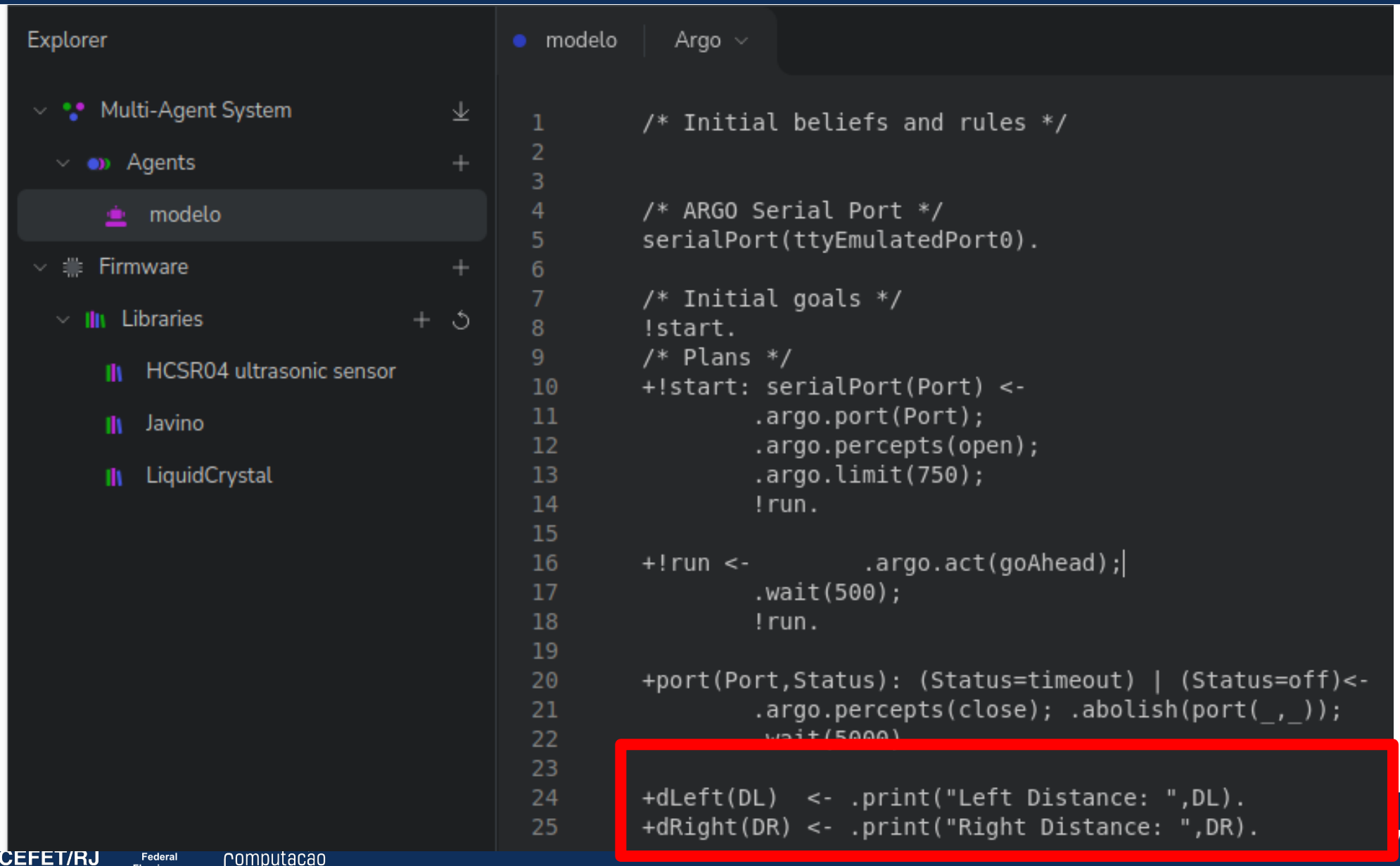
Plano *run*
Para ir para frente (somente)

Exemplo: protótipo ChonBot



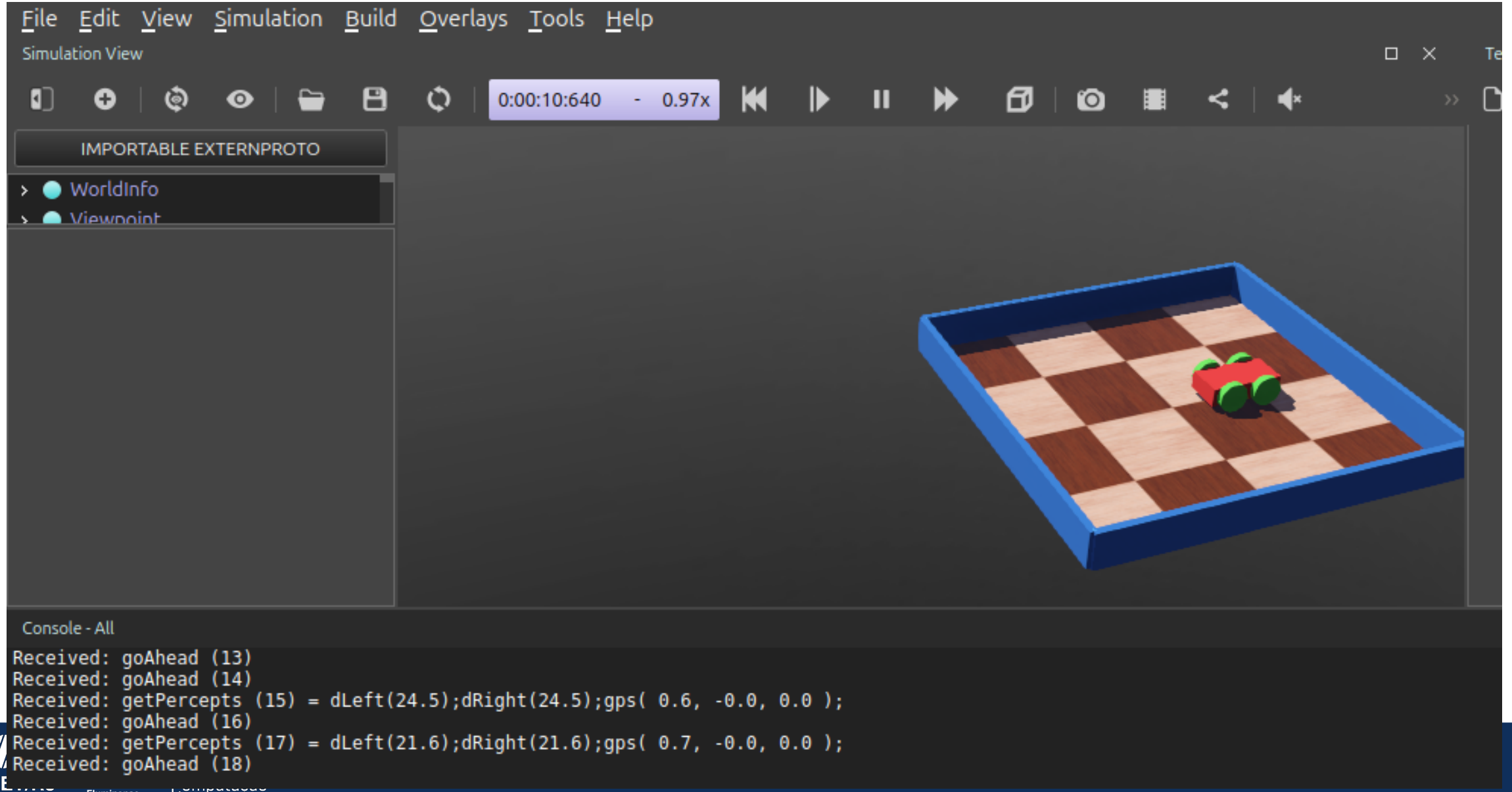
Plano de controle de erros

Exemplo: protótipo ChonBot



Print no log do SMA

Exemplo: protótipo ChonBot



Exemplo: protótipo ChonBot

Agradecimentos

OBRIGADO!

pantoja@cefet-rj.br
nilson.lazarin@cefet-rj.br