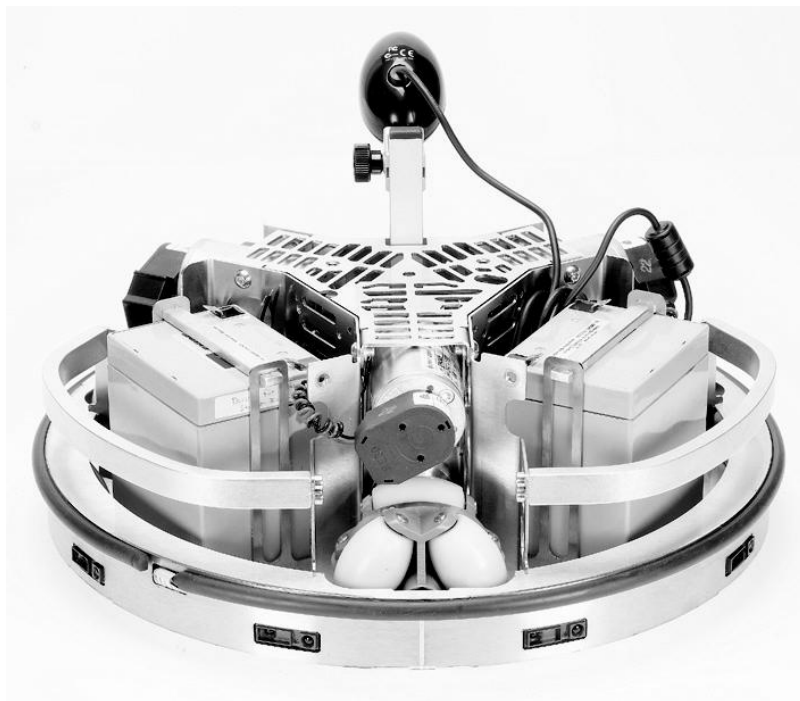




Robótica Móvel - Robotino®

Robotino® - Hardware Modularidade

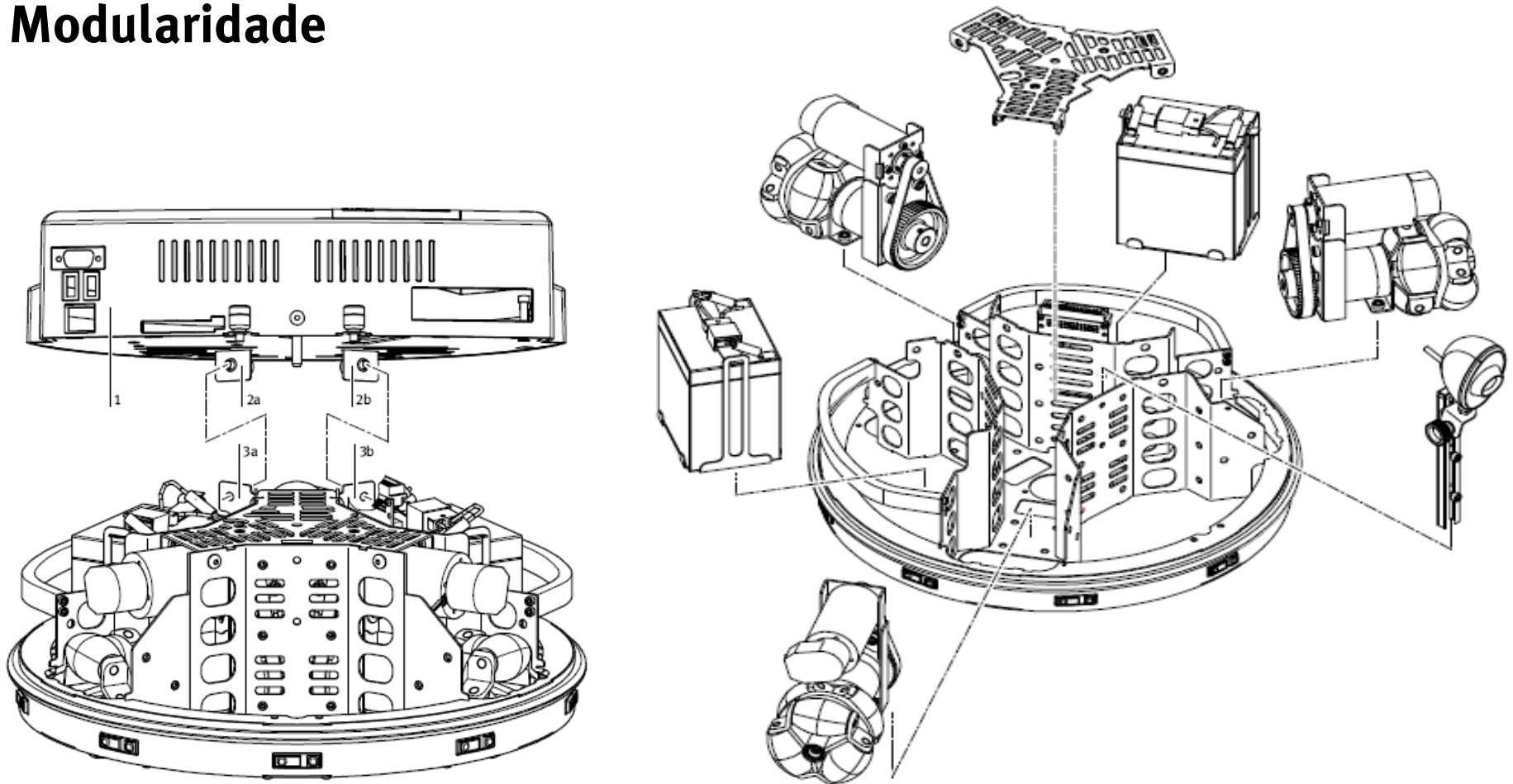


Estrutura Mecânica

- Rodas Omnidirecionais;
- 3 motores DC com redutores;
- Mobilidade em qualquer direção;
- Chassis em aço com barra protetora de borracha;
- Fácil acesso às baterias;
- Fácil acesso à rodas, motores e transmissão;

Sistema de robótica móvel da FESTO voltado para aprendizado, treinamento e pesquisa.

Robotino® - Hardware Modularidade



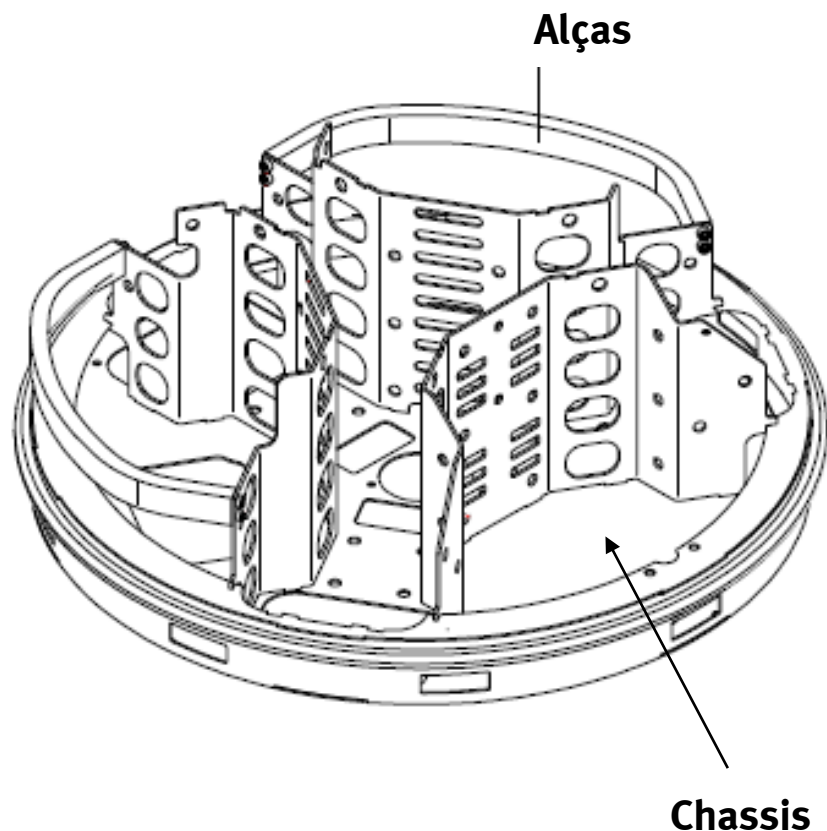
Robotino® - Hardware



Dimensões:

- Diâmetro: 370 mm
- Altura: 210 mm (sem câmera)
- Peso: aprox. 11 kg
- Componentes industriais
- Conexão à CPU através de conector único

Robotino® - Hardware: Chassis

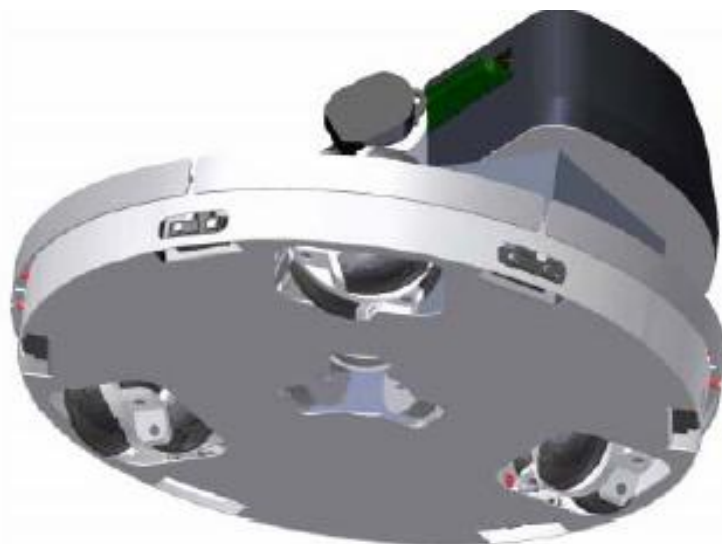


- Plataforma de aço inoxidável.
- Moldado a laser.
- Acomoda as baterias, motores, sensores de distância e sensor anti-colisão.
- Espaço adicional para montagem de outros acessórios, sensores ou atuadores.

Robotino® - Hardware: Módulos de Acionamento



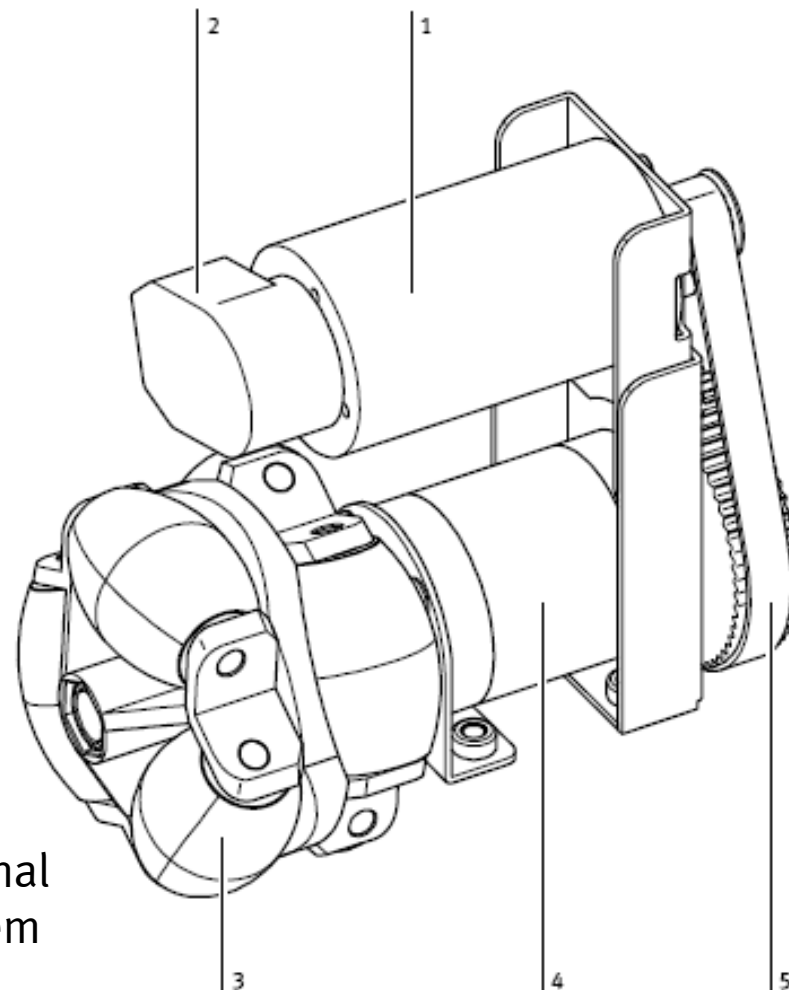
- Motor DC (motor de corrente contínua)
- Sistema de transmissão com relação de 16:1
- Rodas Omnidirecionais e correia dentada
- Encoder incremental



Robotino® - Hardware: Módulos de Acionamento

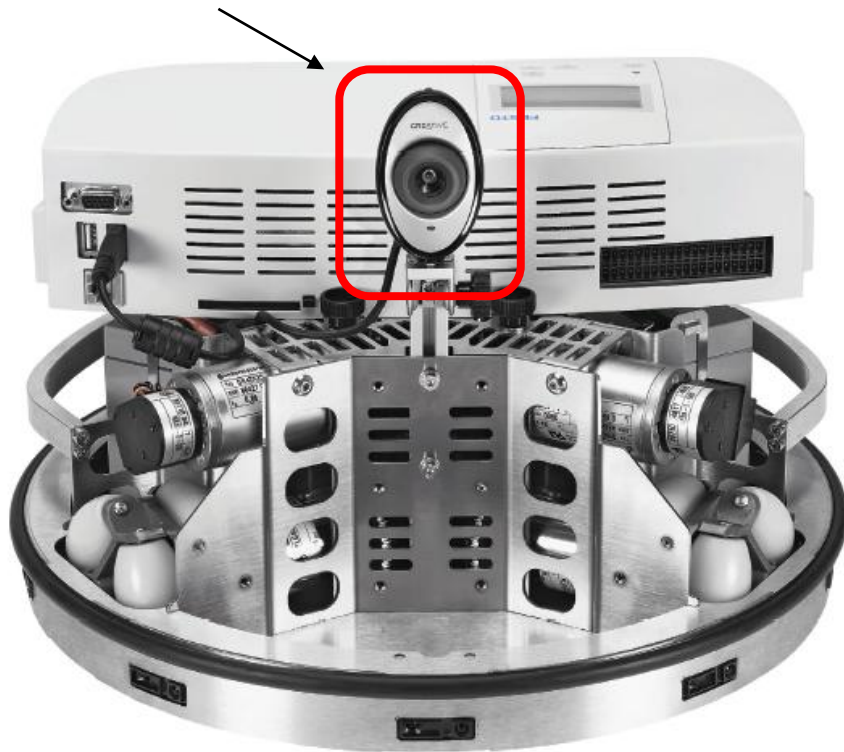
- Motor DC
- Voltagem nominal 24 VDC
- Velocidade nominal 3600 RPM
- Torque nominal 3,8 Ncm 3.8
- Corrente nominal 0,9 A
- Starting torque Ncm 20
- Velocidade sem carga 4200 RPM

- 1 – Motor DC
- 2 – Encoder
- 3 – Rodas Omnidirecional
- 4 – Caixa de engrenagem
- 5 – Correia dentada



Robotino® - Hardware: Câmera

Câmera



- Altura e inclinação ajustáveis.
- Conexão USB.
- Exibição de imagens ao vivo.
- Tratamento de imagens (*Robotino® View*).
- Identificar objetos.
- Rastreamento de caminhos.

Robotino® - Hardware: Unidade de Controle

Unidade de Controle

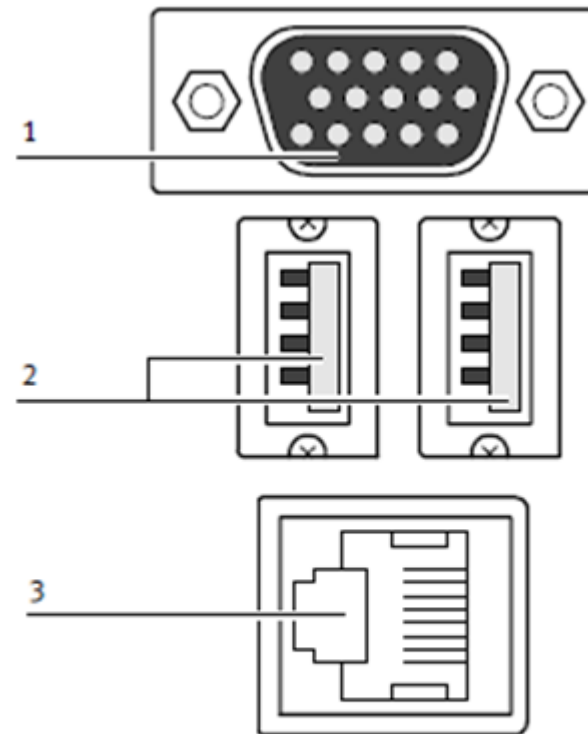


- Processador PC104 300MHz- OS Linux
- Cartão de memória flash 256MB
- Ponto de acesso a rede wireless LAN

Robotino® - Hardware: Unidade de Controle



Interfaces



VGA socket connector (1)

USB ports 1 and 2 (2)

Ethernet interface (3)

Robotino® - Hardware: Placa de I/Os

Entradas e Saídas
Digitais e Analógicas

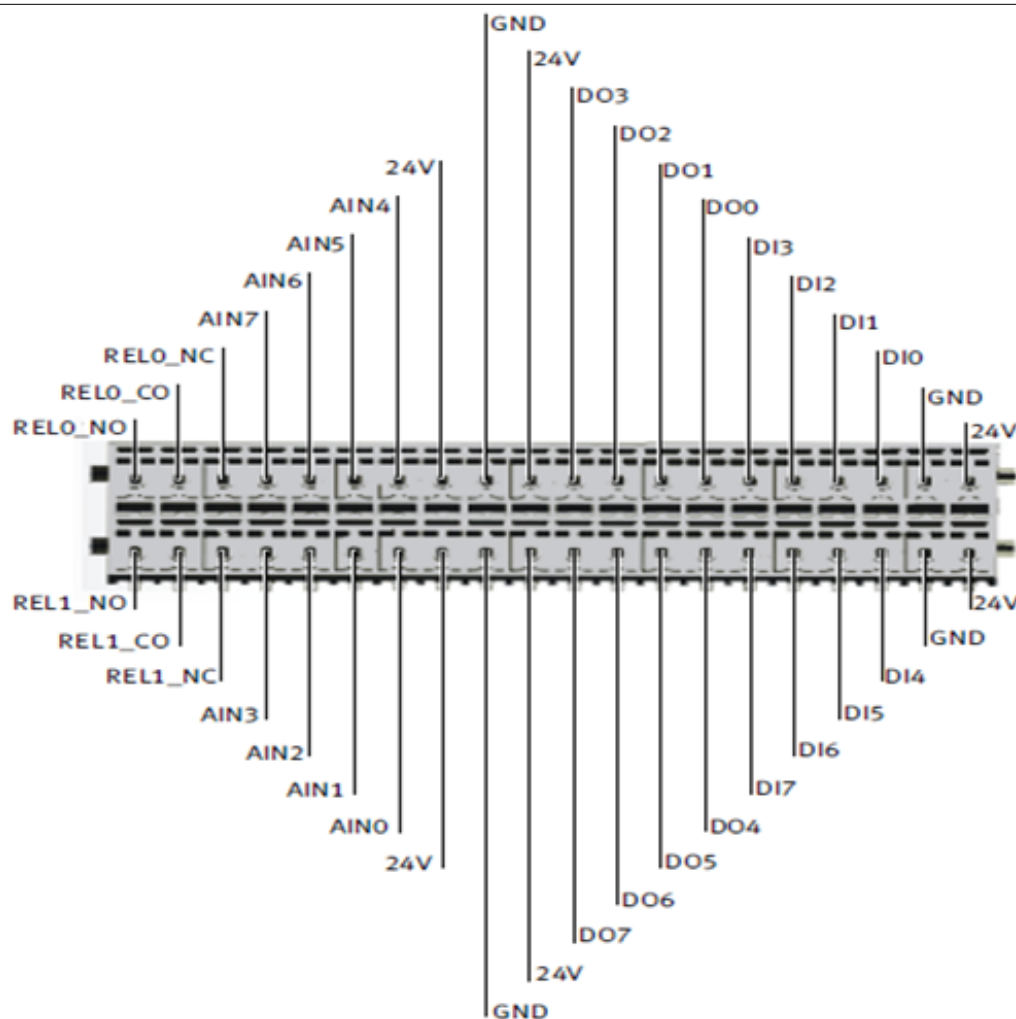


Interface entre a Unidade de Controle e:

- Sensores.
- Motores.
- Demais dispositivos externos.

Robotino® - Hardware: Placa de I/Os

- 8 Entradas Analógicas (0-10V): AIN0 à AIN7.
- 8 Entradas Digitais: DI0 à DI7.
- 8 Saídas Digitais: DO0 à DO7.
- 2 Reles: RELO e REL1 (NA, NF ou Comutador).



Robotino® - Hardware: Baterias

- Duas baterias recarregáveis 12V.
- Fornecimento de 4Ah.
- Alojadas sobre o chassis.
- Duas baterias extras e carregador.



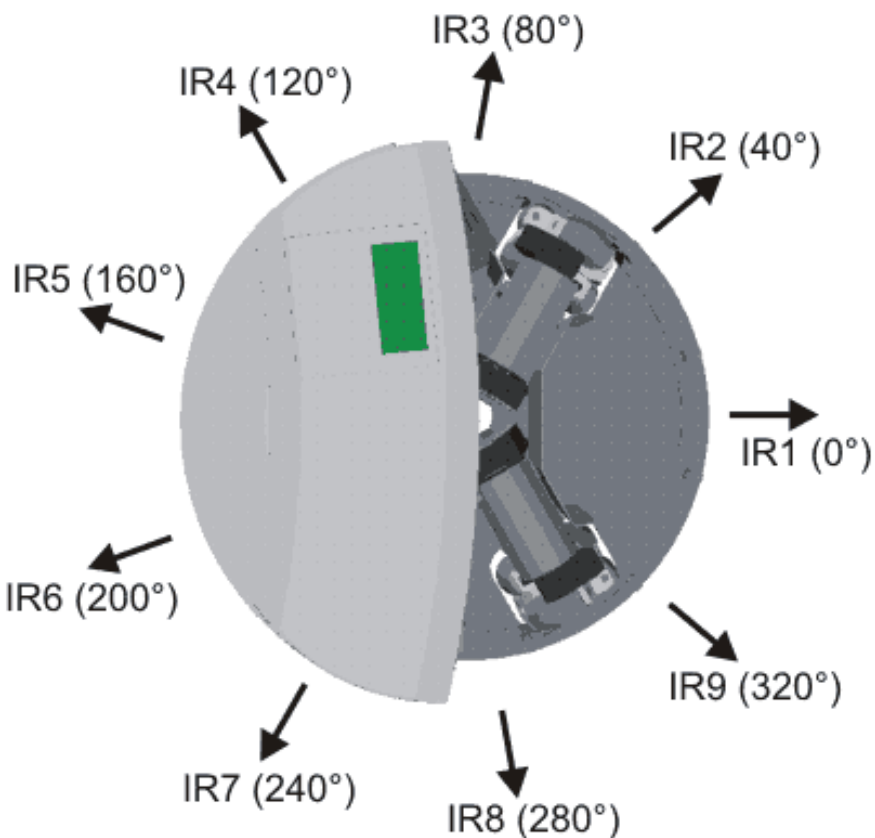
Robotino® - Hardware: Sensores



Sensores:

- Sensores infravermelhos para distância
- Encoder Incremental
- Sensor anti-colisão
- Sensor indutivo analógico
- Sensores ópticos digitais

Robotino® - Hardware: Sensores



Sensores Infravermelhos:

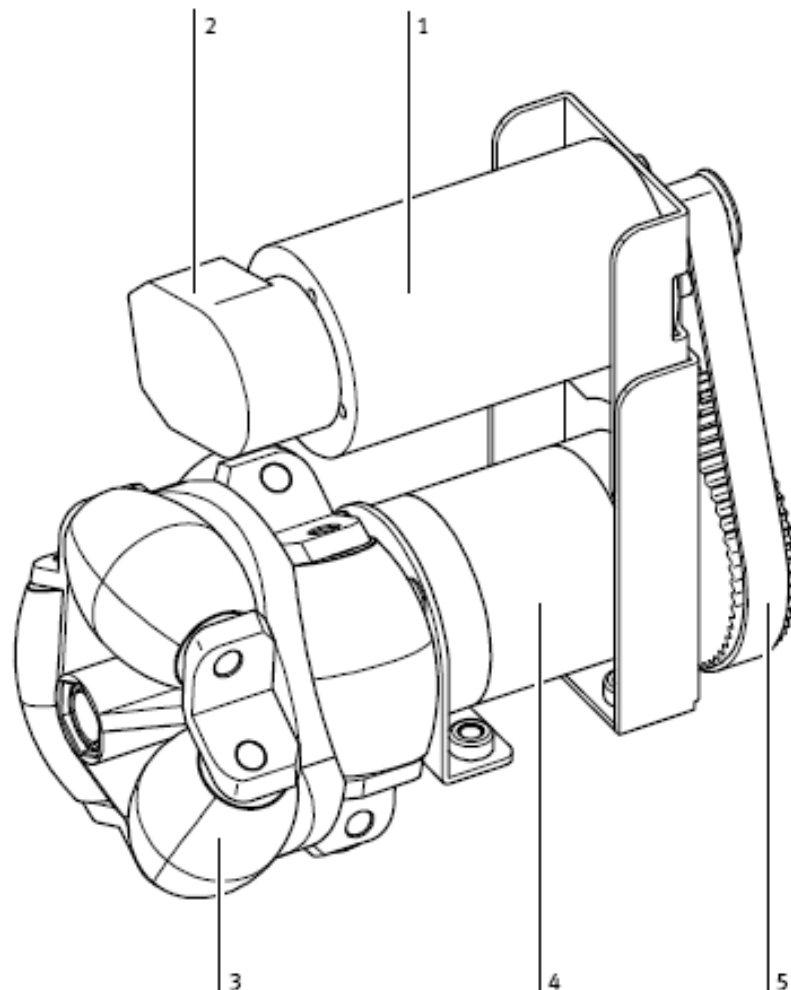
- IR1...IR9
- 40° de ângulo entre cada
- Medições de distâncias entre 4 a 30cm
- Converte um valor de distância em um sinal analógico de saída

Robotino® - Hardware: Sensores

Encoder Incremental

- Medição da velocidade dos motores (RPM)
- Utilizados para ajuste da velocidade do Robotino®

- 1 – Motor DC
- 2 – Encoder**
- 3 – Rodas Omnidirecionais
- 4 – Caixa de engrenagem
- 5 – Correia dentada



Robotino® - Hardware: Sensores

Sensor anti-colisão

- Preso ao redor de todo o Robotino®
- Comuta quando uma pressão é aplicada
- Detecção de colisão com objetos em quaisquer pontos da circunferência do Robotino®



Sensor anti-colisão

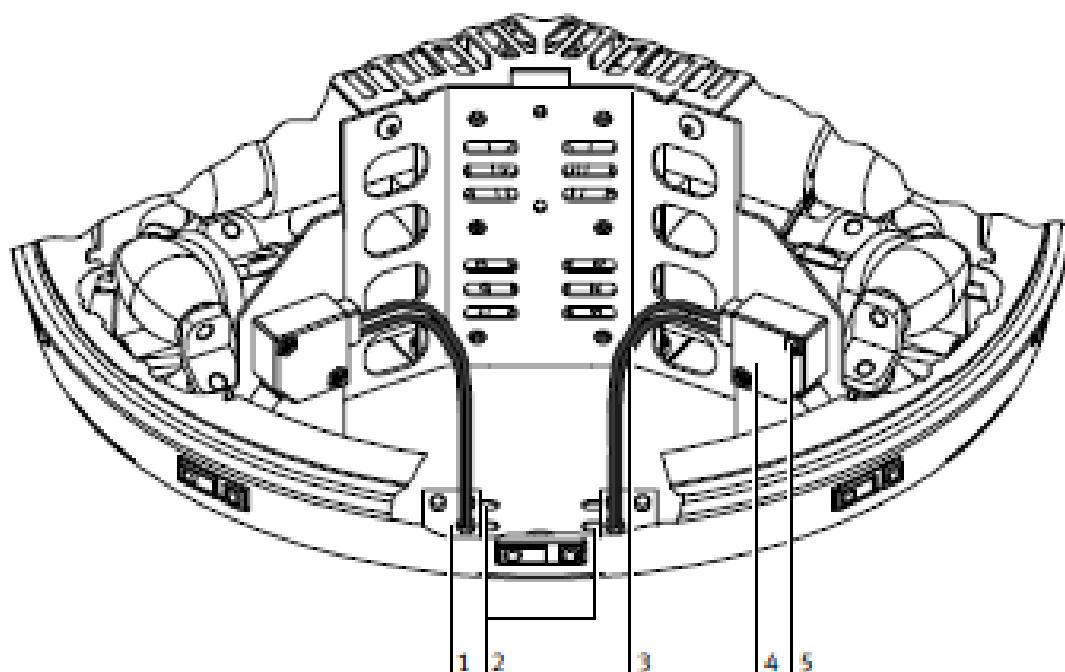
Robotino® - Hardware: Sensores



Sensor Indutivo Analógico

- Sinal de intensidade variável
- Implementação de controle de rota
- Deve ser preso ao chassis do Robotino®
- Conectado à placa de I/Os

Robotino® - Hardware: Sensores



Fibre-optic mounting bracket (1)

Mounting slots (2)

Fibre-optic cables (3)

Fibre-optic unit (4)

Mounting screws and holes (5)

Sensores Ópticos Digitais

- Reflexão Difusa
- Utilização de fibras óticas
- Devem ser presos ao chassis do Robotino®
- Conectados à placa de I/Os

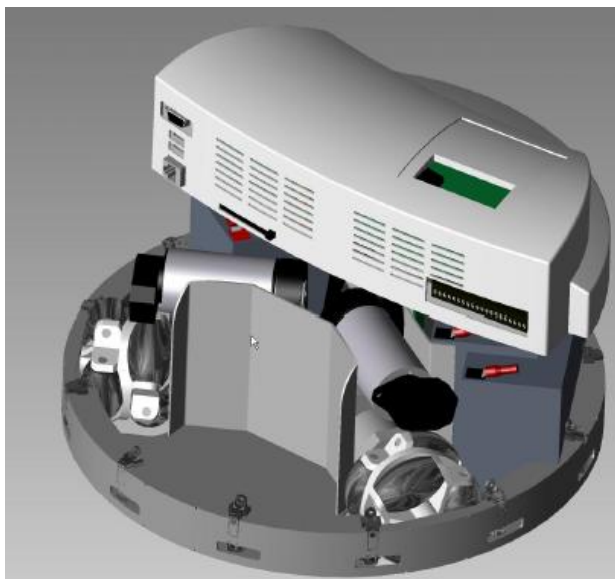
Robotino® - Hardware: Sensores



**Plataforma aberta para inserção
de outros sensores/dispositivos:**

- Sensores ultrassônicos
- Sensor de Navegação
- 8 entradas digitais
- 8 saídas digitais
- 8 entradas analógicas
- 2 relês

Robotino® - Hardware: Rede Wireless LAN

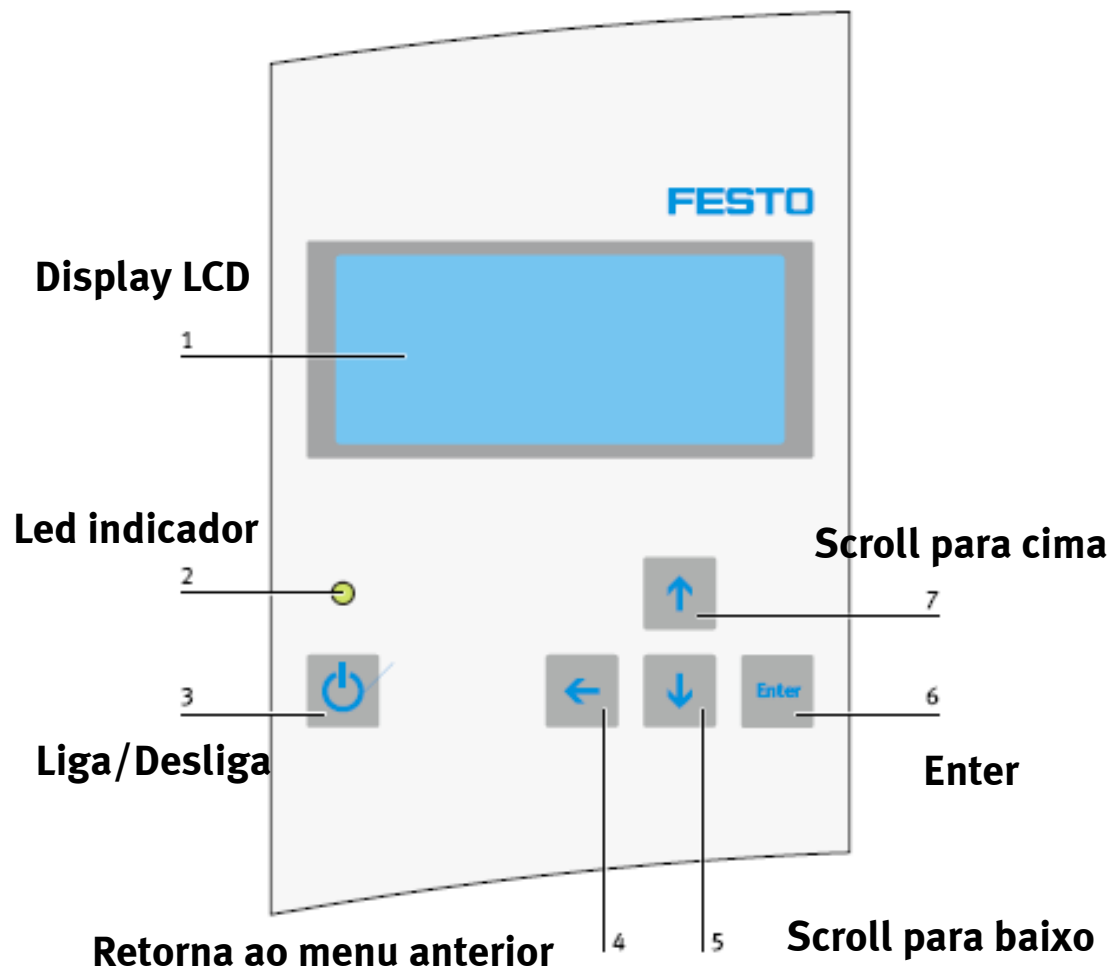


Comunicação através de um ponto de rede wireless:

- Transmissão de dados com um alto alcance de transmissão (até 100 m).
- Fácil e rápida configuração.
- Baixo consumo de corrente.
- Integração com o software *Robotino® View*.

Robotino® - Hardware: IHM e Teclado

- Inicialização do Robotino®.
- Seleção do idioma.
- Informações sobre o sistema
(ex: versão, IP ..)
- Estado das baterias.
- Configurações de rede.
- Seleção de aplicações DEMO.
- Entre outras..



Robotino® - Comissionamento:



Utilização da plataforma (base)

- Mantenha o Robotino® sobre a plataforma durante a realização de testes!
- Desta forma podem ser realizados diversos testes de funcionamento de sensores e motores sem o risco de colisões.
- Possibilidade de conexão ao carregador de baterias.

Robotino® - Comissionamento:

- Pressione o botão de liga-desliga até que o LED acenda.
- Duas barras serão mostradas em todo o display.
- Após aproximadamente 30 segundos a mensagem de início ira aparecer no display.
- A última barra indica o estado das baterias e informa qual é a versão do Robotino®.



Robotino® - Comissionamento:

Pressione a tecla ENTER para acessar o menu principal.

Por padrão o display utilizará o idioma inglês.

Este menu principal contem os seguintes itens:

- Languages
- State of charge
- DEMOs
- Network

Utilizando-se as teclas direcionais para cima e para baixo, você pode mover o sinalizador (▸) para cima ou para baixo.

Pressione o ENTER para acessar o item desejado.



Robotino® - Comissionamento:

- *Languages*: seleção do idioma do Robotino®
- *State of charge*

O display ira te informar a quantidade disponível de carga, a voltagem e corrente atuais das baterias.

- *DEMOS*

Círculo

Adiante

Quadrado

Itinerante

Seguir uma linha



Robotino® - Comissionamento:

- *Network*

Este menu apresenta o endereço IP atual, que pode ser alterado.

Os valores contidos no endereço IP podem ser alterados utilizando as setas direcionais para cima e para baixo.

Pressione ENTER no último dígito do IP para acessar a máscara de sub-rede.



Robotino® - Comissionamento:

Configurando a rede WLAN

- Ative sua WLAN. Suas configurações de WLAN devem permitir a atribuição de um endereço IP para o WLAN.
- Como precaução, posicione o Robotino® sobre a plataforma. Ligue o Robotino® e aguarde a finalização das configurações iniciais.
- Anote o IP que será mostrado no display.

Robotino® - Comissionamento:

Configurando a rede WLAN

- Inicie a busca de redes sem fio disponíveis. Uma rede com o nome **robotino x.x** deverá aparecer na lista das redes disponíveis.
- As seguintes configurações de rede devem ser realizadas para o correto funcionamento da WLAN:
 - Atribuição automática do código de rede (SSID)
 - Atribuição automática do endereço IP

Robotino®View – Ambiente Gráfico de Programação

Introdução:

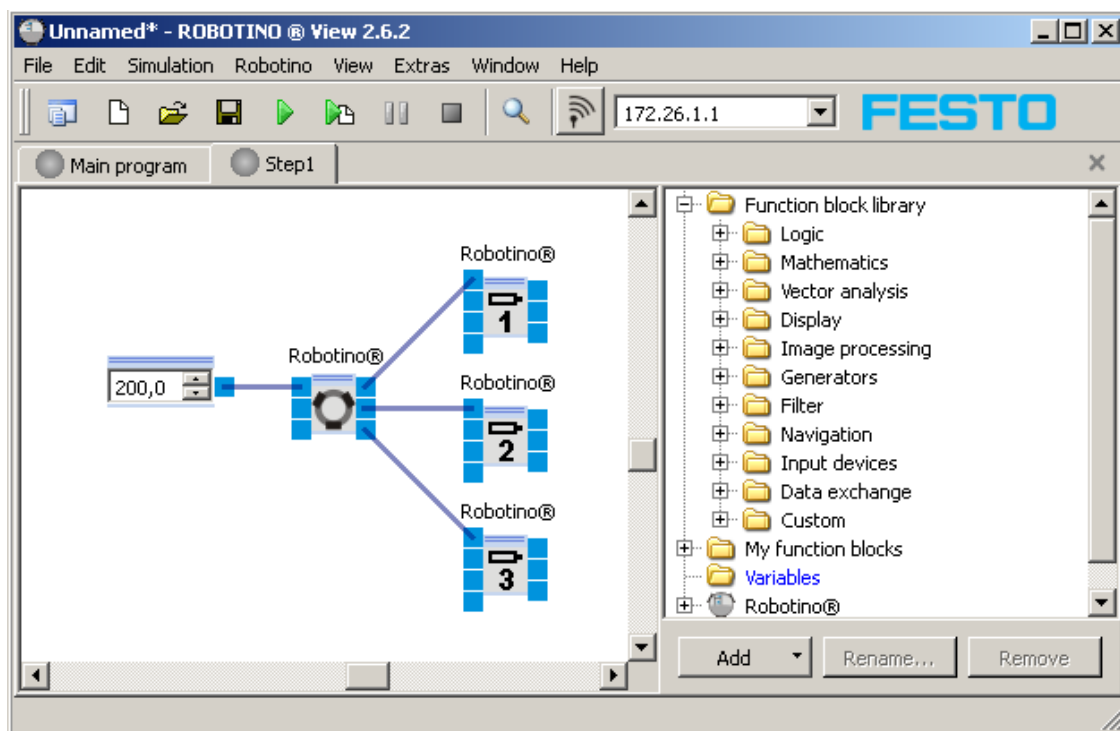
- Software gráfico de programação;
- Programação gráfica / sequencial;
- Criar e executar diversas aplicações;
- Estabelecer a comunicação através da WLAN;



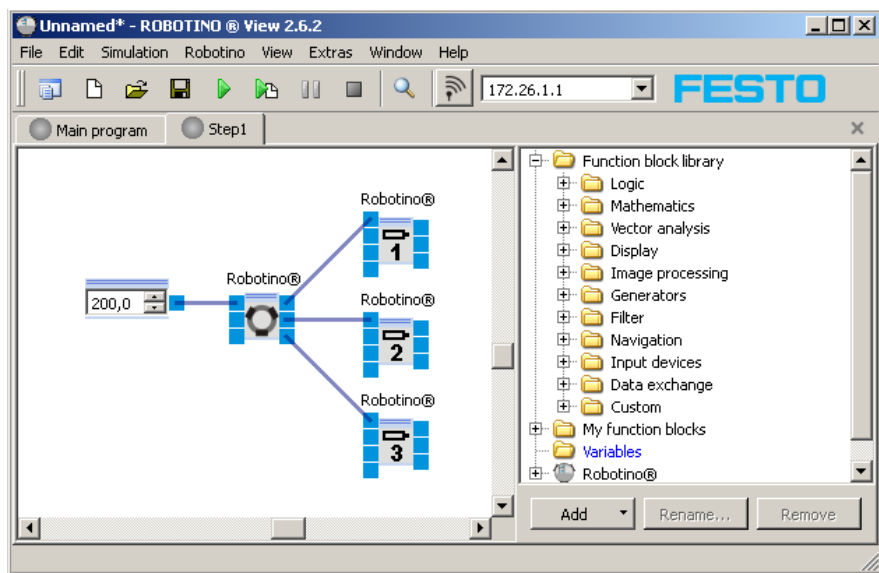
Robotino®View – Ambiente Gráfico de Programação

Programação simplificada:

- Selecione os blocos de função da extensa biblioteca interativa;
- Conexão através de interfaces de entradas e saída bem definidas;
- Ambiente gráfico amigável;
- Permite o ajuste dos parâmetros dos blocos de função, de acordo com a aplicação;



Robotino®View – Ambiente Gráfico de Programação



Rede Wireless



comunicação

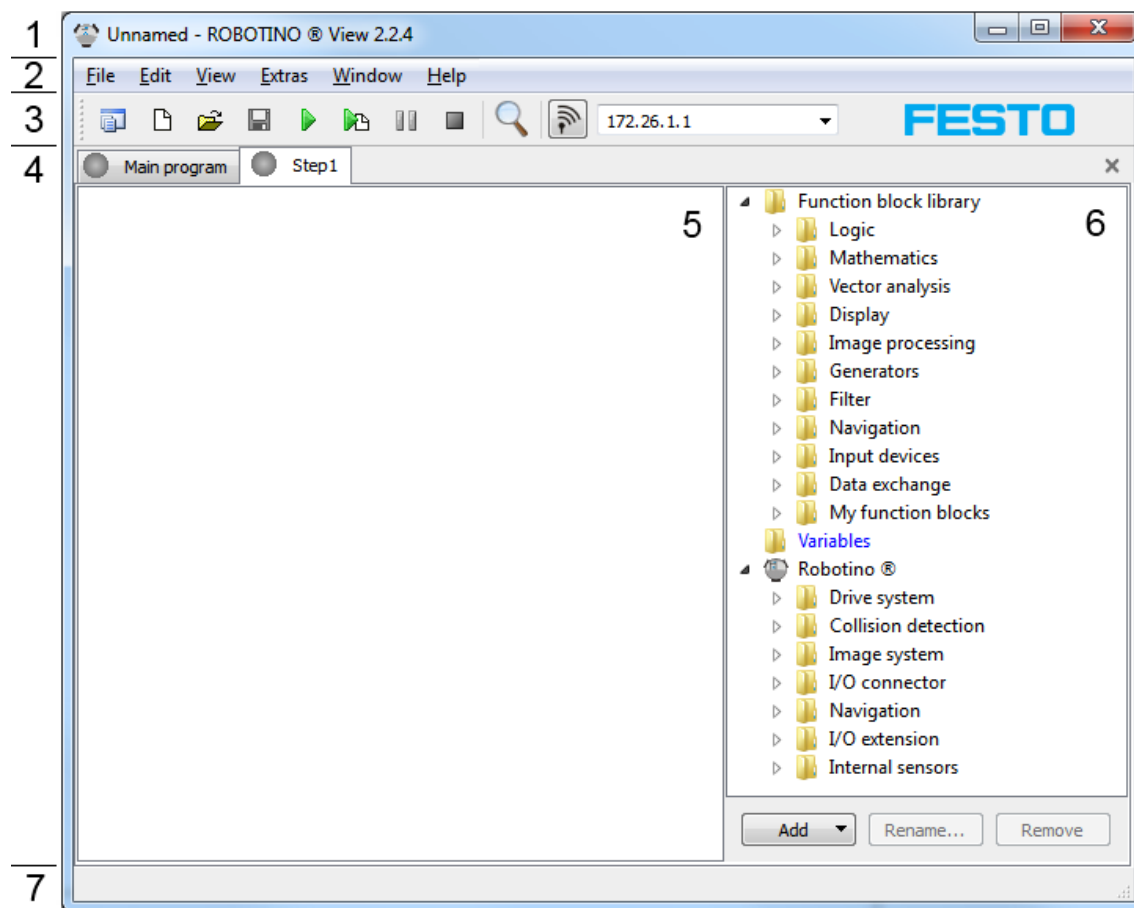


- Não necessita de download

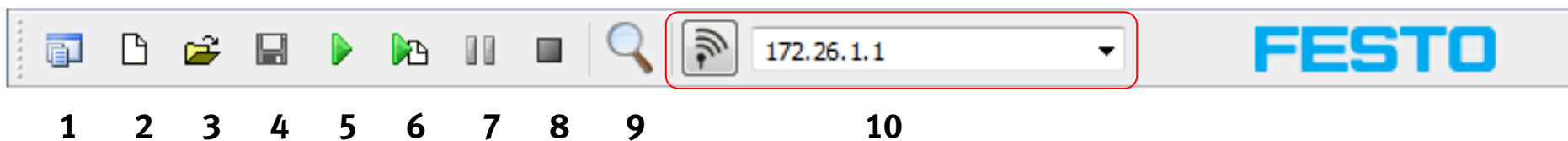
Integração com qualquer PC que possua cartão de rede Wireless

Robotino®View – Ambiente Gráfico de Programação

1. Barra de Títulos
2. Barra de Menus
3. Barra de Ferramentas
4. Seletor de Programas
5. Área de Trabalho
6. *Function Block Library*
7. Barra de Status



Robotino®View – Ambiente Gráfico de Programação

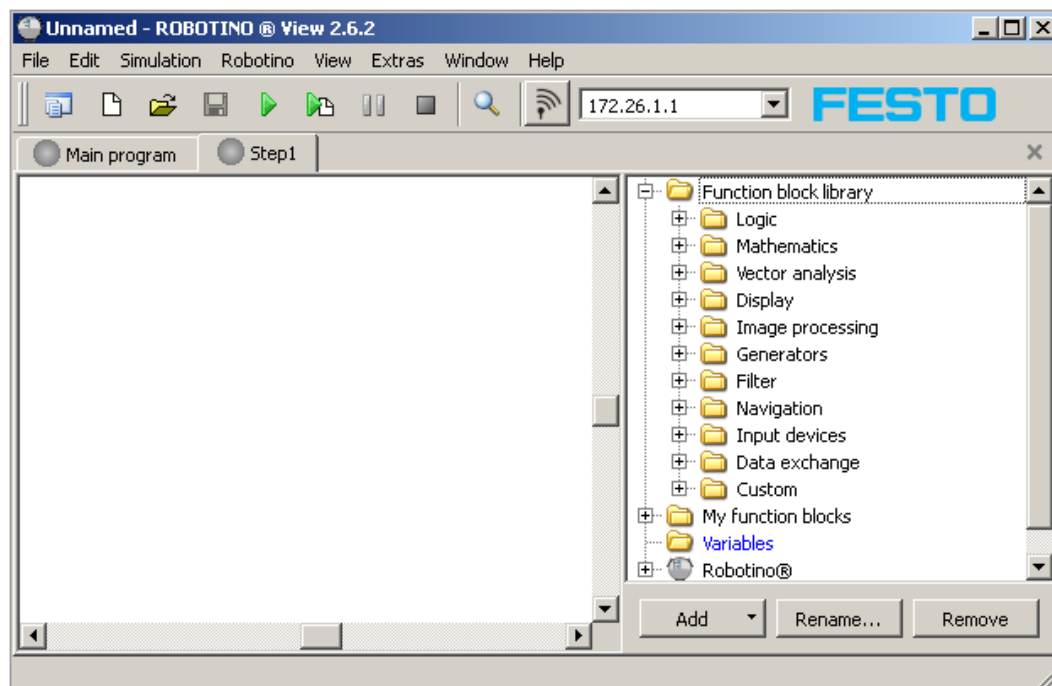


- 1 – Criar novo Projeto
- 2 – Criar novo sub-programa
- 3 – Abrir um Projeto já existente
- 4 – Salvar o Projeto atual
- 5 – Rodar o programa principal
- 6 – Rodar o sub-programa em questão
- 7 – Pausar a simulação
- 8 – Parar a simulação
- 9 – Zoom
- 10 – Fornecer IP e estabelecer conexão

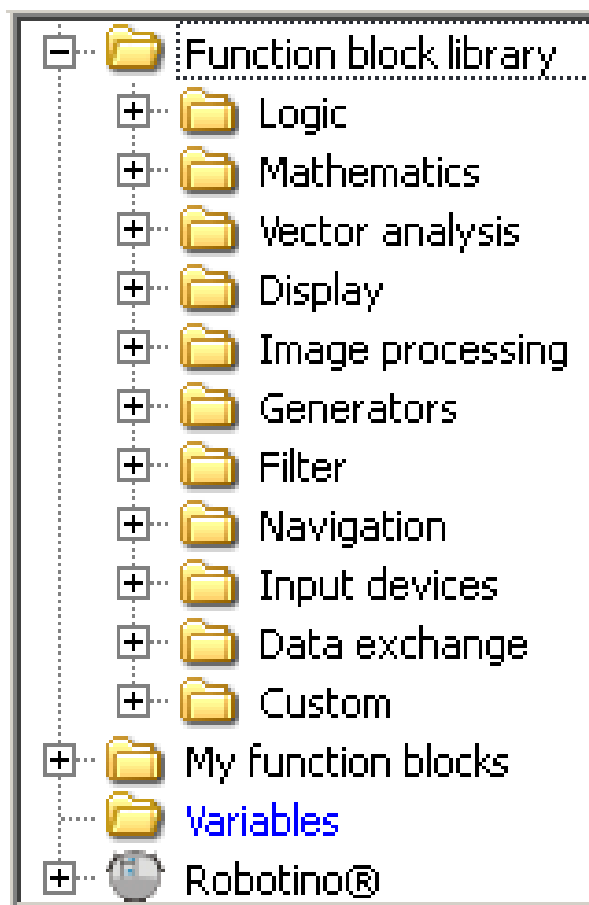
Robotino®View – Ambiente Gráfico de Programação

Novo Programa – Step1:

Ao abrir o *Robotino® View* um programa em branco é aberto na tela e pode-se iniciar a programação.



Robotino®View – Ambiente Gráfico de Programação



Robotino®View – Ambiente Gráfico de Programação

Function Block Library

- Contém toda a extensa biblioteca de funções disponíveis para programação do Robotino®.

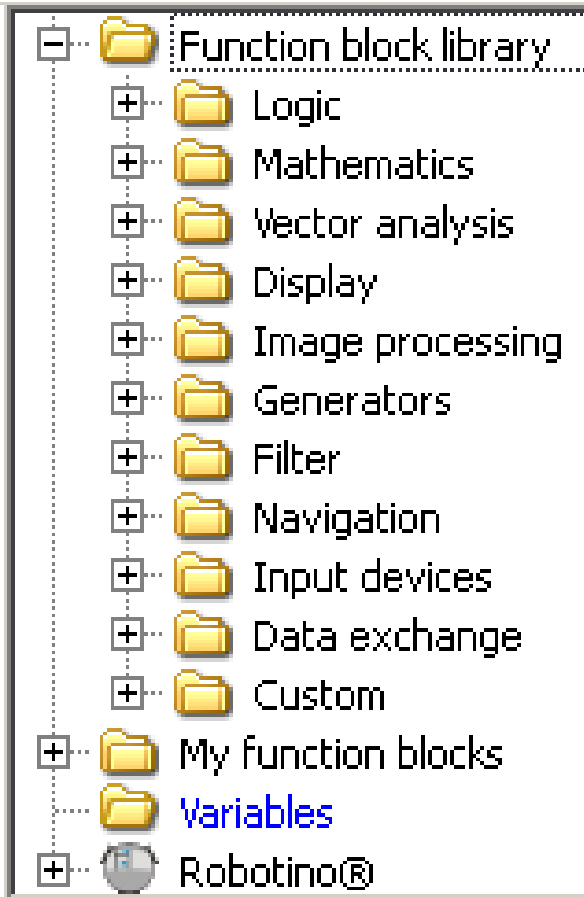
Variables

- Contém os blocos de função responsáveis por escrever e ler variáveis globais.

Robotino

- Contém os blocos de função relacionados com os dispositivos de hardware do Robotino®.

Robotino®View – Ambiente Gráfico de Programação



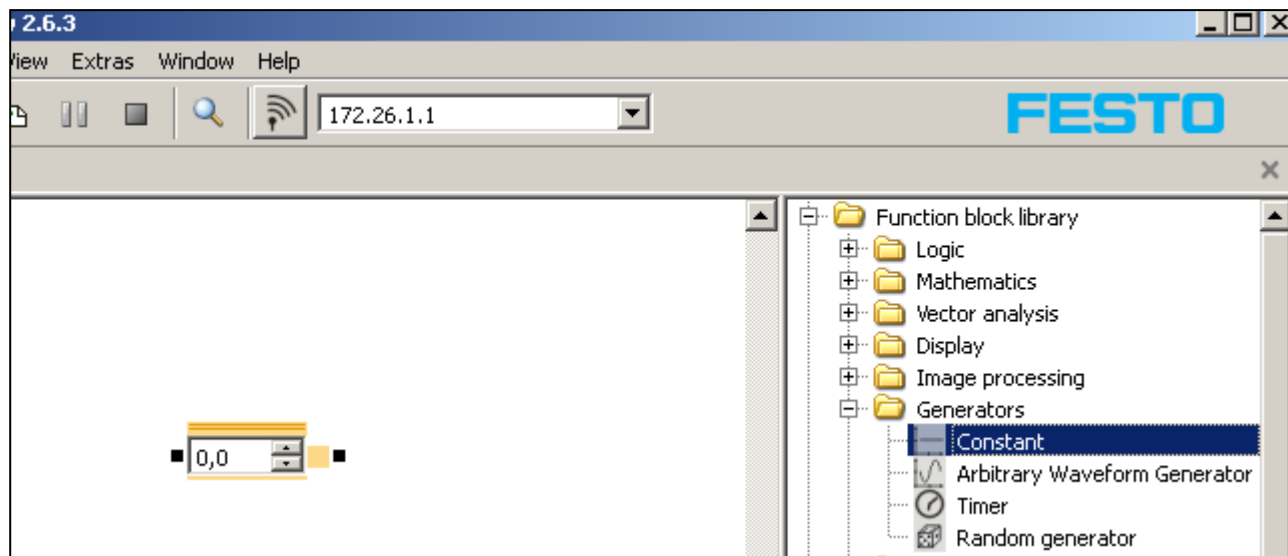
The screenshot displays the 'Function block library' tree in the Robotino®View software. The tree is organized into a hierarchical structure with folders and sub-items. Each folder is represented by a yellow folder icon, and each sub-item is represented by a plus sign icon. The sub-items are listed to the right of the folders, with a corresponding description in Portuguese.

Folder	Sub-item	Description
Function block library	Logic	-> Funções Lógicas (AND, OR ...)
	Mathematics	-> Funções Matemáticas (+, - ...)
	Vector analysis	-> Operações Com Vetores
	Display	-> Gráficos
	Image processing	-> Processamento de Imagens
	Generators	-> Geradores de Sinal
	Filter	-> Filtros de Sinal
	Navigation	-> Navegação (Atingir pontos específicos)
	Input devices	-> Dispositivos de Controle (Joystick ...)
	Data exchange	-> Troca de sinais com outros dispositivos
	Custom	-> Scrip (Lua)
My function blocks		-> Blocos de função criados pelo usuário
	Variables	-> Variáveis do sistema
Robotino®		-> Hardware (Sensores, Câmera ...)

Robotino®View – Ambiente Gráfico de Programação

Exemplo - Inserindo os blocos:

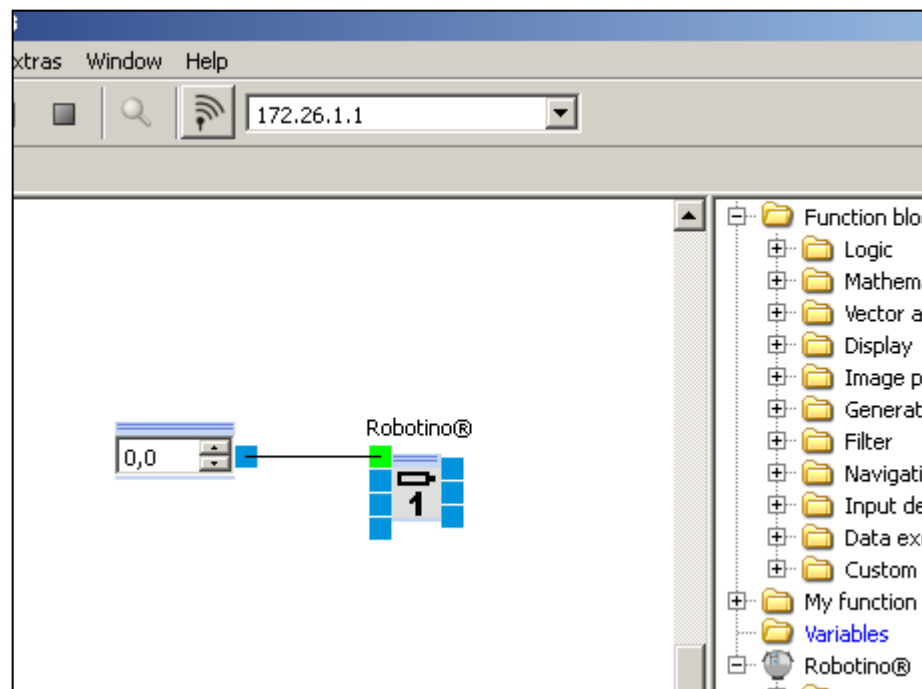
Arraste um bloco de interesse para a área de trabalho!



Robotino®View – Ambiente Gráfico de Programação

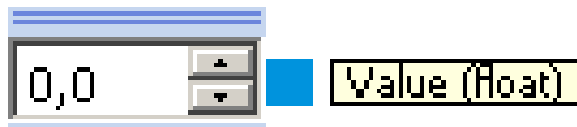
Exemplo - Realizando as interligações:

Clique com o botão esquerdo do mouse na conexão de origem e depois na conexão de destino:



Robotino®View – Funções Principais

Generators: Possui os blocos de função para gerar sinais diversos.



- **Constant:** Gera um sinal de valor constante
 - **Saída:** valor constante definido pelo usuário



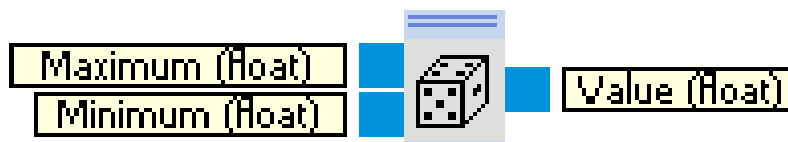
- **Arbitrary Waveform Generator:** Gerar formas de onda ajustáveis.
 - **Saída:** Forma de onda desejável

Robotino®View – Funções Principais

Generators: Possui os blocos de função para gerar sinais diversos.



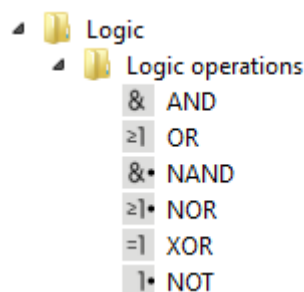
- **Timer**
 - **Saída:** Tempo em milissegundos
 - **Entrada:** Reset



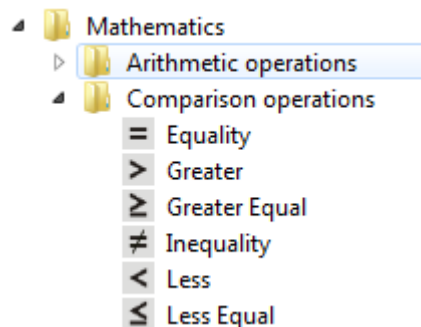
- **Random Generator:** Cria números aleatórios dentro de uma faixa especificada.
 - **Entradas:** Máximo e mínimo da faixa aleatória
 - **Saída:** Número aleatório compreendido entre a faixa.

Robotino®View – Funções Principais

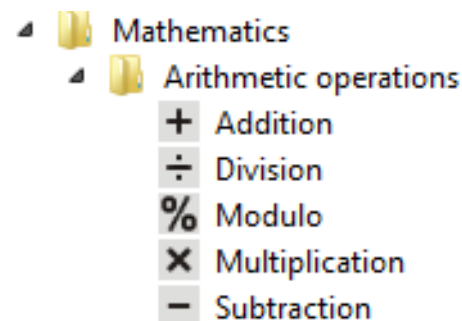
- Blocos lógicos: OU, E, NÃO, XOR, NÃO-E, NÃO-OU



- Operações de comparação: igual, maior, maior ou igual, menor, menor ou igual, diferente

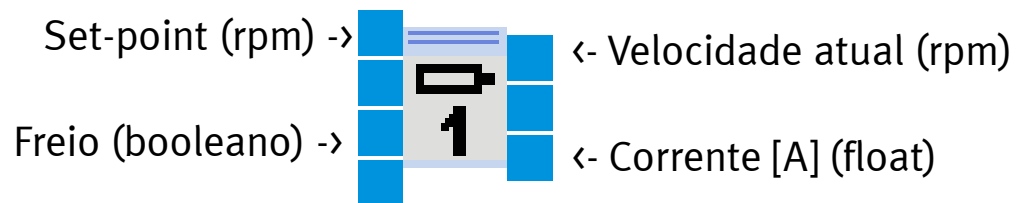
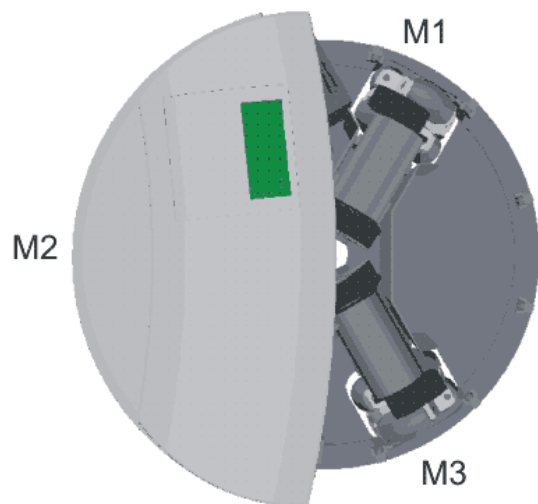


- Operações aritméticas: soma, divisão, multiplicação, subtração, módulo



Robotino®View – Funções Principais

Locomoção:

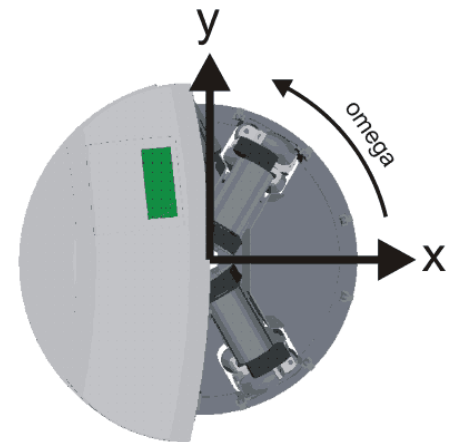


Robotino®View – Funções Principais

Locomoção:

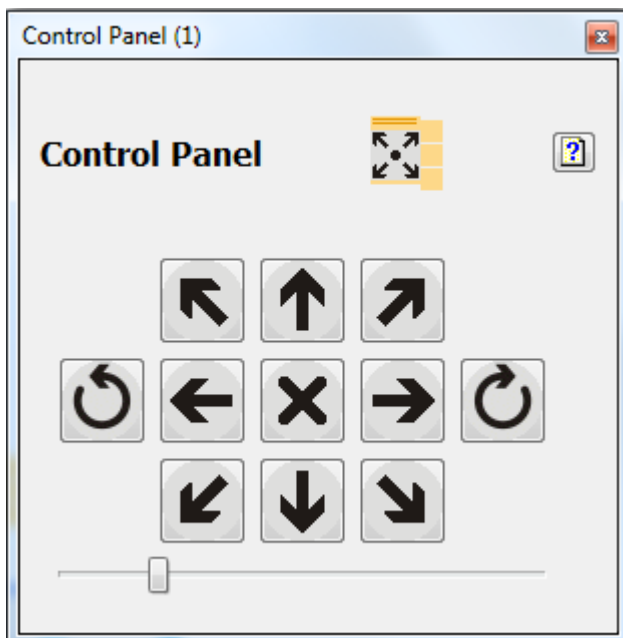
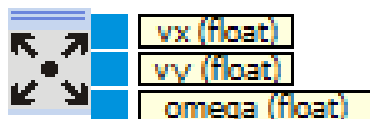


OMNIDRIVE

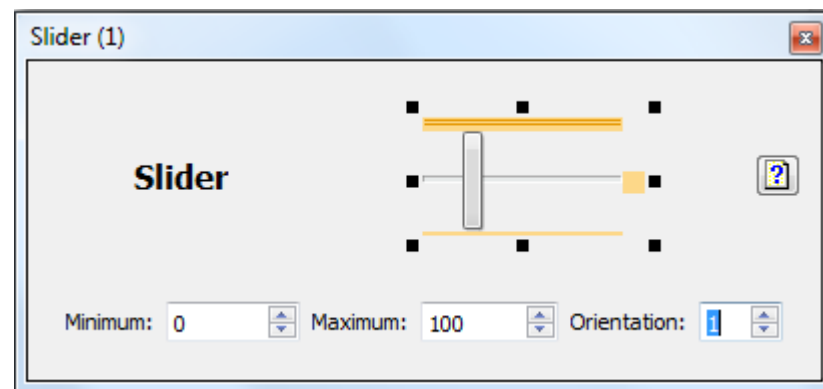


Robotino®View – Funções Principais

Locomoção: Control Panel



Slider



Robotino®View – Funções Principais

Hardware anti-colisão:

Robotino®



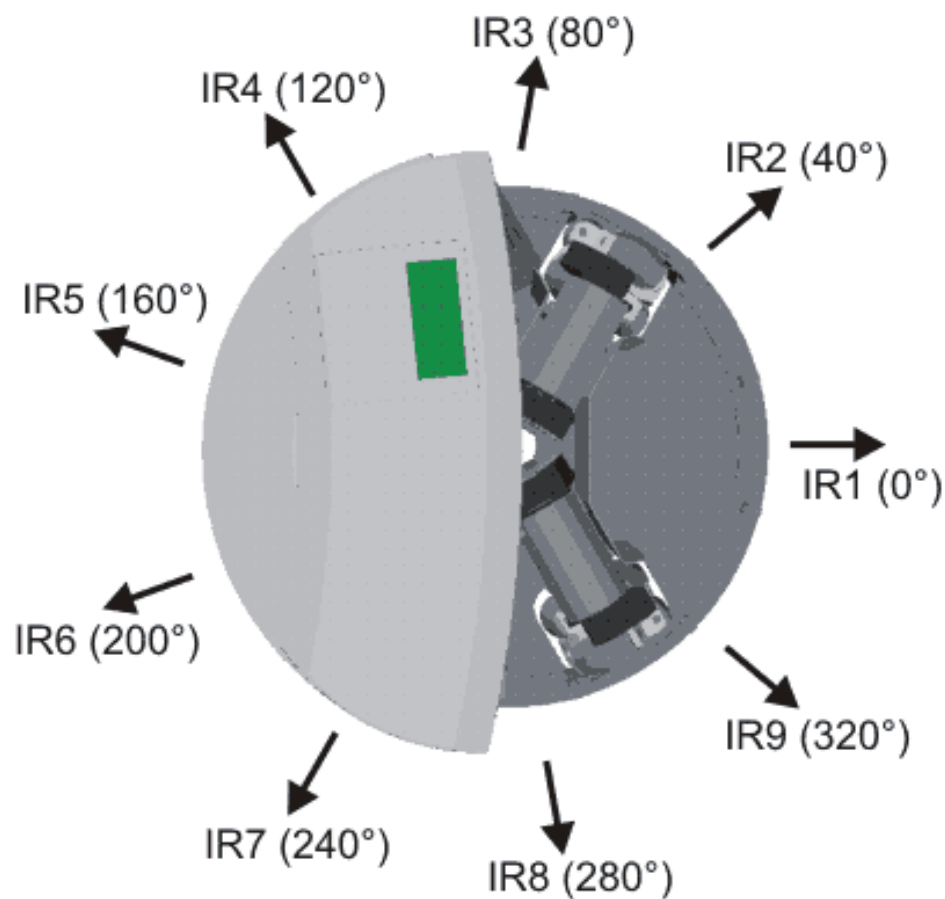
-> Valor (V)

-> Ângulo (deg)

Distance

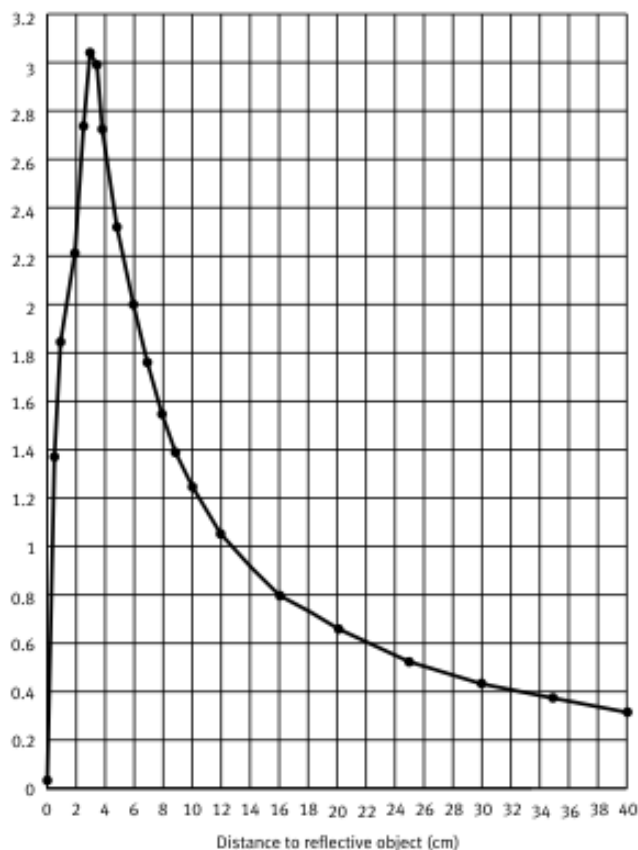
*Detecção de 4 a 30 cm

*Saída 0 a 2,55 V



Robotino®View – Funções Principais

Hardware anti-colisão:



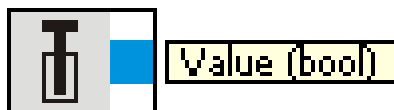
Curva característica dos sensores IR – tensão de saída x distância (cm)

Robotino®View – Funções Principais

Hardware anti-colisão:

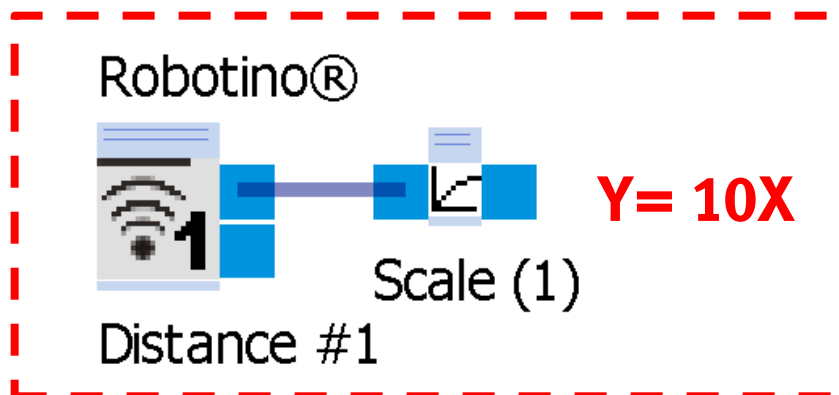
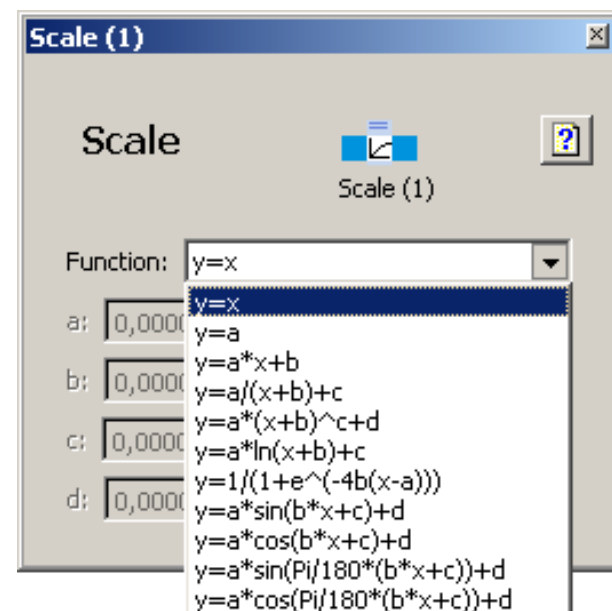
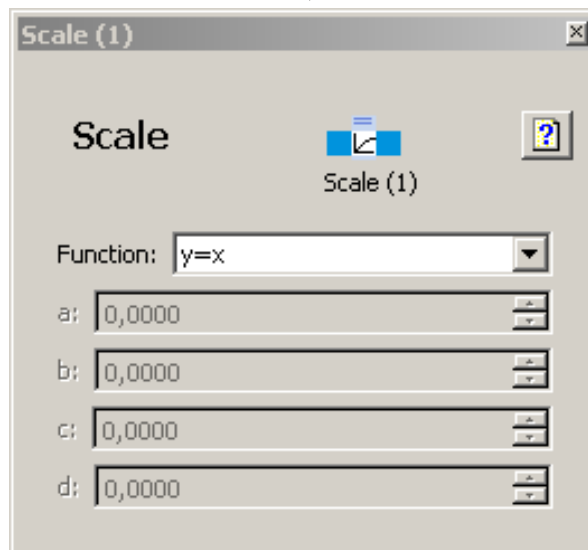
Bumper – um sensor de toque é integrado em toda a extensão do Robotino®. Caso seja tocado, um sinal é enviado.

Robotino ®



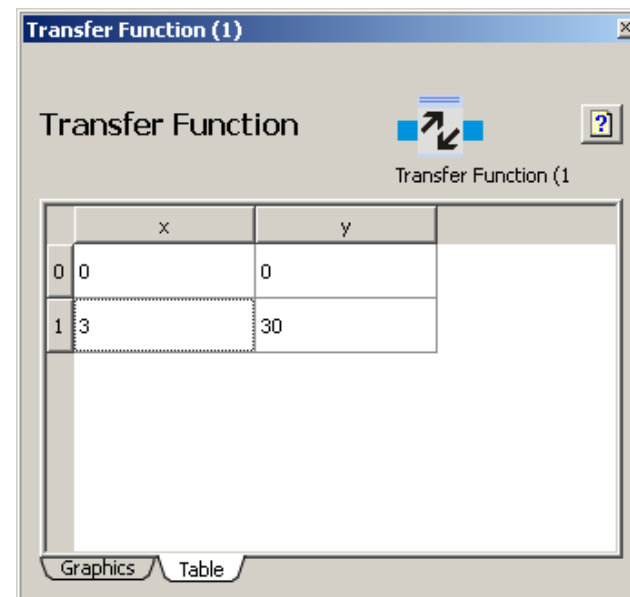
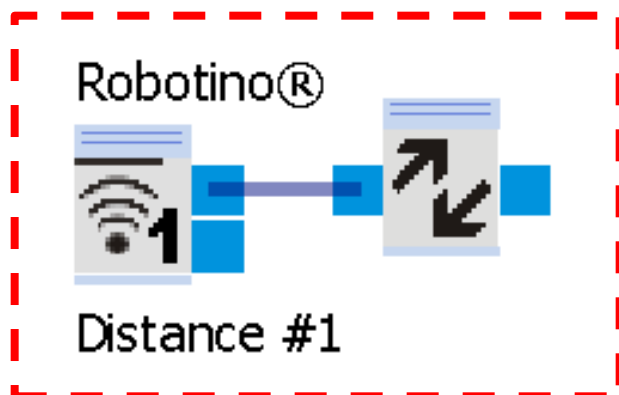
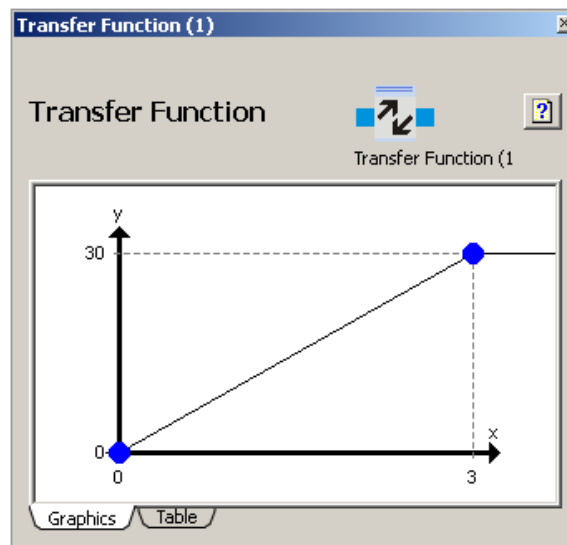
Robotino®View – Funções Principais

Mathematics - Functions:

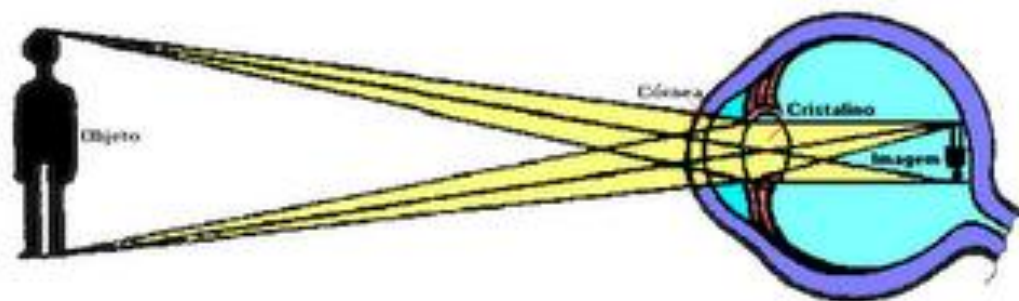


Robotino®View – Funções Principais

Mathematics - Functions:



Robótica Móvel - Imagem



A imagem no olho humano é formada na retina, onde se encontram os cones e bastonetes, responsável pela percepção das cores e luminosidade.

O sinal é então enviado ao cérebro, que interpreta esses sinais e forma a imagem.

No olho humano, a percepção das cores se divide basicamente em três tipos diferentes: vermelho, verde e azul.

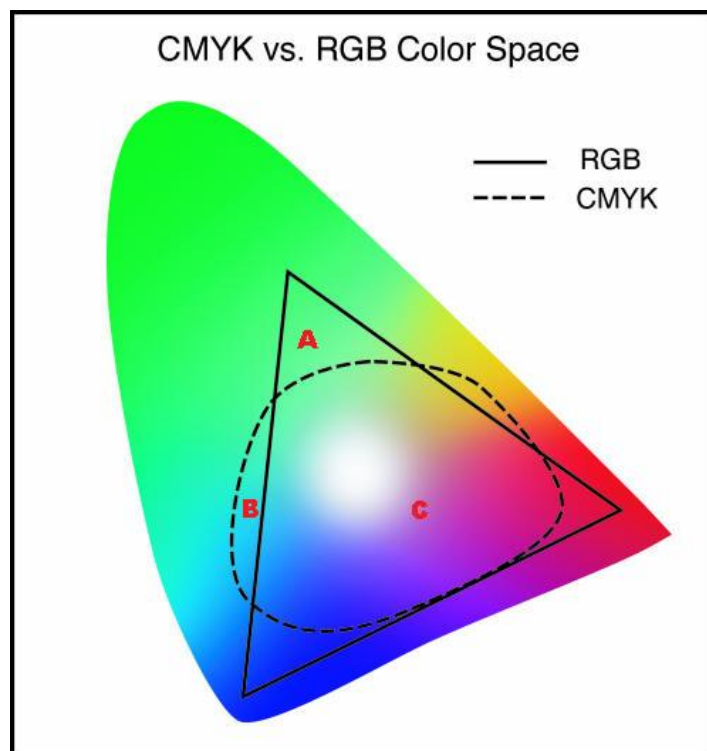
Robótica Móvel - Imagem

A imagem digital é formada por pequenos pontos, denominados pixels. Quanto maior o número de pixels, maior será a imagem.

A quantidade de pixels numa área define a resolução da imagem, logo, quanto mais pixels uma área contiver, maior será a sua resolução.



Robótica Móvel - Imagem



A informação da cor numa imagem digital depende de cada equipamento, ou seja, da forma como ele armazena os valores de cores individuais.

O sistema RGB é o mais utilizado, por ser parecido com o modo como o ser humano capta a imagem.

A formação da cor depende de fatores como iluminação e superfície do objeto.

Robótica Móvel - Imagem

Câmera do Robotino® : Características:



- ✓ Imagem de até 1280 x 960 pixels
- ✓ Vídeo de até 640 x 480 pixels (30 fps)
- ✓ Conexão USB
- ✓ Sensores CMOS-VGA

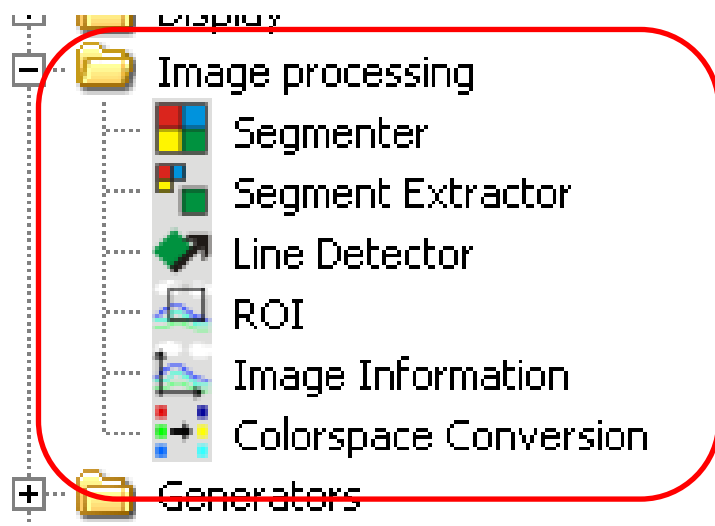
Robotino®View – Tratamento de Imagens

- O *Tratamento e Processamento de Imagens* é um processo para se obter informações e editar imagens digitais.
- Objetivos: Atingir, através de manipulação das imagens, certas especificações e características, de tal maneira que sejam obtidas informações corretas para tomada de decisões nos sistemas robóticos.
- Exemplos: Rastreamento de um objeto em determinada região evitando colisões, definição de rota ao longo de linhas com determinadas cores, etc.

Robotino®View – Tratamento de Imagens

Image Processing:

Contém os blocos de função que implementam o tratamento de imagens do *Robotino® View*.

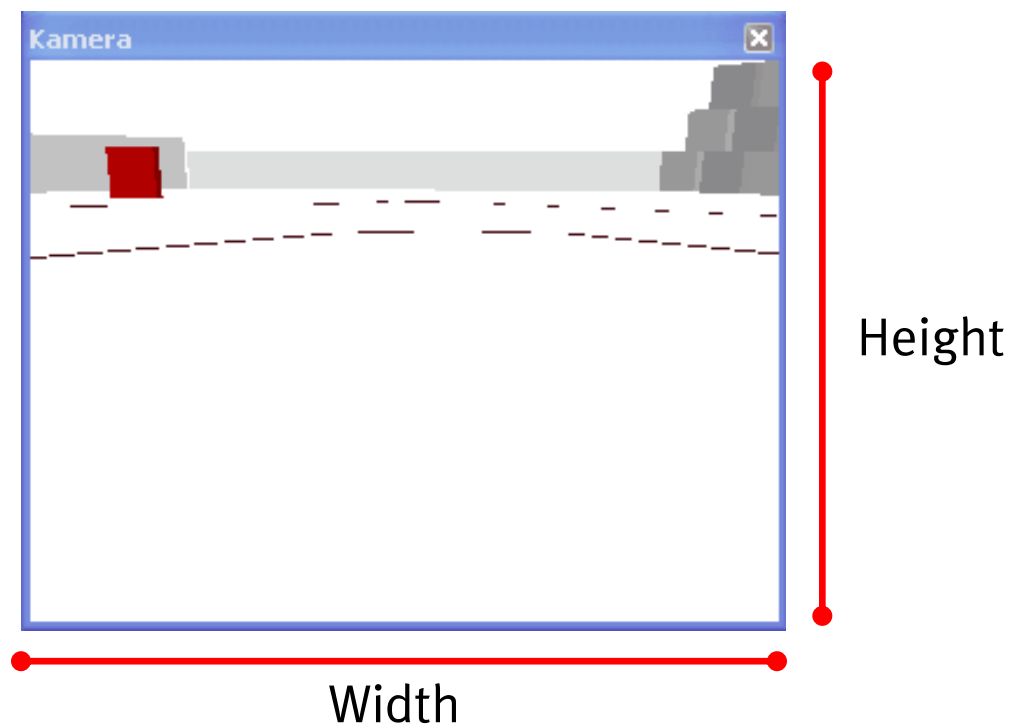


Robotino®View – Tratamento de Imagens

Image Processing:



Image Information



Bloco de função que fornece as dimensões (largura e altura) de determinada imagem, em número de *Pixels*

Robotino®View – Tratamento de Imagens

Image Processing:

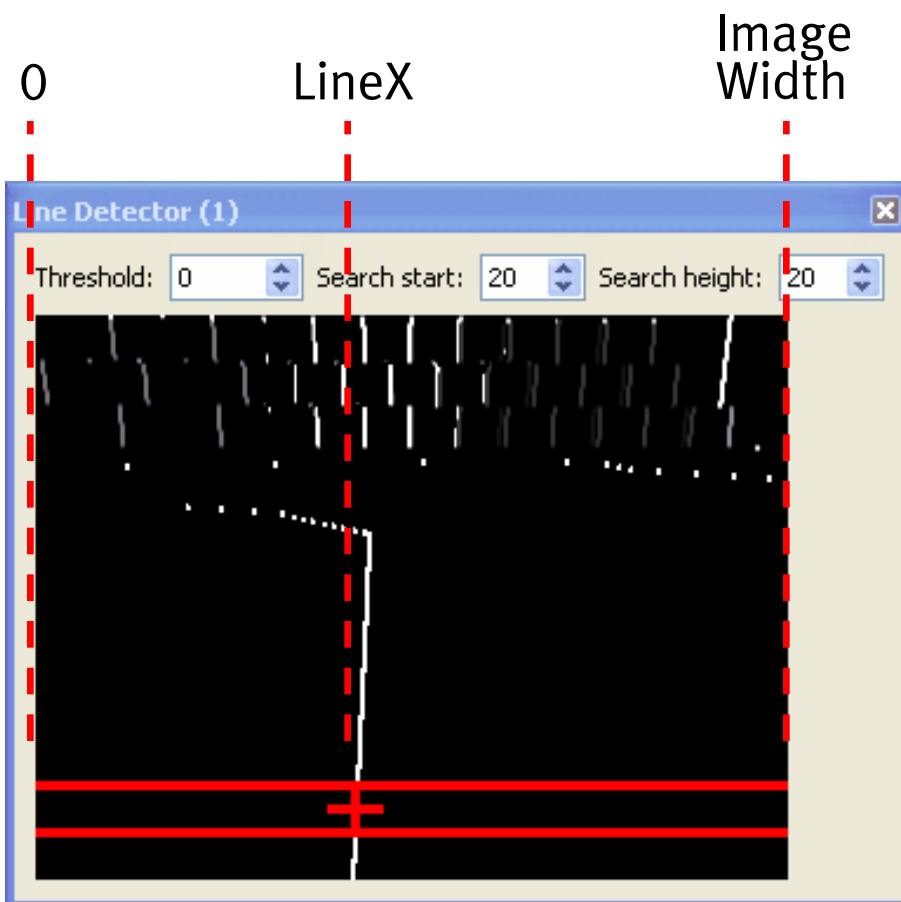


Line Detector

* Quanto maior o valor do Treshold maior o cancelamento de ruídos (0 - 255).

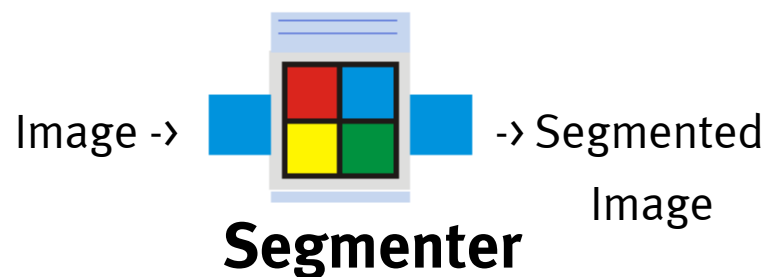
Search height ->

Search start ->



Robotino®View – Tratamento de Imagens

Image Processing:



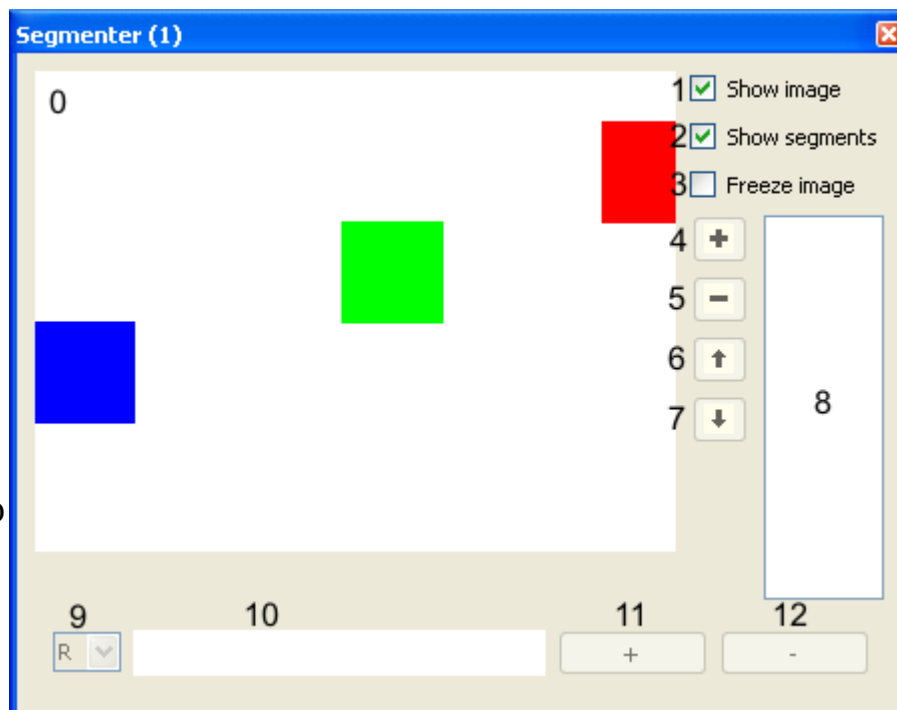
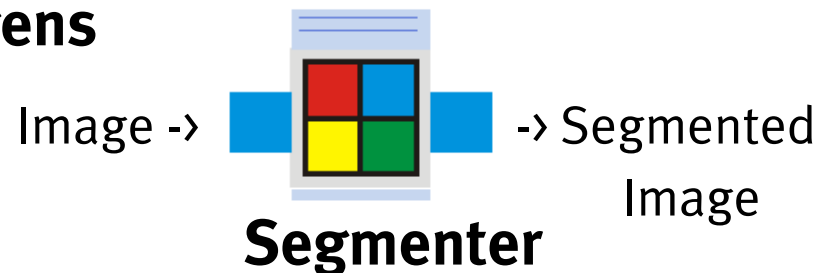
O bloco de função *Segementer* particiona a imagem de entrada em múltiplos segmentos.

A imagem de saída é formada pelos segmentos que foram encontrados.

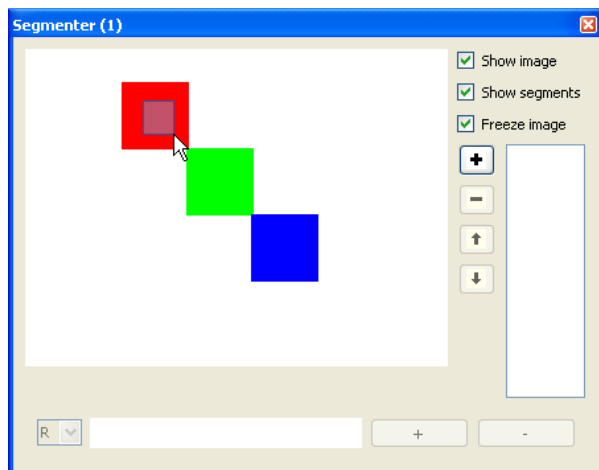
Robotino®View – Tratamento de Imagens

Image Processing:

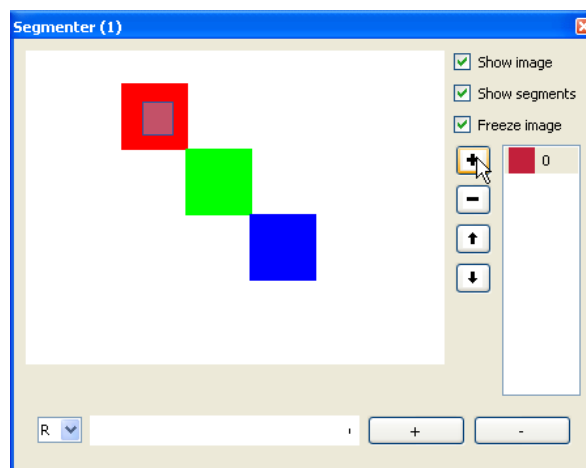
- 0 – Imagem/Segmentos
- 1 – Imagem é mostrada
- 2 – Segmentos são mostrados
- 3 – Congela a imagem
- 4 – Inclui o segmento selecionado
- 5 – Exclui um segmento
- 6 – Move o segmento para cima
- 7 – Move o segmento para baixo
- 8 – Lista de segmentos selecionados
- 9 – Seleciona a cor (R, G ou B) do segmento para otimização
- 10 – Escala de cor (R, G ou B)
- 11 – Altera a escala de cor (+)
- 12 – Altera a escala de cor (-)



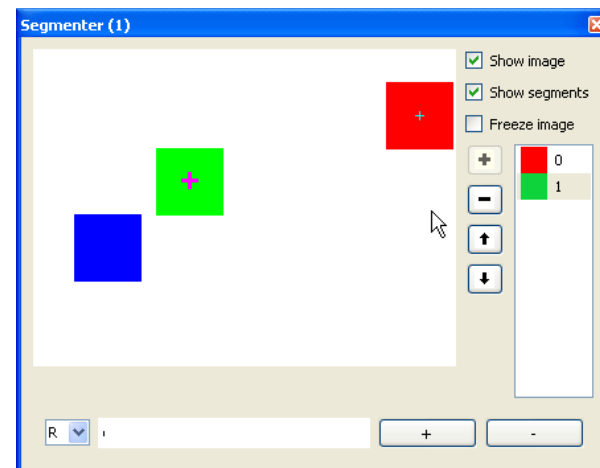
Robotino®View – Tratamento de Imagens



1 – Selecione o segmento



2 – Clique em (+) para adicionar o segmento na lista



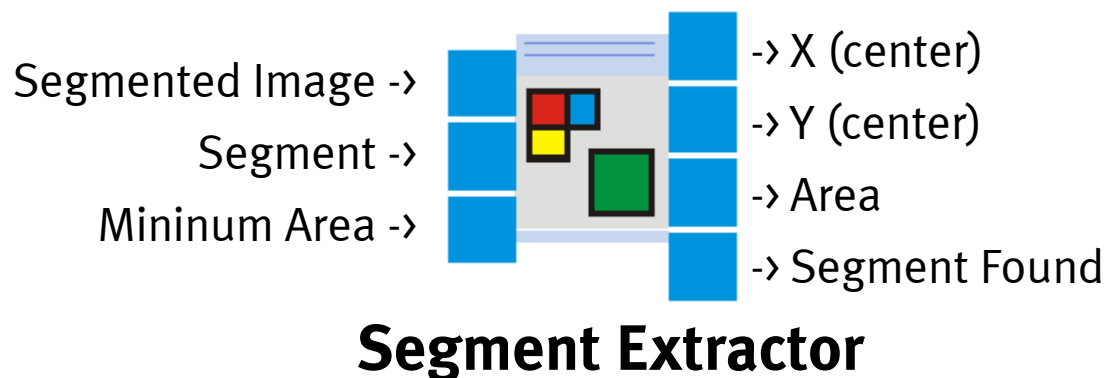
3 – Cruzes em negrito marcam os centros de massa dos segmentos da lista



Segmenter

Robotino®View – Tratamento de Imagens

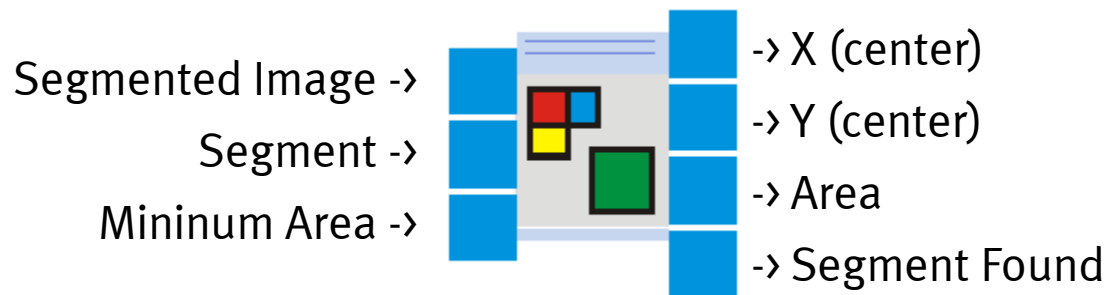
Image Processing:



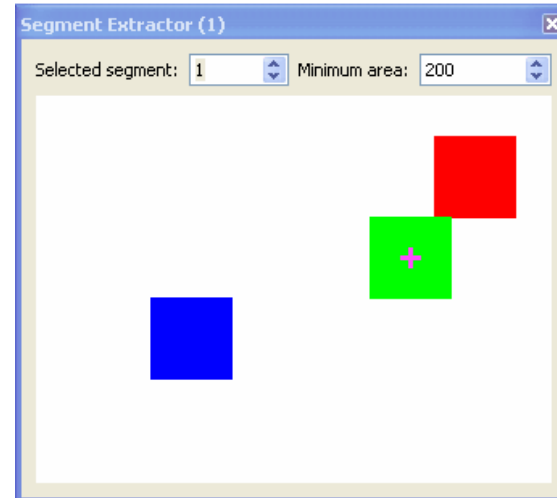
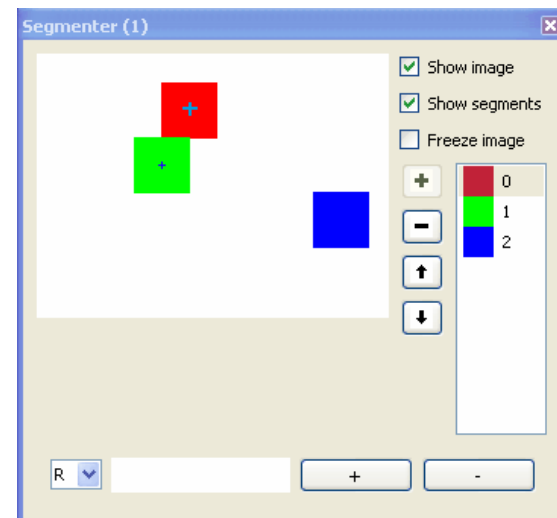
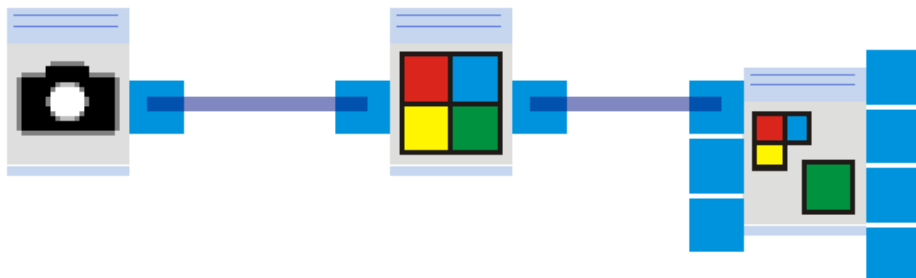
Obtém a posição e dimensões de segmentos de imagens, através de uma lista de segmentos de interesse, fornecidas pelo *Segmenter*.

Robotino®View – Tratamento de Imagens

Image Processing:

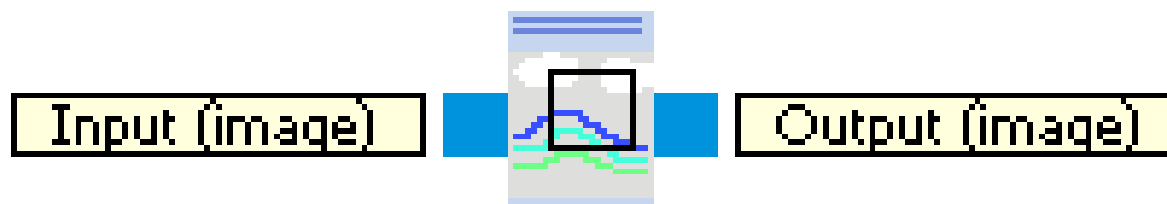


Segment Extractor



Robotino®View – Tratamento de Imagens

Image Processing:

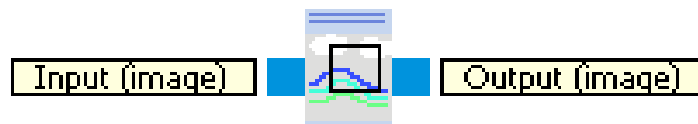
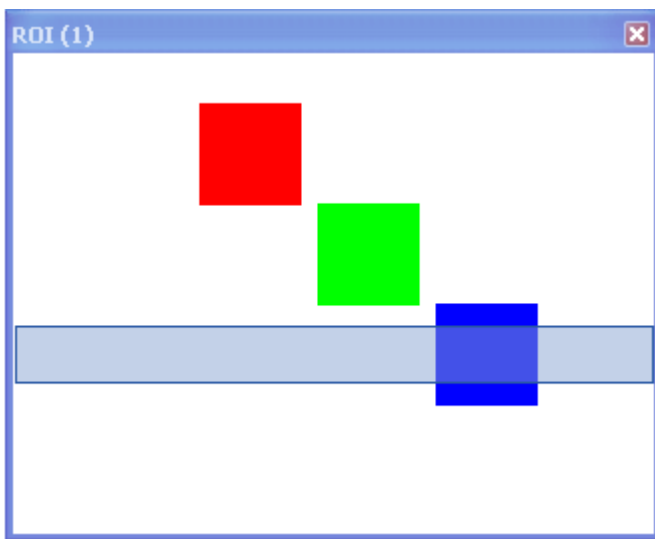


ROI (Region Of Interest)

Seleciona uma região de interesse na imagem de entrada.

Robotino®View – Tratamento de Imagens

Image Processing:

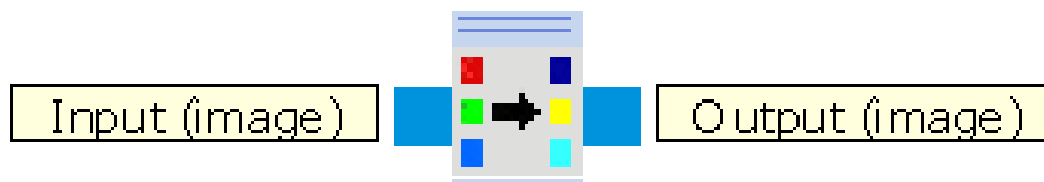


ROI (Region Of Interest)

Marca-se a região de interesse com o mouse.

Robotino®View – Tratamento de Imagens

Image Processing:



Colorspace Conversion

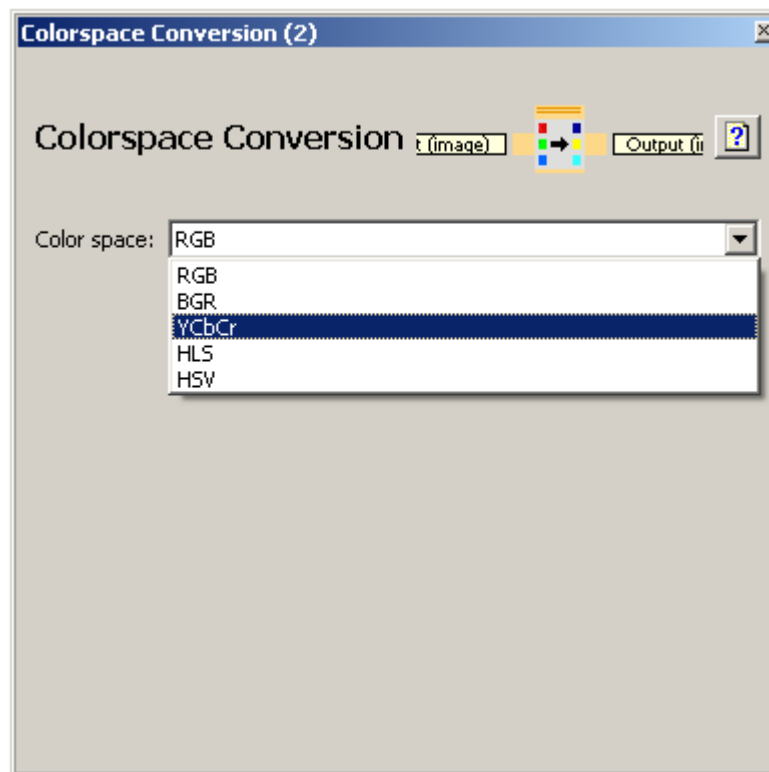
Bloco de função utilizado para se converter o sistema de cores das imagens.

Robotino®View – Tratamento de Imagens

Image Processing:

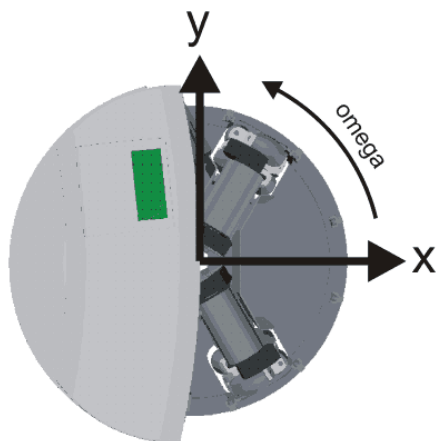


Colorspace Conversion



Robótica Móvel - Navegação

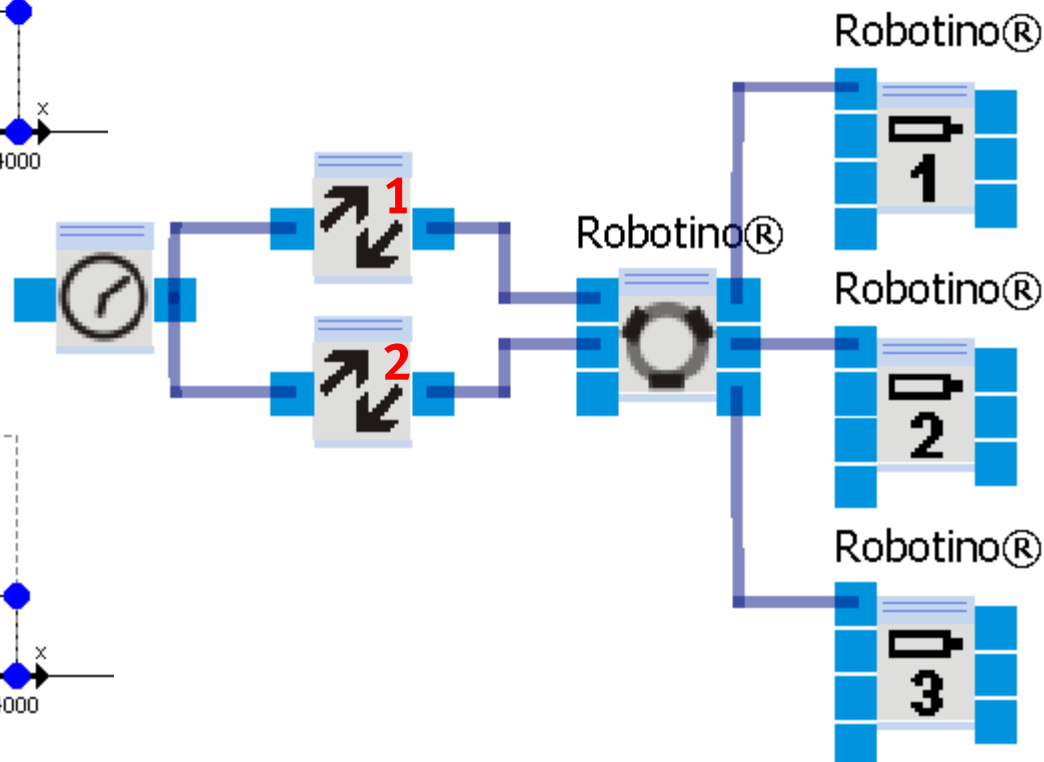
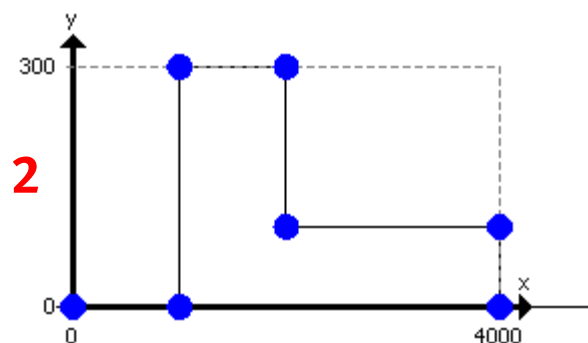
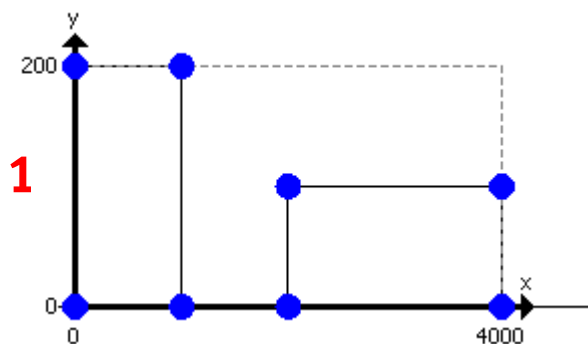
Definições:



- ✓ Sistema de coordenadas é um sistema para representar um determinado espaço com dimensões de forma matemática.
- ✓ Para que não se tenha ambigüidade de pontos, o sistema deve ter uma origem e uma orientação.
- ✓ No Robotino[®], cada ponto é dado por três coordenadas: x, y e ω .

Robotino®View – Navegação

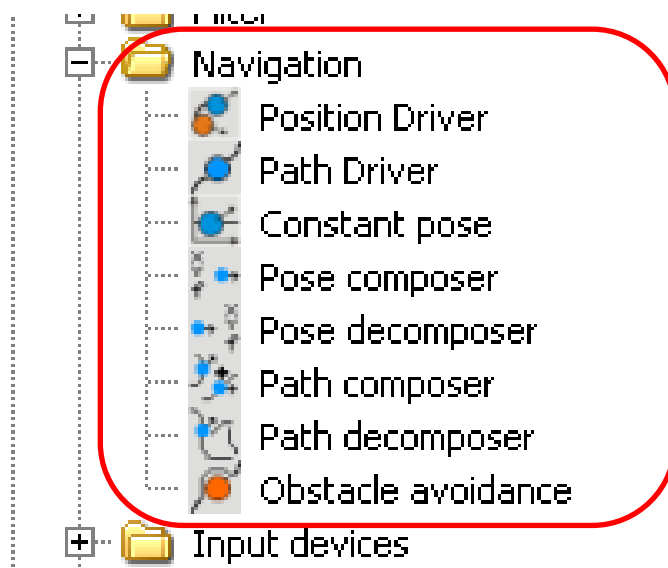
Exemplo:



Robotino®View – Navegação

Navigation:

Esta categoria compreende os blocos de função utilizados para realizar navegação com o Robotino®.



Robotino®View – Navegação

Navigation:

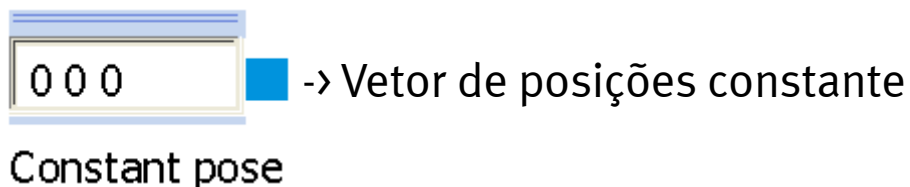
- A navegação no Robotino® será realizada através de vetores de posição.
- Um vetor de posições(*pose*) especifica coordenadas em “ x “, “ y “ e uma coordenada angular “ phi “ (orientação).
- Exemplo: vetor de posições (100 50 45)

$$\begin{aligned} & x = 100 \text{ mm} \\ \Rightarrow & y = 50 \text{ mm} \\ & \text{phi} = 45^\circ \end{aligned}$$

Robotino®View – Navegação

Navigation:

Constant pose: bloco de função utilizado para gerar vetores de posições constantes.

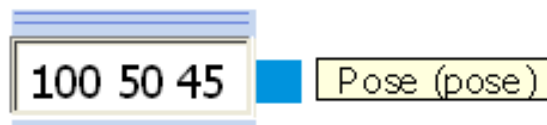


Na caixa de entrada são especificadas as coordenadas do vetor de posições, que devem ser separadas por espaços.

Robotino®View – Navegação

Navigation:

Constant pose: Exemplo:



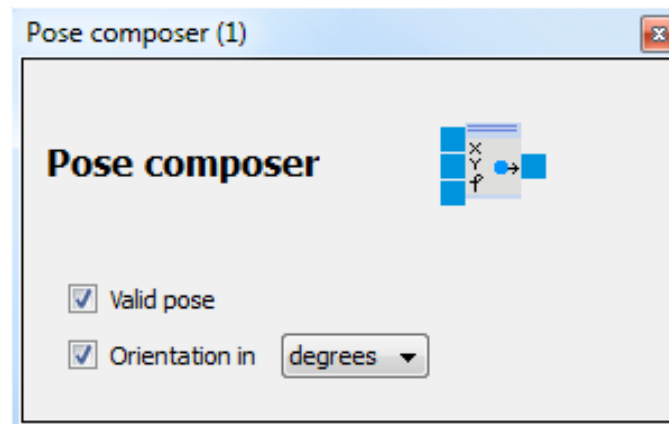
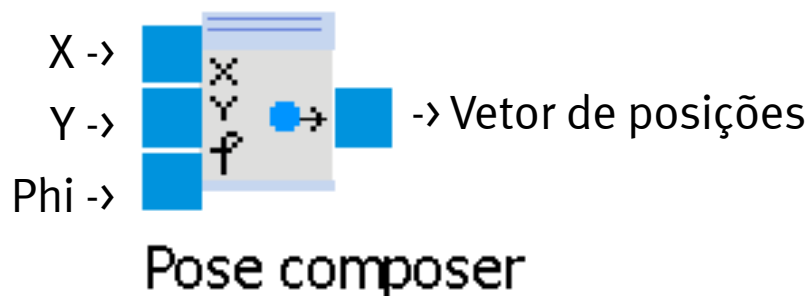
Neste vetor de posições foi especificado:

- 100mm na direção “x”
- 50mm na direção “y”
- Orientação de 45 graus.

Robotino®View – Navegação

Navigation:

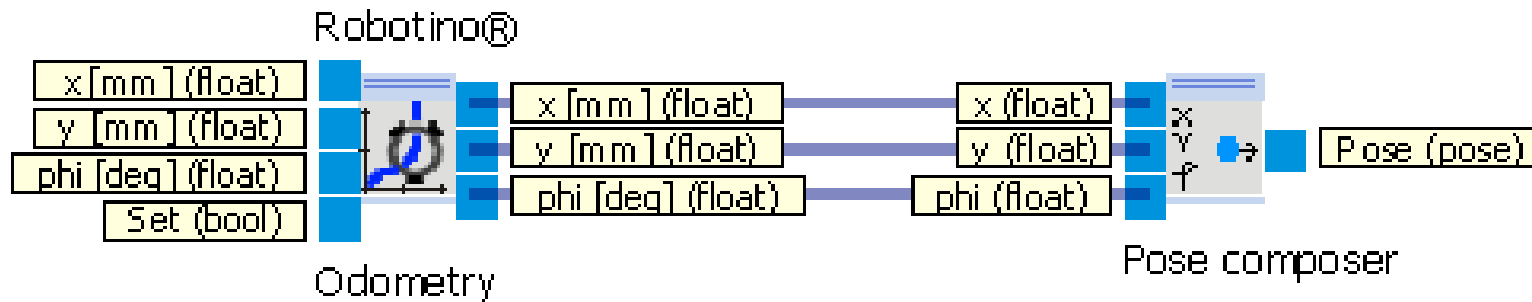
Pose composer: bloco de função utilizado para gerar vetores de posições, cujas coordenadas podem variar.



Robotino®View – Navegação

Navigation:

Pose composer: Exemplo:

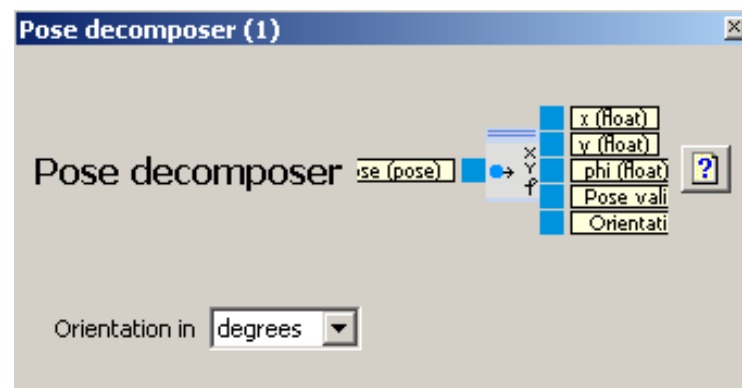
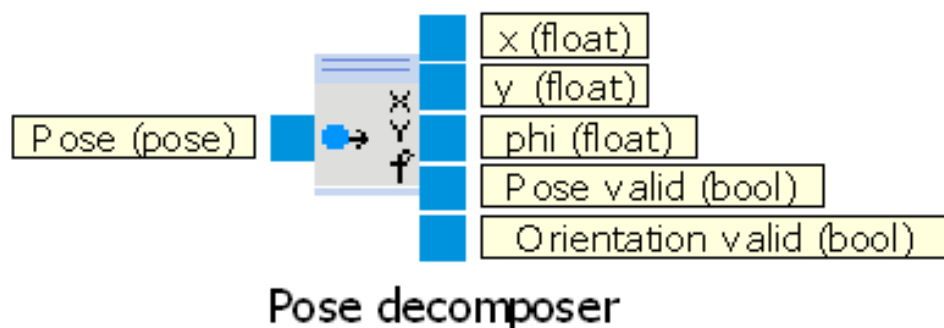


Neste exemplo utiliza-se o *Pose composer*, e outro bloco de função (*Odometry*), para gerar os vetores de posições atuais do Robotino®.

Robotino®View – Navegação

Navigation:

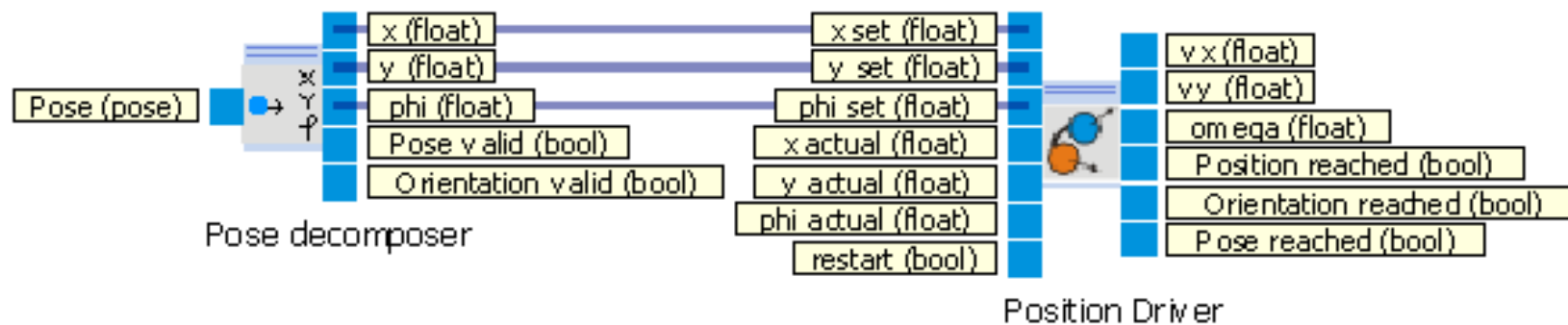
Pose decomposer: bloco de função utilizado para decompor vetores de posições em suas coordenadas individuais.



Robotino®View – Navegação

Navigation:

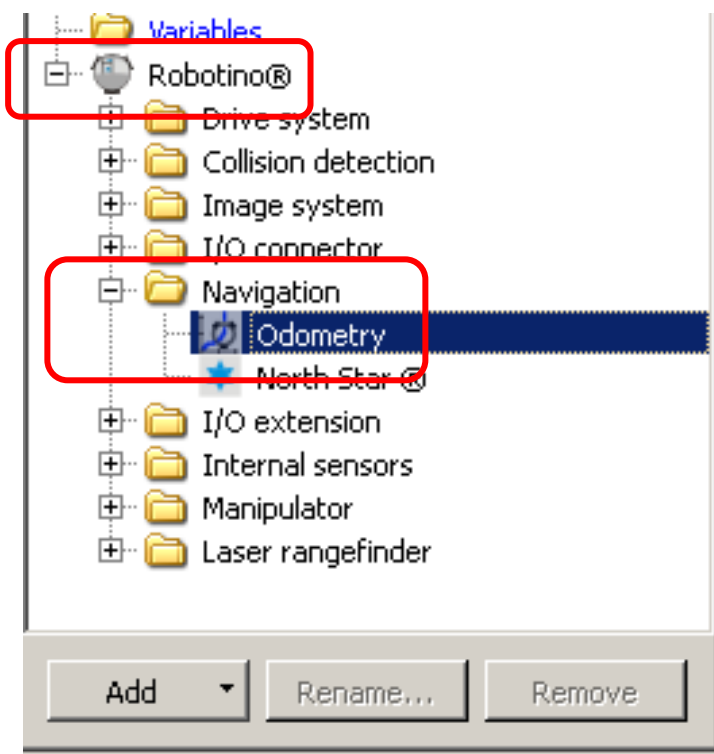
Pose decomposer: Exemplo:



Neste exemplo utiliza-se o *Pose decomposer*, e outro bloco de função (*Position driver*), para gerar dados de navegação.

Robotino®View – Navegação

Odometry:

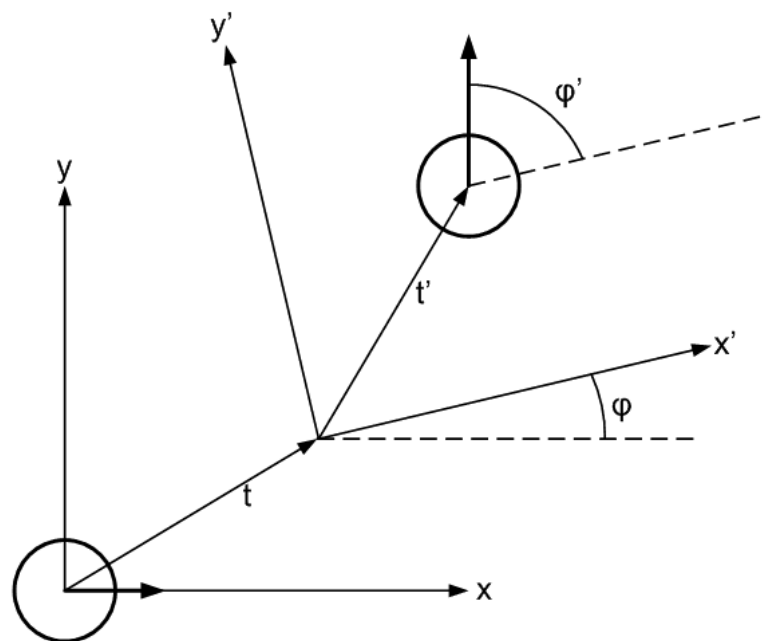


Odometria é o uso dos dados coletados, a partir do movimento dos motores, para estimar mudanças de posição do robô ao longo do tempo.

Robotino[®]View – Navegação

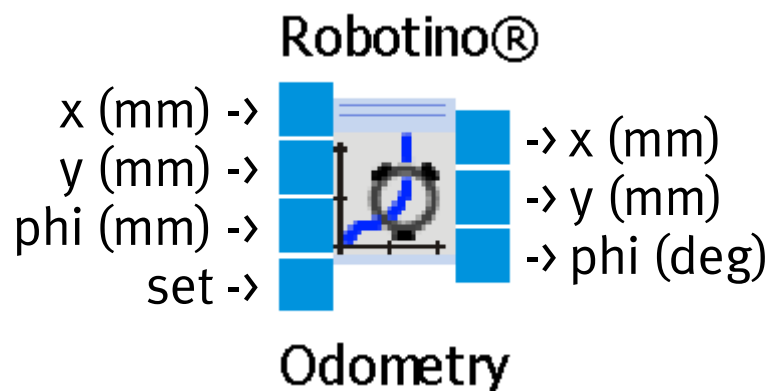
Odometry:

- No *Robotino[®] View*, o bloco de função *Odometry* (Odômetro) está disponível na biblioteca **Robotino**.
- A função do Odômetro é de estimar a posição atual do Robotino[®] através das informações de movimento de cada um dos motores ao longo do tempo.



Robotino®View – Navegação

Odometry:



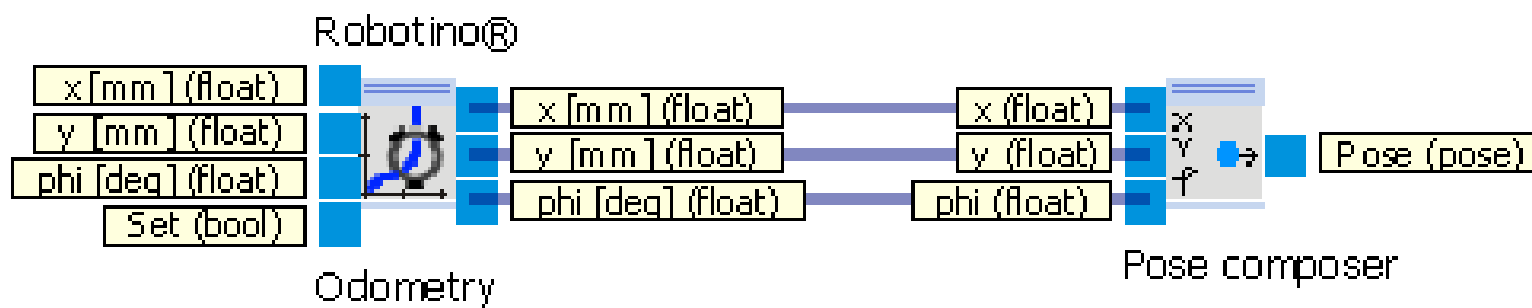
Entrada: Coordenadas e orientação iniciais.

Saída: Coordenadas e orientação atuais.

Set: Habilita o Odômetro.

Robotino®View – Navegação

Odometry: Exemplo:

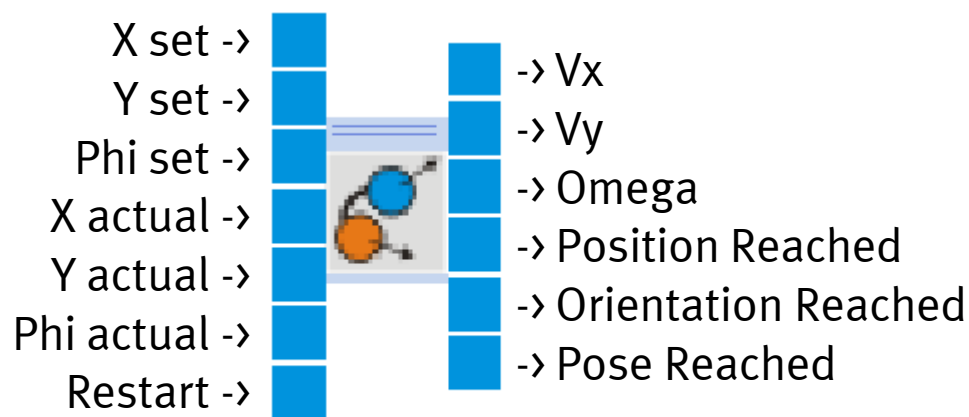


Neste exemplo utiliza-se o *Odometry*, para obter os dados de navegação, e posteriormente gerar um vetor de posições atuais.

Robotino[®]View – Navegação

Navigation:

Position Driver



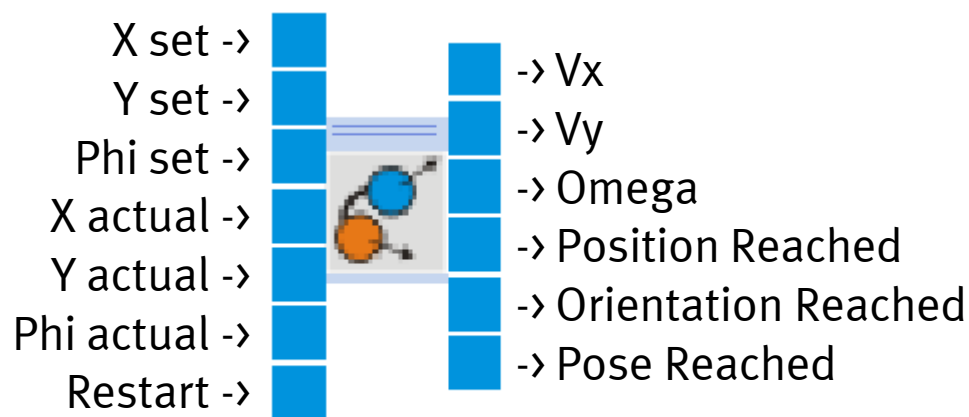
Position Driver

O bloco de função *Position Driver* é utilizado para guiar o Robotino[®] à uma determinada posição (coordenada) desejada.

Robotino®View – Navegação

Navigation:

Position Driver



Position Driver

O *Position* Driver gera os valores de velocidades linear e angular de referência que fazem com que o Robotino® atinja esta posição desejada.

Robotino®View – Funções Principais

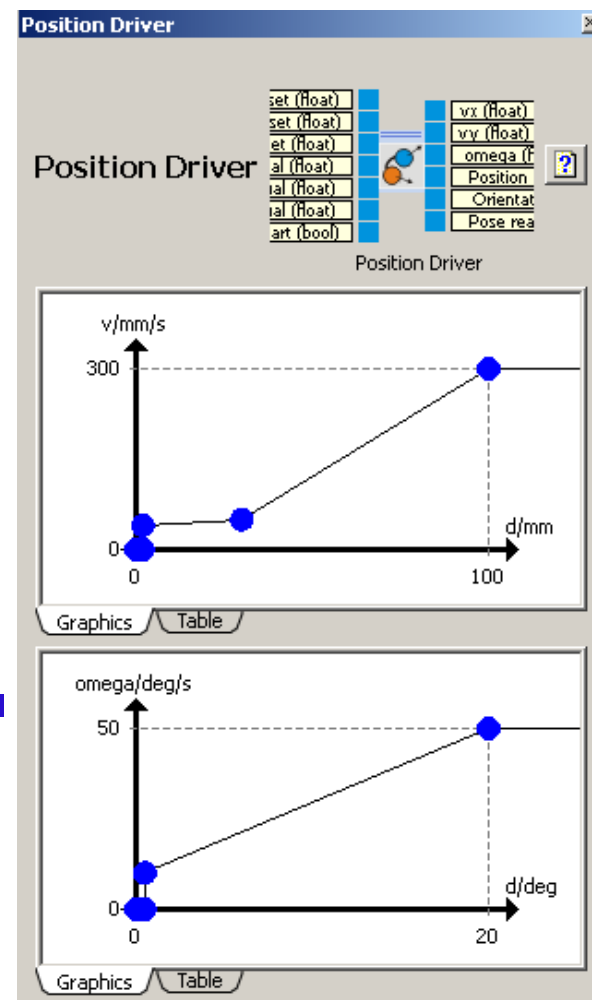
Navigation:

Position Driver: Mapas de velocidade

Mapas de velocidade **linear** e **angular**

Linear: mapeamento da velocidade linear (mm/s) do Robotino® de acordo com a distância linear até a posição desejada.

Angular: mapeamento da velocidade angular (°/s) do Robotino® de acordo com a defasagem angular entre o Robotino® e a orientação desejada.



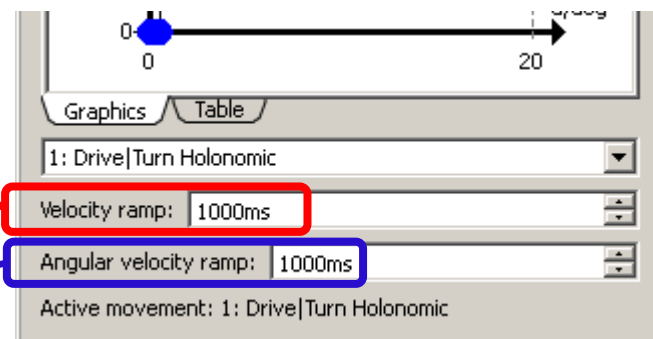
Robotino®View – Funções Principais

Navigation:

Position Driver: Tempos de resposta

Tempo em milissegundos para que o Robotino® atinja 100% da velocidade linear desejada

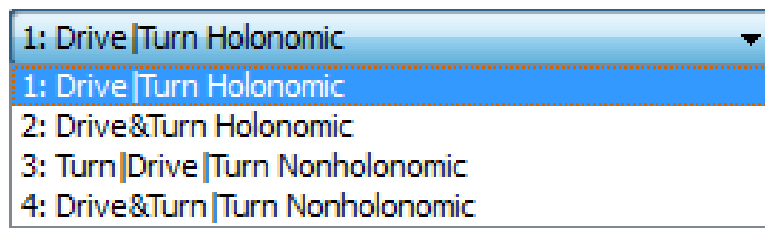
Tempo em milissegundos para que o Robotino® atinja 100% da velocidade angular desejada



Robotino®View – Funções Principais

Navigation:

Position Driver: Tipos de movimentos

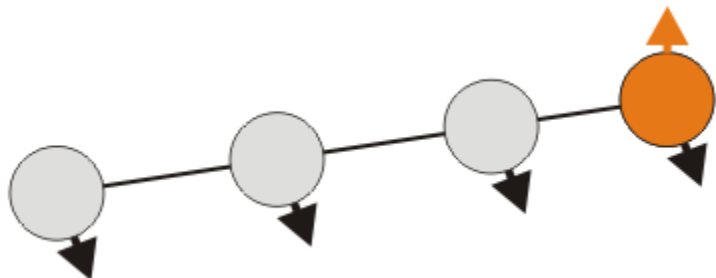


Robotino®View – Funções Principais

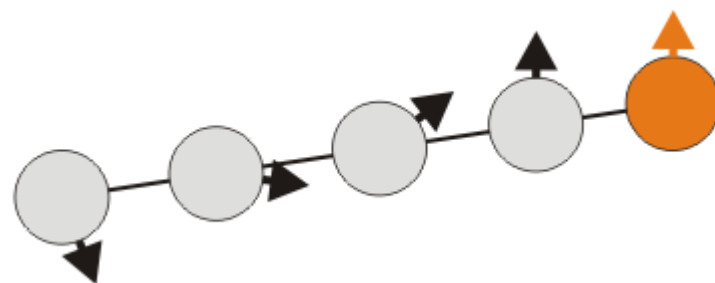
Navigation:

Position Driver: Tipos de movimentos

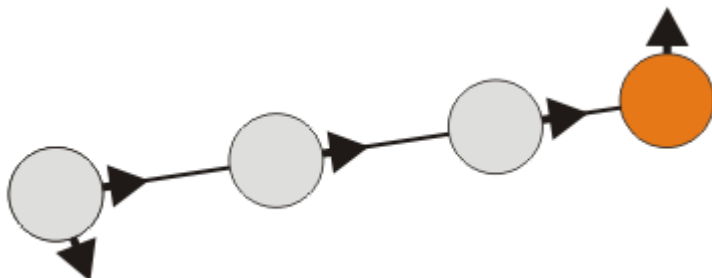
DriveTurn Holonomic



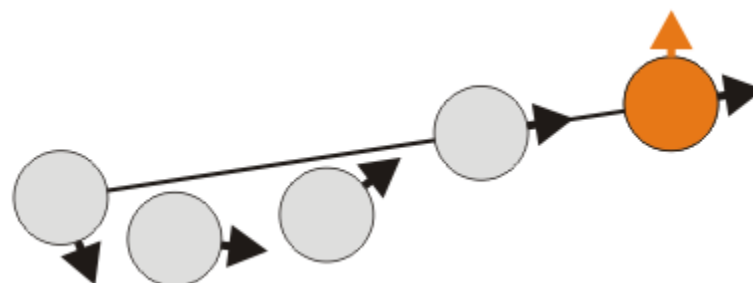
Drive&Turn Holonomic



TurnlDriveTurn Nonholonomic



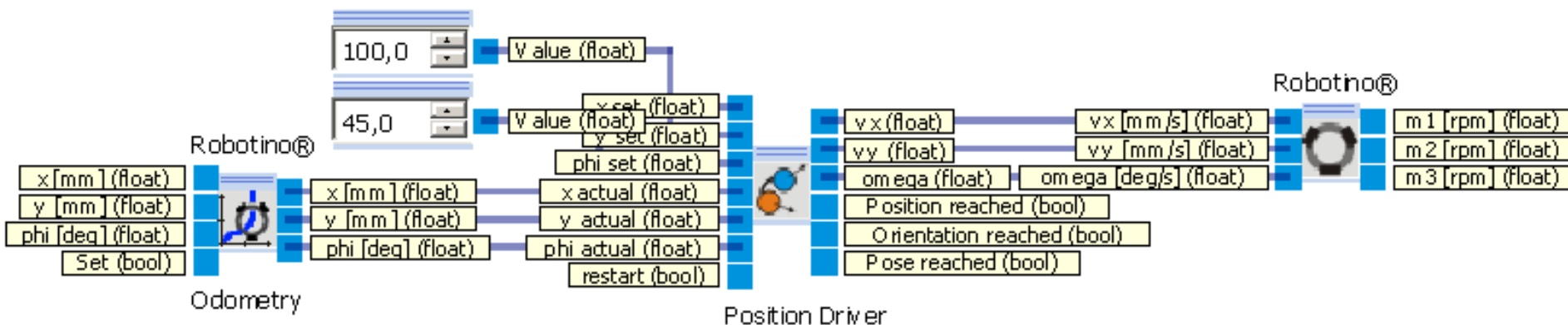
Drive&TurnlTurn Nonholonomic



Robotino®View – Navegação

Navigation:

Position Driver: Exemplo:

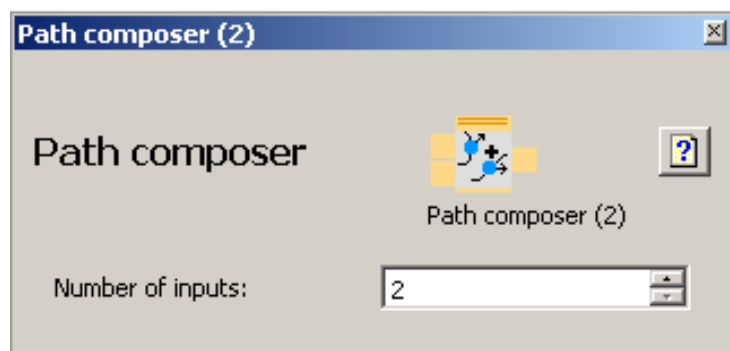
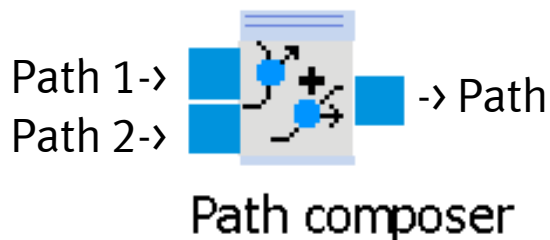


Neste exemplo, utiliza-se o *Odometry*, para obter os dados de navegação e posteriormente utilizar estes dados como coordenadas de referência no bloco de função *Position Driver*.

Robotino®View – Navegação

Navigation:

Path composer

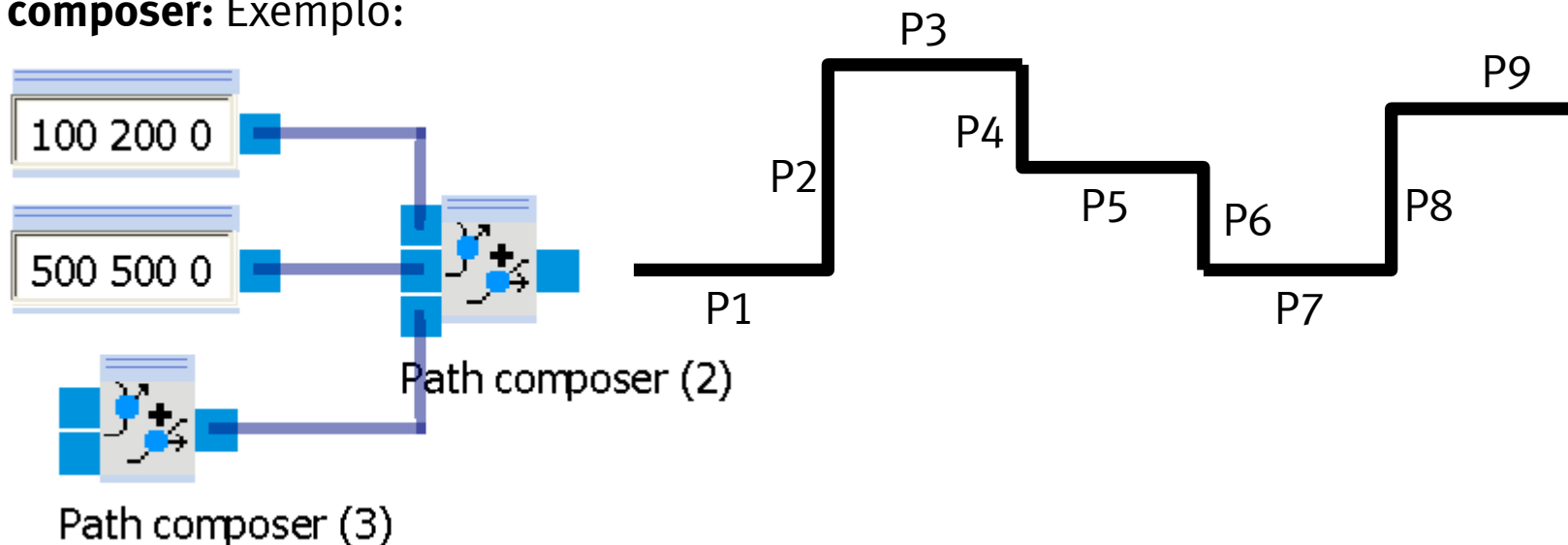


O bloco de função *Path composer* gera trajetórias para o Robotino® através da informação de dois ou mais vetores de posições de entrada.

Robotino®View – Navegação

Navigation:

Path composer: Exemplo:

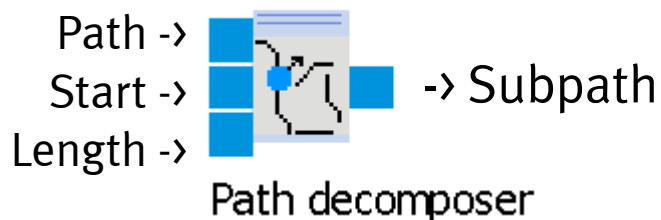


Neste exemplo, uma determinada trajetória é gerada a partir de 3 vetores de entrada no bloco de função *Path composer (2)*

Robotino®View – Navegação

Navigation:

Path decomposer



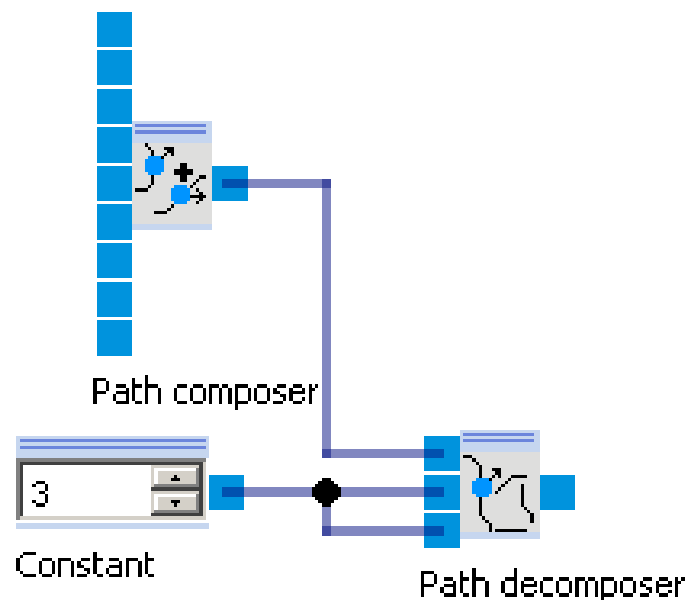
O bloco de função *Path decomposer* fragmenta a trajetória que é fornecida em sua entrada *Path*.

Deve ser informada a coordenada de início (*Start*) e o comprimento do caminho (*Length*) que se quer segmentar.

Robotino®View – Navegação

Navigation:

Path decomposer: Exemplo:



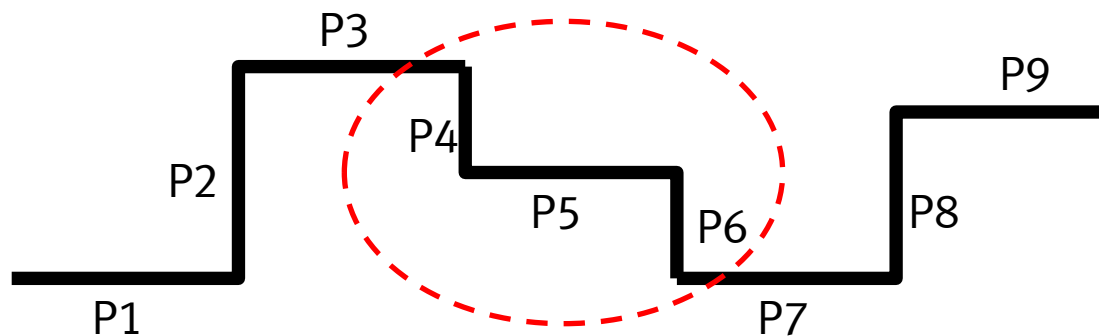
Neste exemplo, utiliza-se o *Path decomposer* para segmentar o *Path* composto de 9 segmentos. Escolhe-se o terceiro como segmento inicial e um *Subpath* resultante de comprimento 3.

Robotino®View – Navegação

Navigation:

Path decomposer: Exemplo:

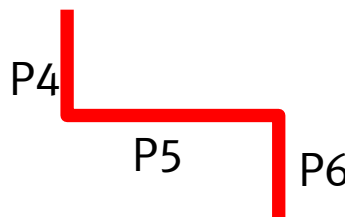
Path ->



Start = 3

Length = 3

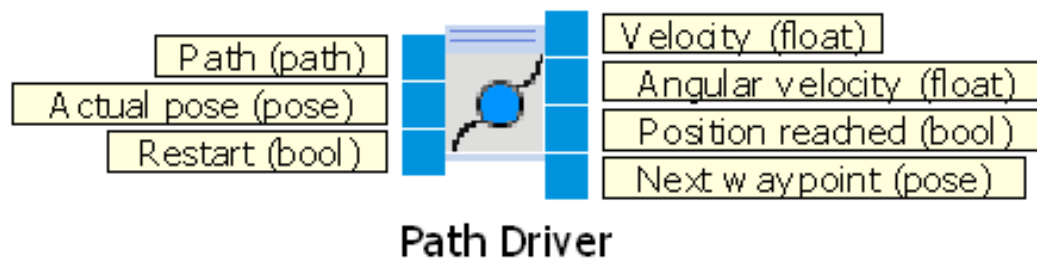
Subpath ->



Robotino®View – Navegação

Navigation:

Path driver



Com a utilização do bloco de função *Path Driver* torna-se possível guiar o Robotino® através de trajetórias completas.

Robotino®View – Navegação

Navigation:

Path driver



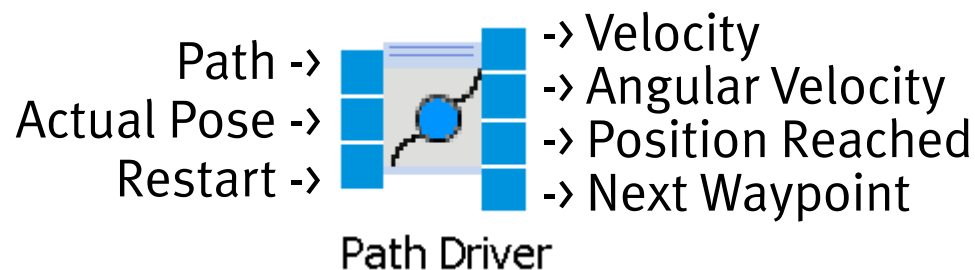
Path Driver

Através da trajetória fornecida (*Path*) e o vetor de posição atual (*Actual Pose*) calcula-se as velocidades linear e angular necessárias para guiar o Robotino® através desta trajetória dada.

Robotino[®]View – Navegação

Navigation:

Path driver



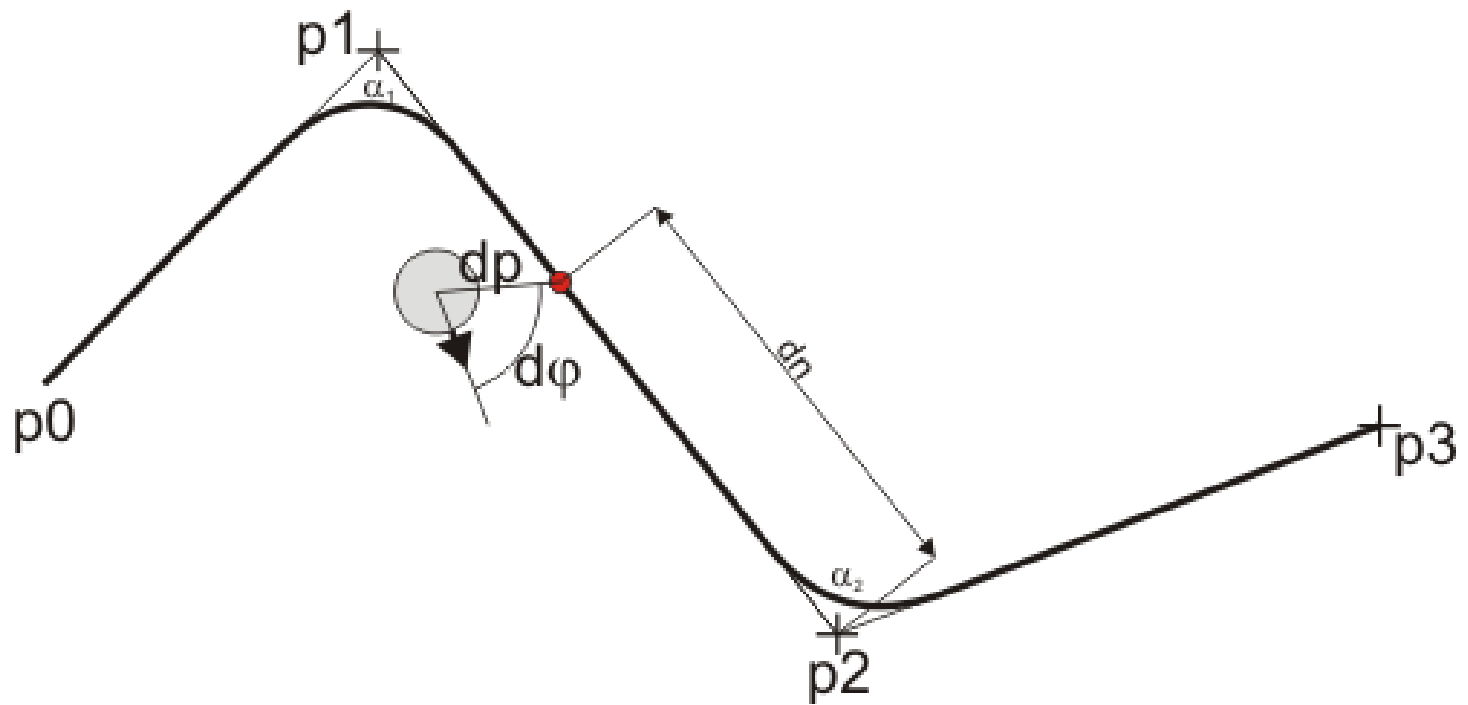
Obs: O sinal ***Next Waypoint*** disponibiliza o vetor de posições (*Pose*) do próximo ponto que o Robotino[®] irá alcançar.

Robotino[®]View – Navegação

Navigation:

Path driver:

Configurações

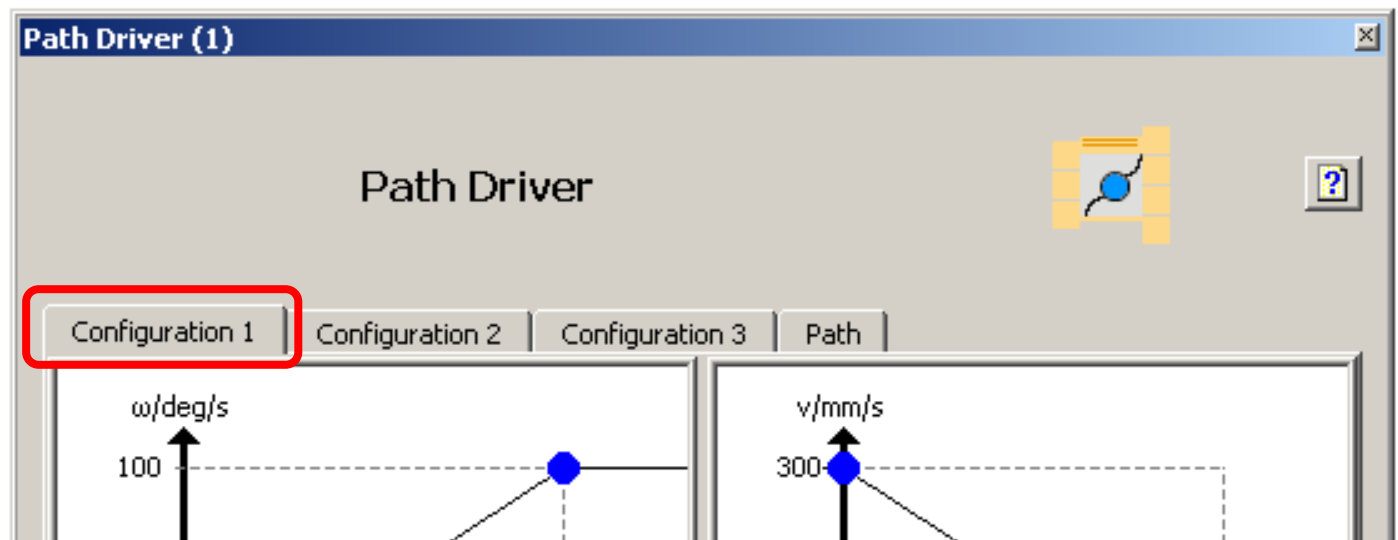


Robotino®View – Navegação

Navigation:

Path driver:

Configuração 1

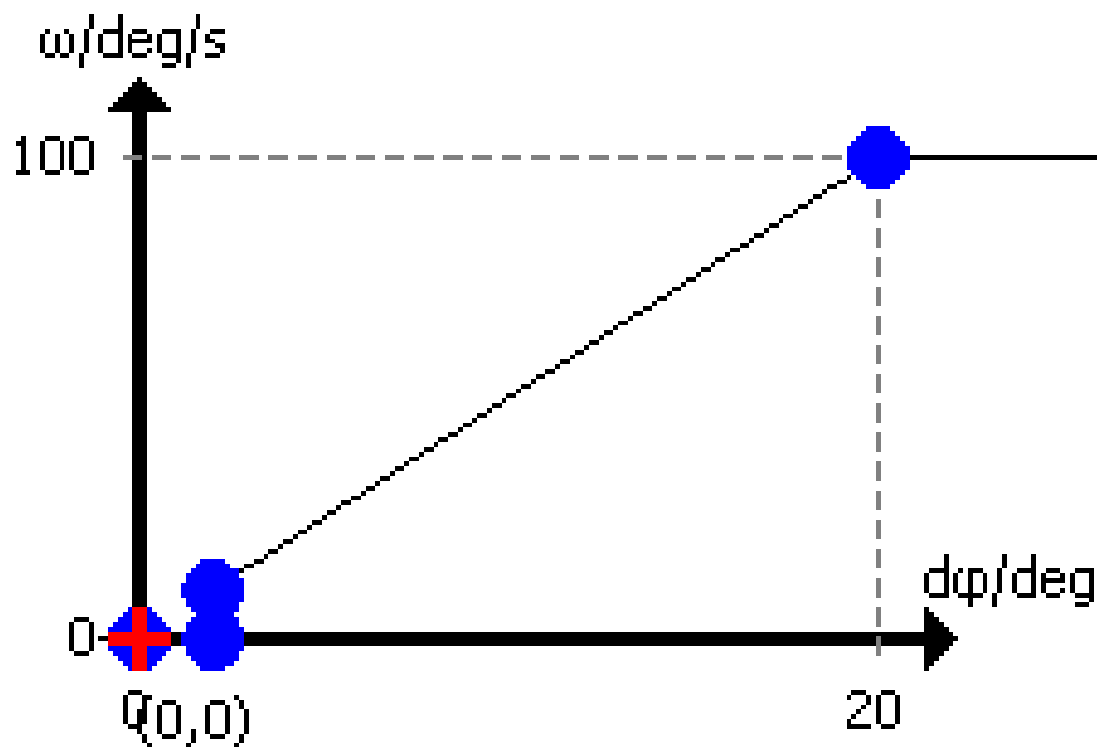


Robotino[®]View – Navegação

Navigation:

Path driver:

Configuração 1



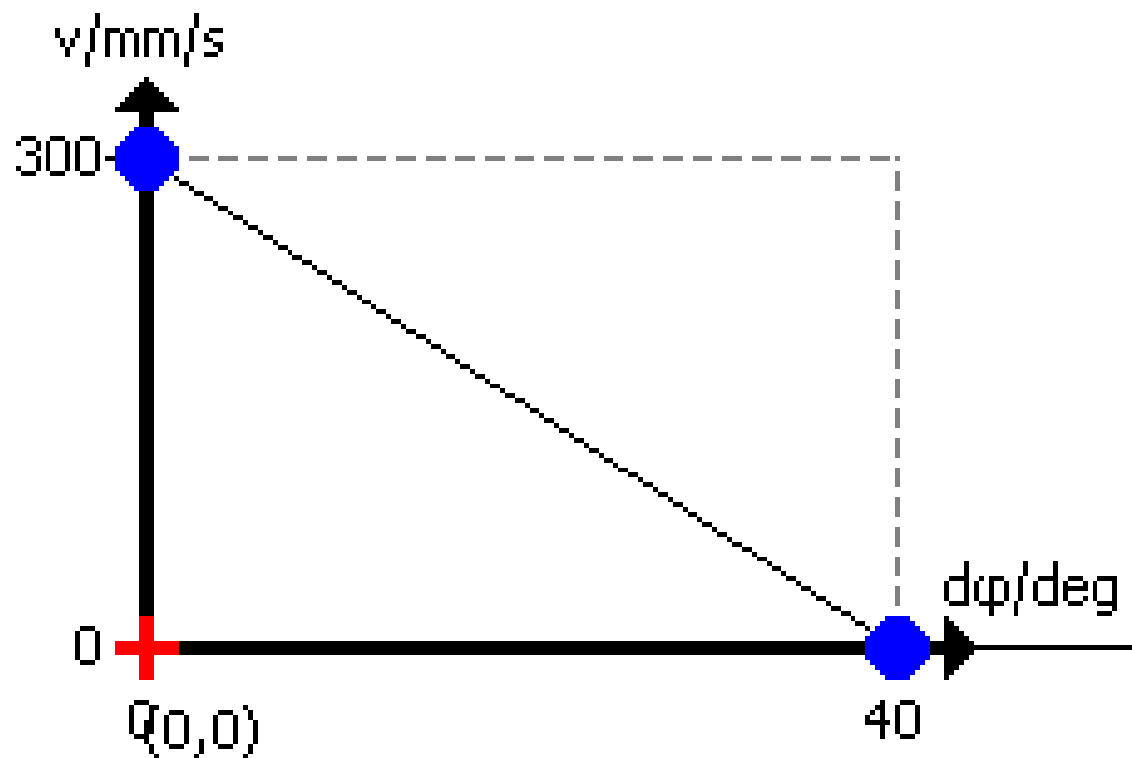
Relação entre velocidade angular e o erro do ângulo

Robotino®View – Navegação

Navigation:

Path driver:

Configuração 1



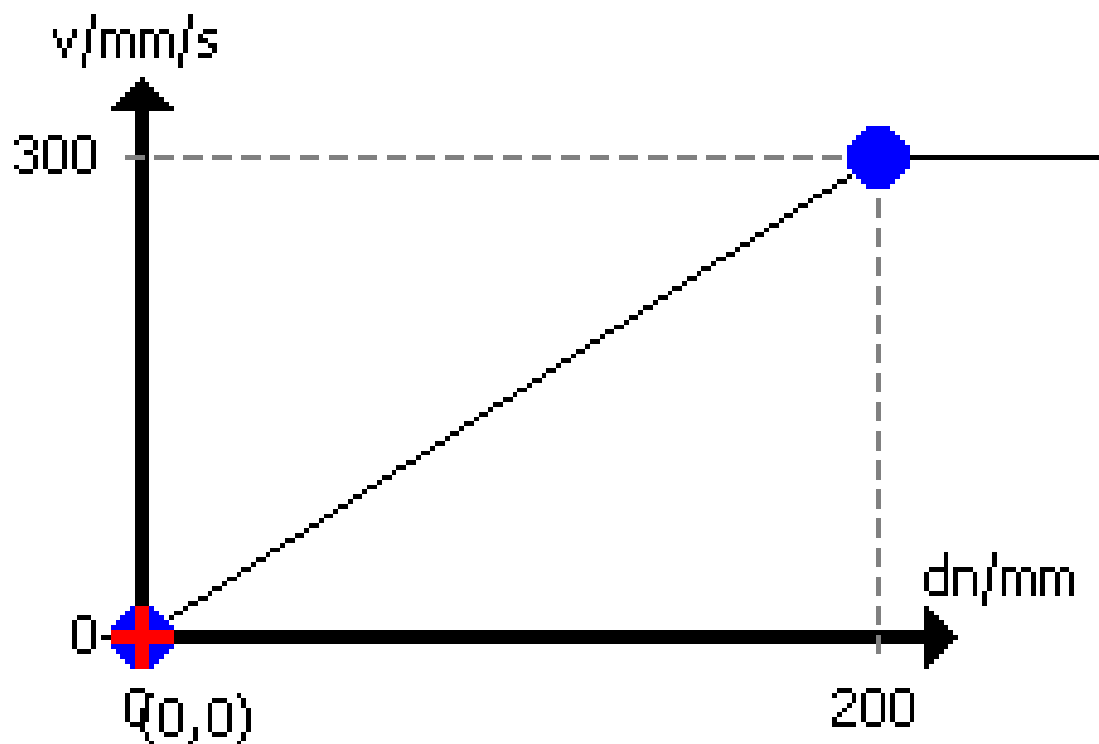
Relação entre velocidade linear e o erro do ângulo

Robotino®View – Navegação

Navigation:

Path driver:

Configuração 1



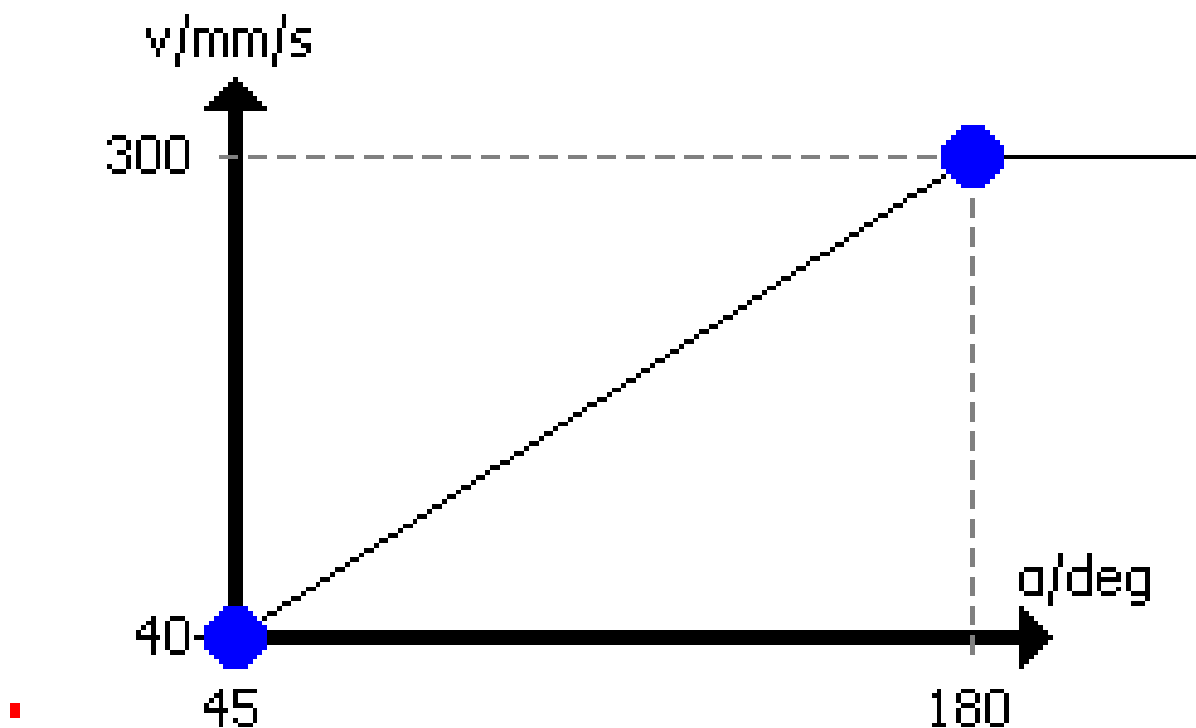
Relação entre velocidade linear e a distância até o próximo ponto;

Robotino[®]View – Navegação

Navigation:

Path driver:

Configuração 1



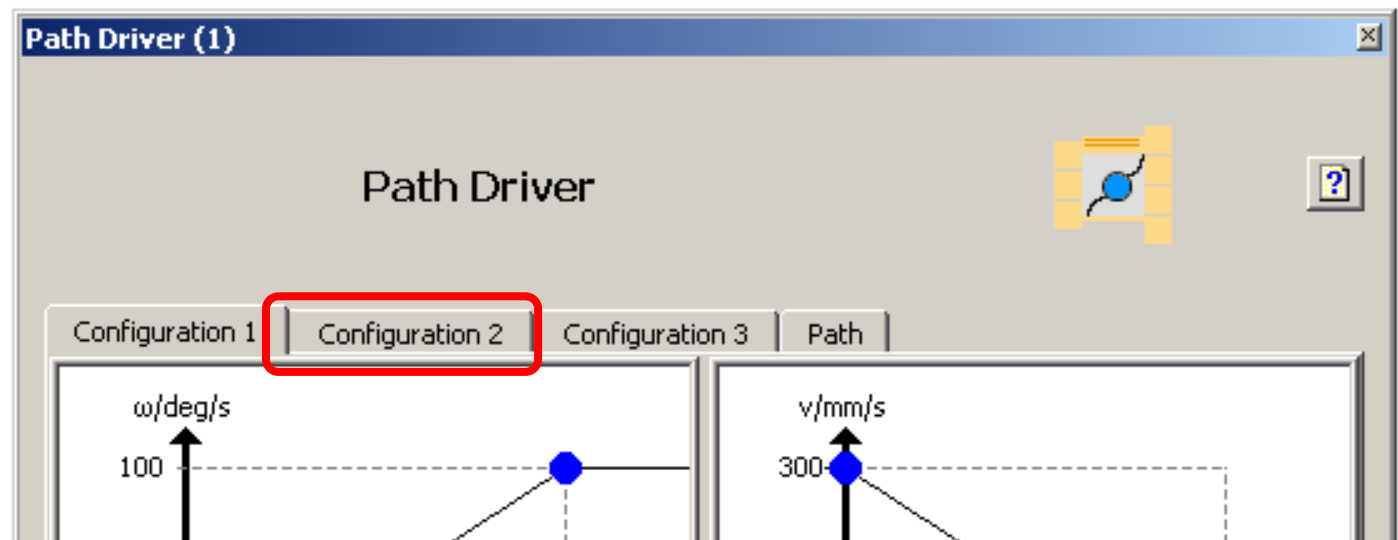
Relação entre velocidade linear e o ângulo até próximo caminho;

Robotino®View – Navegação

Navigation:

Path driver:

Configuração 2

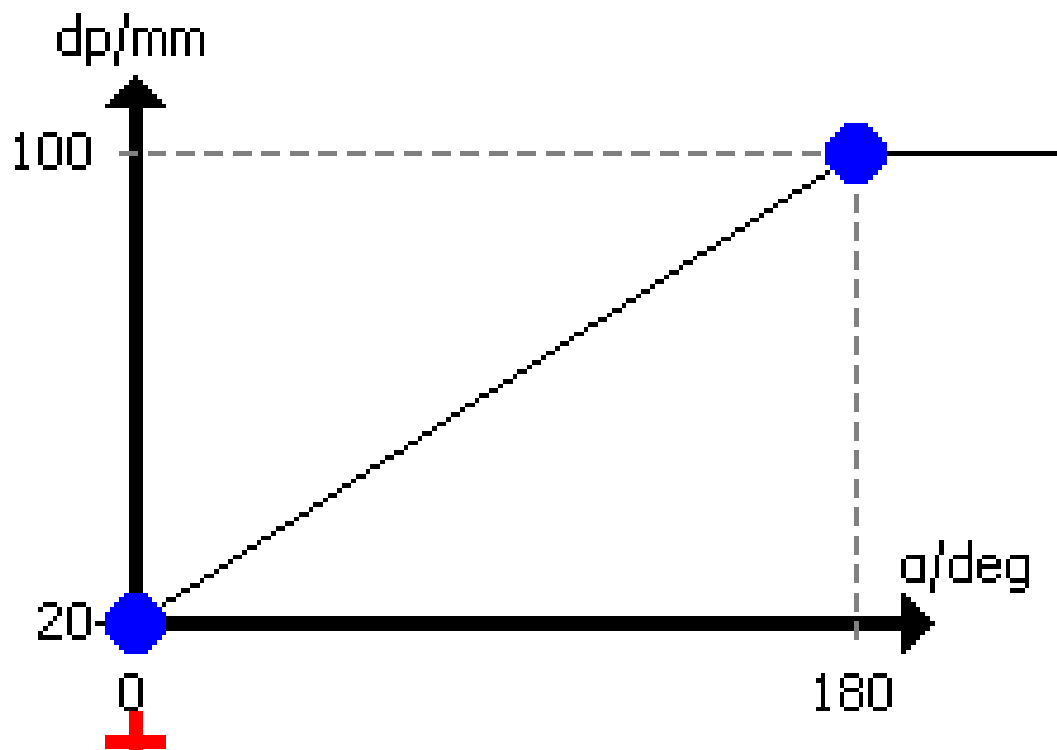


Robotino[®]View – Navegação

Navigation:

Path driver:

Configuração 2



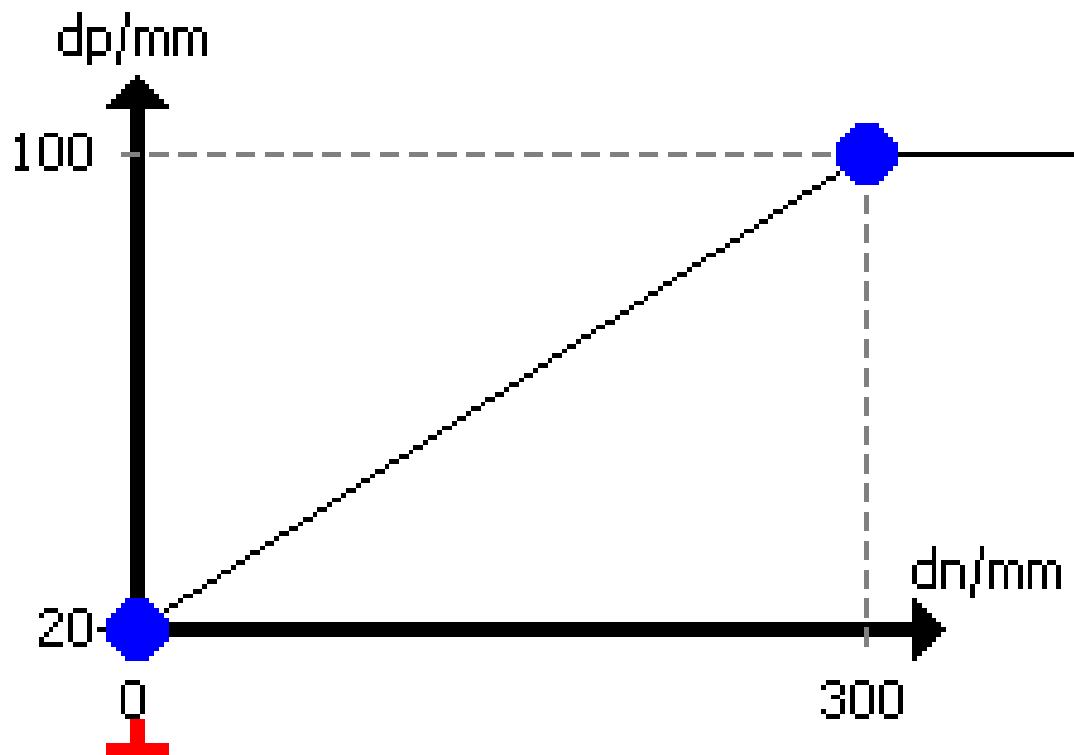
Relação entre a distância do ponto virtual e o ângulo até o próximo caminho

Robotino[®]View – Navegação

Navigation:

Path driver:

Configuração 2



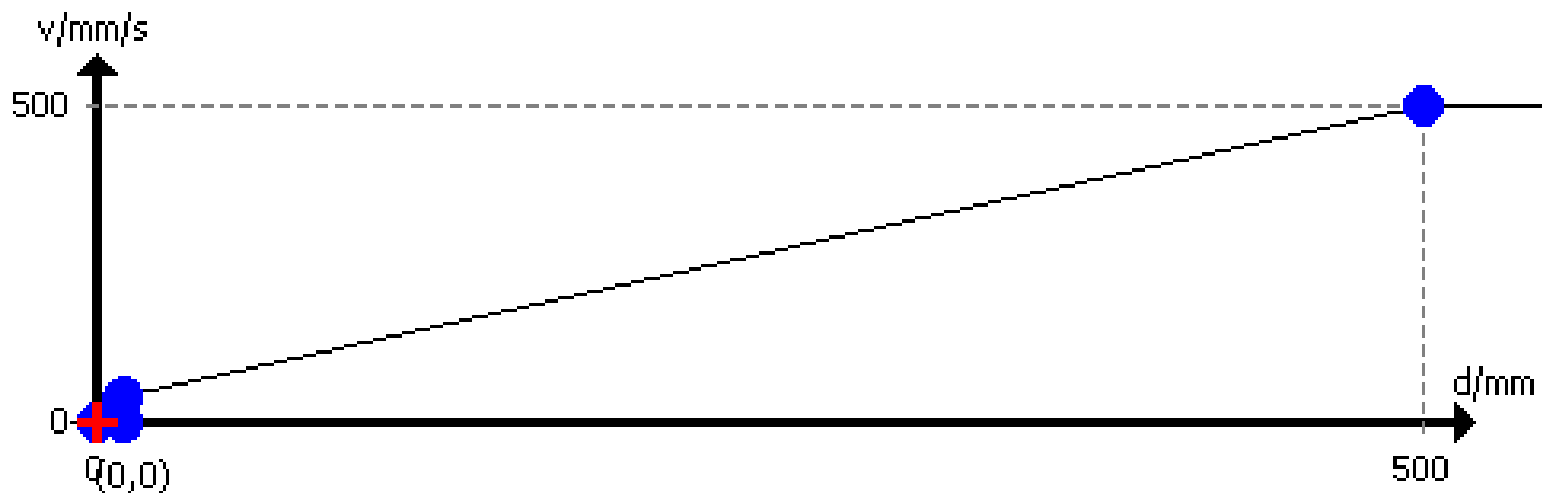
Relação entre a distância do ponto virtual e a distância até o próximo ponto;

Robotino[®]View – Navegação

Navigation:

Path driver:

Configuração 2



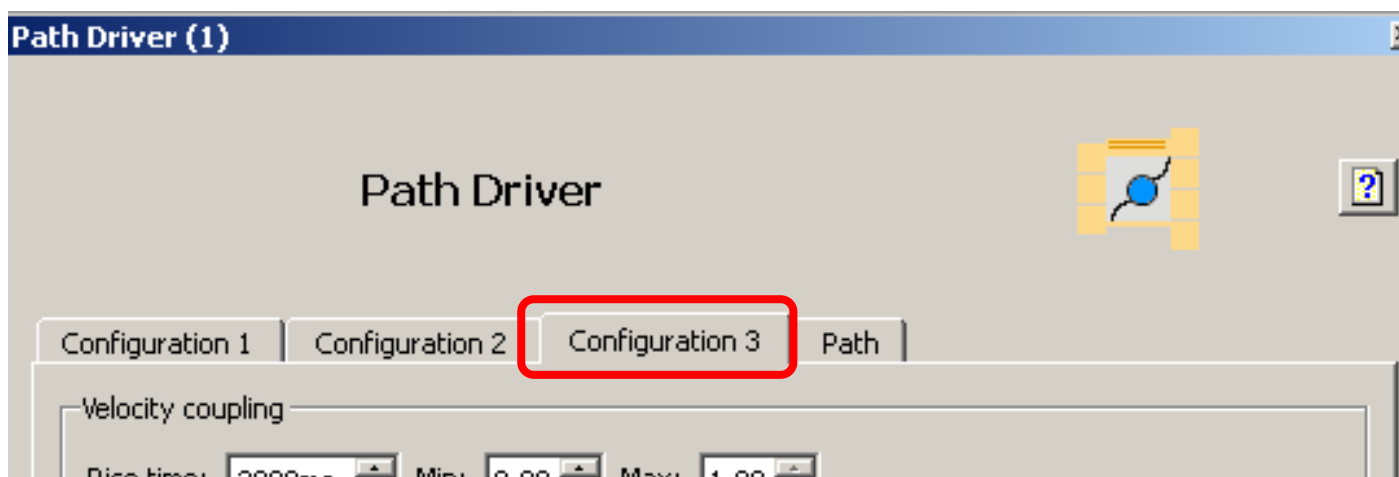
Relação entre velocidade linear e a distância até o fim do caminho;

Robotino®View – Navegação

Navigation:

Path driver:

Configuração 3

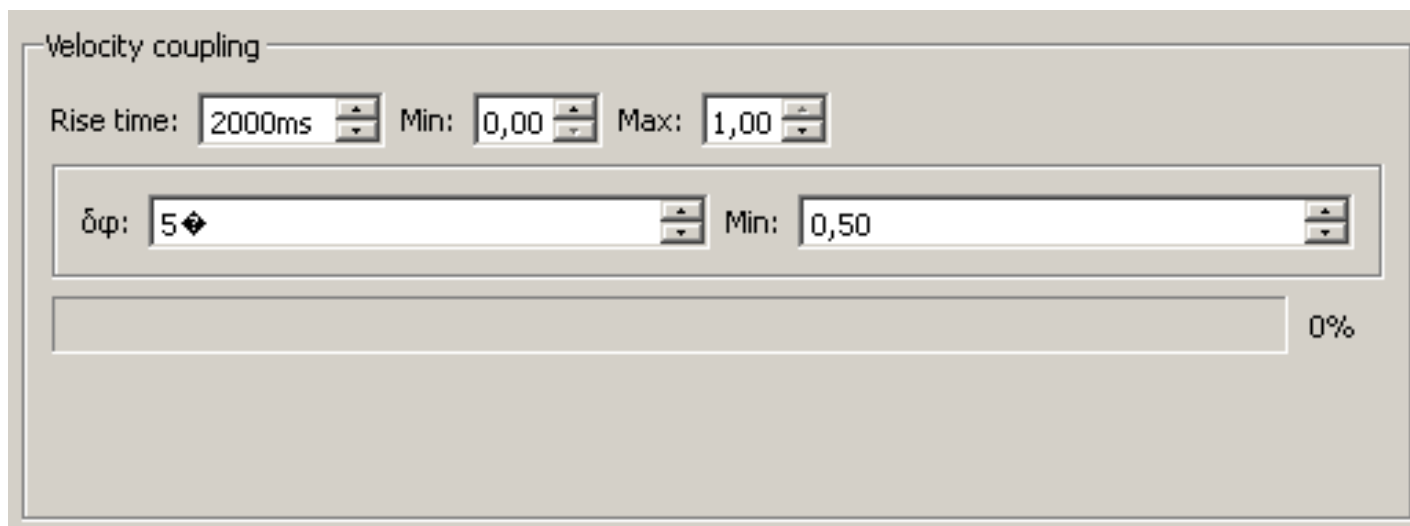


Robotino®View – Navegação

Navigation:

Path driver:

Configuração 3



The screenshot shows a software window titled "Velocity coupling". It contains several input fields with up/down arrows for adjustment:

- Rise time: 2000ms
- Min: 0,00
- Max: 1,00
- $\delta\varphi$: 5
- Min: 0,50

At the bottom right, there is a progress bar and a label "0%".

Adaptação do fator de acoplamento entre a velocidade calculada devido as configurações em 1 e 2 e a velocidade real.

Robotino®View – Navegação

Navigation:

Path driver:

Configuração 3

Angular velocity coupling

Rise time: Min: Max:

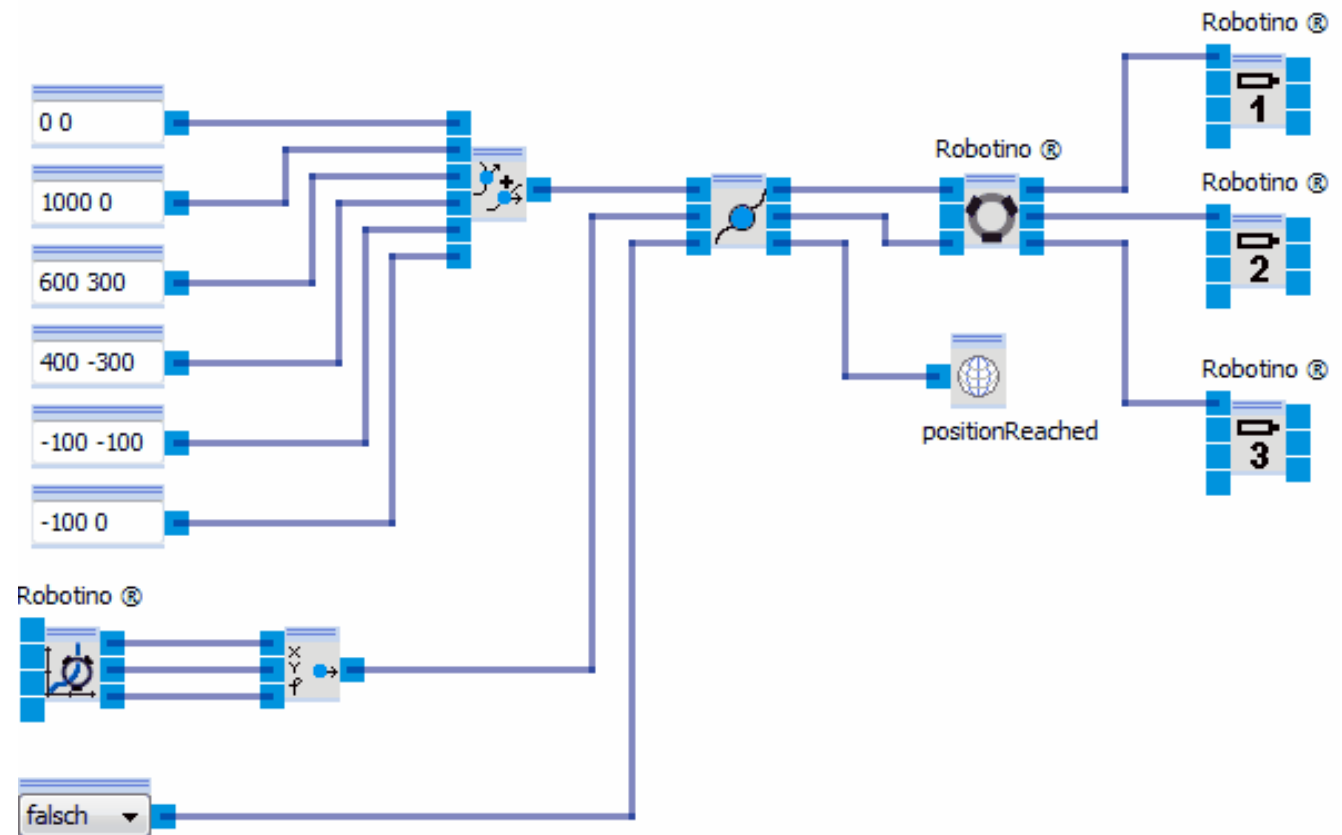
$\delta\varphi$: Min:

Adaptação do fator de acoplamento entre a velocidade angular calculada devido as configurações em 1 e 2 e a velocidade angular real.

Robotino®View – Navegação

Navigation:

Path driver: Exemplo



Robotino®View – Funções Principais

Navigation:

Path ->

Obstacle Pose ->

Obstacle Radius ->

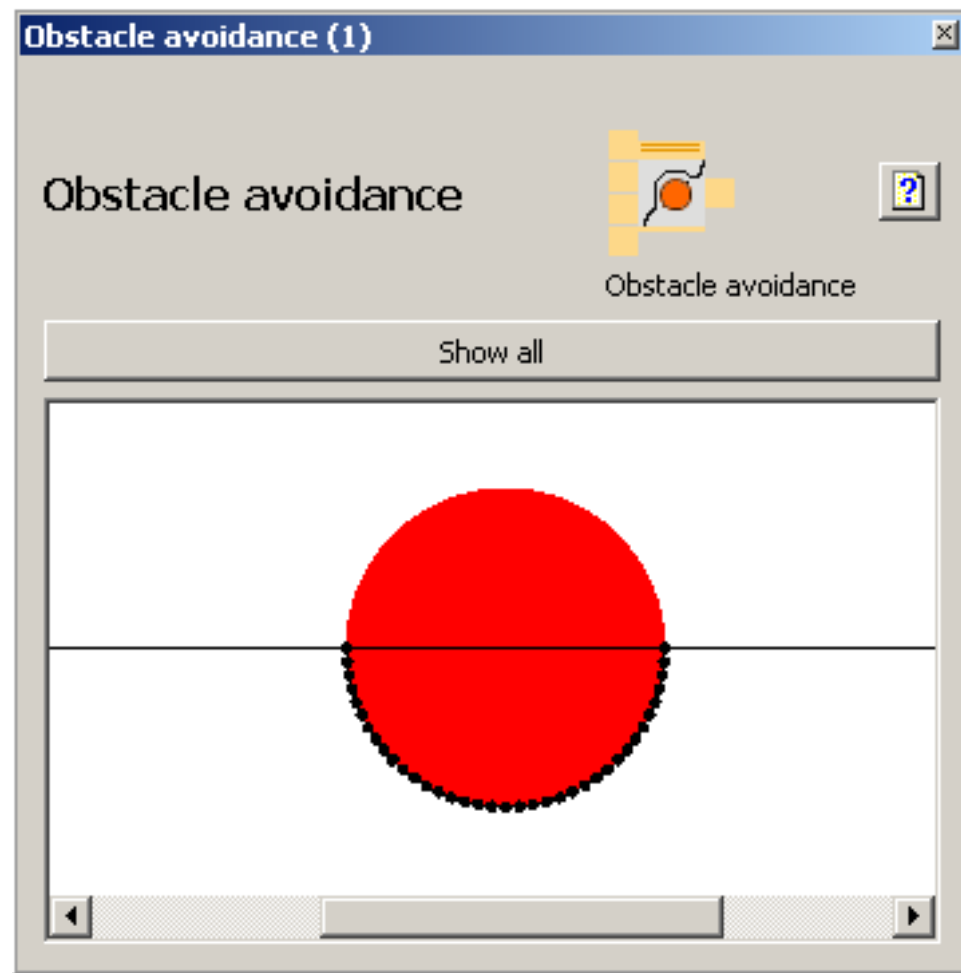
Angular Distance ->



-> Detour

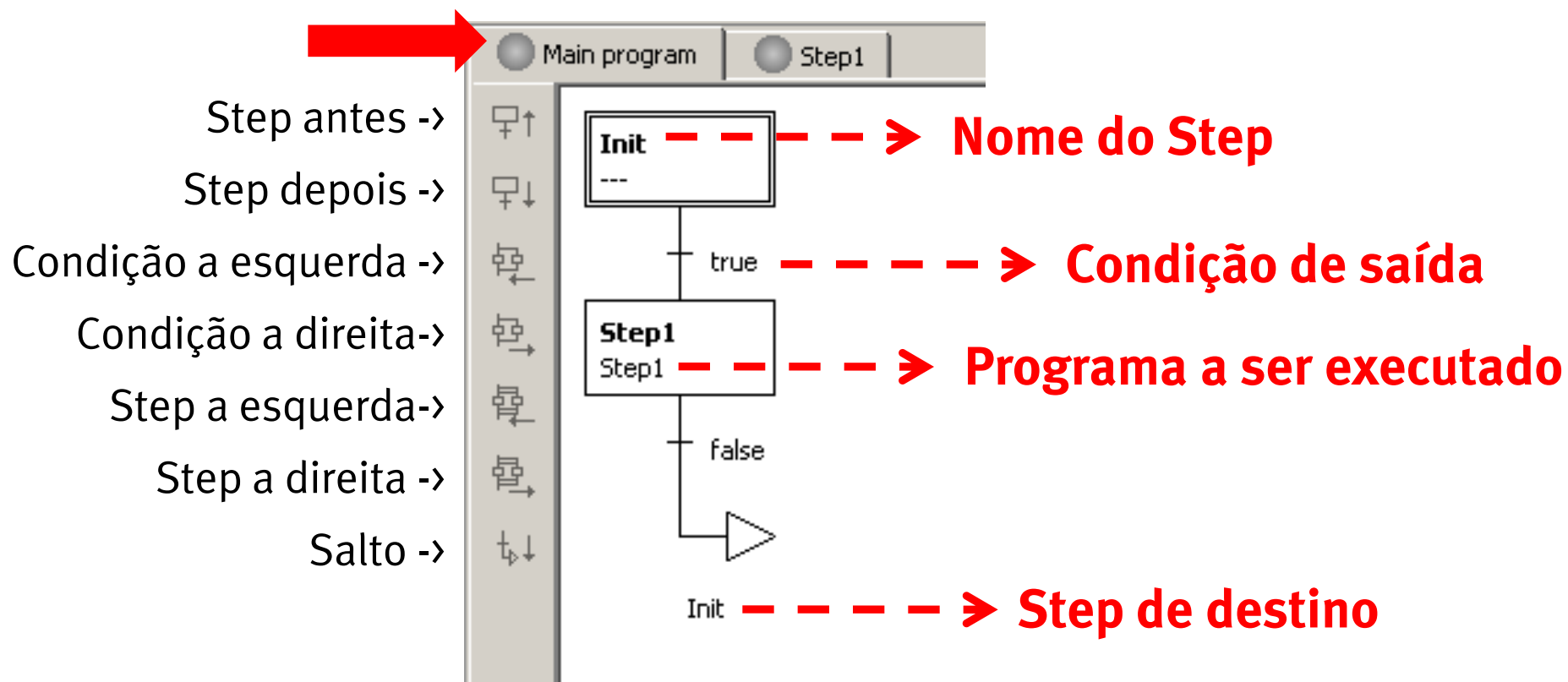
Obstacle avoidance

* **Angular Distance:** Distância angular entre os pontos do caminho de desvio.



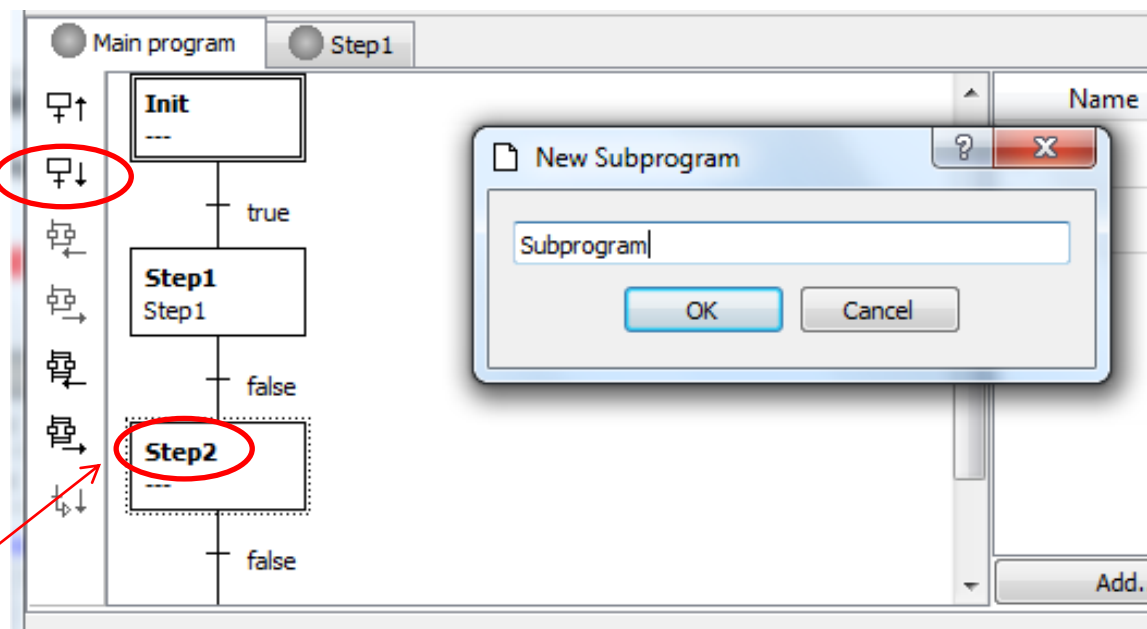
Robotino®View – Ambiente Gráfico de Programação

Programação Sequencial:



Robotino®View – Ambiente Gráfico de Programação

Insira um step



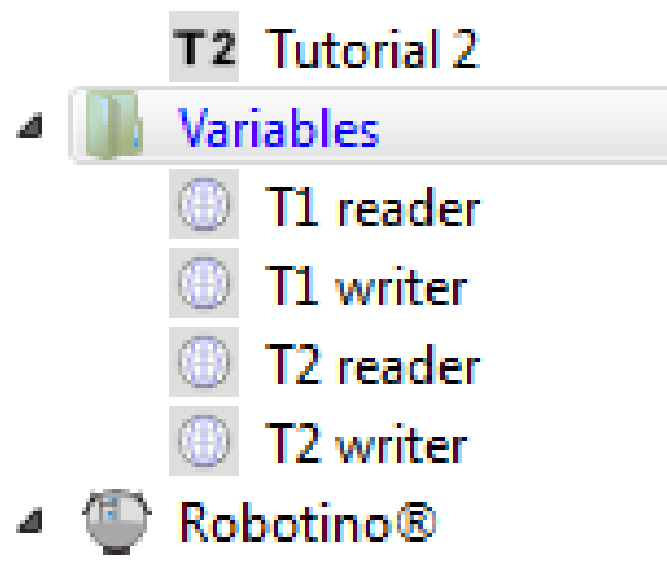
Duplo-clique para inserir
um novo subprograma

Robotino®View – Ambiente Gráfico de Programação

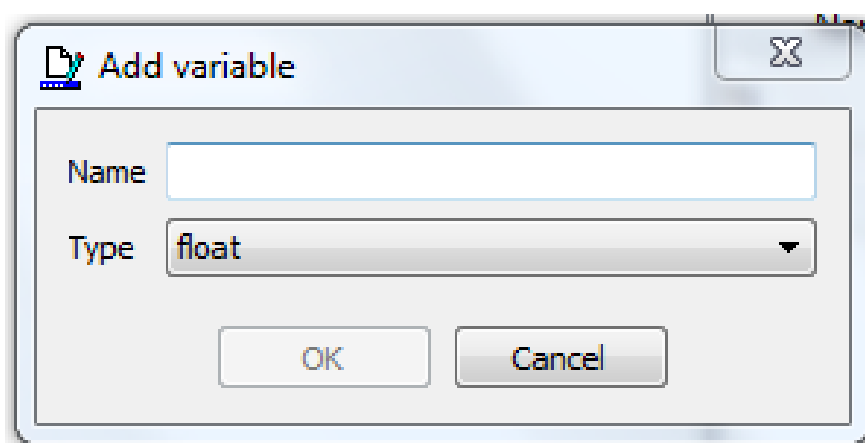
Variáveis globais podem ser lidas e escritas em qualquer subprograma.

No programa principal, elas são usadas para realizar as transições.

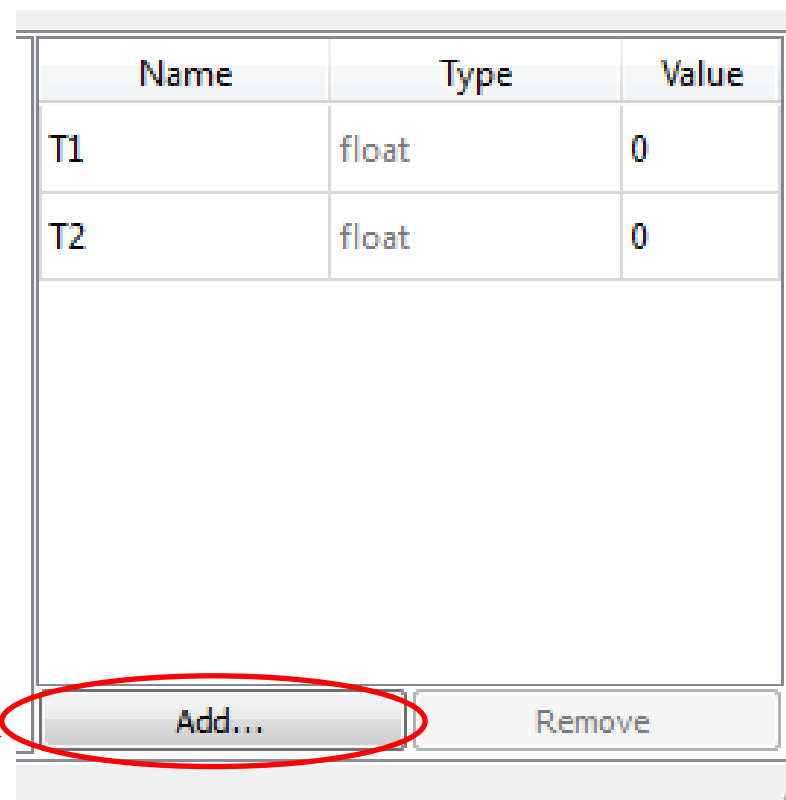
Podem ser do tipo: float, string ou pose.



Robotino®View – Ambiente Gráfico de Programação



A dialog box titled "Add variable" with a close button in the top right corner. It contains a "Name" text field, a "Type" dropdown menu currently set to "float", and "OK" and "Cancel" buttons at the bottom.



A table showing a list of variables. The "Add..." button at the bottom is circled in red, and a red arrow points from the text "Para adicionar" to it.

Name	Type	Value
T1	float	0
T2	float	0

Buttons: Add..., Remove

Para adicionar

Robotino®View – Ambiente Gráfico de Programação

EXEMPLO

