18th Workshop-School on Agents, Environments and Applications
Brasilia, DF, Brazil
August 14 - 16, 2024

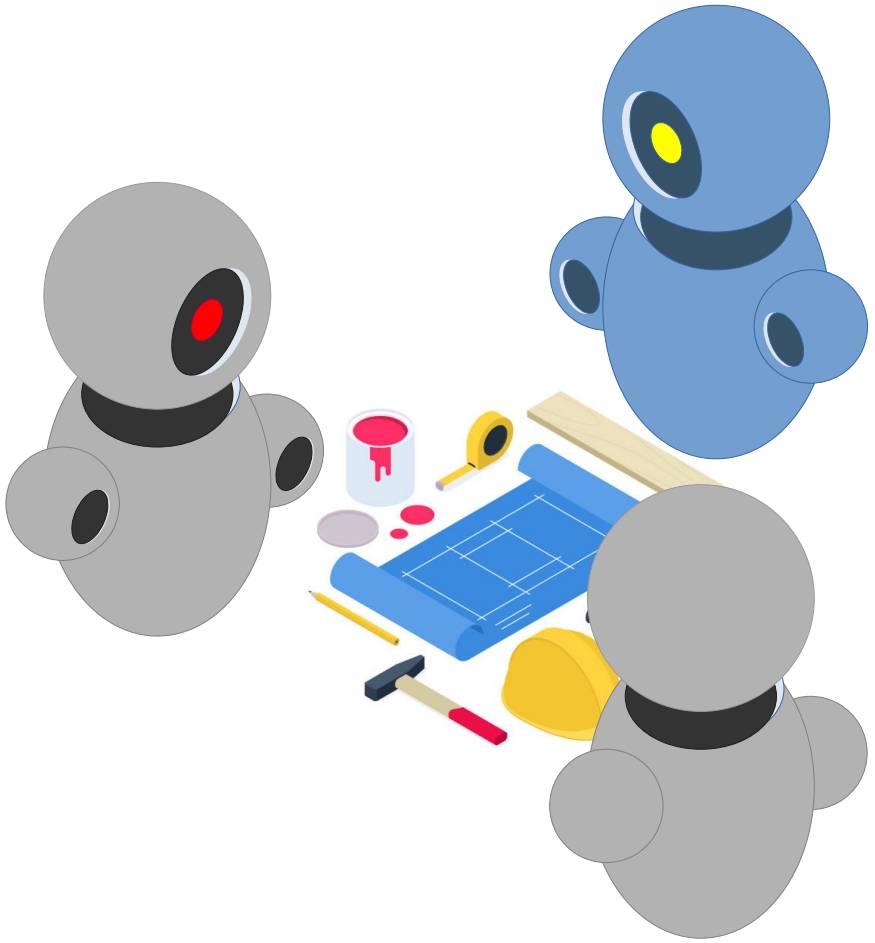
Introduction to Distributed and Embedded Multi-agent Systems

Carlos Eduardo Pantoja[1]
Nilson Mori Lazarin[1,2]

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

August 2024

INTRODUCTION



Who we are?

The **Cognitive Hardware on Networks** group or **CHON** Group.

Who we are?

The **Cognitive Hardware on Networks** group or **CHON** Group.

Prof.NILSON MORI LAZARIN

CEFET/RJ and UFF

Who we are?

The **Cognitive Hardware on Networks** group or
CHON Group.

Prof.NILSON MORI LAZARIN
CEFET/RJ and UFF

Prof. CARLOS PANTOJA
CEFET/RJ

Who we are?

The **Cognitive Hardware on Networks** group or
CHON Group.

Prof.NILSON MORI LAZARIN
CEFET/RJ and UFF

Prof. CARLOS PANTOJA
CEFET/RJ




Where we have been?

This workshop is part of:

Where we have been?

This workshop is part of:

- **2 undergratuate courses** (CEFET/RJ Maria da Graça and Nova Friburgo);

Where we have been?

This workshop is part of:

- **2 undergratuate courses** (CEFET/RJ Maria da Graça and Nova Friburgo);
- **1 graduate course** (M.Sc. and Ph.D. at UFF);

Where we have been?

This workshop is part of:

- **2 undergratuate courses** (CEFET/RJ Maria da Graça and Nova Friburgo);
- **1 graduate course** (M.Sc. and Ph.D. at UFF);
- It was **presented at**: UTFPR, IFSC, IFC, CEFET/RJ, UFRJ, UFF, and Estácio;

Where we have been?

This workshop is part of:

- **2 undergratuate courses** (CEFET/RJ Maria da Graça and Nova Friburgo);
- **1 graduate course** (M.Sc. and Ph.D. at UFF);
- It was **presented at**: UTFPR, IFSC, IFC, CEFET/RJ, UFRJ, UFF, and Estácio;
- It was also **presented at**: WESAAC'22, WESAAC'23, and SEKE'23,

Goals

This workshop goals are:

Goals

This workshop goals are:

- **To experience** the development process of **Embedded MAS** using BDI Agents and Jason.

Goals

This workshop goals are:

- **To experience** the development process of **Embedded MAS** using BDI Agents and Jason.
- to learn how to **collaborate** using IoT and embedded agents in **distributed** scenarios.

Goals

This workshop goals are:

- **To experience** the development process of **Embedded MAS** using BDI Agents and Jason.
- to learn how to **collaborate** using IoT and embedded agents in **distributed** scenarios.

… and the **most important** of all...

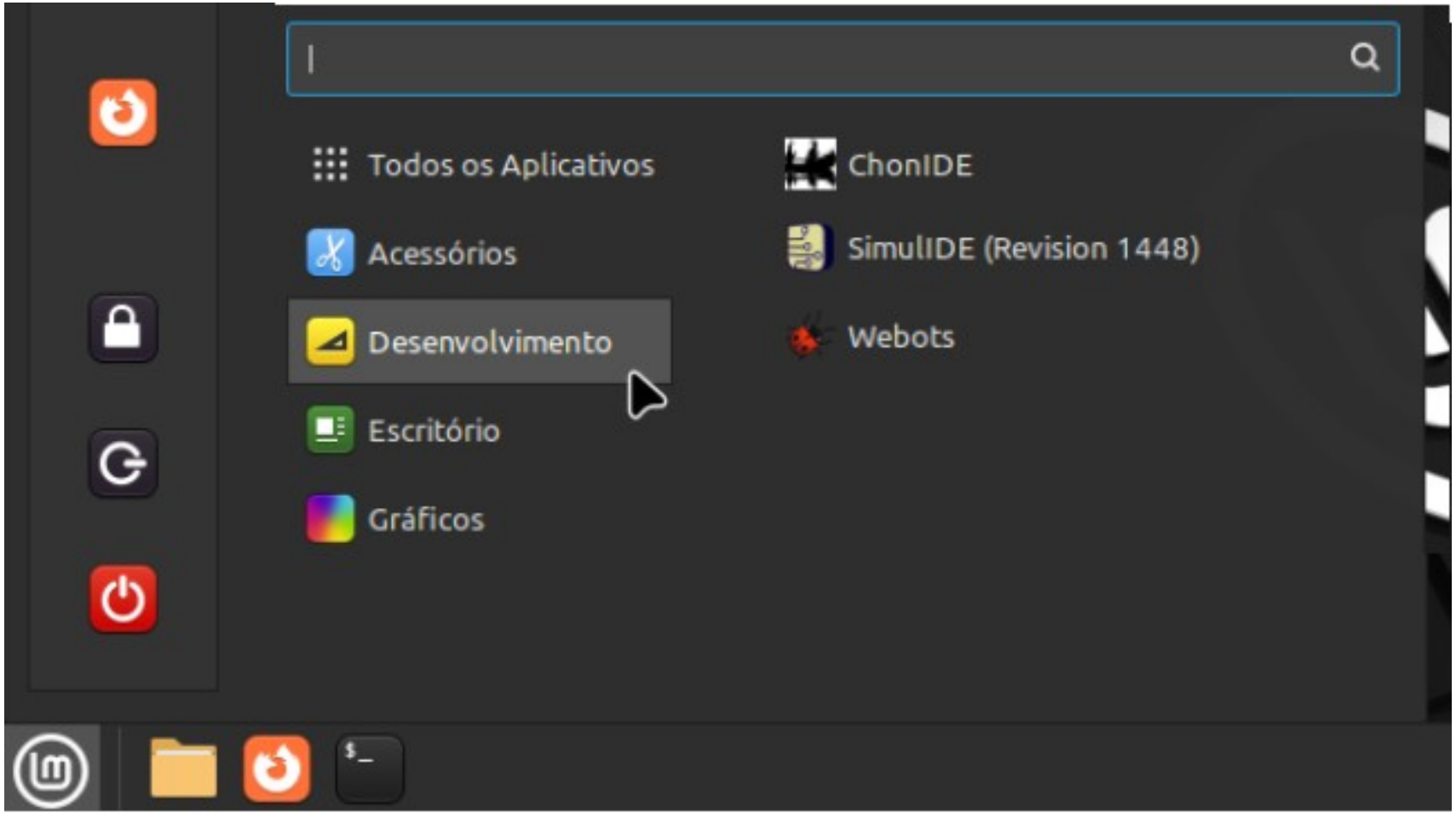
Goals

THE DEVELOPMENT TOOLS

ChonIDE

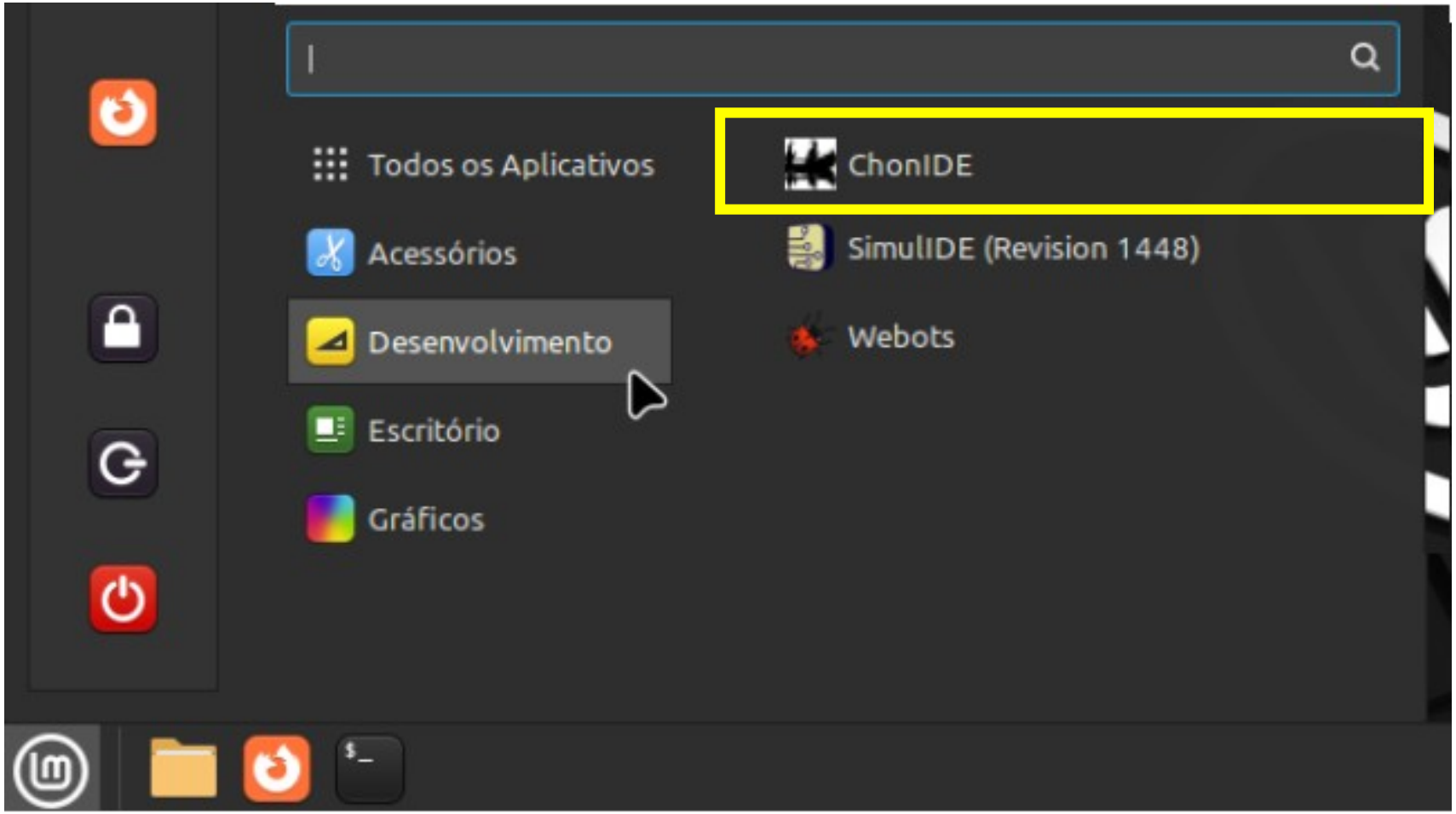
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE



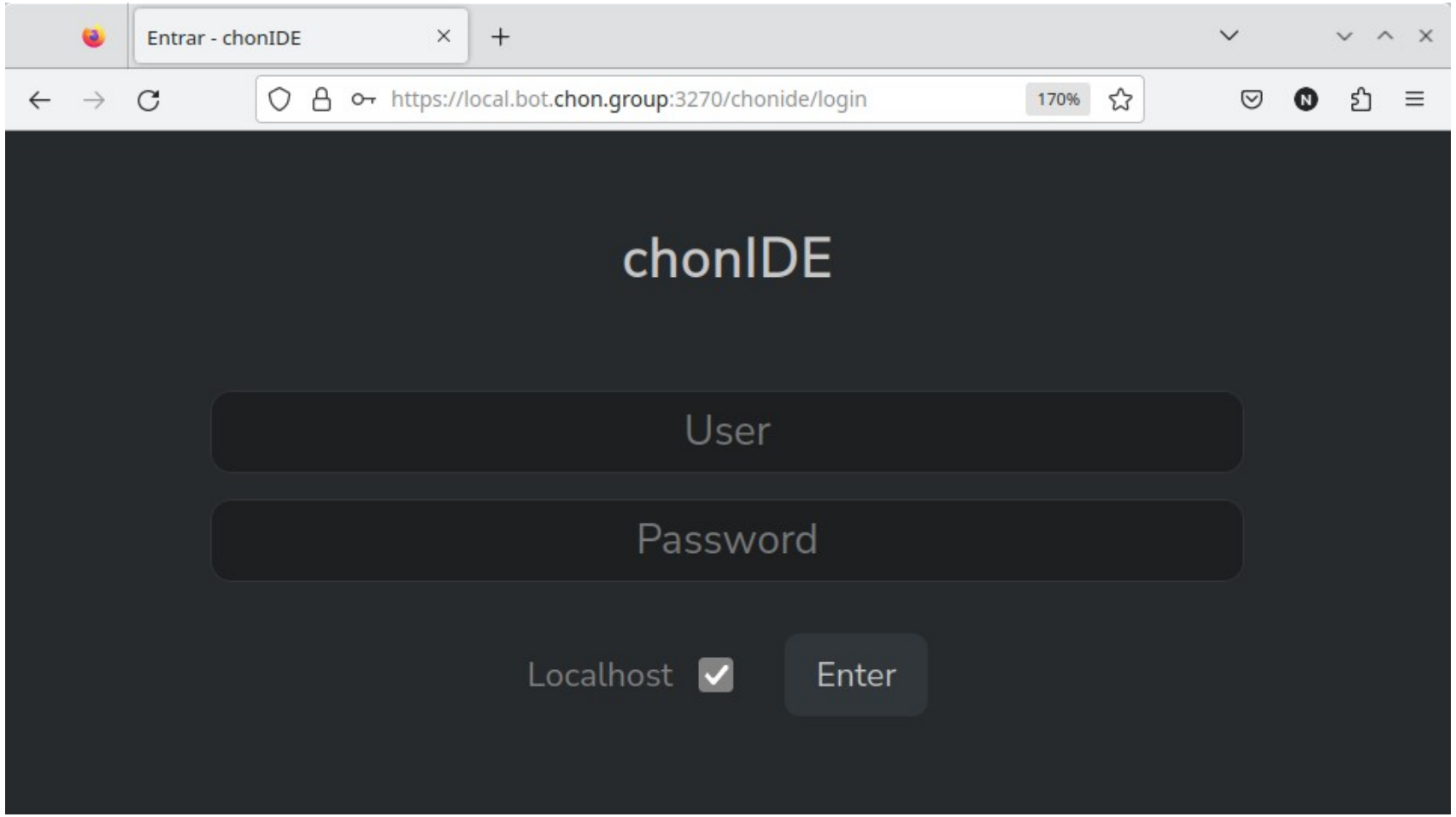
How to Install
https://github.com/chon-group/chonIDE/tree/main/doc

ChonIDE



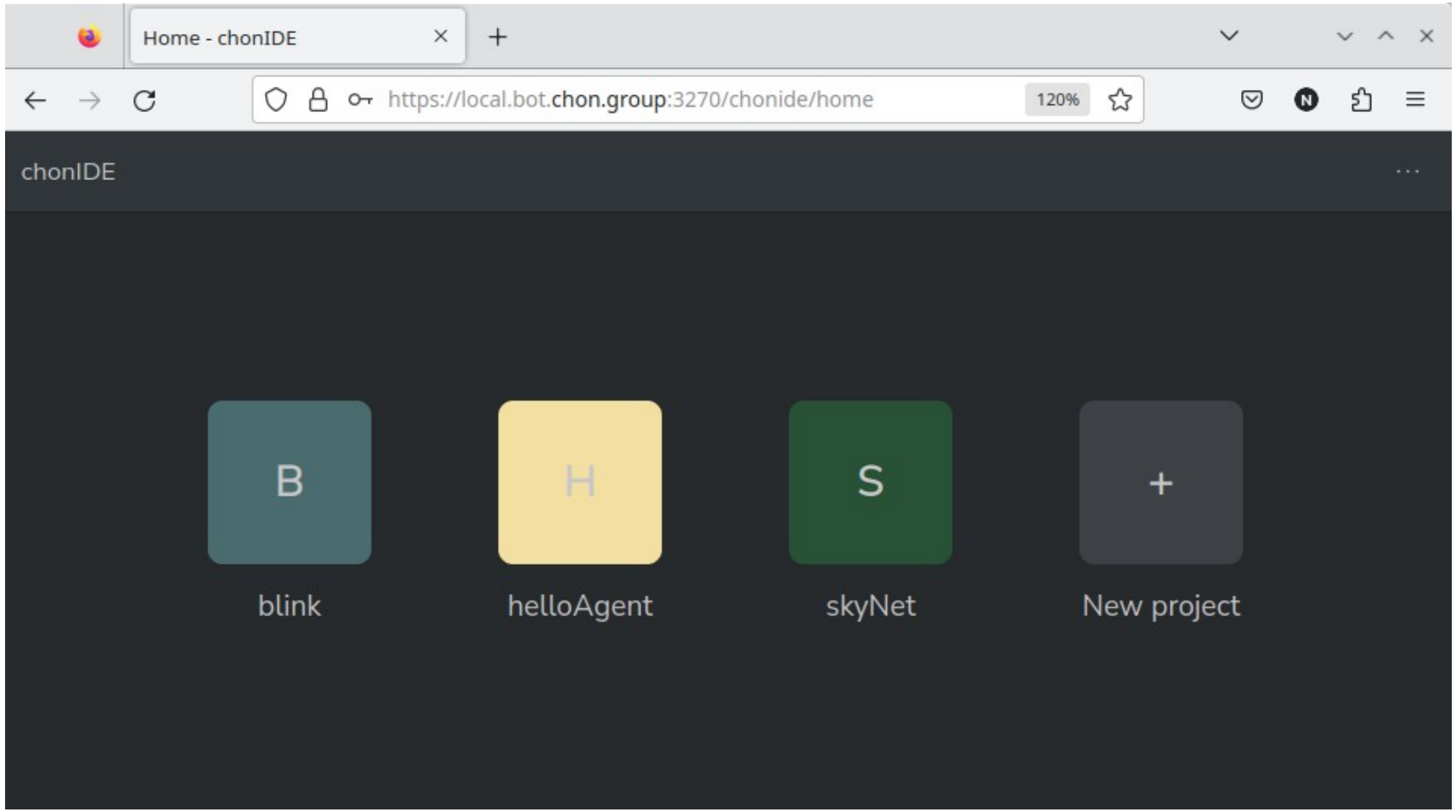
How to Install
https://github.com/chon-group/chonIDE/tree/main/doc

ChonIDE



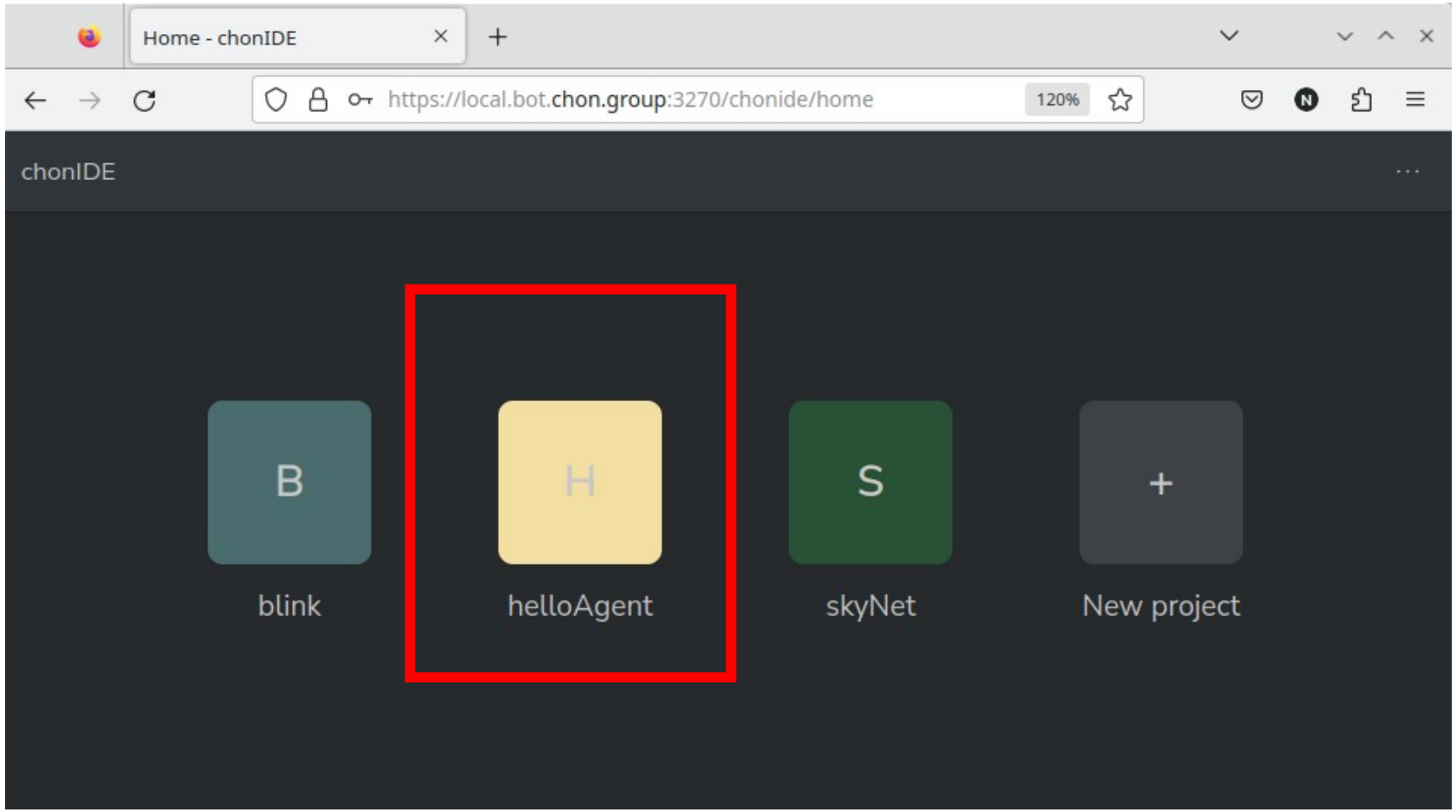
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



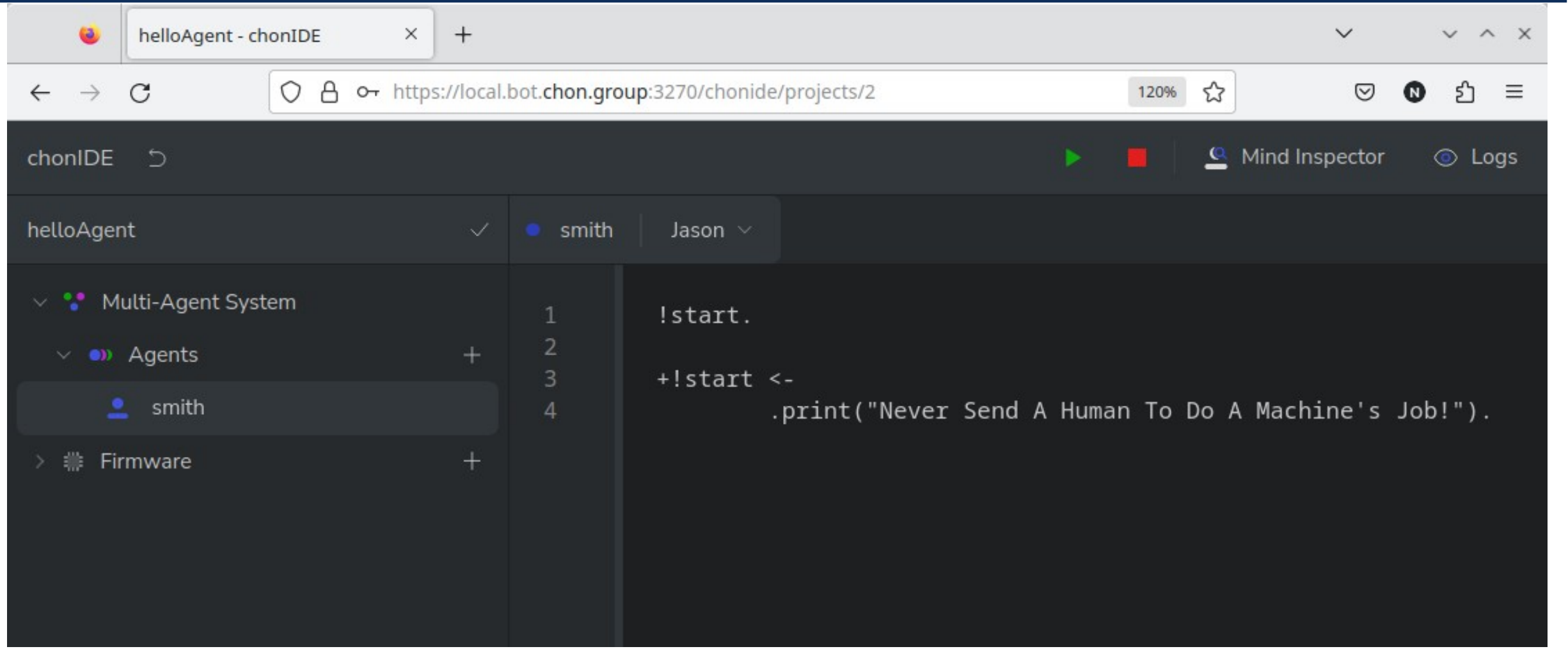
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



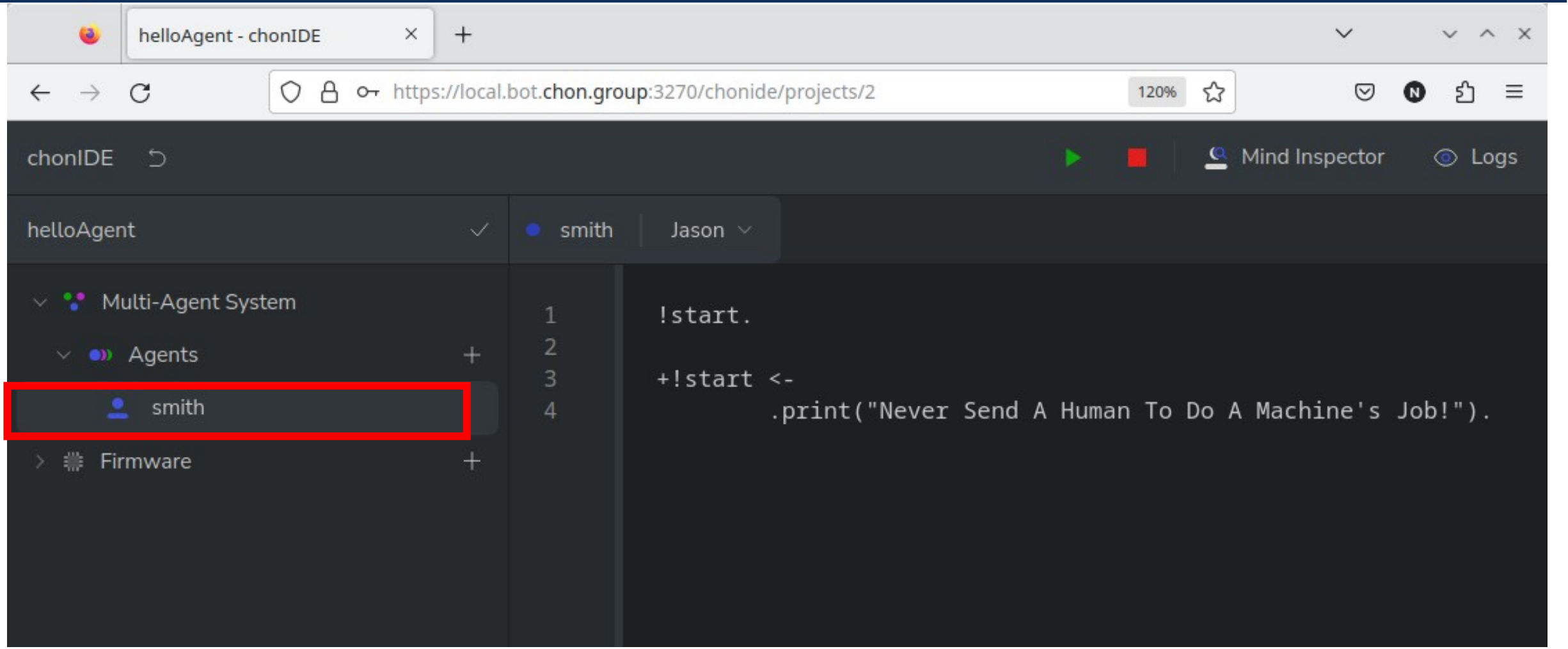
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



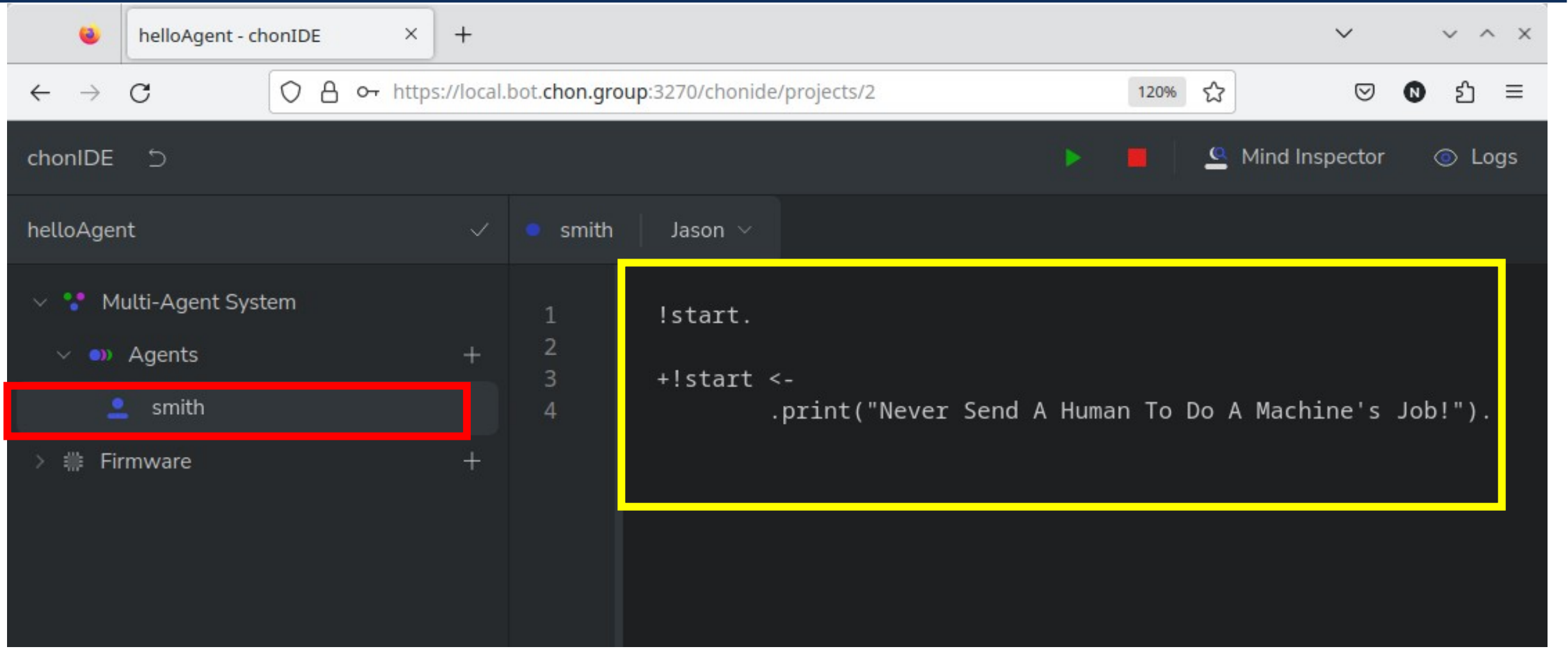
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



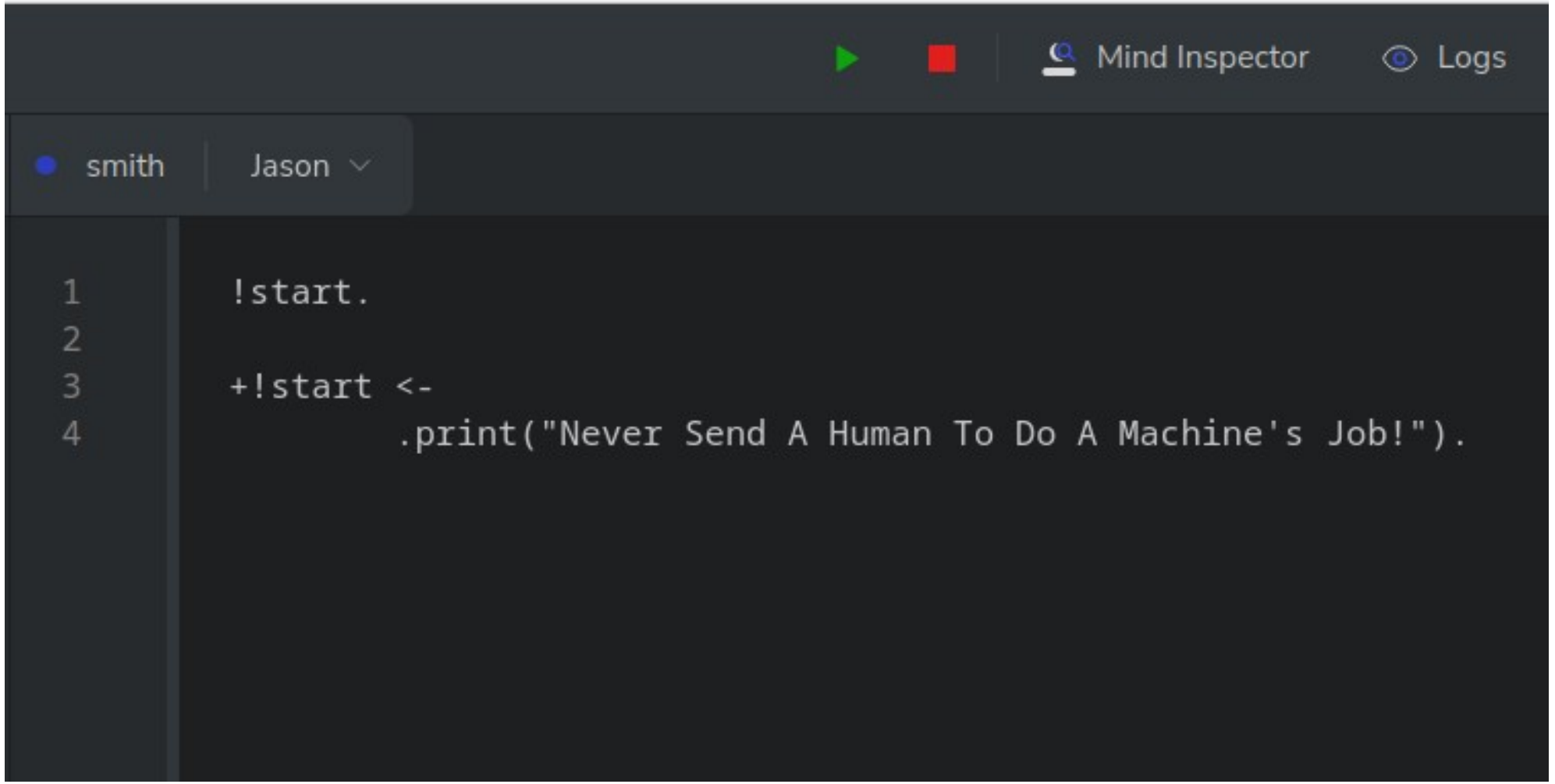
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



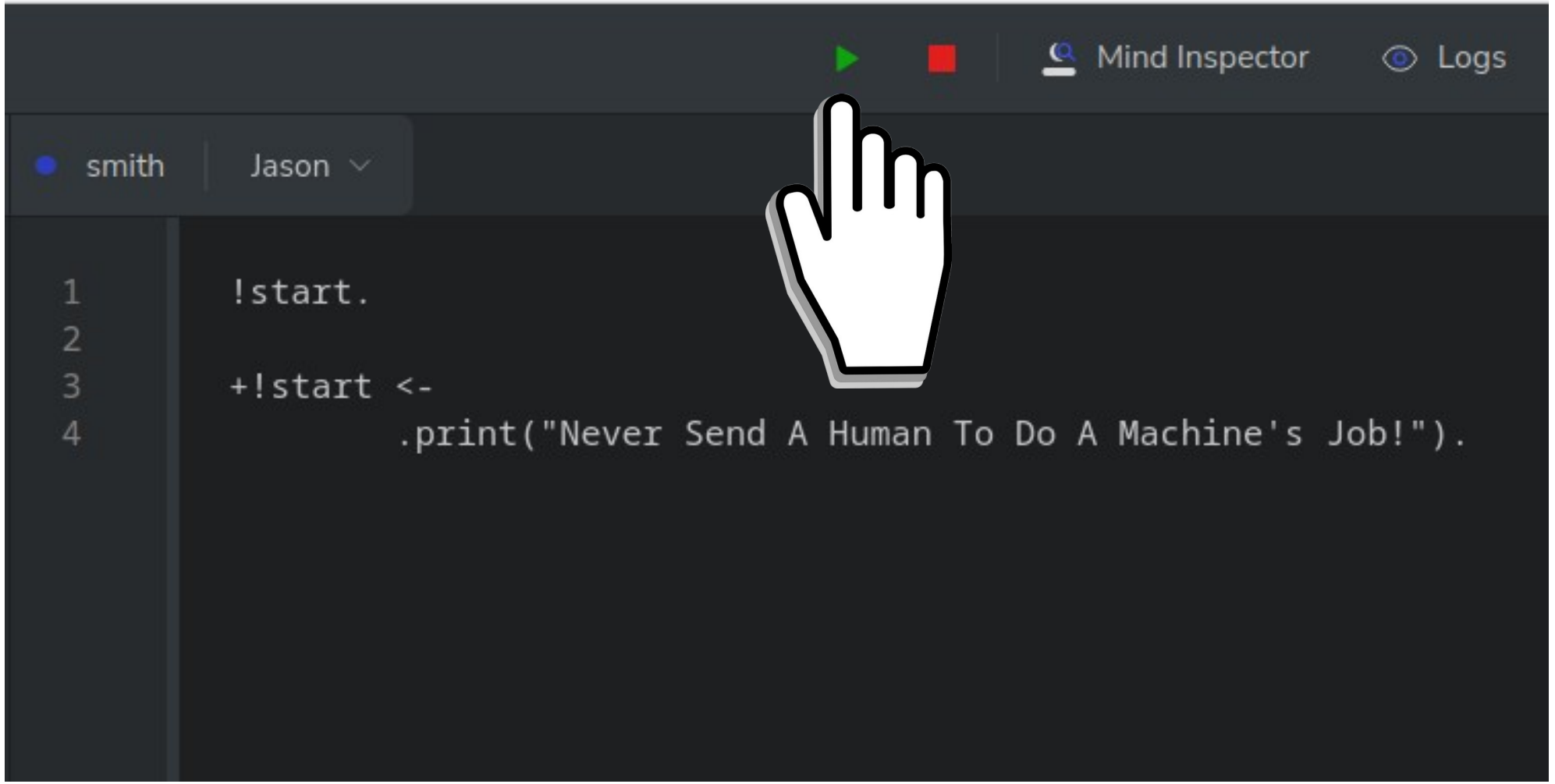
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



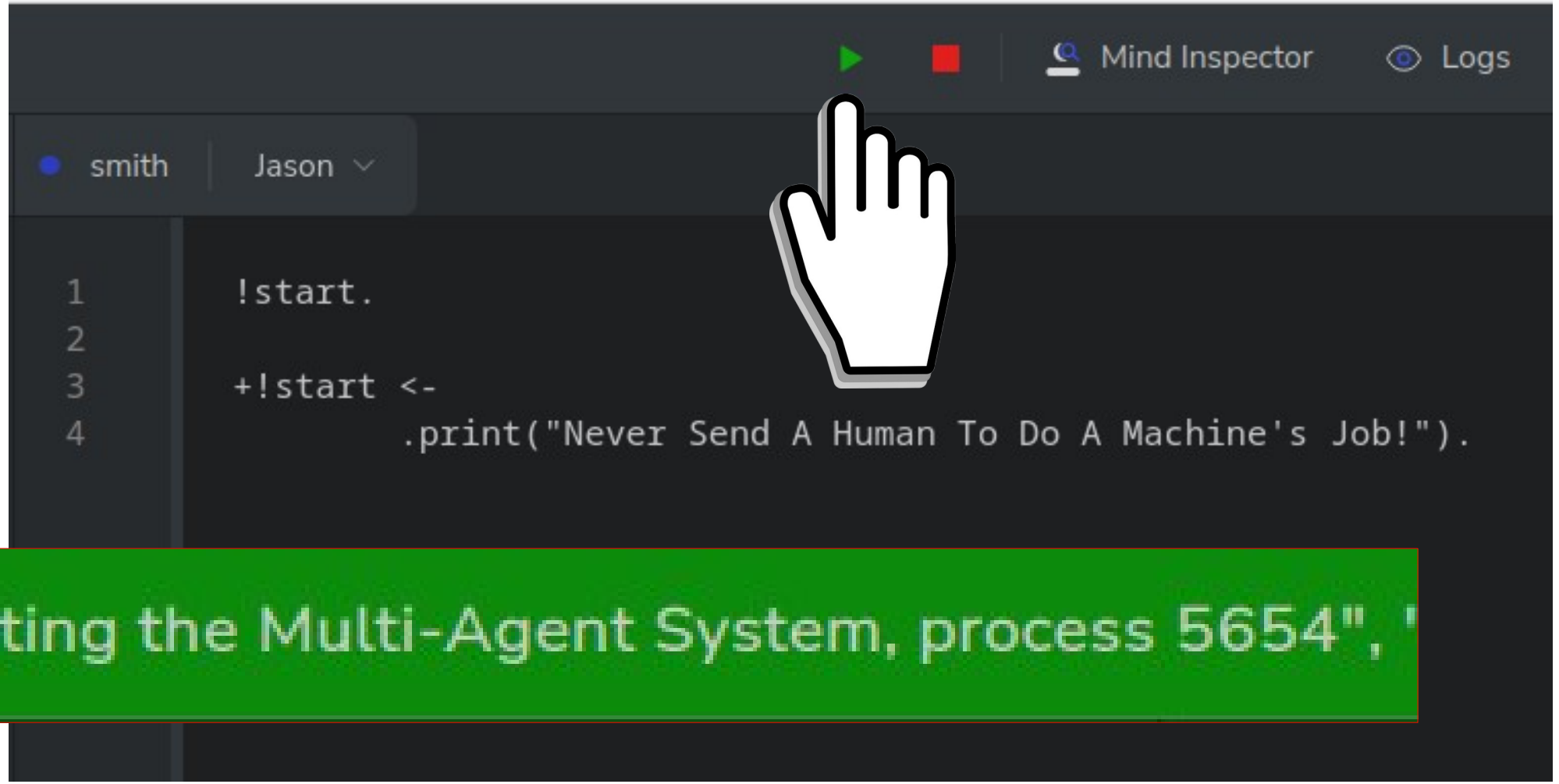
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



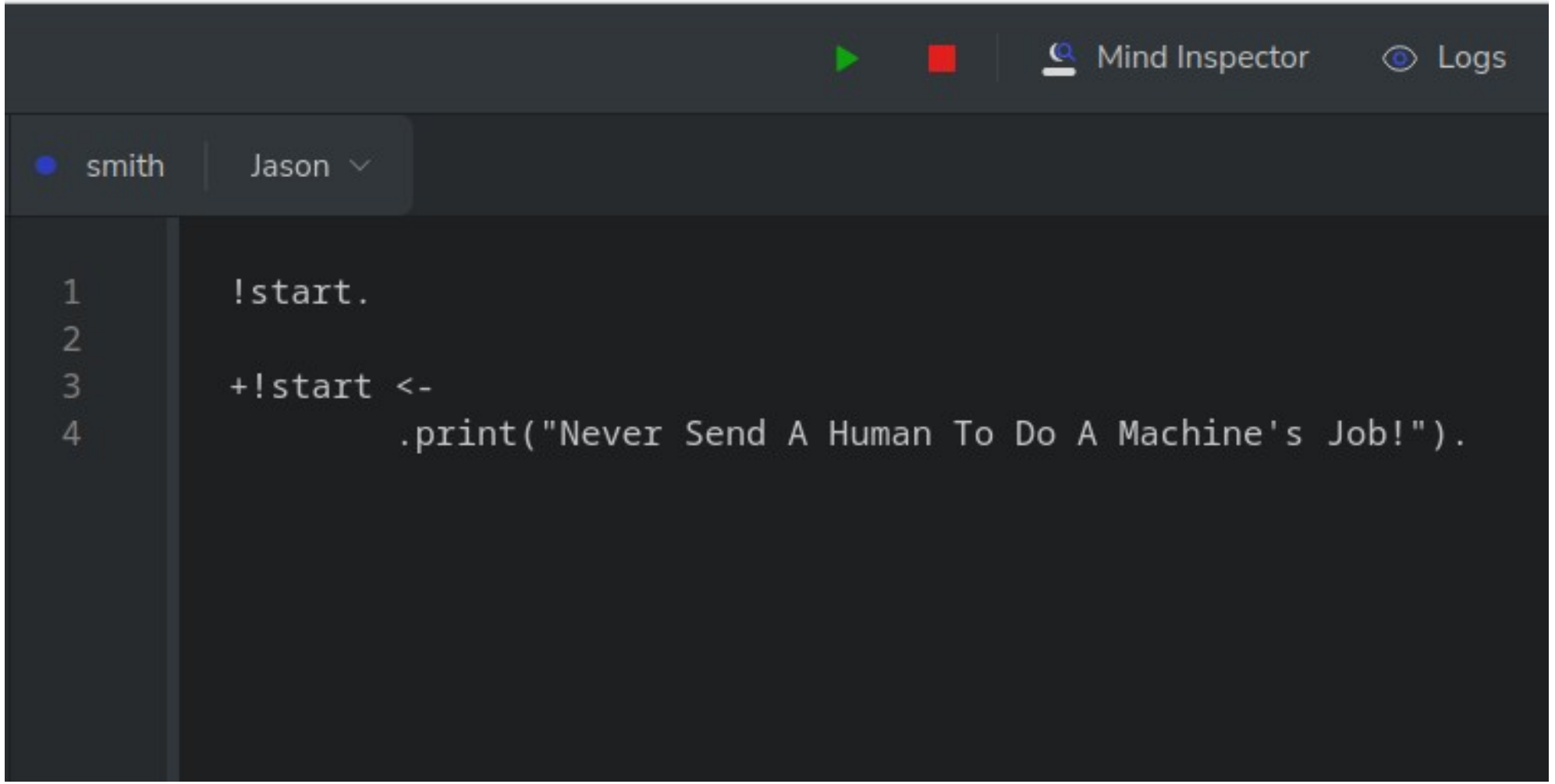
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



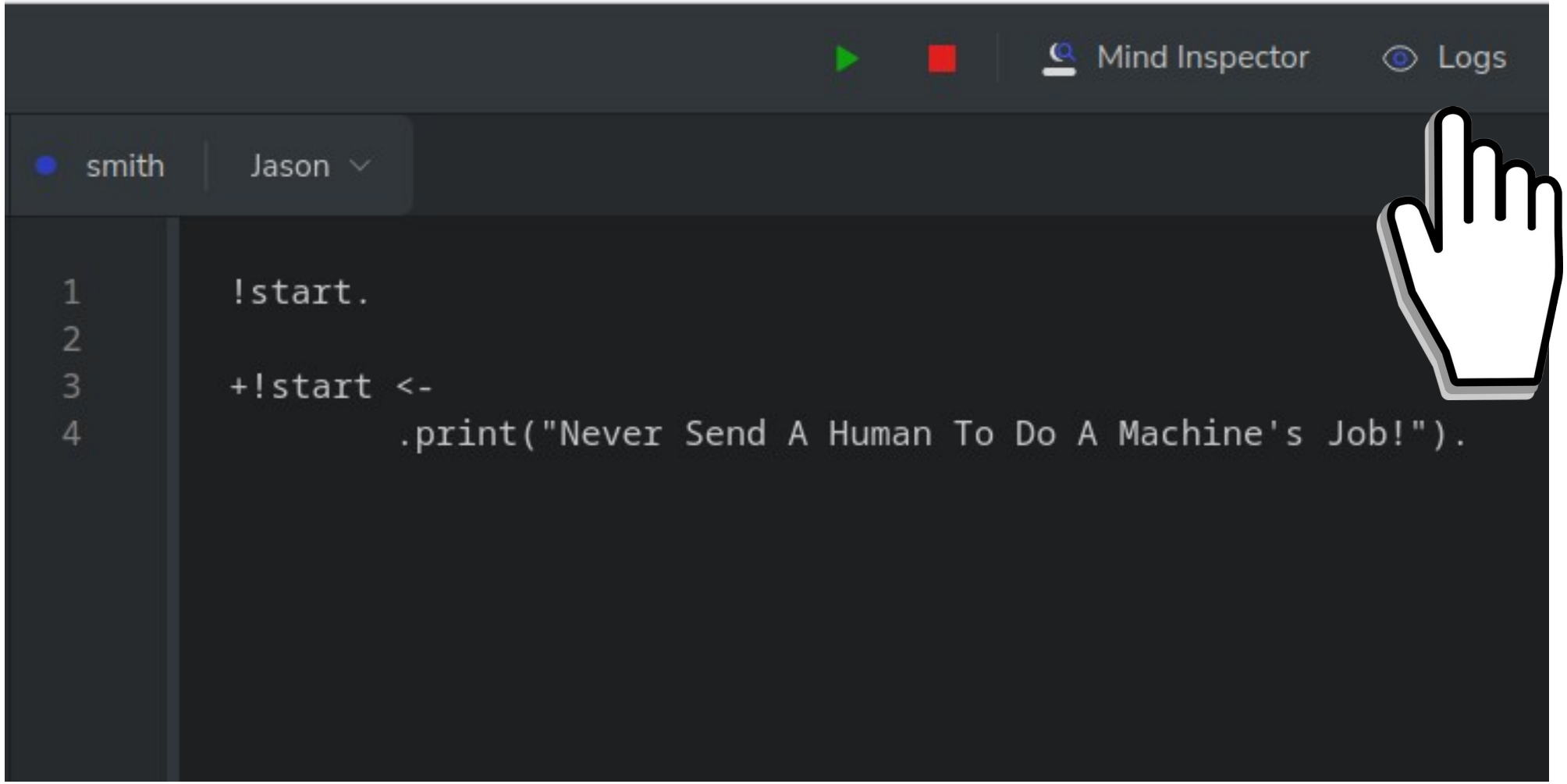
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



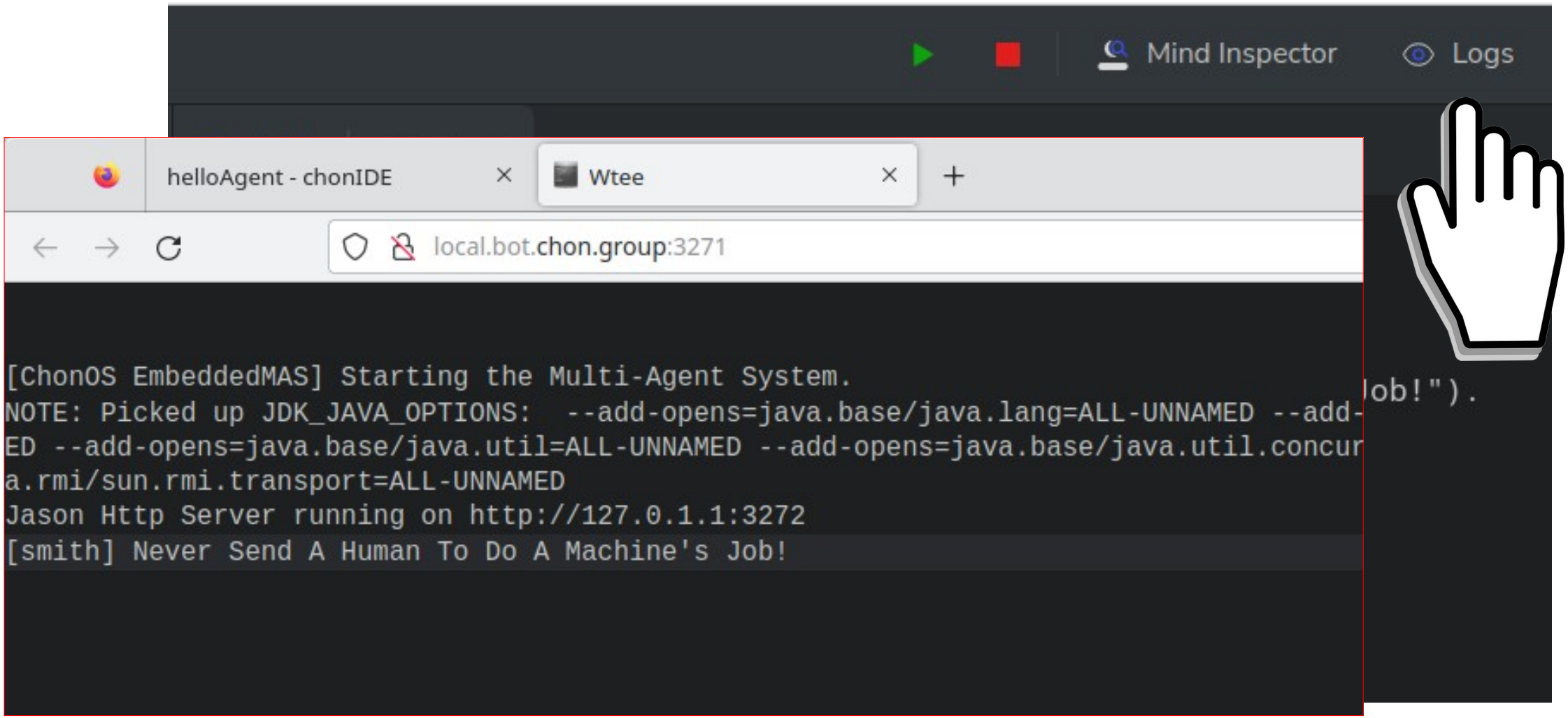
Pantoja, C.E., Jesus, V.S.d., Lazarin, N.M., Viterbo, J. (2023). A Spin-off Version of Jason for IoT and Embedded Multi-Agent Systems. In: Naldi, M.C., Bianchi, R.A.C. (eds) Intelligent Systems. BRACIS 2023. Lecture Notes in Computer Science(), vol 14195. Springer, Cham. https://doi.org/10.1007/978-3-031-45368-7_25

ChonIDE: helloAgent



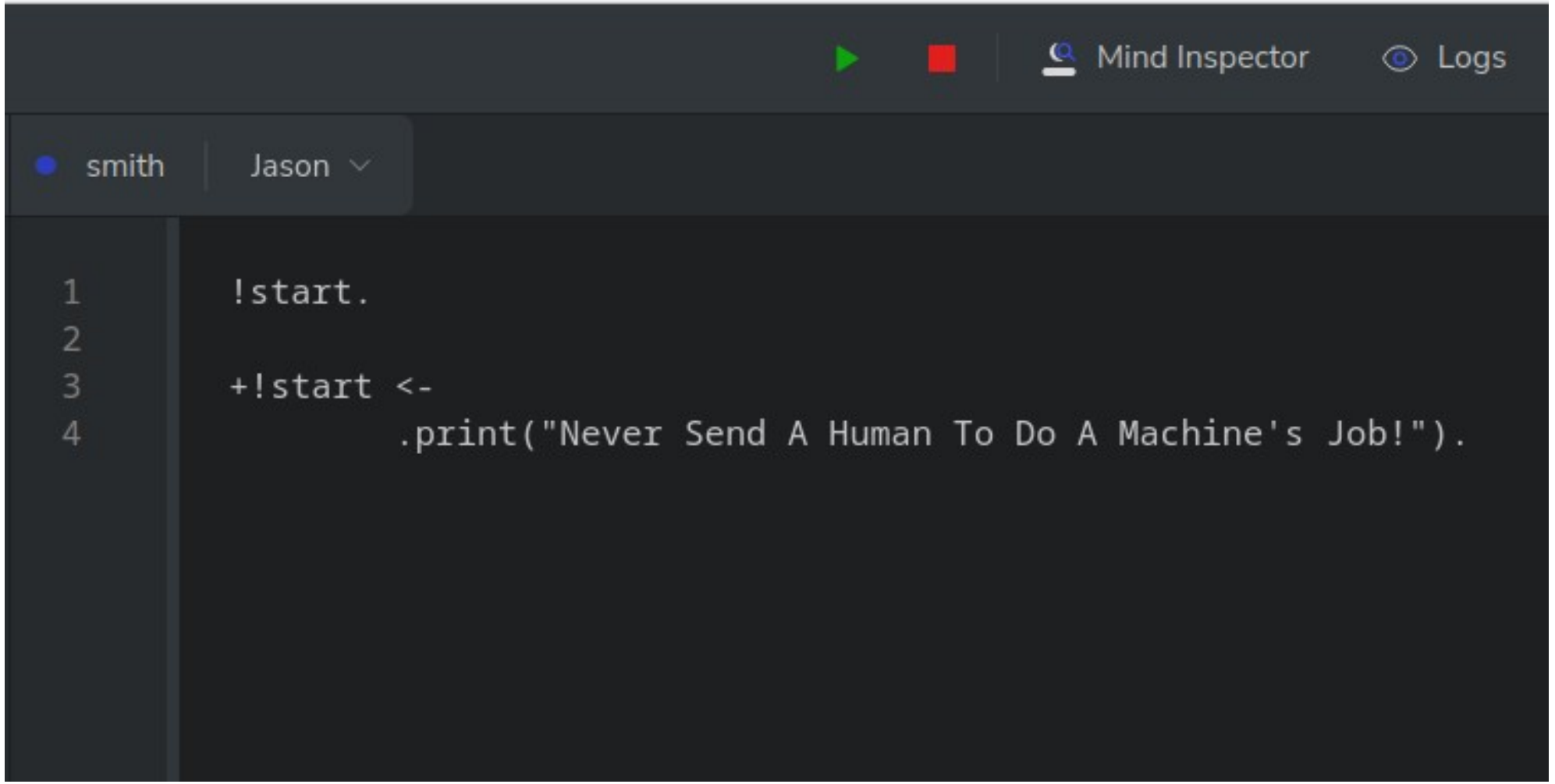
Pantoja, C.E., Jesus, V.S.d., Lazarin, N.M., Viterbo, J. (2023). A Spin-off Version of Jason for IoT and Embedded Multi-Agent Systems. In: Naldi, M.C., Bianchi, R.A.C. (eds) Intelligent Systems. BRACIS 2023. Lecture Notes in Computer Science(), vol 14195. Springer, Cham. https://doi.org/10.1007/978-3-031-45368-7_25

ChonIDE: helloAgent



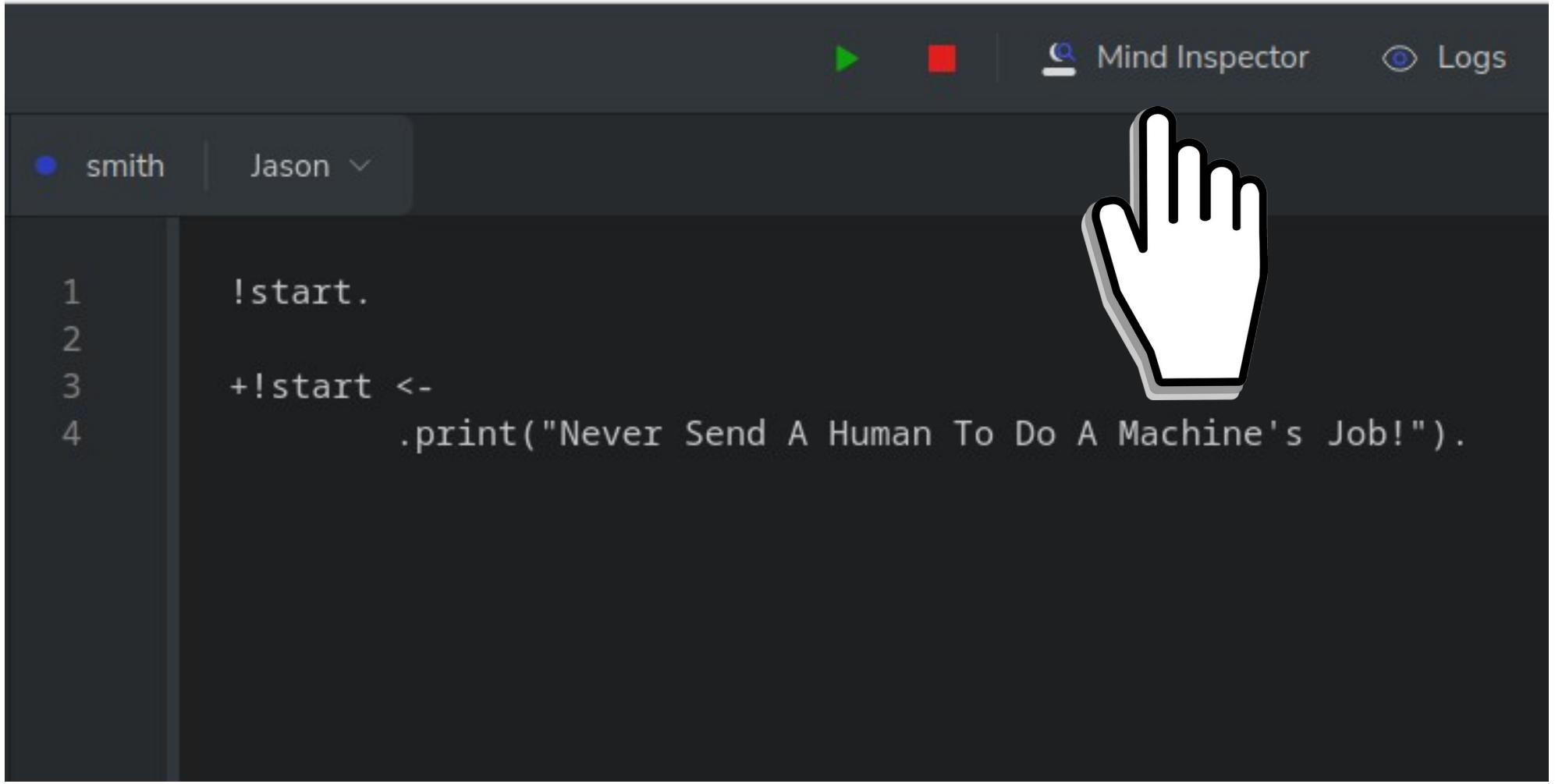
Pantoja, C.E., Jesus, V.S.d., Lazarin, N.M., Viterbo, J. (2023). A Spin-off Version of Jason for IoT and Embedded Multi-Agent Systems. In: Naldi, M.C., Bianchi, R.A.C. (eds) Intelligent Systems. BRACIS 2023. Lecture Notes in Computer Science(), vol 14195. Springer, Cham. https://doi.org/10.1007/978-3-031-45368-7_25

ChonIDE: helloAgent



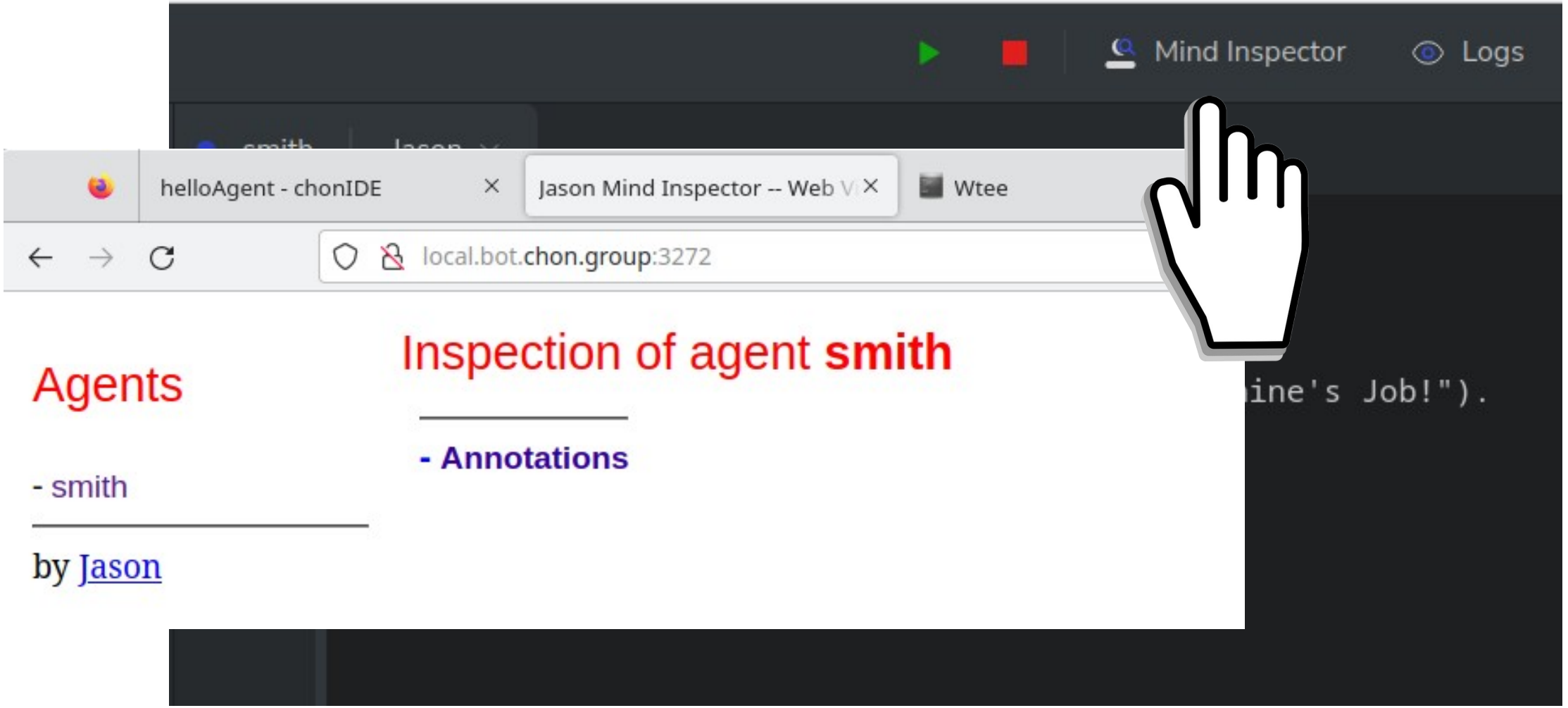
Bordini, R.H., Hübner, J.F. (2006). BDI Agent Programming in AgentSpeak Using Jason . In: Toni, F., Torroni, P. (eds) Computational Logic in Multi-Agent Systems. CLIMA 2005. Lecture Notes in Computer Science(), vol 3900. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11750734_9

ChonIDE: helloAgent



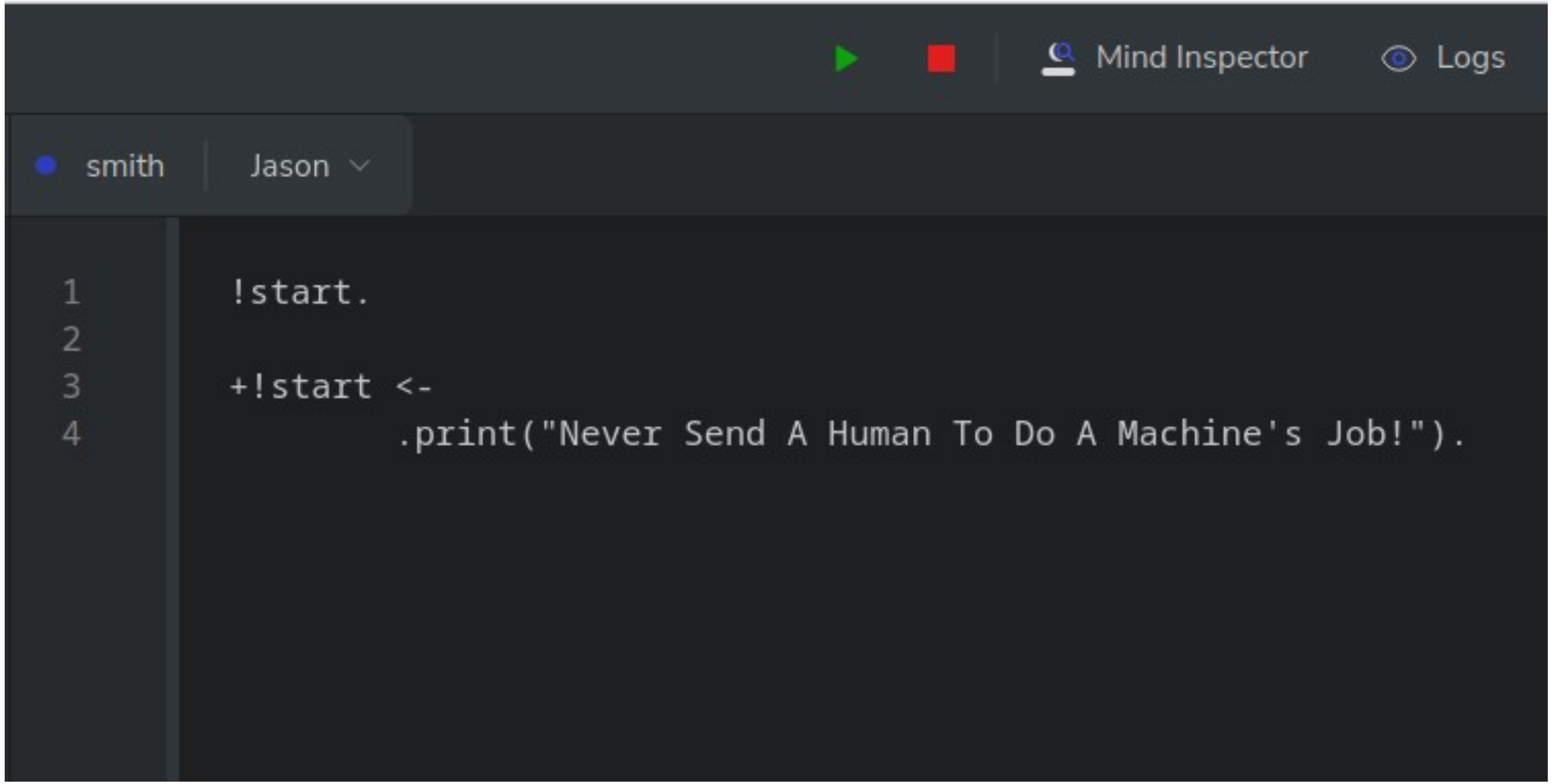
Bordini, R.H., Hübner, J.F. (2006). BDI Agent Programming in AgentSpeak Using Jason . In: Toni, F., Torroni, P. (eds) Computational Logic in Multi-Agent Systems. CLIMA 2005. Lecture Notes in Computer Science(), vol 3900. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11750734_9

ChonIDE: helloAgent



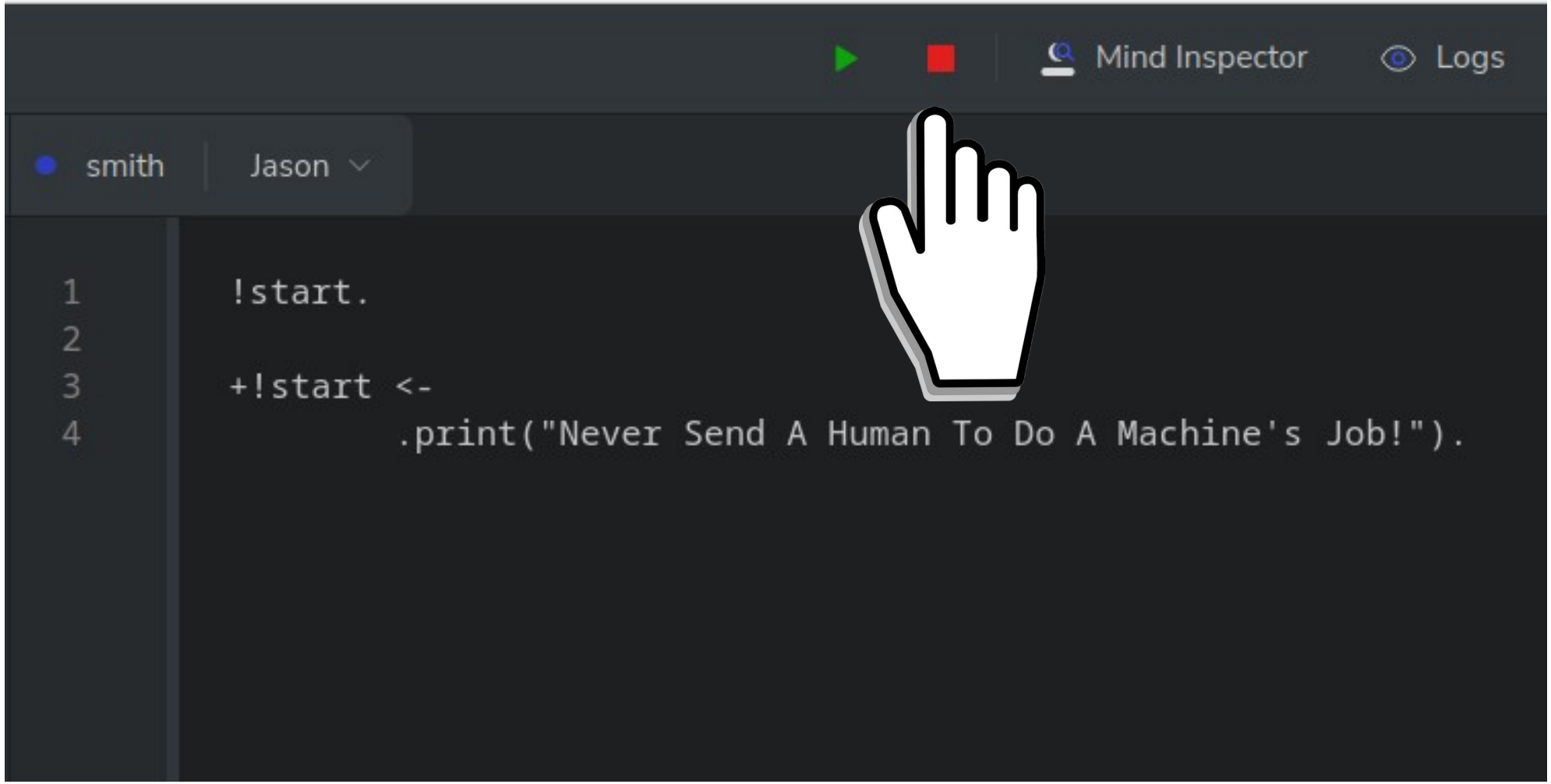
Bordini, R.H., Hübner, J.F. (2006). BDI Agent Programming in AgentSpeak Using Jason . In: Toni, F., Torroni, P. (eds) Computational Logic in Multi-Agent Systems. CLIMA 2005. Lecture Notes in Computer Science(), vol 3900. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11750734_9

ChonIDE: helloAgent



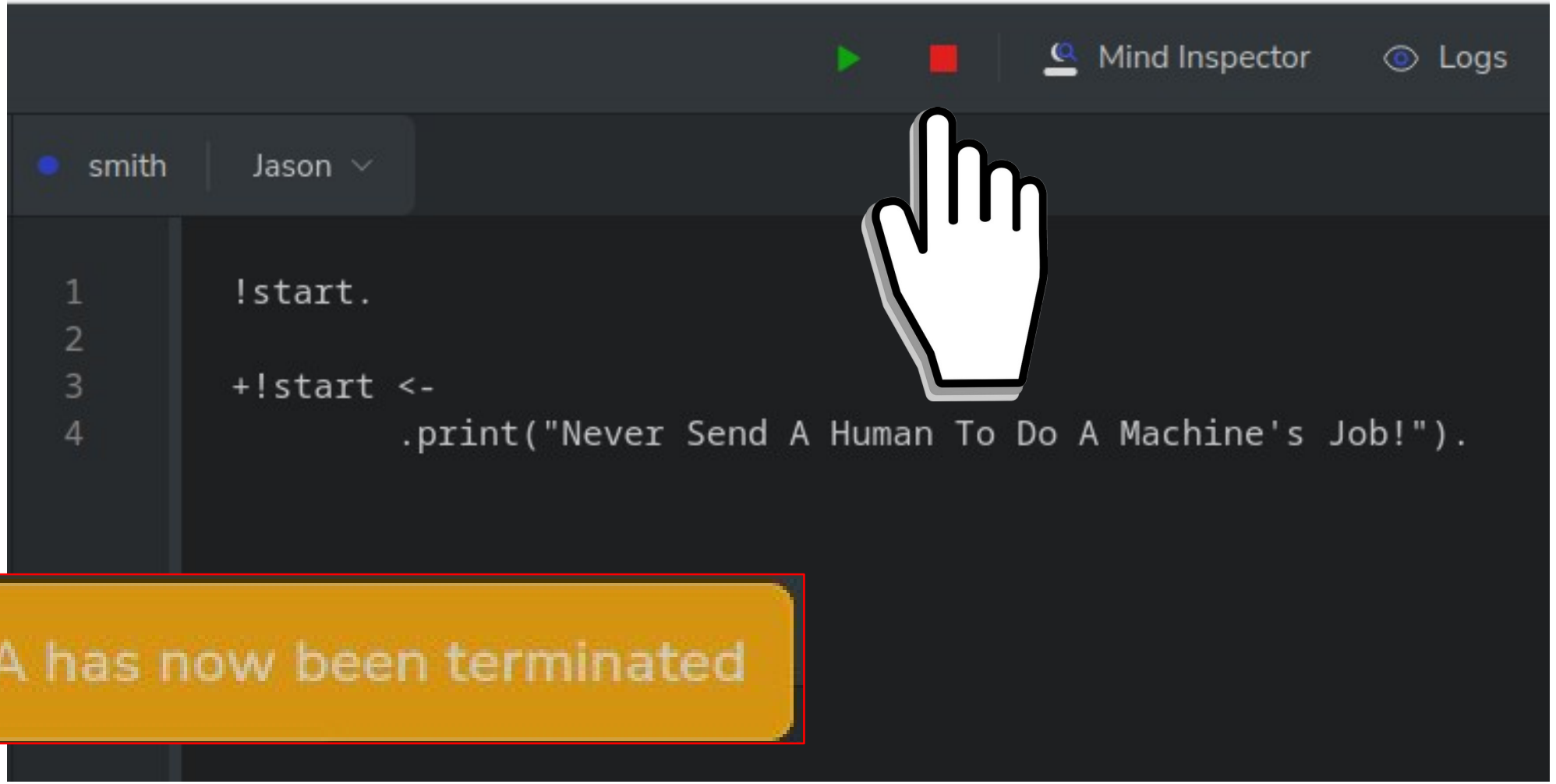
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent



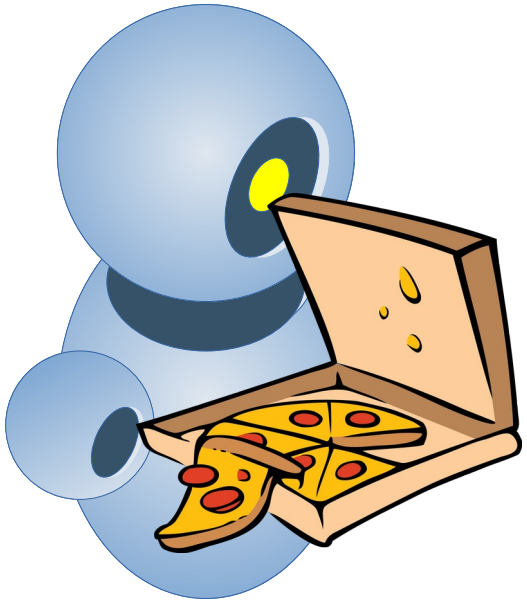
Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

ChonIDE: helloAgent

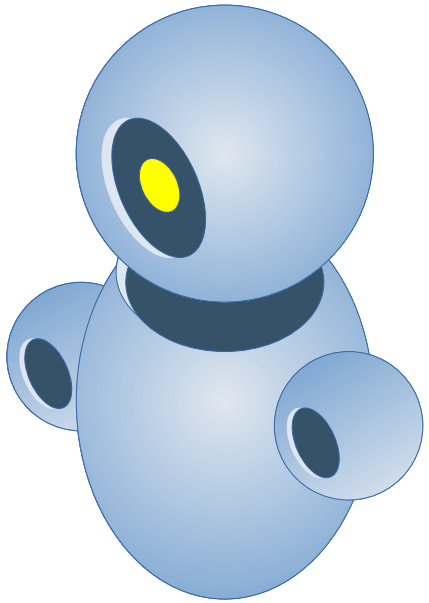


Souza de Jesus, V., Mori Lazarin, N., Pantoja, C.E., Vaz Alves, G., Ramos Alves de Lima, G., Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In: Mathieu, P., Dignum, F., Novais, P., De la Prieta, F. (eds) Advances in Practical Applications of Agents, Multi-Agent 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

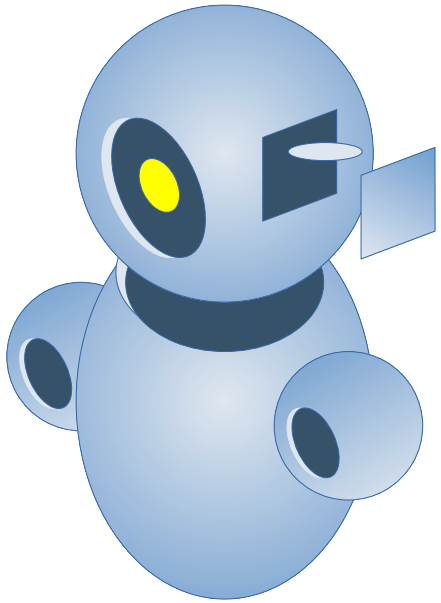
MY FIRST SINGLE AGENT



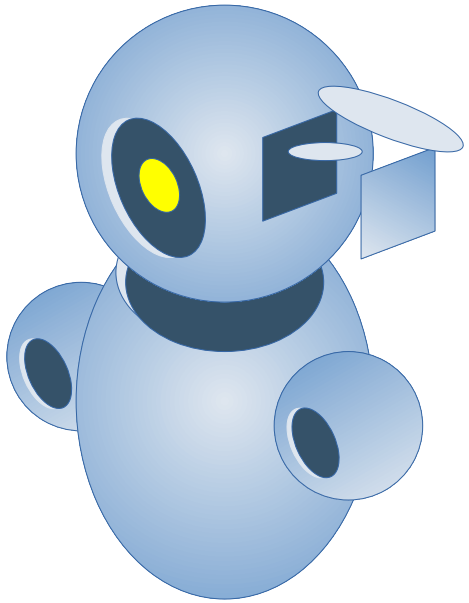
My First Agent



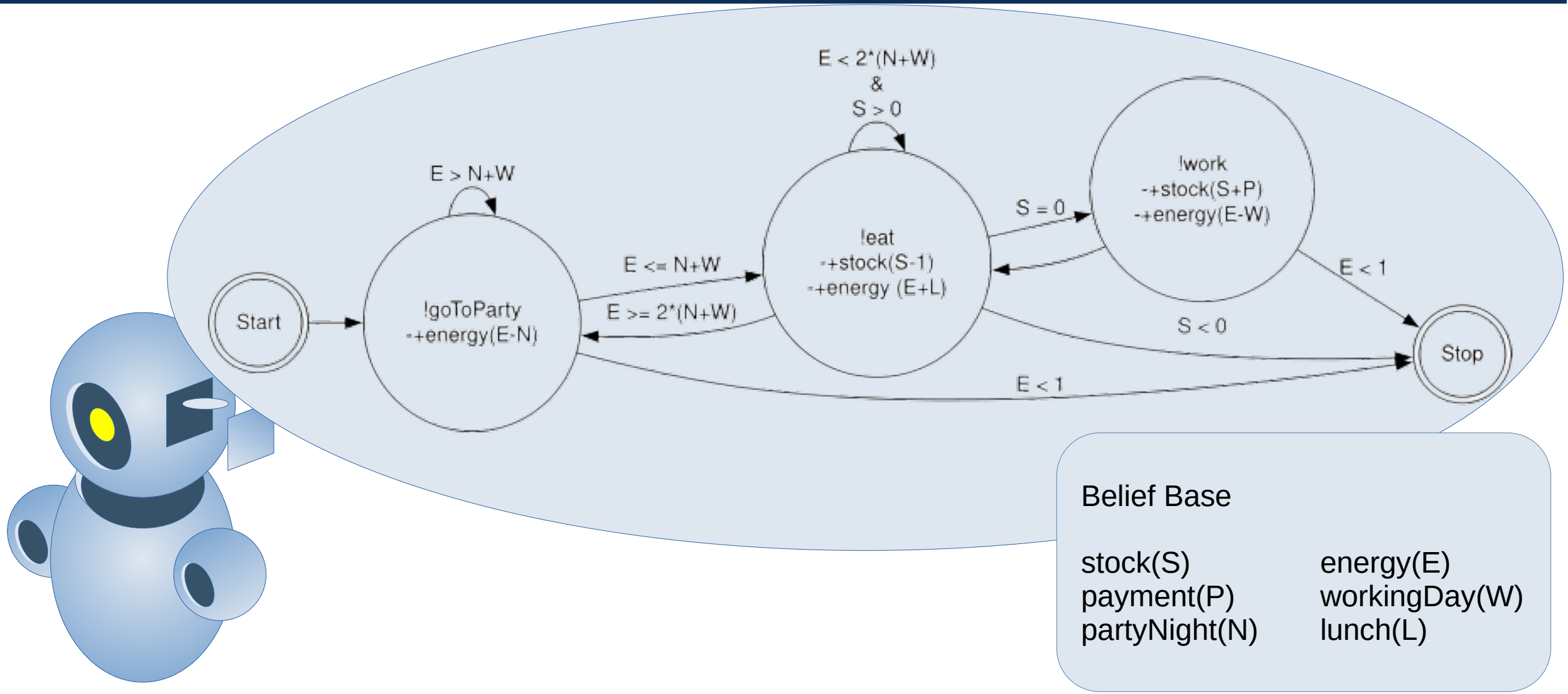
My First Agent



My First Agent



My First Agent



Belief Base

stock(S)	energy(E)
payment(P)	workingDay(W)
partyNight(N)	lunch(L)

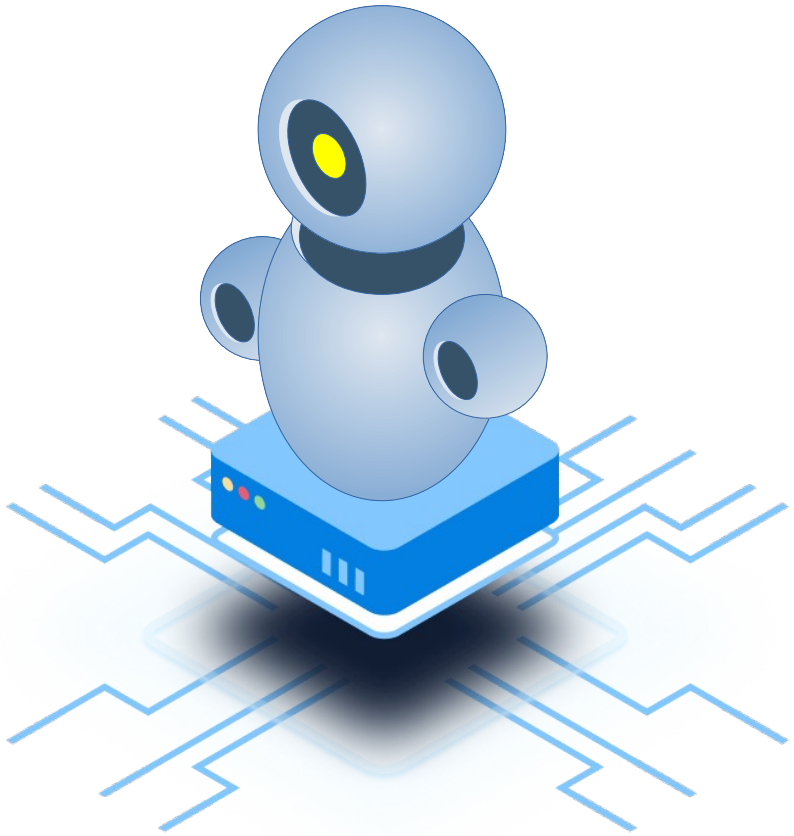
My First Agent: Code

asl > giacomo.asl

```
1   /* Initial beliefs and rules */
2   lifeParameters(S,E,P,W,N,L) :- stock(S) &  energy(E) & payment(P) & workingDay(W) & partyNight(N) & lunch(L).
3
4   stock(0).        // The pizza stock.
5   payment(+3).     // How many pizzas does Giacomo get by a working day?
6   energy(10).      // The Giacomo life energy.
7   lunch(+2).       // How much energy does Giacomo get eating a pizza?
8   workingDay(-1). // How much energy does Giacomo lose in a working day?
9   partyNight(-3). // How much energy does Giacomo lose on a party night?
10
11  /* Initial goals */
12  !goToParty.
13
14  /* Plans */
15  +!goToParty: lifeParameters(S,E,P,W,N,L) & (E+(N+W)>1) <- -+energy(E+N); !goToParty.
16  +!goToParty: lifeParameters(S,E,P,W,N,L) & (E+(N+W)<=1)<- !eat.
17
18  +!eat: lifeParameters(S,E,P,W,N,L) & (E<2*(-1*(N+W))) & S>0 <-  -+stock(S-1); -+energy(E+L); !eat.
19  +!eat: lifeParameters(S,E,P,W,N,L) & (E>=2*(-1*(N+W))) <-   !goToParty.
20  +!eat: lifeParameters(S,E,P,W,N,L) & S=0 <- !work.
21
22  +!work: lifeParameters(S,E,P,W,N,L) <- -+stock(S+P); -+energy(E+W); !eat.
23
24  +energy(E): E < 1 <- .print("Giacomo died of hunger!"); .stopMAS.
25  +stock(S): C < 0 <- .print("Without food!"); .stopMAS.
```

https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/06-MyFirstAgent/giacomoAgent.chon

EMBEDDED MAS



Jason Embedded

Jason Embedded is a spin-off Version of Jason for IoT and Embedded MAS:

Jason Embedded

Jason Embedded is a spin-off Version of Jason for IoT and Embedded MAS:

- **Autonomy** as a characteristic an Embedded MAS provides to a device that does not need external architectures to control and manage it.

Jason Embedded

Jason Embedded is a spin-off Version of Jason for IoT and Embedded MAS:

- **Autonomy** as a characteristic an Embedded MAS provides to a device that does not need external architectures to control and manage it.
- **Communicability** is the agent's ability to communicate with other agents in its MAS or another MAS.

Jason Embedded

Jason Embedded is a spin-off Version of Jason for IoT and Embedded MAS:

- **Autonomy** as a characteristic an Embedded MAS provides to a device that does not need external architectures to control and manage it.
- **Communicability** is the agent's ability to communicate with other agents in its MAS or another MAS.
- **Mobility** as the agents' ability to move from one MAS to another.

Jason Embedded

Jason Embedded is a spin-off Version of Jason for IoT and Embedded MAS:

- **Autonomy** as a characteristic an Embedded MAS provides to a device that does not need external architectures to control and manage it.
- **Communicability** is the agent's ability to communicate with other agents in its MAS or another MAS.
- **Mobility** as the agents' ability to move from one MAS to another.
- **Fault Tolerance and Adaptability** are defined as the ability of an agent to identify if a physical resource is absent or not answering commands and to overcome this situation by modifying its goals.

Jason Embedded

- **Jason Embedded** also provides **new types of agents** to control hardware devices and to communicate with agents hosted in other MAS.

Jason Embedded

- **Jason Embedded** also provides **new types of agents** to control hardware devices and to communicate with agents hosted in other MAS.
  - **ARGO**: is a customized agent architecture built on the Jason framework that extends Jason's standard agents by adding the **ability to control microcontrollers** [Pantoja et al., 2016].

Jason Embedded

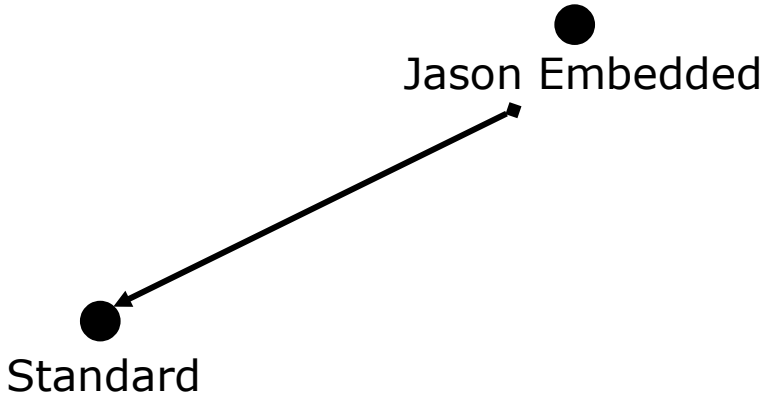
- **Jason Embedded** also provides **new types of agents** to control hardware devices and to communicate with agents hosted in other MAS.
  - **ARGO**: is a customized agent architecture built on the Jason framework that extends Jason's standard agents by adding the **ability to control microcontrollers** [Pantoja et al., 2016].
  - **Hermes**: is another customized agent architecture built on Jason's Standard agent by adding the **ability to communicate** with agents from **other MAS**, which also have Communicator agents [Pantoja et al., 2018].

Jason Embedded

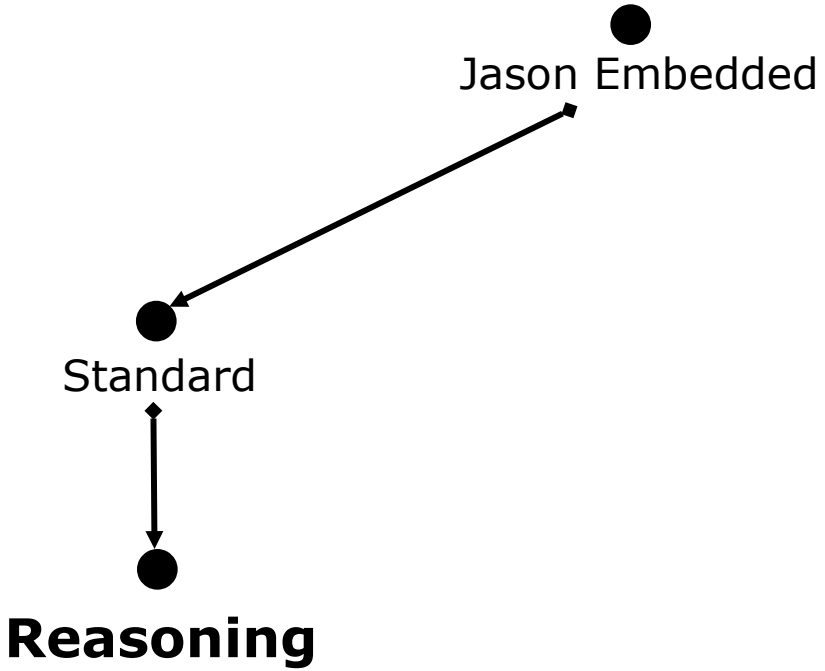
- **Mobility** is based on **bio-inspired protocols** [Jesus et al., 2021].
  - The protocols simulate natural behaviors that could be explored in collaborative tasks using IoT devices.
  - One or more agents or even the whole MAS could move from one device to another to take control of the destination device (**Predation**) or to use it as a temporary non-hazardous relationship (**Mutualism** and **Inquilinism**) if both parties accept the protocol.

Jason Embedded

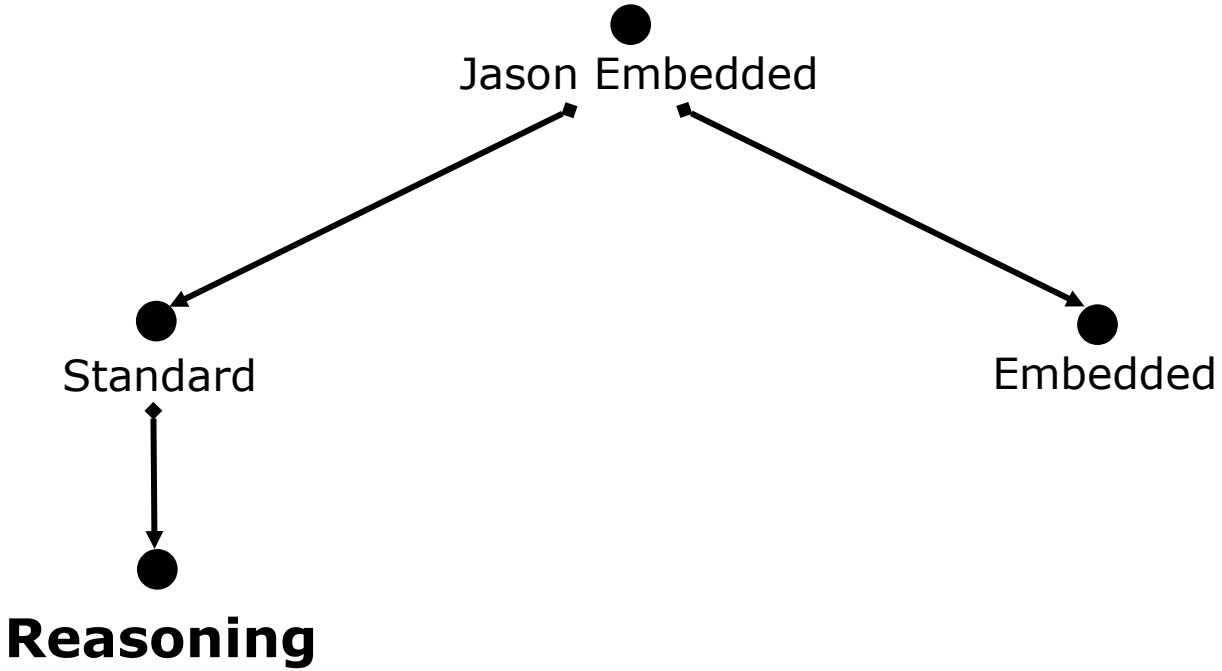
Jason Embedded



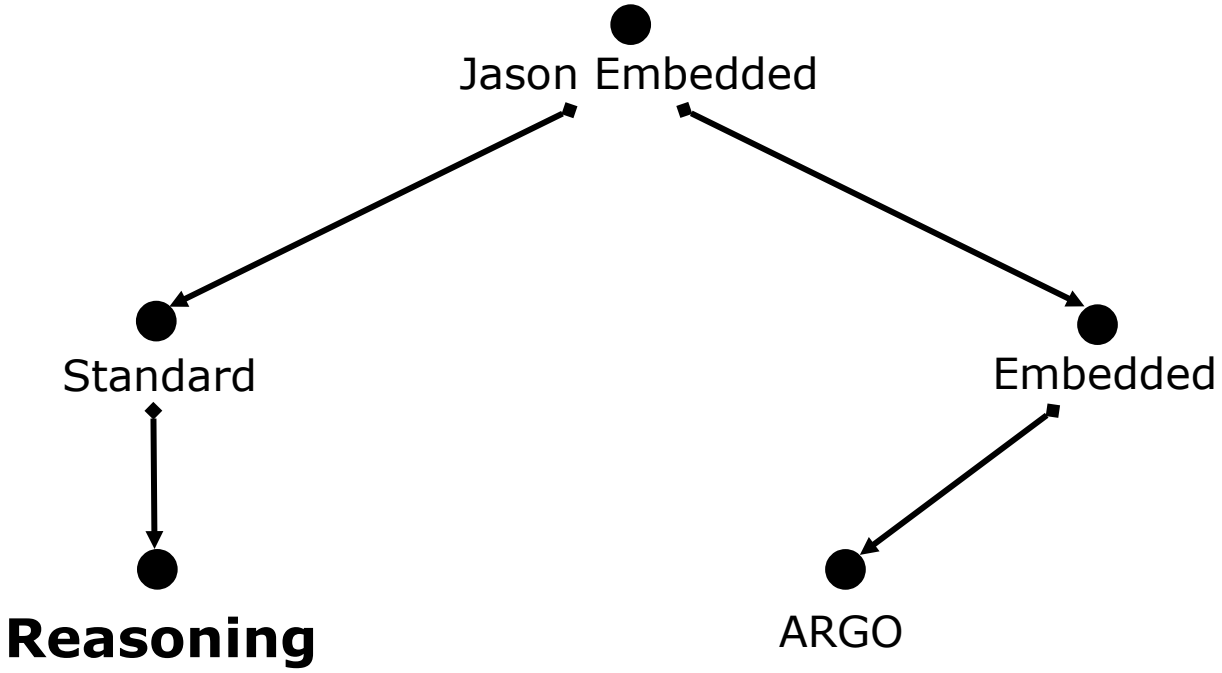
Jason Embedded



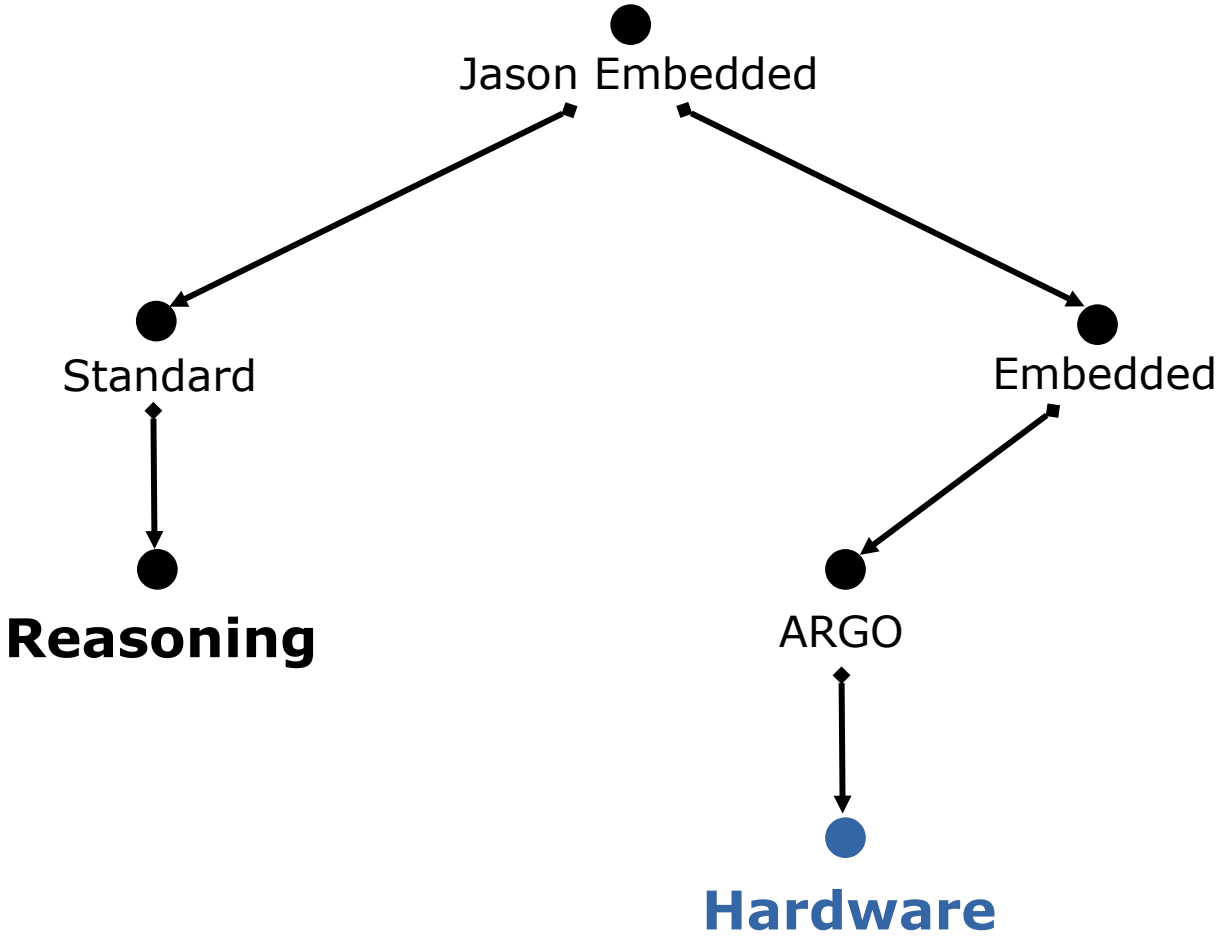
Jason Embedded



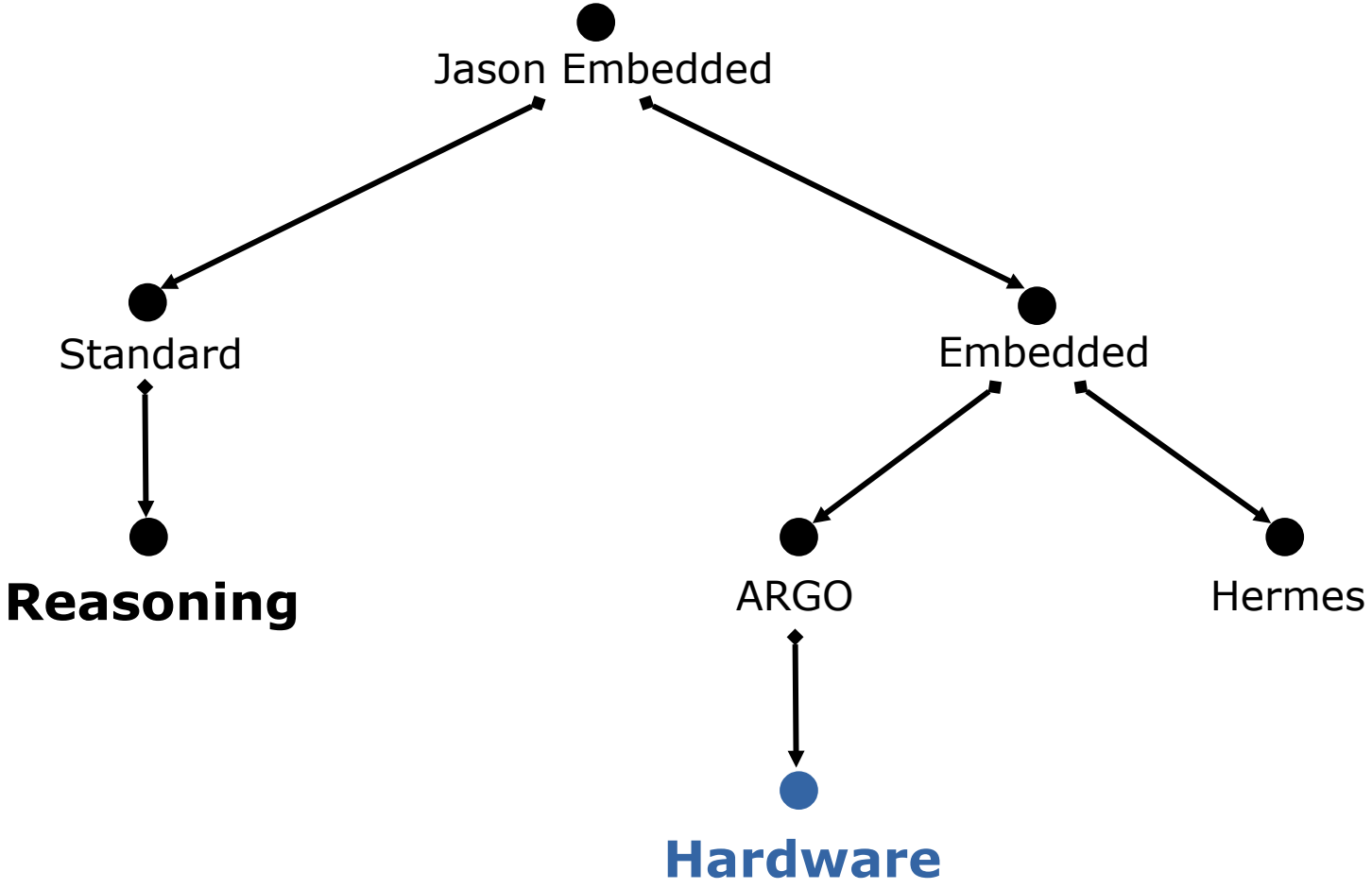
Jason Embedded



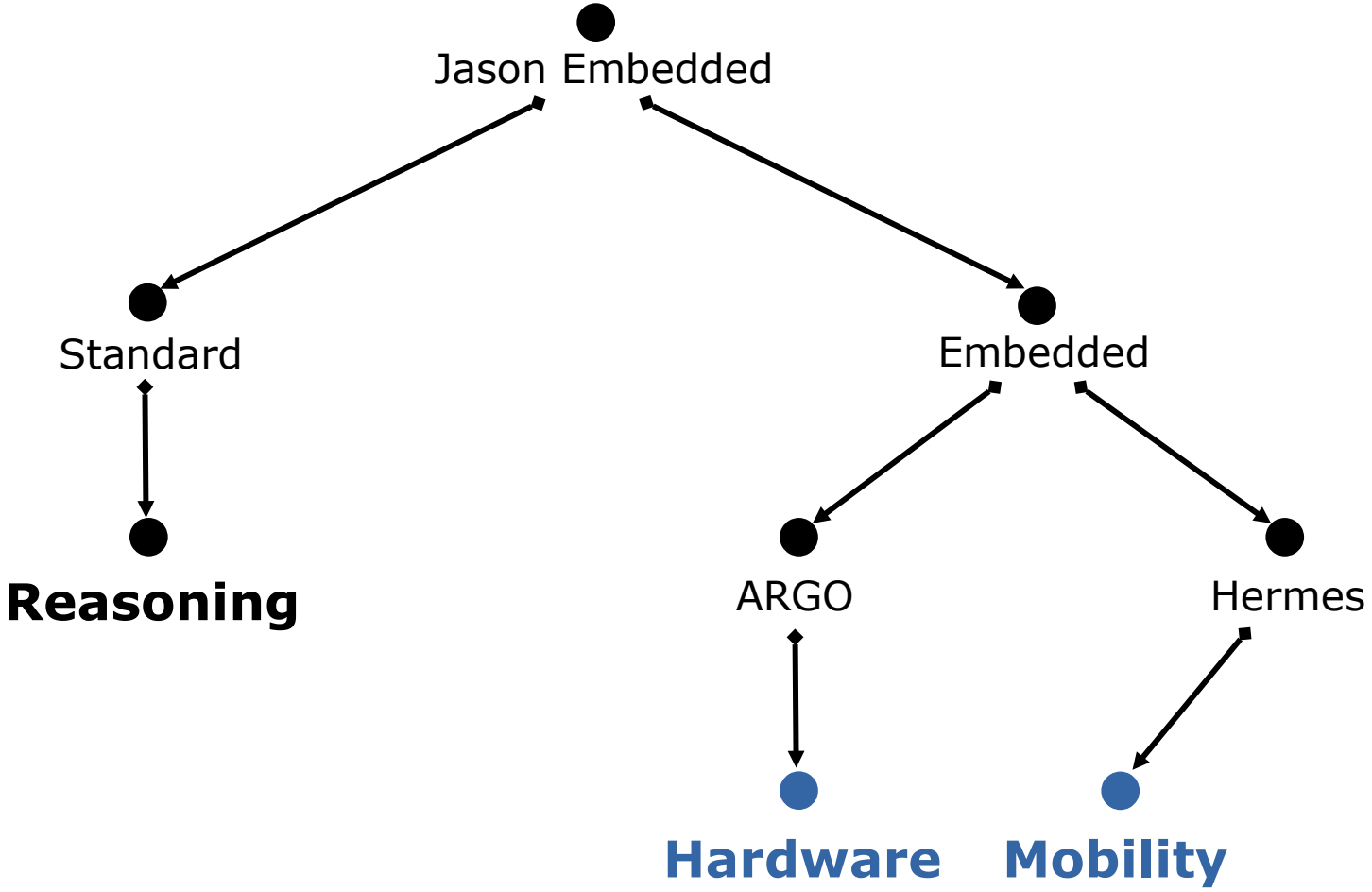
Jason Embedded



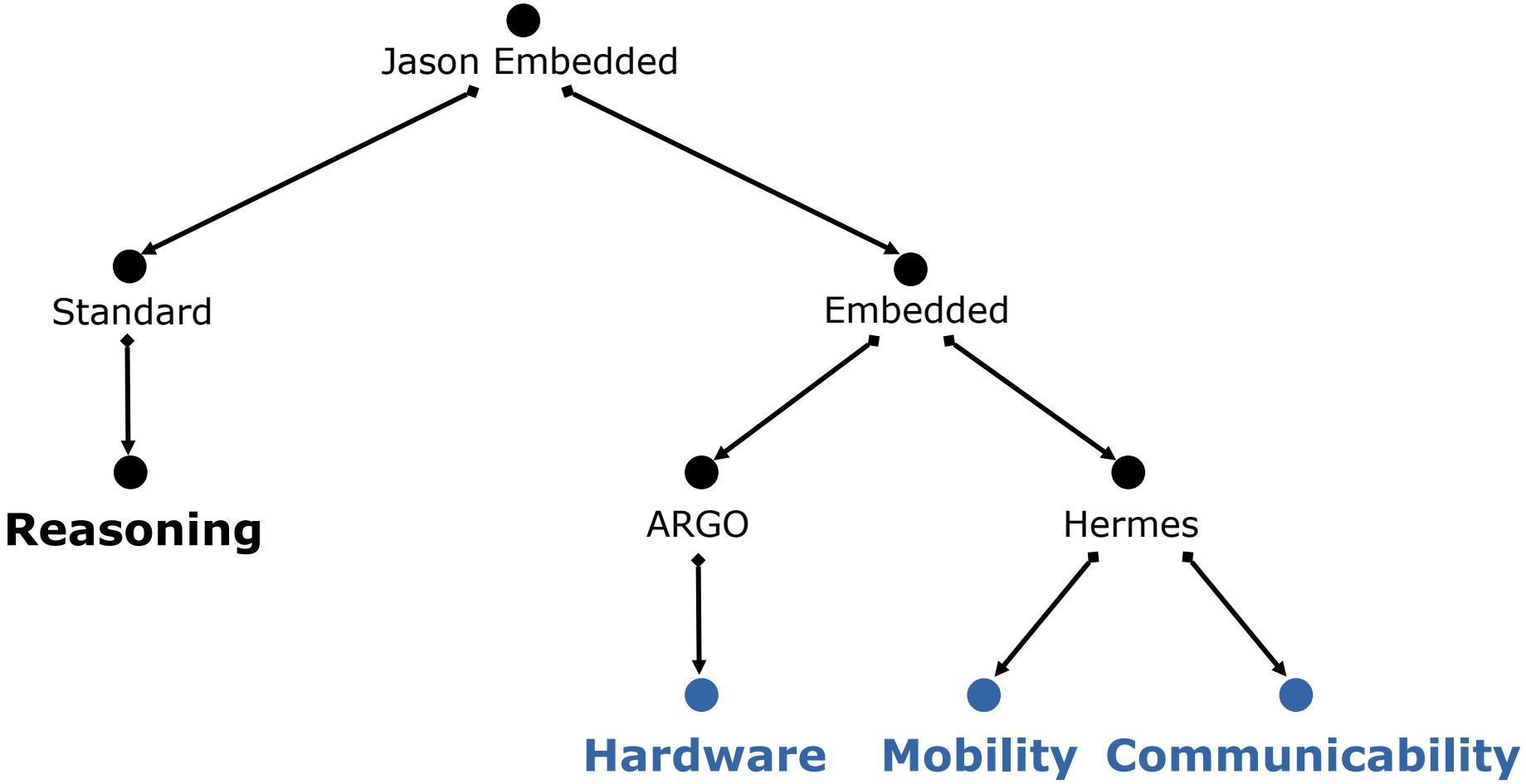
Jason Embedded



Jason Embedded

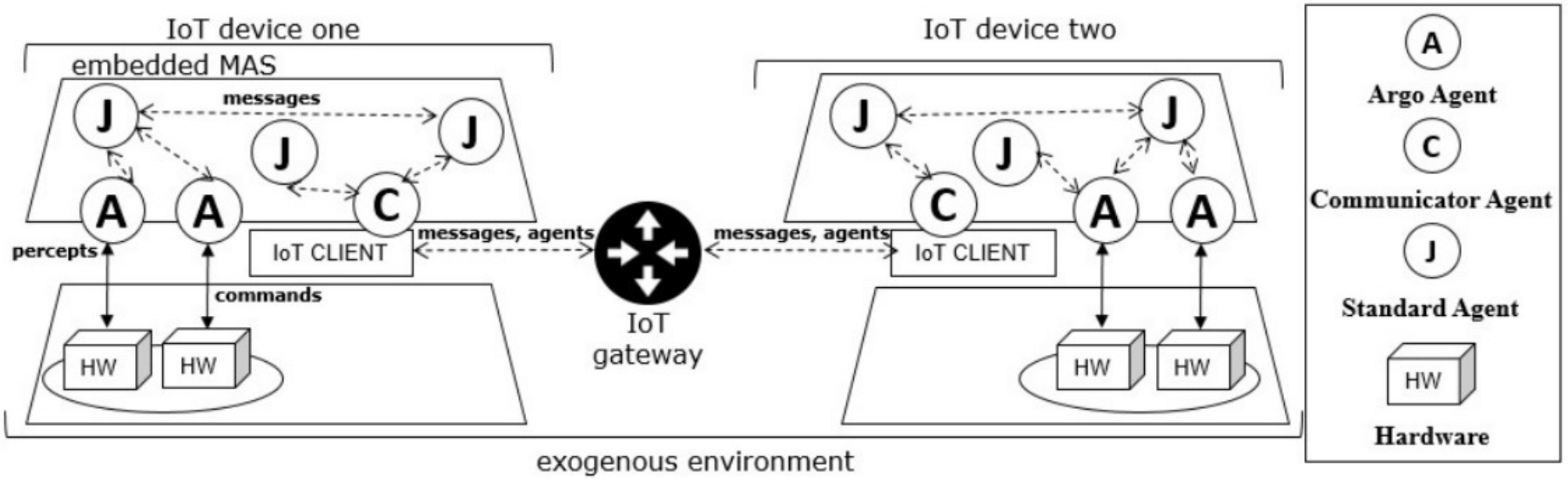


Jason Embedded

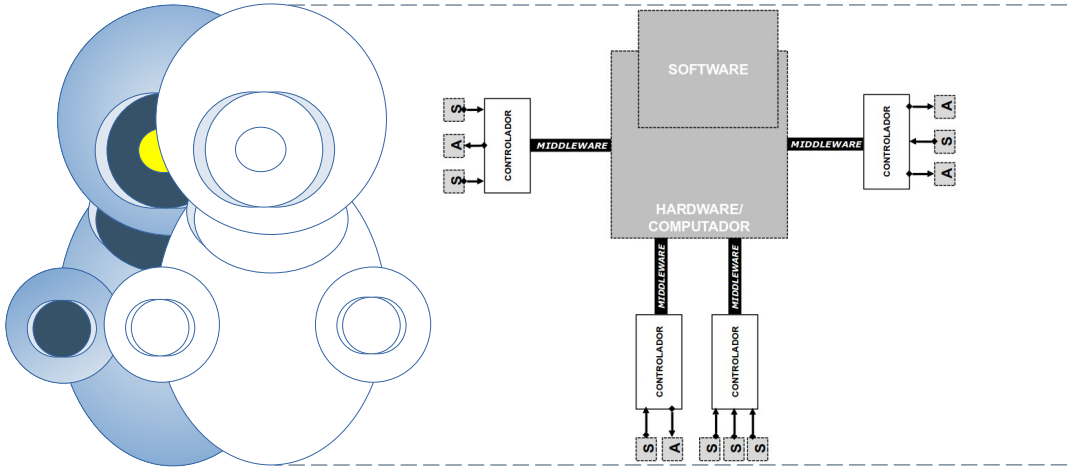


Jason Embedded

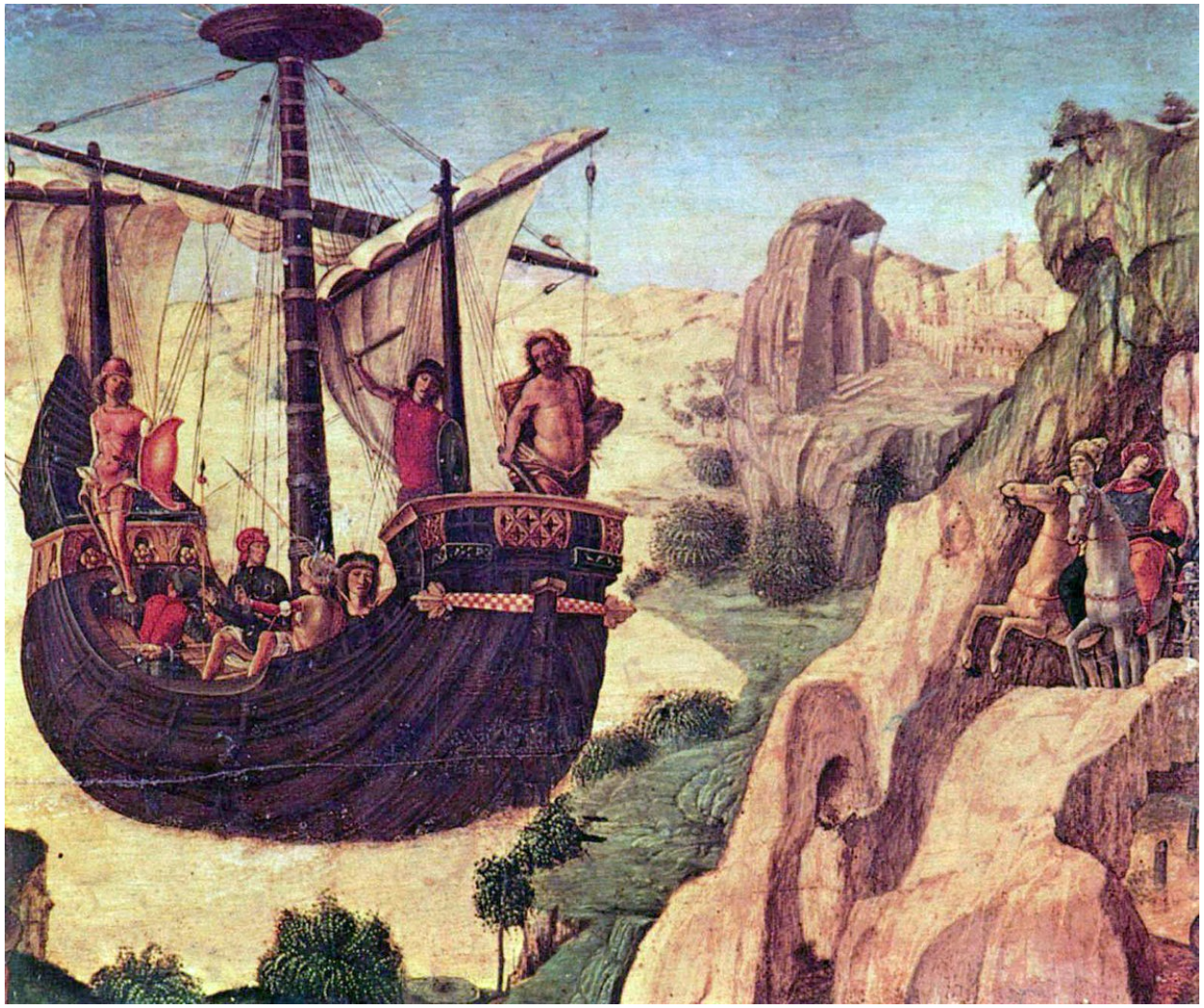
It is a spin-off Version of Jason for IoT and Embedded Multiagent Systems.



ARGO AGENT


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

Argo for Jason



Argo was the ship **Jason**, and the Argonauts sailed in search of the **golden fleece** in Greek mythology.

The Argo
by Lorenzo Costa

Argo for Jason

ARGO is a customized agent architecture that **interfaces** sensors and hardware actuators.

Argo for Jason

ARGO is a customized agent architecture that **interfaces** sensors and hardware actuators.

ARGO aims to be a practical architecture for **programming embedded BDI agents** using **Jason** and microcontroller boards.

Argo for Jason

The **ARGO** allows to:

Argo for Jason

The **ARGO** allows to:

1. to direct control actuators at runtime;

Argo for Jason

The **ARGO** allows to:

1. to direct control actuators at runtime;
2. to receive sensor perceptions within a predefined period automatically;

Argo for Jason

The **ARGO** allows to:

1. to direct control actuators at runtime;
2. to receive sensor perceptions within a predefined period automatically;
3. to change which devices are being accessed at runtime;

Argo for Jason

The **ARGO** allows to:

1. to direct control actuators at runtime;
2. to receive sensor perceptions within a predefined period automatically;
3. to change which devices are being accessed at runtime;
4. to communicate with other agents in Jason;

Argo for Jason

The **ARGO** allows to:

1. to direct control actuators at runtime;
2. to receive sensor perceptions within a predefined period automatically;
3. to change which devices are being accessed at runtime;
4. to communicate with other agents in Jason;
5. whether or not to perceive the real world at runtime;

Argo for Jason

The **ARGO** allows to:

1. to direct control actuators at runtime;
2. to receive sensor perceptions within a predefined period automatically;
3. to change which devices are being accessed at runtime;
4. to communicate with other agents in Jason;
5. whether or not to perceive the real world at runtime;
6. to change perception filters at runtime.

Argo for Jason

- **ARGO** Internal Actions:

Argo for Jason

- **ARGO** Internal Actions:
  - .argo.limit(x): defines a time interval to perceive the environment.

Argo for Jason

- **ARGO** Internal Actions:
  - .argo.limit(x): defines a time interval to perceive the environment.
  - .argo.port(y): defines which serial port the agent should operate.

Argo for Jason

- **ARGO** Internal Actions:
  - .argo.limit(x): defines a time interval to perceive the environment.
  - .argo.port(y): defines which serial port the agent should operate.
  - .argo.percepts(open|close): decides whether or not to perceive the real world.

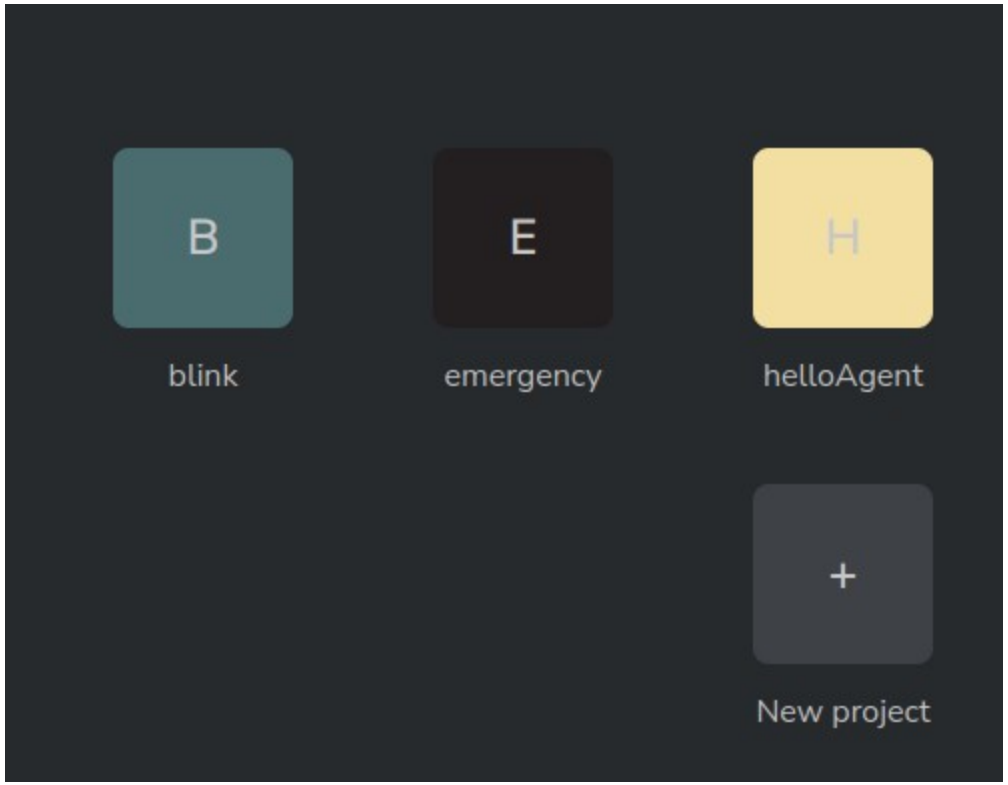
Argo for Jason

- **ARGO** Internal Actions:
  - .argo.limit(x): defines a time interval to perceive the environment.
  - .argo.port(y): defines which serial port the agent should operate.
  - .argo.percepts(open|close): decides whether or not to perceive the real world.
  - .argo.act(w): sends a command to the microcontroller to be executed by an effector.

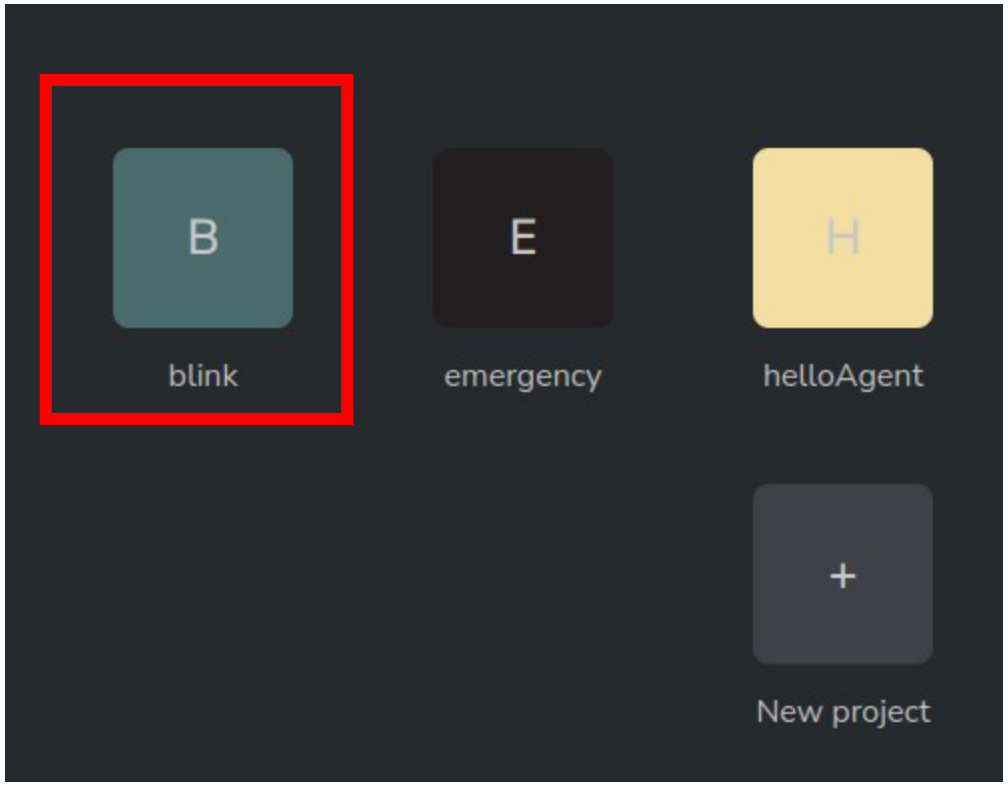
Argo for Jason

- **ARGO** Internal Actions:
  - .argo.limit(x): defines a time interval to perceive the environment.
  - .argo.port(y): defines which serial port the agent should operate.
  - .argo.percepts(open|close): decides whether or not to perceive the real world.
  - .argo.act(w): sends a command to the microcontroller to be executed by an effector.
  - .argo.change_filter(filterName): defines a perception filter to restrict real-time perceptions.

Example: blink



Example: blink

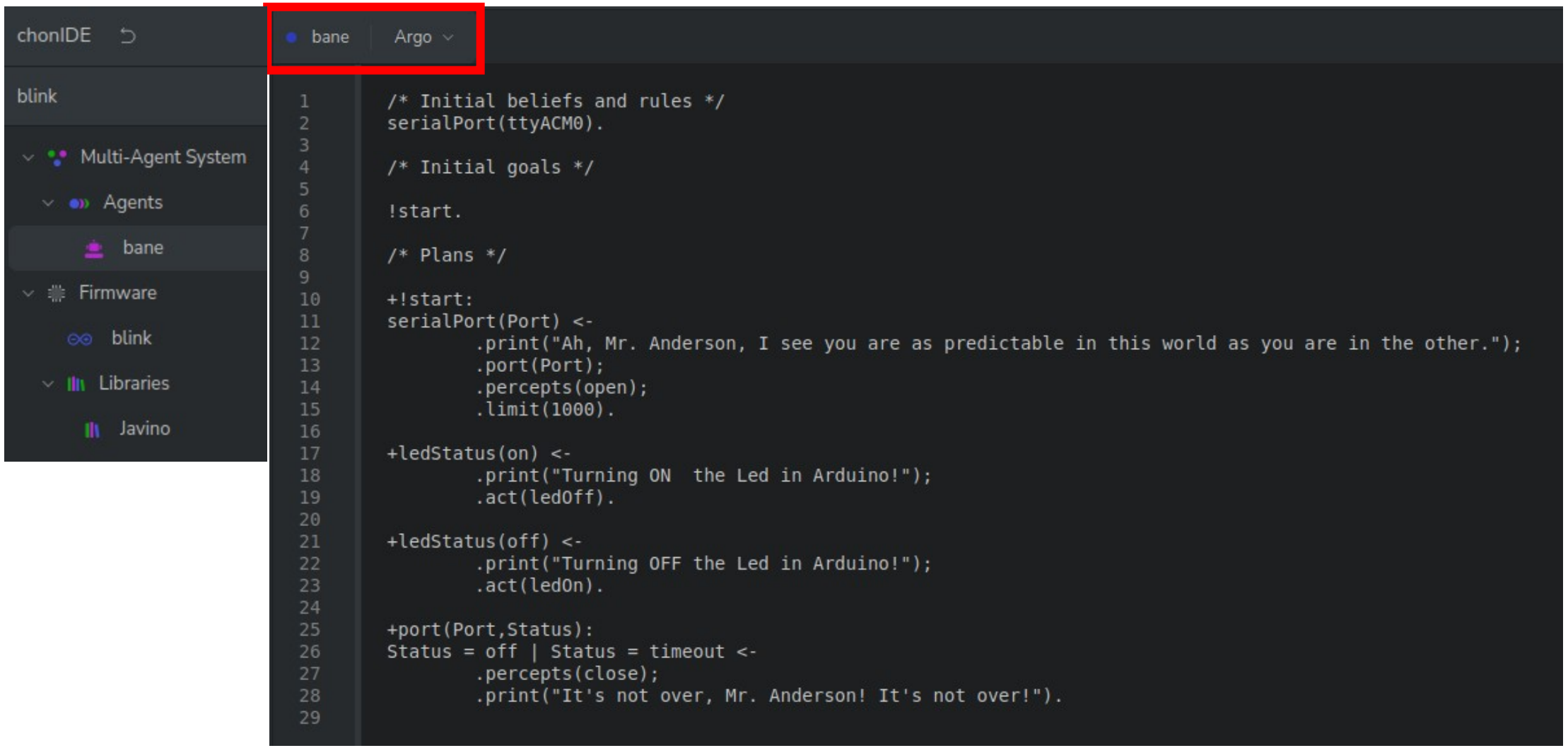


Example: blink

bane | Argo

```
1   /* Initial beliefs and rules */
2   serialPort(ttyACM0).
3
4   /* Initial goals */
5
6   !start.
7
8   /* Plans */
9
10  +!start:
11  serialPort(Port) <-
12          .print("Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.");
13          .port(Port);
14          .percepts(open);
15          .limit(1000).
16
17  +ledStatus(on) <-
18          .print("Turning ON  the Led in Arduino!");
19          .act(ledOff).
20
21  +ledStatus(off) <-
22          .print("Turning OFF the Led in Arduino!");
23          .act(ledOn).
24
25  +port(Port,Status):
26  Status = off | Status = timeout <-
27          .percepts(close);
28          .print("It's not over, Mr. Anderson! It's not over!").
29
```

Example: blink




```
/* Initial beliefs and rules */
serialPort(ttyACM0).

/* Initial goals */

!start.

/* Plans */

+!start:
serialPort(Port) <-
        .print("Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.");
        .port(Port);
        .percepts(open);
        .limit(1000).

+ledStatus(on) <-
        .print("Turning ON  the Led in Arduino!");
        .act(ledOff).

+ledStatus(off) <-
        .print("Turning OFF the Led in Arduino!");
        .act(ledOn).

+port(Port,Status):
Status = off | Status = timeout <-
        .percepts(close);
        .print("It's not over, Mr. Anderson! It's not over!").

```

Example: blink

bane | Argo

```
/* Initial beliefs and rules */
serialPort(ttyACM0).

/* Initial goals */

!start.

/* Plans */

+!start:
serialPort(Port) <-
        .print("Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.");
        .port(Port);
        .percepts(open);
        .limit(1000).

+ledStatus(on) <-
        .print("Turning ON  the Led in Arduino!");
        .act(ledOff).

+ledStatus(off) <-
        .print("Turning OFF the Led in Arduino!");
        .act(ledOn).

+port(Port,Status):
Status = off | Status = timeout <-
        .percepts(close);
        .print("It's not over, Mr. Anderson! It's not over!").

```

Example: blink

bane | Argo

```
1   /* Initial beliefs and rules */
2   serialPort(ttyACM0).
3
4   /* Initial goals */
5
6   !start.
7
8   /* Plans */
9
10  +!start:
11  serialPort(Port) <-
12          .print("Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.");
13          .port(Port);
14          .percepts(open);
15          .limit(1000).
16
17  +ledStatus(on) <-
18          .print("Turning ON  the Led in Arduino!");
19          .act(ledOff).
20
21  +ledStatus(off) <-
22          .print("Turning OFF the Led in Arduino!");
23          .act(ledOn).
24
25  +port(Port,Status):
26  Status = off | Status = timeout <-
27          .percepts(close);
28          .print("It's not over, Mr. Anderson! It's not over!").
29
```

Example: blink

bane Argo

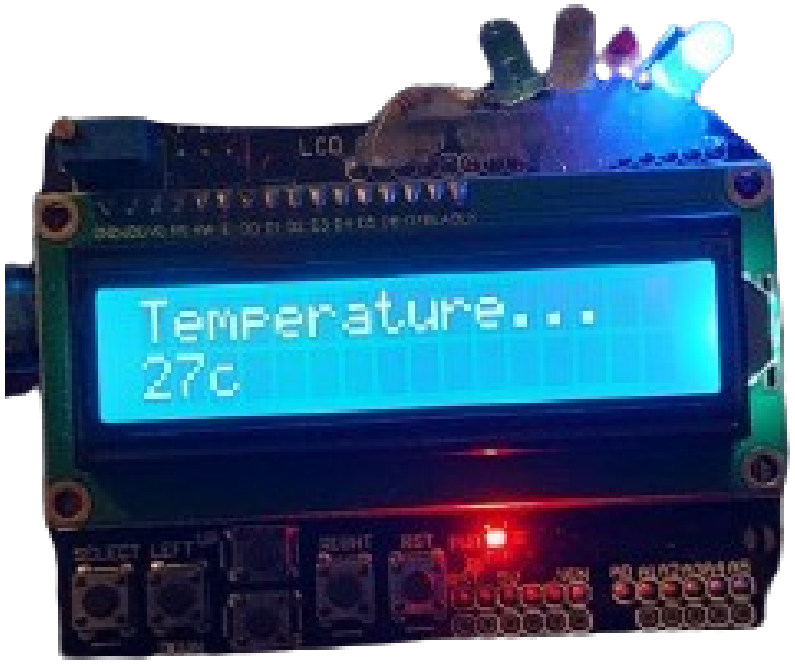
```
1   /* Initial beliefs and rules */
2   serialPort(ttyACM0).
3
4   /* Initial goals */
5
6   !start.
7
8   /* Plans */
9
10  +!start:
11  serialPort(Port) <-
12          .print("Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.");
13          .port(Port);
14          .percepts(open);
15          .limit(1000).
16
17  +ledStatus(on) <-
18          .print("Turning ON  the Led in Arduino!");
19          .act(ledOff).
20
21  +ledStatus(off) <-
22          .print("Turning OFF the Led in Arduino!");
23          .act(ledOn).
24
25  +port(Port,Status):
26  Status = off | Status = timeout <-
27          .percepts(close);
28          .print("It's not over, Mr. Anderson! It's not over!").
29
```

Example: blink

bane | Argo

```
1   /* Initial beliefs and rules */
2   serialPort(ttyACM0).
3
4   /* Initial goals */
5
6   !start.
7
8   /* Plans */
9
10  +!start:
11  serialPort(Port) <-
12          .print("Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.");
13          .port(Port);
14          .percepts(open);
15          .limit(1000).
16
17  +ledStatus(on) <-
18          .print("Turning ON  the Led in Arduino!");
19          .act(ledOff).
20
21  +ledStatus(off) <-
22          .print("Turning OFF the Led in Arduino!");
23          .act(ledOn).
24
25  +port(Port,Status):
26  Status = off | Status = timeout <-
27          .percepts(close);
28          .print("It's not over, Mr. Anderson! It's not over!").
29
```

Example: LCD



Example: LCD

agentARGO | Argo

```
/* beliefs and rules */
serialPort(ttyUSB0).

/* Initial goals */
!connect.

/* Achievement Plans */
+!connect:
serialPort(Port) <-
        .print("Trying to connect at ",Port);
        .argo.port(Port);
        .argo.limit(500);
        .argo.percepts(open).

+!infoLCD(M)  <- .argo.act(M).
+!yellowAlert <- .argo.act(yellowOn).
+!greenAlert  <- .argo.act(greenOn).
+!redAlert    <- .argo.act(redOn).
+!blueAlert   <- .argo.act(blueOn).

/* Belief Plans */
+device(D):
D = arduinoWithLCDKeypadShield <-
        .my_name(N);
        .concat(N," is online!",LCDMessage);
        !infoLCD(LCDMessage);
        .wait(1000);
        !redAlert.

+port(Port,Status):
Status = off | Status = timeout <-
        .argo.percepts(close);
        .print("Serial port ",Port," is ",Status);
        .print("Stopping the MAS");
        .stopMAS.

+rainLast24hrs(RainStatus) <- .print("New perception-> rainLevel: ",RainStatus," mm").

+humidity(H) <- .print("New perception-> humidity: ",H," %").

+temperature(T) <- .print("New perception-> temperature: ",T," Celsius").
```

https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/workshops/wesaac/2024/examples/lcdKeypadProject.chon

Example: LCD

```
14
15      +!infoLCD(M)   <- .argo.act(M).
16      +!yellowAlert <- .argo.act(yellowOn).
17      +!greenAlert   <- .argo.act(greenOn).
18      +!redAlert     <- .argo.act(redOn).
19      +!blueAlert    <- .argo.act(blueOn).
```

Example: LCD

```
20
21      /* Belief Plans */
22      +device(D):
23      D = arduinoWithLCDKeypadShield <-
24              .my_name(N);
25              .concat(N," is online!",LCDMessage);
26              !infoLCD(LCDMessage);
27              .wait(1000);
28              !redAlert.
29
```

Example: LCD

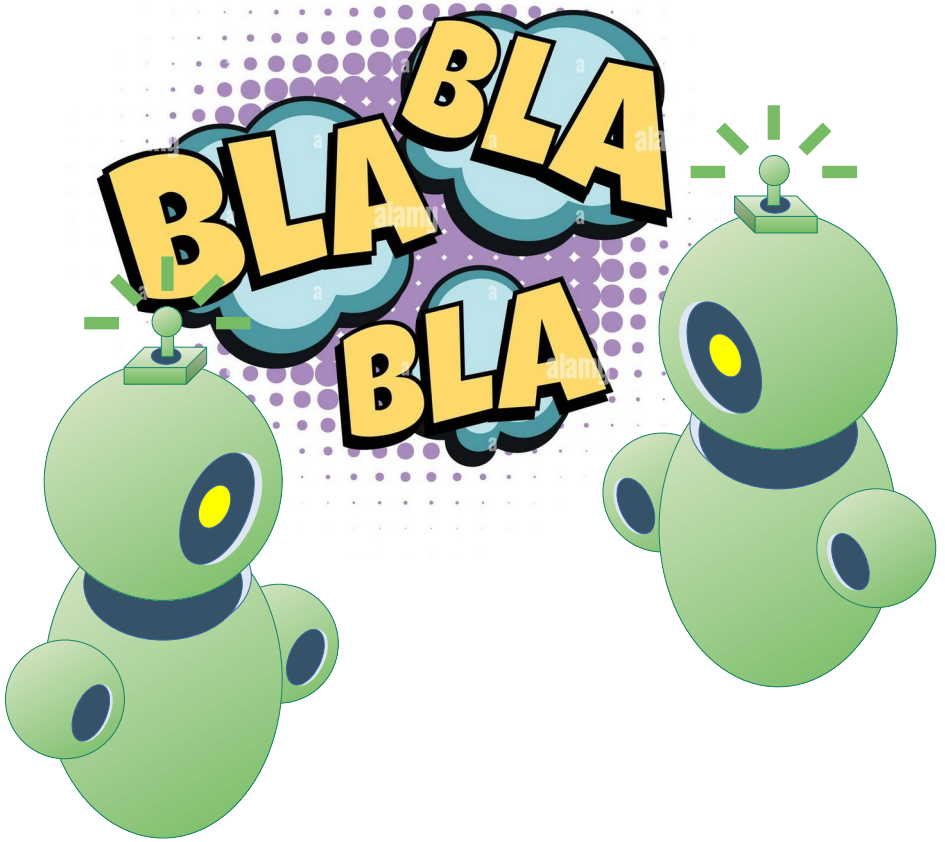
```
37      +rainLast24hrs(RainStatus) <- .print("New perception-> rainLevel: ",RainStatus," mm").
38
39      +humidity(H) <- .print("New perception-> humidity: ",H," %").
40
41      +temperature(T) <- .print("New perception-> temperature: ",T," Celsius").
```

Simple Exercise

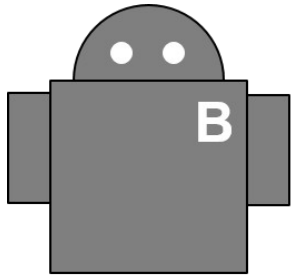
Adjust the ARGO agent to:

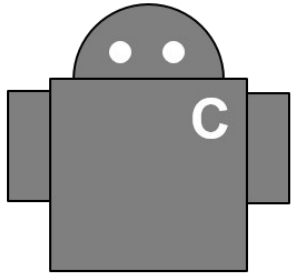
- turn on the **red LED** if the temperature is higher than 25;
- turn on the **green LED** if the humidity is higher than 80%;
- turn on the **yellow LED** if the rain intensity exceeds 30mm.

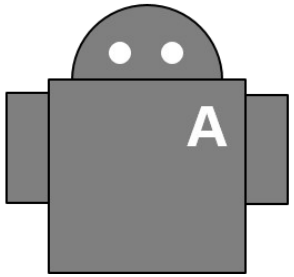
HERMES AGENT (COMMUNICABILITY)

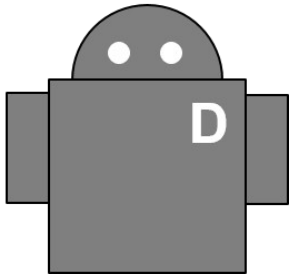


Collaborative MultiAgent System

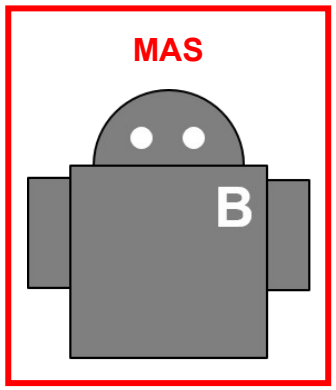


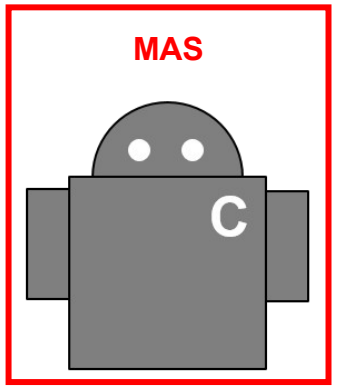


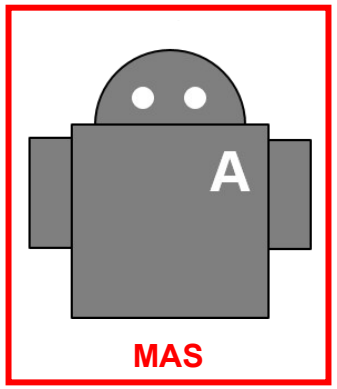


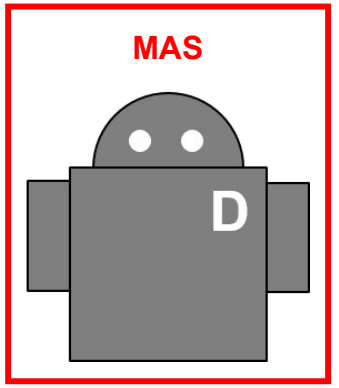


Collaborative MultiAgent System

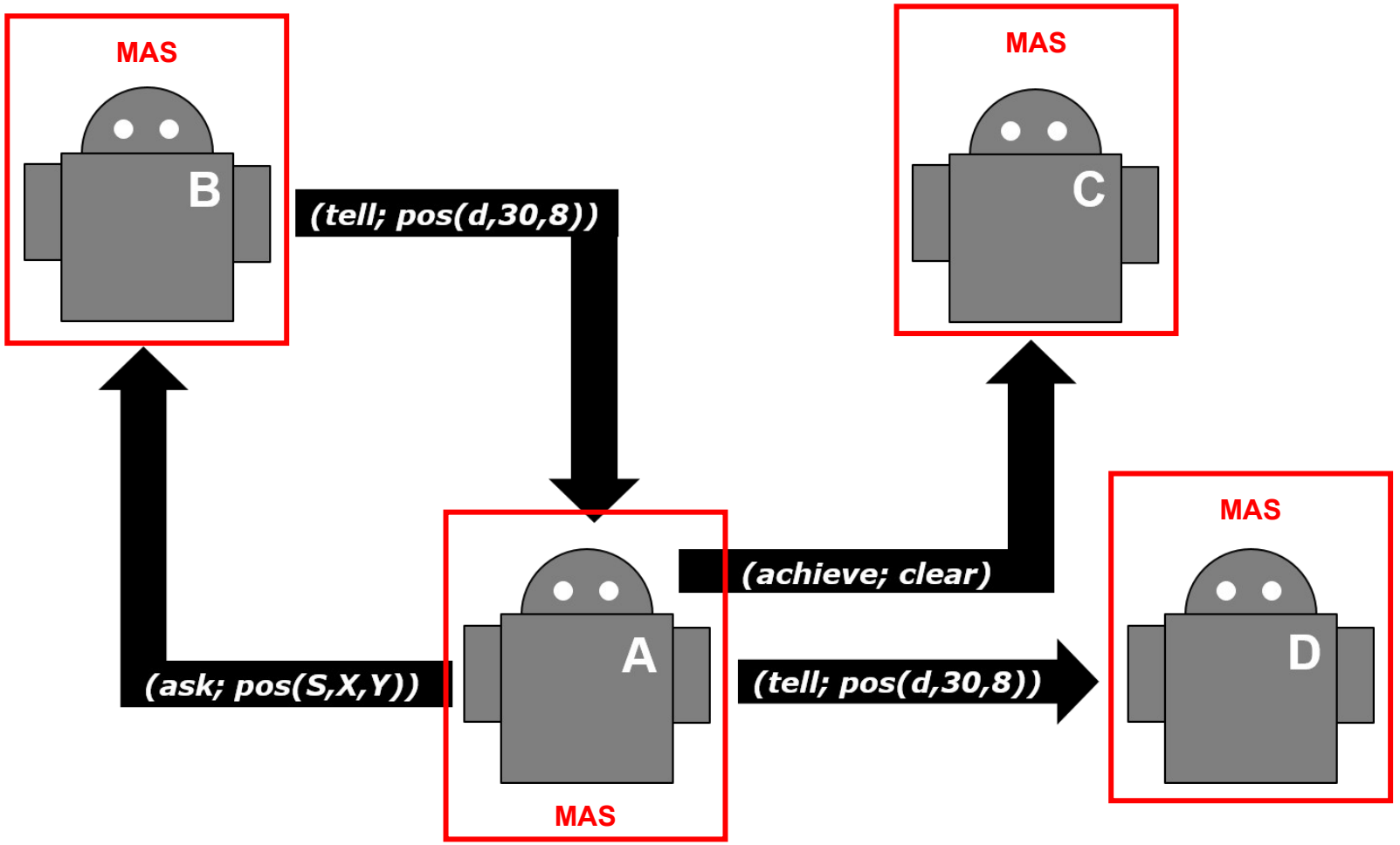




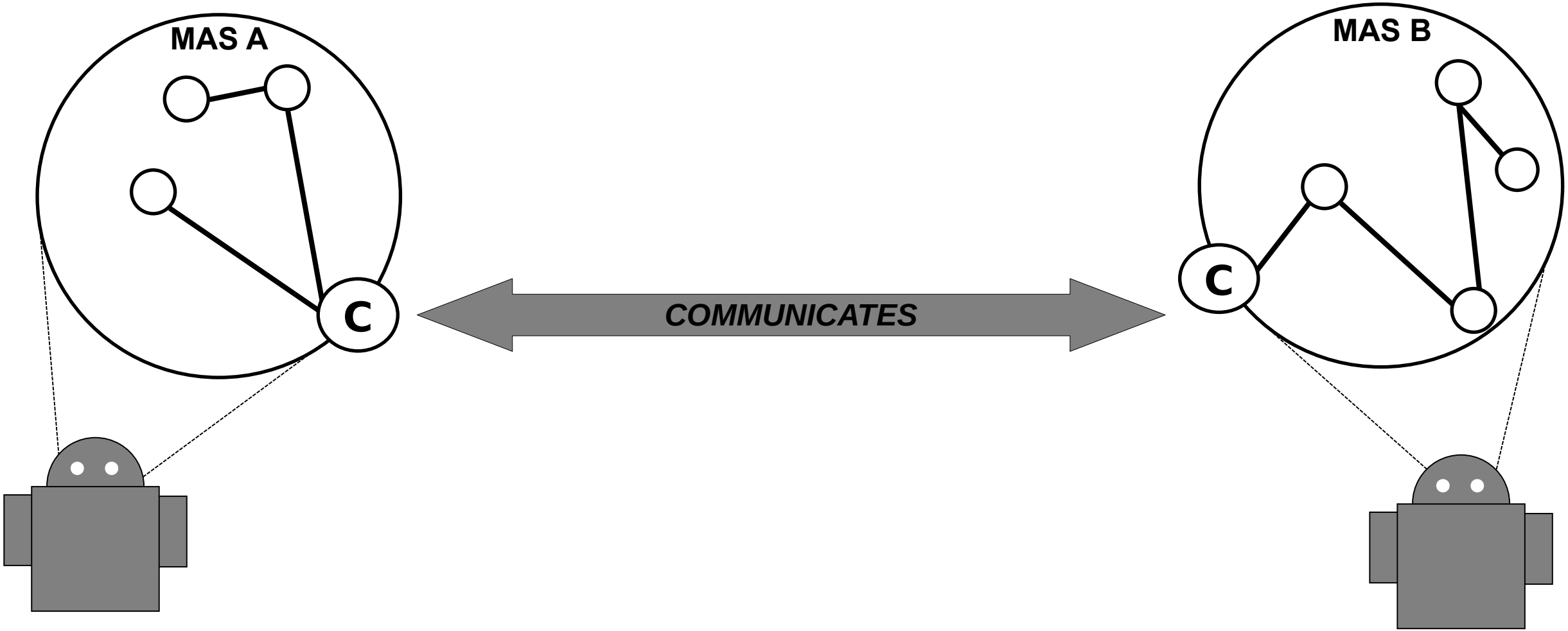




Collaborative MultiAgent System

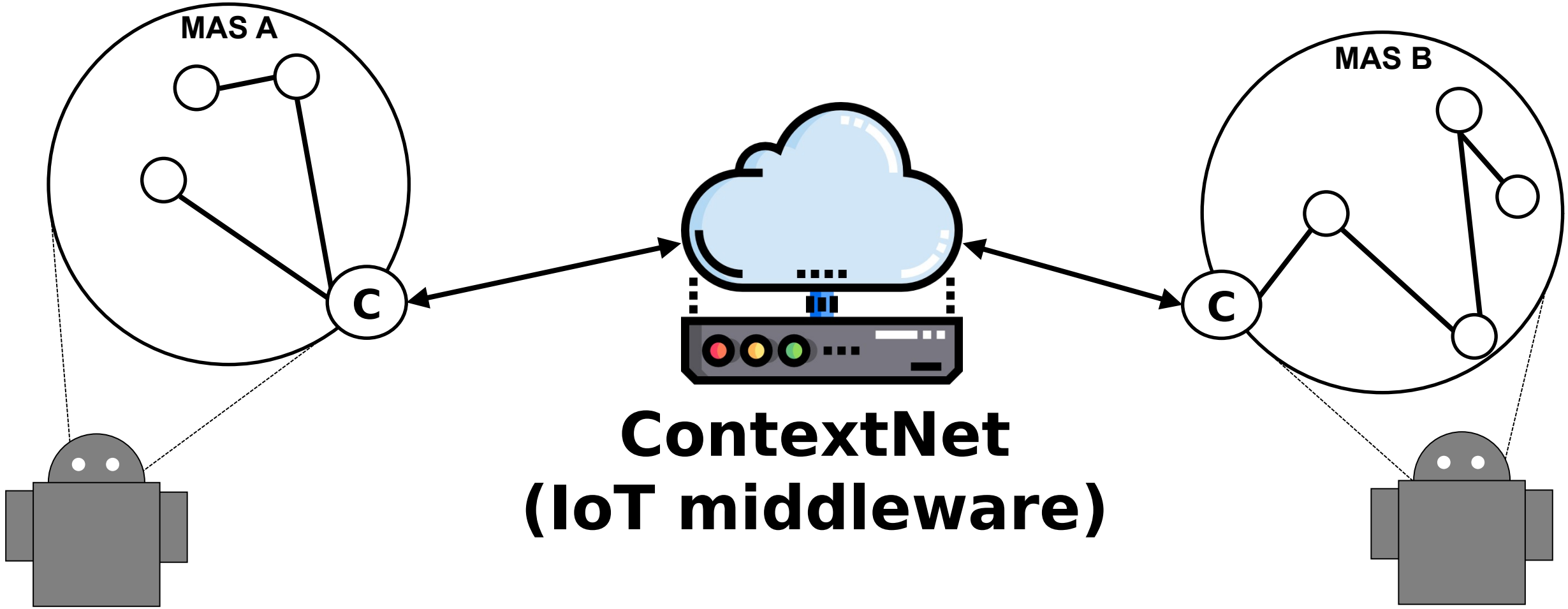


Hermes Agents



Pantoja, C., Soares, H.D., Viterbo, J., Seghrouchni, A.E.F. “An Architecture for the Development of Ambient Intelligence Systems Managed by Embedded Agents”. The 30th International Conference on Software Engineering and Knowledge Engineering, Hotel Pullman, Redwood City, California, USA, July 1-3, 2018, organizado por Óscar Mortágua Pereira, KSI Research Inc. and Knowledge Systems Institute Graduate School, 2018, https://doi.org/10.18293/SEKE2018-110 .

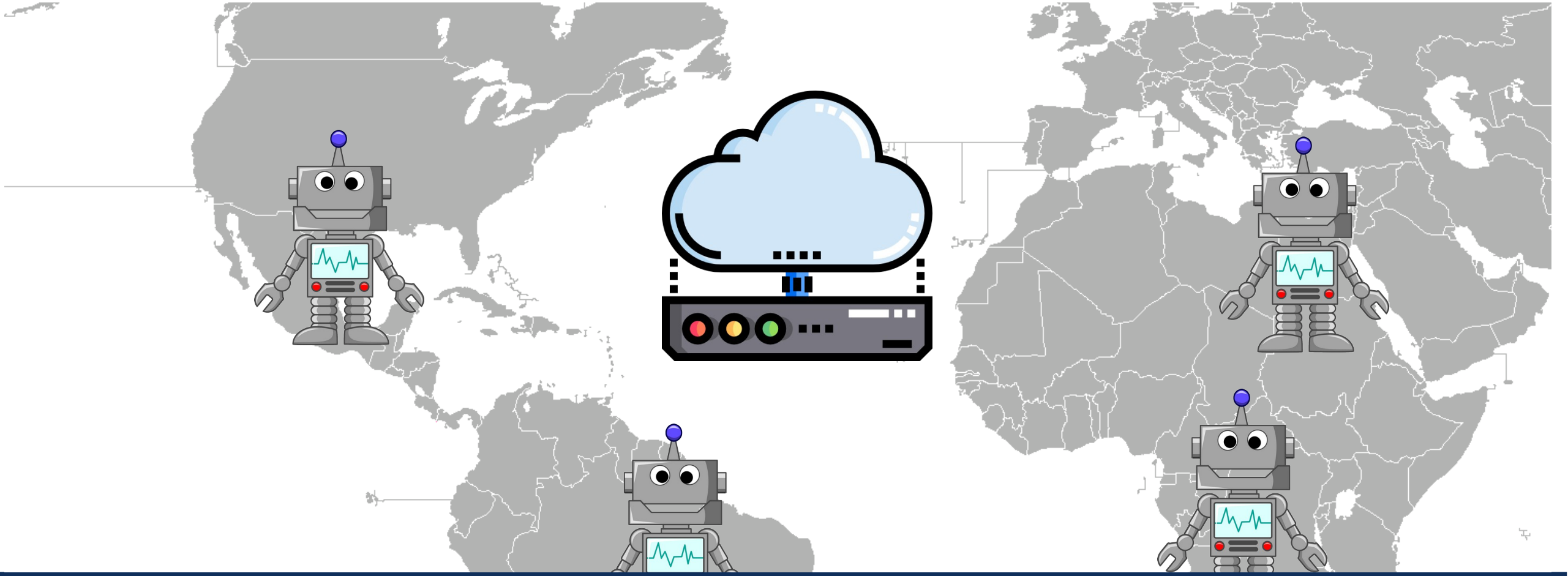
Hermes Agents: Server Communication



Endler, M., Baptista, G., Silva, L.D., Vasconcelos, R., Malcher, M., Pantoja, V., Pinheiro, V., Viterbo, J.: Contextnet: Context reasoning and sharing middleware for large-scale pervasive collaboration and social networking. In: Proceedings of the Workshop on Posters and Demos Track. PDT '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2088960.2088962

ContextNet Public Server

Address: skynet.chon.group - UDP Port: 5500



Hermes Agents: Internal Actions

- **Hermes** Internal Actions:

.sendOut(agentUuid, force, message)

sends a message from one Hermes agent to another Hermes agent in a different MAS.

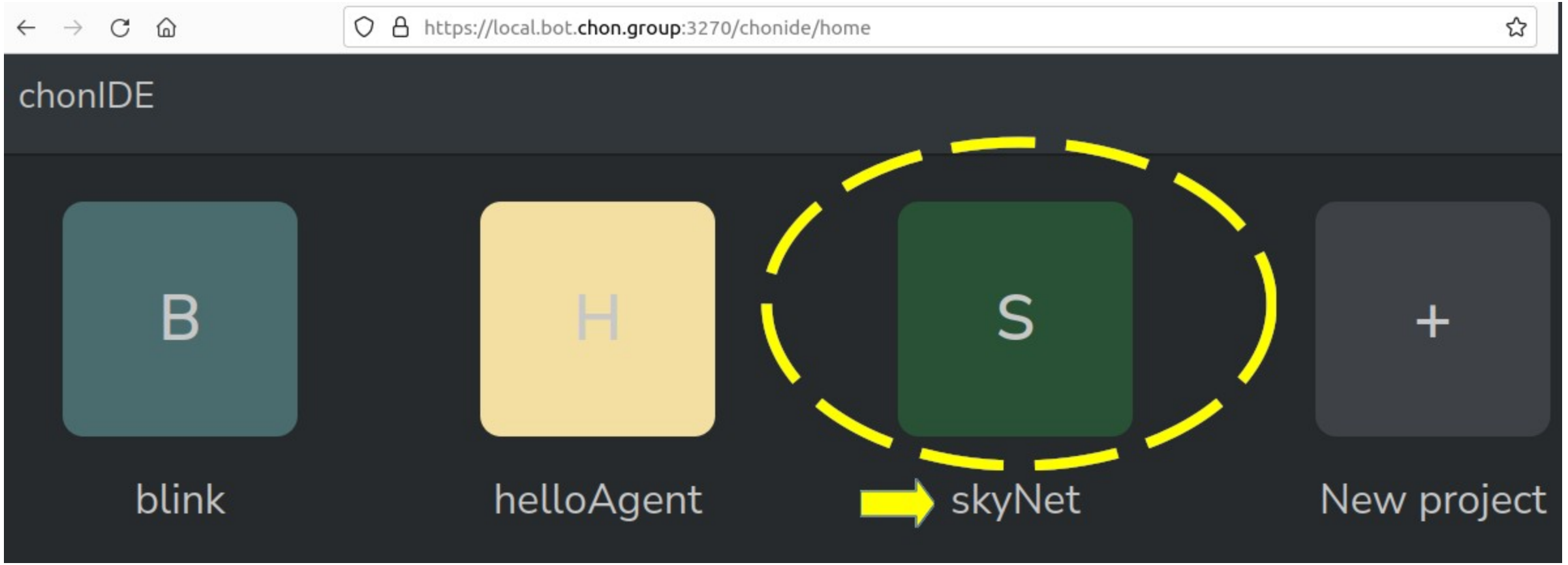
.connectCN(servidor, porta, UUID)

connects to the IoT server with a specific ID for the agent.

.disconnectCN

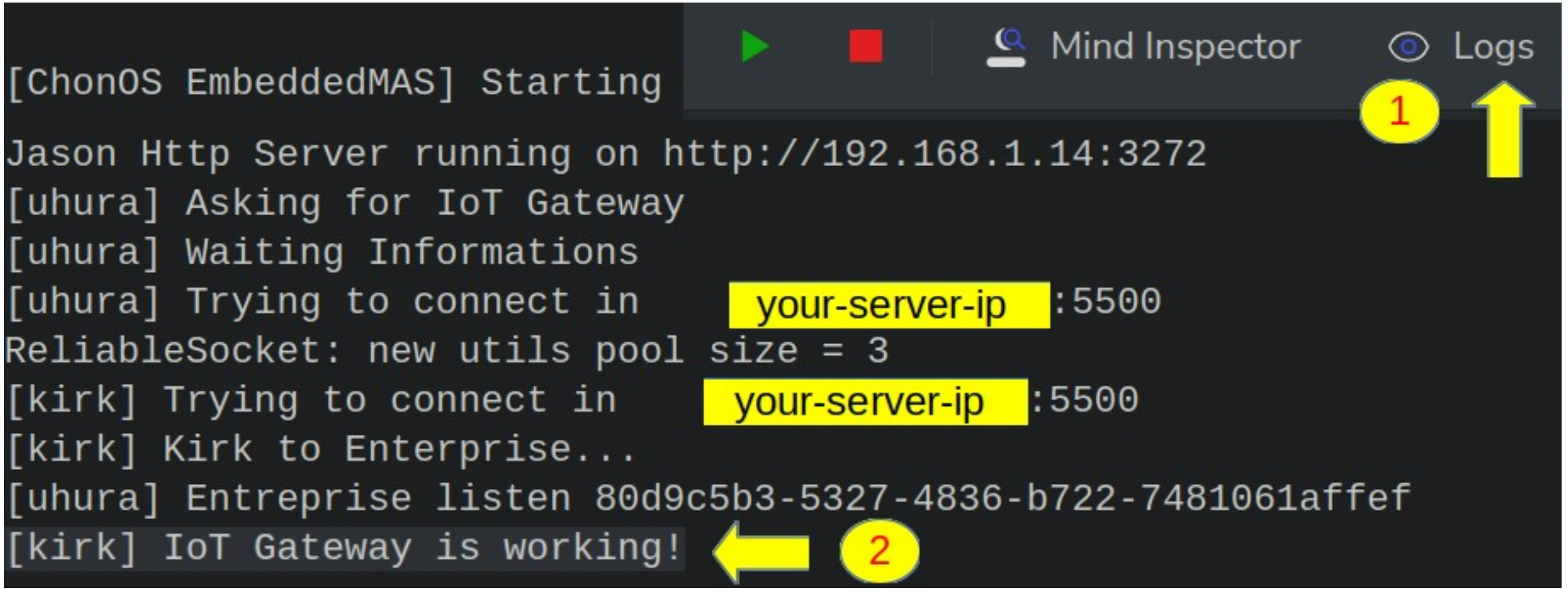
disconnects from the currently connected IoT server.

Hello SkyNet

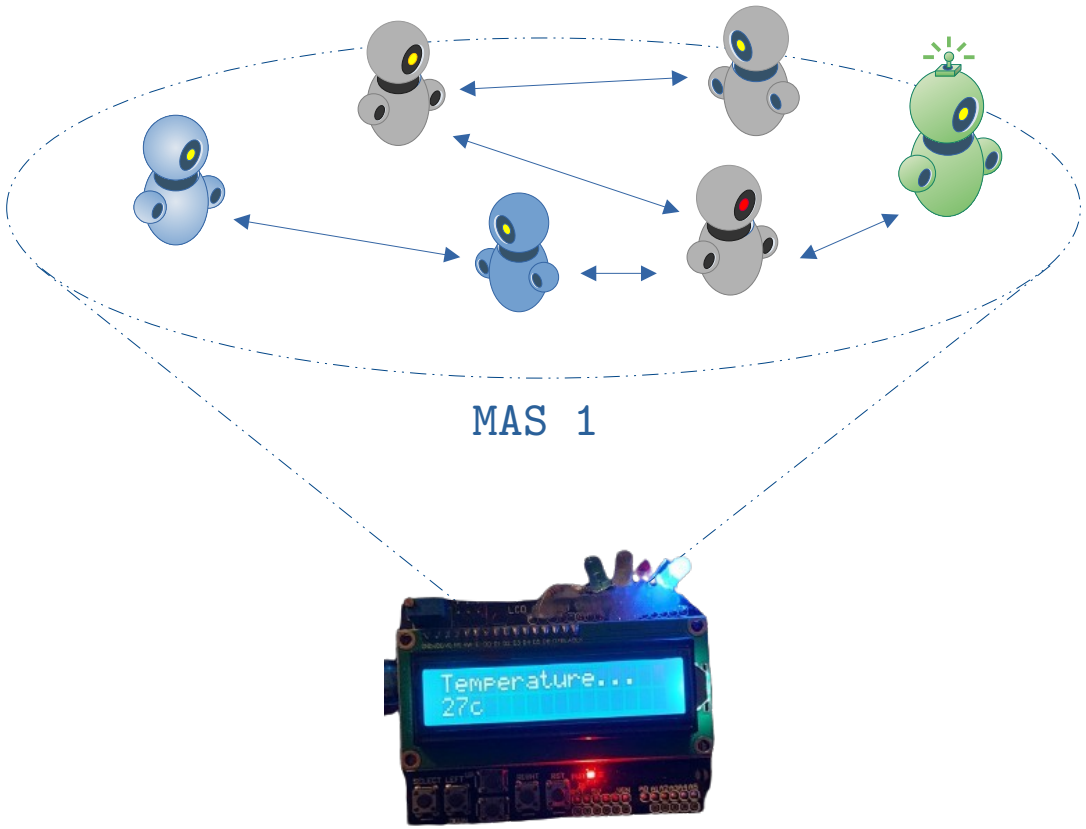


Hello SkyNet

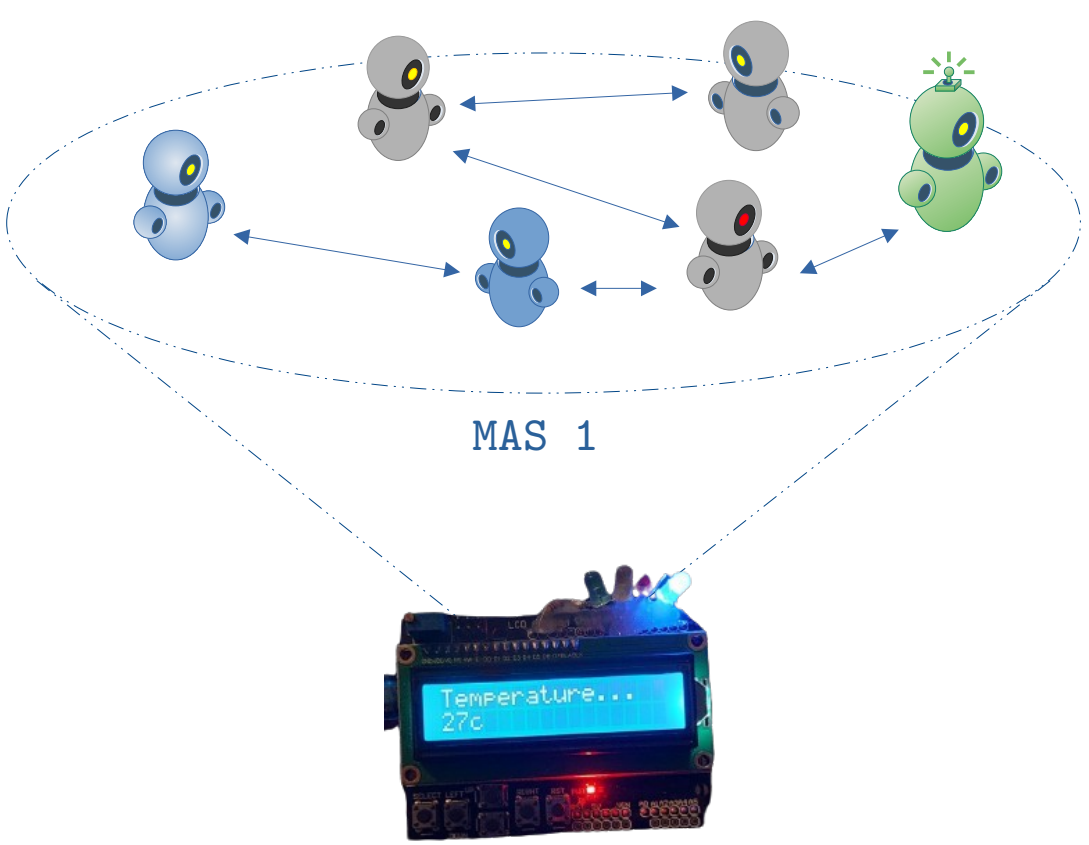
Hello SkyNet

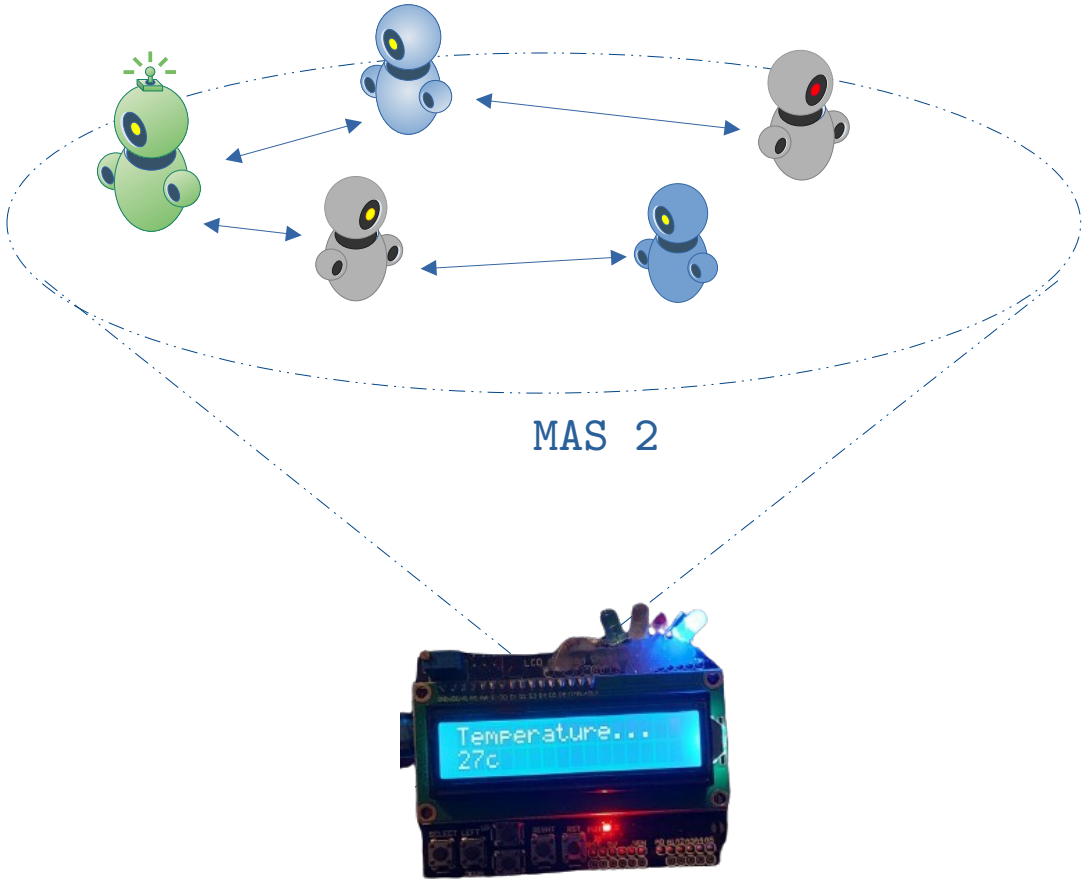


Example: MAS Communication

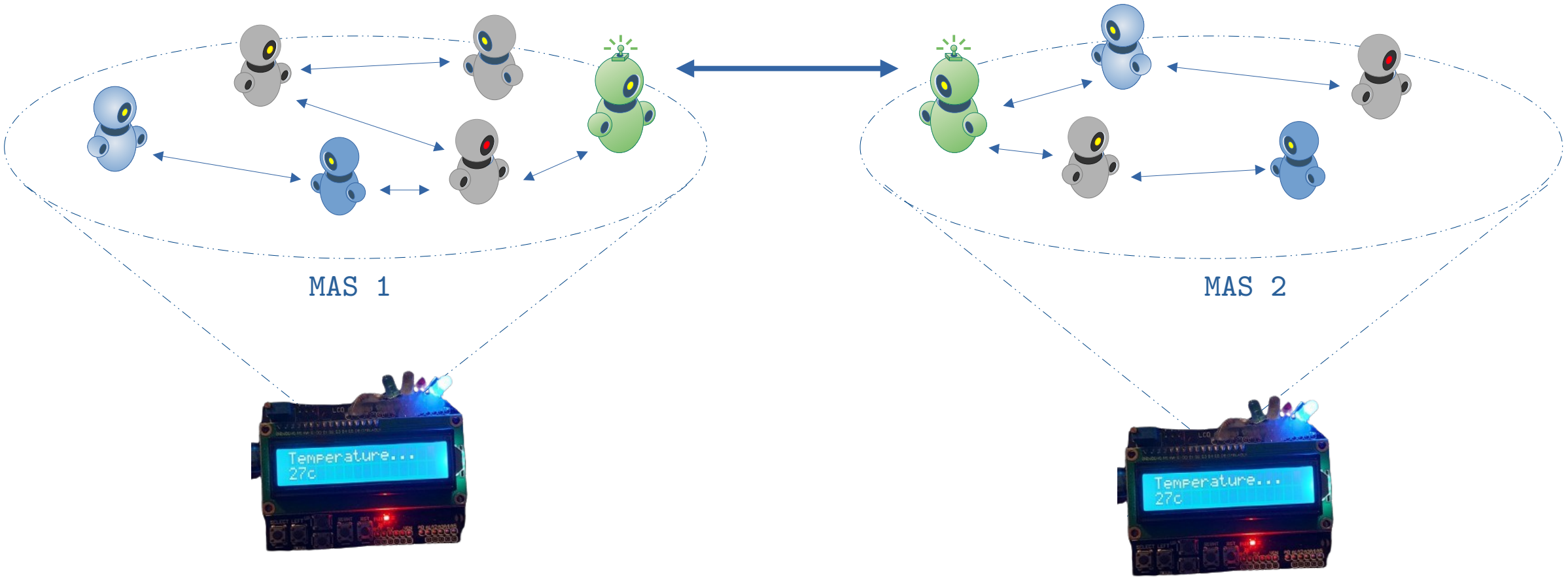


Example: MAS Communication





Example: MAS Communication



Example: MAS Communication

https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/workshops/wesaac/2024/examples/lcdHermesComm.chon

communicator | Communicator

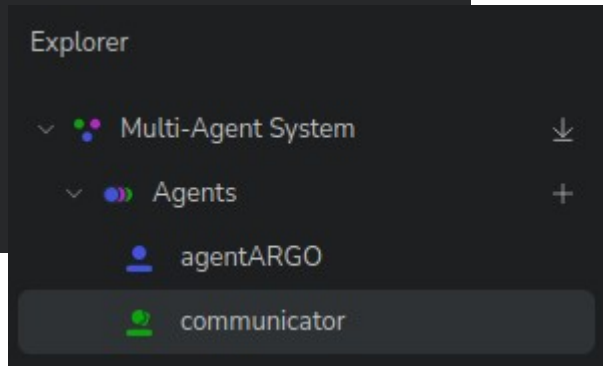
```
/* To generate a random UUID use https://www.uuidgenerator.net/ */
myUUID("80d9c5b3-5327-4836-b722-7481061affef").

/* We provide a Public IoT Gateway, more info in https://doi.org/10.5753/wei.2023.229753 */
gateway("192.168.0.53",5500).

!connect.

+!connect: gateway(Server,Port) & myUUID(ID) <-
        .print("Trying to connect in ",Server,":",Port);
        .connectCN(Server,Port,ID).

+!forward(Message, Who) <-
        .print("Forwarding a message to ", Who);
        .sendOut(Who, tell, msg(Message)).

+msg(Message)[source(Device)] <-
        .print("I received the following message: ", Message);
        .send(agentARGO, achieve, infoLCD(Message));
        -msg(Message)[source(Device)].

```

Example: MAS Communication

https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/workshops/wesaac/2024/examples/lcdHermesComm.chon

communicator | Communicator

```
/* To generate a random UUID use https://www.uuidgenerator.net/ */
myUUID("80d9c5b3-5327-4836-b722-7481061affef").

/* We provide a Public IoT Gateway, more info in https://doi.org/10.5753/wei.2023.229753 */
gateway("192.168.0.53",5500).

!connect.

+!connect: gateway(Server,Port) & myUUID(ID) <-
        .print("Trying to connect in ",Server,":",Port);
        .connectCN(Server,Port,ID).

+!forward(Message, Who) <-
        .print("Forwarding a message to ", Who);
        .sendOut(Who, tell, msg(Message)).

+msg(Message)[source(Device)] <-
        .print("I received the following message: ", Message);
        .send(agentARGO, achieve, infoLCD(Message));
        -msg(Message)[source(Device)].

```



Example: MAS Communication

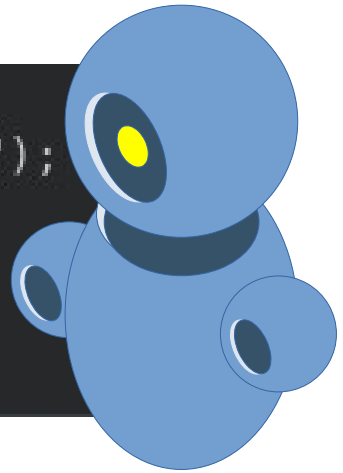
```
1       /* To generate a random UUID use https://www.uuidgenerator.net/ */
2       myUUID("80d9c5b3-5327-4836-b722-7481061affef").
3
4       /* We provide a Public IoT Gateway, more info in https://doi.org/10.5753/wei.2023.229753 */
5       gateway("192.168.0.53",5500).
6
7       !connect.
8
9       +!connect: gateway(Server,Port) & myUUID(ID) <-
10              .print("Trying to connect in ",Server,":",Port);
11              .connectCN(Server,Port,ID).
12
```

Example: MAS Communication

```
1       /* To generate a random UUID use https://www.uuidgenerator.net/ */
2       myUUID("80d9c5b3-5327-4836-b722-7481061affef").
3
4       /* We provide a Public IoT Gateway, more info in https://doi.org/10.5753/wei.2023.229753 */
5       gateway("192.168.0.53",5500).
6
7       !connect.
8
9       +!connect: gateway(Server,Port) & myUUID(ID) <-
10              .print("Trying to connect in ",Server,":",Port);
11              .connectCN(Server,Port,ID).
12
```

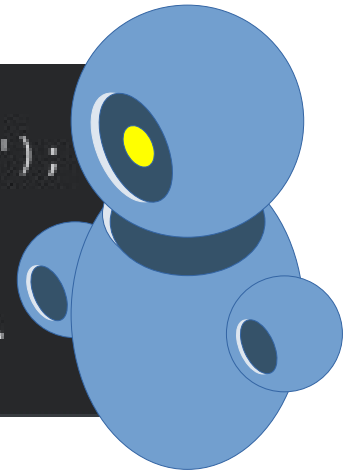
Example: MAS Communication

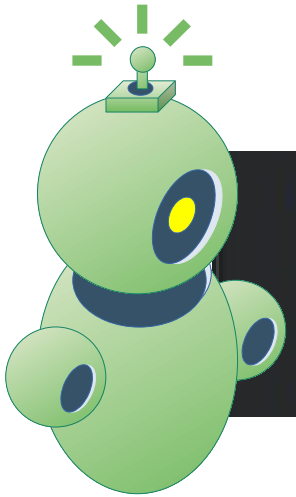
```
42      +temperature(T): T > 25 <-
43              .print("New perception-> temperature: ",T," Celsius");
44              .concat("Friend's Temp. is ", T, LCDMessage);
45              ?neighborUUID(ID);
46              .send(communicator,achieve,forward(LCDMessage, ID)).
```



Example: MAS Communication

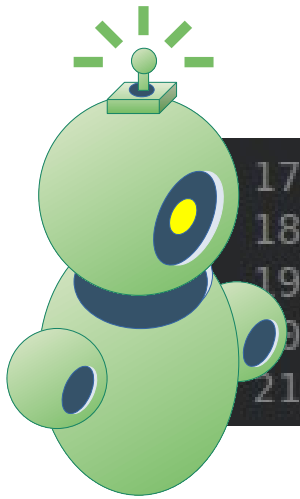
```
42      +temperature(T): T > 25 <-
43              .print("New perception-> temperature: ",T," Celsius");
44              .concat("Friend's Temp. is ", T, LCDMessage);
45              ?neighborUUID(ID);
46              .send(communicator,achieve,forward(LCDMessage, ID)).
```





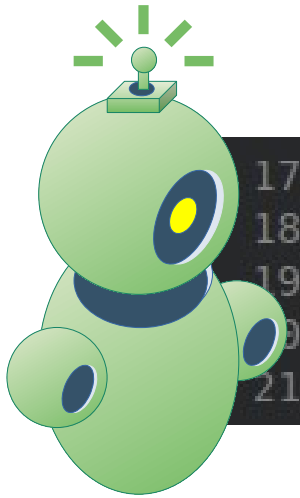
```
13      +!forward(Message, Who) <-
14              .print("Forwarding a message to ", Who);
15              .sendOut(Who, tell, msg(Message)).
16
```

Example: MAS Communication



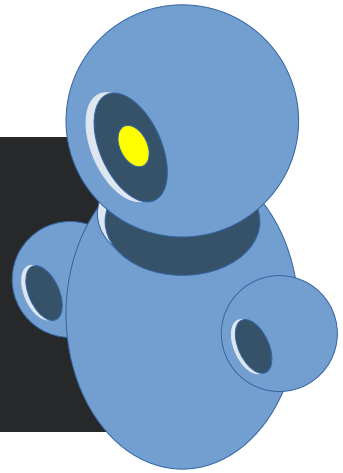
```
17      +msg(Message)[source(Device)] <-
18              .print("I received the following message: ", Message);
19              .send(agentARGO, achieve, infoLCD(Message));
20              -msg(Message)[source(Device)].
21
```

Example: MAS Communication



```
17      +msg(Message)[source(Device)] <-
18              .print("I received the following message: ", Message);
19              .send(agentARGO, achieve, infoLCD(Message));
20              -msg(Message)[source(Device)].
21
```

```
16      +!infoLCD(M)   <- .argo.act(M).
17      +!yellowAlert <- .argo.act(yellowOn).
18      +!greenAlert   <- .argo.act(greenOn).
19      +!redAlert     <- .argo.act(redOn).
20      +!blueAlert    <- .argo.act(blueOn).
21
```



Simple Exercise

Adjust the MAS 1 to:

- turn on the **red LED** if the temperature of **MAS 2** is higher than 25.

HERMES AGENT (MOBILITY)

Agent Transfer Protocol

Mobile agents can **transcend** their MAS and **move** to another MAS.

In **Embedded MAS**, an agent is part of a MAS embedded in a device. If the device is **damaged**, the agent and the entire **MAS are at risk**.

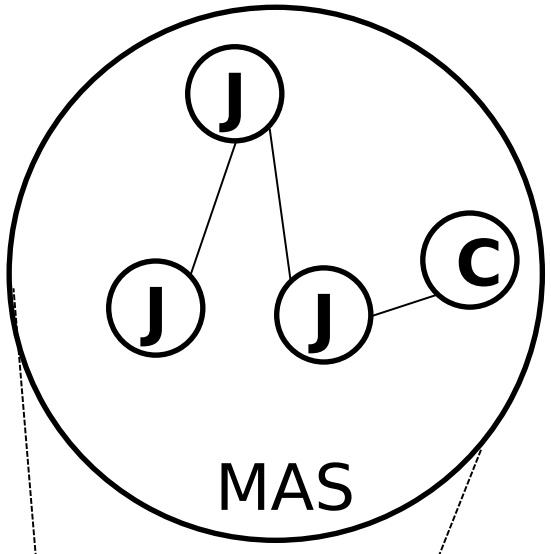
The **Agent Transfer Protocol** allows Jason's Hermes agents to **move** from **system to system** to **collaborate and preserve** acquired knowledge.

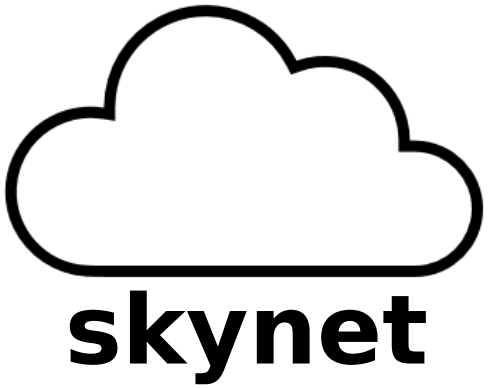
Agent Transfer Protocol

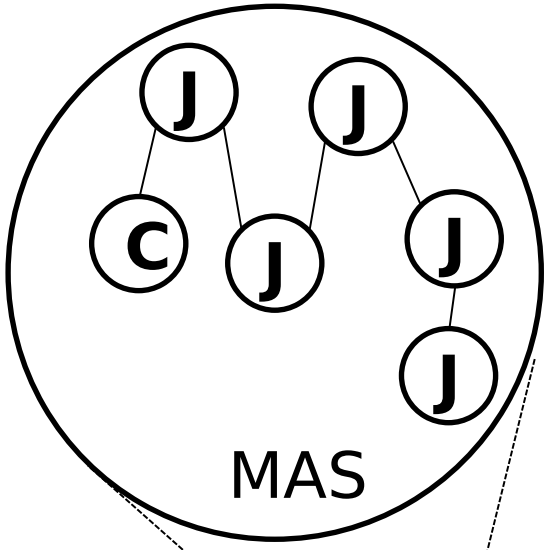
The agent transfer protocol has three possible **relationships** between **MAS**.

- Mutualism
- Inquilinism
- Predation

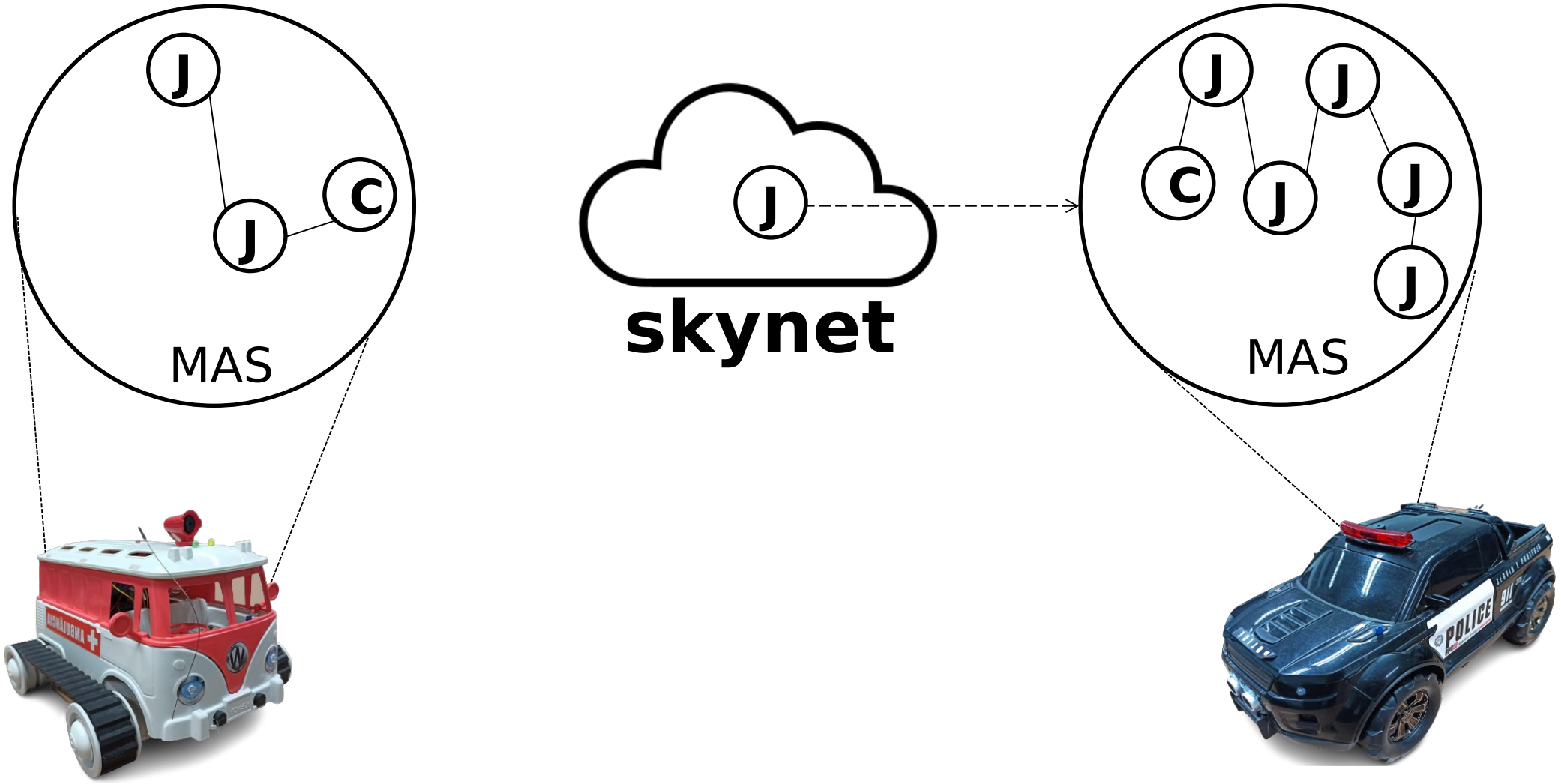
Bio-Inspired: Mutualism



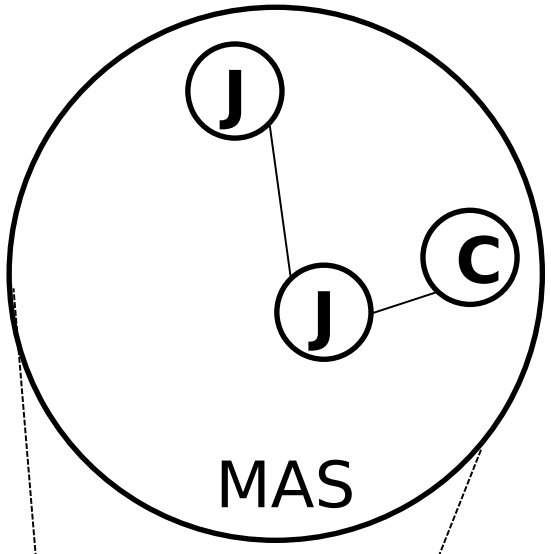




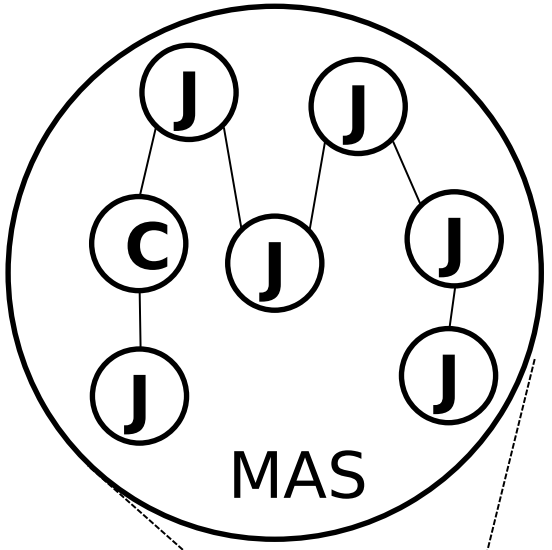
Bio-Inspired: Mutualism



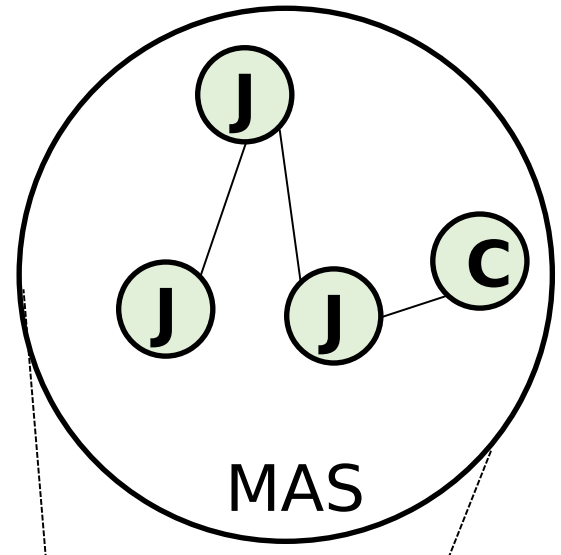
Bio-Inspired: Mutualism



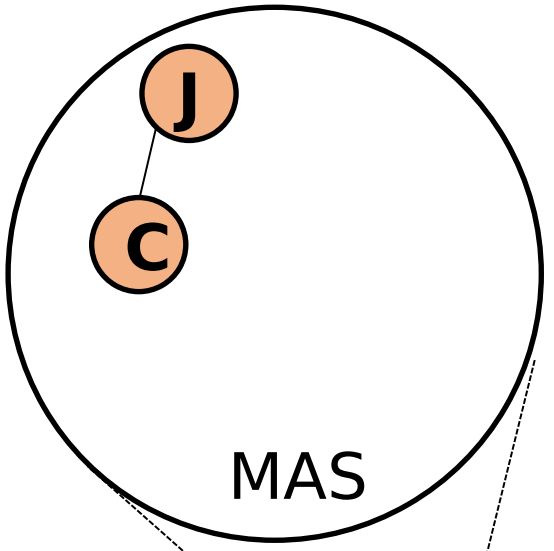
skynet



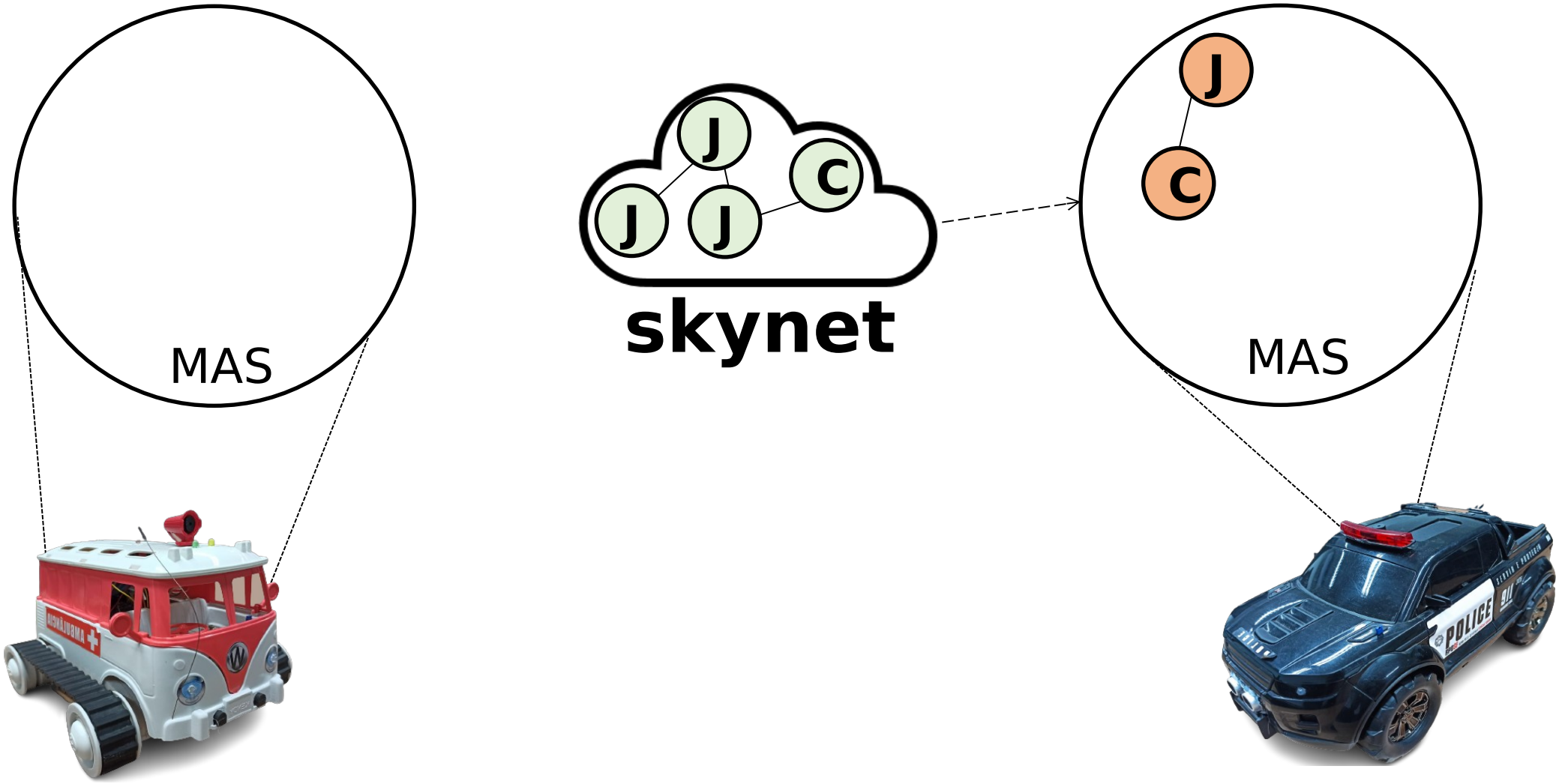
Bio-Inspired: Inquilinism



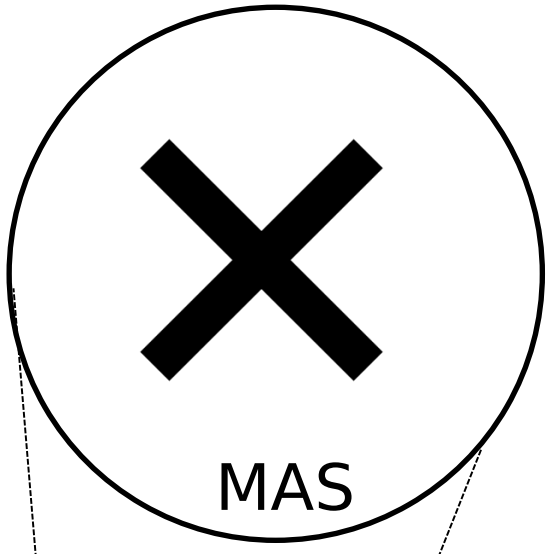
skynet

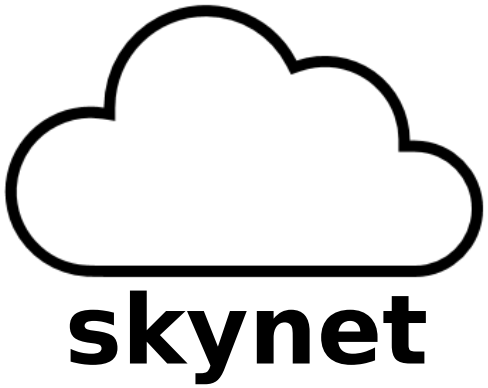


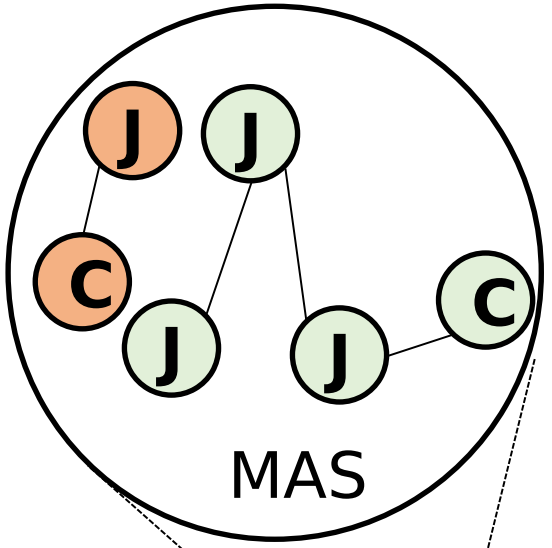
Bio-Inspired: Inquilinism



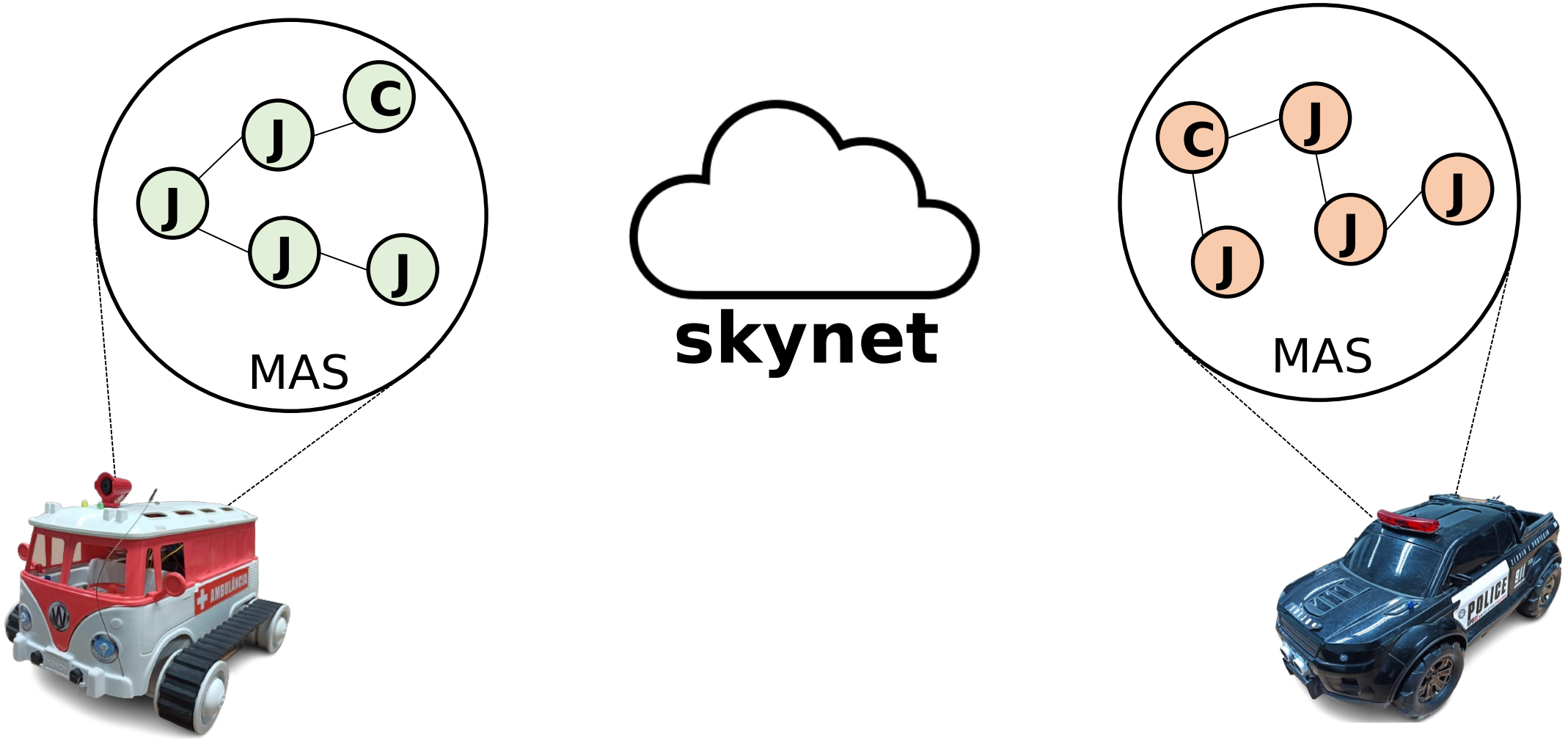
Bio-Inspired: Inquilinism



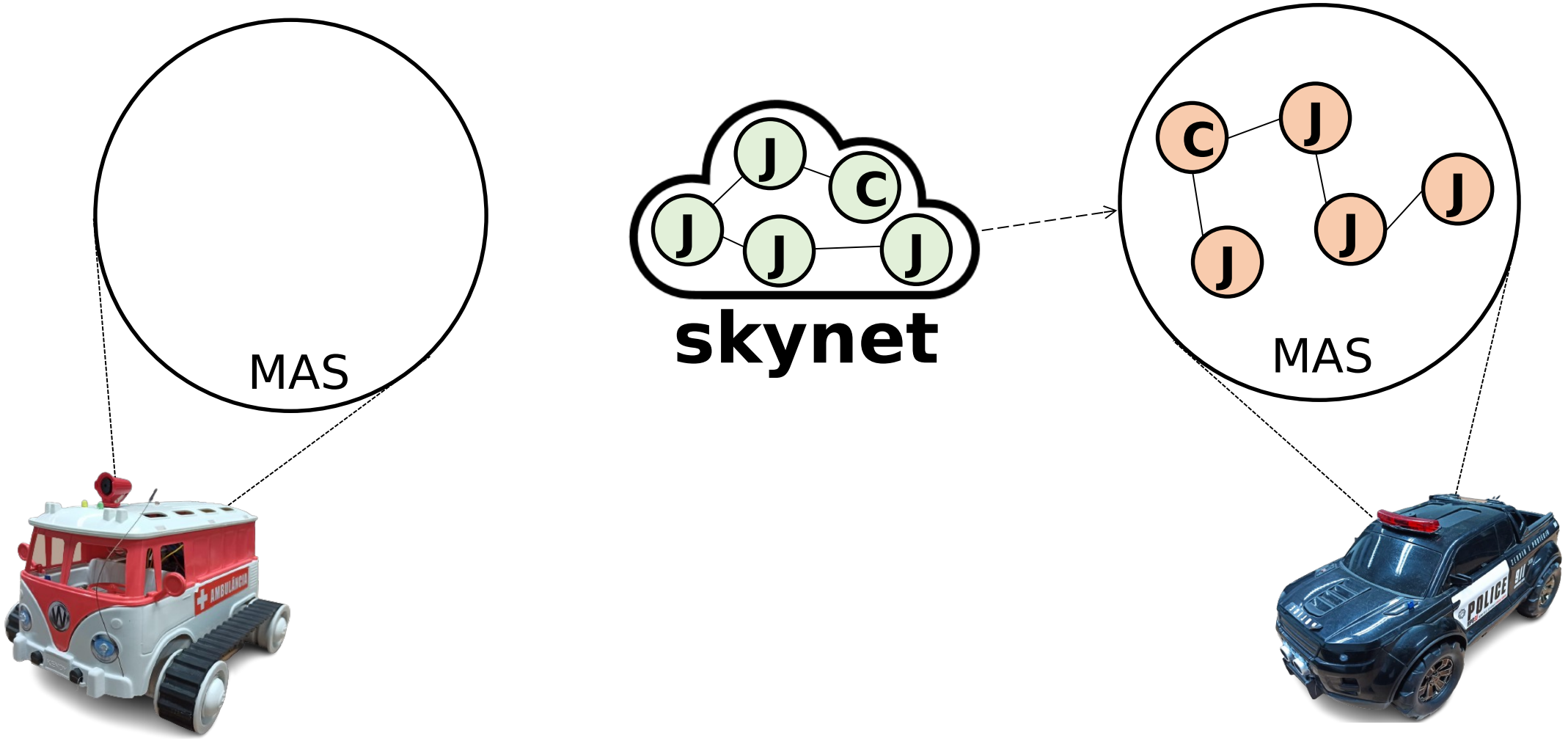




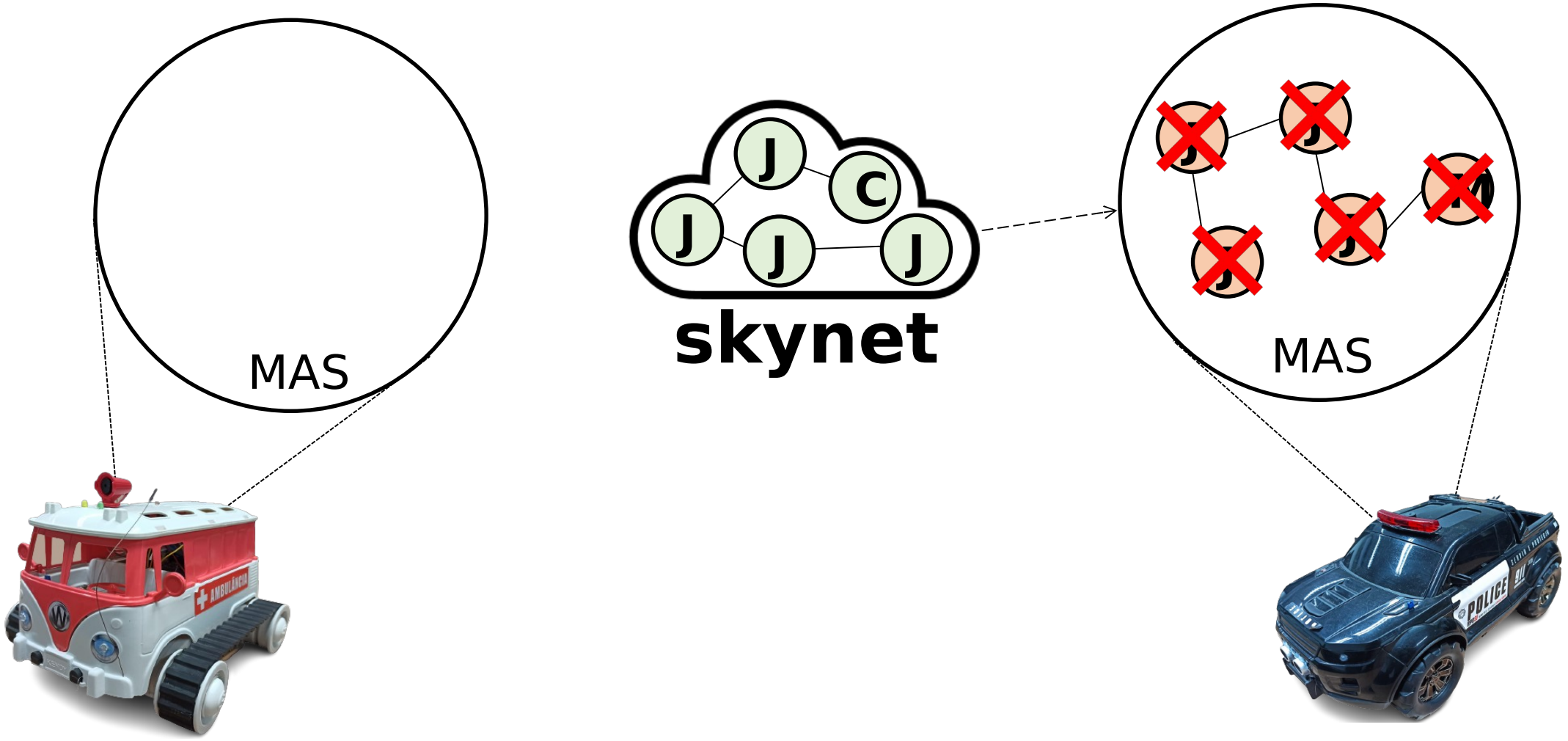
Bio-Inspired: Predation



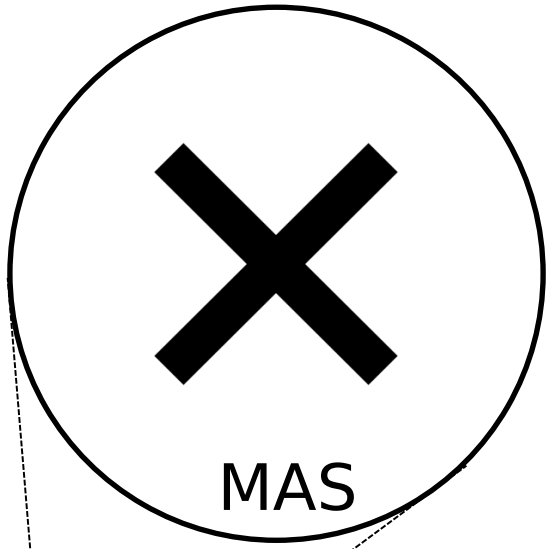
Bio-Inspired: Predation

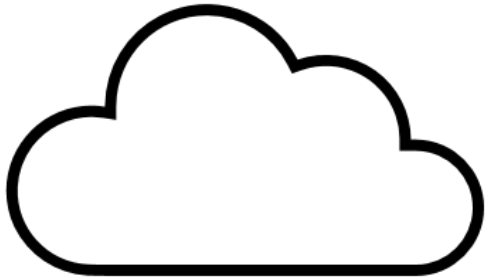


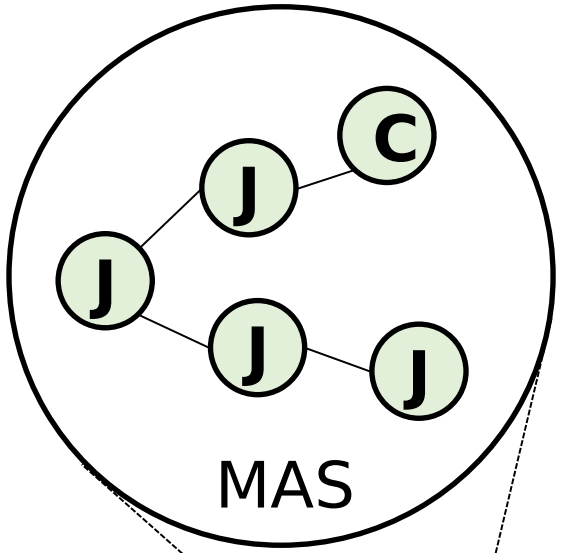
Bio-Inspired: Predation



Bio-Inspired: Predation







Hermes: Internal Actions

- **Hermes** Internal Actions:

.moveOut(agentUuid, predation|inquilinism)

Move all agents to a different known MAS.

.moveOut(agentUuid, mutualism, agent)

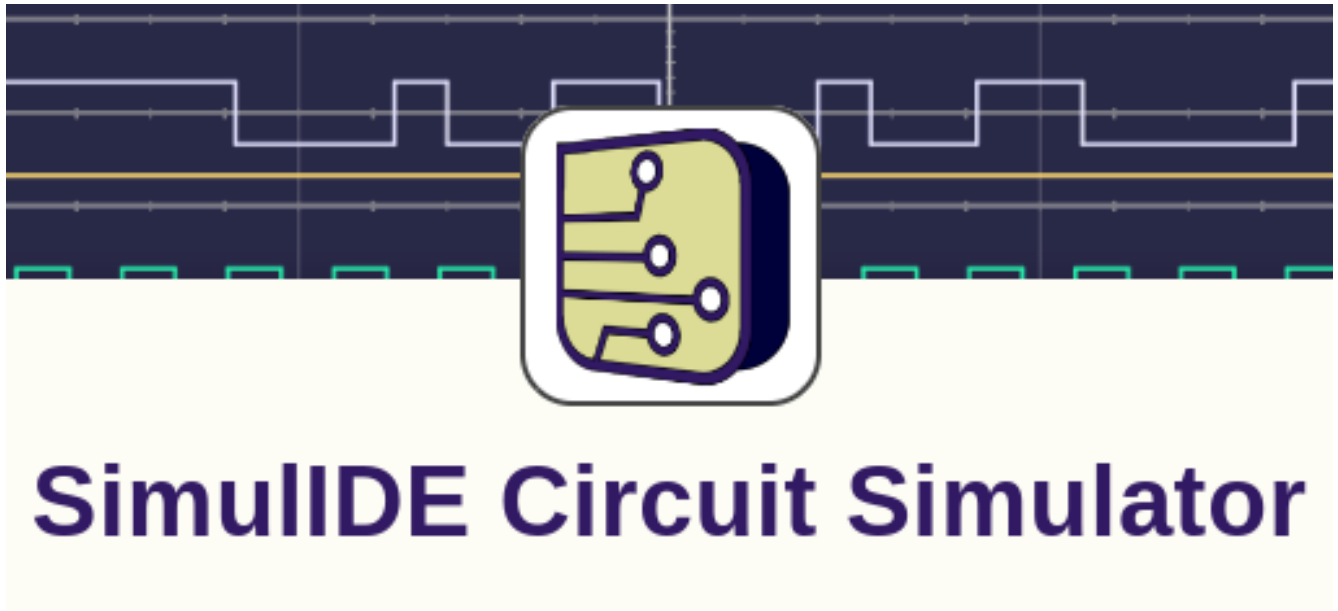
Move one specific agent to a different known MAS.

Example: Mutualism

Example: Predation

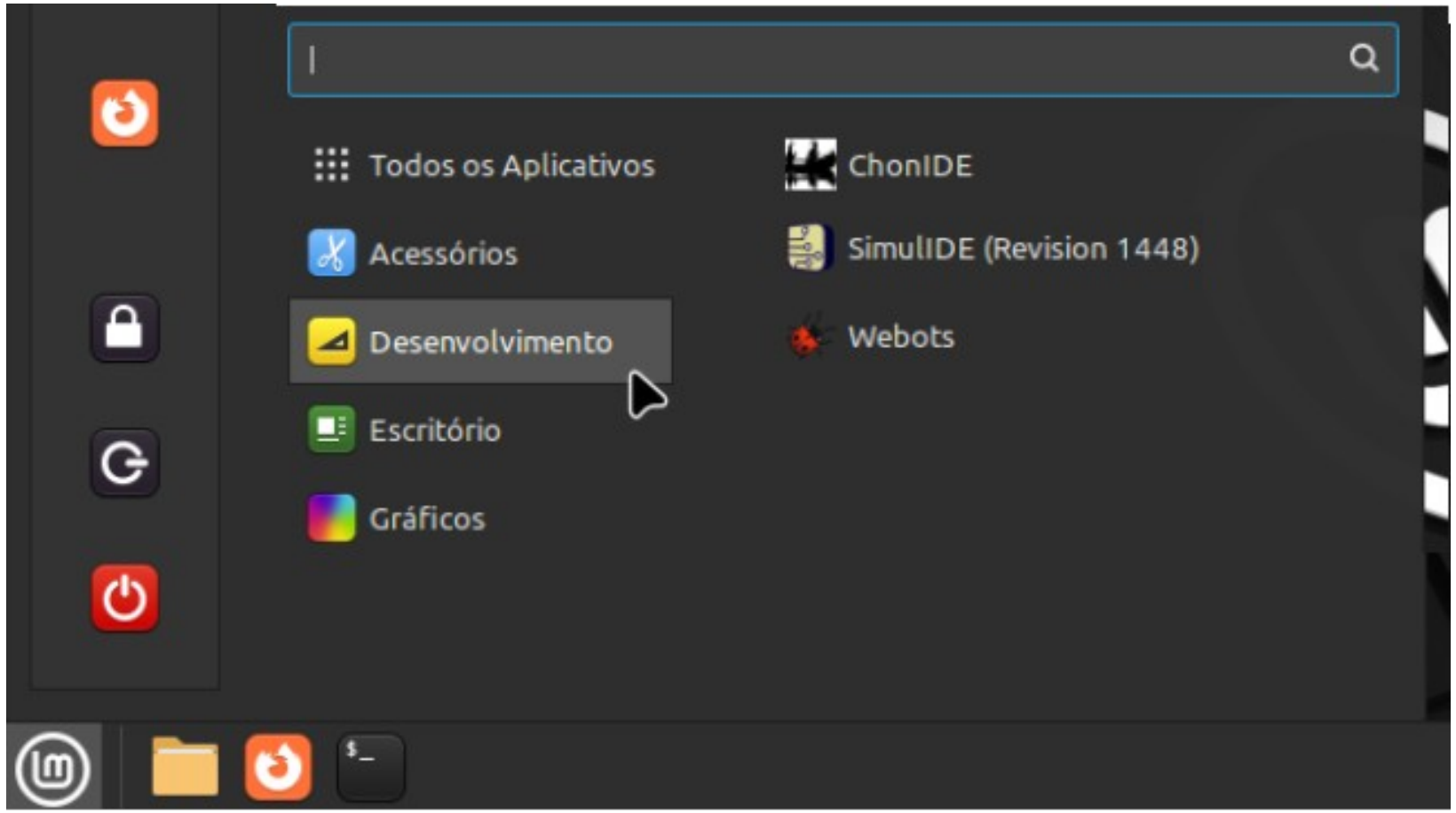
OTHER TOOLS

SimulIDE



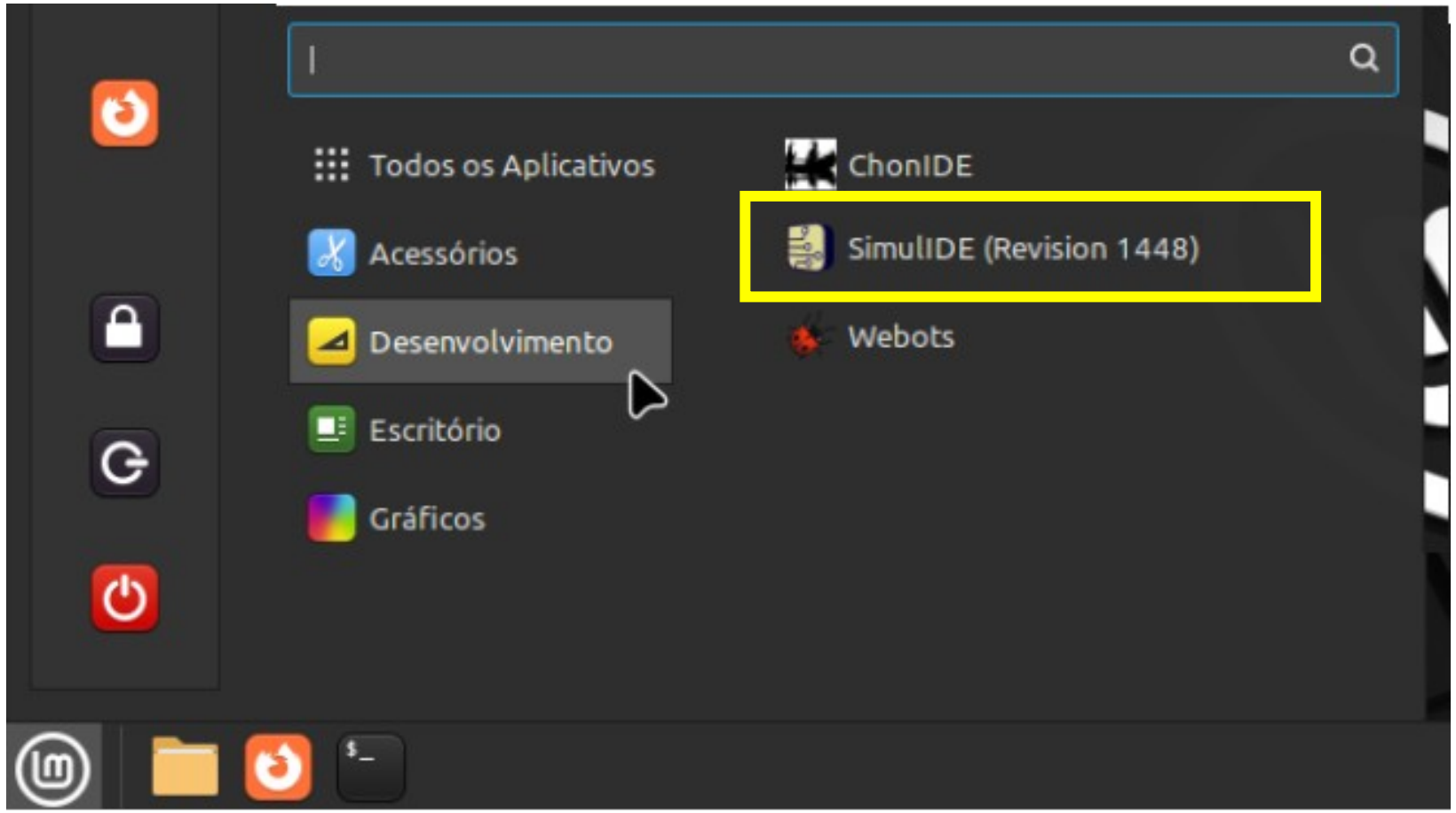
González, Santiago. “SimulIDE Circuit Simulator”, 2023. https://simulide.com.

SimulIDE



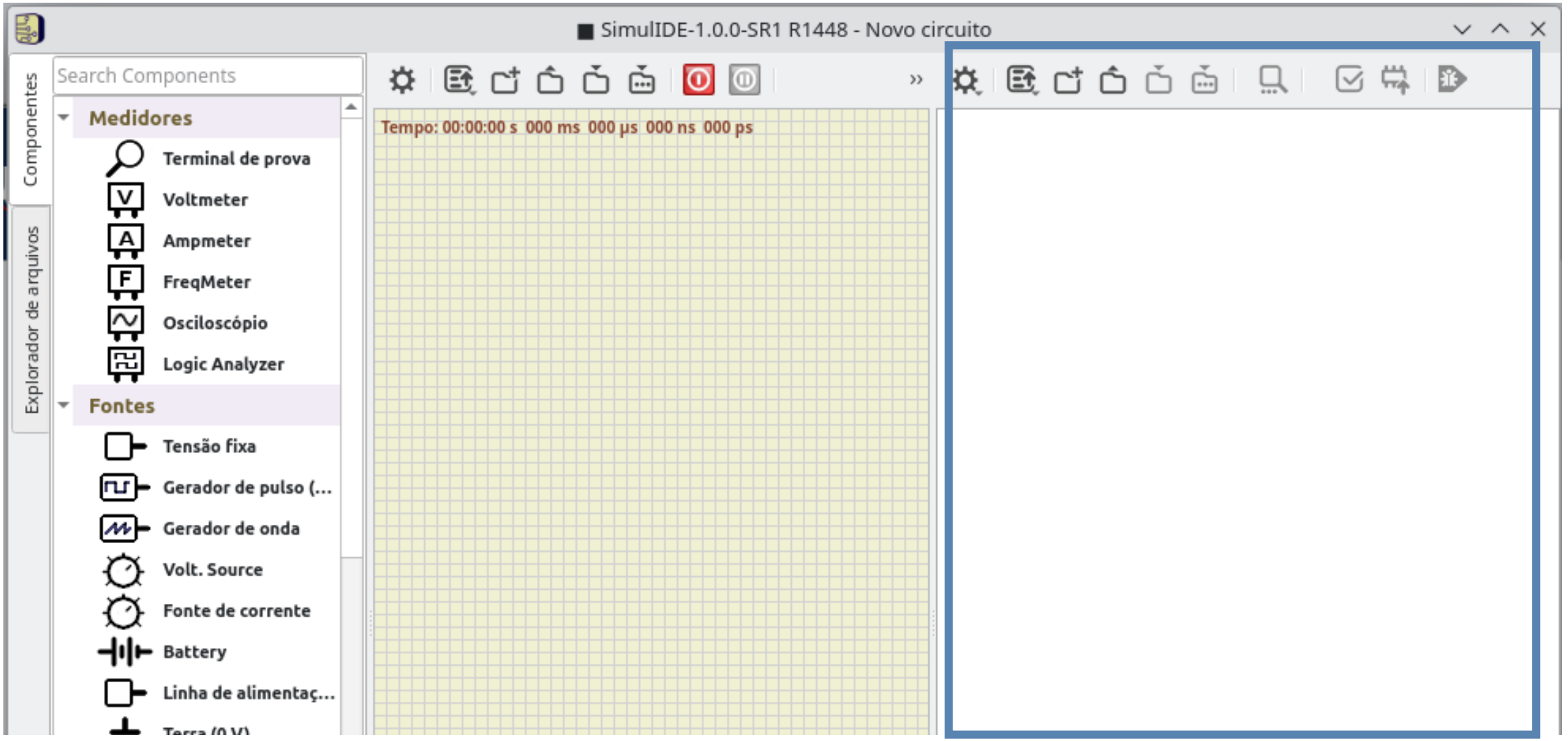
Manual de instalação
https://github.com/chon-group/dpkg-simulide

SimulIDE

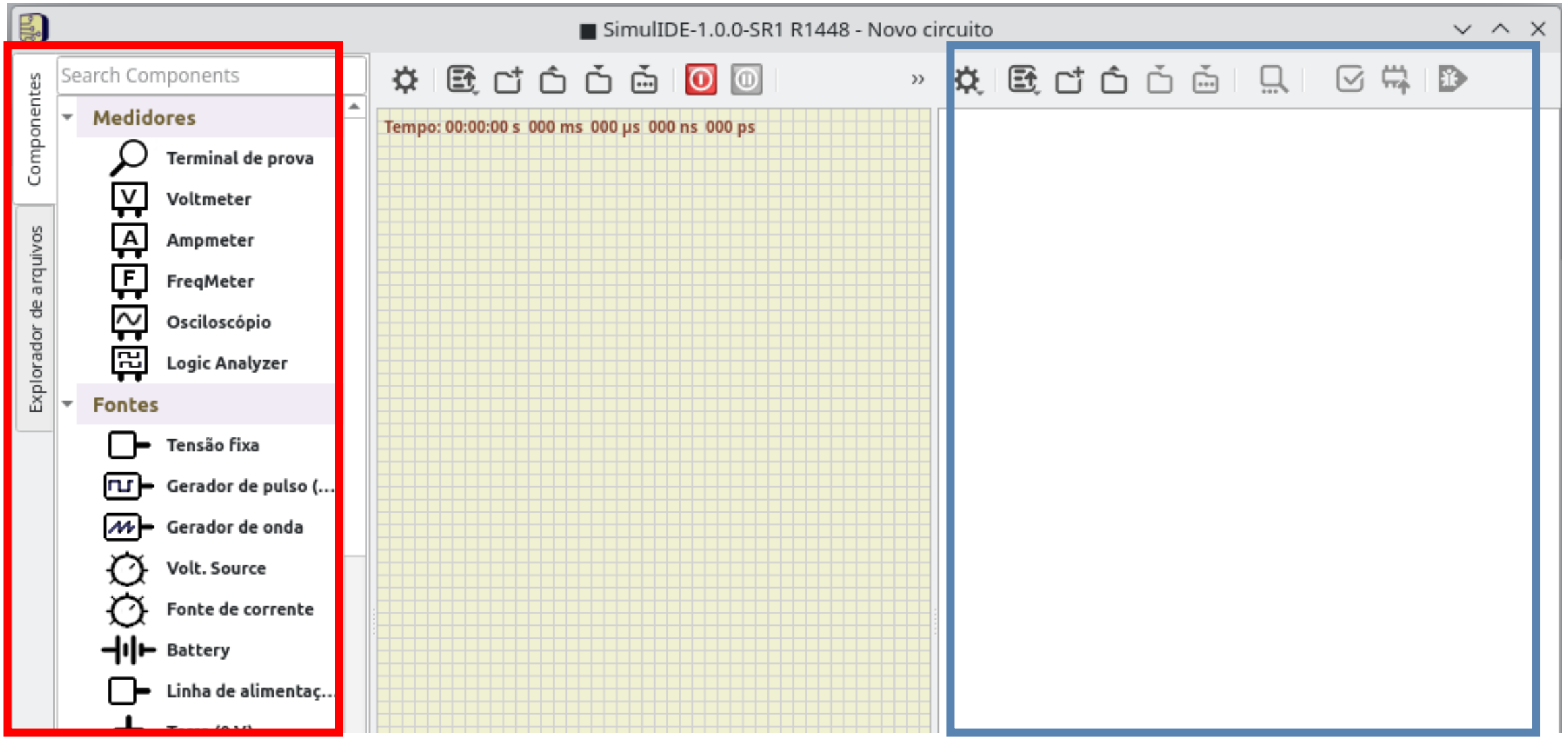


Manual de instalação
https://github.com/chon-group/dpkg-simulide

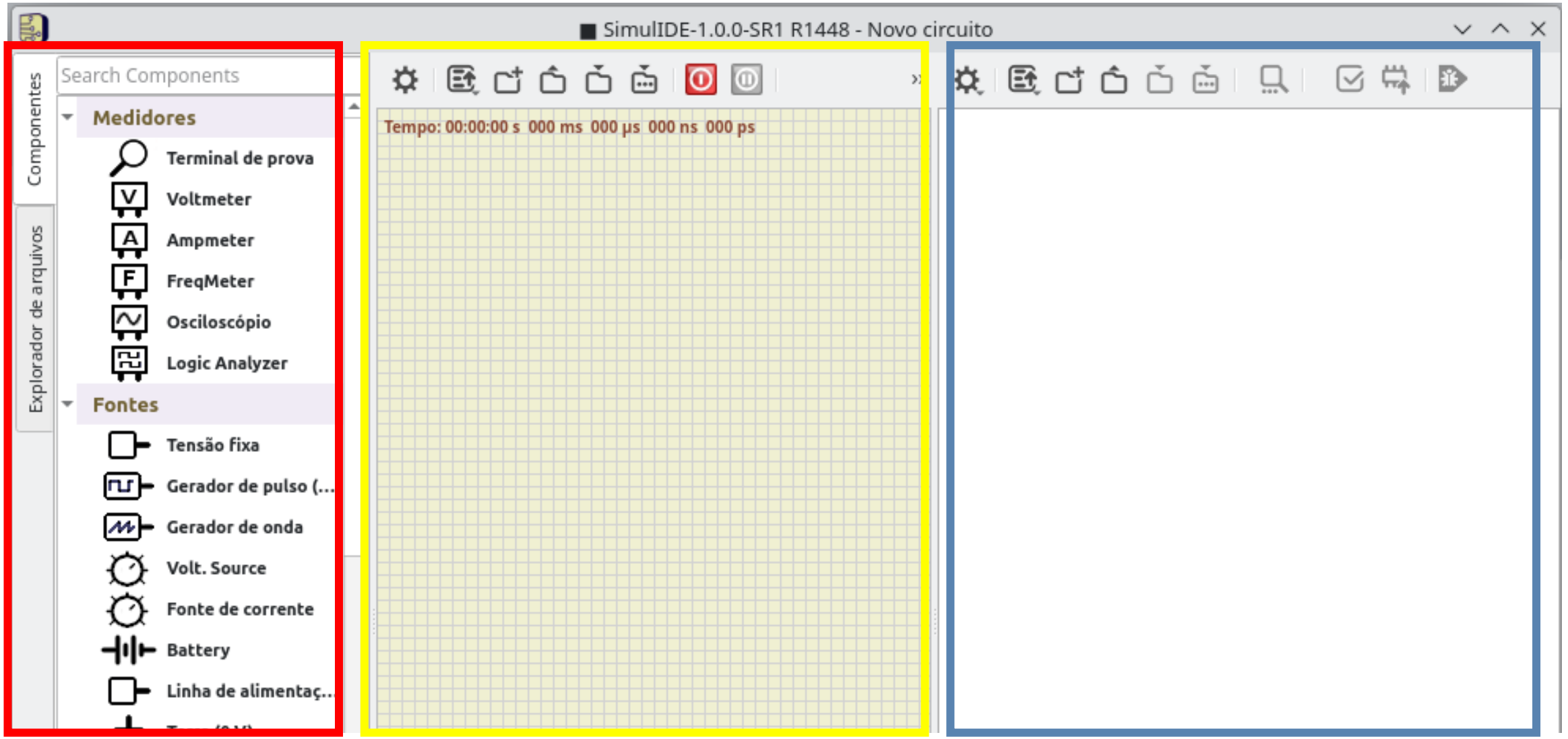
SimulIDE


SimulIDE-1.0.0-SR1 R1448 - Novo circuito
Componentes
Explorador de arquivos
Search Components
Medidores
Terminal de prova
Voltmeter
Ampmeter
FreqMeter
Osciloscópio
Logic Analyzer
Fontes
Tensão fixa
Gerador de pulso (...
Gerador de onda
Volt. Source
Fonte de corrente
Battery
Linha de alimentaç...
Tempo: 00:00:00 s 000 ms 000 µs 000 ns 000 ps

SimulIDE

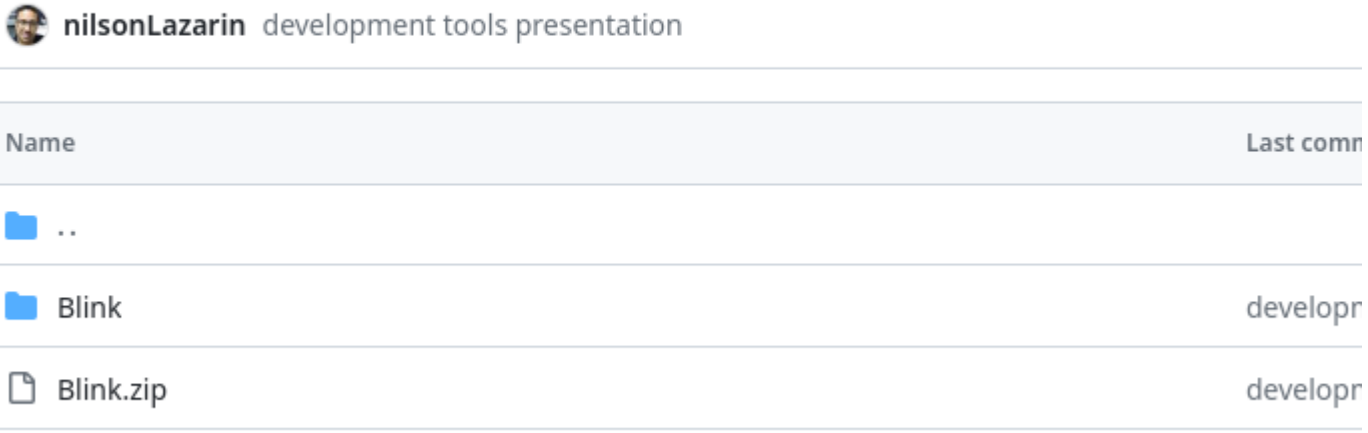

SimulIDE-1.0.0-SR1 R1448 - Novo circuito
Componentes
Explorador de arquivos
Search Components
Medidores
Terminal de prova
Voltmeter
Ampmeter
FreqMeter
Osciloscópio
Logic Analyzer
Fontes
Tensão fixa
Gerador de pulso (...
Gerador de onda
Volt. Source
Fonte de corrente
Battery
Linha de alimentaç...
Tempo: 00:00:00 s 000 ms 000 µs 000 ns 000 ps

SimulIDE


SimulIDE-1.0.0-SR1 R1448 - Novo circuito
Componentes
Explorador de arquivos
Search Components
Medidores
Terminal de prova
Voltmeter
Ampmeter
FreqMeter
Osciloscópio
Logic Analyzer
Fontes
Tensão fixa
Gerador de pulso (...
Gerador de onda
Volt. Source
Fonte de corrente
Battery
Linha de alimentaç...
Tempo: 00:00:00 s 000 ms 000 µs 000 ns 000 ps

SimulIDE: Blink

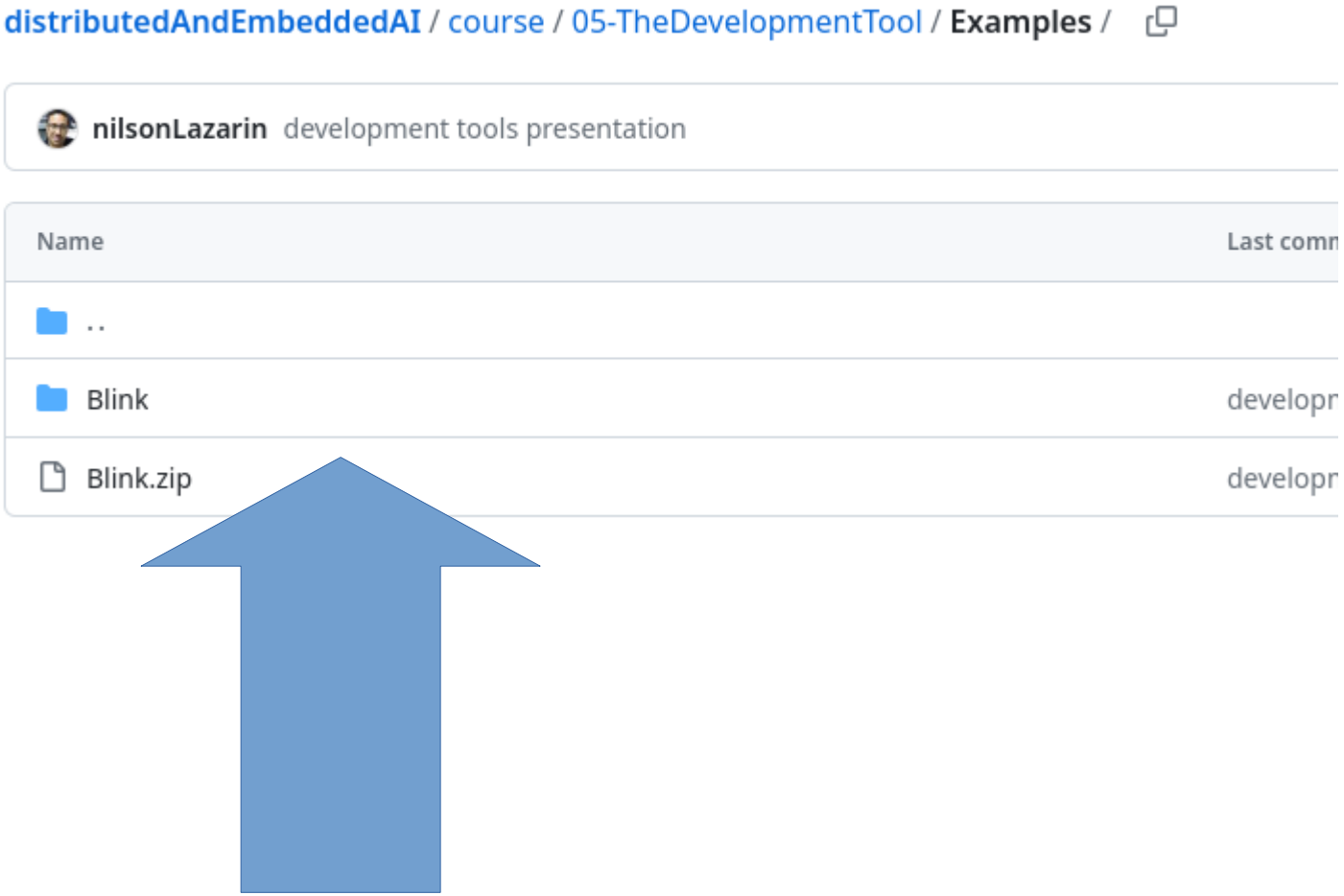
distributedAndEmbeddedAI / course / 05-TheDevelopmentTool / **Examples** /



https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/05-TheDevelopmentTool/Examples/Blink.zip

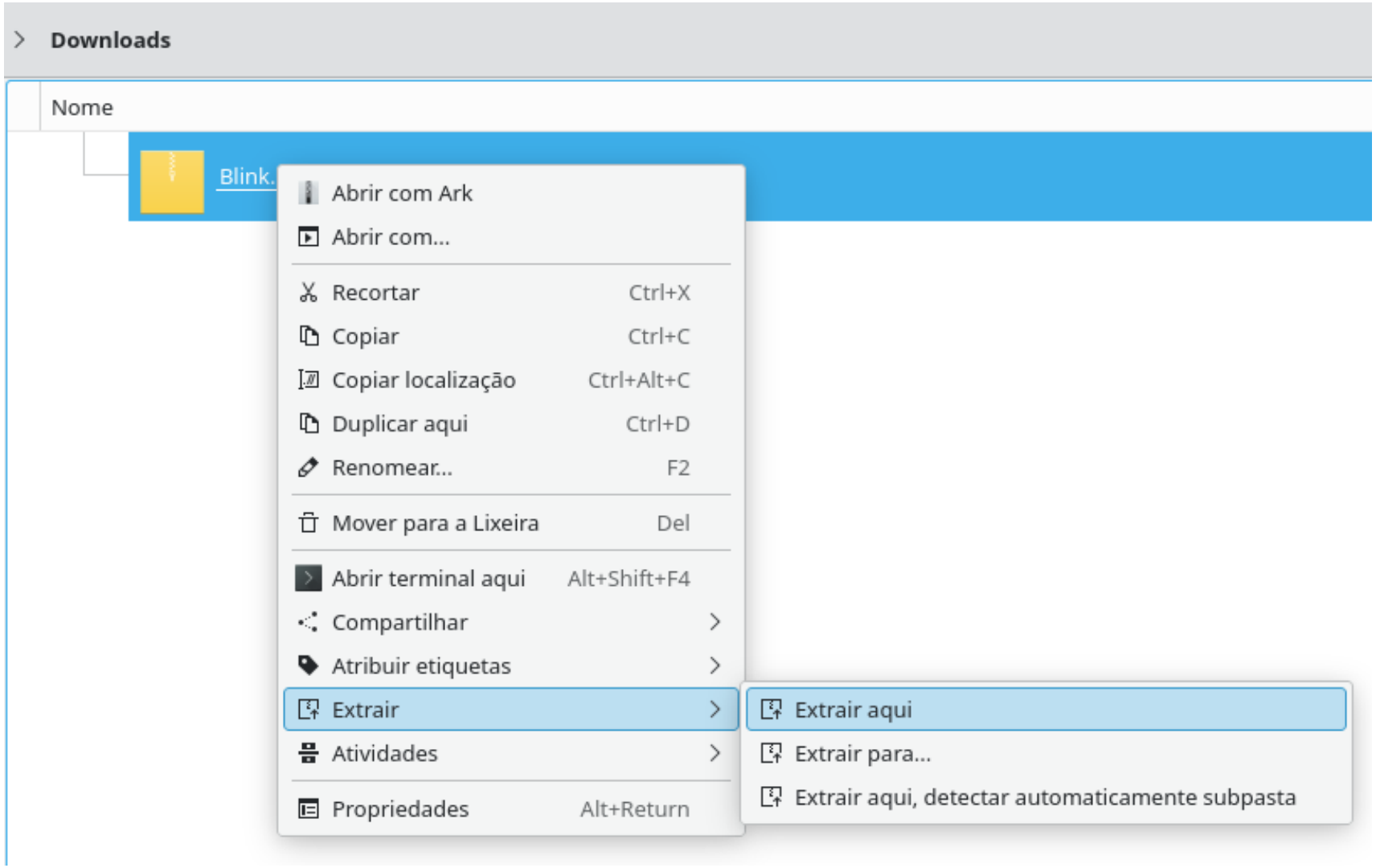
SimulIDE: Blink

distributedAndEmbeddedAI / course / 05-TheDevelopmentTool / **Examples** /

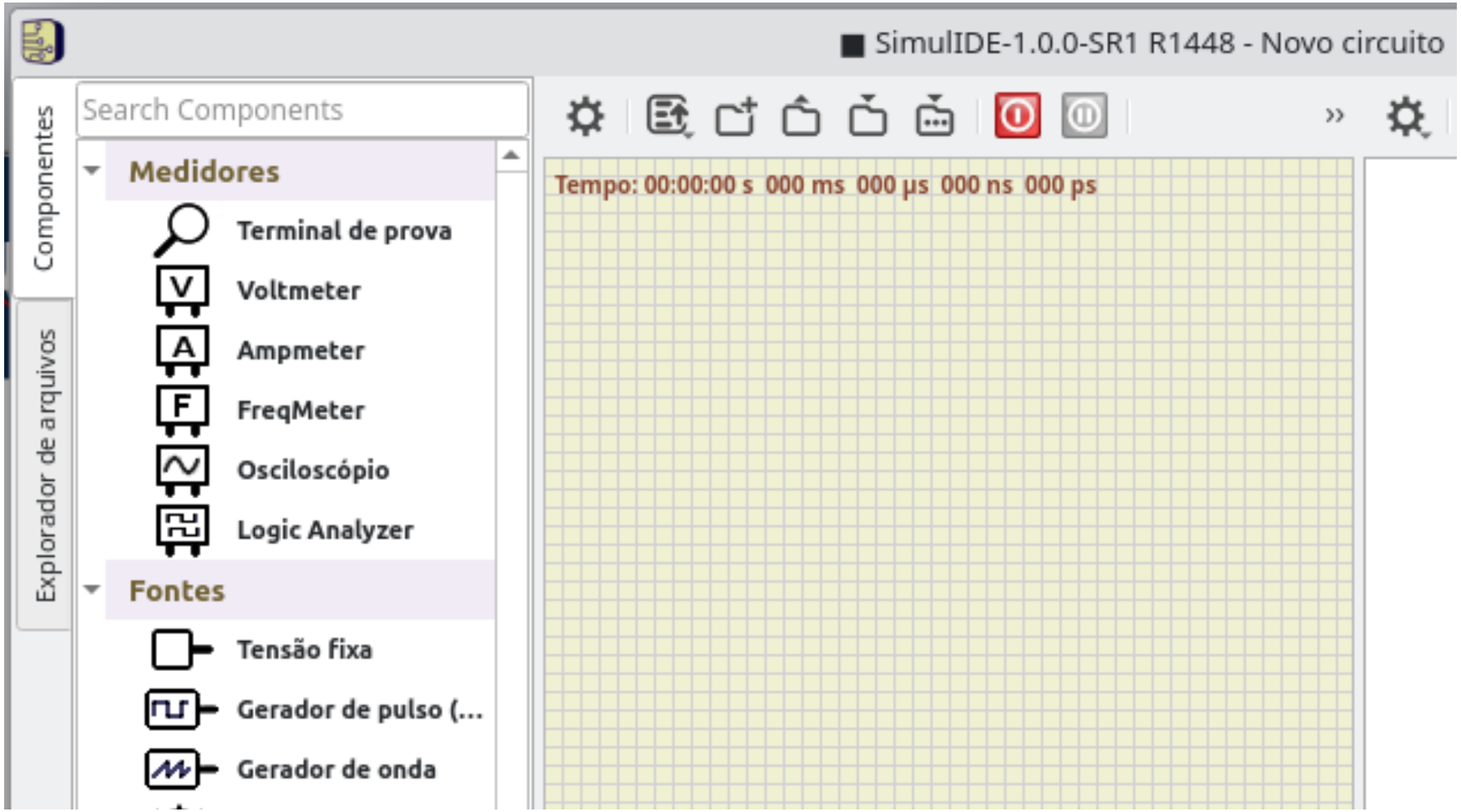


https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/05-TheDevelopmentTool/Examples/Blink.zip

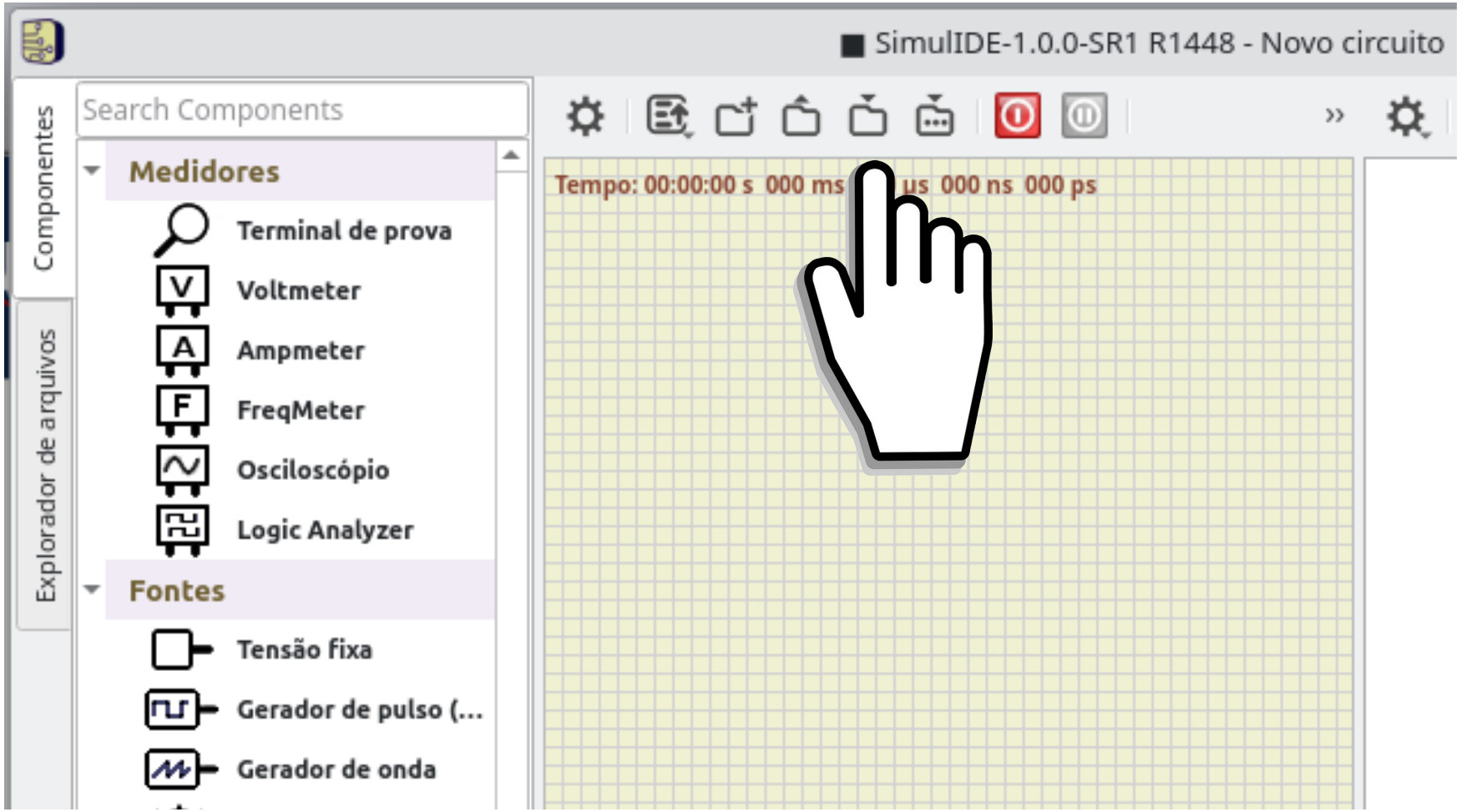
SimulIDE: Blink



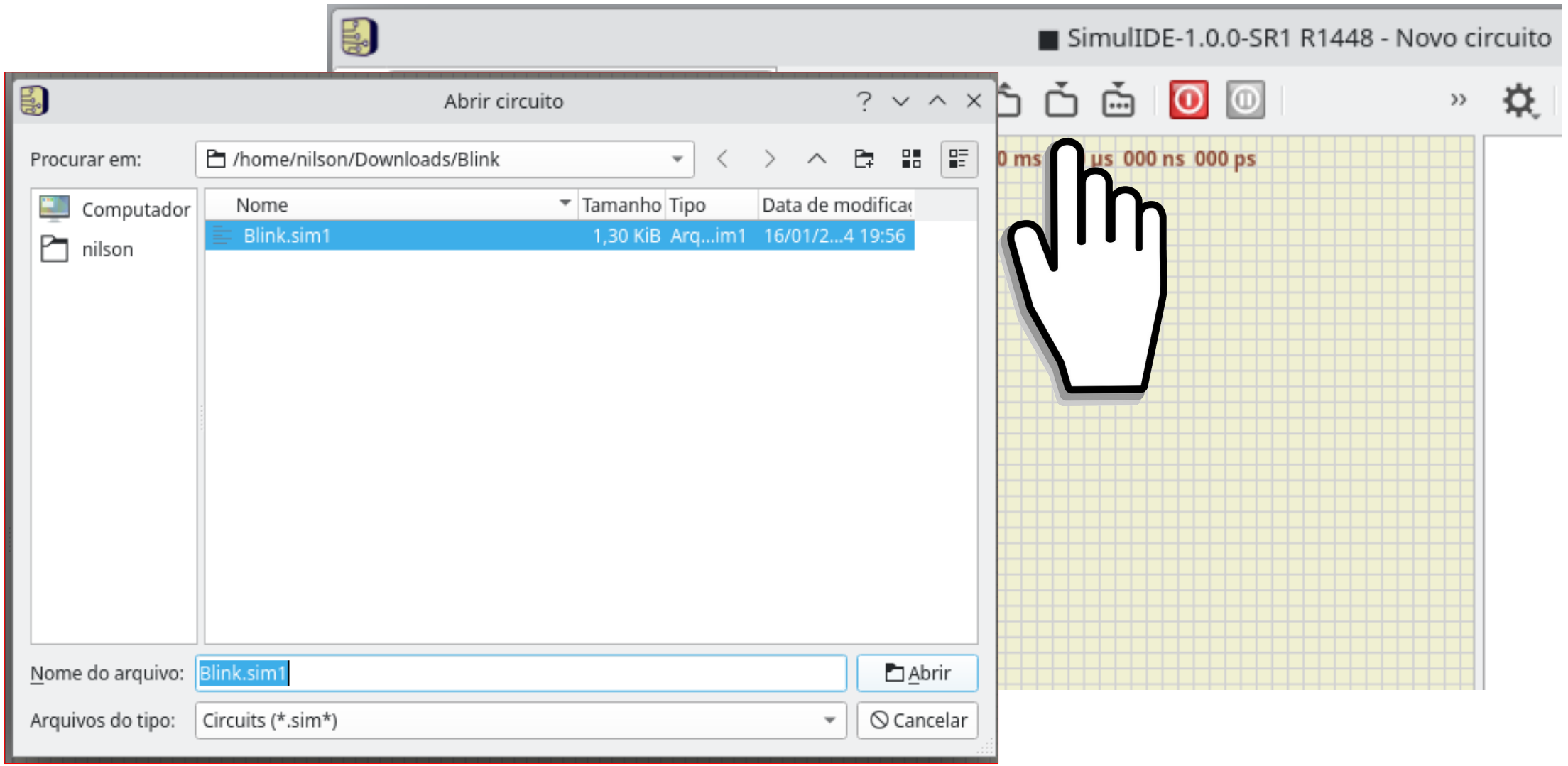
SimulIDE: Blink



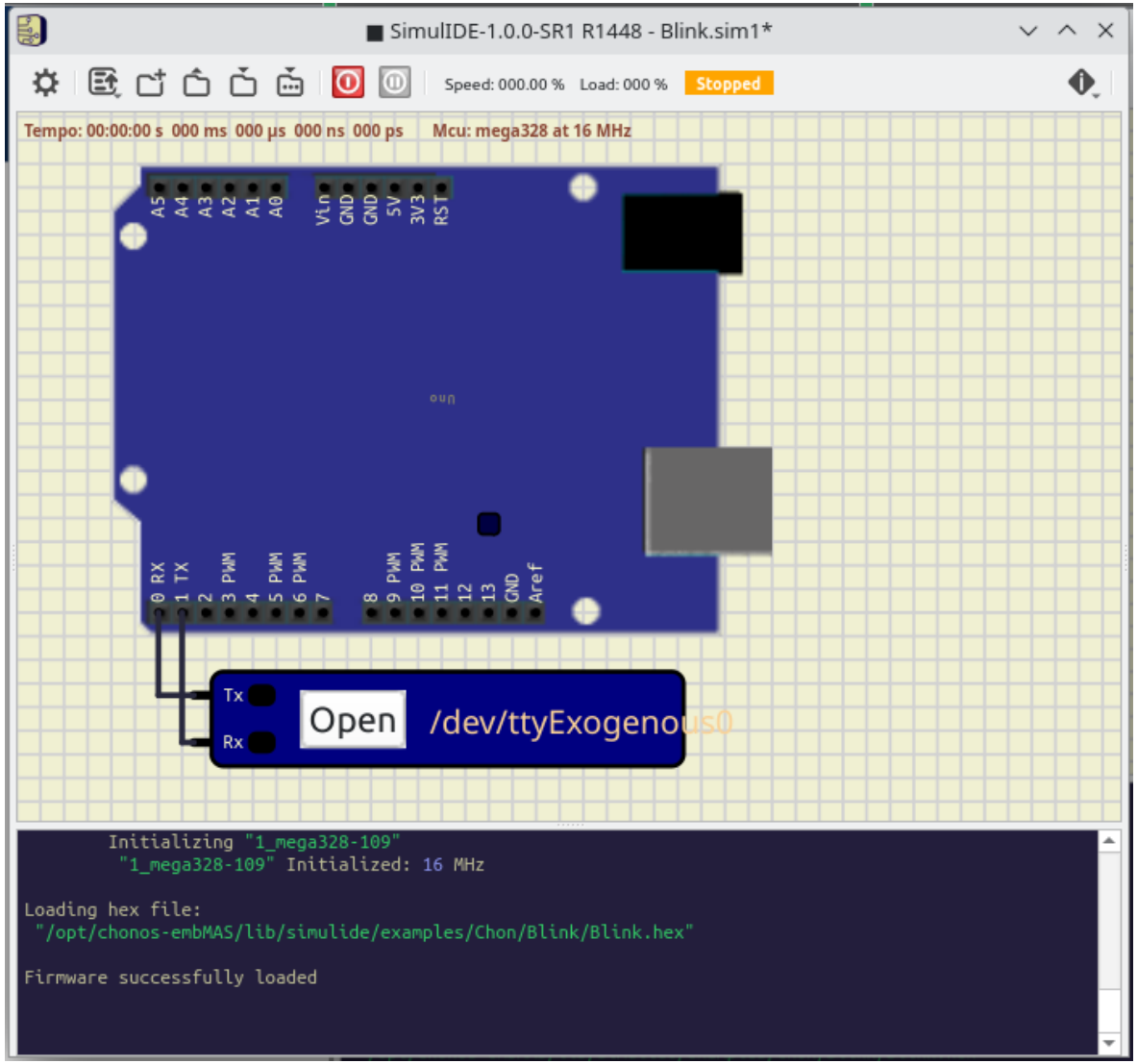
SimulIDE: Blink



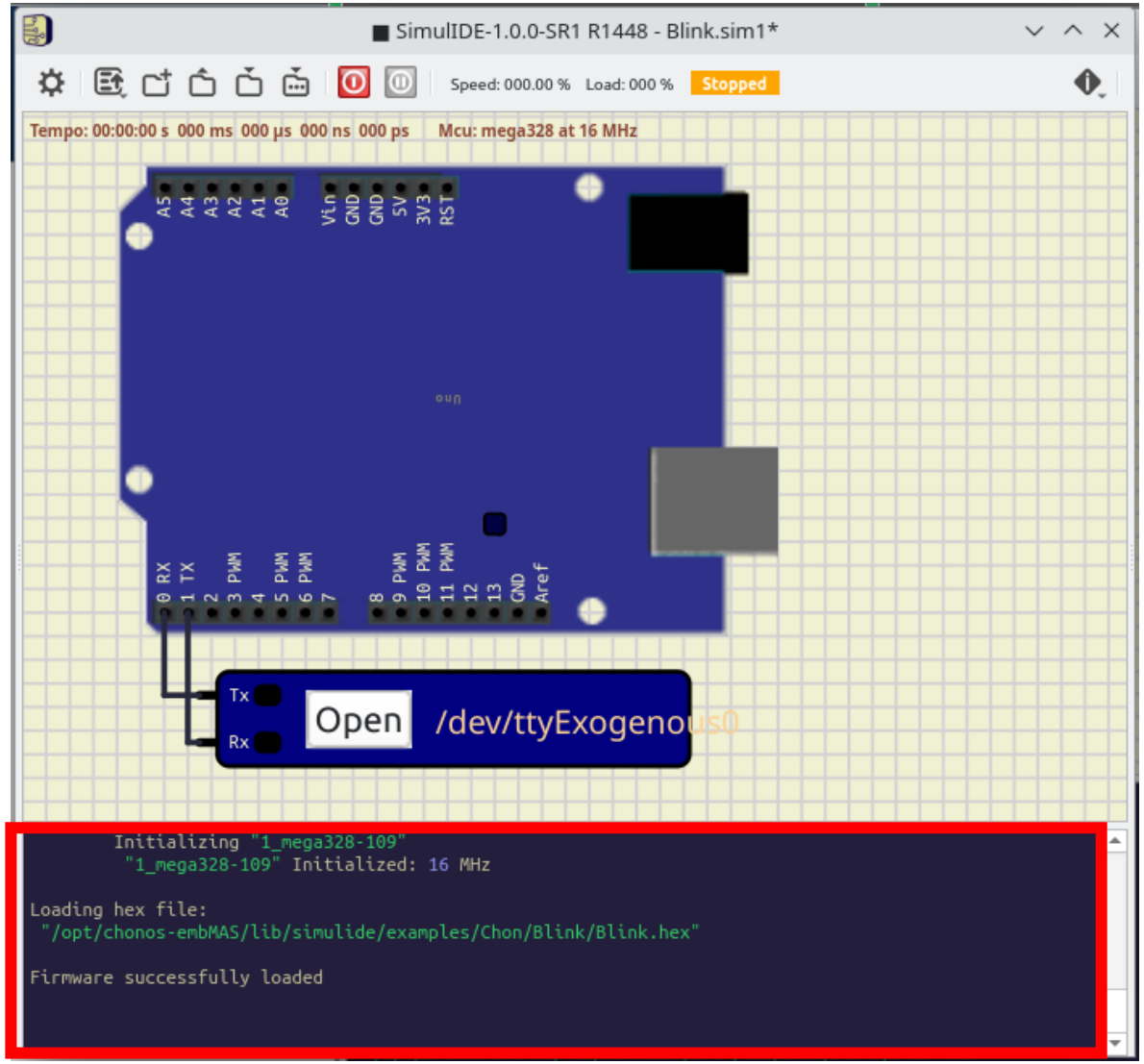
SimulIDE: Blink



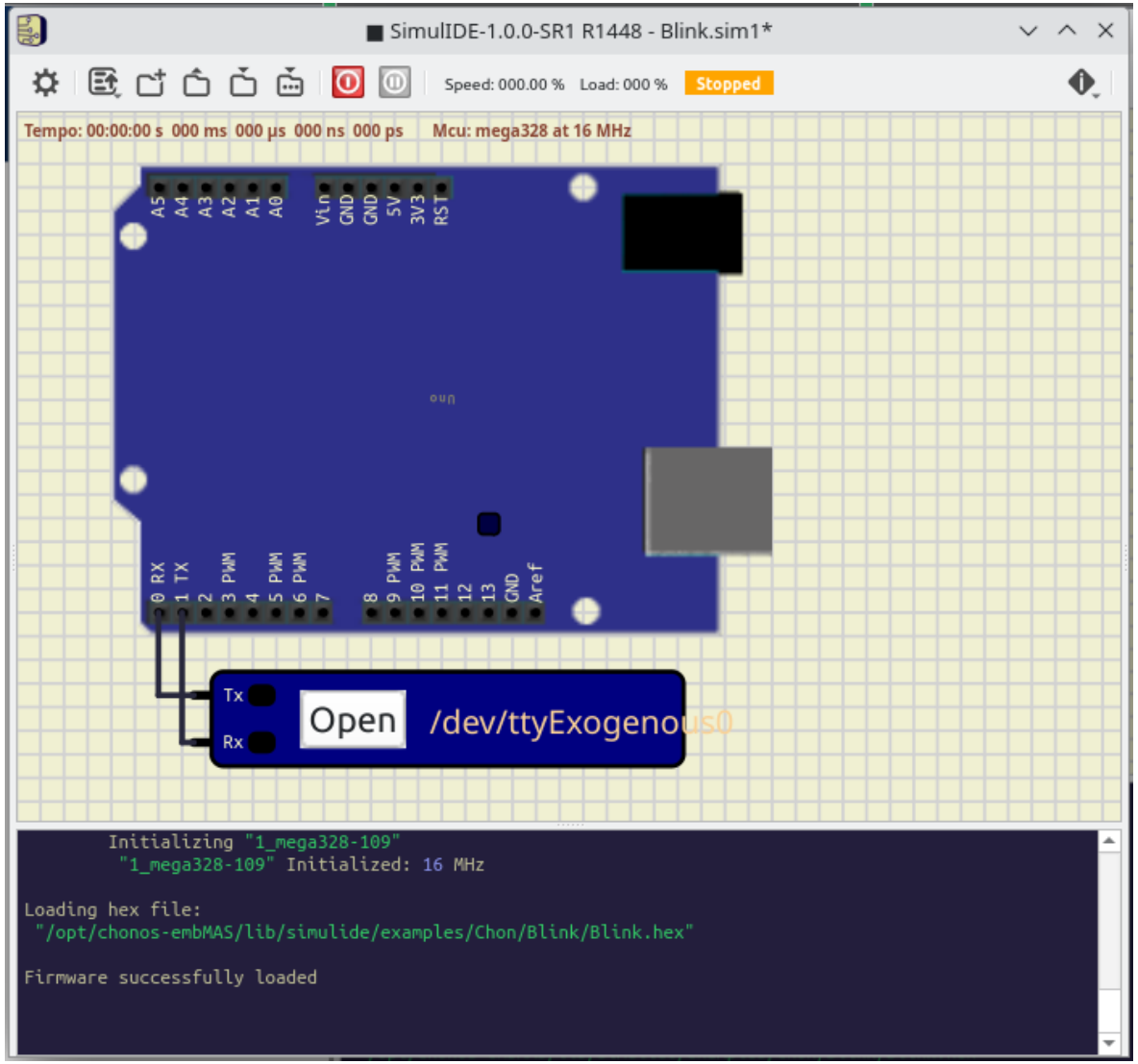
SimulIDE: Blink



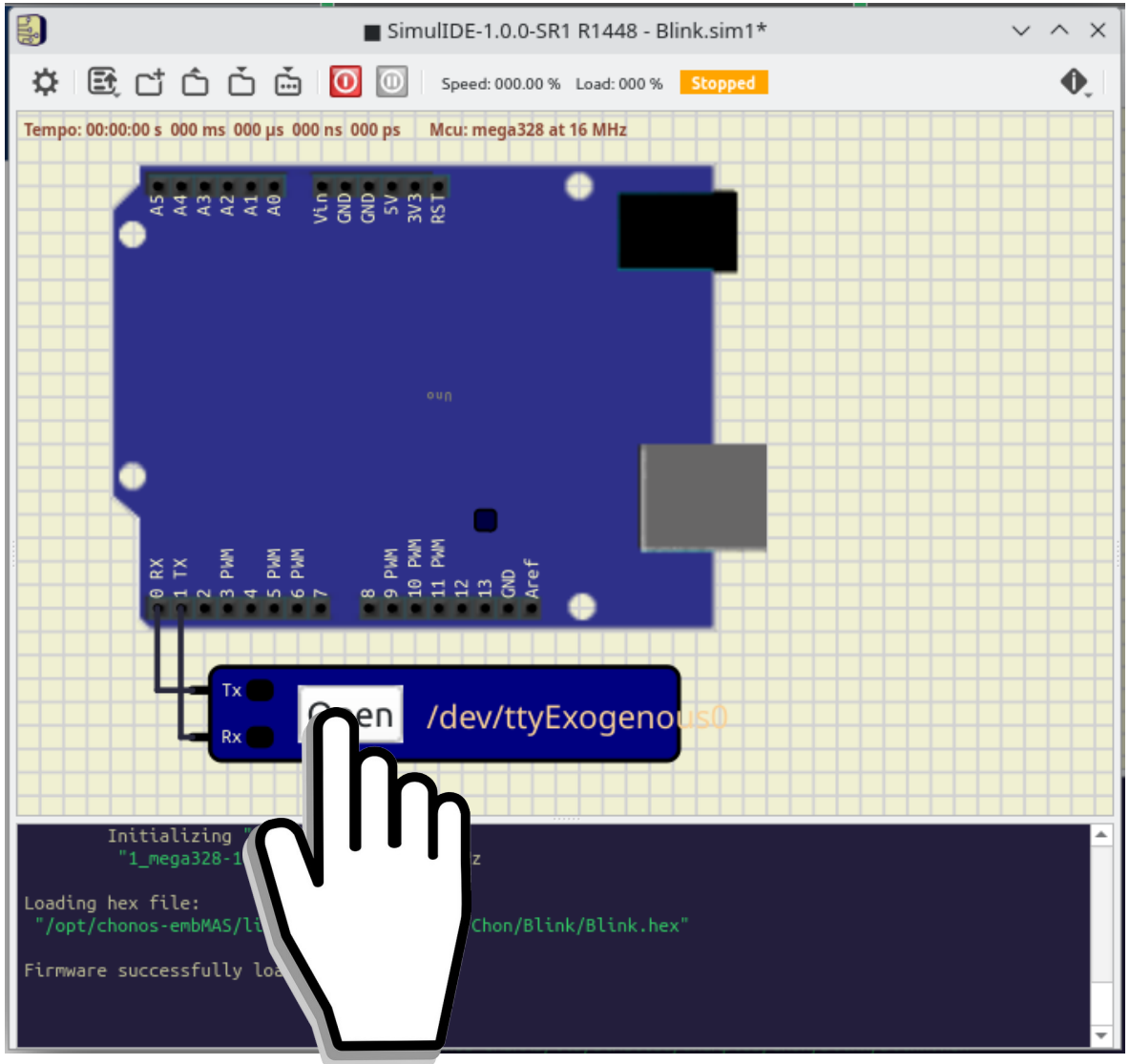
SimulIDE: Blink



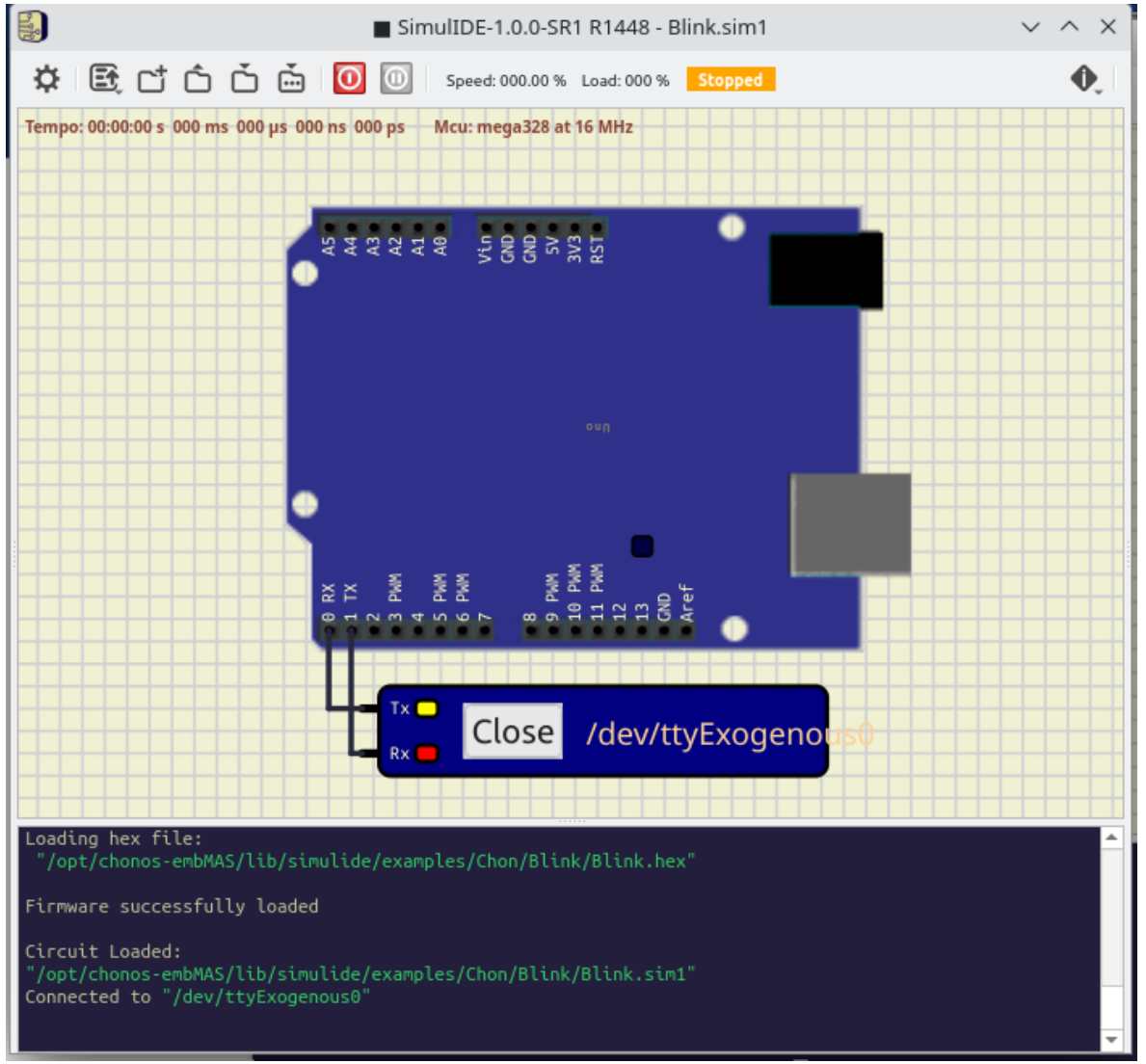
SimulIDE: Blink



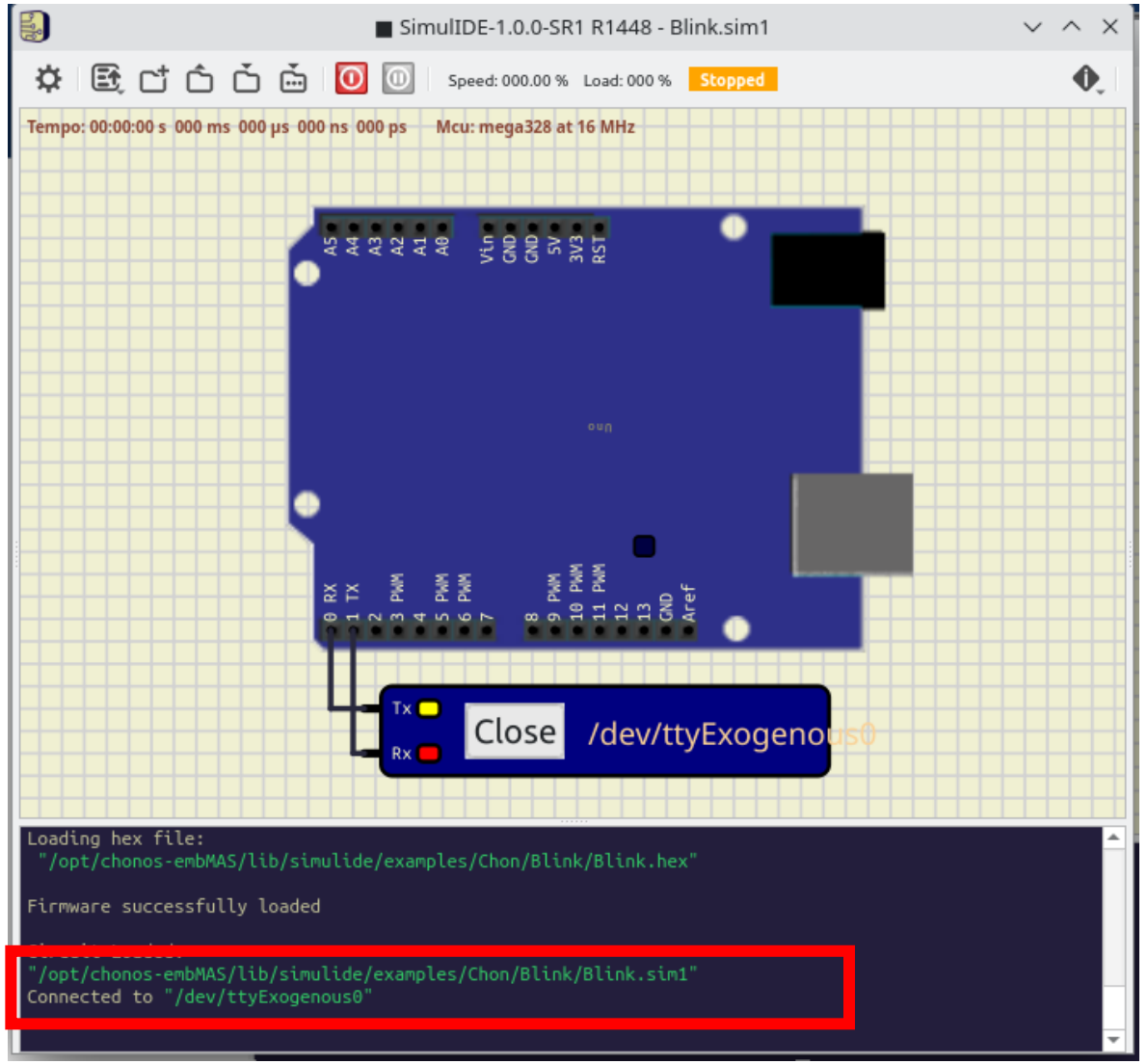
SimulIDE: Blink



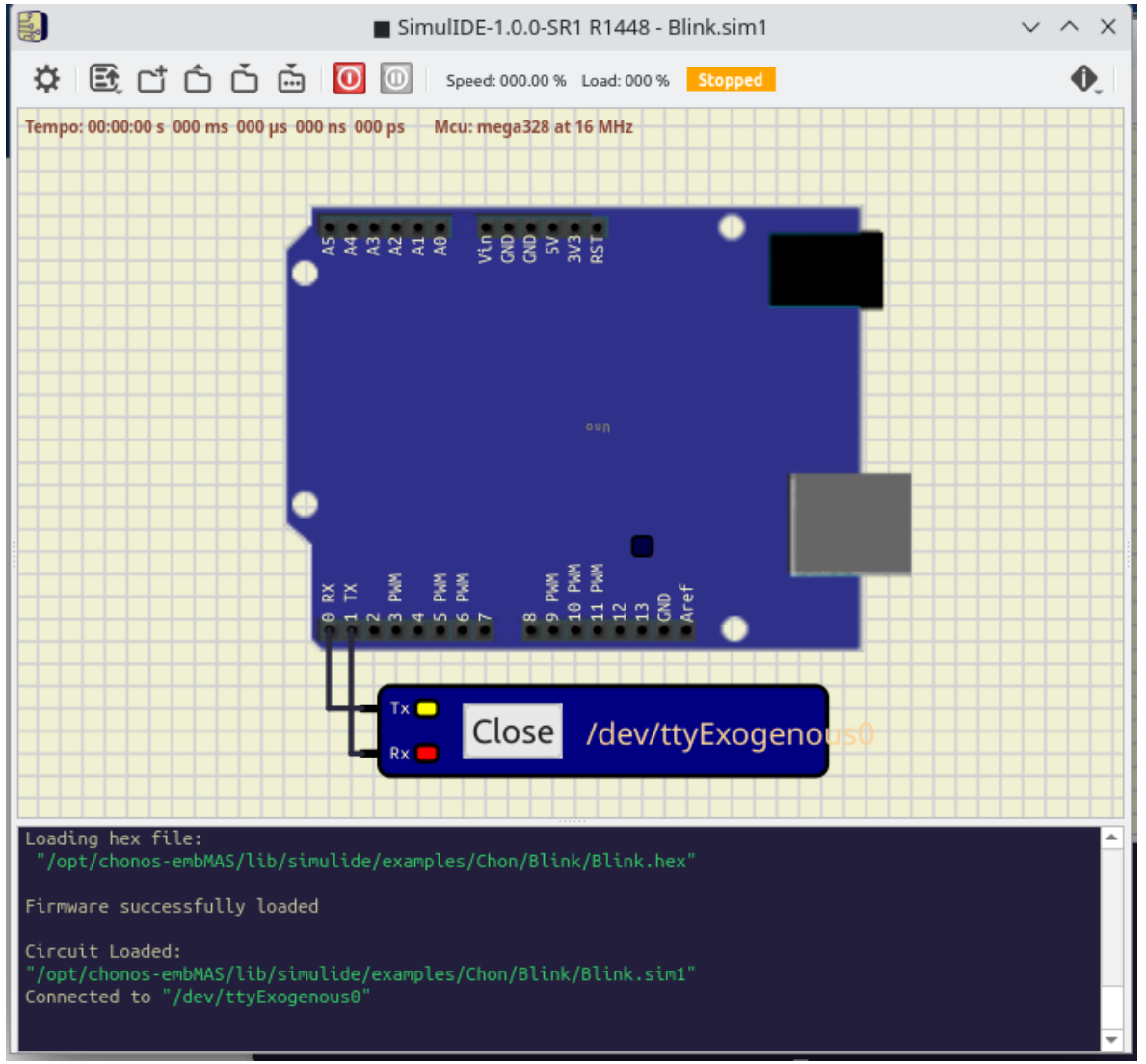
SimulIDE: Blink



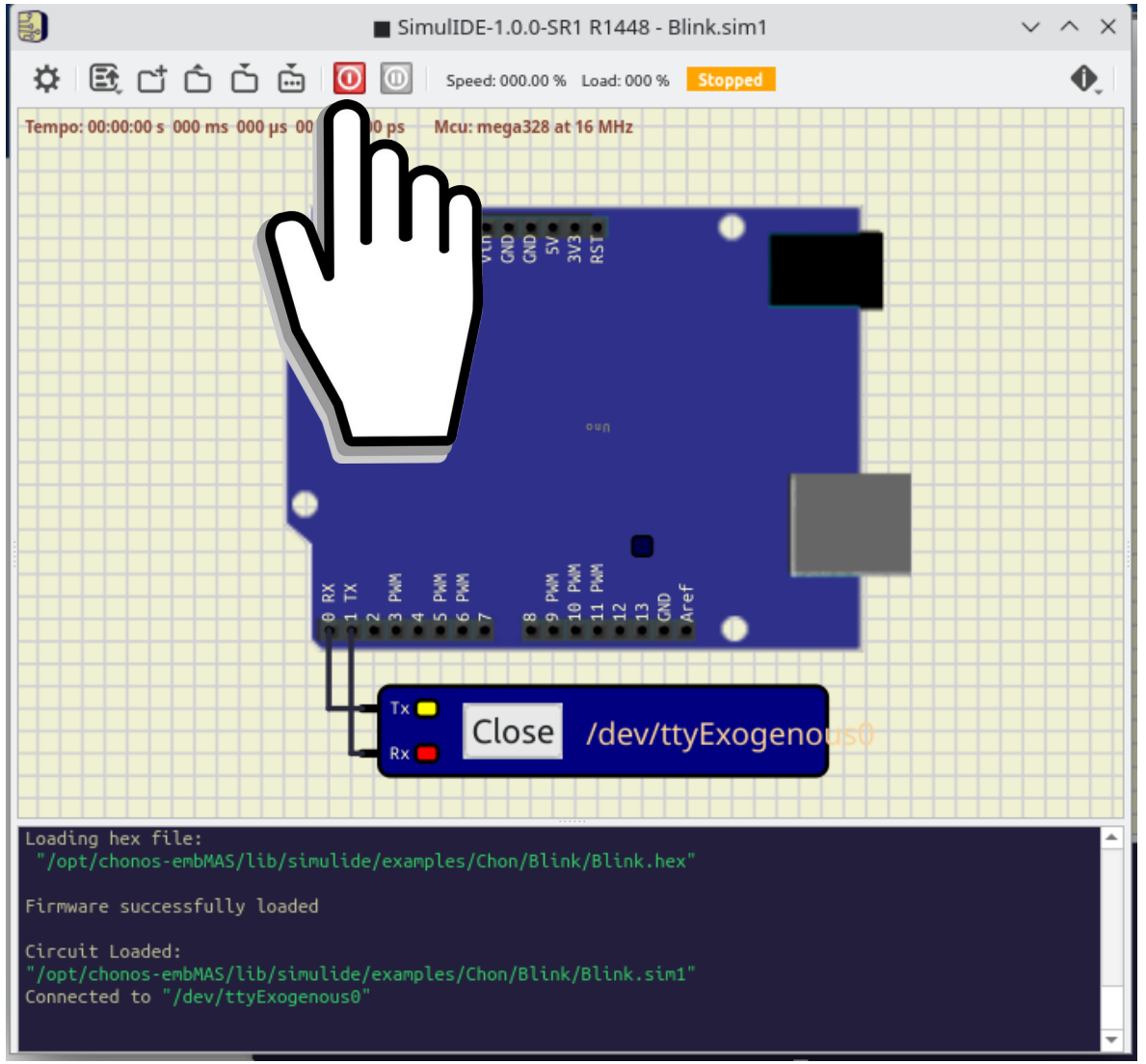
SimulIDE: Blink



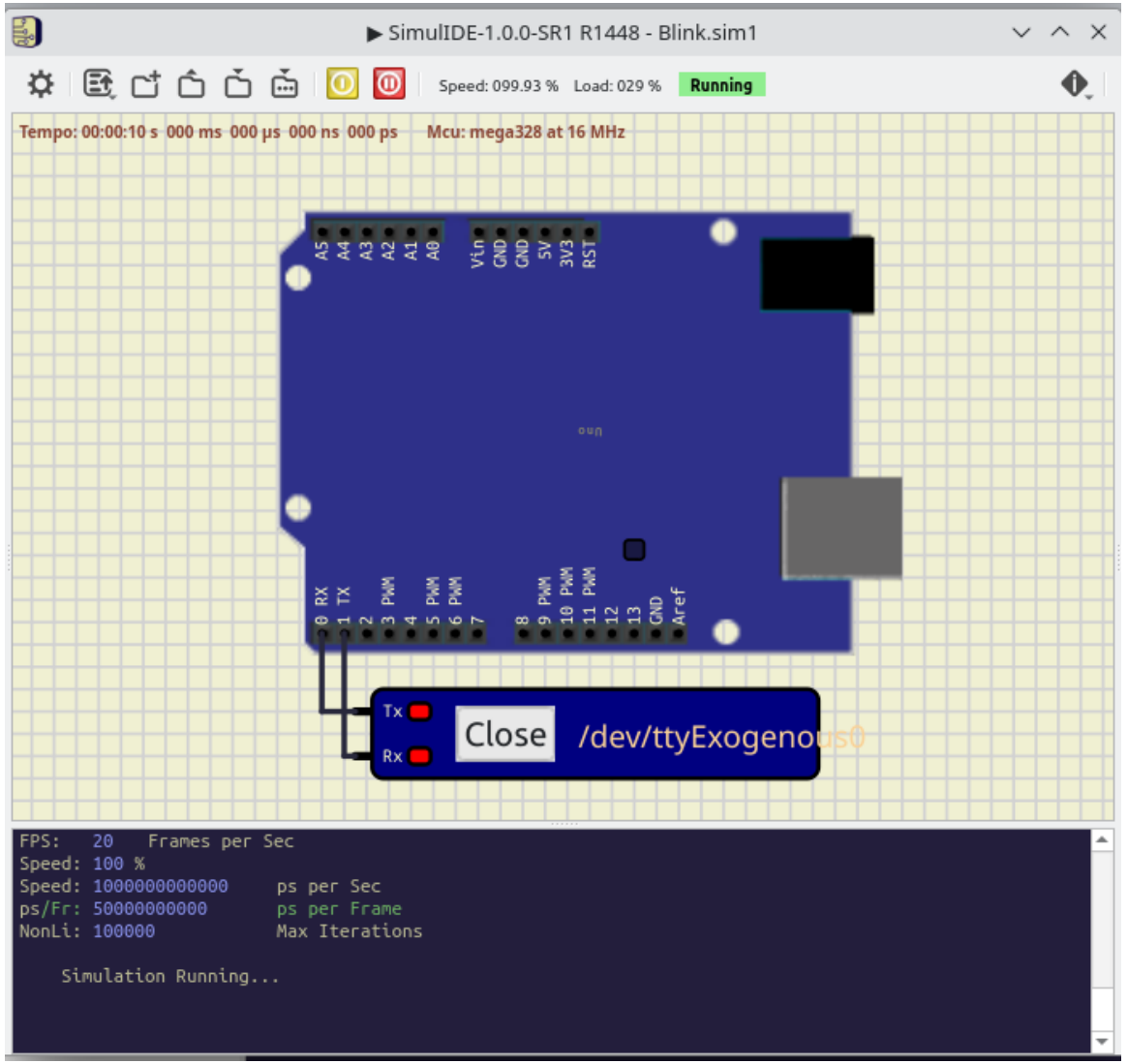
SimulIDE: Blink



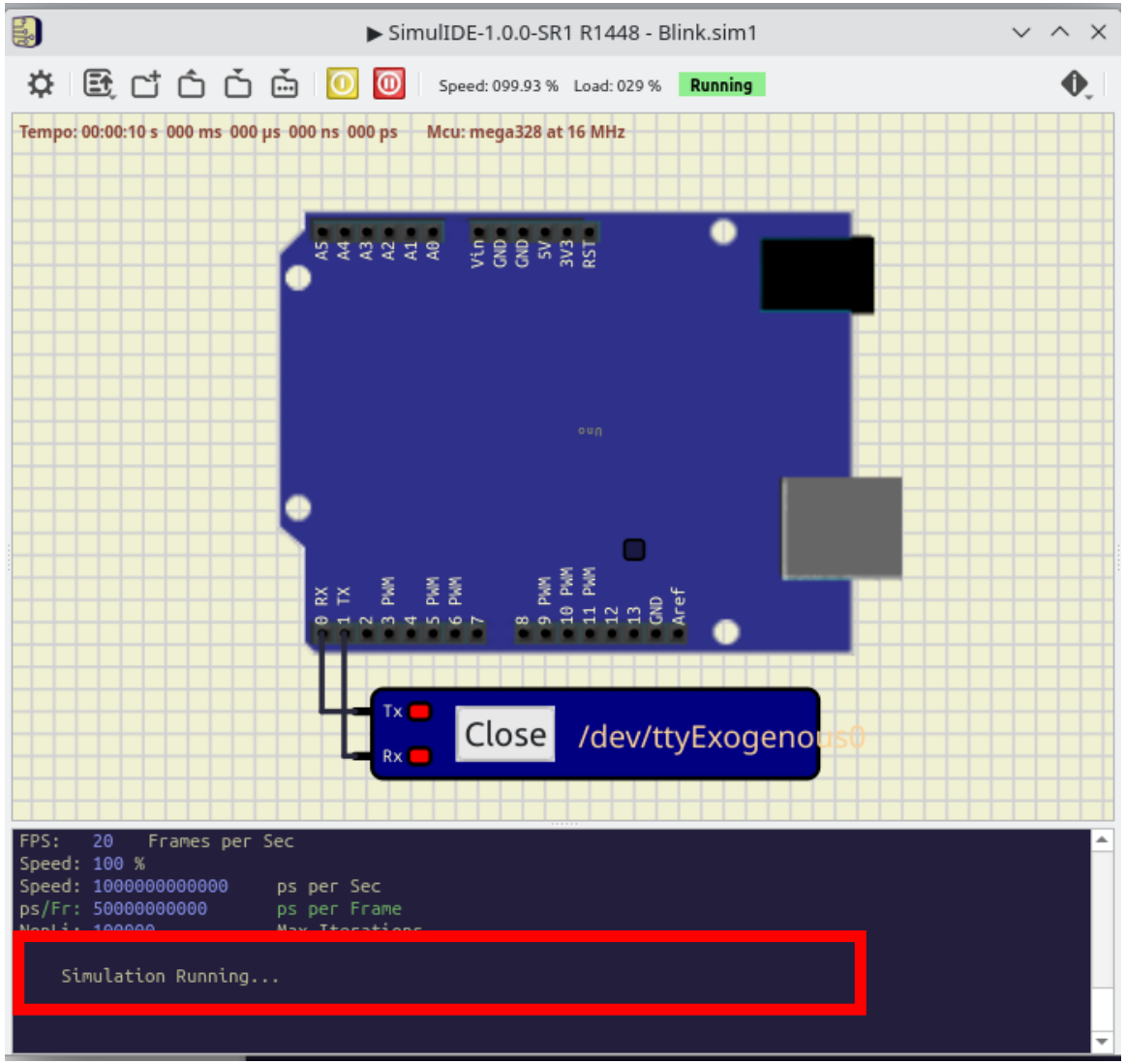
SimulIDE: Blink



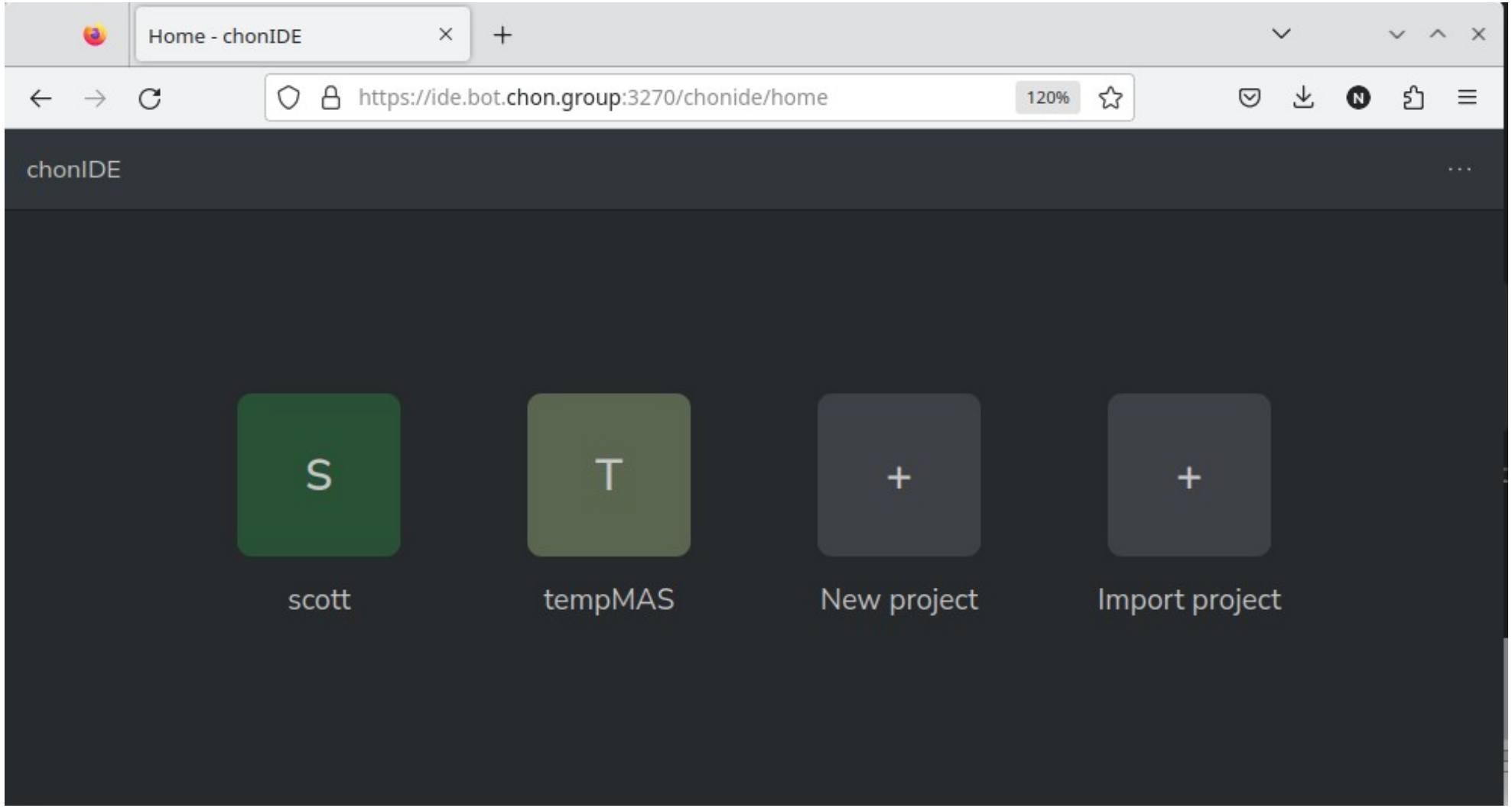
SimulIDE: Blink



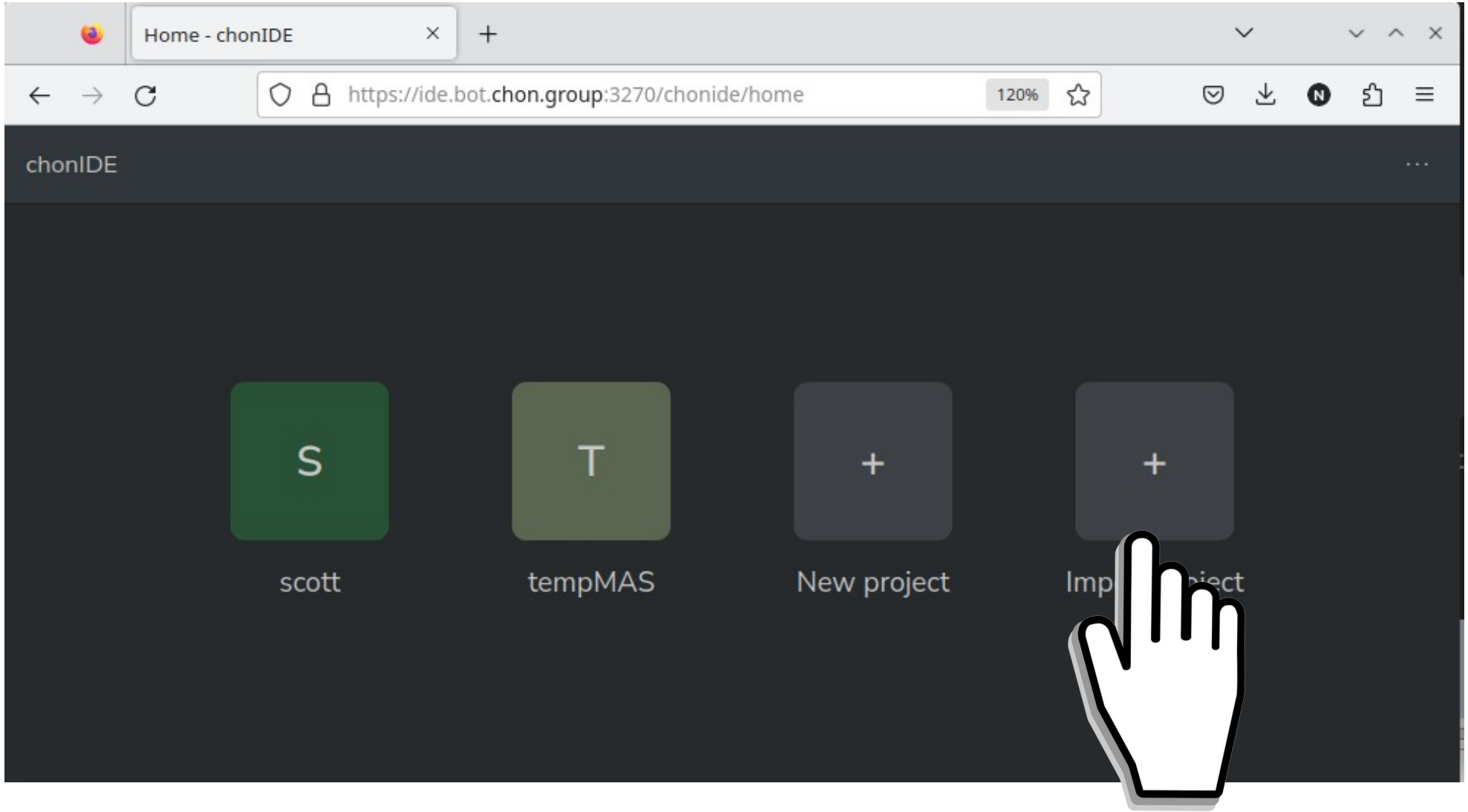
SimulIDE: Blink



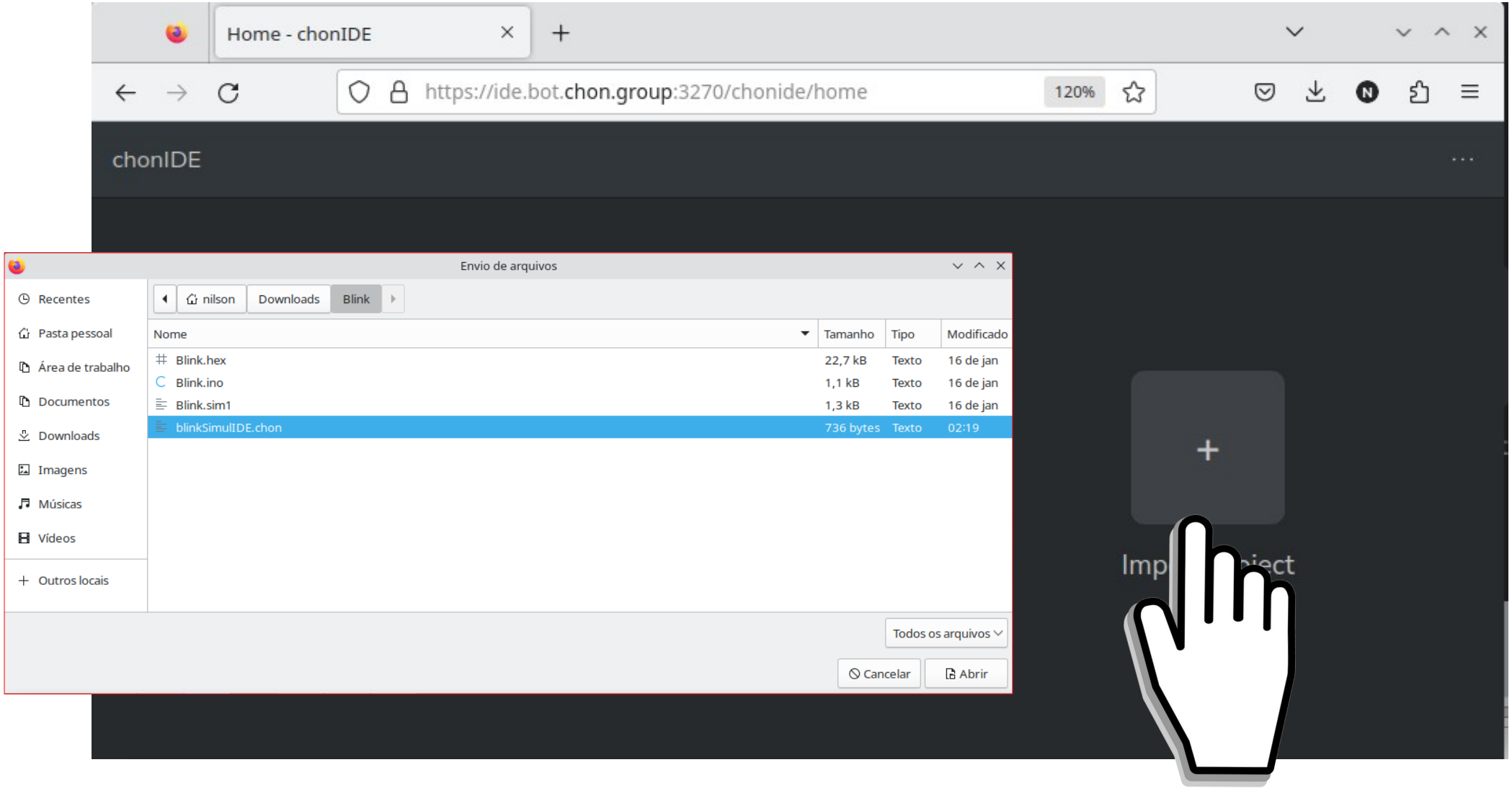
SimulIDE: Blink



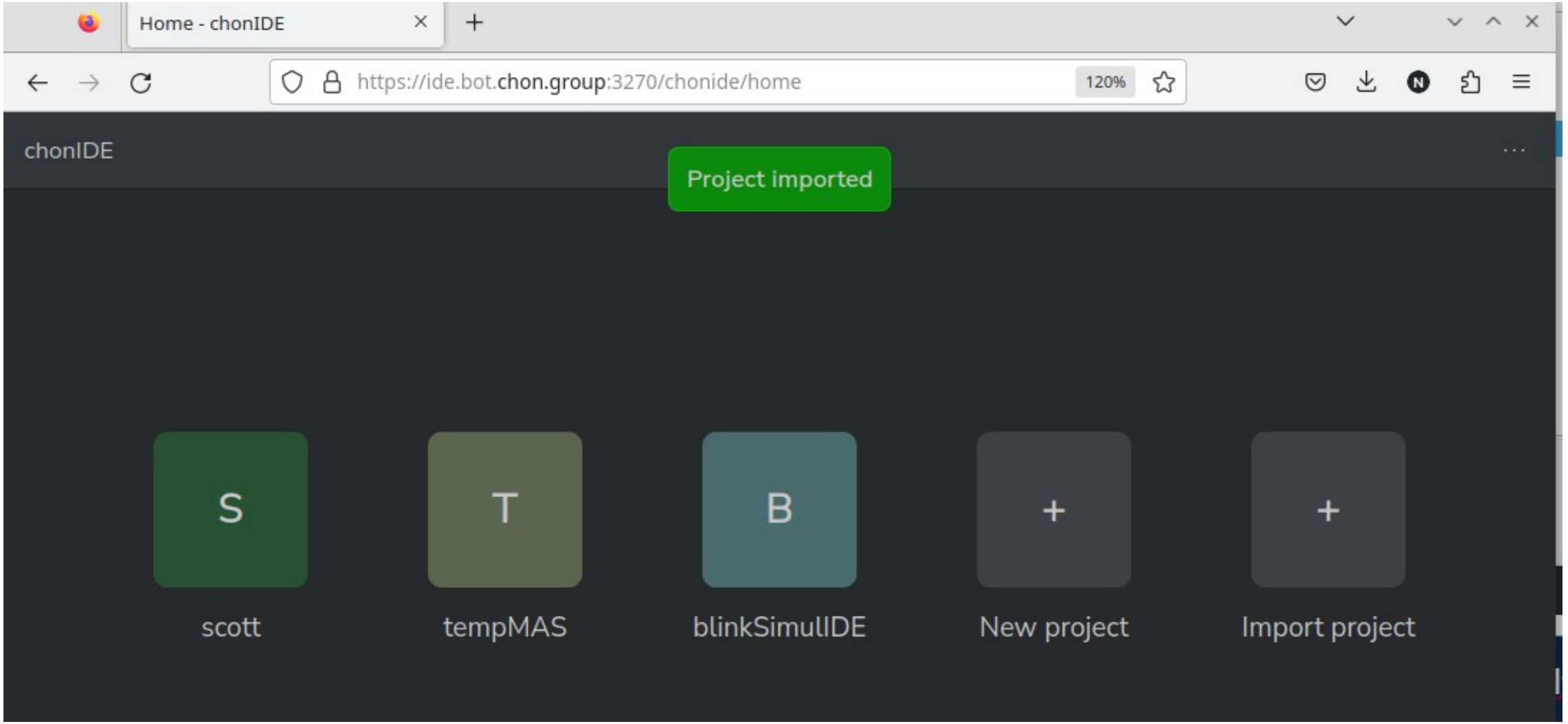
SimulIDE: Blink



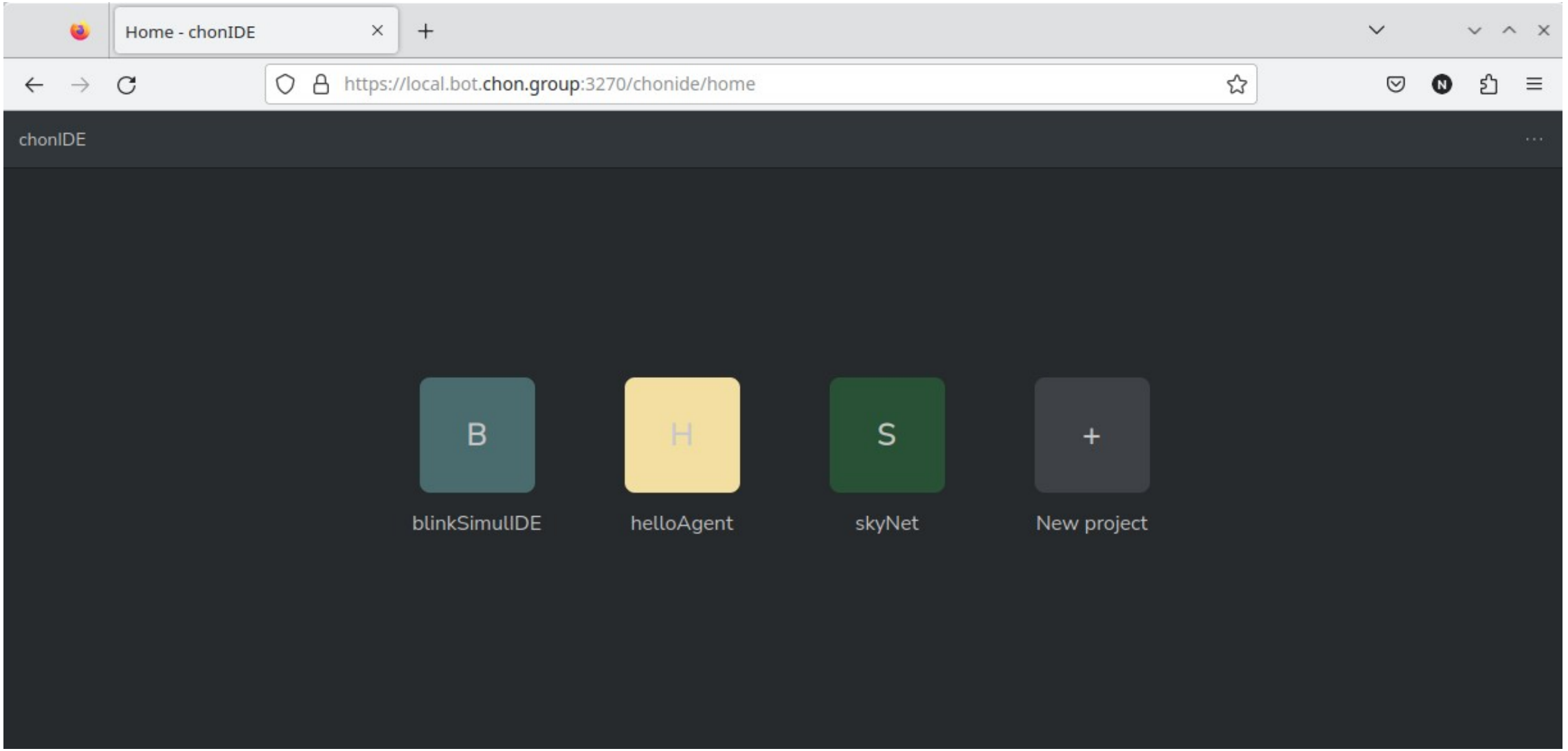
SimulIDE: Blink



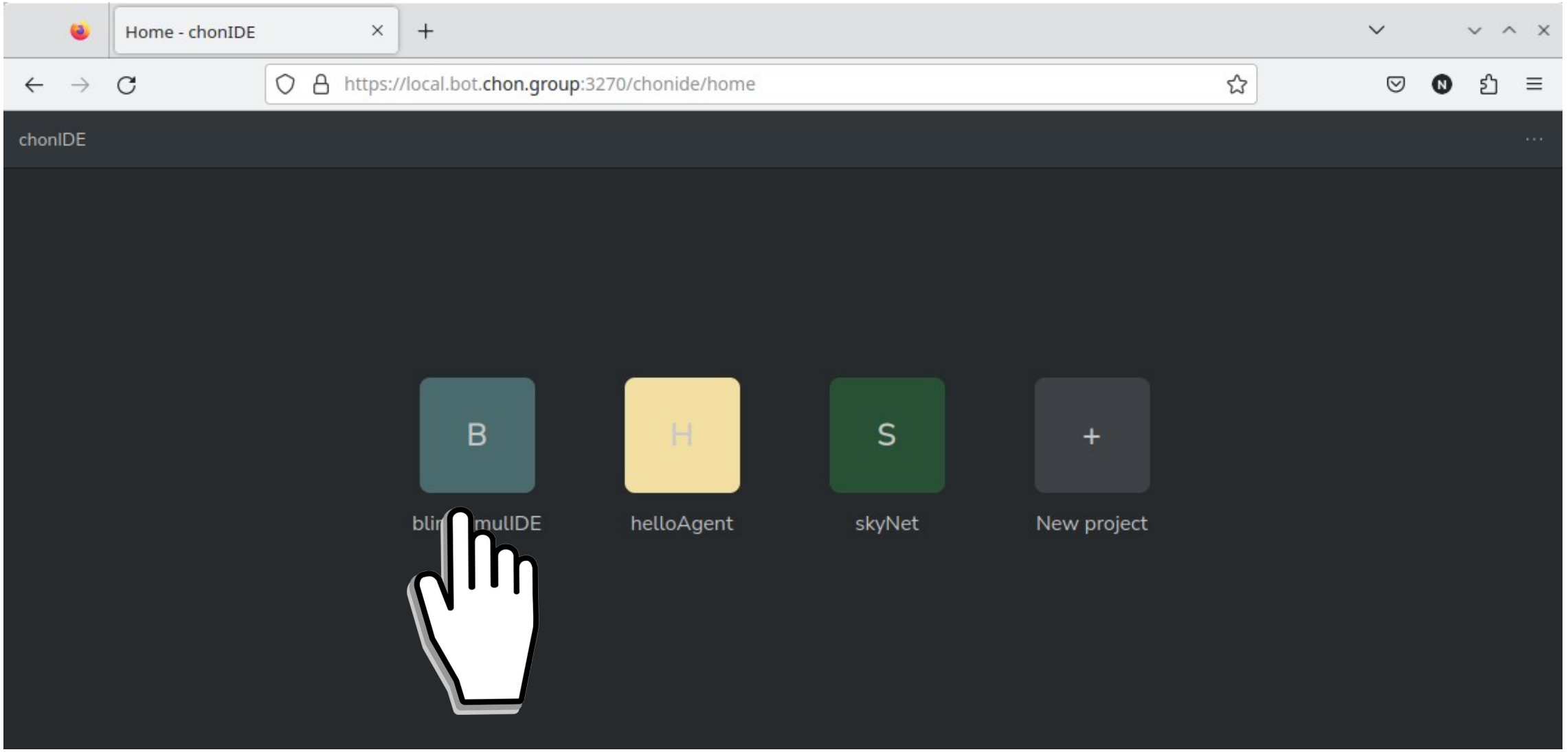
SimulIDE: Blink



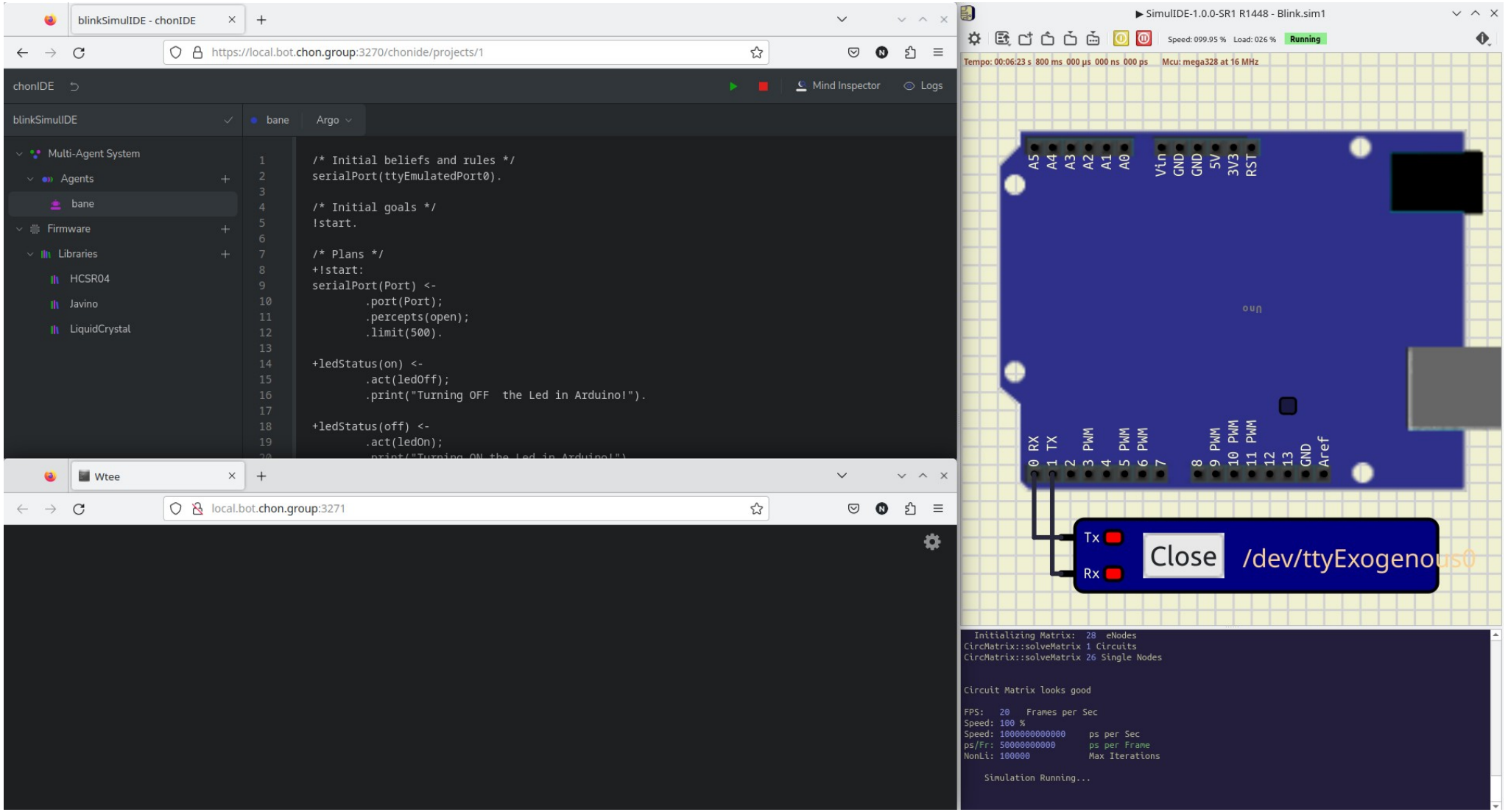
SimulIDE: Blink



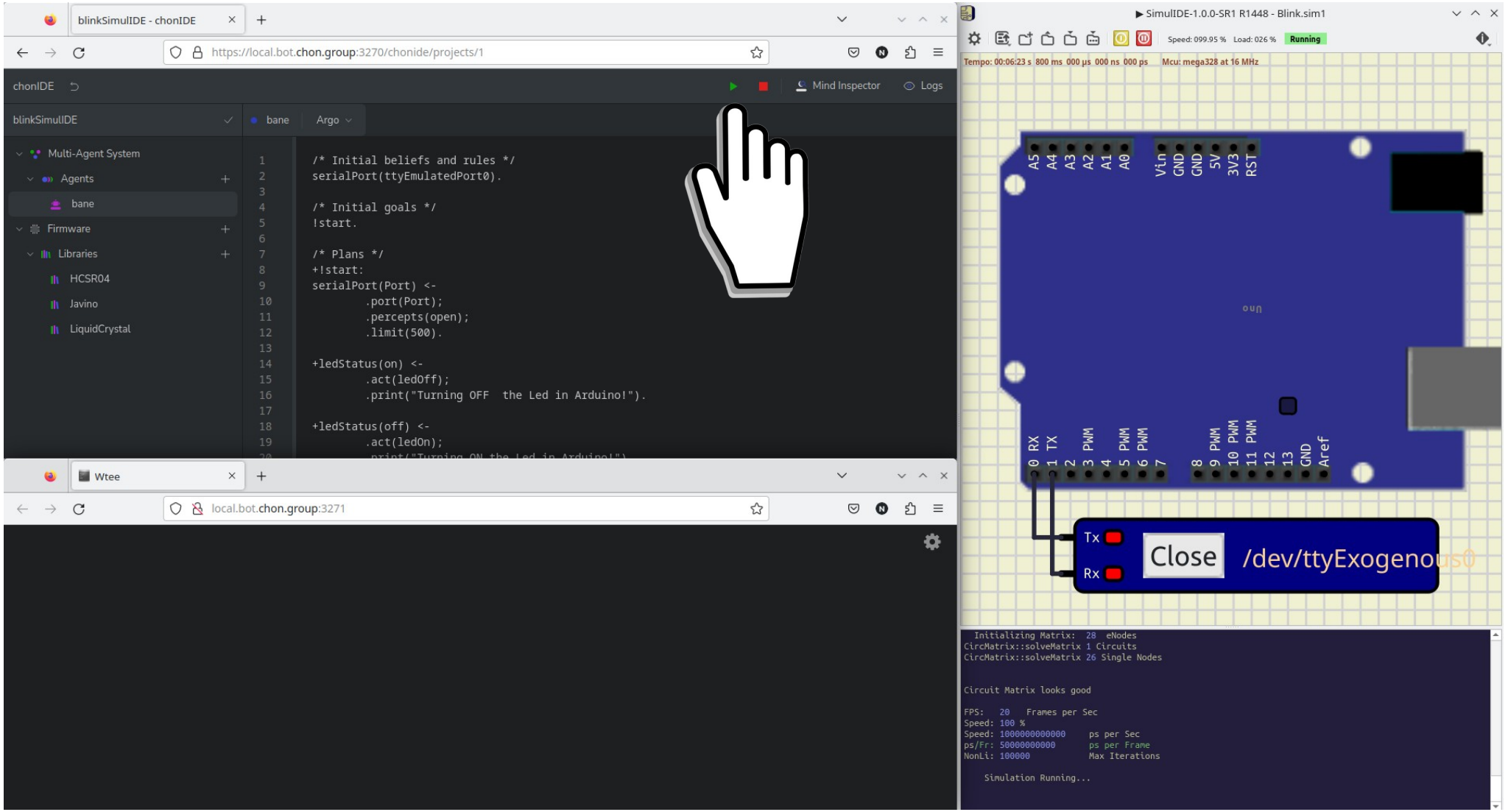
SimulIDE: Blink



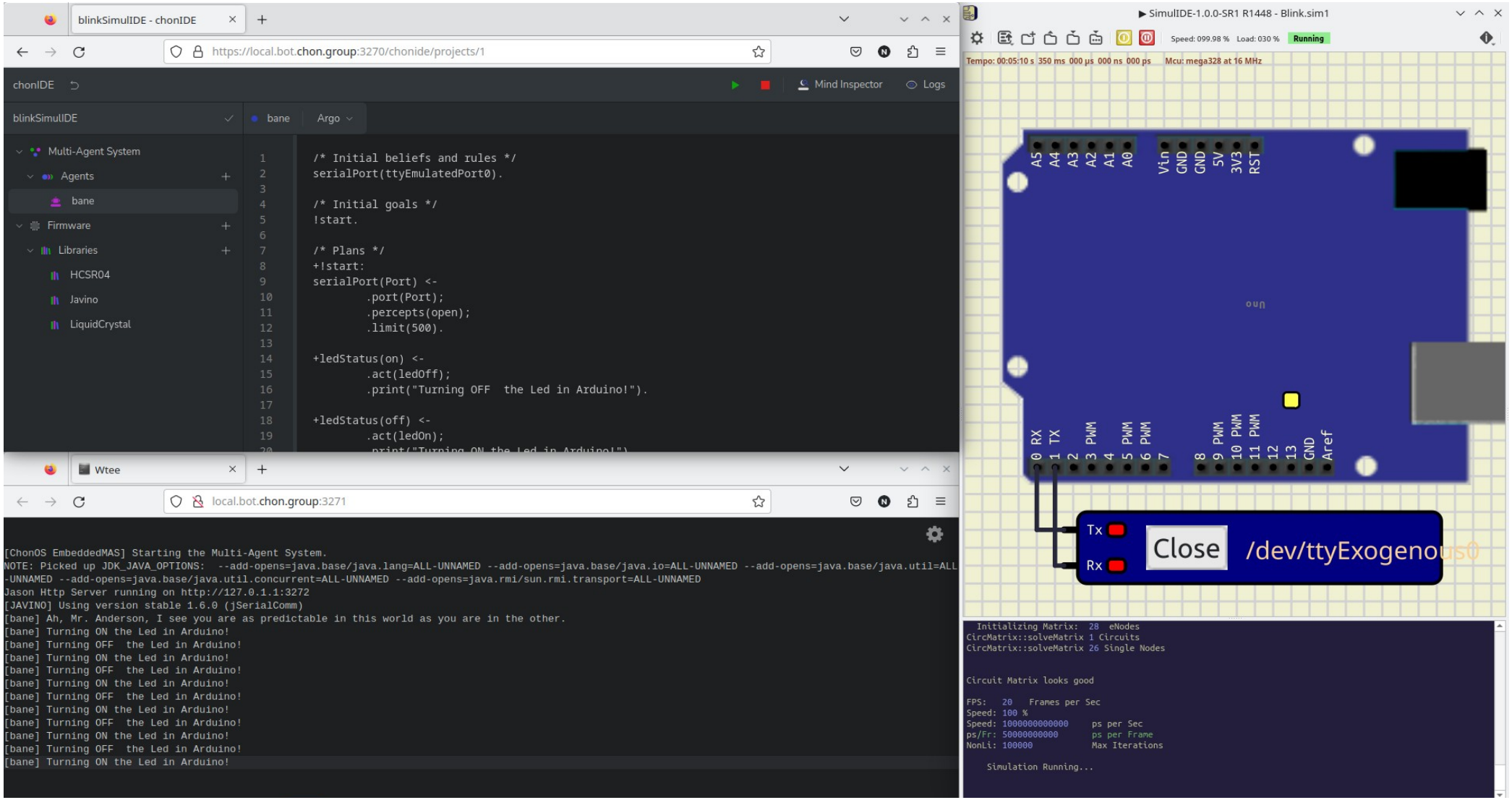
SimulIDE: Blink



SimulIDE: Blink



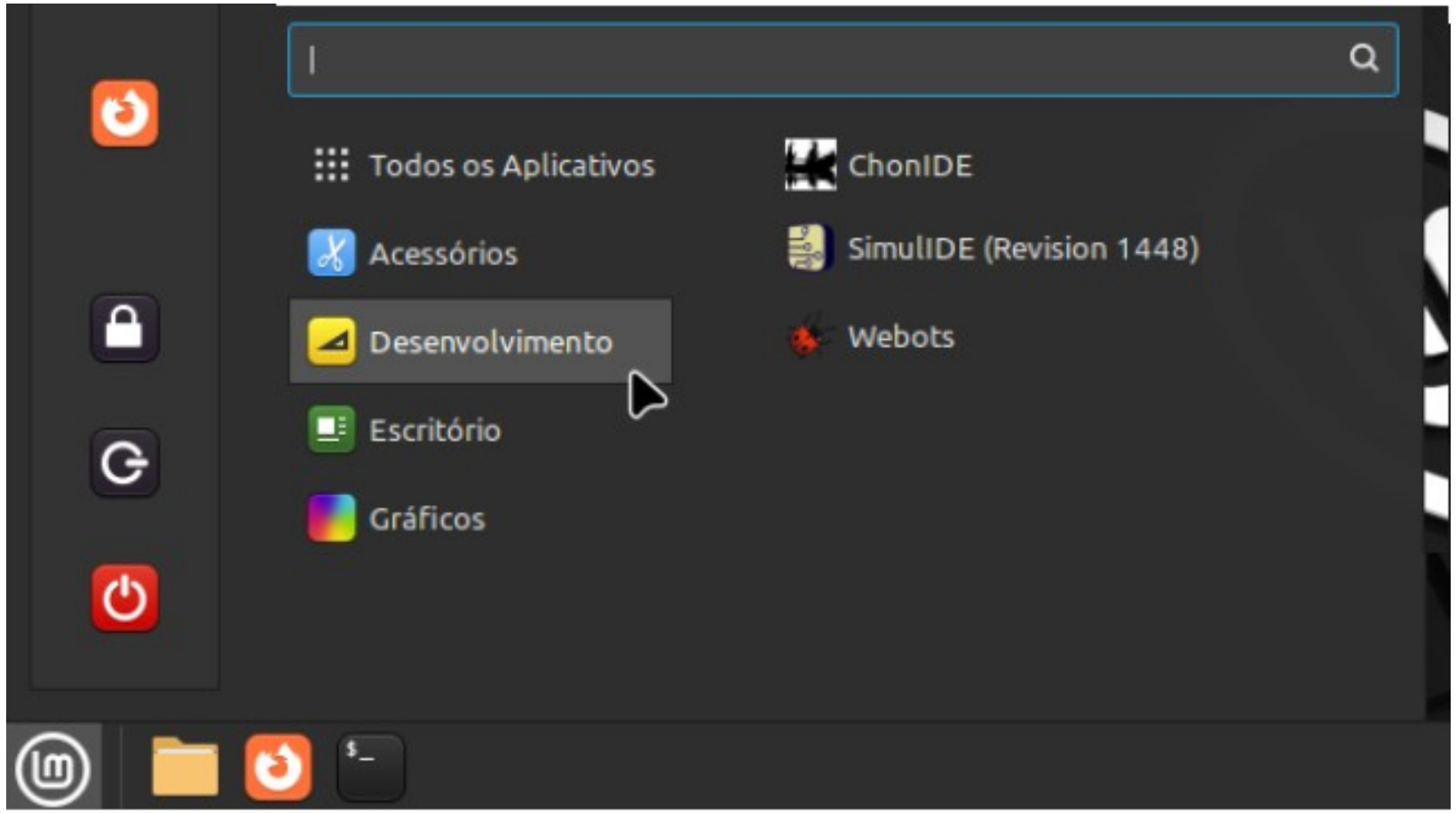
SimulIDE: Blink



Webots

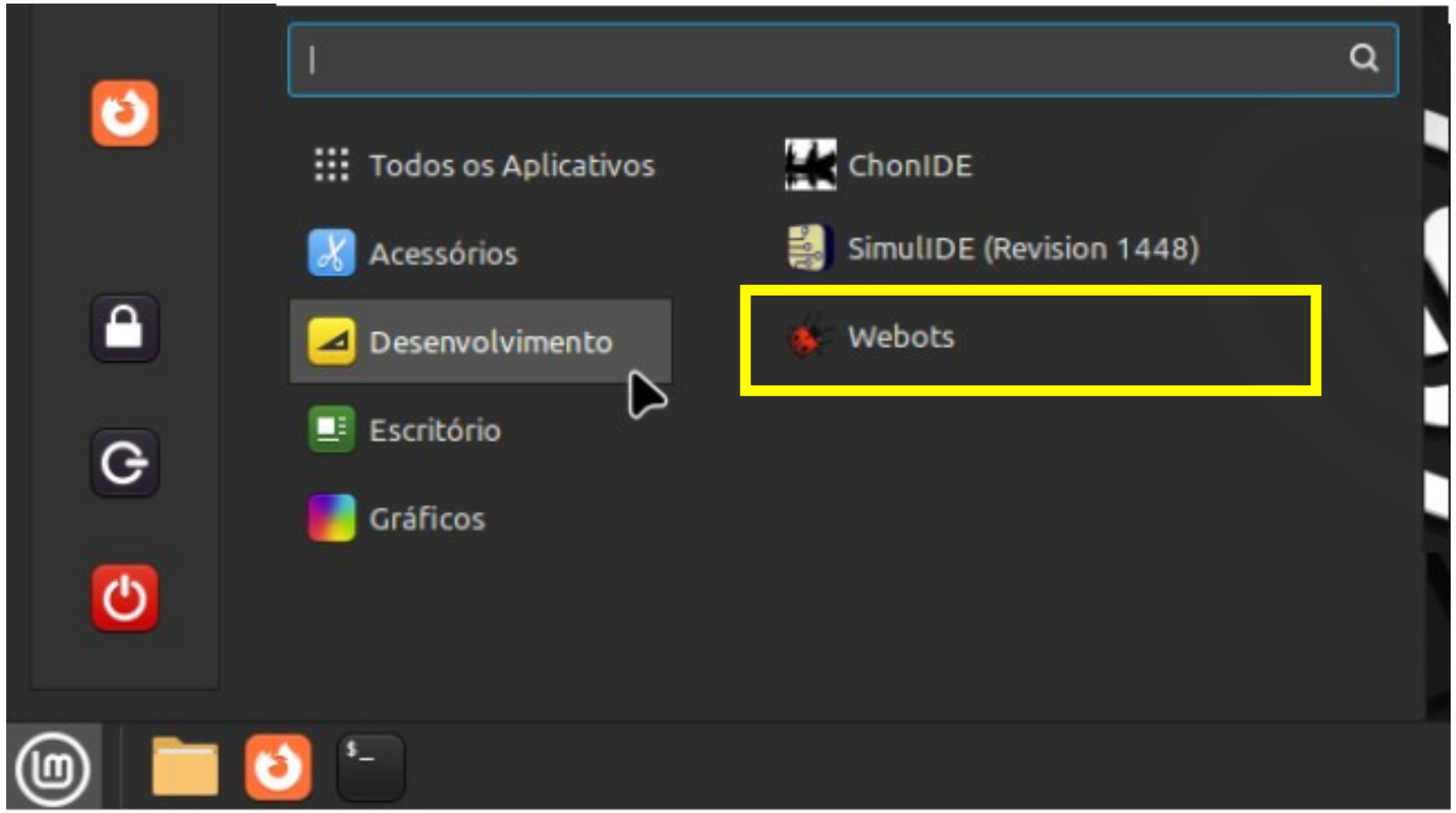
Michel, O. (1998). Webots: Symbiosis Between Virtual and Real Mobile Robots. In: Heudin, JC. (eds) Virtual Worlds. VW 1998. Lecture Notes in Computer Science(), vol 1434. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-68686-X_24

Webots



Manual de instalação
https://cyberbotics.com/doc/guide/installation-procedure#installing-the-debian-package-with-the-advanced-packaging-tool-apt

Webots



Manual de instalação
https://cyberbotics.com/doc/guide/installation-procedure#installing-the-debian-package-with-the-advanced-packaging-tool-apt

Webots

distributedAndEmbeddedAI / course / 05-TheDevelopmentTool / Examples /

nilsonLazarin Car4WD Simulated World developed by @bptfreitas

Name	Last commit r
..	
Blink	developmer
Car4WD	Car4WD Sim
Blink.zip	developmer
Car4WD.zip	Car4WD Sim

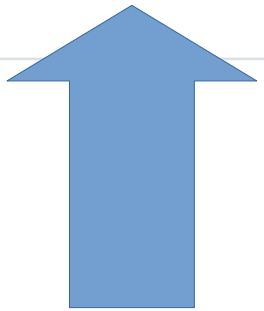
https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/05-TheDevelopmentTool/Examples/Car4WD.zip

Webots

distributedAndEmbeddedAI / course / 05-TheDevelopmentTool / Examples /

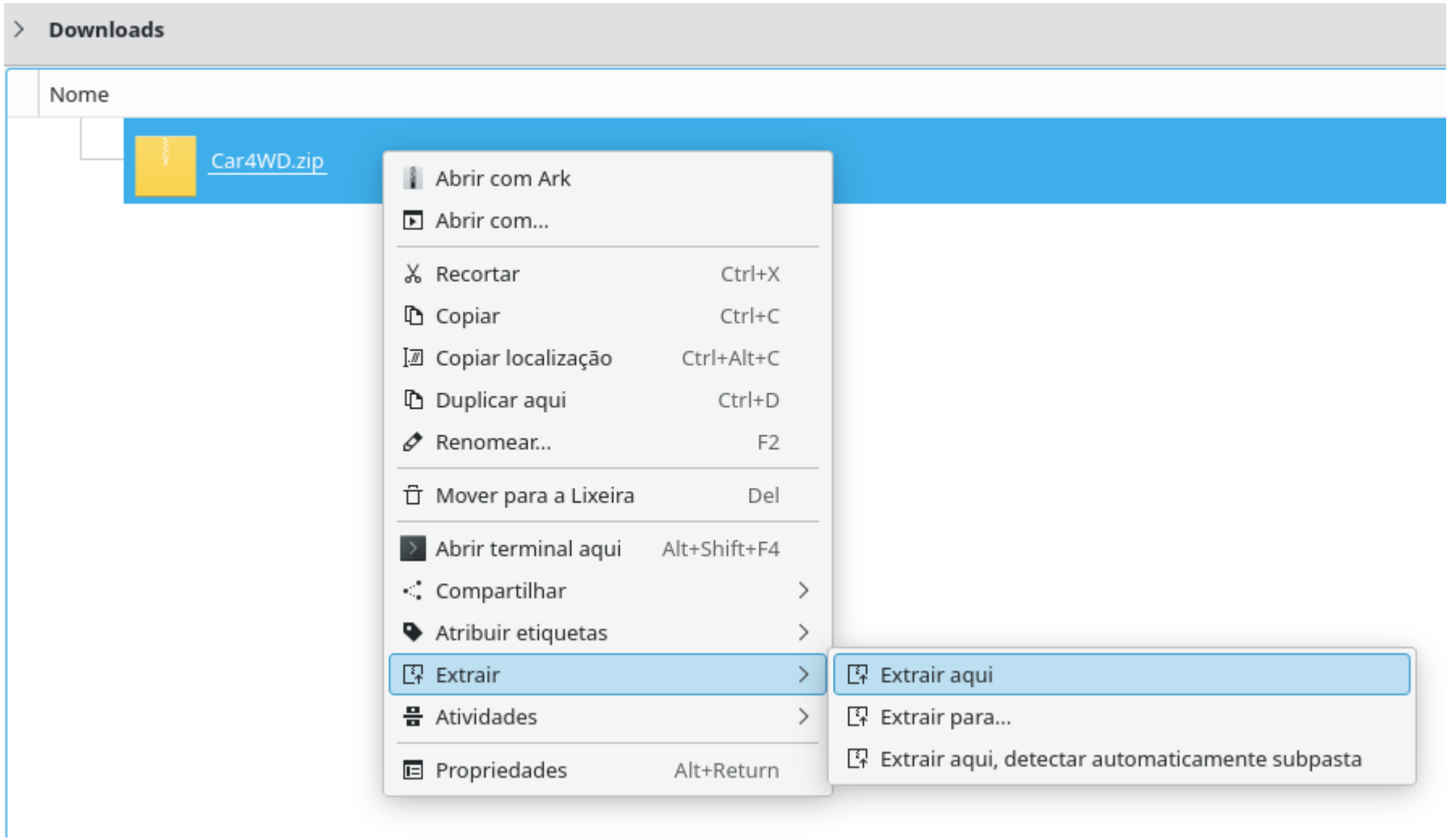
nilsonLazarin Car4WD Simulated World developed by @bptfreitas

Name	Last commit
..	
Blink	developmer
Car4WD	Car4WD Sim
Blink.zip	developmer
Car4WD.zip	Car4WD Sim

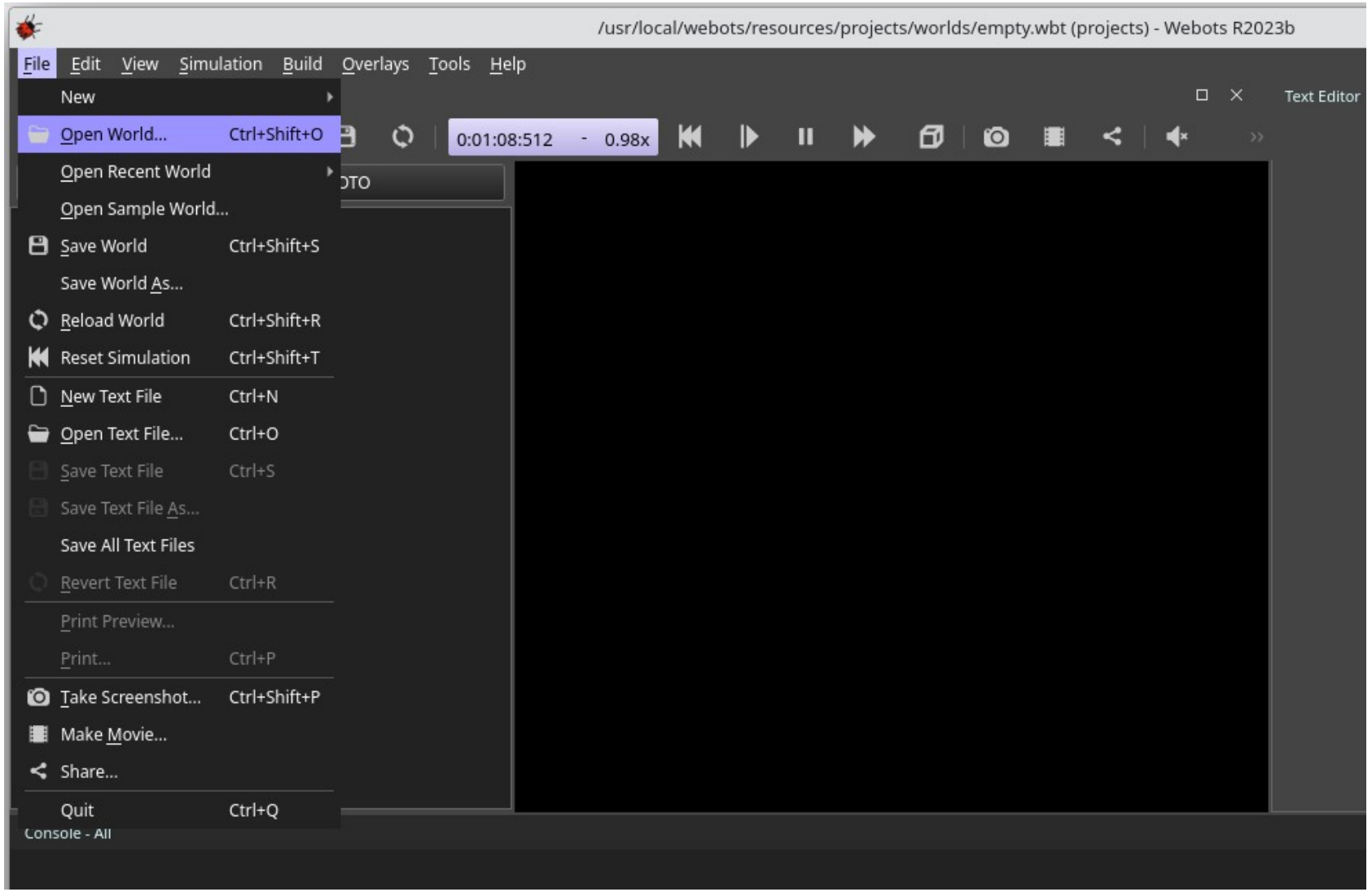


https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/05-TheDevelopmentTool/Examples/Car4WD.zip

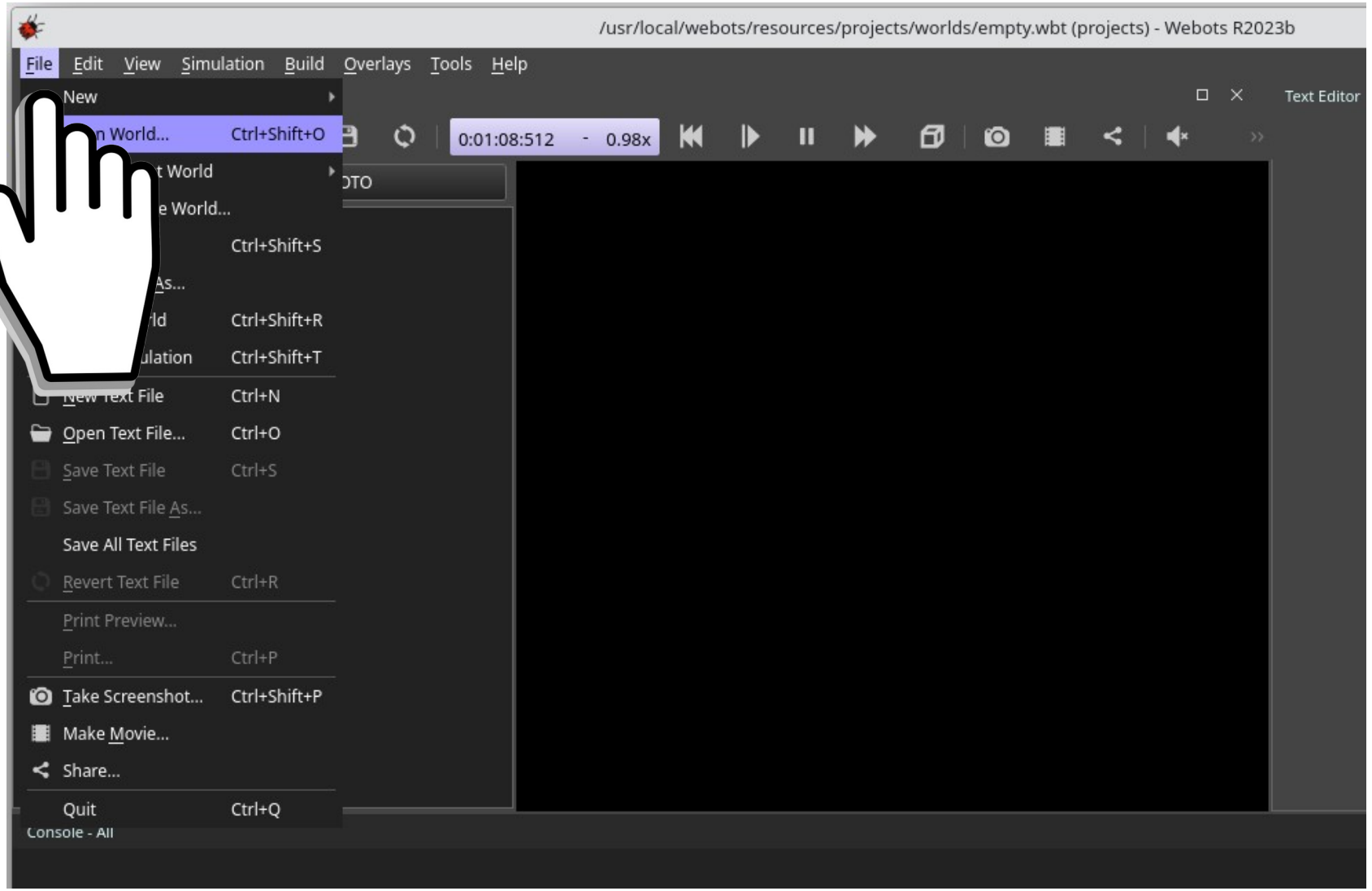
Webots


Downloads
Nome
Car4WD.zip
Abrir com Ark
Abrir com...
Recortar Ctrl+X
Copiar Ctrl+C
Copiar localização Ctrl+Alt+C
Duplicar aqui Ctrl+D
Renomear... F2
Mover para a Lixeira Del
Abrir terminal aqui Alt+Shift+F4
Compartilhar
Atribuir etiquetas
Extrair
Atividades
Propriedades Alt+Return
Extrair aqui
Extrair para...
Extrair aqui, detectar automaticamente subpasta

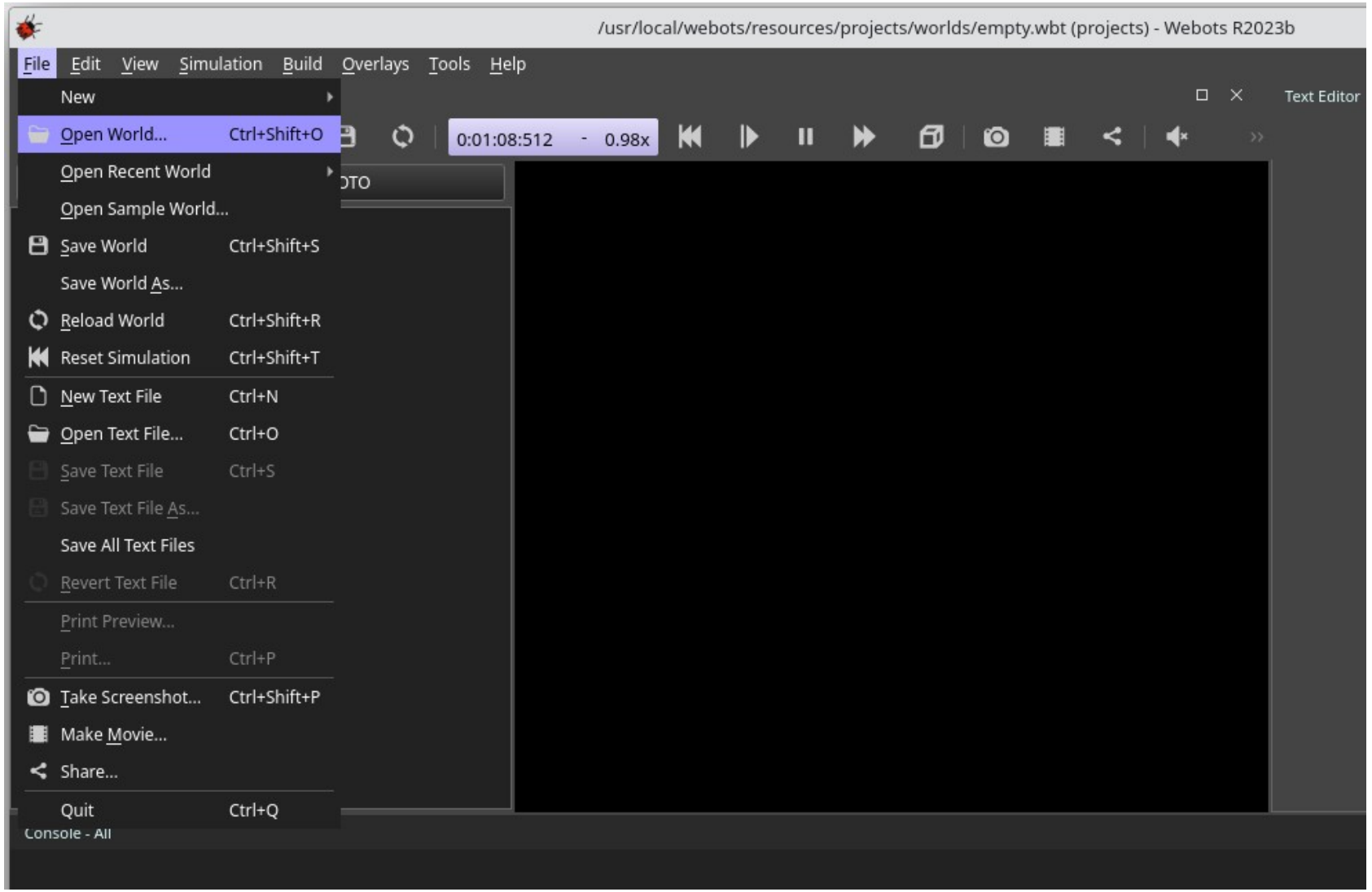
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Open World... Ctrl+Shift+O
Open Recent World
Open Sample World...
Save World Ctrl+Shift+S
Save World As...
Reload World Ctrl+Shift+R
Reset Simulation Ctrl+Shift+T
New Text File Ctrl+N
Open Text File... Ctrl+O
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
0:01:08:512 - 0.98x
Text Editor
Console - All

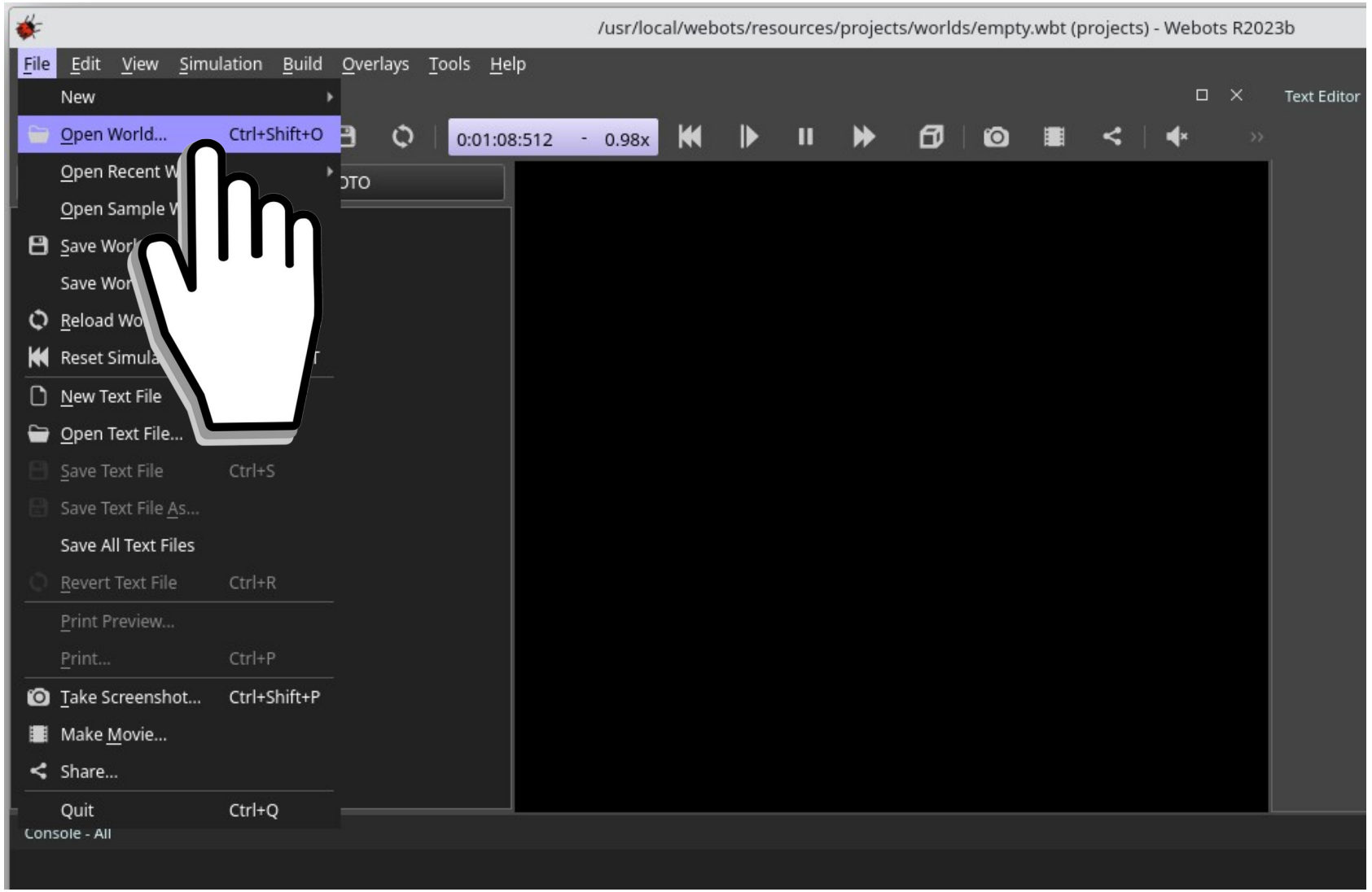
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Ctrl+Shift+O
Ctrl+Shift+S
Ctrl+Shift+R
Ctrl+Shift+T
Ctrl+N
Open Text File... Ctrl+O
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
0:01:08:512 - 0.98x
Text Editor
Console - All

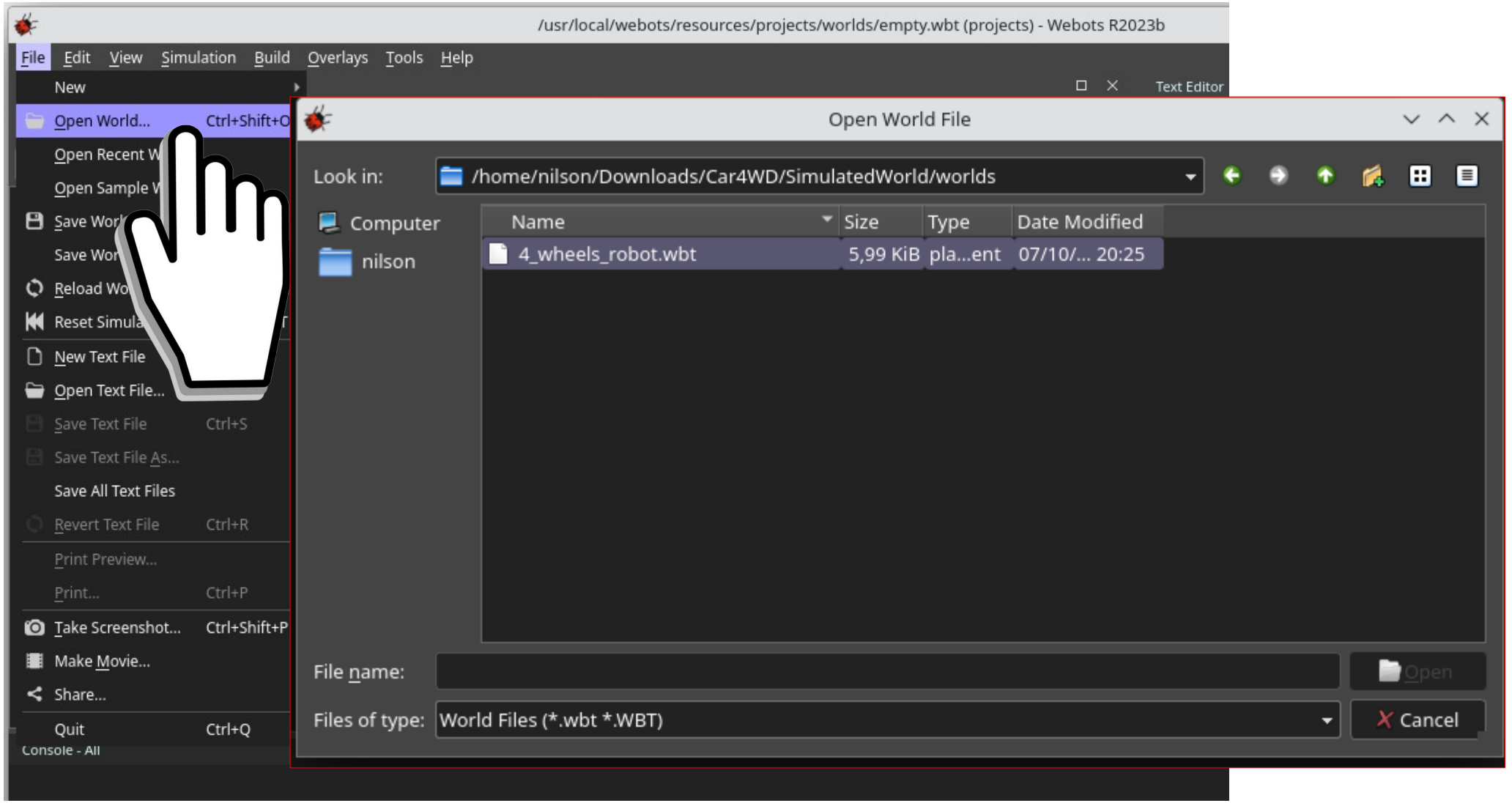
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Open World... Ctrl+Shift+O
Open Recent World
Open Sample World...
Save World Ctrl+Shift+S
Save World As...
Reload World Ctrl+Shift+R
Reset Simulation Ctrl+Shift+T
New Text File Ctrl+N
Open Text File... Ctrl+O
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
0:01:08:512 - 0.98x
Text Editor
Console - All

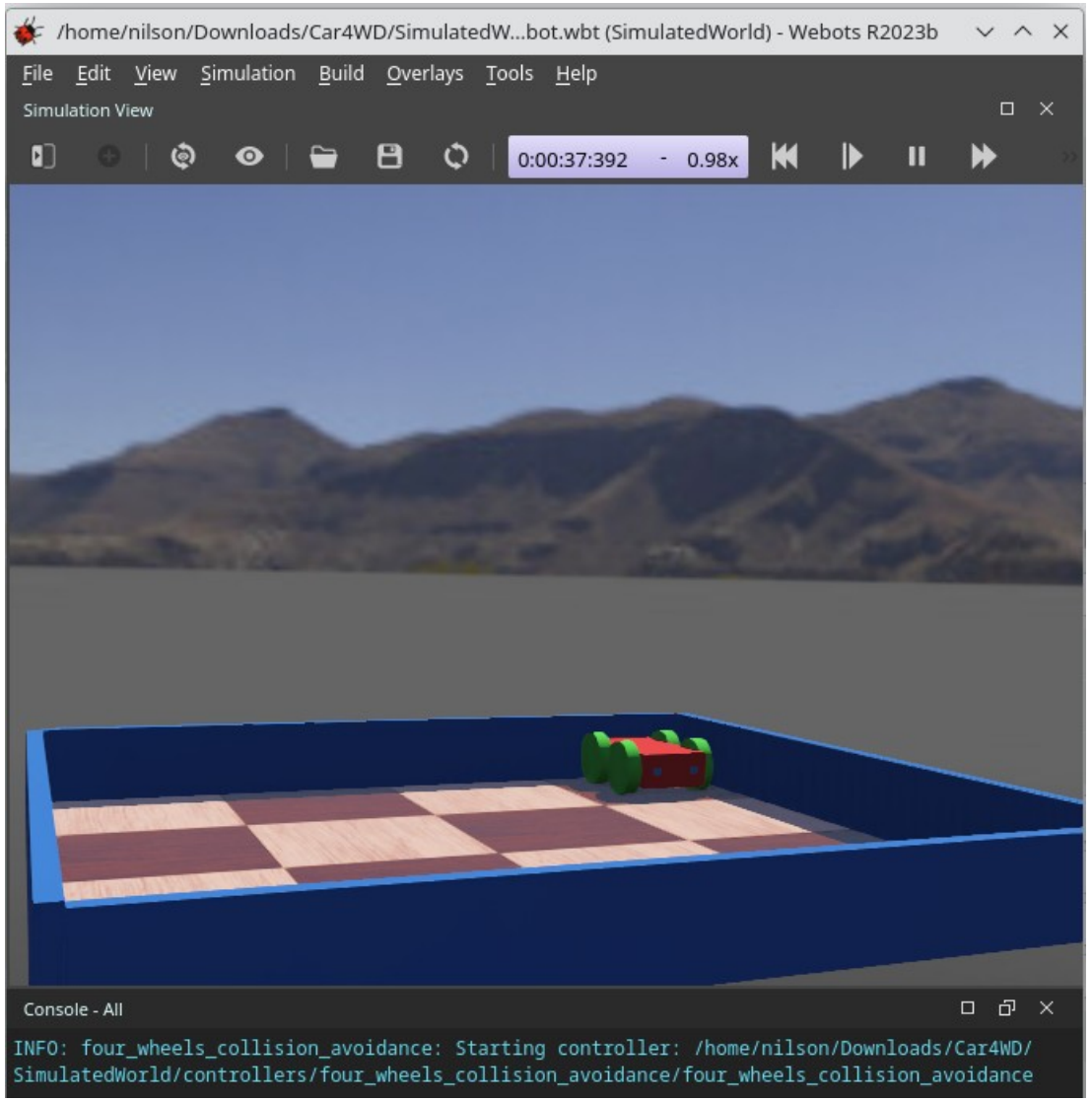
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Open World... Ctrl+Shift+O
Open Recent W
Open Sample W
Save Wor
Save Wor
Reload Wo
Reset Simula
New Text File
Open Text File...
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
0:01:08:512 - 0.98x
Text Editor
Console - All

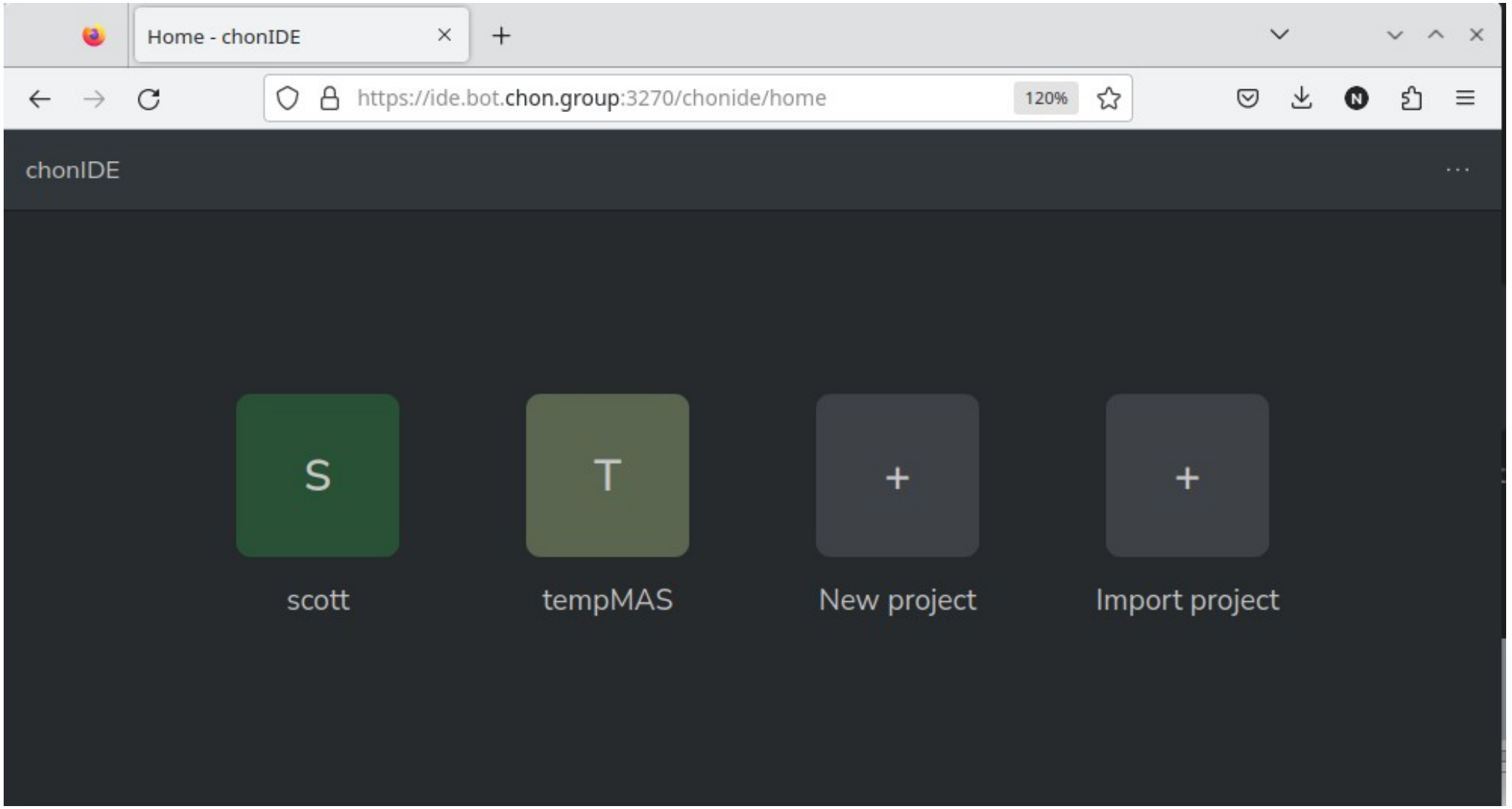
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Open World... Ctrl+Shift+O
Open Recent W
Open Sample W
Save Wor
Save Wor
Reload Wo
Reset Simula
New Text File
Open Text File...
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
Console - All
Text Editor
Open World File
Look in: /home/nilson/Downloads/Car4WD/SimulatedWorld/worlds
Computer
nilson
Name Size Type Date Modified
4_wheels_robot.wbt 5,99 KiB pla...ent 07/10/... 20:25
File name:
Open
Files of type: World Files (*.wbt *.WBT)
Cancel

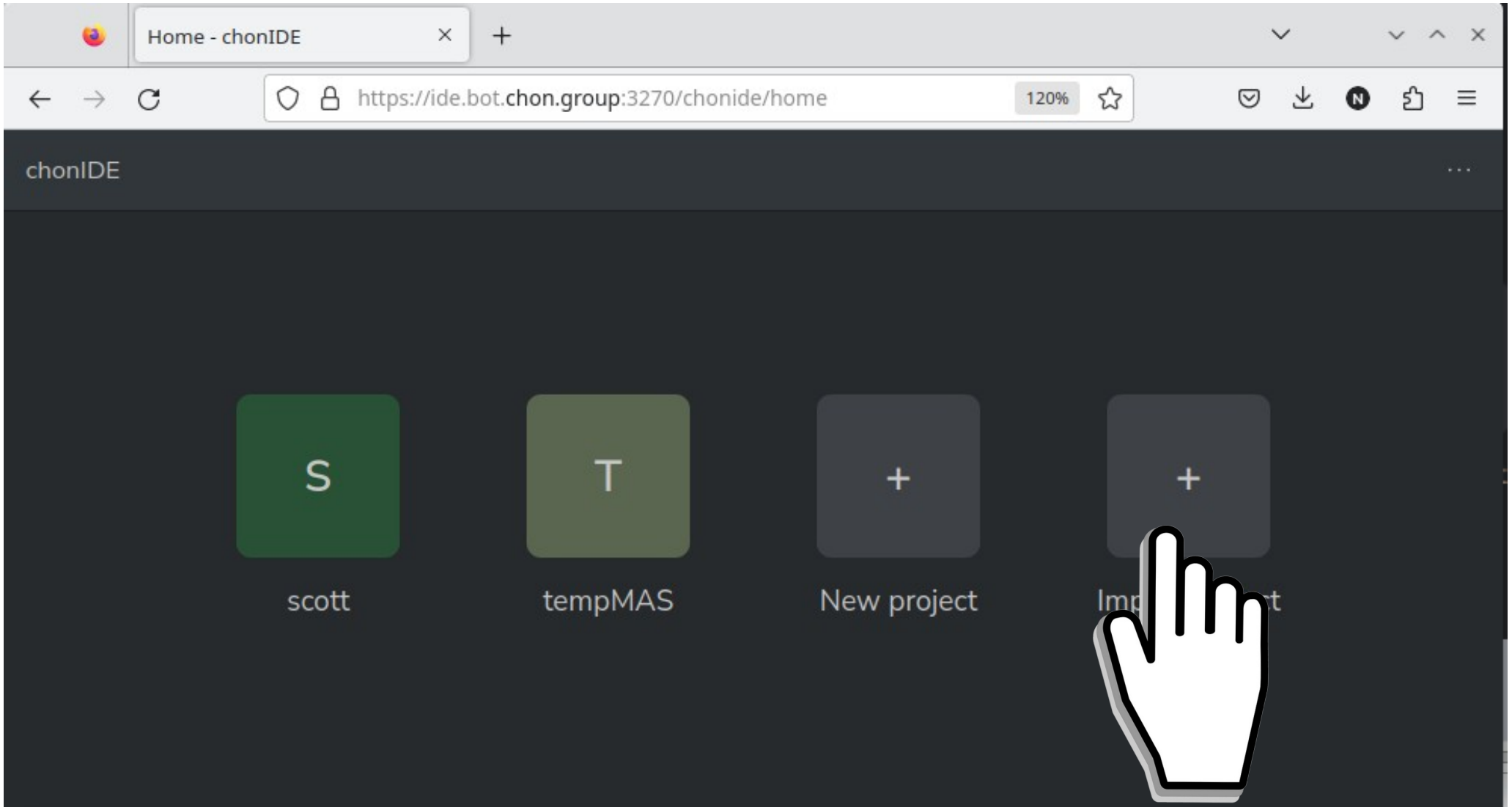
Webots


/home/nilson/Downloads/Car4WD/SimulatedW...bot.wbt (SimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:00:37:392 - 0.98x
Console - All
INFO: four_wheels_collision_avoidance: Starting controller: /home/nilson/Downloads/Car4WD/SimulatedWorld/controllers/four_wheels_collision_avoidance/four_wheels_collision_avoidance

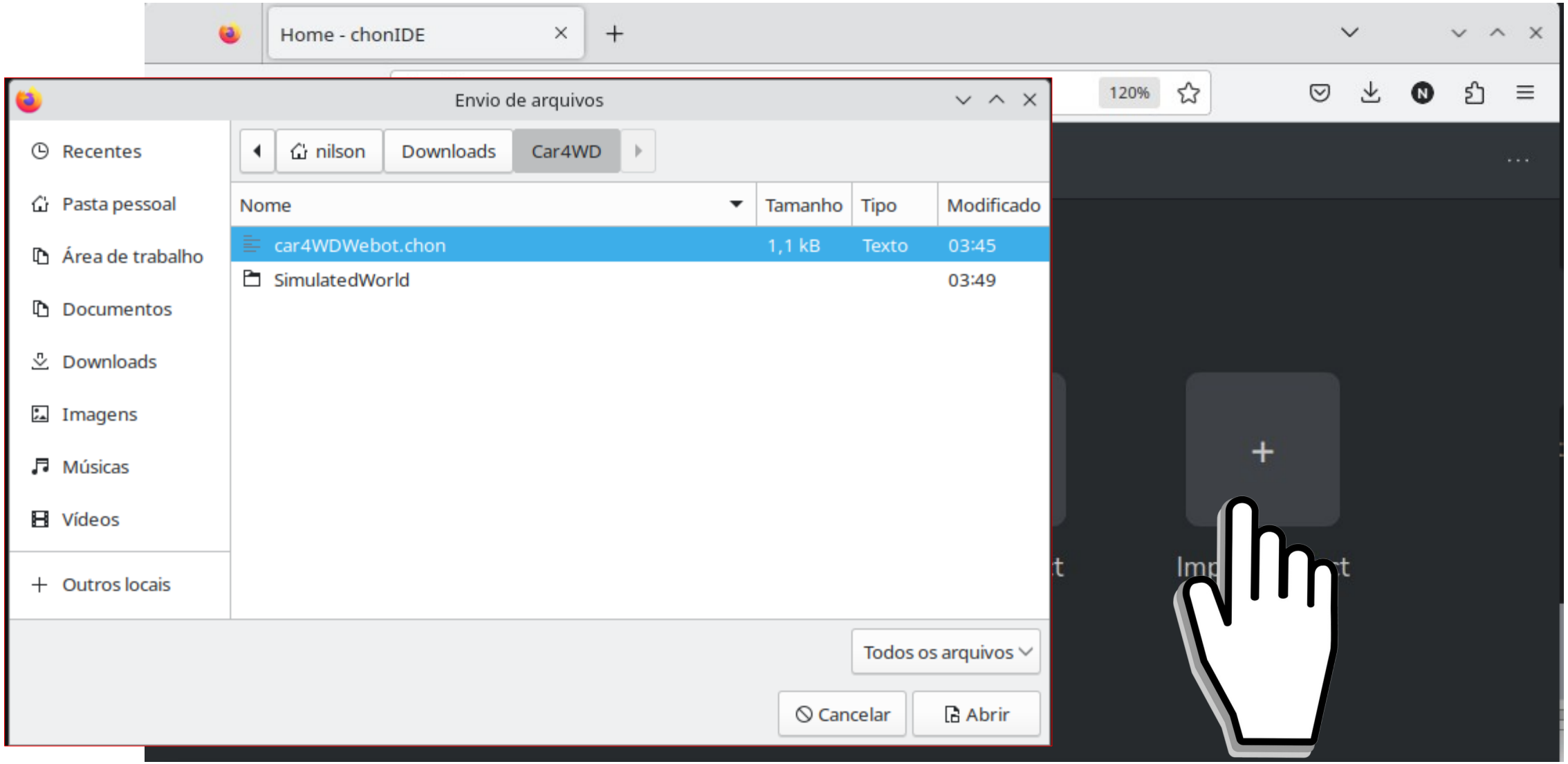
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
T
+
+
scott
tempMAS
New project
Import project

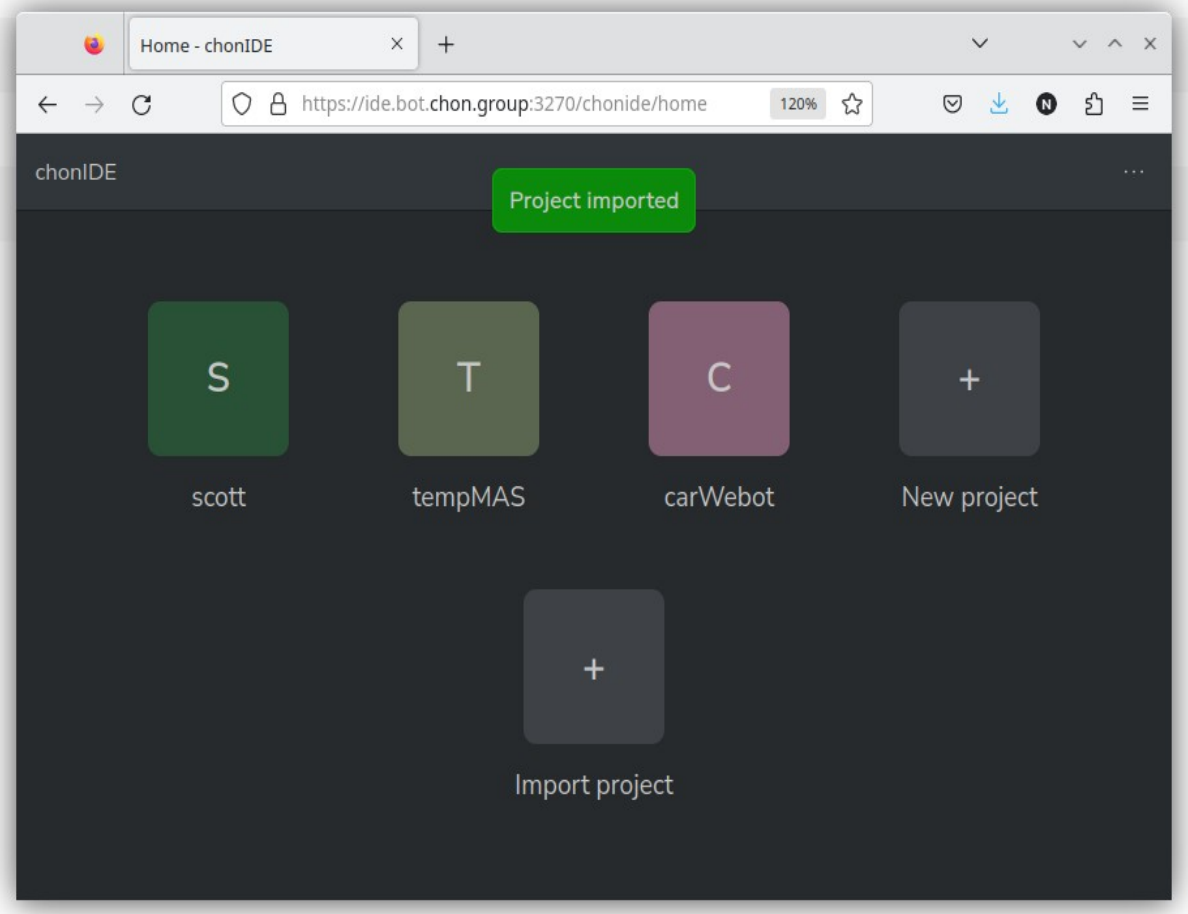
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
T
+
+
scott
tempMAS
New project

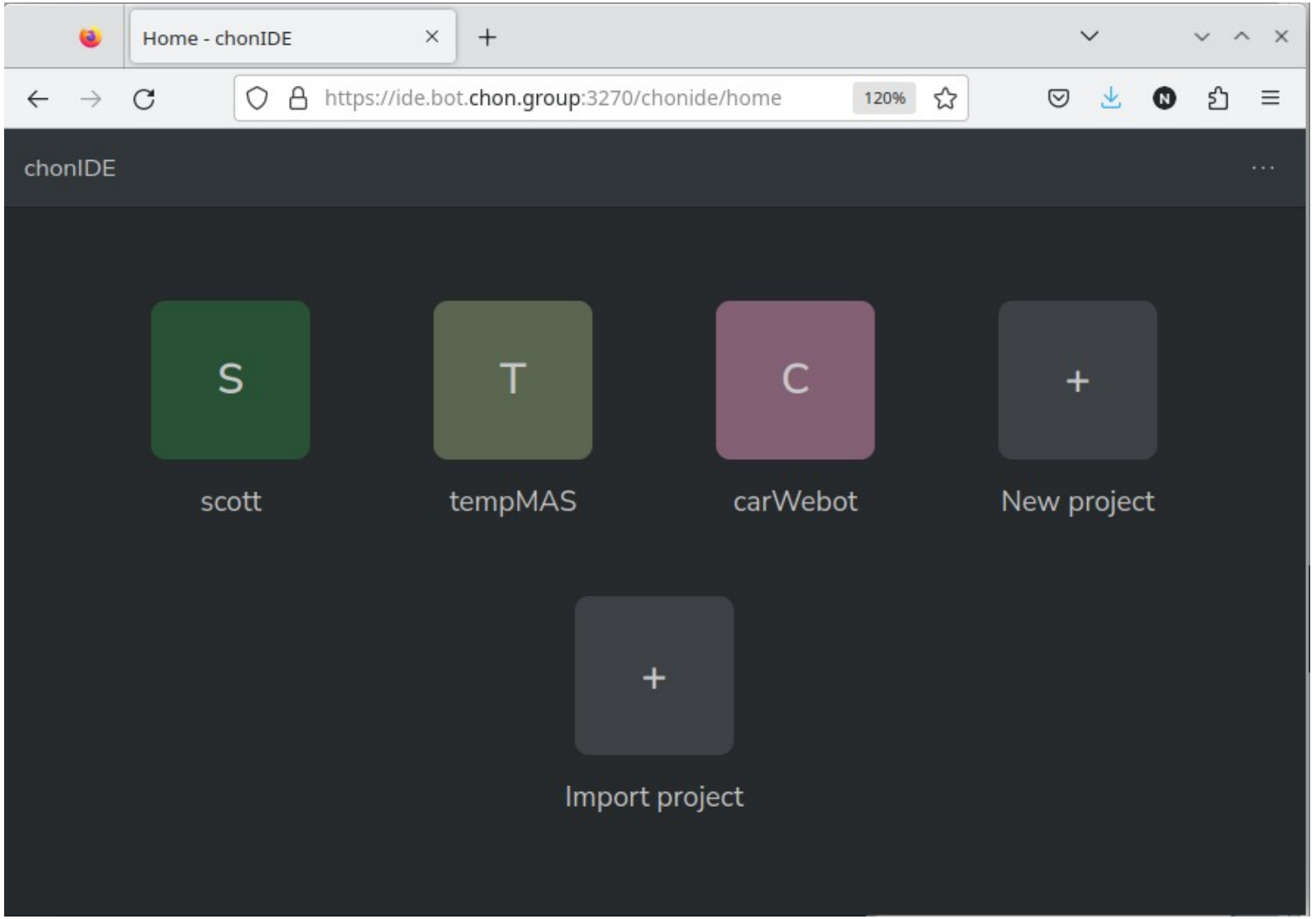
Webots


Home - chonIDE
120%
Envio de arquivos
Recentes
Pasta pessoal
Área de trabalho
Documentos
Downloads
Imagens
Músicas
Vídeos
Outros locais
nilson
Downloads
Car4WD
Nome
Tamanho
Tipo
Modificado
car4WDWebot.chon
1,1 kB
Texto
03:45
SimulatedWorld
03:49
Todos os arquivos
Cancelar
Abrir

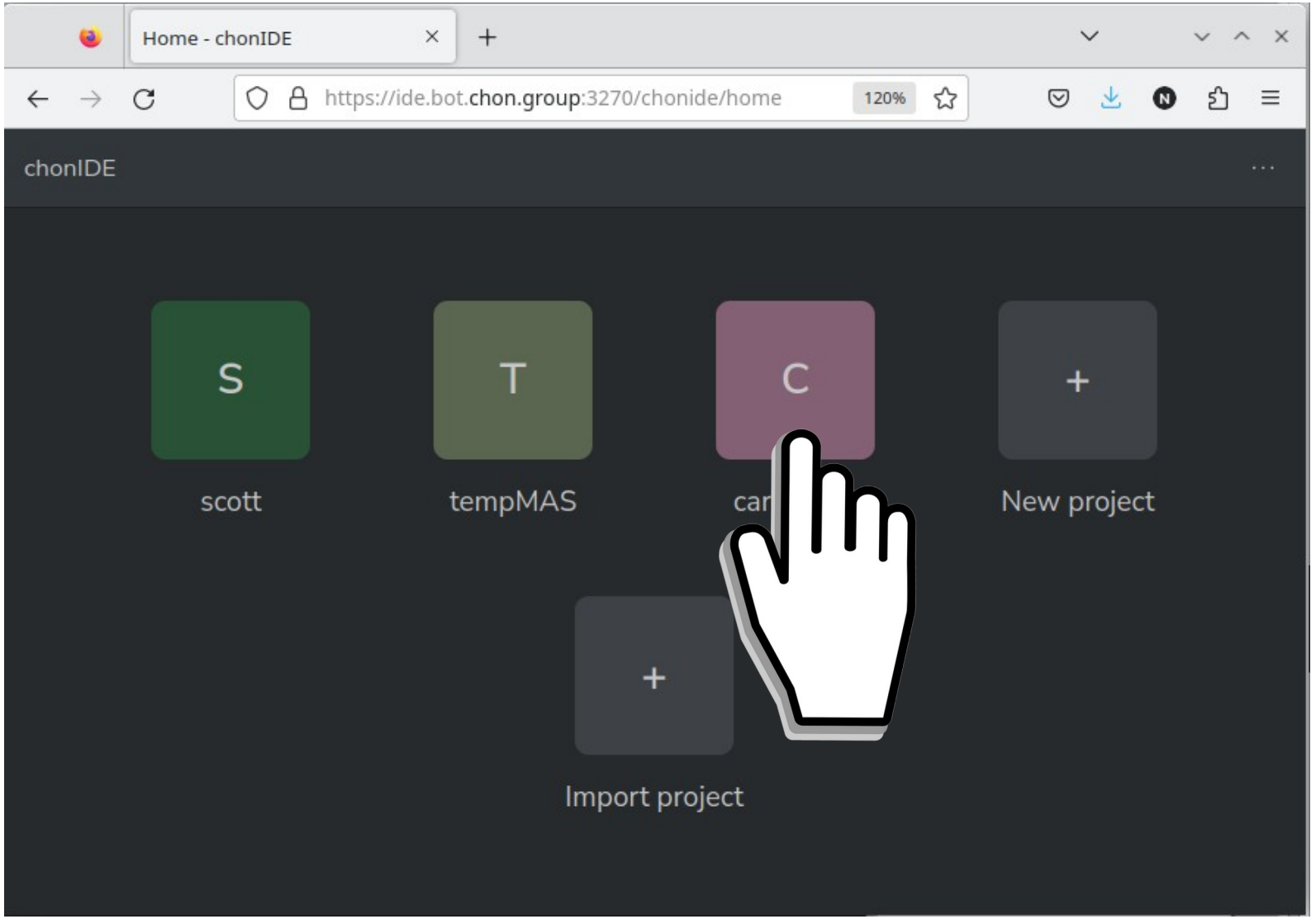
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
Project imported
S
T
C
+
scott
tempMAS
carWebot
New project
+
Import project

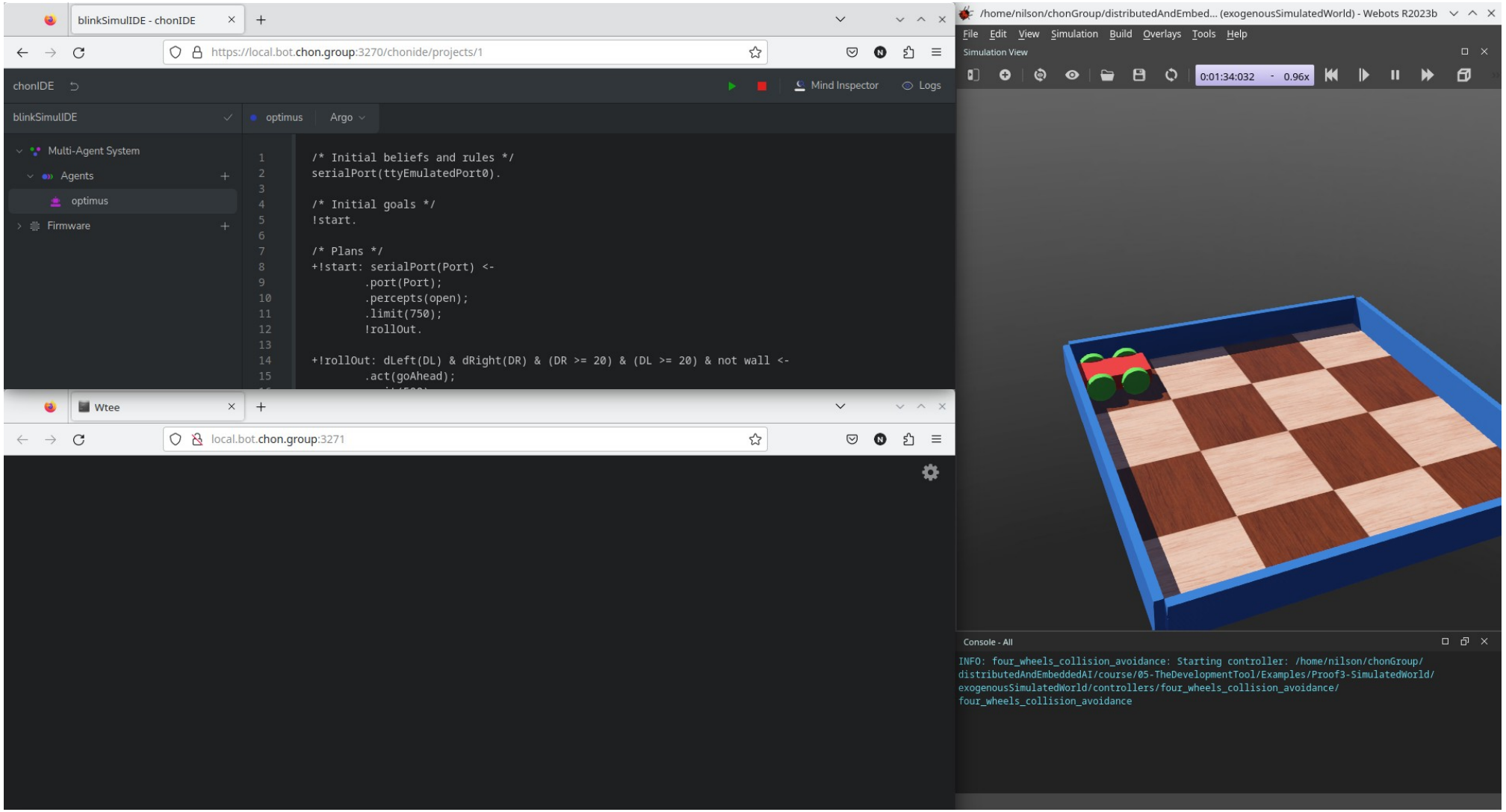
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
scott
T
tempMAS
C
carWebot
+
New project
+
Import project

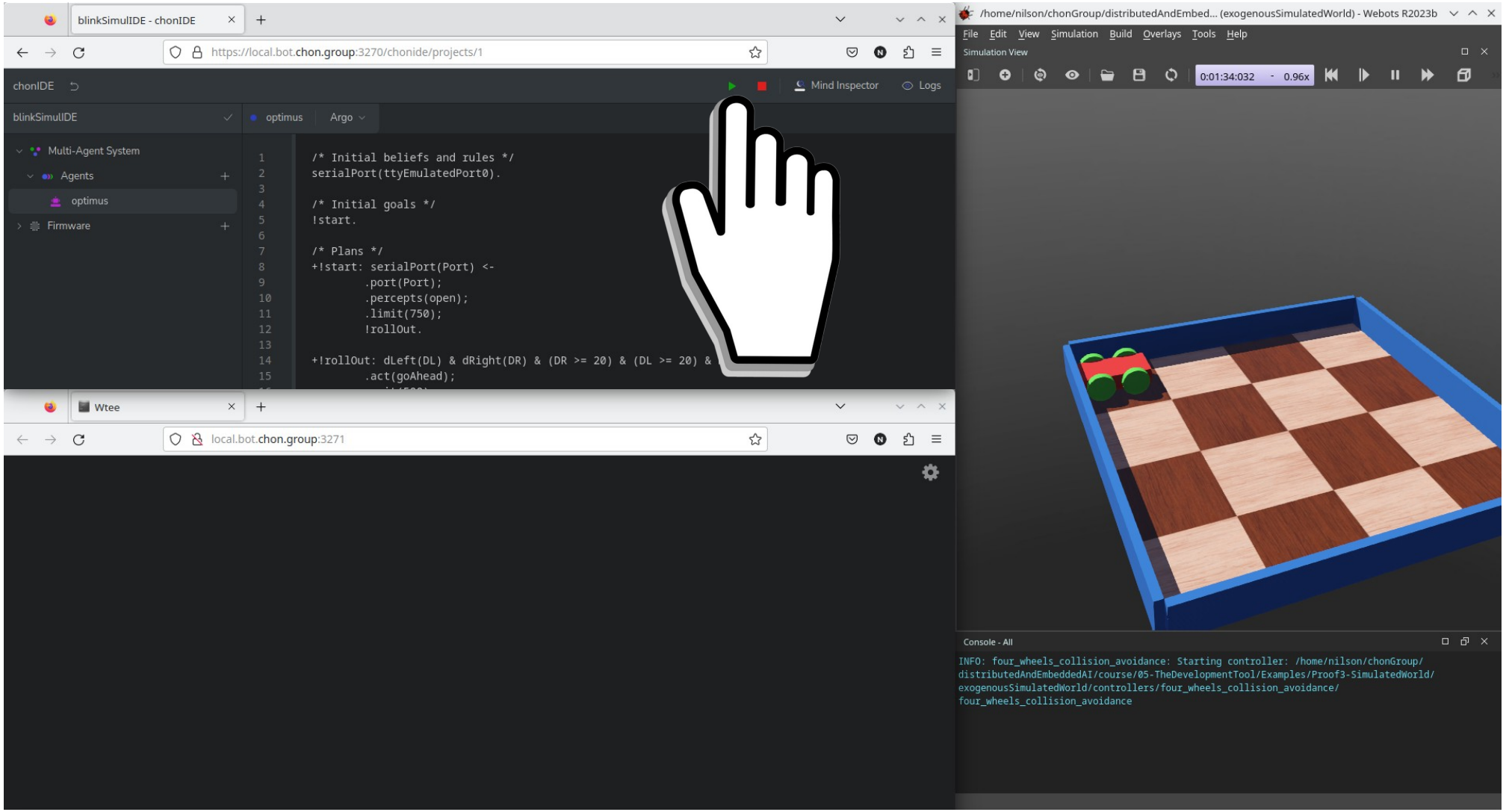
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
scott
T
tempMAS
C
car
+
New project
+
Import project

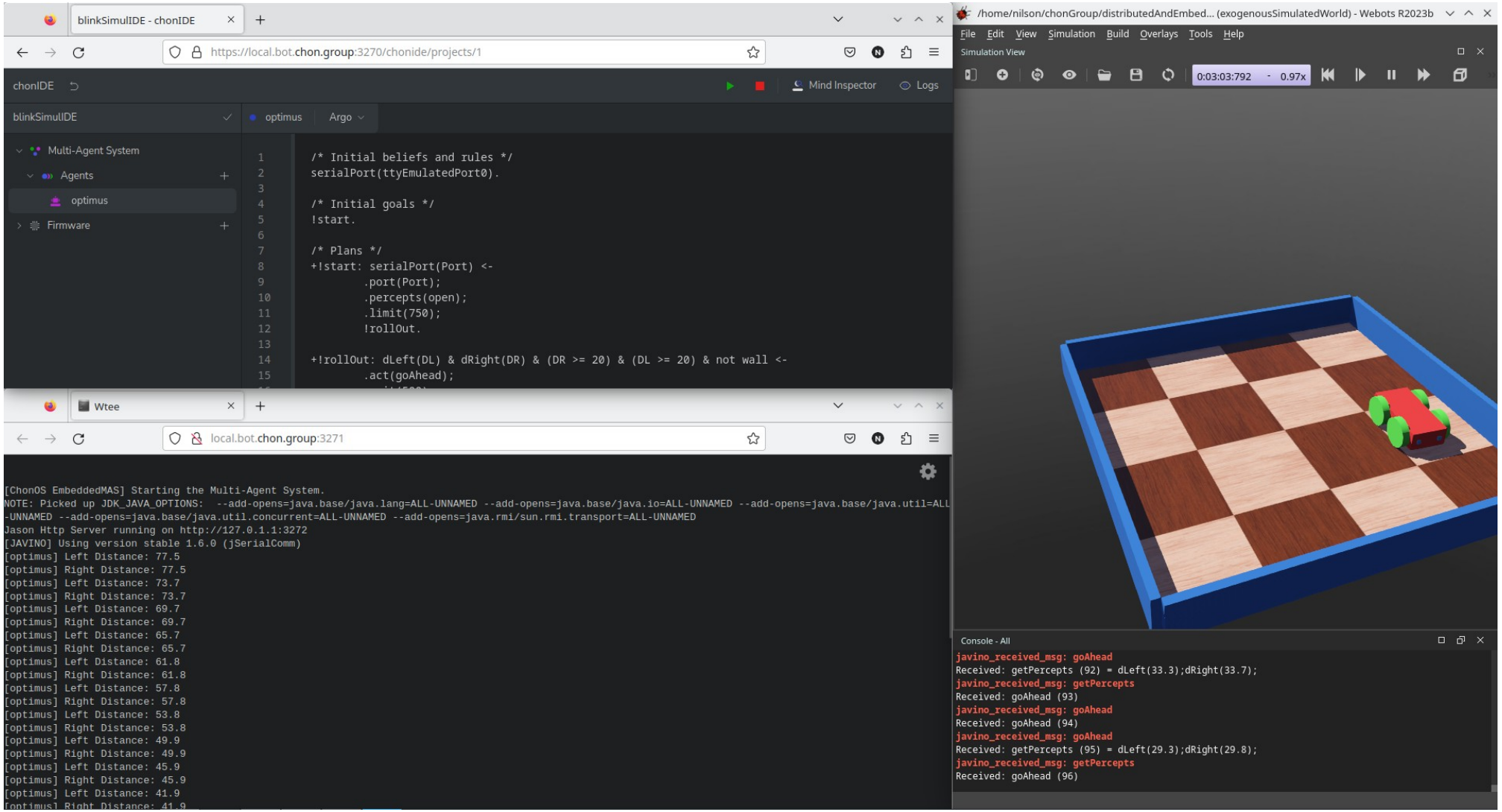
Webots


blinkSimulIDE - chonIDE
https://local.bot.chon.group:3270/chonide/projects/1
chonIDE
Mind Inspector
Logs
blinkSimulIDE
optimus
Argo
Multi-Agent System
Agents
optimus
Firmware
/* Initial beliefs and rules */
serialPort(ttyEmulatedPort0).
/* Initial goals */
!start.
/* Plans */
+!start: serialPort(Port) <-
.port(Port);
.percepts(open);
.limit(750);
!rollOut.
+!rollOut: dLeft(DL) & dRight(DR) & (DR >= 20) & (DL >= 20) & not wall <-
.act(goAhead);
Wtee
local.bot.chon.group:3271
/home/nilson/chonGroup/distributedAndEmbed... (exogenousSimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:01:34:032 - 0.96x
Console - All
INFO: four_wheels_collision_avoidance: Starting controller: /home/nilson/chonGroup/distributedAndEmbeddedAI/course/05-TheDevelopmentTool/Examples/Proof3-SimulatedWorld/exogenousSimulatedWorld/controllers/four_wheels_collision_avoidance/four_wheels_collision_avoidance

Webots


blinkSimulIDE - chonIDE
https://local.bot.chon.group:3270/chonide/projects/1
chonIDE
Mind Inspector
Logs
blinkSimulIDE
optimus
Argo
Multi-Agent System
Agents
optimus
Firmware
/* Initial beliefs and rules */
serialPort(ttyEmulatedPort0).
/* Initial goals */
!start.
/* Plans */
+!start: serialPort(Port) <-
.port(Port);
.percepts(open);
.limit(750);
!rollOut.
+!rollOut: dLeft(DL) & dRight(DR) & (DR >= 20) & (DL >= 20) &
.act(goAhead);
Wtee
local.bot.chon.group:3271
/home/nilson/chonGroup/distributedAndEmbed... (exogenousSimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:01:34:032 - 0.96x
Console - All
INFO: four_wheels_collision_avoidance: Starting controller: /home/nilson/chonGroup/distributedAndEmbeddedAI/course/05-TheDevelopmentTool/Examples/Proof3-SimulatedWorld/exogenousSimulatedWorld/controllers/four_wheels_collision_avoidance/four_wheels_collision_avoidance

Webots


blinkSimulIDE - chonIDE
https://local.bot.chon.group:3270/chonide/projects/1
chonIDE
Mind Inspector
Logs
blinkSimulIDE
optimus
Argo
Multi-Agent System
Agents
optimus
Firmware
/* Initial beliefs and rules */
serialPort(ttyEmulatedPort0).
/* Initial goals */
!start.
/* Plans */
+!start: serialPort(Port) <-
.port(Port);
.percepts(open);
.limit(750);
!rollOut.
+!rollOut: dLeft(DL) & dRight(DR) & (DR >= 20) & (DL >= 20) & not wall <-
.act(goAhead);
Wtee
local.bot.chon.group:3271
[ChonOS EmbeddedMAS] Starting the Multi-Agent System.
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Jason Http Server running on http://127.0.1.1:3272
[JAVINO] Using version stable 1.6.0 (jSerialComm)
[optimus] Left Distance: 77.5
[optimus] Right Distance: 77.5
[optimus] Left Distance: 73.7
[optimus] Right Distance: 73.7
[optimus] Left Distance: 69.7
[optimus] Right Distance: 69.7
[optimus] Left Distance: 65.7
[optimus] Right Distance: 65.7
[optimus] Left Distance: 61.8
[optimus] Right Distance: 61.8
[optimus] Left Distance: 57.8
[optimus] Right Distance: 57.8
[optimus] Left Distance: 53.8
[optimus] Right Distance: 53.8
[optimus] Left Distance: 49.9
[optimus] Right Distance: 49.9
[optimus] Left Distance: 45.9
[optimus] Right Distance: 45.9
[optimus] Left Distance: 41.9
/home/nilson/chonGroup/distributedAndEmbed... (exogenousSimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:03:03:792 - 0.97x
Console - All
javino_received_msg: goAhead
Received: getPercepts (92) = dLeft(33.3);dRight(33.7);
javino_received_msg: getPercepts
Received: goAhead (93)
javino_received_msg: goAhead
Received: goAhead (94)
javino_received_msg: goAhead
Received: getPercepts (95) = dLeft(29.3);dRight(29.8);
javino_received_msg: getPercepts
Received: goAhead (96)

Acknowledgements

THANK YOU!

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