Programação em Fluxo para a Internet das Coisas com Soletta / Intel Edison

Bruno Dilly / Leandro Dorileo
bruno.dilly@intel.com / *leandro.maciel.dorileo@intel.com*

Sumário

- O que é Internet das Coisas ?
- Soletta
- Programação baseada em fluxo
- O que é aprendizado de máquina ?
  - Processamento local ou nas nuvens?
  - O SML.
- Protótipos.
- Robótica com Soletta?
- Mão na massa
- Comunicação

O que é Internet das Coisas ?

- Pequenos dispositivos com processamento.
- Comunicação e troca de dados entre eles.
- O que fazer com tantos dados ?
- Privacidade

Projetos do Soletta

- Base do Soletta
- Editor textual
- Editor visual
- Aprendizado de máquina
- Prototipos

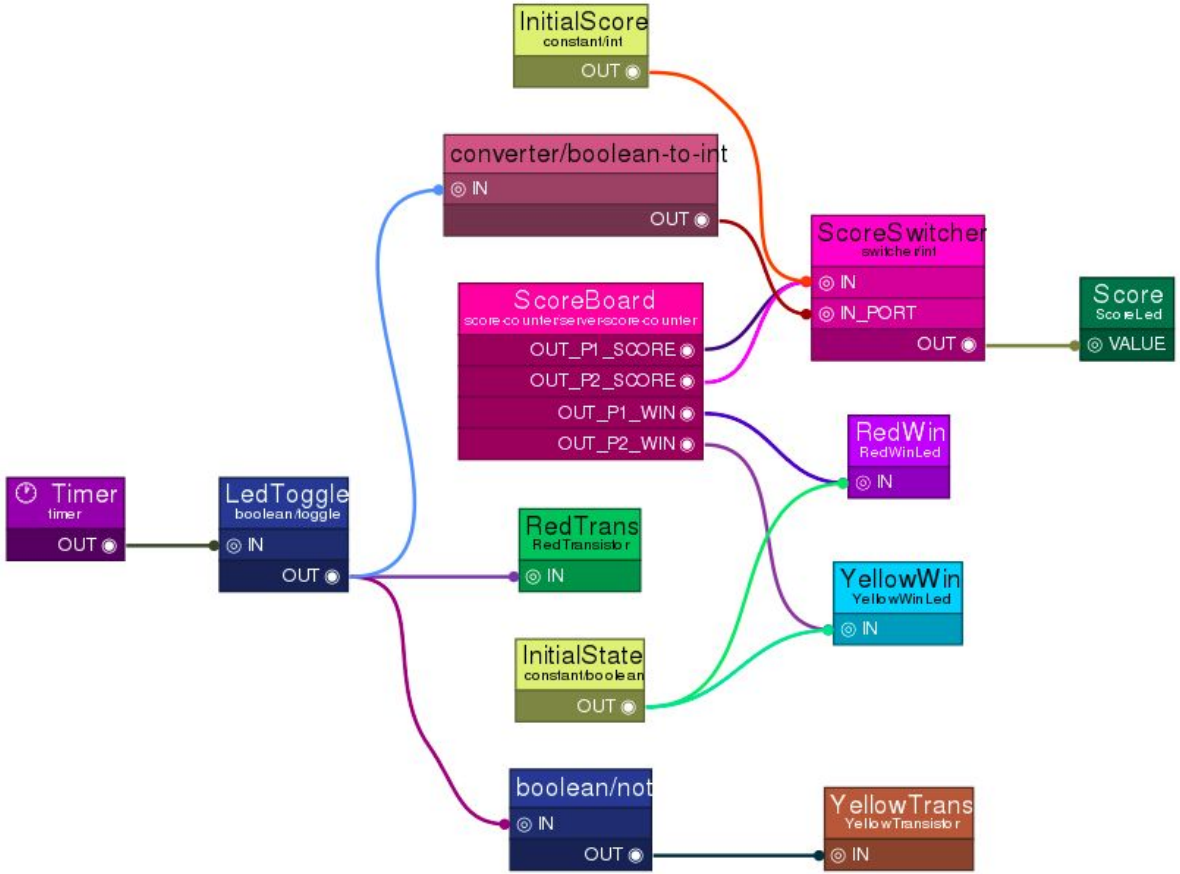
Soletta - base

- Varios sistemas operacionais: Linux, Contiki, Riot
- Abstração da plataforma
- API uniforme para loop de eventos
- Primitivas de I/O basica uniforme (GPIO, AIO, PWM, I2C)
- Programação em alto nível (C / C++, JavaScript, FBP)
- Programação em fluxo.
- Comunicação: OIC, CoAP, MQTT, Bluetooth e HTTP (cliente e servidor)

FBP

- Programação baseada em fluxo (Flow-Based Programming)
- “Fábrica de Dados”
- Orientado a componente
- Visual
- Paradigma diferente
- Uma aplicação FBP é uma rede de processos “caixa preta” (nós), que transferem dados através de mensagens passando por conexões predefinidas.
- A rede é definida externamente aos processos, e essa lista de conexões é interpretada por um programa executor.

Exemplo de FBP - Placar



Soletta™

Coisas legais da FBP

- Aplicações são mais fáceis de entender e manter.
- Componentes podem ser reutilizados
- A prototipagem é rápida
- Dá para testar componentes isoladamente
- Comunicação melhor entre designers, programadores e usuários.

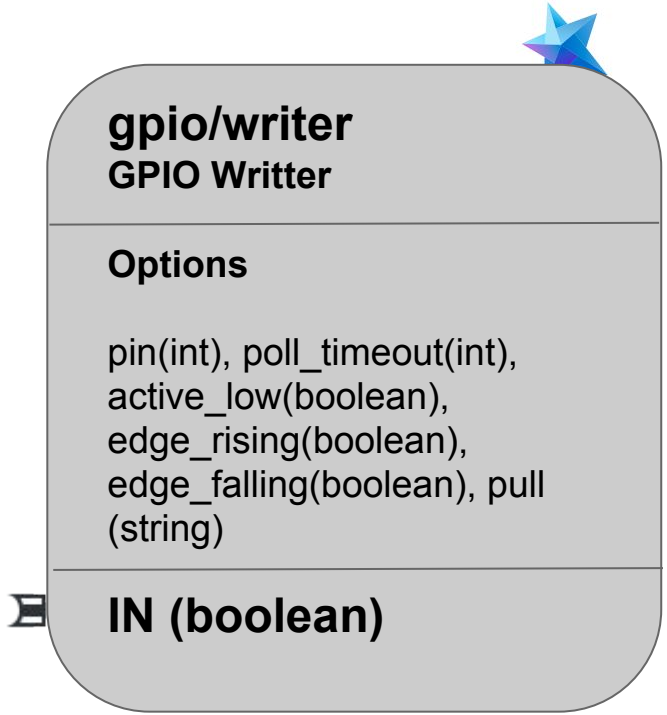
Elementos da FBP

- Tipos de Nós
- Nós (processos)
- Portas nomeadas
- Conexões externas aos nós
- Pacotes de informação

Tipos de Nós

Cada tipo de nó tem portas e opções de inicialização:

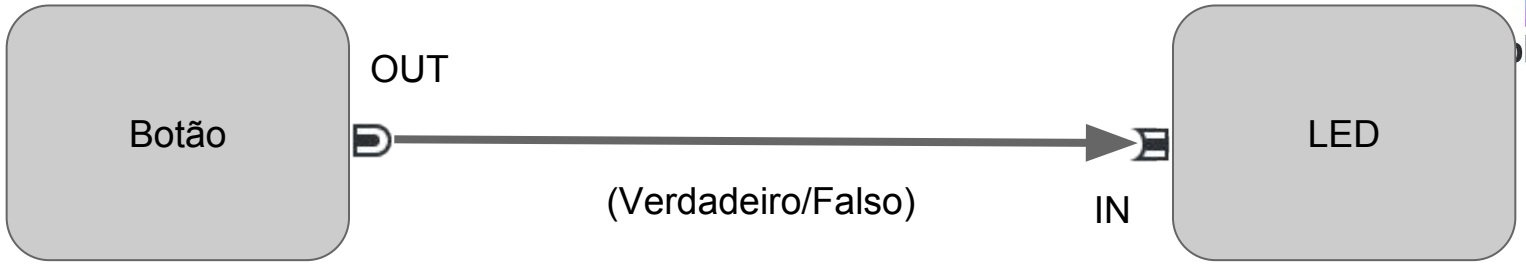
- Portas (tipos de pacotes)
- Comportamento



(Em FBP)

```
led(gpio/writer:pin=10)
```

Nós: Fluxo com dois nós



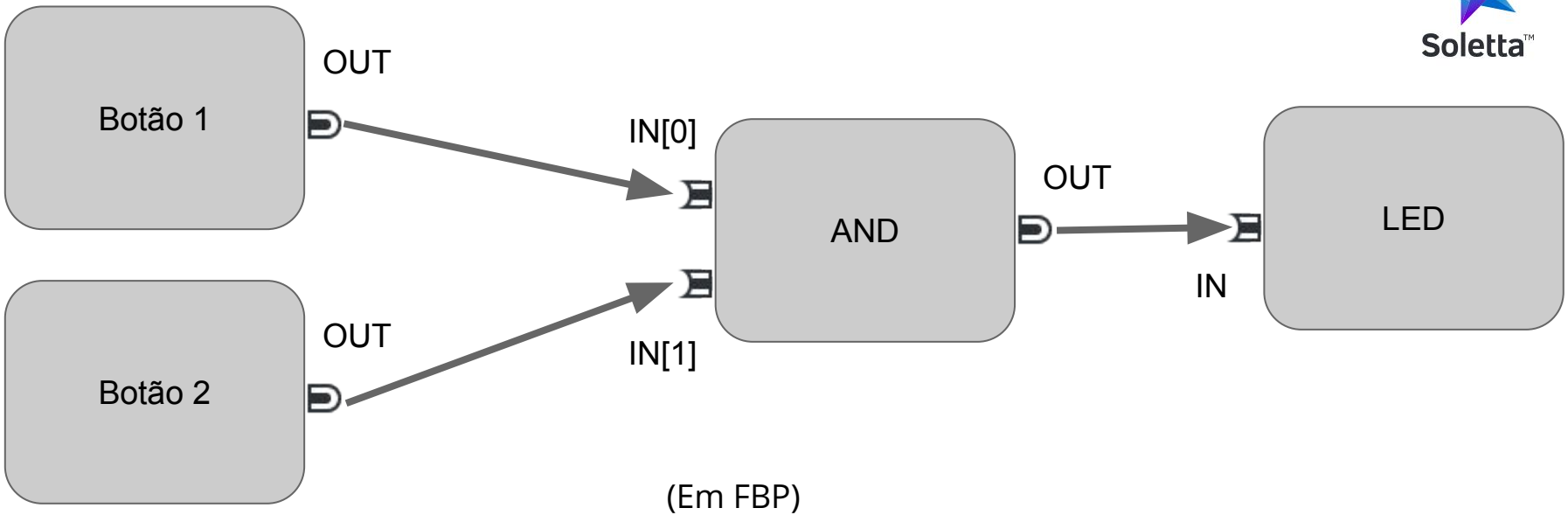
Botão envia um pacote com valor “Verdadeiro” quando é pressionado e “Falso” quando é solto.

LED acende quando recebe pacote “Verdadeiro” e apaga com pacote “Falso”

(Em FBP)

```
botao OUT -> IN led
```

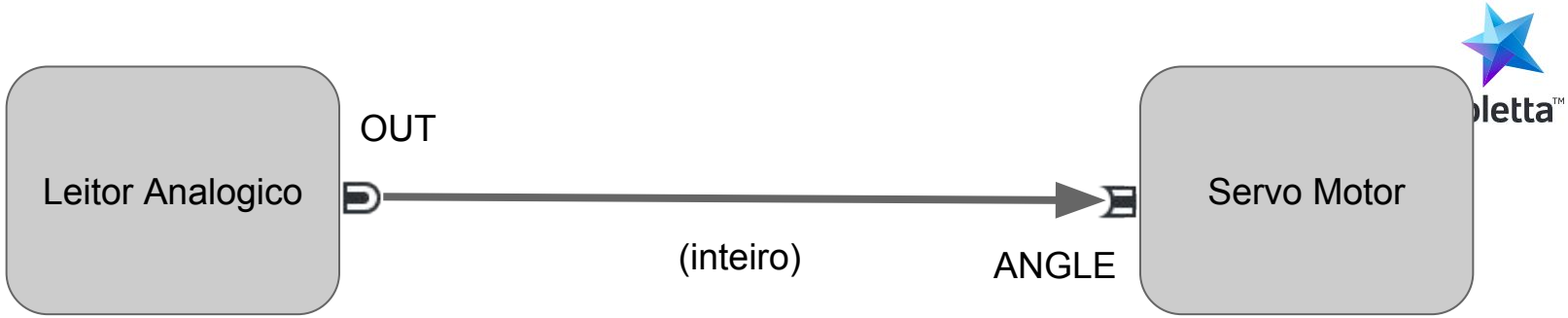
Conexões: portas com multiplas conexões



(Em FBP)

```
botao1 OUT -> IN[0] and
botao2 OUT -> IN[1] and
   and OUT -> IN led
```

Packets: types



Portas tem um tipo de pacote associado. Pode ser boleano, inteiro, fracional e outros, inclusive customizados.

O tipo de pacote é checado pela base.

(FBP input language)

```
aio OUT -> ANGLE servo
```

Elementos da FBP (para o Soletta)

- Cada nó tem um conjunto de portas de entrada e saída
- Cada porta tem um tipo de pacote definido
  - Algumas portas aceitam qualquer tipo
- Nós se comunicam com outros através de pacotes

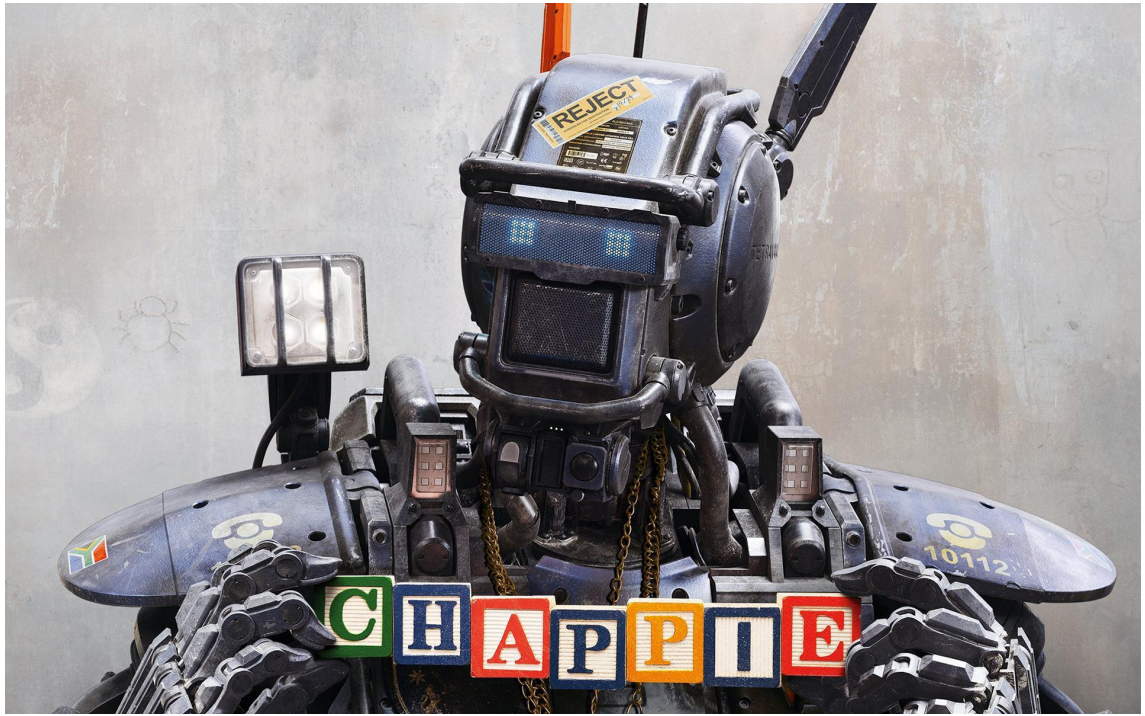
Sintaxe da FBP


```
No1(TipoNo:opcao1=valor1,opcao2=valor2) PORTA_SAIDA -> PORTA_ENTRADA No2
```

- “No1” e “No2” são nomes unicos.
- TipoNo é um componente a ser instanciado
- Opcoes sao separadas por virgula. São usadas para parametrizar os nos.
- “PORTA_SAIDA” é o nome da porta de saída do No1 usada na conexão
- “PORTA_ENTRADA” é o nome da porta de entrada do No2. Tipo compativel com “PORTA_SAIDA”.

O que é aprendizado de máquina?

Soletta™



Processamento local ou nas nuvens?

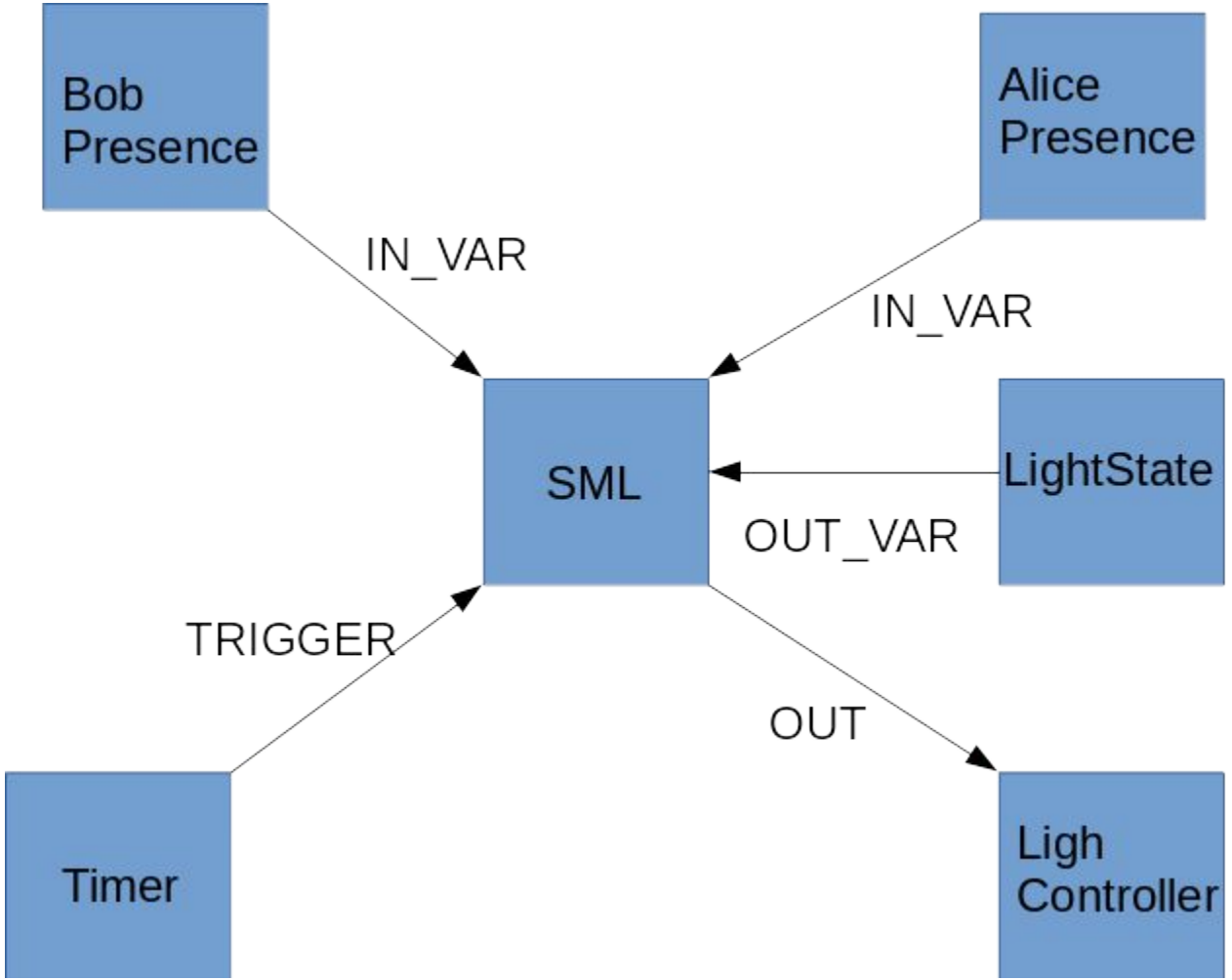
- Dispositivos pequenos podem não ter acesso a *Internet*.
- Acesso restrito a *Internet*.
- Privacidade e segurança.
- Tempo de resposta rápido.

O Soletta Machine Learning

- *Fuzzy* e Rede neural
- Suporte para o Soletta
- Extensível
- Não é necessário conhecimentos profundos sobre algoritmos de aprendizado de máquina.

SML

- Usuário define entradas e saídas.
- De tempos em tempos usuário fornece ao SML o estado do sistema.
  - Caso esteja treinado o SML irá fazer uma predição para o(s) valore(s) de saída.


Bob Presence
Alice Presence
IN_VAR
IN_VAR
SML
LightState
OUT_VAR
TRIGGER
OUT
Timer
Ligh Controller

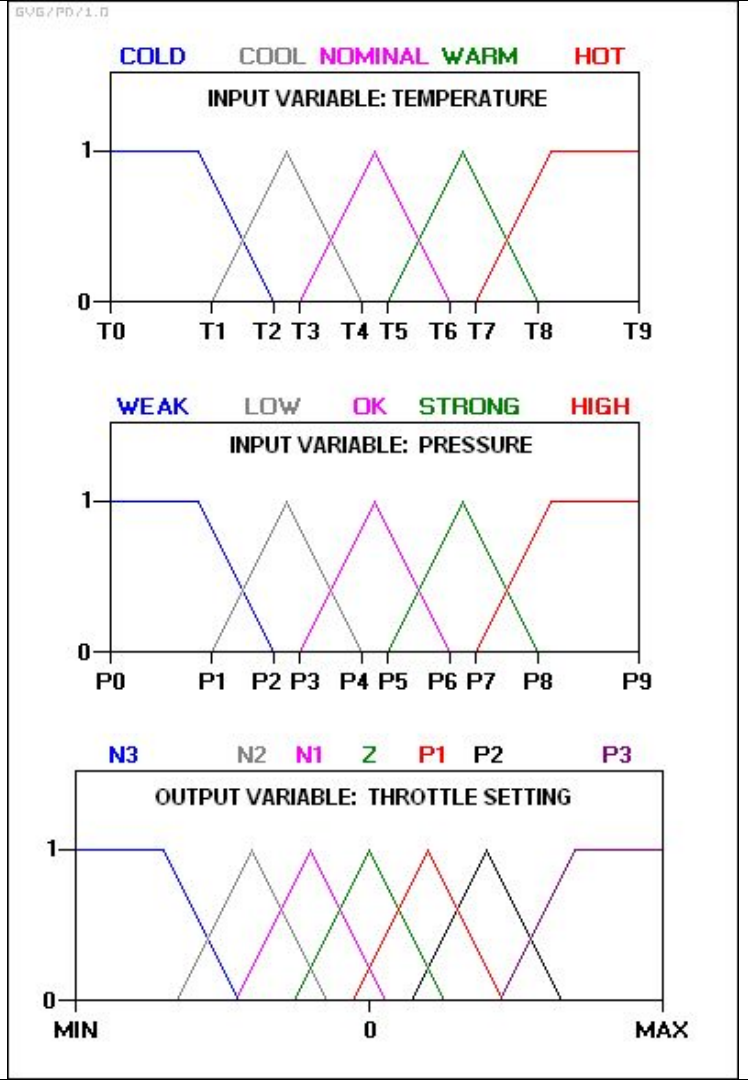
Soletta™

Código do *flow*


```
BobPresence(gpio/reader:pin=12) OUT -> IN_VAR ANN(machine-
learning/neural-network:initial_required_observations=10)

AlicePresence(gpio/reader:pin=13) OUT -> IN_VAR ANN

LightState(gpio/reader:pin=14) OUT -> IN LightStateTagger(machine-
learning/tagger:tag=LightState) OUT -> OUT_VAR ANN

Timer(timer:interval=100) OUT -> TRIGGER ANN

ANN OUT -> IN _(machine-learning/filter:tag=LightState) OUT -> IN
LightController(gpio/writer:pin=15)
```

SML - *Fuzzy*

- Usado quando há vários graus de certeza para um determinado variável.
- Termos.
- Regras de produção.


COLD
COOL
NOMINAL
WARM
HOT
INPUT VARIABLE: TEMPERATURE
1
0
T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
WEAK
LOW
OK
STRONG
HIGH
INPUT VARIABLE: PRESSURE
1
0
P0
P1
P2
P3
P4
P5
P6
P7
P8
P9
N3
N2
N1
Z
P1
P2
P3
OUTPUT VARIABLE: THROTTLE SETTING
1
0
MIN
0
MAX

SML - Fuzzy

- IF temperature IS cool AND pressure IS weak,THEN throttle is P3.
- IF temperature IS cool AND pressure IS low, THEN throttle is P2.
- IF temperature IS cool AND pressure IS ok, THEN throttle is Z.
- IF temperature IS cool AND pressure IS strong, THEN throttle is N2.

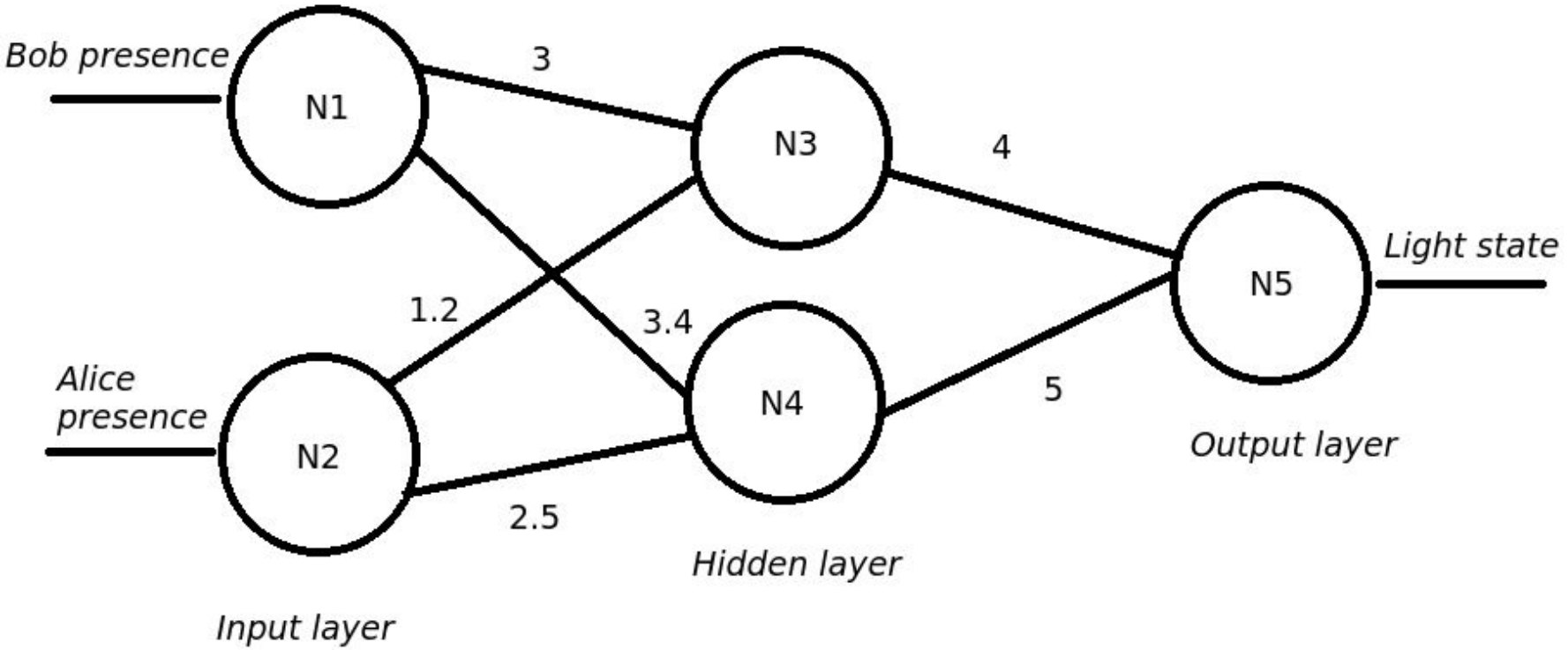
SML - Fuzzy

- Regras de produção são criadas automaticamente.
- Os termos são criados pelo usuário ou automaticamente.

SML - Rede neural

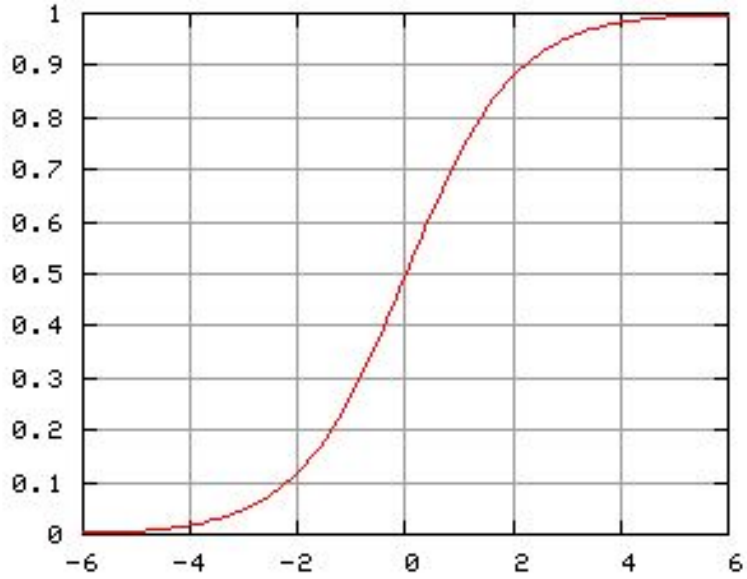
- Aproximador de função.
- *Feedforward neural network.*

SML - Rede neural FeedFoward



SML - Rede neural

- Cada neurônio tem uma função de ativação.
- Sigmoide é a mais comum.



SML - Rede neural

- Treinamento utilizando retropropagação.
  - Dados são coletados.
  - Pesos inicializados de forma randomica.
  - Dados de entrada alimentação a rede neural para obter um valor de saída.
  - Calcule o erro da rede
  - Calcule o erro da *hidden layer* para *output layer*
  - Calcule o erro da *input layer* para *hidden layer*
  - Atualize os pesos

Dificuldades

- Validar os resultados é difícil.
- Como criar as simulações?
- Até que ponto as simulações são realistas?
- Consequências de predições erradas.

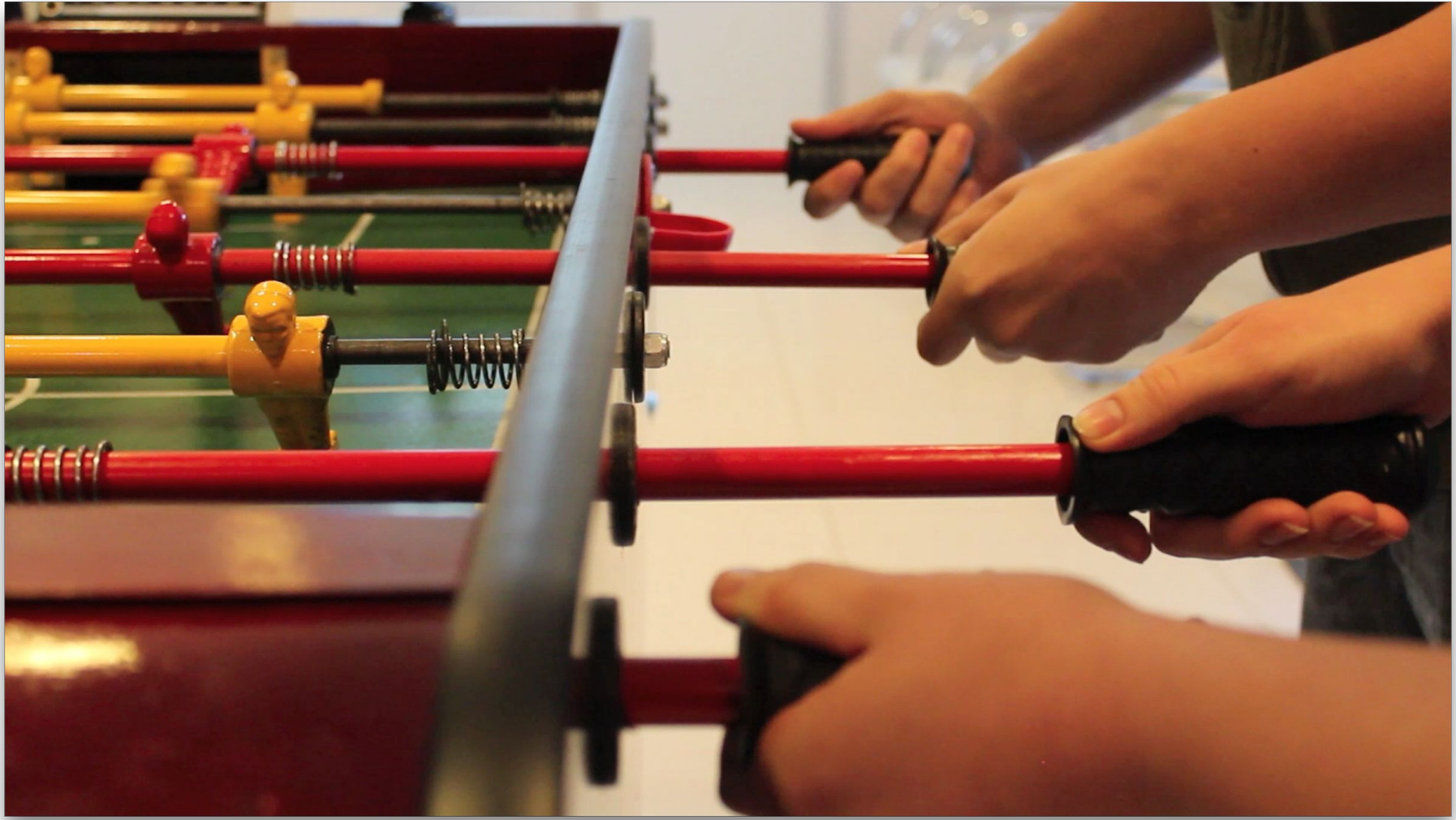
Protótipos

- Controle lâmpadas.
- Predição do resultado de jogos de pebolim.
- Controle de irrigação.

Soletta™

Controle de lâmpada

- Cor da lâmpada é ajustada de acordo com o usuário.
- Cor única quando os dois usuários estão na sala.
- Primeiro teste com dados reais.
- Ajudou a encontrar *bugs*.



Mesa de pebolim inteligente

- Jogadores e placar final de jogos de pebolim coletados por mesa de pebolim (no protótipo os resultados foram coletados manualmente)
- 366 jogos registrados
  - Algoritmo *Fuzzy* - predição em 40% dos jogos - 75% de acerto
  - Algoritmo Redes Neurais - predição em 99% dos jogos - 62% de acerto

Soletta™

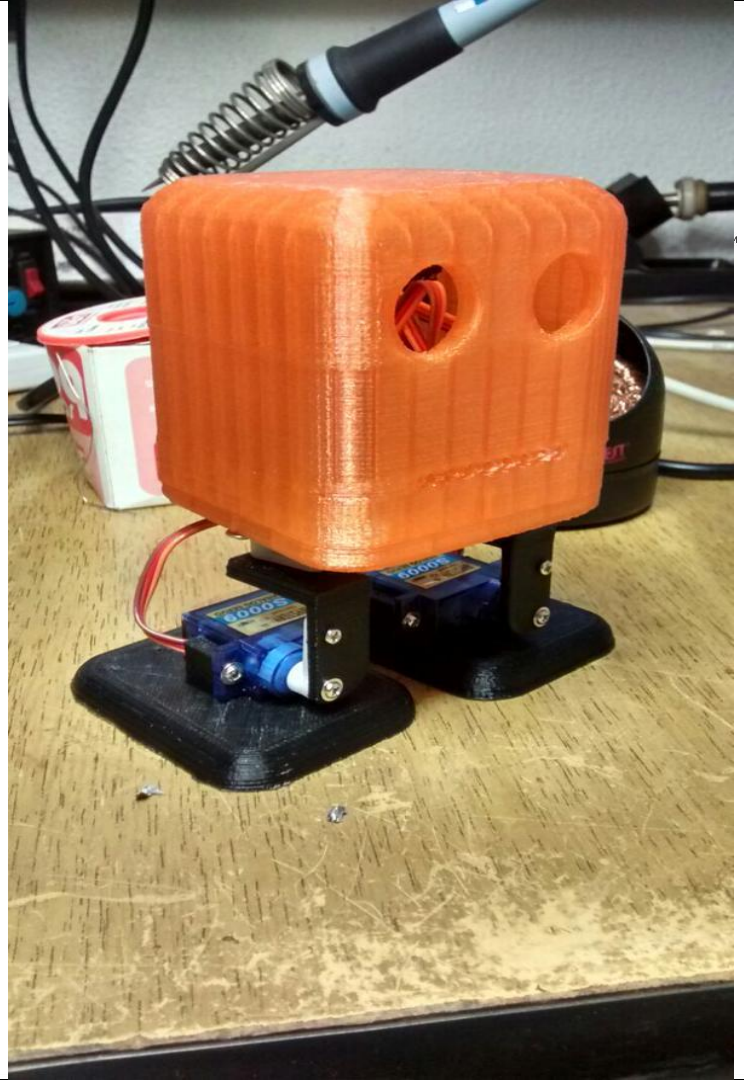
Controle de irrigação

- Duas plantas - cada irrigador utilizava um algoritmo diferente.
- *Flower power* era usado para medir a umidade do solo.
- Informações eram coletadas através de uma API *Web*.
- Solo era regado pelo usuário quando umidade <= 25 %.

Controle de irrigação - resultados e problemas

- Resultados
  - Rede neural aprendeu que a planta deveria ser irrigada por volta de 25% de umidade com apenas um *input* do usuário.
  - *Fuzzy* apenas regava em situações muito específicas.
- Problema
  - Não era possível “dizer” ao SML que a planta estava sendo irrigada demais.

Robótica com Soletta? O pequeno Bob



Contatos

IRC: #soletta @ irc.freenode.net

Listas de email: http://lists.solettaproject.org

Repositorios: https://github.com/solettaproject/

Wiki: https://github.com/solettaproject/soletta/wiki

bruno.dilly@intel.com

leandro.maciel.dorileo@intel.com