

Introduction to Distributed and Embedded Multi-agent Systems

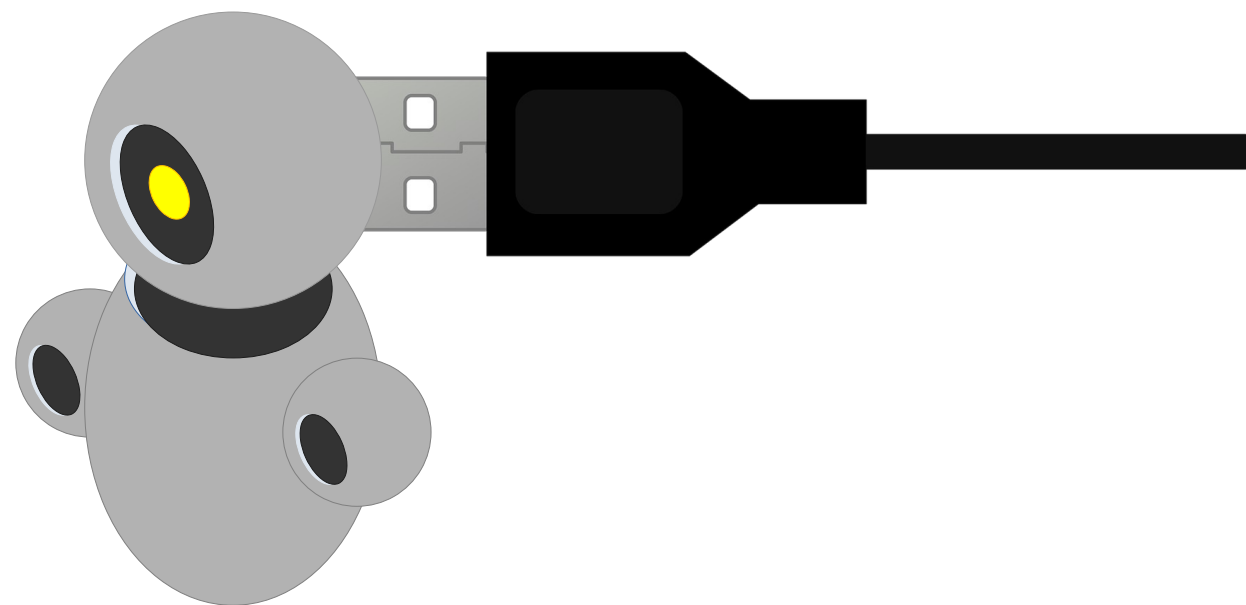
Carlos Eduardo Pantoja¹
Nilson Mori Lazarin^{1,2}

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



Junho, 2024

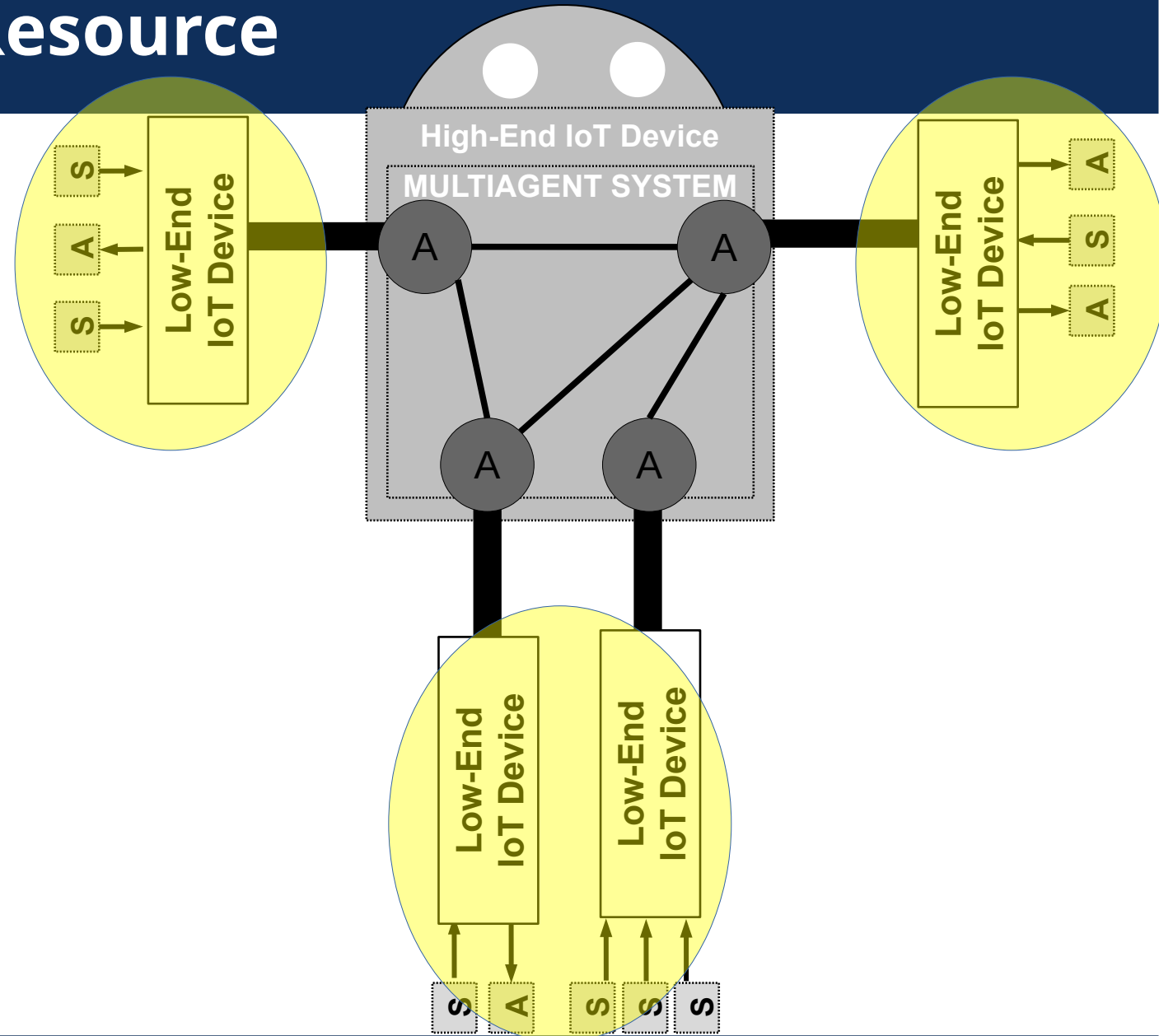
Agent-operable Physical Resource



Agent-operable Physical Resource

É um Low-End IoT Device que captura os dados brutos do ambiente, através dos sensores acoplados e os transforma em percepções bem definidas para interpretação do agente.

Além disso, altera o estado do ambiente, através dos atuadores acoplados, conforme as ações determinados pelo agente.



- O Javino é um protocolo que:
 - Permite a troca de mensagem entre hardware e linguagens de programação de alto nível;
 - É composto por dupla biblioteca para comunicação serial;
 - Provê detecção de erros através de uma rotina de verificação da recepção de dados.

- O Javino é um protocolo que:
 - **Permite a troca de mensagem entre hardware e linguagens de programação de alto nível;**
 - É composto por dupla biblioteca para comunicação serial;
 - Provê detecção de erros através de uma rotina de verificação da recepção de dados.

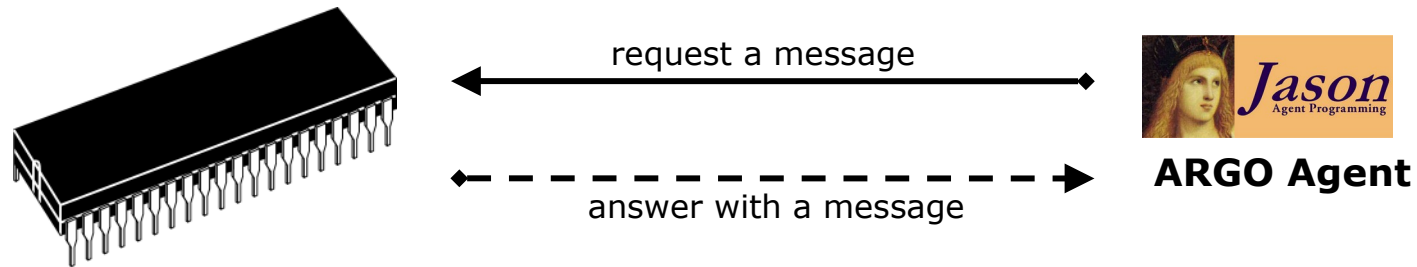
- O Javino é um protocolo que:
 - Permite a troca de mensagem entre hardware e linguagens de programação de alto nível;
 - **É composto por dupla biblioteca para comunicação serial;**
 - Provê detecção de erros através de uma rotina de verificação da recepção de dados.

- O Javino é um protocolo que:
 - Permite a troca de mensagem entre hardware e linguagens de programação de alto nível;
 - É composto por dupla biblioteca para comunicação serial;
 - **Provê detecção de erros através de uma rotina de verificação da recepção de dados.**

Javino: Operation Modes

- **Request Mode**

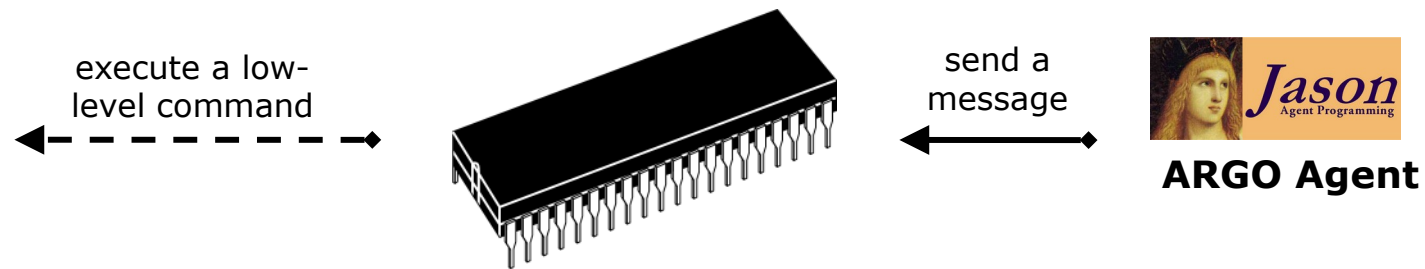
- do software para o hardware;
- o hardware responde com uma mensagem.



Javino: Operation Modes

- **Send Mode**

- do software para o hardware;
- o hardware executa uma ação.



Javino: Aplicação em Arduino

Para **iniciar a comunicação em ARDUINO**:

- inclui-se a **biblioteca** Javino;
- cria-se uma **variável** do tipo Javino para manipulação da comunicação com o Java;
- determina-se a **velocidade da comunicação** pela porta serial;

```
#include <Javino.h>
Javino j;

void setup(){
    Serial.begin(9600);
}
```

Javino: Aplicação em Arduino

Além disso, o Javino fornece **recursos** de verificação, leitura e escrita que são chamados nas seguintes funções Arduino:

```
javino.perceive(getExogenousPerceptions);
```



Define qual função será chamada quando o agente requisitar as percepções do ambiente exógeno através do ciclo de raciocínio.

Javino: Aplicação em Arduino

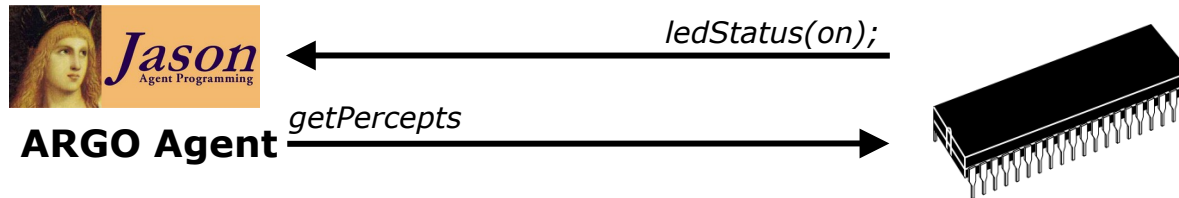
```
javino.perceive(getExogenousPerceptions);
```



```
void getExogenousPerceptions(){  
    if(digitalRead(13)==1)  
        javino.addPercept("ledStatus(on)");  
    else  
        javino.addPercept("ledStatus(off)");  
}
```

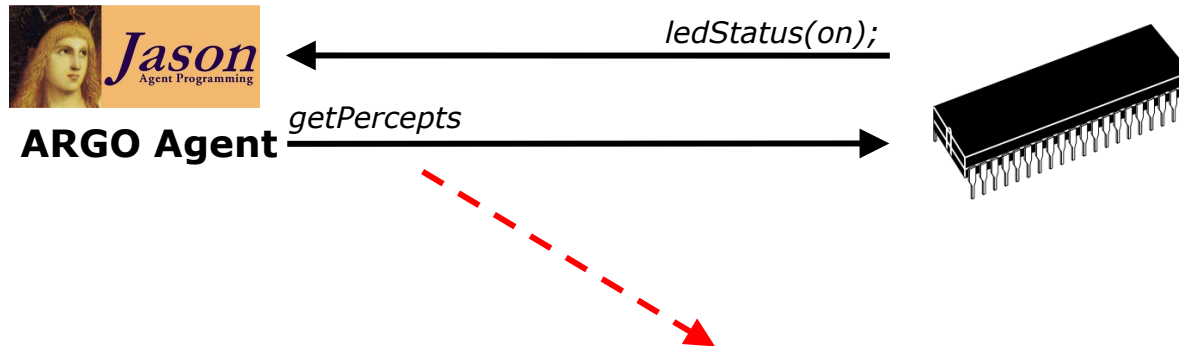
Exemplos

- Para isso, um procedimento de envio das percepções deve ser programado em resposta a um estímulo ***getPercepts*** (modo request):



Exemplos

- Para isso, um procedimento de envio das percepções deve ser programado em resposta a um estímulo ***getPercepts*** (modo request):



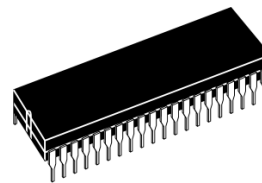
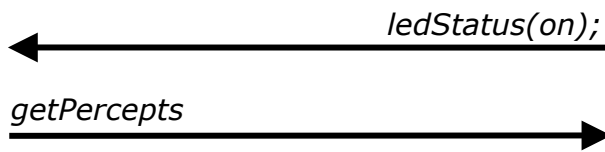
`javino.perceive(getExogeneousPerception)`

Exemplos

- Para isso, um procedimento de envio das percepções deve ser programado em resposta a um estímulo **getPercepts** (modo request):



ARGO Agent

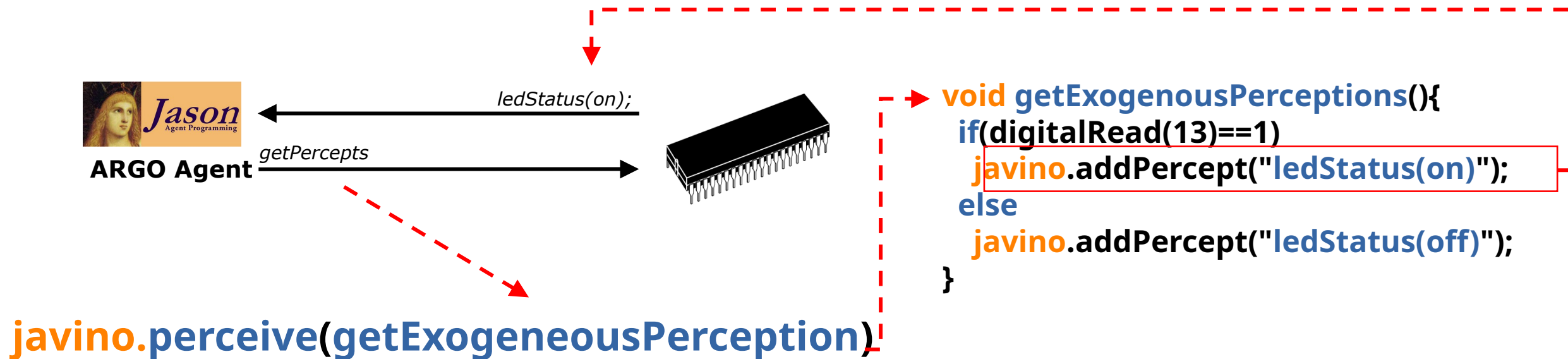


```
void getExogenousPerceptions(){  
    if(digitalRead(13)==1)  
        javino.addPercept("ledStatus(on)");  
    else  
        javino.addPercept("ledStatus(off)");  
}
```

`javino.perceive(getExogeneousPerception)`

Exemplos

- Para isso, um procedimento de envio das percepções deve ser programado em resposta a um estímulo **getPercepts** (modo request):



Javino: Aplicação em Arduino

Além disso, o Javino fornece **recursos** de verificação, leitura e escrita que são chamados nas seguintes funções Arduino:

```
javino.act["ledOn"] = ledOn;  
javino.act["ledOff"] = ledOff;
```



Define qual função será chamada quando o agente enviar um comando para ser executado por um efetuator no ambiente exógeno.

Javino: Aplicação em Arduino

```
javino.act["ledOn"] = ledOn;  
javino.act["ledOff"] = ledOff;
```



```
void ledOn(){  
    digitalWrite(13, HIGH);  
}
```

```
void ledOff(){  
    digitalWrite(13, LOW);  
}
```

Javino: Aplicação em Arduino

Além disso, o Javino fornece **recursos** de verificação, leitura e escrita que são chamados nas seguintes funções Arduino:

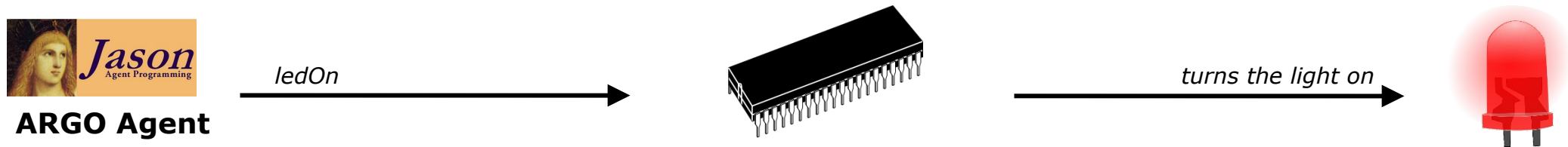
```
javino.addPercept("ledStatus(off)");
```



A cada requisição o microcontrolador envia uma **mensagem de até 256 caracteres** ao agente com todas as crenças adicionadas pelo designer.

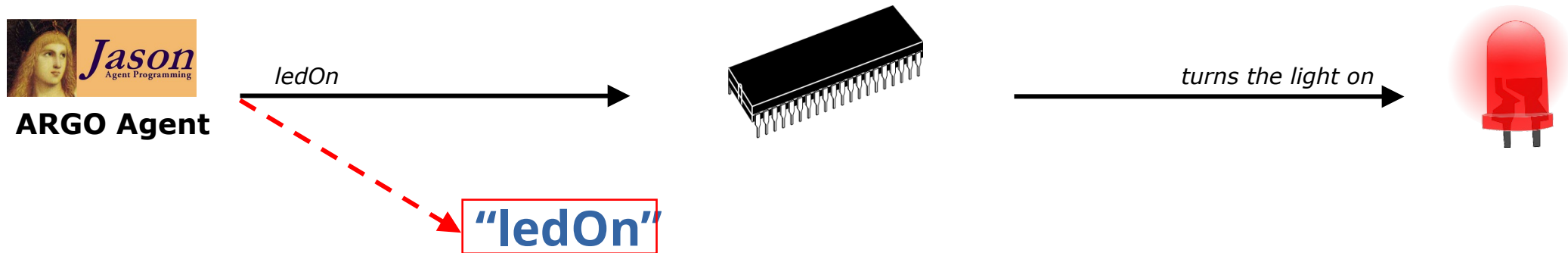
Exemplos

- O Javino precisa ser programado em modo send e request na controladora
- Para cada led um procedimento de ativação deve ser programado em resposta a um estímulo (modo send):



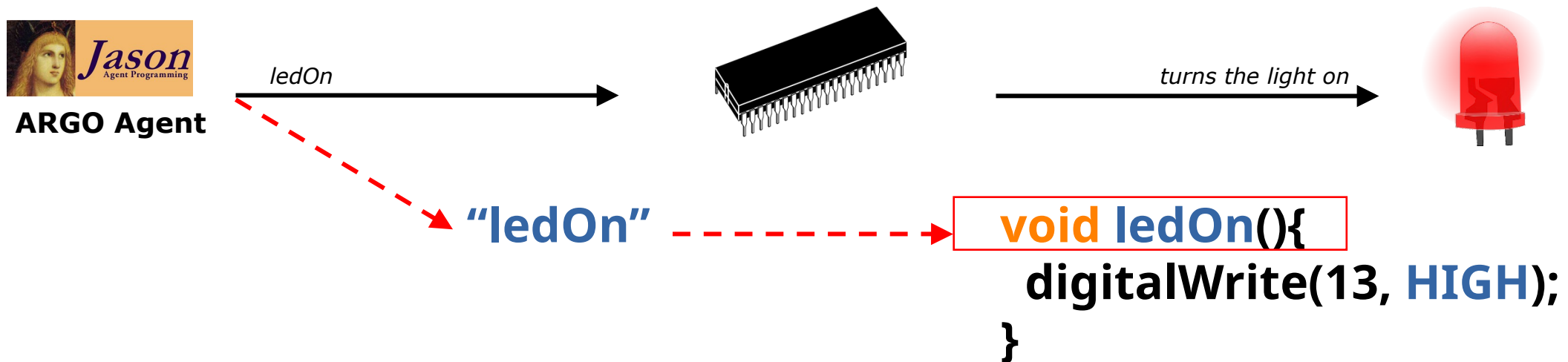
Exemplos

- O Javino precisa ser programado em modo send e request na controladora
- Para cada led um procedimento de ativação deve ser programado em resposta a um estímulo (modo send):



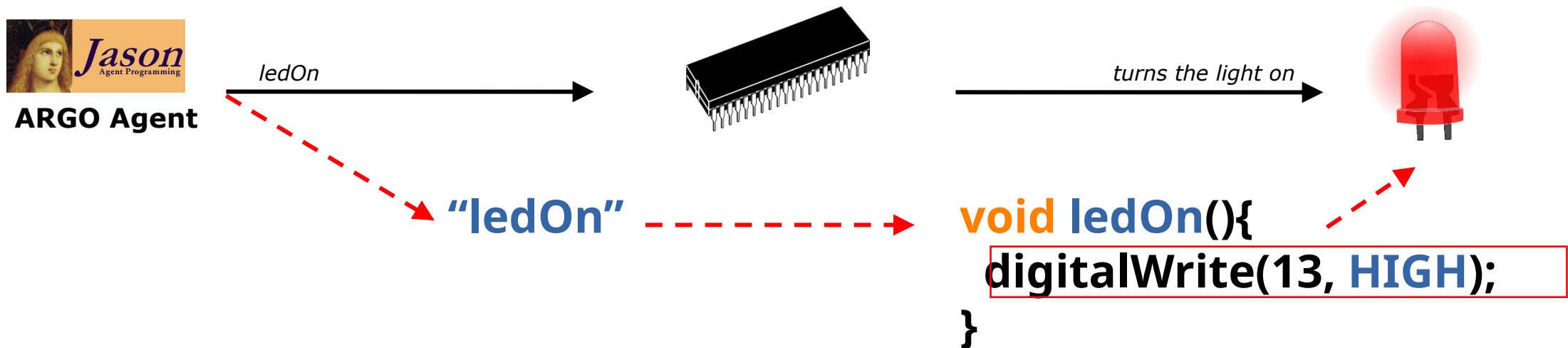
Exemplos

- O Javino precisa ser programado em modo send e request na controladora
- Para cada led um procedimento de ativação deve ser programado em resposta a um estímulo (modo send):



Exemplos

- O Javino precisa ser programado em modo send e request na controladora
- Para cada led um procedimento de ativação deve ser programado em resposta a um estímulo (modo send):

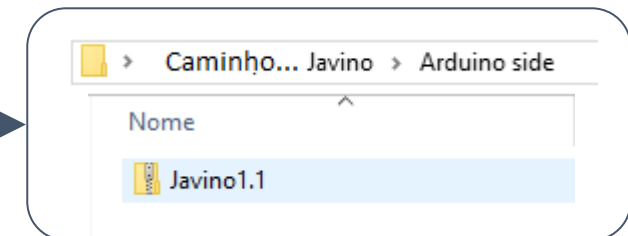
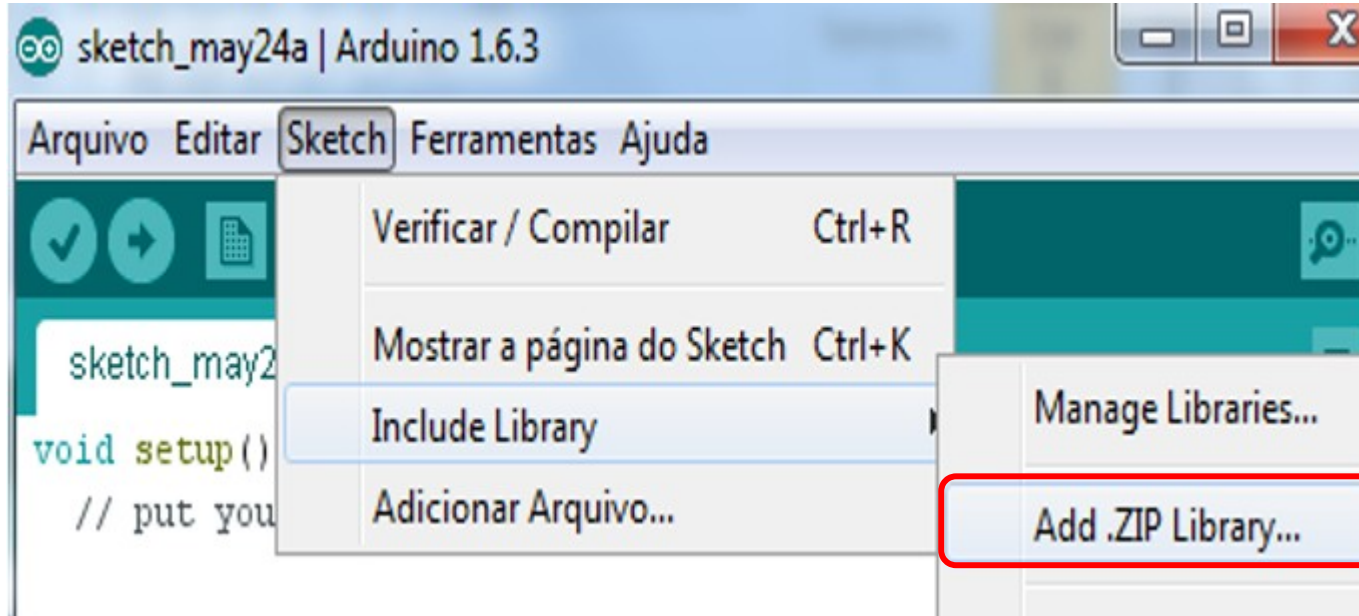


Javino: Aplicação em Arduino

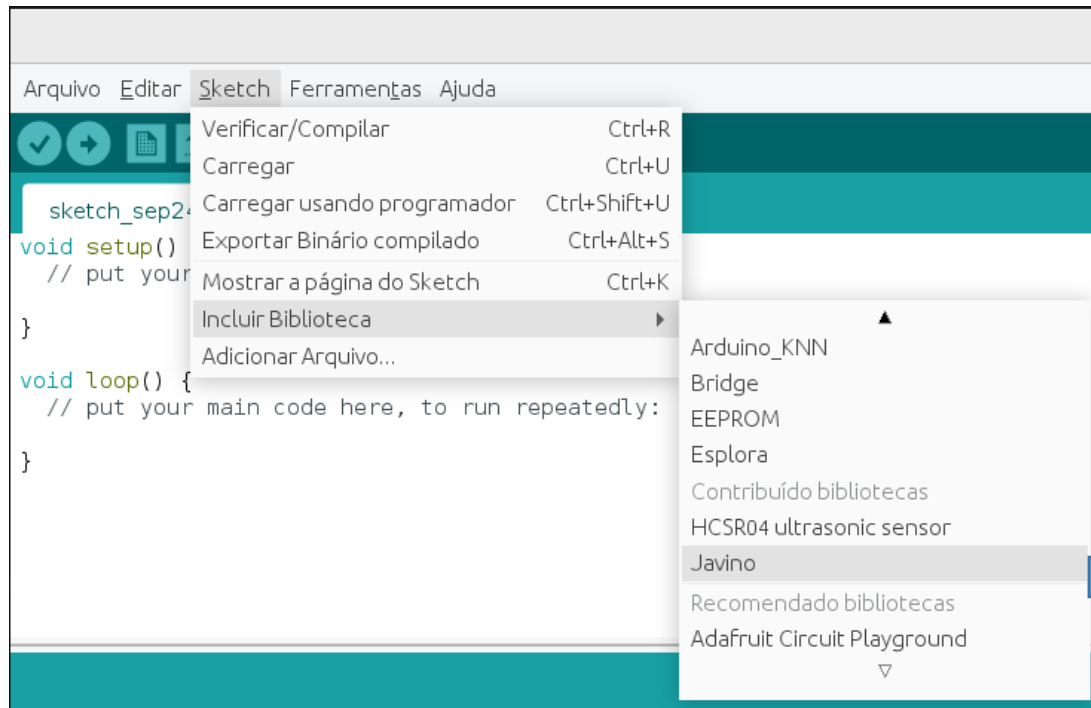


EXEMPLO:

Em seu projeto, adicione a biblioteca Javino para arduino:

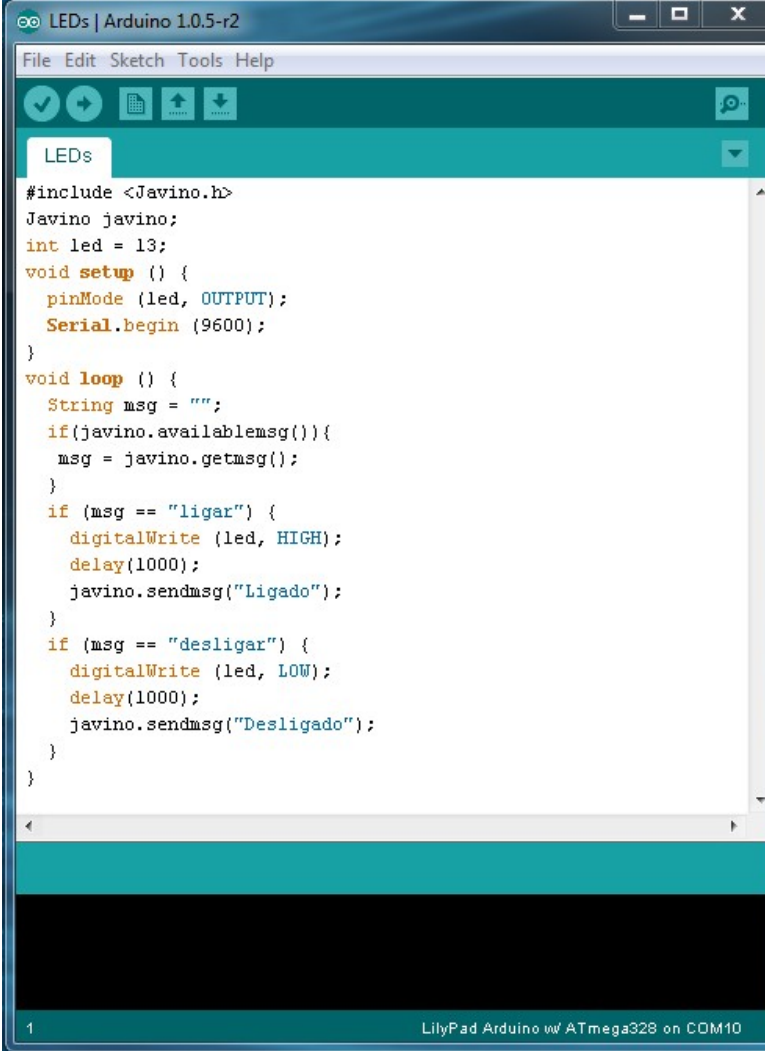


Javino: Importando o Javino



Acionamento de Led via Javino

Neste exemplo, o arduino **receberá** mensagens externas e **executará** uma determinada tarefa e **retornará** com uma resposta.



```
#include <Javino.h>
Javino javino;
int led = 13;
void setup () {
  pinMode (led, OUTPUT);
  Serial.begin (9600);
}
void loop () {
  String msg = "";
  if(javino.availablemsg()){
    msg = javino.getmsg();
  }
  if (msg == "ligar") {
    digitalWrite (led, HIGH);
    delay(1000);
    javino.sendmsg("Ligado");
  }
  if (msg == "desligar") {
    digitalWrite (led, LOW);
    delay(1000);
    javino.sendmsg("Desligado");
  }
}
```

Javino: Request Mode no Arduino

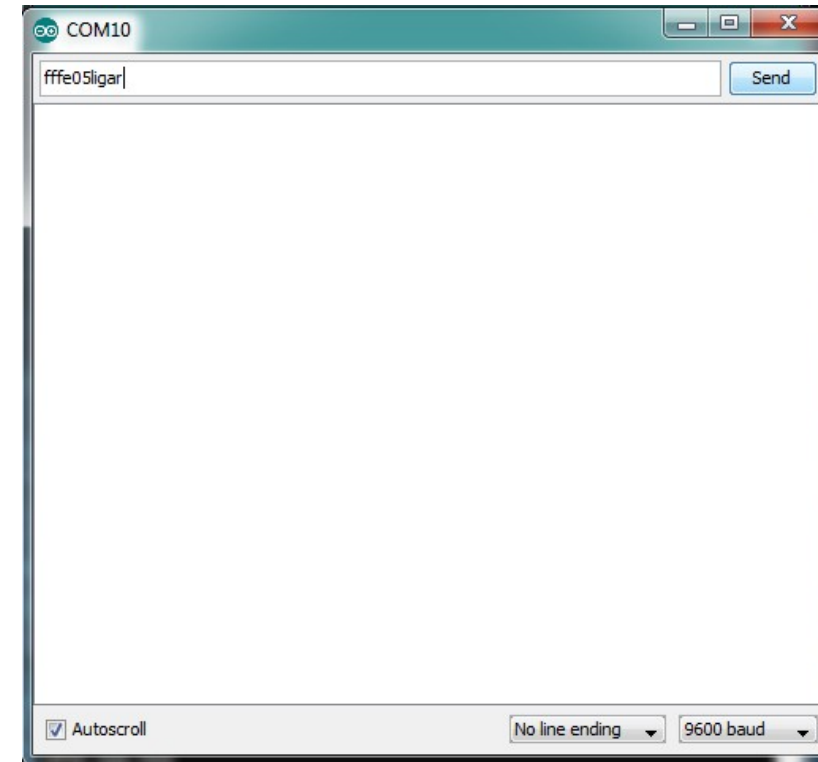
Enviando o comando de **ligar** o Led.

fffe 05 ligar

Pré-âmbulo para
verificação de
recebimento da
mensagem

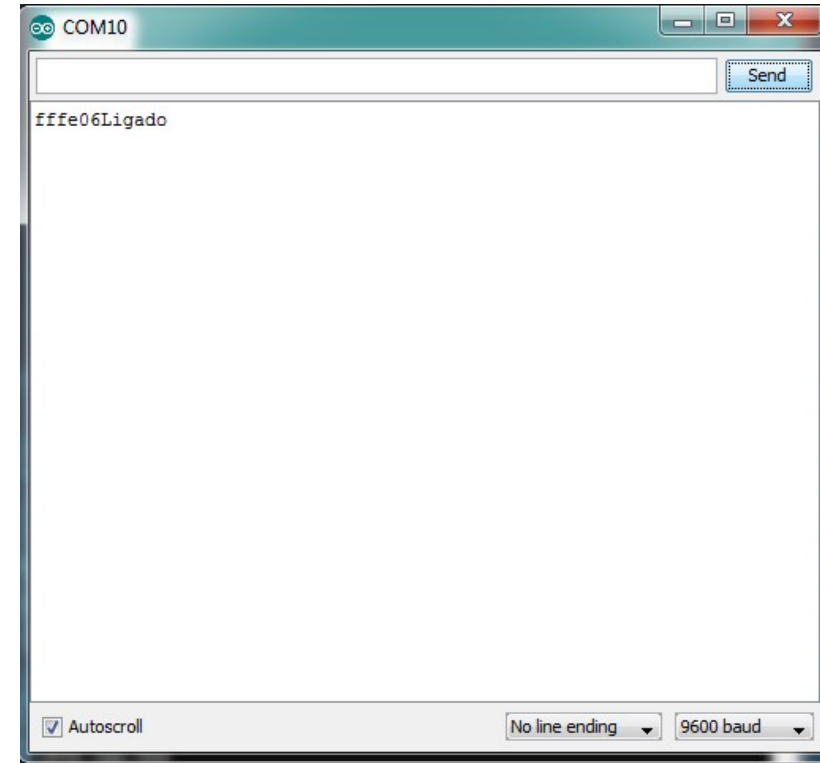
Tamanho do
conteúdo a ser
enviado em
hexadecimal

Conteúdo a ser
enviado



Javino: Request Mode no Arduino

Resultado do comando de **ligar**
o Led.



OBRIGADO!

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

