

ÍNDICE

1	Introdução – controlar modelos fischertechnik através do ROBO Pro	
	443
1.1	Instalação do ROBO Pro.....	443
1.2	Instalação do controlador USB para a interface ROBO.....	444
1.3	Primeiros passos.....	447
2	Breve teste do hardware antes da programação	451
2.1	Ligação da interface ao PC.....	451
2.2	Para que a conexão esteja correta – as configurações da interface.....	451
2.3	Se a conexão não for a correta –sem conexão à interface!.....	452
2.4	Tudo funcionando – o teste da interface.....	453
3	Level 1: O seu primeiro programa de controle	455
3.1	Criar um novo programa.....	455
3.2	Os elementos do programa de controle.....	456
3.3	Introduzir elementos do programa, deslocar e alterar.....	456
3.4	Conexão dos elementos do programa.....	459
3.5	Teste do primeiro programa de controle.....	460
3.6	Outros elementos do programa.....	462
3.6.1	Tempo de espera.....	462
3.6.2	Aguardar pela entrada.....	463
3.6.3	Contador de impulsos.....	463
3.6.4	Ciclo de contagem.....	464
3.7	Modo Online ou Download – Qual é a diferença?.....	464
3.8	Dicas e truques.....	467
4	Level 2: Trabalhar com os subprogramas	468
4.1	O seu primeiro subprograma.....	469
4.2	A biblioteca de subprogramas.....	473
4.2.1	Utilização da biblioteca.....	473
4.2.2	Utilização da biblioteca pessoal.....	473
4.3	Editar símbolos de subprogramas.....	474
5	Level 3: Variáveis, painéis de comando & Co	476
5.1	Variáveis e comandos.....	476
5.2	Variáveis e vários processos.....	478
5.3	Painéis de comando.....	478
5.4	Temporizador.....	482
5.5	Entradas de comando para subprogramas.....	483
5.6	Listas (Arrays).....	486
5.7	Operadores.....	487
6	Ativar módulos de ampliação e várias interfaces	491
6.1	Módulos de ampliação.....	491
6.2	Várias Interfaces.....	491
6.3	Alocação da interface nos subprogramas.....	493
6.4	Dicas & Truques.....	494
6.5	Alterar o número de série da interface ou da versão Firmware.....	494
7	Vista geral dos elementos do programa	496
7.1	Elementos básicos (Level 1).....	496
7.1.1	Início.....	496
7.1.2	Final.....	496
7.1.3	Ramificação digital.....	496
7.1.4	Ramificação analógica.....	497
7.1.5	Tempo de espera.....	498
7.1.6	Saída do motor.....	498
7.1.7	Saída de lâmpada (Level 2).....	499
7.1.8	Aguardar pela entrada.....	500
7.1.9	Contador de impulsos.....	501
7.1.10	Ciclo de contagem.....	501
7.2	Subprograma I/O (Level2-3).....	502
7.2.1	Entrada do subprograma (Level 2).....	502
7.2.2	Saída do subprograma (Level 2).....	502
7.2.3	Entrada de comando do subprograma (Level 3).....	502
7.2.4	Saída de comando do subprograma (Level 3).....	503
7.3	Variável, lista, ... (Level 3):.....	503
7.3.1	Variável, (global).....	503
7.3.2	Variável local.....	504
7.3.3	Constante.....	505
7.3.4	Variável do temporizador.....	505
7.3.5	Lista.....	506
7.4	Comandos (Level 3).....	508
7.4.1	= (Atribuição).....	509
7.4.2	+ (Mais).....	509
7.4.3	- (Menos).....	509
7.4.4	Direita.....	509

7.4.5	Esquerda	509
7.4.6	Parar	510
7.4.7	Ligar	510
7.4.8	Desligar	510
7.4.9	Texto	510
7.4.10	Anexar valor	510
7.4.11	Deletar valor(es).....	510
7.4.12	Trocar os valores	511
7.5	<i>Comparações, Aguardar por, ... (Level3)</i>	511
7.5.1	Ramificação (com entrada de dados)	511
7.5.2	Comparação com constante	512
7.5.3	Comparação.....	512
7.5.4	Tempo de espera	512
7.5.5	Aguardar por.....	513
7.5.6	Contador de impulsos	513
7.6	<i>Interface de Entrada/ -Saída,</i>	514
7.6.1	Entrada digital	514
7.6.2	Entrada analógica	514
7.6.3	Entrada IR:	516
7.6.4	Saída do motor.....	516
7.6.5	Saída da lâmpada	517
7.6.6	Entrada do painel de comando	518
7.6.7	Saída do painel de comando	519
7.7	<i>Operadores.....</i>	519
7.7.1	Operadores aritméticos.....	520
7.7.2	Operadores lógicos	520
8	Vista geral dos elementos de comando e painéis de comando.....	521
8.1	<i>Indicações</i>	521
8.1.1	Aparelho de medição	521
8.1.2	Indicação de texto	522
8.1.3	Lâmpada de sinalização	523
8.2	<i>Elementos de comando.....</i>	524
8.2.1	Botão.....	524
8.2.2	Regulador.....	525
9	Funções de desenho.....	526

1 Introdução – controlar modelos fischertechnik através do ROBO Pro

Com certeza já se perguntou como é que funcionam os robôs, quando executam determinadas tarefas como que por mágica. Mas não apenas com os robôs verdadeiros, nós tratamos em muitas outras áreas a técnica do controle e automatização; também na fischertechnik. Já no próximo capítulo iremos criar em conjunto um pequeno programa de controle para um portão de garagem automático e, com isso, aprender como é possível executar e testar este tipo de tarefas de controle com a ajuda do software ROBO Pro para Windows. O ROBO Pro é por isso muito fácil de utilizar. Na superfície de comando gráfico é possível criar os programas de controle, mais concretamente os organigramas e, mais tarde, os diagramas de fluxo dos dados, como vamos ver, apenas exclusivamente com a ajuda do mouse.

De maneira a que possa controlar os seus modelos fischertechnik através do PC, necessita, além do software de controle ROBO Pro, de uma interface como meio de ligação entre computador e modelo. Este converte os comandos do software de tal forma, que, por exemplo, é possível controlar motores e processar sinais de sensores. Existe a ROBO Interface, n.º de art. 93293 e a já mais antiga Intelligent Interface, n.º de art. 30402. Pode utilizar ambas as interfaces juntamente com o ROBO Pro. Além disso, o ROBO Pro suporta apenas o modo online da Intelligent Interface. A interface paralela anterior n.º de art. 30520 já não é suportada pelo ROBO Pro.

Mais algumas palavras relacionadas com a estrutura deste manual de instruções. Ele subdivide-se em duas partes. A primeira parte do Capítulo 1 ao Capítulo 4 descreve os procedimentos básicos da programação com o ROBO Pro. Lá obtém muitas informações e fundos relacionados com a programação em geral e ao modo de operação do software ROBO Pro.

A segunda parte é abrangida pelos capítulos 5 a 7 e faz uma introdução às funções para programas avançados.

Os capítulos a partir do capítulo 8 são antes de mais, material de consulta. Após ter concluído a primeira parte, estará familiarizado com a utilização do ROBO Pro, e se quiser informações mais específicas, encontrará aí a explicação detalhada relativa a cada elemento do programa.

Comecemos então imediatamente! Com certeza já está curioso relativamente às possibilidades que tem de programar o seu modelo fischertechnik através do software ROBO Pro. Divirta-se!

1.1 Instalação do ROBO Pro

As condições para a instalação do ROBO Pro são:

- um PC compatível com IBM com processador Pentium com pelo menos 600 MHz de frequência de relógio, 32 MB RAM e aprox. 20 MB de capacidade de memória livre no disco rígido
- Um monitor e uma placa gráfica com uma definição de no mínimo 1024x768 pixels. Em caso de monitores com tubos de imagem, a taxa de varrimento deve ser de, no mínimo, 85 Hertz, de modo a que se possa obter uma imagem sem brilho. As telas planas TFT proporcionam uma imagem sem brilho com qualquer taxa de varrimento, de modo que, numa tela plana TFT, a taxa de varrimento não é relevante.
- Microsoft, Windows, versão Windows 95, 98, ME, NT4.0, 2000 ou XP

- Uma interface USB livre ou uma interface RS232 livre COM1 a COM4 para a ligação da interface ROBO — n.º de art. 93293 — ou uma interface RS232 livre COM1 a COM4 para ligação da Intelligent Interface antiga — n.º de art. 30402

Em primeiro lugar tem de iniciar o computador e aguardar, até o sistema operativo (Windows) ter carregado totalmente. A interface ROBO somente deve ser conectada ao computador após uma instalação com sucesso. Insira o CD de instalação na drive de CD-ROM. O programa de instalação no CD é iniciado automaticamente.

- Pressione o botão **Continuar** na primeira janela de boas-vindas do programa de instalação.
- A segunda janela **Notas importantes** contém notas atuais importantes relativas à instalação do programa ou sobre o próprio programa. Pressione também aqui o botão **Continuar**.
- Na terceira janela **Acordos de licença** é reproduzido o contrato de licença para o ROBO Pro. Tem de aceitar o contrato de licença com **Sim**, antes de poder avançar para a próxima janela com **Continuar**.
- Introduza o seu nome na próxima janela **Informações do utilizador**.
- Na janela **Tipo de instalação** pode escolher entre a **Instalação Express** ou uma **Instalação definida pelo utilizador**. Na Instalação definida pelo utilizador pode excluir componentes individuais da instalação. Se instalou uma nova versão do ROBO Pro sobre uma versão mais antiga e alterou alguns dos programas de exemplos, pode excluir os exemplos da instalação através da instalação definida pelo utilizador. Caso contrário os programas de exemplo alterados são **substituídos sem aviso** durante a instalação. Se escolher a instalação definida pelo utilizador e pressione **Continuar**, surge uma janela adicional, na qual pode escolher os componentes.
- Na janela **Diretório de destino** pode seleccionar a pasta ou o caminho de diretório, no qual o programa ROBO Pro deve ser instalado. Geralmente este caminho é C:\Programas\ROBO Pro. No entanto, pode introduzir também um outro diretório.
- A instalação é efetuada ao pressionar **Terminar**, na última janela. Assim que a instalação está concluída — isso dura apenas alguns segundos — o programa informa que a instalação teve sucesso. Se existirem problemas, é apresentada uma mensagem de erro, que o deveria ajudar a resolver o problema.

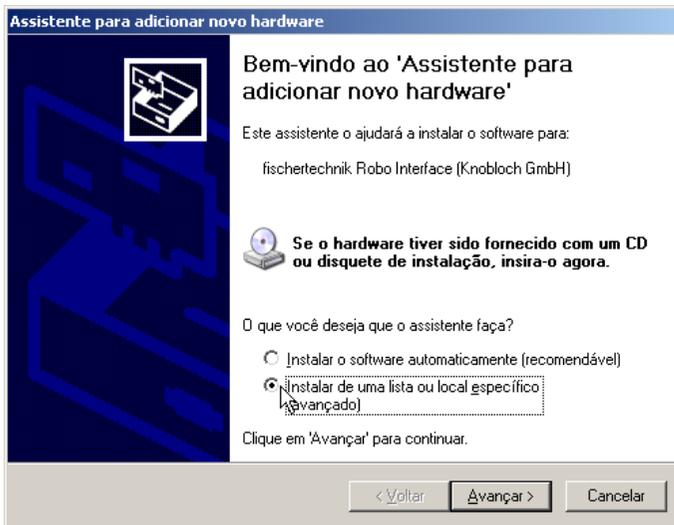
1.2 Instalação do controlador USB para a interface ROBO

Este passo só é necessário, quando a nova interface ROBO deve ser conectada à interface USB. A interface ROBO também pode ser conectada à interface serial COM1–COM4. Para a antiga Intelligent Interface este passo é desnecessário, uma vez que a Intelligent Interface apenas pode ser conectada em série. As versões do Windows mais antigas Windows 95 e Windows NT 4.0 não suportam a interface USB. Em caso de utilização do Windows 95 ou NT 4.0 a interface ROBO só pode ser conectada através da interface serial. Sendo assim, não é necessário instalar um controlador.

Nota importante para a instalação com o Windows 2000 e Windows XP:

O controlador USB só pode ser instalado por um utilizador, que possui direitos de administração do PC. Se o programa de instalação informar que não pode instalar o controlador USB, terá que pedir ao seu administrador do sistema, para instalar o controlador ou ROBO Pro sem instalar este controlador. Depois poderá conectar a sua interface mas apenas através da interface serial um pouco mais lenta.

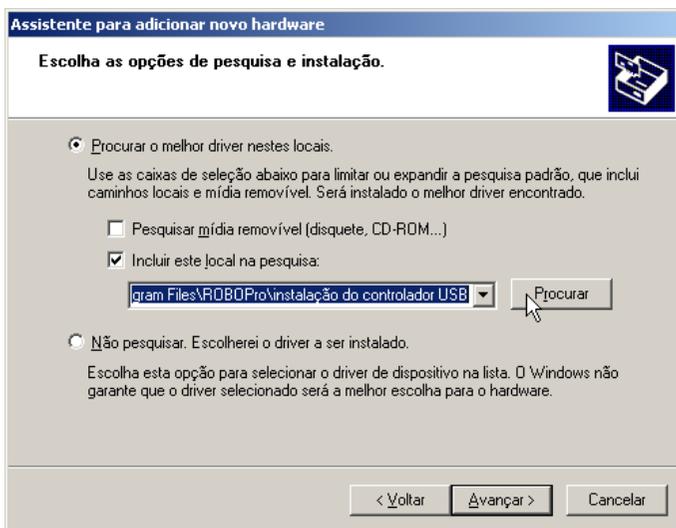
Para instalar o controlador USB tem de conectar primeiro a interface ROBO com um cabo USB ao seu computador e alimentá-lo com corrente. O Windows detecta automaticamente, que a interface está conectada e apresenta a seguinte janela:



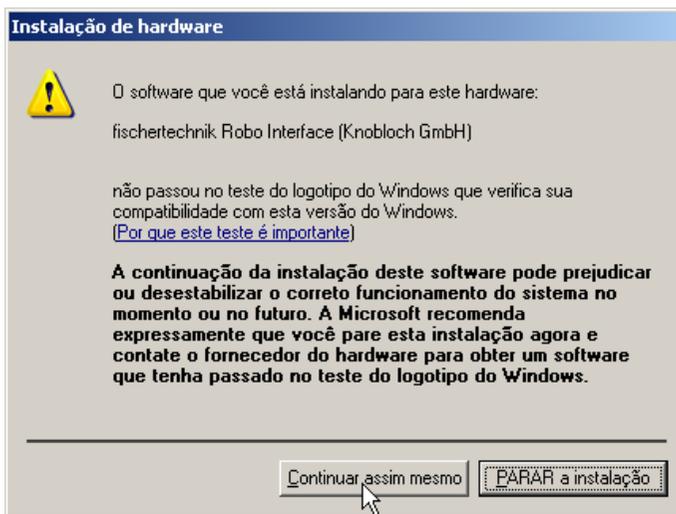
A janela poderá ter uma aparência ligeiramente diferente do apresentado, dependendo do sistema operativo!

Aqui tem de seleccionar **Instalar software a partir de uma lista ou fonte determinada** e premir **Continuar**.

Na janela seguinte desativa **Procurar meios de alternância** e ativa **Procurar a fonte seguinte**. A seguir, pressione **Procurar** e selecione o subdiretório **Instalação do controlador USB** no diretório, no qual o ROBO Pro está instalado (o diretório padrão é C:\ROBOPro):



Depois de ter pressionado em **Continuar**, surge no Windows XP provavelmente a seguinte mensagem:



O controlador USB ainda vai ser verificado pela Microsoft. Assim que a verificação estiver concluída o controlador é marcado pela Microsoft, de modo a que esta mensagem deixa de surgir. Para instalar o controlador pressione em **Prosseguir instalação**.

Finalmente, surge a seguinte mensagem:



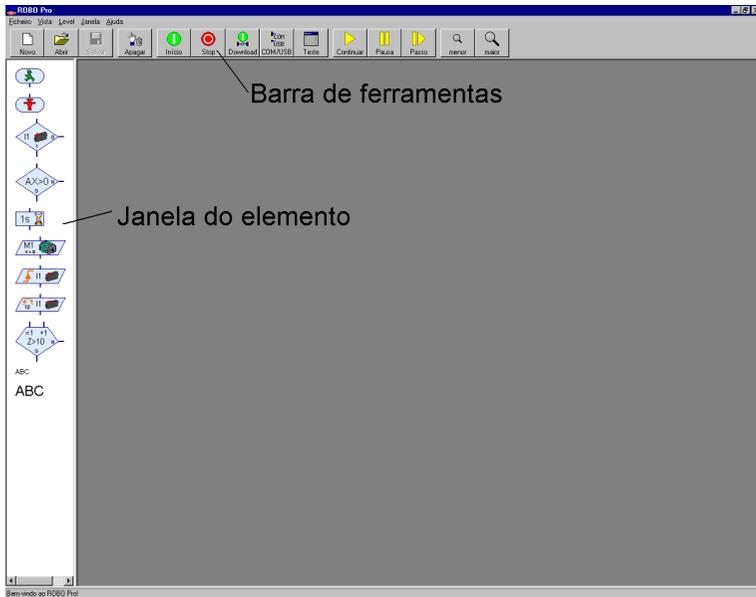
Pressione **Terminar** para concluir a instalação do controlador USB.

1.3 Primeiros passos

Curioso? Então inicie o programa ROBO Pro. Para isso clique no botão iniciar na barra de tarefas e selecione a seguir **Programas** ou **Todos os Programas** e **ROBO Pro**. Nesta pasta do menu iniciar encontra os seguintes registros:



Com o registro desinstalar pode desinstalar o ROBO Pro. O registro Ajuda abre o ficheiro de ajuda relativo ao ROBO Pro e o registro ROBO Pro abre o programa ROBO Pro. Selecione agora o registro **ROBO Pro** para iniciar o software.

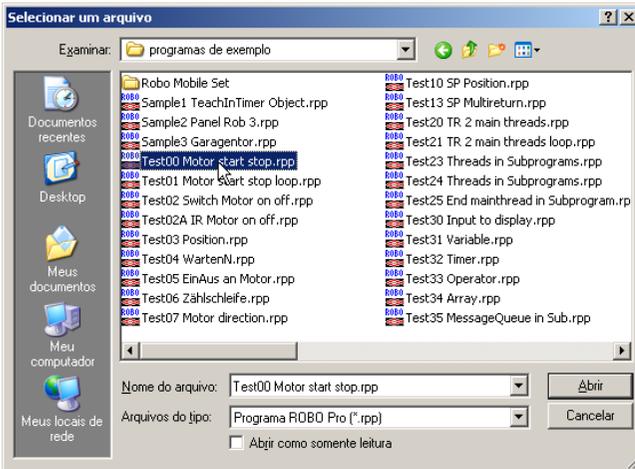


A janela tem na parte superior uma barra de menu e uma barra de ferramentas com diferentes botões de comando, bem como uma janela com elementos do programa do lado esquerdo. Se visualizar duas janelas sobrepostas, na coluna da margem esquerda, o ROBO Pro não está ajustado para o **Level 1**. Para adaptar a funcionalidade do ROBO Pro ao conhecimento crescente, pode ajustar o ROBO Pro do Level 1 para principiantes ao Level 5 para peritos. Controle agora no menu **Level**, se no **Level 1: Principiantes** se encontra um visto. Se não, mude para o Level 1.

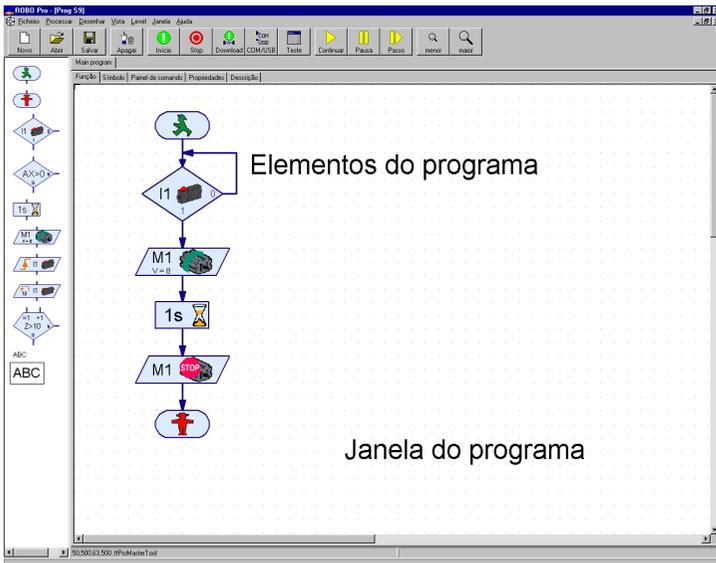
Agora tem a possibilidade de criar um novo arquivo do programa ou de abrir um arquivo do programa já existente. Pretendemos criar um novo arquivo do programa apenas no capítulo 3, quando colocarmos um novo programa de comando. Para conhecer a superfície de comando, abrimos um dos exemplos de programas já existentes. Para isso clique no menu **Arquivo** no registro **Abrir** ou utilize o botão **Abrir** na barra de ferramentas. Os arquivos de exemplo encontram-se no diretório **C:\Programas\ROBO Pro\programas de exemplo**.



Abrir



Abra o arquivo **Test00 Motor start Stopp.rpp**:



Aqui pode visualizar qual é a aparência um programa ROBO Pro simples. Os organigramas dos programas de comando são criados na janela de programas ao programar, com os elementos do programa da janela de elementos. Os organigramas terminados podem depois ser verificados e testados com uma interface fischertechnik. Mas sempre com calma, vamos conhecer a

programação passo a passo nos próximos capítulos! Depois de ter recebido uma primeira impressão da superfície de comando, feche o arquivo de programa através do comando **Terminar** novamente no menu **Arquivo**. Pode responder à consulta se pretende salvar o arquivo com **Não**.

2 Breve teste do hardware antes da programação

Para que também possamos testar os programas de comando, que vamos criar mais tarde, a interface tem de ser conectada ao PC, isso é lógico. Mas, dependendo da interface utilizada (interface ROBO, n.º de art. 93293 ou mais antiga Intelligent Interface n.º de art. 30402) o software também tem de ser ajustado e a conexão tem de ser testada de forma correspondente. Pretendemos abordar isso no seguinte capítulo.

2.1 Ligação da interface ao PC

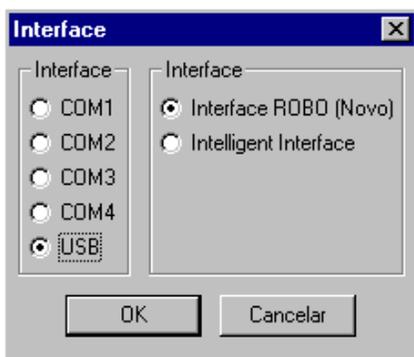
Isto não deveria ser um problema. O cabo de ligação fornecido com a interface é conectado à interface e a uma interface do PCs:

- No caso de uma interface ROBO (n.º de art. 93293) pode ser utilizada uma interface USB ou uma interface serial COM1 a COM4.
- No caso de uma interface ROBO (n.º de art. 30402) pode ser utilizada uma interface USB ou uma interface serial COM1 a COM4.

As conexões desta interface encontram-se normalmente na parte de trás do seu computador. A localização exata das diferentes conexões está descrita detalhadamente nas instruções de serviço do seu PC, por favor consulte-o. Também pode encontrar frequentemente conexões USB na parte frontal do PC. Não se esqueça de alimentar a interface com corrente (transformador ou bateria). As ligações individuais da interface estão descritas detalhadamente nas instruções de serviço da respectiva interface.

2.2 Para que a conexão esteja correta – as configurações da interface

Para que a conexão entre o PC e a interface funcione corretamente, a respectiva interface utilizada tem que ser configurada no ROBO Pro. Inicie para isso o ROBO Pro através do registo **ROBO Pro** no menu iniciar em **Programas** ou **Todos os programas** e **ROBO Pro**. Pressione depois na barra de ferramentas o botão **COM/USB**. Surge a seguinte janela:

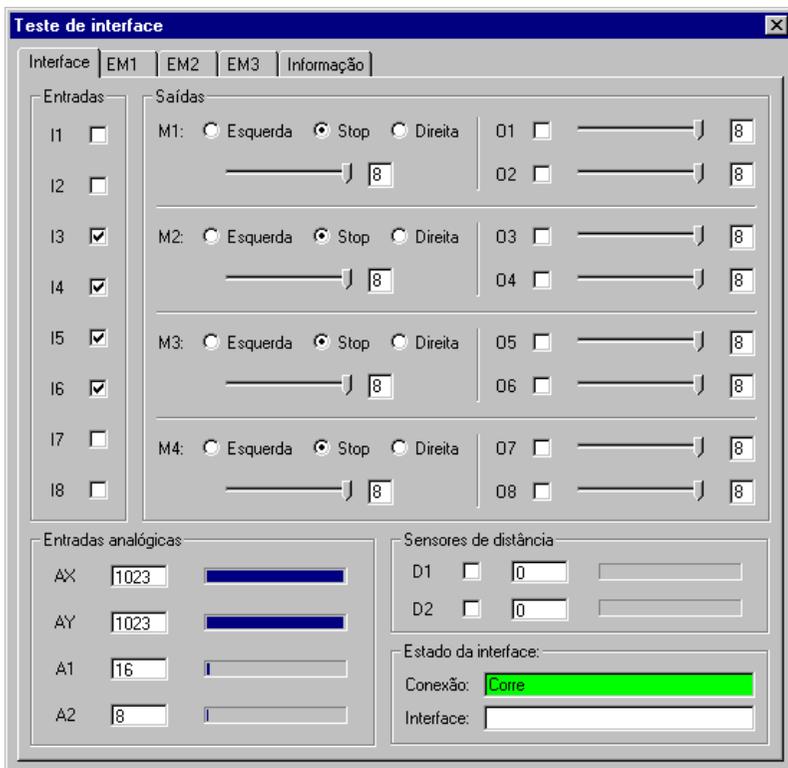


Aqui pode seleccionar tanto a interface como o tipo da interface. A Intelligent Interface suporta como já mencionado apenas as interfaces seriais COM1–COM4.



Teste

Depois de ter efetuado as configurações corretas feche a janela com OK. Abra agora a janela para o teste da interface através do botão **Teste** na barra de ferramentas:



Este apresenta as entradas e saídas existentes na interface. A coluna verde em baixo, à direita apresenta o estado de conexão do PC à interface:

- **Conexão com a interface OK** confirma uma conexão correta à interface
- **Sem conexão com a interface** indica que a conexão não foi estabelecida corretamente e que o PC não conseguiu estabelecer uma conexão à interface. A coluna aparece em cor vermelha.

Para poder alterar as configurações da interface e da conexão, tem de ser fechada a janela de teste (com o X em cima, à direita) e, tal como descrito anteriormente, selecionar uma outra interface através do botão COM/USB na barra de ferramentas.

Depois de configurar a conexão do PC e da interface e de aparecer a coluna verde na janela de **Teste**, pode passar para o capítulo seguinte.

Se não for este o caso, talvez possa recorrer às dicas do próximo capítulo.

2.3 Se a conexão não for a correta –sem conexão à interface!?

Se no ROBO ou Intelligent Interface, apesar da interface serial COM (interface em cima) corretamente configurada, surge a mensagem **Sem conexão à interface**, deve verificar os

seguintes pontos. Eventualmente, pode ter que contar com a ajuda de um "perito em computadores":

- **Alimentação de corrente:**
A interface está sendo corretamente alimentada com corrente? Se estiver utilizando pilhas ou bateria para alimentação de corrente, então pode acontecer que a bateria gasta já não fornece corrente suficiente. Se a corrente da bateria desce para menos de 6 V, o processador da interface do ROBO deixa de funcionar. Neste caso, acende o diodo luminoso vermelho durante algum tempo ou não acende qualquer diodo luminoso. Quando a corrente está em condições, alguns dos diodos luminosos verdes piscam. Na Intelligent Interface anterior, não é possível determinar se a corrente é suficiente para o processador simplesmente através dos diodos luminosos. Quando a corrente é insuficiente, tem de carregar novamente a bateria ou utilizar pilhas novas, ou então, se possível, testar a interface com um transformador.
- **A interface funciona de todo?**
Pode saber isto testando um outro aparelho serial, como por exemplo um modem externo, na interface.
- Há um conflito com um outro controlador de aparelho na mesma interface (por ex. modem)? Eventualmente, este controlador pode ter de ser desativado (ver manual do Windows ou dos aparelhos).
- Apenas para Windows NT/2000/XP e o Intelligent Interface anterior: Se um Intelligent Interface anterior, na inicialização do PC, já está conectada ao computador e à alimentação de corrente, é infelizmente comutada do Windows NT para o modo Download. Para estabelecer novamente a conexão ao PC tem apenas de interromper por alguns instantes a alimentação de corrente na interface. Isso não é possível na nova interface ROBO.
- Se ainda não conseguir estabelecer qualquer conexão com a interface, a interface ou o cabo de conexão têm provavelmente uma avaria. Nesse caso, contacte o serviço de assistência fischertechnik (endereço: ver Menu: „?“ / **Informação sobre**).

2.4 Tudo funcionando – o teste da interface

Depois da conexão estar corretamente configurada, podemos testar, com ajuda do teste da interface, a própria interface e os modelos que estão conectados à interface. Como já mencionado, a janela de teste mostra as várias entradas e saídas da interface:



Teste

- **Entradas digitais I1–I8**
I1–I8 são as entradas digitais da interface. São ligados aqui os chamados sensores. As entradas digitais só podem assumir os estados 0 e 1 ou Sim e Não. Podem ser ligados nas entradas digitais, como sensores, interruptores (mini-botão) mas também fototransistores (sensores de luz) ou contatos reed (sensores magnéticos).
Pode testar o funcionamento destas entradas, ligando na interface um mini-botão ao I1, por exemplo (n.º de art. 37783). (use no botão os contatos 1 e 3). Logo que prime o botão, surge um visto na indicação do I1. Se ligou o botão noutro ponto (contatos 1 e 2), surge imediatamente o visto e desaparece quando se pressionar o botão
- **Saídas do motor M1–M4**
M1 – M4 são as saídas da interface. São conectados aqui os chamados atuadores. Estes podem ser, por ex., motores, eletroímãs ou lâmpadas. As 4 saídas do motor podem ser comandadas em termos de velocidade a 8 níveis e de direcção. Para o comando da velocidade é usado o controle deslizante. Para além disso, é indicada a velocidade em

números ao lado do controlador deslizante. Se desejar testar uma saída, conecte um motor a esta saída, por ex. M1.

- **Saídas das lâmpadas O1–O8**

As saídas do motor podem ser alternativamente utilizadas como saídas independentes. Desta forma, é possível ativar lâmpadas, e também motores que apenas se podem deslocar numa direção, (por ex. cinta transportadora). Se desejar testar estas saídas, conecte uma conexão da lâmpada à saída, por ex. O1. Conecte a outra conexão da lâmpada à tomada da massa da interface (\perp).

- **Entradas analógicas AX–AY**

As entradas analógicas AX e AY medem a resistência do sensor conectado. Podem ser conectados aqui resistores NTC para medição da temperatura, potenciômetro, fotoresistor ou fototransistores.

- **Entradas analógicas A1–A2**

Estas duas entradas medem uma corrente de 0–10V.

- **Sensores de distância D1–D2**

Apenas podem ser conectados sensores de distância especiais às entradas de sensores de distância D1 e D2. Existem sensores de distância D1 e D2 para entradas digitais e para entradas analógicas.

- **Módulos de extensão EM1–EM3**

Quando os módulos de extensão estão conectados à interface (até três extensões I/O nr.º de art. 93294 na ROBO Interface, mas no máximo um módulo de extensão n.º de art. 16554 no caso do Intelligent Interface), estes podem ser ativados ligando através do registro no lado superior dos módulos de ampliação.

3 Level 1: O seu primeiro programa de controle

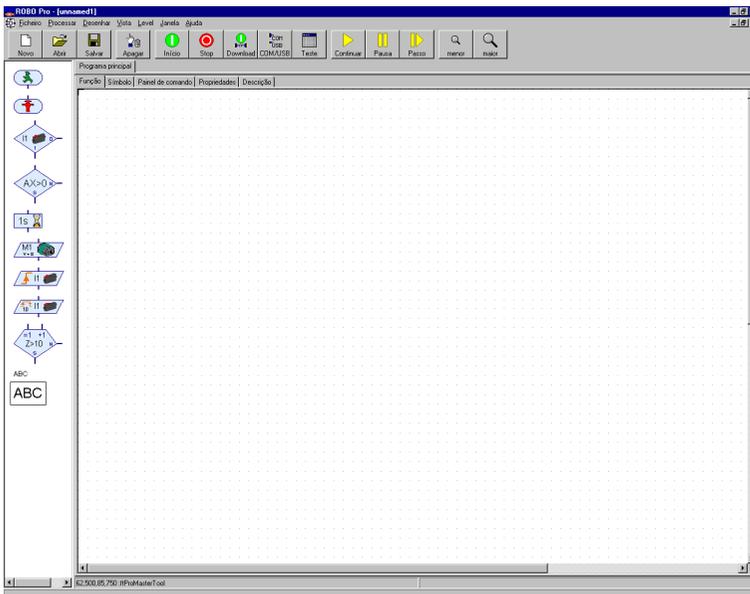
Depois de ter testado no último capítulo o hardware, ou seja, a interface e os interruptores e motores a ela ligados, vamos agora tratar da programação. Mas o que significa realmente "Programação"? Imagine agora que está conectado, por ex., um robô à nossa interface. Este robô é contudo tão bobo que não consegue funcionar sozinho. Felizmente, somos um pouquinho mais espertos. Podemos dizer ao robô exatamente aquilo que ele tem de fazer. Como? O que aconteceu quando, no capítulo anterior, colocamos com o botão do mouse a saída do motor M1 em „Esquerda“? Isso, nós ligamos o motor. Se este motor, por ex., movesse as pinças do nosso robô, teríamos dito ao robô simplesmente o seguinte: "Agarra esse objeto!" Nós não queremos realizar cada passo manualmente, o robô deve fazê-lo "automaticamente". Para isso, temos de memorizar cada passo individual, de forma a que o robô os possa realizar uns após aos outros, por isso, temos de criar um programa que ative o robô em nosso lugar. Em linguagem técnica, isso se chama logicamente um programa de controle.

3.1 Criar um novo programa

Com o software ROBO Pro temos nas nossas mãos uma ferramenta fantástica, para lançar programas de controle desse género e testá-los com a ajuda de uma interface conectada. Não tenha medo, não pretendemos programar imediatamente um robô. Para já, ficamos satisfeitos com a realização de trabalhos de controle simples. Para isso, temos de criar um novo programa. Na caixa de ferramentas pode encontrar o registro „Novo“. Quando clica nesse registro com o botão esquerdo do mouse, é criado um novo programa vazio:



Novo



Agora pode ver uma grande superfície de desenho branca, na qual pode introduzir a seguir o seu primeiro programa. Se na margem esquerda estiverem apresentadas duas janelas uma por cima da outra, coloque no menu o **Level** em **Level 1**: para **Principiantes**.

3.2 Os elementos do programa de controle

Podemos agora tratar de criar o nosso primeiro programa. Vamos fazer isto com a ajuda de um exemplo concreto:

Descrição do funcionamento:

Imagine um portão de garagem que é aberto automaticamente. Talvez tenha até um destes portões em sua casa! Quando o carro se aproxima da garagem, basta pressionar um botão no comando e o portão da garagem, é aberto através de um motor. O motor tem de trabalhar até o portão da garagem ser completamente aberto.

Mas é bastante complicado descrever em palavras um processo de comando. Por isso, se utiliza, para a representação da sequência de ações e condições necessárias a este processo, os chamados **organigramas**. A condição para a ação "Ligação do Motor" no caso de este comando, é que o botão seja premido. A leitura de um organigrama desse gênero é muito fácil: Seguir sempre os passos de acordo com as setas! Estas indicam de forma precisa o modo de funcionamento do comando – os passos individuais só podem ser realizados de acordo com a sequência indicada pelas setas, nunca de outra forma. Caso contrário, não precisaríamos de ter todo este trabalho – certo?

Com a ajuda do nosso software ROBO Pro podemos elaborar de forma precisa este organigrama e, dessa forma, criar o **programa de controle** para o hardware conectado (interface, motores, interruptores, etc.). O resto é assumido pelo software, o mesmo que acontece nos casos de grandes utilizações industriais! Dessa forma podemos nos concentrar totalmente na elaboração do organigrama.

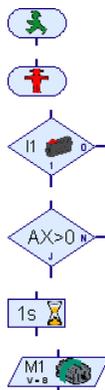
O organigrama é composto a partir dos elementos do programa. Outro termo novo? Muito mais fácil! No ROBO Pro os elementos individuais, com os quais é criado o organigrama, são chamados elementos do programa. A ação "Ligar motor" quer dizer simplesmente que a interface deve ligar este motor que está ligado à interface! Os elementos de programa disponíveis podem ser encontrados na janela do elemento na margem esquerda.

3.3 Introduzir elementos do programa, deslocar e alterar

Agora trataremos de criar o organigrama para o nosso comando de portão de garagem com os elementos do programa que se encontram na janela do elemento. Todos os elementos do programa disponíveis podem ser obtidos na janela do elemento e introduzidos na janela do programa.

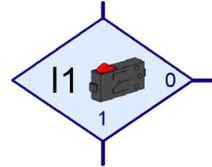
Introdução de elementos do programa

Para isso, tem de mover o mouse para o símbolo do elemento do programa desejado e clicar uma vez sobre ele com o botão esquerdo do mouse. Depois mova o mouse para a janela do programa (é a grande superfície branca) para o ponto desejado e clique novamente. Pode também deslocar o elemento do programa com o botão do mouse pressionado da janela do elemento para a janela do programa. Um programa começa sempre com um elemento inicial. O elemento inicial é o elemento



arredondado com o semáforo pedestre em movimento. O melhor é tentar logo com este elemento do programa: Clique uma vez com o botão esquerdo do mouse sobre o elemento inicial na janela do programa, mova o mouse em cima para a janela do programa e clique lá uma vez com o botão esquerdo do mouse.

A seguir, no organigrama do programa um elemento que pede uma entrada e, em função do estado da entrada, se ramifica num ou noutro caminho. Clique, na janela do elemento, sobre o elemento representado à direita e mova depois o mouse para debaixo do elemento inicial anteriormente introduzido. Quando a entrada superior do elemento de ramificação se encontra um ou dois pontos de encaixe abaixo da saída do elemento inicial, surge na janela do programa uma linha de conexão. Se clicar mais uma vez no botão esquerdo do mouse, é introduzido o elemento de ramificação e conectado automaticamente ao elemento inicial.



Deslocação de elementos do programa e de grupos

Um elemento do programa pode ser deslocado, também após a introdução, para o ponto desejado com o botão esquerdo do mouse pressionado. Se desejar deslocar vários elementos ao mesmo tempo, pode criar um quadro envolvendo os elementos, com o botão esquerdo do mouse pressionado. Para isso, tem de clicar com o botão esquerdo do mouse numa área **vazia**, manter o botão pressionado e formar com o mouse um retângulo que contenha os elementos desejados. Os elementos no retângulo são representados agora com uma margem vermelha. Se desloca um dos elementos a vermelho com o botão esquerdo do mouse, todos os elementos são agora deslocados em conjunto. Também pode marcar a vermelho elementos isolados, clicando com o botão esquerdo do mouse sobre os elementos, com a tecla de maiúsculas pressionada. Se clicar com o botão esquerdo do mouse numa área vazia, todos os elementos marcados a vermelho são novamente apresentados de forma normal.

Copiar elementos do programa e grupos

Existem duas formas de copiar os elementos do programa. Pode utilizar exatamente o mesmo procedimento utilizado para a deslocação, mas antes de deslocar os elementos, clique a tecla **CTRL** no teclado. Desta forma, os elementos não são deslocados mas sim copiados. Através desta função, pode copiar elementos, mas apenas dentro do programa. Se desejar copiar elementos de um programa para outro, pode utilizar a **área de transferência** do Windows. Selecione apenas alguns elementos, tal como descrito no capítulo anterior sobre deslocação de elementos. Se clicar agora **CTRL+C** no teclado, ou se chamar o item de menu **Processar / Copiar**, são copiados todos os elementos selecionados para a área de transferência do Windows. Pode agora mudar para um outro programa e introduzir de novo nesse programa os elementos com **CTRL+V** ou **Processar / Introduzir**. Pode também introduzir várias vezes, os elementos copiados uma vez. Se desejar deslocar elementos de um programa para outro, pode utilizar no início, em vez de **CTRL+C** ou **Processar / Copiar**, a função **CTRL+X** ou **Processar / Cortar**.

Apagar elementos e função Anular

Também é muito fácil apagar os elementos. Pode apagar todos os elementos marcados a vermelho (ver capítulo anterior), clicando a tecla Delete (**Del**) no teclado. Também pode apagar elementos individuais com a função Apagar. Para isso, clique primeiro sobre o botão representado na barra de ferramentas e depois sobre o elemento que pretende apagar. Experimente-o de imediato. Pode depois voltar a desenhar o elemento apagado. Para recuperar o elemento



apagado, pode também utilizar a função **Anular** no Menu **Processar**. Através deste item de menu pode anular todas as alterações no programa.

Processar as propriedades dos elementos do programa

Se clicar com o botão **direito** do mouse sobre um elemento do programa, na janela do programa, surge uma janela de diálogo, onde pode alterar as propriedades dos elementos. A janela Propriedades para um elemento de ramificação está representado à direita.

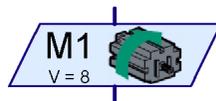
- Com os botões **I1** até **I8** pode inserir que entrada da interface é consultada.
- A seleção **interface / extensão** só é explicada no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** você pode seleccionar uma imagem para sensor conectado na entrada. As entradas digitais na maioria das vezes são utilizadas com botões mas também com frequência com fototransistores ou contatos Reed.
- Em **Trocar as ligações 1/0** pode trocar a posição das saídas 1 e 0 da ramificação. Normalmente a saída 1 encontra-se em baixo e a saída 0 do lado direito. Mas, torna-se mais prático quando a saída 1 se encontra do lado direito. Pressione **trocar as ligações 1/0**, para que seja efetuada a troca das ligações 1 e 0 tão logo se feche a janela com OK.



Nota: Se ligar um mini-botão como contato de trabalho na ligação 1 e 3 do botão, a ramificação do programa vai para o ramo 1 quando o interruptor está a ser pressionado, senão vai para o ramo 0.

Se ligar um mini-interruptor como contato de rutura na ligação 1 e 2 do botão, a ramificação do programa vai para o ramo 1 quando o interruptor não está a ser pressionado, senão vai para o ramo 0.

O próximo elemento de programa no nosso comando do portão da garagem é um elemento do motor. Introduza-o no programa, como nos dois elementos anteriores, nomeadamente sob o elemento de ramificação. Será melhor voltar a posicionar o elemento, de forma a ficar automaticamente conectado com o elemento acima.



Com o elemento do motor pode ligar ou desligar um motor, uma lâmpada ou um eletroimã. Pode voltar a abrir a janela de propriedades para o elemento do motor com um clique direito no mouse sobre o elemento.

- Através dos botões **M1** a **M4**, pode selecionar qual a saída da interface que deve ser ativada.
- Em **Imagem** pode selecionar uma imagem que representa o componente fischertechnik conectado à saída.
- A seleção **interface / extensão** só é explicada no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Ação** pode acionar de que maneira a saída deve ser influenciada. Um motor pode iniciar ou parar, com o sentido de rotação esquerdo ou direito. Pode ligar ou desligar uma lâmpada.
- Em **Velocidade/intensidade** pode ajustar com que velocidade o motor deve rodar ou com que intensidade a lâmpada deve brilhar. Os valores possíveis se situam entre 1 e 8.

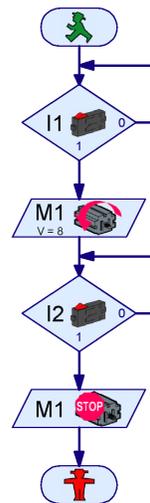
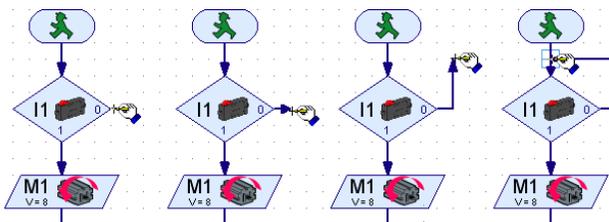


Para o organigrama do nosso programa é necessário o comando **Motor M1 esquerda com velocidade 8**.

3.4 Conexão dos elementos do programa

Depois de saber como se introduz elementos num programa de controle, podemos tratar de concluir o nosso programa de controle. Pense novamente na descrição do funcionamento do comando do portão da garagem – não falta qualquer coisa? Isso, na realidade nós ligamos o motor pressionando o botão, mas depois do portão estar aberto, ele tem de ser desligado de novo automaticamente! Na prática, isto é efetuado com um chamado interruptor de fim de curso. É um botão, que está aplicado no portão da garagem, que é acionado depois do motor ter aberto totalmente o portão. E tal como com a ligação do motor, este sinal pode ser utilizado para desligar novamente o motor. Para a consulta do interruptor de fim de curso podemos utilizar novamente o elemento de ramificação.

Introduza ainda outro elemento de ramificação no seu programa de controle, que consulte o interruptor de fim de curso na saída I2. Não se esqueça de clicar com o botão direito do mouse sobre o elemento, e de mudar a entrada para I2. Logo que o portão da garagem esteja aberto e o interruptor de fim de curso esteja acionado, o motor deve parar. Isto consegue-se, através do elemento do motor. Utilize primeiro o mesmo elemento utilizado para a ligação do motor. Se clicar com o botão direito do mouse sobre o elemento, pode alterar a função do elemento em **Parar motor**. O programa é concluído através de um elemento final. O seu programa deve estar apresentado do lado direito. Se colocou os elementos uns após os outros, sempre com uma distância de um ou dois pontos de encaixe, então a maior parte das entradas e saídas já estão conectadas entre si através das setas de curso do programa. A saída Não (N) de ambas as ramificações, ainda não está, contudo, ligada. Enquanto o botão na entrada I1 não for pressionado, o programa deve retroceder e consultar novamente o interruptor. Para desenhar esta linha, clique com o mouse sobre o pontos apresentados embaixo da imagem.



Nota: Se alguma linha não estiver corretamente conectada com outra linha, ou com uma conexão, isso é indicado através de um retângulo verde na ponta da seta. Neste caso, deve estabelecer novamente a ligação através da deslocação da linha, ou apagando e desenhando de novo. Caso contrário, o processo do programa não funcionará neste ponto.

Apagar as linhas de curso do programa

O processo para apagar linhas é exatamente igual ao processo de apagar elementos do programa. Clique simplesmente com o botão esquerdo do mouse sobre a linha, para que fique marcada a vermelho. Pressione o botão Delete (**Del**) no teclado, para apagar a linha. Pode também selecionar várias linhas, se mantiver pressionada a tecla de maiúsculas e se clicar depois com o botão esquerdo do mouse sobre as linhas umas após às outras. Além disso, para marcar várias linhas, pode também criar um quadro envolvendo todas as linhas. Agora pode apagar de uma vez todas as linhas marcadas a vermelho através da tecla **DEL**.

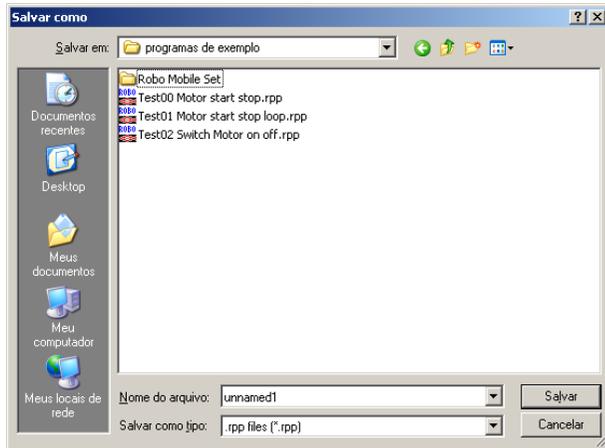
3.5 Teste do primeiro programa de controle

Para poder testar o nosso primeiro programa de controle, é necessário construir um pequeno modelo. Para isso basta ligar um botão na interface I1 e I2 e um motor ao M1.

Nota: A conexão da interface ao PC e a configuração da interface já foi efetuada no capítulo anterior e pode voltar a ser consultada.

Antes de testar o programa de controle, é necessário salvar o arquivo do programa no disco rígido do computador. Clique com o mouse no comando **Salvar como** no menu **Arquivo**. Surgirá a seguinte janela de diálogo:

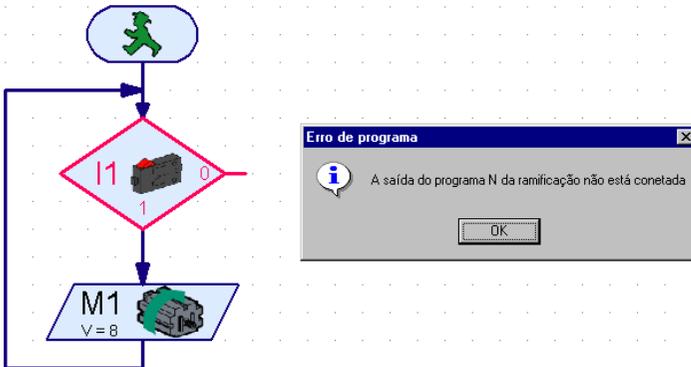
Selecione em "Salvar em" o diretório em que pretende salvar o programa. Atribua em "Nome do arquivo" um nome ainda não atribuído, por exemplo GARAGENTOR e confirme com um clique em "Salvar" com o botão esquerdo do mouse.



Para testar o programa, clique o botão de início à esquerda na barra de ferramentas. Em primeiro lugar, o ROBO Pro testa se todos os elementos do programa estão conectados corretamente. Se um elemento ou outra coisa qualquer não estiver em condições, fica selecionado a vermelho, e é indicada uma mensagem de erro descrevendo o que não está correto. Por exemplo, se tiver esquecido de ligar a saída Não (N) da ramificação do programa, este terá este aspecto:



Início

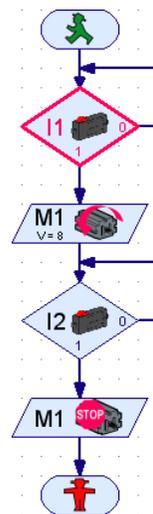


Se visualizar uma mensagem de erro, deverá ser efetuada a correção no local sinalizado. Caso contrário, o programa não é iniciado.

Nota: Pode encontrar um esclarecimento pormenorizado sobre este tipo de funcionamento e sobre o tipo de funcionamento "Modo Download" no capítulo 3.7 na página 464.

O primeiro elemento de ramificação é marcado a vermelho. O processo aguarda, neste elemento do programa, por uma ocorrência, nomeadamente a pressão do botão em I1 que deve abrir o portão da garagem. Enquanto o botão não for pressionado, o programa ramifica para a saída Não (N) da ramificação do programa e volta a ir para o início da ramificação. Clique agora o botão que está conectado à entrada I1 da interface. Assim está cumprida a condição para avançar e o motor é ligado. O processo aguarda no passo seguinte que o interruptor fim de curso seja pressionado na entrada I2. Logo que o interruptor fim de curso accione em I2, o programa ramifica para a segunda saída do motor e volta a ligar o motor. Por fim, atinge - se o final do programa. Surge a mensagem de que o programa foi terminado.

Deu tudo certo? Parabéns! Criou e testou o seu primeiro programa de controle. Se não funcionar não desanime, verifique tudo mais uma vez, certamente que existe em algum lugar um pequeno erro escondido. Todos os programadores erram e, geralmente, aprende-se com os erros. Por isso, coragem!!!



3.6 Outros elementos do programa

Quando tiver experimentado o seu primeiro programa de controle num modelo de portão de garagem correto, o portão encontra-se aberto. Como é que voltará a fechar? Claro que podemos voltar a pôr o motor a trabalhar, pressionando um botão! Mas queremos testar uma outra solução e, nesse caso, conhecer um novo elemento do programa. Para isso, salva, em primeiro lugar, o programa com um outro nome (necessitaremos novamente do organigrama atual mais tarde). Depois, utilize o item do menu **Salvar como ...** no menu **Arquivo** e introduza aí um nome de arquivo ainda não utilizado.

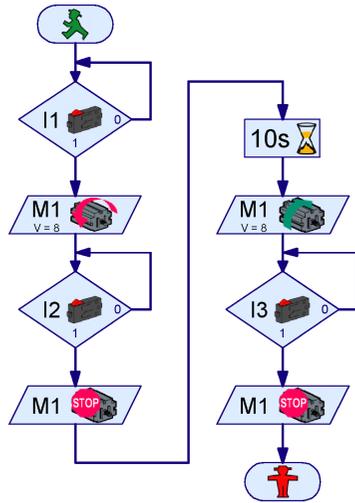
3.6.1 Tempo de espera

Antes de podermos expandir o organigrama, tem de apagar a conexão entre “desligar motor” e “fim do programa” e deslocar para baixo o elemento final. Pode inserir o novo elemento do programa entre ambos os elementos. O portão da garagem deve ser fechado automaticamente após 10 segundos. Para isso, pode utilizar o elemento do programa **Tempo de espera** representado à direita. Pode ajustar o tempo de espera para limites alargados, clicando normalmente com o botão direito do mouse no elemento. Insira o tempo de espera desejado de 10 segundos. Para fechar o portão da garagem, o motor tem de funcionar naturalmente na outra direção, ou seja, para a direita. O motor é desligado por um interruptor fim de curso em I3.

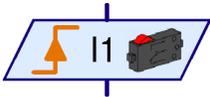




O seu organigrama pronto deve ser semelhante ao que está representado à direita. Para uma melhor representação, os novos elementos do programa estão deslocados para a direita. Se no organigrama não existir qualquer tipo de erro, pode testar o controle alargado do portão da garagem com o botão de **Início**. Ao ativar o botão em I1 é ligado o motor, ao ativar o I2 volta a ser desligado. Com isso, o portão da garagem é fechado. Agora, o elemento do programa Tempo de espera é rodeado a vermelho por 10 segundos, é o nosso tempo de espera configurado. A seguir, o motor é conectado em outro sentido de rotação até o botão em I3 ser acionado. Tente mais uma vez alterar o tempo de espera.



3.6.2 Aguardar pela entrada



Além do elemento Tempo de espera, existem ainda outros dois elementos que aguardam até prosseguir com a execução do programa. O elemento **Aguardar pela entrada** representado à esquerda aguarda até à

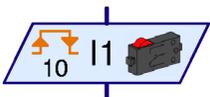
entrada da interface ter um determinado estado ou se alterar para um determinado tipo. Há 5 variantes deste elemento:

Símbolo					
Aguardar por	Entrada=1 (fechada)	Entrada=0 (aberta)	Troca 0-1 (abrir depois de fechada)	Troca 1-0 (fechado depois de aberto)	Troca qualquer (1-0 ou 0-1)
Mesma função apenas com ramificação					



Para isso, pode também utilizar uma combinação de elementos de ramificação, mas com o elemento **Aguardar pela entrada** se torna mais simples e mais compreensível.

3.6.3 Contador de impulsos

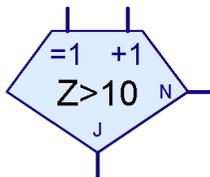


Muitos modelos robóticos da Fischertechnik utilizam também rodas dentadas movidas por impulsos. Estas rodas dentadas acionam um botão, em cada rotação, 4 vezes. Com essas rodas movidas por impulsos pode ligar um motor numa quantidade definida de rotações, em vez de um determinado tempo. Para isso, tem de contar a

quantidade de impulsos na entrada do interface. Com este objetivo existe o **elemento de**

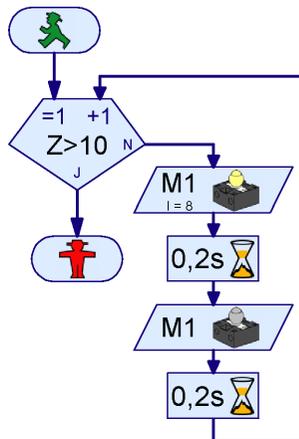
contador de pulsos apresentado à esquerda, que aguarda por uma quantidade configurada de impulsos. Também no caso deste elemento pode configurar se são avaliadas como impulsos, alterações ou apenas trocas 0-1 ou apenas 1-0. No caso de rodas movidas por impulsos se aguarda, geralmente, por alterações de forma que em 4 dentes se atinja uma resolução de 8 passos por rotação.

3.6.4 Ciclo de contagem



Com o elemento do ciclo de contagem pode executar, várias vezes, de forma muito simples, uma determinada parte do programa. O programa representado liga, por exemplo, uma lâmpada em **M1** 10x e volta a desligar. O elemento do ciclo de contagem

tem um contador incorporado. Quando o ciclo de contagem é ativado em **=1** entrada, o contador é colocado em 1. Quando o ciclo de contagem é ativado em **+1** entrada, o contador é contado em 1. Conforme o contador seja ou não maior do que um valor indicado por si, o ciclo de contagem se ramifica para a saída Sim (**S**) ou para Não (**N**). A saída Sim também é utilizada quando o ciclo é executado tantas vezes quantas as que indicou no valor do contador. Se ainda forem necessários outros ciclos, o ciclo de contagem ramifica-se numa saída Não. Como acontece no elemento de ramificação, pode trocar também a saída Sim e o Não peça janela Propriedades.



3.7 Modo Online ou Download – Qual é a diferença?



Início

Até aqui, testamos os nossos programas de controle no **Modo Online** indicado. Pode seguir o processo do programa na tela, porque o elemento ativo foi marcado a vermelho na tela. Utilize o modo Online para entender programas ou procurar erros nos mesmos.



Pausa

No modo Online pode também interromper e prosseguir com o programa, apertando o botão **Pausa**. É muito prático, quando deseja pesquisar alguma coisa no seu modelo, sem parar totalmente o programa. Mesmo quando tenta executar um programa, a função Pausa pode ser uma grande ajuda.



Passo

Com o botão **Passo** pode executar o programa em passos individuais elemento por elemento. Cada vez que pressiona o botão Passo, o programa passa para o próximo elemento do programa. Quando executa um elemento **tempo de espera** ou um elemento **Esperar por**, pode demorar um pouco até que o programa chegue ao próximo elemento.



Download

Se possuir uma ROBO Interface (interface não inteligente), pode utilizar também o **modo Download** em vez do modo Online. No modo Online são executados os programas do seu computador. O seu computador envia ordens de controle como “ligar motor” à interface. Para isso é necessário que a interface esteja conectada ao computador, enquanto o programa roda. No modo Download é executado o programa pela própria interface. O seu computador salva o programa na interface. Quando isso acontecer, a conexão entre o computador e a interface pode ser separada. Agora a interface pode executar o programa de controle independentemente do

computador. O modo Download é importante, por exemplo, no caso de programação de robôs móveis, nos quais um cabo de conexão entre o PC e robôs ser muito incomodo. Apesar disso, os programas de controle devem ser testados em primeiro lugar no modo Online, pois assim os possíveis erros são mais fáceis de se encontrar. O programa testado pode ser transferido por download para a ROBO Interface. Na ROBO Interface o cabo avariado pode ser substituído através do circuito radioelétrico **ROBO RF Data Link**, art. n.º 93295. Para isso, o modelo está móvel também no modo online.

O modo Online tem também vantagens relativamente ao modo Download. Em comparação com a interface, um computador tem muito mais memória principal e pode calcular de forma muito mais rápida. Esta é uma vantagem no caso de programas grandes. Além disso, podem ser ativadas paralelamente mais interfaces no modo Online, mesmo ROBO Interfaces e Intelligent Interfaces.

Os dois tipos de funcionamento em resumo

Tipo de funcionamento	Vantagem	Desvantagem
Online	<ul style="list-style-type: none"> • A execução do programa pode ser seguida na tela • A execução é muito rápida também em programas grandes • Podem ser ativadas paralelamente mais interfaces • Suporta a Intelligent Interface antiga • Os painéis de comando podem ser utilizados • O programa pode ser interrompido e retomado. 	<ul style="list-style-type: none"> • O computador e a interface têm de ficar conectados entre si
Download	<ul style="list-style-type: none"> • O computador e a interface podem ser separadas depois do Download 	<ul style="list-style-type: none"> • Não suporta a Intelligent Interface antiga • A execução do programa não pode ser seguida na tela • O programa só pode ativar até 3 módulos de ampliação



Download

Utilizar o modo Download

Se possuir a nova ROBO Interface, pode transferir o controle do portão da garagem com o botão **Download** para a barra de ferramentas na interface. Em primeiro lugar, é indicada à direita a janela de diálogo. A ROBO Interface dispõe de outros locais de memória do programa, uma área **RAM** (Random Access Memory) e duas áreas **Flash**. Um programa perde-se na RAM, se desligar a interface da alimentação de corrente ou se a bateria estiver vazia. Pelo contrário, um programa memorizado em Flash permanece por muitos anos na interface mesmo sem tensão. Claro que, apesar disso, pode substituir programas em Flash. O download na RAM torna-se claramente mais rápido e é, por isso, recomendado para fins de teste.



Nas duas áreas Flash pode colocar dois programas diferentes, por exemplo dois procedimentos diferentes para um robô móvel. Pode selecionar, iniciar e parar os dois programas com a tecla **Prog**. Se a opção **Iniciar programa para download** estiver ativa, o programa é iniciado imediatamente para download. Enquanto o programa roda, na ROBO Interface pisca o LED verde **Prog 1** (programa carregado em Flash 1) ou **Prog 2** (programa carregado em Flash 2) perto da tecla **Prog**. Se for carregado um programa na RAM, piscam ambos os LEDs. Para interromper o programa, pressione a tecla **Prog**. A seguir, o LED acende permanentemente. Para mudar do programa 1 para o programa 2 mantém a tecla Prog pressionada até o LED do programa 1 ou 2 desejado acender. Para iniciar o programa, pressione novamente a tecla.

No caso de robôs móveis, faz mais sentido a opção **Iniciar programas com o botão na Interface**. Se não possuir qualquer RF Data Link, tem de marcar sobretudo o cabo, antes de o seu programa colocar o robô em movimento. Neste caso, tem de selecionar, em primeiro lugar, o programa desejado com a tecla Prog na interface e iniciá-lo com uma nova pressão.

Se a última opção **Iniciar automaticamente ao ligar** estiver ativa, o programa 1 é iniciado automaticamente, logo que a interface seja alimentada com corrente. Com isso, pode, por exemplo, alimentar com corrente a sua interface com temporizador e iniciar o programa todos os dias à mesma hora. Nem a interface tem de permanecer conectada permanentemente, nem o programa tem de ser iniciado todas as vezes após a conexão através da tecla Prog.

Notas:

Quando um programa é carregado em Flash ou é executado a partir de Flash, os programas carregados em RAM são perdidos, porque os programas Flash também utilizam a memória RAM.

Uma descrição pormenorizada das funções da ROBO Interface pode ser consultada também nas instruções de serviço da interface.

3.8 Dicas e truques

Alterar linhas de conexão

Se deslocar elementos, o ROBO Pro esforça-se por adaptar razoavelmente as linhas de conexão. Se não gostar de uma linha adaptada, pode alterar facilmente as linhas de conexão, ao mesmo tempo que clica na linha com o botão esquerdo do mouse e desloca a linha com o botão do mouse pressionado. Dependendo do local onde se encontre o mouse na linha, é deslocado o ângulo ou o canto da linha. Isso é indicado por diferentes cursores do mouse:



Se o mouse se encontrar numa linha de conexão vertical, pode deslocar toda a linha vertical com o botão esquerdo do mouse pressionado.



Se o mouse se encontrar numa linha de conexão horizontal, pode deslocar toda a linha horizontal com o botão esquerdo do mouse pressionado.



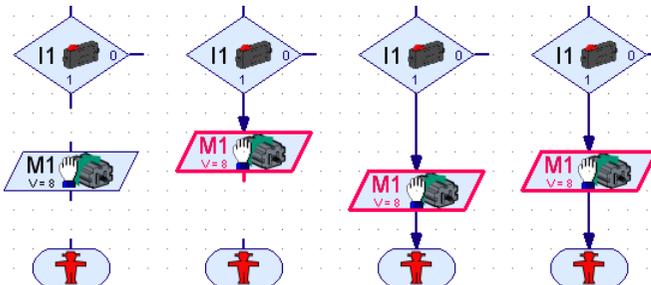
Se o mouse se encontrar numa linha de conexão diagonal, é introduzido um novo ponto na linha de conexão, quando pressiona o botão esquerdo do mouse. Tem de manter pressionado o botão esquerdo do mouse, e só soltá-lo quando o mouse se encontrar no local onde o novo ponto deverá estar.



Se o mouse se encontrar próximo de um ângulo ou ponto final de uma linha de conexão, pode deslocar o ponto com o botão esquerdo do mouse pressionado. Um ponto final da linha conectada pode ser deslocado apenas para uma outra conexão adequada de um elemento do programa. Neste caso, o ponto final da linha de conexão é ligada com este ponto de conexão. Caso contrário, o ponto não é deslocado.

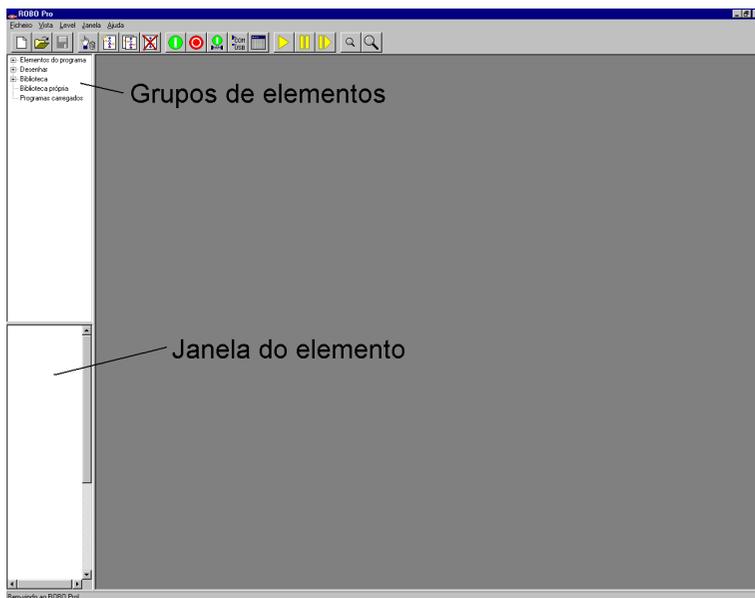
Outras linhas de conexão

As linhas de conexão podem ser criadas também pelo deslocamento dos elementos do programa. Quando se desloca um elemento do programa de forma que a sua entrada se encontra um ou dois pontos de encaixe abaixo da saída de um outro, é criada entre ambos os elementos uma linha de conexão. Isso é válido também para uma saída que é deslocada por uma entrada. A seguir, pode deslocar o elemento do programa para a sua posição final ou designar outras conexões para as entradas e saídas restantes:



4 Level 2: Trabalhar com os subprogramas

Depois de ter criado e testado com sucesso o seu primeiro programa de comando, está pronto para o ROBO Pro Level 2. Seleccione agora no menu **Level** o registro **Level 2: Subprograma** desligado. Com certeza que reparou imediatamente na diferença: A janela do elemento desapareceu e, em vez dela, tem duas janelas sobrepostas no lado esquerdo:



Mas não precisa de ter medo! A janela do elemento ainda está lá, mas está livre. No Level 2 existem outros elementos do programa, que poderia se tornar complicado englobar tudo numa janela. Por isso, os elementos estão agrupados a partir do Level 2. Os elementos estão organizados em grupos de forma idêntica às pastas de arquivos no disco rígido do seu computador. Se no lado esquerdo da janela superior seleccionar um grupo, surgem todos os elementos deste grupo na janela inferior. Os elementos do Level 1 encontram-se no grupo **Elementos do programa / Elementos básicos**. Como a janela do elemento só está completa até metade, tem de utilizar a barra de deslocamento à direita na janela do elemento para indicar os elementos inferiores.

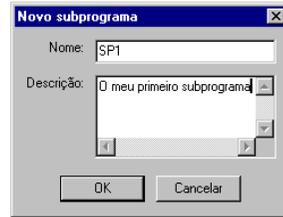
Mas ainda sobre o mesmo tema: os subprogramas! Embora os nossos organigramas em questão ainda não sejam tão extensos para se perder a clareza, pode imaginar que isso aconteça facilmente no caso de grandes projetos com organigramas extensos. De repente, a folha de trabalho enche-se de componentes, por todo o lado há linhas de conexão e na tela tem de se deslocar permanentemente a barra de deslocamento para cima e para baixo. "Onde estava esta ou aquela saída?" Brevemente – está iminente um pequeno caos! O que fazer? Não é possível restabelecer alguma ordem? Sim – denomina-se de **subprograma!**

4.1 O seu primeiro subprograma

Um subprograma é muito semelhante aos programas que conheceu até agora. De forma a poder analisar mais pormenorizadamente, tem de criar, em seguida, um novo programa e nele um novo subprograma vazio. Clique no programa **Novo** e, a seguir, o botão **UP Novo** na barra de ferramentas. Surge uma janela em que pode inserir o nome e uma descrição do subprograma.

O nome não deve ser demasiado longo (aprox. 8-10 letras), caso contrário o símbolo do subprograma torna-se demasiado grande. Mais tarde, pode alterar todos os registos que realizou até aqui.

Logo que feche a janela **Novo subprograma** com **OK**, surge na barra de subprogramas o novo subprograma:



Pode alternar, em qualquer momento, entre o programa principal e o subprograma, ao mesmo tempo que clica no nome do programa na barra de subprogramas. Como ambos os programas ainda estão vazios, ainda não é possível visualizar qualquer diferença.

Desejamos agora organizar o controle do portão da garagem do capítulo anterior (ver capítulo 3.6 *Outros elementos do programa* na página 462) em subprogramas. O programa é composto por quatro unidades funcionais:

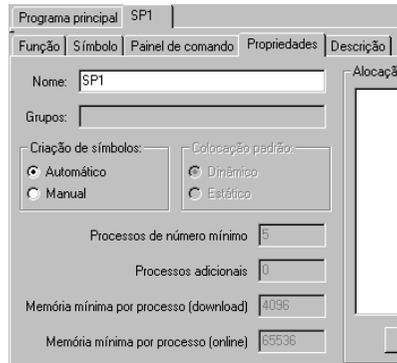
- Aguardar até pressionar botão I1
- Abrir portão
- Aguardar 10 segundos
- Fechar portão

Desejamos distribuir a abertura e o fecho apenas em dois subprogramas. Pode chamar ambos os subprogramas, a partir do programa principal, com apenas um símbolo. A espera pelo botão I1 e o tempo de espera de 10 segundos permanecem no programa principal, pois ambos são compostos por apenas um elemento. Criou um novo programa com um subprograma denominado **Subprograma 1**. **Abrir** e **Fechar** seriam os melhores nomes para os dois subprogramas. Pode mudar o nome do subprograma já criado, selecionando, em primeiro lugar, o subprograma 1 pela barra de subprogramas, desde que já não esteja selecionado.

Mude, em seguida, com a barra de funções para a janela Propriedades do subprograma, clicando em **Propriedades**. Aqui pode alterar o nome de **UP 1** para **Abrir**. A maioria dos outros campos podem ser alterados apenas no nível avançado ou até mesmo no nível profissional. O ponto **Criação de símbolos** é explicado mais à frente.

Se clicar em **Descrição** na barra de funções, pode alterar a descrição anteriormente introduzida, apesar de o "O meu primeiro programa" já estar corretamente definido.

Na barra de funções clique em **Função**, para que possa programar a função do subprograma.



Agora visualiza novamente a janela de programas em que, no capítulo anterior, introduziu os elementos do programa para o seu primeiro programa ROBO Pro. Preste atenção para o fato de na barra de subprogramas o subprograma **Abrir** estar selecionado:

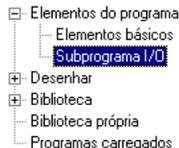
Programa principal | **Abrir**

Está preparado para programar o seu primeiro subprograma? Comecemos então! Mas por onde se começa realmente um subprograma? Boa pergunta! Começou sempre um programa com o elemento inicial. Um subprograma começa com um elemento idêntico, a entrada do subprograma. O elemento se chama assim porque através deste elemento a execução do programa principal entra no subprograma. Aqui não pode utilizar nenhum elemento inicial porque não deve ser iniciado nenhum novo processo.



	Elemento inicial	Inicia um novo processo independente
	Entrada do subprograma	Aqui, a execução do programa é transferida do programa principal para o subprograma

Pode encontrar a entrada do subprograma na janela do grupo de elementos em **Subprograma I/O**. Coloque agora a entrada do subprograma superior na janela do programa para o subprograma **Abrir**. Ao elemento de entrada do subprograma pode atribuir um outro nome como **Entrada**, mas isso só é necessário quando escreve um subprograma com mais entradas.



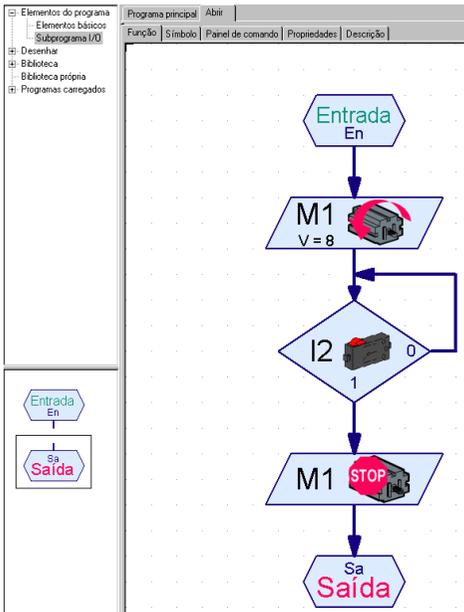
O outro decurso do programa no subprograma é idêntico à parte **Abrir** do programa principal atual: Ligue o motor M1 com sentido de rotação para a esquerda, aguarde até o botão estar fechado na entrada I2 e volte a desligar o motor.

Para terminar o subprograma, utilize uma saída do subprograma. A diferença entre a saída do subprograma e o elemento de parada é a mesma que entre a entrada do subprograma e o início do processo.



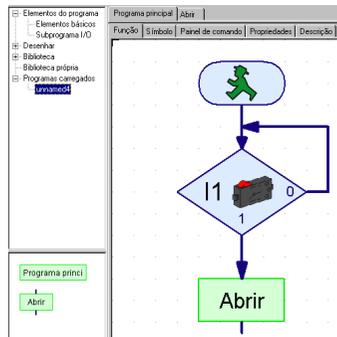
	Elemento de parada	Termina a execução do programa de um processo independente
	Saída do subprograma	Aqui, a execução do programa é transferida do programa principal para o subprograma

O seu subprograma pronto deve ter o seguinte aspecto:



Preste atenção para o fato de ter inserido o subprograma em **Abrir**, e não em **Programa principal**. Na barra de subprogramas volte a mudar de **Abrir** para **Programa principal**. Visualize agora a janela de programas vazia do programa principal. Insira no programa principal, como é costume, um elemento inicial (sem entrada de subprograma!). Também efetue a consulta do botão em **I1**, que deve abrir o portão da garagem, no programa principal.

Pode inserir agora o seu novo subprograma como um elemento habitual de programa no seu programa principal (ou um outro subprograma). Encontra isso na janela de grupos de elementos em **Programas carregados** e nos nomes dos arquivos do seu programa. Se ainda não tiver salvo o arquivo, o nome é **indefinido1**. Se ainda não tiver carregado nenhum arquivo de programa, pode selecionar também na janela de seleção subprogramas que pertencem a outros arquivos. Desta forma, pode utilizar subprogramas de um outro arquivo.



No grupo de elementos **Programas carregados / indefinido1** encontra dois símbolos verdes de subprogramas. O primeiro com o nome **Programa principal** é o símbolo do programa principal. Esse é raramente utilizado como subprograma, mas é possível, por exemplo, quando controla maquinaria e desenvolveu os comandos para as máquinas individuais como programas principais. O segundo símbolo com o nome **Abrir** é o símbolo do seu novo subprograma. **Abrir** é o nome que introduziu em Propriedades. Introduza agora o símbolo de subprograma no seu programa principal de forma a se habitue aos elementos habituais do programa. É muito simples!

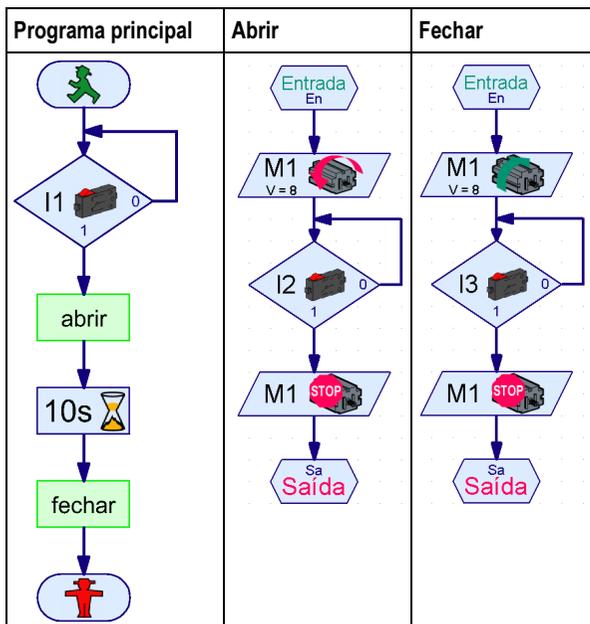


UP Novo

Podemos terminar o seu programa principal com um elemento de parada, e experimentá-lo. O portão abre por meio da pressão do botão I1, mas para fechar não programamos. Para isso, tem de instalar um outro subprograma. Pressione o botão **UP Novo** na barra de ferramentas e introduza na janela **Novo subprograma** o nome **Fechar**. Não precisa introduzir uma descrição, e esta não efetua danos, para que mais tarde saiba para que finalidade o subprograma foi pensado.

Introduza agora na janela de programas para o subprograma **Fechar** o programa para fechar o portão da garagem. Começa de novo com uma entrada do subprograma. O motor **M1** deve funcionar, em primeiro lugar, para a direita. Logo que o botão fim de curso seja fechado em **I3**, o motor **M1** deve parar. O subprograma é concluído novamente com uma saída do subprograma.

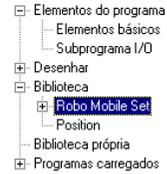
Passa de novo da barra de subprogramas para o programa principal. Quando terminou há pouco o seu programa principal com um elemento de parada para o experimentar, tem de voltar a apagar o elemento de parada. Depois do portão da garagem estar aberto, ele deverá ficar neste estado durante 10 segundos antes de voltar a se fechar. Depois de um tempo de espera de 10 segundos introduza a partir do grupo de elementos **Programas carregados / indefinido1** o símbolo **Fechar** para o subprograma. O programa principal e ambos os subprogramas devem ter o seguinte aspecto:



O programa inicia no elemento de parada em **Programa principal**. O programa aguarda, então, até o botão **I1** ser pressionado. Para isso, pode utilizar também o elemento **Aguardar pela entrada** (ver capítulo 7.1.8 *Aguardar pela entrada* na página 500). Após o botão **I1** ter sido pressionado, o programa principal encontra a consulta do subprograma **Abrir**. A execução do programa muda para a entrada do subprograma **Abrir**. O subprograma **Abrir** abre o portão da garagem e chega a uma saída do subprograma. Neste ponto, o programa ramifica-se novamente para o programa principal. Após o final do subprograma **Abrir**, aguarda-se 10 segundos no programa principal. A seguir, a execução do programa muda para o subprograma **Fechar**, que volta a fechar o portão da garagem. Depois regressa ao subprograma **Fechar**, o programa principal se encontra com um elemento de parada, através do qual o programa é concluído.

4.2 A biblioteca de subprogramas

Pode copiar simplesmente subprogramas de um arquivo para outro, ao mesmo tempo que carrega ambos os arquivos e introduz um subprograma de um arquivo através do grupo de elementos **Programas carregados** num outro arquivo. Para subprogramas frequentemente utilizados ainda é mais simples, é com a **Biblioteca**. ROBO Pro inclui uma biblioteca de subprogramas prontos que pode utilizar facilmente. Além disso, pode instalar uma biblioteca própria para guardar os subprogramas que são utilizados mais vezes.

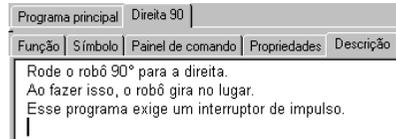


4.2.1 Utilização da biblioteca

A **biblioteca** subdivide-se, em primeiro lugar, em dois grupos principais. No grupo **Módulos** encontra subprogramas que pode utilizar para modelos de determinados módulos. No grupo **Geral** encontra subprogramas que pode utilizar para todos os modelos possíveis. A maioria destes subprogramas do grupo **Geral** exige, porém, técnicas do nível 3 que só são explicadas no próximo capítulo.

Para cada módulo Computing, como por exemplo o ROBO Mobile Set, existe no grupo **Módulos** um subgrupo próprio. Este é, por vezes, subdividido em modelos que pode encontrar nas instruções de montagem do módulo. Se seleccionar o módulo ou um dos modelos, na janela do elemento são indicados os subprogramas prontos para este modelo.

Se indicar com o mouse um dos símbolos do subprograma, é indicada uma breve descrição. Se introduzir um subprograma no seu programa, pode indicar uma descrição pormenorizada, seleccionado o subprograma na barra de subprogramas e, a seguir, clicar em **Descrição** na barra de funções:



Atenção: Quando introduz um subprograma da biblioteca, são introduzidos em parte outros subprogramas, que são utilizados por este subprograma. Pode voltar a deletar todos os subprogramas, seleccionando no menu **Editar** a função **Undo**.

4.2.2 Utilização da biblioteca pessoal

Depois de se ter ocupado um pouco com o ROBO Pro, terá com certeza um subprograma próprio que utiliza mais vezes. Para que não precise de procurar e carregar o respectivo arquivo, pode instalar uma biblioteca de subprogramas pessoal que funcione como a biblioteca predefinida. A biblioteca pessoal é composta por um ou mais arquivos ROBO Pro que são salvados numa pasta. Para cada arquivo numa pasta é indicado um grupo próprio na selecção de grupos.

Pode indicar a pasta em que pretende salvar a sua biblioteca pessoal no menu **Arquivo** em **Diretório biblioteca pessoal**. O diretório padrão para a biblioteca pessoal é o C:\Programas\ROBOProbiblioteca pessoal. Se tiver um diretório de biblioteca pessoal no seu computador, recomenda-se a criação e utilização de uma pasta própria.

Dica: No início pode indicar a pasta em que salva os seus programas ROBO Pro em **Diretório biblioteca pessoal**. Assim, tem acesso rápido a todos os arquivos na sua pasta de trabalho em todos os subprogramas.

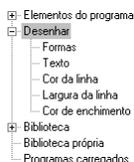
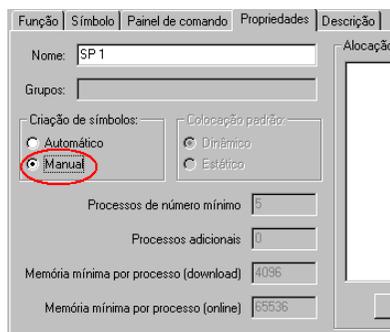
Organizar a sua biblioteca pessoal

No ROBO Pro não existe qualquer função especial para alterar uma biblioteca. Apesar disso, isso efetua-se de uma forma muito simples. Se desejar introduzir ou remover subprogramas de um grupo de bibliotecas, tem de carregar, em primeiro lugar, o respectivo arquivo. Encontra estes arquivos no diretório que configurou em **Diretório biblioteca pessoal**. Agora pode, por exemplo, carregar um segundo arquivo e dele retirar um subprograma do grupo **Programas carregados** no programa principal da biblioteca. No caso de uma biblioteca, o programa principal não é um programa mas sim uma coleção de todos os subprogramas da biblioteca. No caso de bibliotecas, o programa principal não é indicado na janela de elementos. Claro que também pode apagar ou alterar subprogramas de uma biblioteca.

Quando altera e salva um arquivo de biblioteca, tem de selecionar no menu **Arquivo** o item do menu **Atualizar biblioteca pessoal**. Para isso, a lista de arquivos é atualizada na janela de grupos.

4.3 Editar símbolos de subprogramas

Como viu no capítulo anterior, o ROBO Pro cria automaticamente símbolos de subprogramas verdes para os seus subprogramas. Mas também pode desenhar alguns símbolos que melhor exemplifiquem o que os seus subprogramas realizam. Para isso, na janela Propriedades do subprograma tem de mudar de criação automática de símbolos para criação manual de símbolos. A seguir, pode editar na barra de funções de **Propriedades** para **símbolo** e aí editar o símbolo do subprograma. Encontra as funções de desenho na janela de grupos de elementos em **Desenhar**.



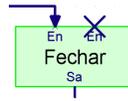
Em **Símbolos / Formas**

encontra todos os elementos gráficos básicos mais comuns como retângulo, círculo, elipse, polígonos entre outros. Em **Símbolos / Texto** encontra objetos de texto em diferentes tamanhos de letra. Em outros grupos em **Desenhar**, encontra funções para alterar a cor e as propriedades semelhantes dos elementos selecionados. A utilização correta das funções dos caracteres vem descrita no capítulo 9 *Funções de desenho* na página 526. Preste também

atenção às funções no menu principal em **Desenhar**.

Pode também deslocar as ligações do subprograma, mas não pode deletar ou introduzir novas ligações. No símbolo do programa existe sempre uma ligação para cada entrada ou saída do subprograma da função do subprograma. Os elementos de conexão são também criados automaticamente se mudar para a criação de símbolos manual.

Logo que saia da janela de edição de símbolos, são adaptadas respectivamente todas as consultas do subprograma no programa principal ou em outros subprogramas. Preste atenção para o fato de, ao deslocar ligações de um subprograma, poder ocorrer uma pequena confusão na consulta do subprograma quando as ligações já estavam ligadas. Os pontos finais das linhas de conexão deixam de terminar na ligação correta, que é indicada através de uma cruz no ponto final da linha e na ligação (ver figura). Geralmente, basta fazer clique uma vez com o botão esquerdo do mouse em algum lugar linha de conexão. A linha é automaticamente transferida. Em subprogramas com muitas ligações pode acontecer que ainda tenha que editar a linha.

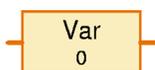


5 Level 3: Variáveis, painéis de comando & Co

Pense em alterar o ROBO Pro no menu **Level** em **Level 3** (ou superior)!

Imagine que descobre num corredor lateral até aqui inexplorado de um museu uma máquina fascinante que pretende reproduzir na fischertechnik. Na exploração da máquina se esquece do tempo e não repara que todos os outros visitantes do museu já saíram. Só quando o museu fechou, é que estudou ao pormenor o funcionamento da máquina para a conseguir reproduzir. Mas, infelizmente, tem de passar uma noite desconfortável e solitária no museu antes de poder ir embora. Para que tal não volte a acontecer, pede ao diretor do museu para programar um dispositivo que conta todos os visitantes que entram e saem, liga uma lâmpada vermelha de aviso caso ainda estejam visitantes no museu. Mas como se faz isso? Como pode contar algo com o ROBO Pro? A resposta está em: **variáveis**.

5.1 Variáveis e comandos

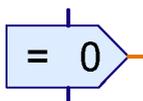


Uma variável é um elemento que consegue memorizar um número. Na janela Propriedades das variáveis insere um **Nome** que deve indicar que número deve estar salvo nas variáveis.

Em **Valor inicial** pode indicar que número deve ser salvo nas variáveis no início do programa. A configuração **Tipo de variável** é explicada no capítulo 7.3.2 *Variável local* na página 504).



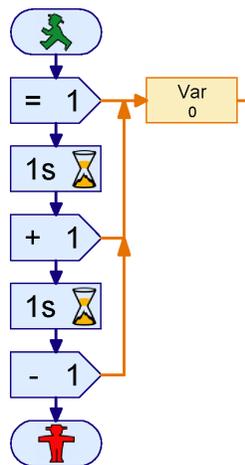
Pode alterar o número salvo, enviando os comandos à variável. Uma variável compreende 3 diferentes comandos: =, + e -. O comando = substitui o número salvo por um novo número. Os comandos + e - acrescentam algo ao número salvo ou deduzem algo daí.



Envia os comandos com um **elemento de comando** à variável. O elemento de comando tem, como a maioria dos outros elementos de comando, em cima uma entrada de programa a azul e em baixo uma saída de programa a azul.

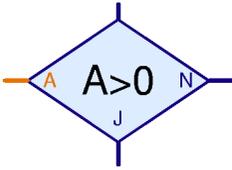
À direita tem algo novo, uma ligação **laranja**. É uma saída de comando. Sempre que for executado um elemento de comando, é enviado um comando a todos os elementos conectados através desta saída. A variável tem do lado esquerdo uma entrada de comando correspondente. Se conectar a saída de comando com a entrada de comando, ROBO Pro desenha, em vez das ligações azuis comuns, uma linha laranja. Através das linhas laranjas, os elementos do programa podem enviar comandos ou mensagens e trocar informações.

O programa à direita envia às variáveis **Var** um comando = 1. Um comando é composto geralmente por um verdadeiro comando como = e um valor como 1. O comando =1 coloca a variável em 1. Após um segundo, o programa envia às variáveis um comando +1.



A variável acrescenta ao seu valor atual 1 e daí resulta o valor 2. Depois de um outro segundo, o programa envia um comando -1. Daí a variável volta a ter o valor 1.

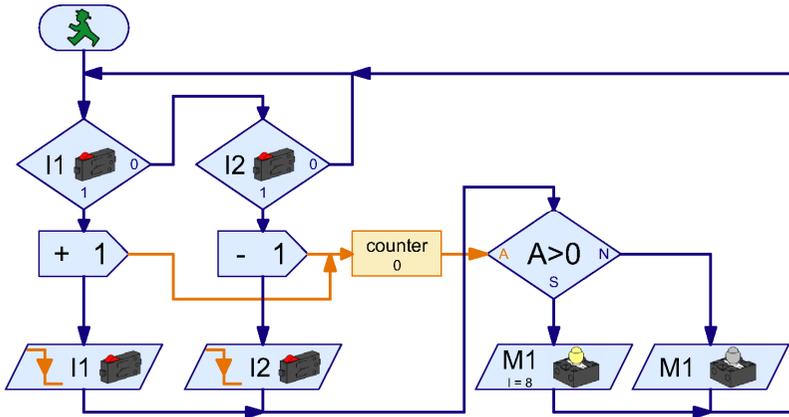
Tente desenhar este programa no ROBO Pro. Encontra o elemento de comando no grupo **Comandos**, a variável no grupo **Variáveis, Temporizador, ...**. Se executar o programa no modo Online, verá como o valor das variáveis se altera.



Poderá dizer: "É tudo muito bonito, mas...": Consegui ver o valor das variáveis, mas e agora o que faço com isso? É muito simples: A variável tem à direita uma ligação cor de laranja, com a qual envia mensagens com o seu valor atual a todos os elementos ligados. Existem alguns elementos no ROBO Pro, que tem uma entrada laranja à esquerda, que pode ligar à saída da variável. Pode encontrar, por exemplo, no grupo **Ramificação, Aguardar, ...** um elemento de ramificação Sim/ Não, o qual não consulta

directamente uma entrada, mas pode consultar um valor qualquer, entre outros, o valor de uma variável.

Assim, o dispositivo de contagem de visitantes para o museu pode ser programado da seguinte forma:



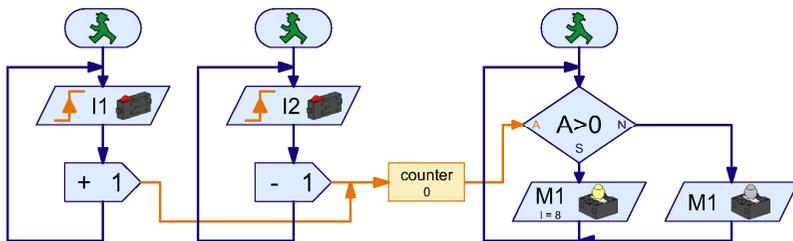
O torniquete na entrada aciona o botão em I1, o torniquete na saída o botão em I2. Logo que I1 esteja pressionado, o programa envia à variável **Contador** um comando + 1. A seguir, o programa aguarda até ao botão ser solto em I1. Com o botão para a saída em I2 comporta-se assim apenas de forma que à variável **Contador** seja enviado um comando - 1. Cada vez que o contador é alterado, o estado do contador é controlado. Quando a variável **Contador** possui um valor > 0 é ligada a lâmpada de aviso vermelha em M1, caso contrário é desligada.



O programa superior desenha e experimenta. Desde que pressione o botão em I1 e o volte a soltar, acende a lâmpada de aviso em M1. Quando aciona o botão em I2, ela se desliga. Se acionar mais vezes I1, tem de acionar I2 tantas vezes quantas as necessárias para que a lâmpada de aviso se desligue de novo. Tente novamente, o que se passa quando entram só 5 visitantes, depois 2, a seguir 3. Quantas vezes tem de acionar o botão em I2 para que a lâmpada de aviso se desligue?

5.2 Variáveis e vários processos

Durante o teste de contagem de visitantes, talvez tenha notado que dá problemas se pressionar simultaneamente os botões em I1 e I2. Enquanto um dos botões estiver a ser pressionado, o programa não consegue mais reagir ao outro botão. Mas a partir do momento que os visitantes à entrada e à saída possam passar simultaneamente pelo respectivo torniquete, podem surgir erros na contagem. Pode resolver este problema, utilizando vários processos paralelos. Até então, todos os programas tinham apenas um elemento inicial. Mas pode utilizar vários elementos iniciais. Todos os processos com um elemento inicial próprio serão processados repetidamente. O técnico fala em **processos secundários**. Com esta técnica pode alterar o programa de contagem de visitantes da seguinte forma:



São agora utilizados processos independentes para I1 e I2. Quando o botão em I1 estiver pressionado, o processo para I2 fica independente e pode continuar a vigiar o botão em I2. Para consultar os valores de contagem espera ligar e desligar a lâmpada de aviso é também utilizado um processo próprio.

Como pode ver, não há qualquer problema em aceder a uma variável a partir de vários processos. Pode, a partir de vários processos, enviar ordens a uma variável e pode utilizar o valor de uma variável em vários processos. Daí que as variáveis se ajustam perfeitamente a trocar informações entre os processos.



O diretor do museu está tão entusiasmado pelo seu contador genial de visitantes, que pede logo a você a solução para outro problema: o museu organizou uma nova exposição. Mas como todos os visitantes querem ver a nova exposição, há uma tal confusão que já ninguém quer ver nada. O diretor quer, por isso, limitar o número de visitantes na exposição em 10. À entrada e à saída da exposição, o diretor colocou um torniquete. O torniquete à entrada pode ser eletronicamente bloqueado. Agora só precisa de um programador competente, ou seja, de você!

Tente desenvolver o programa descrito com ROBO Pro. Em geral funciona como o contador de visitantes. Pode simular o bloqueio eletrônico da entrada através de uma lâmpada vermelha em M1, que deve ser ligado quando se encontram 10 visitantes na exposição.

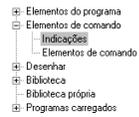
5.3 Painéis de comando

Depois de você ter solucionado o problema com a exposição, o diretor do museu já tem uma nova tarefa: Ele também gostaria de saber quantas pessoas visitaram o museu em um dia. Um programa que sabe somar já não é problema para você, mas como é que você pode indicar um valor? Naturalmente você poderia executar o programa no modo Online e mostrar para o diretor do museu em que variáveis ele pode procurar o valor. Mas para um amador de computadores como o diretor do museu, isso é bastante complicado. Tem de ser algo mais fácil!

Para esses casos existe o painel de comando no ROBO Pro. Um painel de comando é uma página própria na qual você pode traçar indicações e botões de comando. Carregue seu programa contador de visitas e passe para painel de comando na barra de funções.

Funktion | Symbol | Bedienfeld | Eigenschaften | Beschreibung

O painel de comando é primeiramente uma superfície cinzenta vazia. Nessa superfície você coloca indicações e elementos de comando que encontra na janela dos grupos de elementos em **Elementos de comando**. Nos elementos de comando você encontra botões, controles deslizantes e semelhantes. Nas indicações você encontra indicações de texto, lâmpadas de sinalização e indicações com indicador giratório.



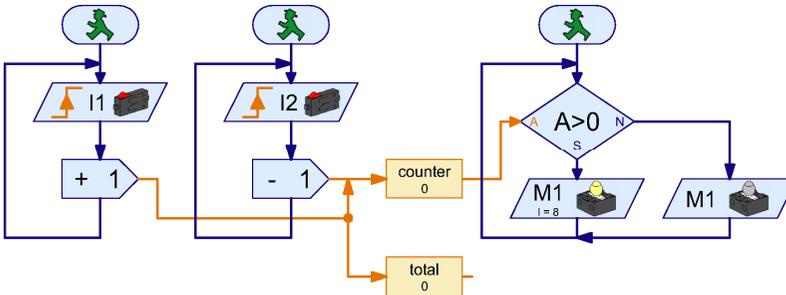
Atenção: Um painel de comando é parte de um subprograma. Se você tiver subprogramas, se certifique que instala o painel de comando em **Programa principal** e não em um subprograma! Mais tarde, como profissional, você poderá instalar mais painéis de comando.

Se você tiver traçado um painel de comando e mais tarde o painel de comando desaparecer, provavelmente você selecionou um subprograma na barra de subprogramas. Passe outra vez para **Programa principal**, e o seu painel de comando certamente estará lá.

Var: 0

Para o dispositivo de contagem de visitantes, você pega em uma **indicação de texto** (a cor é indiferente) da janela do elemento **Elementos de comando/Indicações**, e a coloca no painel de comando. Nessa indicação deve ser apresentado agora a quantidade de visitantes do museu.

A seguir você tem de adicionar uma segunda variável ao seu programa, que conte o número de visitantes na entrada sem que volte a subtrai-los na saída. Para isso, na barra de funções, você muda novamente para **Função** e introduz a variável **Total**, como se segue:



Como você vê, um elemento de comando também pode ser utilizado para enviar simultaneamente um comando a duas variáveis. A variável **Total** não recebe os comandos **-1**, porque comandos ao longo das linhas laranja só são transferidos na direção da seta. Pelo contrário, os comandos **+1** são transmitidos a ambas as variáveis. No entanto, isso deve ser só um exemplo aqui. Normalmente é mais fácil e mais claro utilizar um segundo elemento de comando.



Dica: Quando as linhas laranja se ramificam é frequentemente mais prático traçar as linhas do objetivo desde o princípio. Se no caso do exemplo de cima, quiser desenhar a linha para a variável **Total**, clique primeiro na entrada da variável **Total** e desloque a linha para trás até ao ponto da ramificação. Se, em contrapartida, quiser iniciar uma linha cor de laranja numa

da mesma cor existente, terá de fazer duplo clique com o botão esquerdo do mouse no ponto onde pretende iniciar a nova linha.



Agora você tem uma indicação de texto no painel de comando e uma variável, que vocês pretendem apresentar na indicação. Como se ligam os dois? Visto que a indicação de texto e a variável estão em lados diferentes, é difícil conseguir ligar as duas com uma linha. Por isso, existe um elemento especial, que encaminha o valor, que tem de ser apresentado no painel de comando, para a respectiva indicação. Você pode encontrar o elemento **Saída do painel de comando** representado acima no final do grupo **Entradas, Saídas**. Introduza uma saída do painel de comando no seu programa ao lado das variantes **Total** e ligue a conexão direita das variáveis à conexão da **Saída do painel de comando**.

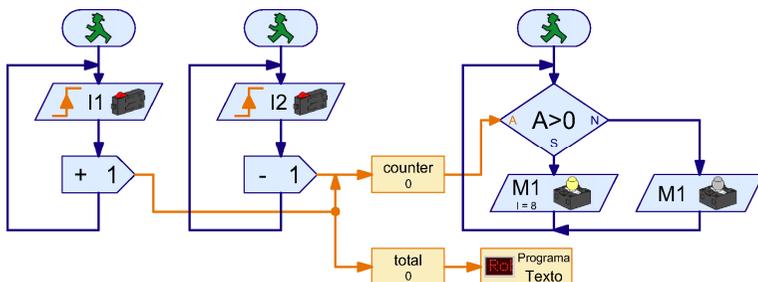
Visto que normalmente você terá mais do que uma indicação num campo de comando, terá de transmitir à saída do painel de comando a qual indicação tem de enviar os valores da variante. Isto se efetua facilmente através da janela Propriedades do elemento. Quando clica com o botão direito do mouse na saída do campo de comando, você pode ver uma lista, na qual estão registradas todas as indicações, que já foram inseridas no painel de comando. Visto que cada subprograma pode ter o seu próprio painel de comando, os diferentes painéis de comando estão classificados de acordo com os subprogramas. No nosso exemplo, não existe qualquer subprograma, apenas o programa principal. Por baixo dele, existe uma indicação com o nome **Texto**. Selecione esta indicação e clique em OK.



Logo que você tiver conectado a saída do painel de comando a uma indicação, o símbolo e a inscrição se alteram respectivamente. A saída do painel de comando utilizada por nós estabelece uma ligação com a indicação de texto com o nome **Texto** no subprograma **PRINCIPAL**.



Depois de ter inserido a saída do painel de comando e de a ter conectado com a indicação de texto, o seu programa terá este aspecto:



Experimente-o de imediato. Logo que tiver iniciado o programa em modo Online, a indicação mostra, no painel de comando, o número de visitantes que passaram o torniquete da entrada.

Nota: Caso pretenda mudar mais do que uma indicação num painel de comando, é importante que você atribua um outro nome a cada uma delas, para que as possa distinguir ao conectar com o programa. Para isso, clique com o botão direito do mouse na indicação do painel de comando. Lá pode introduzir um nome em **ID/Nome**. Quando você conectar uma saída do painel de comando à indicação, este nome surge na janela Seleção da saída do painel de comando. Visto que, de momento, temos apenas uma indicação, o nome não é importante e nós mantemos o nome **Texto**.

O programa ainda não está totalmente perfeito. O que ainda falta é um interruptor para zerar o contador. Para isso não queremos utilizar uma tecla normal, mas sim um botão, que possamos pressionar no painel de comando.

Botão

Podemos encontrar o botão de comando na janela do elemento no grupo **Elementos de comando/Elementos de controle**. Na barra de funções mude para **Painel de comando** e insira um botão no seu painel de comando ao lado da indicação de texto. Naturalmente, a inscrição **Botão** não é completamente adequada, mas é fácil de alterar através da janela Propriedades do botão. Clique no botão com o botão direito do mouse, introduza, por exemplo, 0000 no **texto de inscrição** e confirme com **OK**.

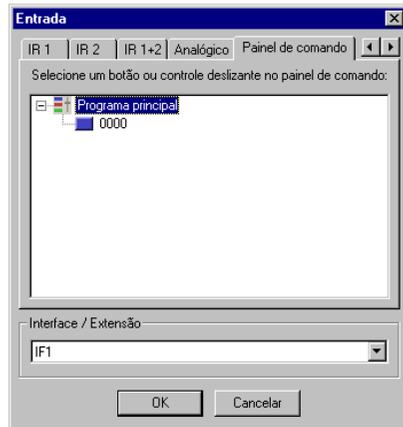


Exatamente como na indicação de texto, nós precisamos de um elemento do programa, que conecte o botão ao processo do programa. Por isso, primeiro mude novamente para **Função** na barra de funções. Na janela do elemento pode encontrar o elemento **Entrada do painel de comando**

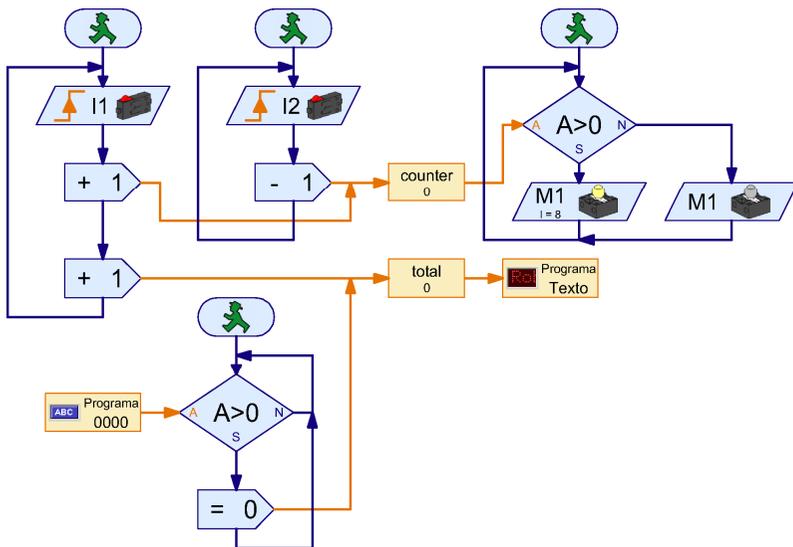
representado no grupo **Entradas, Saídas**. Coloque-o no processo do programa sob o processo efetuado até agora.

Agora você tem ainda que conectar a entrada do painel de comando com o botão do painel de comando. Para isso, clique com o botão direito do mouse no elemento entrada do painel de comando. Os elementos de controle estão classificados como nas indicações de acordo com os subprogramas, visto que cada um pode ter um painel de comando próprio. Selecione o botão **0000** e confirme com **OK**.

Provavelmente reparou, que pode ajustar este elemento relativamente a todos os tipos possíveis de entradas através da tecla Tab da janela Propriedades. Isto será explicado no capítulo a seguir ao próximo *Entradas de dados para subprogramas*.



O valor, que é fornecido pela entrada do painel de comando, pode ser consultado com um elemento de ramificação. Você já utilizou este elemento para consultar a variável. O programa concluído com posição zero terá este aspecto:



Enquanto o botão **0000** estiver pressionado, é enviado ao contador total um comando **= 0**, que repõe o contador a 0.

5.4 Temporizador

Depois dos seus bons resultados, o diretor do museu já não sabe o que fazer sem você e o nomeia conselheiro de computação do museu. Naturalmente, um cargo como este traz muita honra e fama, mas também traz muito trabalho, como por exemplo: No museu existem muitos modelos que se movem quando um botão é pressionado. Mas alguns visitantes pressionam os botões durante demasiado tempo, causando o superaquecimento dos modelos e, conseqüentemente, se torna necessário efetuar reparações com muita frequência. O diretor gostaria que os modelos funcionassem durante o tempo em que o botão está sendo pressionado, mas num período máximo de 30 segundos de cada vez. Quando o modelo funcionar uma vez, deve se suceder uma pausa de 15 segundos, até que este possa ser novamente conectado.

Hmm, isso não é difícil, pensa você: Alguns tempos de espera, umas ramificações do programa e está pronto. Tente à vontade! Mas, passado pouco tempo, você irá descobrir, que isso não é assim tão fácil, isto acontece por dois motivos:

- Durante o período de 30 segundos, o programa tem de consultar o botão para determinar se este foi solto antes dos 30 segundos. Bom, já desistiu. Isto pode ser resolvido através de dois processos, que são executados em simultâneo, ver capítulo 5.2 *Variáveis e vários processos* na página 478.
- Quando um visitante solta o botão após 5 segundos e, a seguir, o volta a pressionar depois de 15 segundos, o tempo de espera de 30 segundos tem de ser novamente reiniciado. Mas do tempo de espera decorreram apenas $5 + 15 = 20$ segundos e, por isso, ainda se encontra ativo. Mesmo com processos decorridos paralelamente, não se pode reiniciar um tempo de espera. Eventualmente, isto funciona com dois tempos de espera em três processos, que você inicia alternadamente, mas isso irá lhe causar dores de cabeça de tanto pensar.

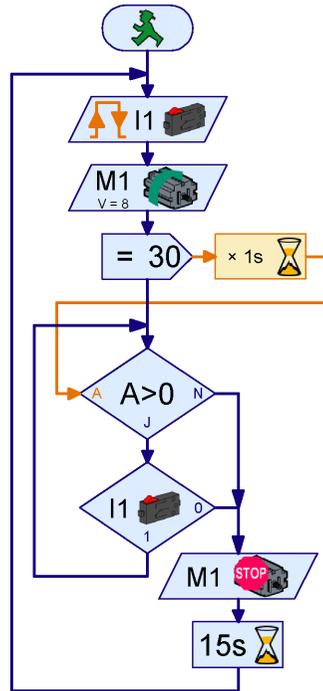


Mas não existe nada mais fácil? Sim, existem as **Variáveis do Temporizador**, ou apenas **Temporizador**. Um temporizador

funciona primeiro como uma variante normal. O temporizador memoriza um número e você pode alterar o número com os comandos =, + e -. A função especial num temporizador é que este repõe automaticamente o número a 0 com regularidade. Você pode ajustar o período de tempo para um intervalo de contagem em intervalos entre um milésimo de segundos e um minuto. Com temporizadores se pode executar tarefas de controle de tempo com mais estilo do que com tempos de espera. Você já compreendeu como pode executar a tarefa com um temporizador?

Correto: Quando o visitante pressionar o botão em I1, você inicia o modelo e programa o temporizador com um comando = 30 x 1 segundo = 30 segundos. Depois você consulta em um ciclo se o tempo de 30 segundos já decorreu ou se o botão em I1 foi solto. Se um dos dois critérios de interrupção for cumprido, você pára o modelo e espera 15 segundos. Depois é tudo inicializado novamente.

É verdade que aos poucos os programas se tornam um pouco mais complexos. Mas tente efetuar uma vez mais esta tarefa: desenvolva um programa com a mesma função com tempos de espera em vez de temporizadores! **Atenção: Essa é uma tarefa muito difícil e pensada só para os que gostam de enigmas complicados e demorados! Todos os outros podem pular para o próximo capítulo.** Existem duas aproximações ao problema para essa tarefa: Você pode utilizar dois tempos de espera, que você inicia alternadamente em processos próprios. Visto que existe uma pausa de 15 segundos, um dos dois tempos de espera acabou no máximo após o segundo ciclo, podendo então ser reiniciado. Existe uma outra alternativa, que é imitar um temporizador com uma variável normal e um elemento **tempo de espera** com um curto tempo de espera de por exemplo um segundo.



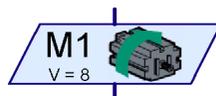
5.5 Entradas de comando para subprogramas

Como sempre, o seu programa funciona perfeitamente e a fischertechnik fica contente porque todos os modelos no museu estão equipados com a ROBO Interface. Infelizmente, como em todas as outras organizações públicas, também o museu não tem dinheiro. Por isso, o diretor quer ter o menor número possível de interfaces. Uma ROBO interface tem sempre quatro saídas de motor e também entradas suficientes para comandar quatro modelos. Visto que a maior parte dos modelos só podem girar numa direção, você pode acionar 8 modelos através das saídas unipolares O1 até O8.

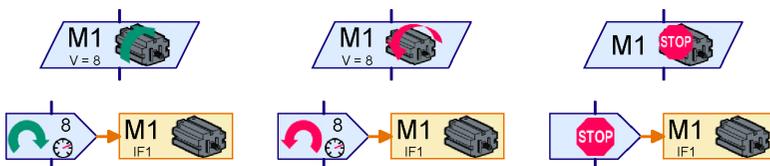
O diretor do museu poupa assim muito dinheiro. Para isso, você tem de copiar o programa 7 vezes e adaptar em todo o lado as entradas e saídas. Ou não? Não se poderia fazer também com subprogramas?

Claro que sim, mas aí teríamos um problema: quando você utiliza as consultas normais dos botões e elementos do motor do grupo **Elementos básicos** em um subprograma, cada chamada do subprograma consulta o mesmo botão e ativa os mesmos motores.

Isso se deve ao fato de, por exemplo em um elemento de saída do motor, o comando de controle para o motor (direita, esquerda ou parar) e o número de saída do motor (M1...M8) constituírem uma unidade. Visto que um subprograma só existe uma vez, está lá sempre o mesmo motor. Se você mudar o número da saída do motor em uma chamada do subprograma, também será alterado em todas as chamadas existentes do subprograma. Você também teria de copiar novamente 7 vezes o subprograma, dar um outro nome a cada subprograma e adaptar manualmente em todo o lado as entradas e saídas.

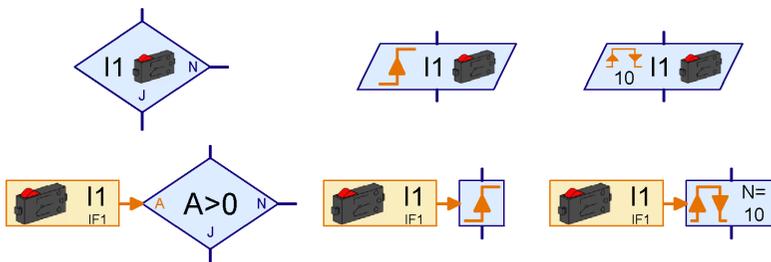


Mas também se pode resolver esse problema de uma maneira muito mais elegante. O truque é separar as ordens de controle dos símbolos do motor. Depois se pode programar as ordens de controle (esquerda, direita, parar) no subprograma e os elementos do motor no programa principal. Depois, no decorrer do programa você envia através de um elemento de comando, que você já conheceu nas variáveis, os comandos esquerda, direita ou parar ao programa principal, onde pode depois reencaimhar para vários motores. Para o motor existe um elemento do motor, que só representa um motor sem determinar o que o motor deve fazer. Esse elemento tem uma entrada de comando, para a qual você pode mandar comandos. Você pode substituir os elementos do grupo elementos básicos por um elemento de comando e por um elemento de motor, da seguinte maneira:



Na linha superior você vê respectivamente um elemento de motor do grupo **Elementos básicos**. Na segunda linha está a combinação de um elemento de comando do grupo **Comandos** e um elemento de motor do grupo **Entradas, Saídas**, que tem exatamente o mesmo efeito. Na realidade, os elementos superiores são só abreviaturas ou simplificações para as combinações na linha inferior. Ambas enviam ao motor **M1** um comando direita, esquerda ou parar.

O mesmo se aplica para a consulta de botões:



Na linha superior você volta a ver elementos de motor do grupo **Elementos básicos**. Na fileira inferior você encontra respectivamente uma combinação de uma entrada digital e um elemento do grupo **Ramificação, Aguardar, ...** Você encontra o elemento laranja **Entrada digital** e o elemento de motor no grupo **Entradas, Saídas**.

Com esse truque você pode separar a lógica de um processo de programa das entradas e saídas. Ainda falta algo. Se os elementos do motor e do botão tiverem de estar no programa principal e os comandos em um subprograma, então tem de existir uma maneira de ligar os botões e os elementos do motor ao subprograma. Você encontra os elementos de conexão necessários para fazer isto no grupo **Subprograma I/O**.

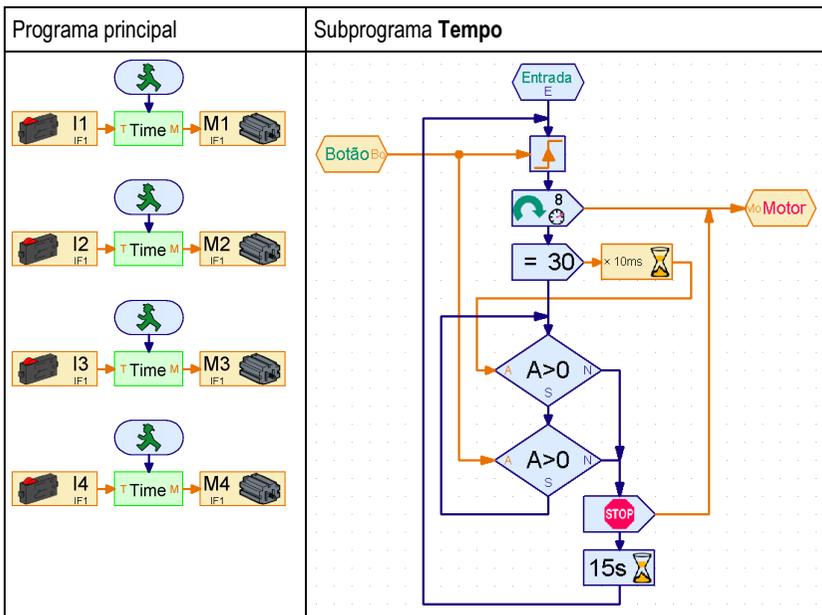


Você pode enviar comandos do exterior para um subprograma utilizando uma entrada de comando do subprograma. O elemento de entrada digital (botão) envia o seu novo valor através da linha laranja quando o estado da entrada se altera (com um chamado “=Comando”). Você pode dar um nome à entrada no campo de diálogo do elemento.



Você pode enviar comandos para o exterior a partir de um subprograma utilizando uma saída de comando do subprograma. Você pode então por exemplo enviar os comandos esquerda, direita ou parar a partir de um subprograma para um motor. Você também pode introduzir um nome nesse elemento no campo de diálogo.

Agora você tem tudo reunido para o seu **Temporizador modelo múltiplo com subprogramas**:



O **subprograma Tempo** é quase igual ao programa do capítulo anterior. No entanto, os elementos **Aguardar pela entrada digital I1** no início e no ciclo foram substituídos pelos elementos **Aguardar por** com plugues de dados para linhas laranja do grupo **Ramificação, Aguardar,...** Ambos estão ligados à entrada de comando do subprograma **Botão**. Os dois elementos de comando do motor no princípio e no fim do programa foram substituídos por elementos de comando. Ambos enviam os seus comandos à saída de comando do subprograma **Motor**.

No **Programa principal** o subprograma **Tempo** é chamado quatro vezes. A entrada de comando **Botão** do subprograma criou automaticamente a conexão **T** no símbolo verde do subprograma no



lado esquerdo. Através da saída de comando **Motor** do subprograma, apareceu a conexão **M** no lado direito. A conexão **T** do símbolo do subprograma está conectada respectivamente com um dos botões **I1** até **I4**. Na conexão **M** é respectivamente conectado um dos motores **M1** até **M4**. Deste modo, cada chamada do subprograma **Tempo** consulta um outro botão e comando um outro motor!

Tente recriar o subprograma anterior e o programa principal e experimente. Em primeiro lugar você tem de conceber o subprograma, porque senão você não consegue inserir o subprograma no programa principal. Se você tiver dificuldades com o subprograma, leia outra vez o capítulo 4 *Level 2: Trabalhar com os subprogramas* na página 468 .

5.6 Listas (Arrays)

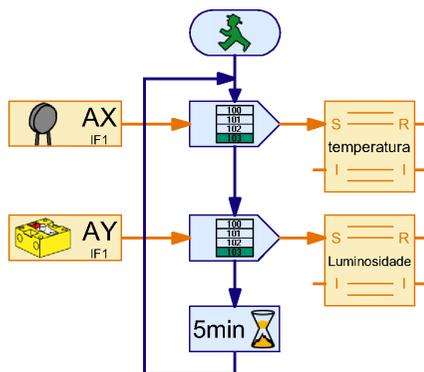
Depois de todas as construções de ensaio no museu estarem equipadas com o seu controle de poupança de custo, o próximo problema do diretor do museu não se faz esperar: em uma sala com peças de exposição antigas e valiosíssimas, surgiram nos últimos tempos oscilações de temperatura danosas. Você pensa que isso poderá estar relacionado com a radiação solar. Para poder provar essa relação, você quer construir um aparelho que grave a luminosidade e a temperatura. A ROBO Interface tem mais entradas analógicas e você já sabe como pode salvar valores com a ajuda de variáveis. Então isso não deve ser problema, pois não? Para poder gravar dois valores a cada 5 minutos acima de 12 horas são necessárias 288 variáveis! No entanto, isto iria se tornar em um programa enorme e complicado. Será possível também simplificar outra vez com subprogramas? Sim, mas existe um caminho muito melhor: o elemento **Lista** (os programadores lhe chamam "Array").

Em uma lista é possível salvar não só um valor, mas também uma lista inteira. No início, a lista está normalmente vazia. Quando você enviar um comando **Anexar** à entrada de dados superiores à esquerda com a designação **S**, o valor inserido neste elemento de comando é anexado ao final da lista. Você pode ajustar a extensão máxima da lista entre 1 e 32767, utilizando a janela Propriedades do elemento **Lista**. Com isto, o programa para a gravação da temperatura e da luminosidade se torna muito fácil:



O sensor de temperatura está conectado na entrada analógica **AX** e sensor de luminosidade na entrada analógica **AY**. O programa lê em um ciclo todos os 5 minutos ambos os valores e os introduz através do comando **Anexar** em uma lista.

Nota: Ao inserir o elemento de comando, você tem de ativar na janela Propriedade a opção **Entrada de dados para valor de comando**. Depois aparece à esquerda no elemento de comando uma entrada de dados à qual você pode conectar a entrada analógica.

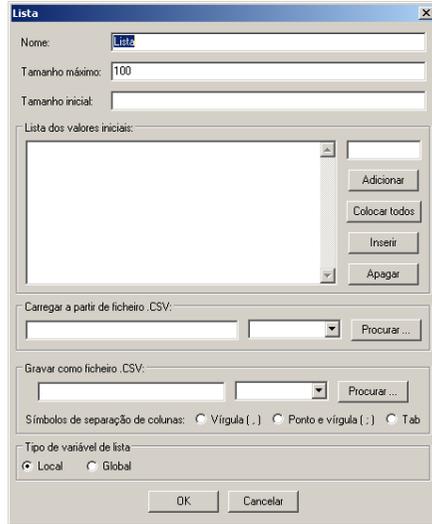


Para testar o programa, é útil diminuir o tempo de ciclo de 5 minutos para alguns segundos.

Agora você se pergunta certamente como é possível você voltar a ler os valores salvados da lista. Para isso existem duas possibilidades: Você pode ler os valores como em uma variável normal e os processar no seu programa. Visto que a lista contém vários elementos, você seleciona primeiro

na entrada de dados à esquerda o número do elemento com a designação **I** que você quer ler. Depois, o valor que este elemento tem é depois entregue no lado direito na saída de dados **R**.

O ROBO Pro pode também salvar todos os valores da lista em um arquivo em seu computador, que depois você pode por exemplo trabalhar em Excel. Visto que agora você só pretende ver e comparar a luminosidade e a temperatura gravadas, isto é seguramente mais prático. O ROBO Pro salva os valores em um chamado **arquivo CSV** (comma separated values = valores separados por vírgulas). Os arquivos CSV são arquivos de texto que têm uma ou mais colunas com uma fileira de dados. Você também pode salvar outras séries de medição como temperatura e luminosidade em várias colunas em um arquivo CSV. As colunas estão separadas por vírgulas. Em países onde se escreve 0,5 com vírgula e não 0.5 com ponto (p. ex. na Alemanha), se utiliza frequentemente como separador para as colunas um **ponto e vírgula (;)**. Se você tiver problemas ao trocar arquivos CSV entre o ROBO Pro e por exemplo o Microsoft Excel, pode alterar o **símbolo de separação das colunas** na janela Propriedades da lista.



Você pode determinar o nome do arquivo CSV e a coluna onde o conteúdo deve ser salvo na janela Propriedades da lista em **Salvar arquivo CSV**. Os dados são salvados quando o programa é terminado em modo Online, ou quando você selecionar no menu **Arquivo** o ponto **Salvar arquivos CSV**, contanto que o programa ainda esteja a funcionar (Online ou Download). No modo Download você pode desconectar a ROBO Interface do PC para a gravação dos dados e depois voltar a conectar para salvar.

Depois de você ter executado o programa anterior no modo Online, você pode abrir o arquivo .CSV criado pelo ROBO Pro com os dados no Microsoft Excel ou em um outro programa de cálculo de tabelas. Se você não tiver nenhum programa de cálculo de tabelas, você também pode utilizar o Windows Editor (Notepad.exe), que encontra no menu inicial do Windows em Acessórios.

Enquanto o programa estiver em modo Online, você também pode visualizar os dados em uma lista, fazendo clique com o botão direito do mouse no elemento lista.

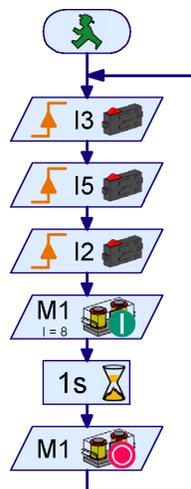
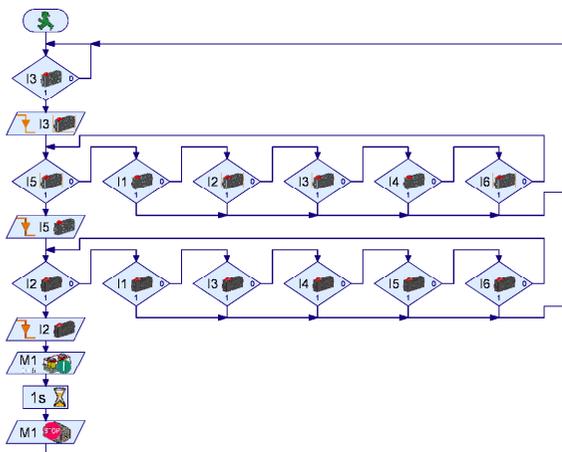
5.7 Operadores

O seu programa para gravar a luminosidade e temperatura funcionou bem, mas a gravação mostra que a temperatura na sala de exposição do museu não tem nada a ver com o sol. Vem-se a descobrir que alguns visitantes confundiram o controle do clima na sala de exposições com um controle do modelo e andaram a pressionar despreocupadamente os botões. Assim, não é de admirar que a temperatura da sala de exposições esteja descontrolada!

Mas este problema resolve-se facilmente com uma fechadura eletrônica com combinação numérica. A fechadura com combinação numérica deve ter um campo com teclas de 1 a 6.

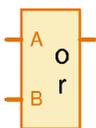
Quando são introduzidos sequencialmente 3 números corretos, a fechadura de teclas deve desbloquear a cobertura do controle de clima através de um magneto.

À primeira vista, esta fechadura de teclas é muito simples: O programa aguarda até que as teclas corretas sejam acionadas na sequência correta. Do lado direito está visível um programa para a combinação 3-5-2. No entanto, à segunda vista, este programa tem um problema: É muito fácil quebrar a fechadura, pressionando 3 vezes sequencialmente todas as teclas de 1 a 6. Assim, tem-se em todo o caso a certeza de que se pressionou a tecla correta. Como já dizia Albert Einstein: „Deve-se fazer as coisas da maneira mais simples possível – mas não mais simples“. O programa não deve só questionar se foram pressionadas as teclas corretas, mas também se foi pressionada alguma tecla incorreta. Assim, o programa tem o seguinte aspecto:



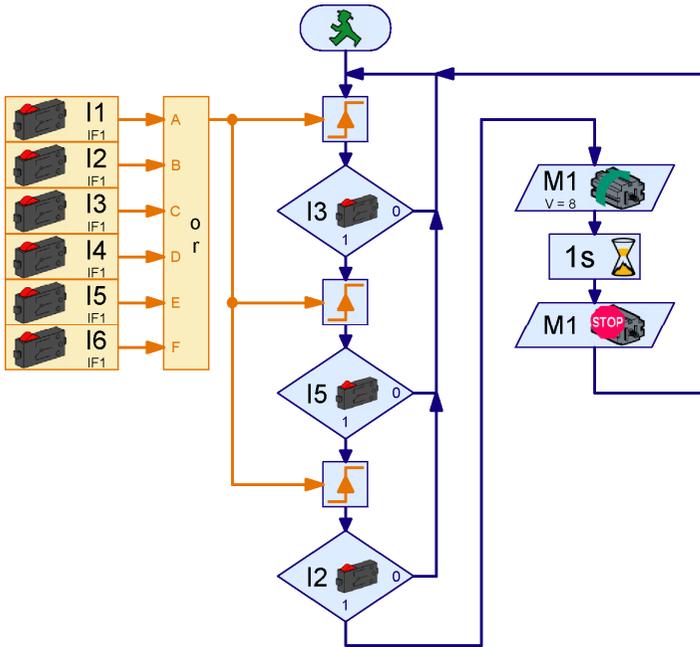
Este programa só desbloqueia a fechadura, se as teclas 3 5 2 forem pressionadas sem premir outras teclas por meio deste código. Se, por exemplo, for pressionada a tecla 3, o programa espera primeiro até que a tecla seja solta. Se a seguir for pressionada outra tecla que não seja o 5, o

programa reinicia. O programa funciona bem, mas não de um modo simples e fácil. Adicionalmente é relativamente difícil alterar o código. Mas não tenha medo, também pode ser feito de um modo correto e fácil através dos **operadores**. Existem vários tipos de operadores. Encontram-se no **elementos de programas** no grupo **operadores**. Para a fechadura com combinação numérica precisamos, em primeiro lugar, de um **operador ou**.



Podem ser conectados vários sinais nas entradas do **operador ou** (Inglês **or**). O operador fornece na saída sempre 1 quando pelo menos uma das saídas é de 1 (ou maior 0). Quando se conectam vários botões à saída **operadores ou**, a saída do operador é sempre de 1, quando pelo menos um botão é pressionado. O número das entradas pode ser regulado até 26 através da janela propriedades do operador.

Então, também se pode conectar todos os 6 botões a um operador. Agora deve estar a questionar-se como se pode deste modo simplificar uma fechadura com combinação numérica? É muito simples: com o operador pode primeiro aguardar todos os passos até que qualquer tecla seja pressionada. Depois pode controlar se é a tecla correta. Agora precisa para cada dígito em vez de 7 elementos de programa, apenas 2.



Os botões nas entradas I1 a I6 são resumidos através de um operador ou com 6 entradas. Quando pelo menos um botão foi pressionado, o operador ou fornece um valor de saída de 1, senão 0. Com um elemento **Aguardar por** o programa aguarda até que um botão seja pressionado. Logo a seguir, é controlado se era a tecla correta. Se sim, aguarda-se que outra tecla seja pressionada. Caso tenha sido pressionada a tecla incorreta, o programa reinicia.

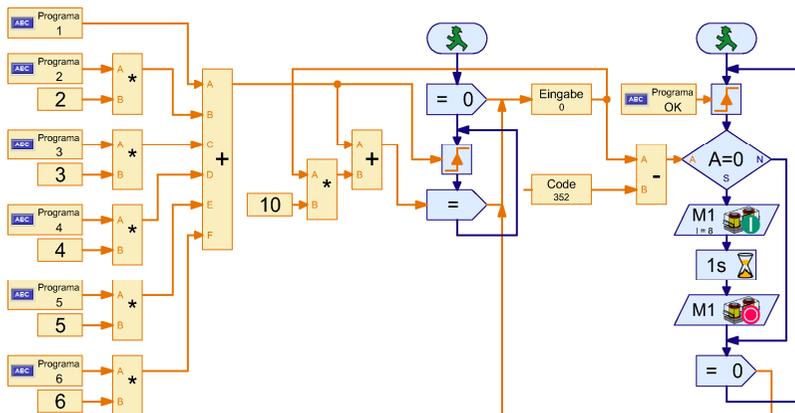


Altere o programa de cima de tal modo que utilize elementos de comando no painel de controle em vez de botões. Para isto, desenhe um painel de comando com 6 botões com as inscrições de 1 a 6. Depois, altere as entradas digitais através da janela propriedades. A ramificação tem de ser substituída através da ramificação com entrada de dados e entrada do painel.



A fechadura com combinação numérica funciona agora perfeitamente, mas alterar o código (3 5 2) continua a não ser fácil. Para isto, é necessário alterar as entradas em três elementos de ramificação. Para o controle do clima no museu não é necessário alterar regularmente o código, mas quando a fechadura de código é, por exemplo, aplicada num sistema de alarme deve querer alterar o código com uma certa regularidade. Mais simples seria a possibilidade de memorizar o código em variáveis. Deste modo seria possível alterar o código automaticamente. Quando, por exemplo, é disparado um alarme silencioso no sistema de alarme, seria possível substituir o código normal por um código de alarme especial.

Para a comparação dos registros com as variáveis de código se tornar possível, é necessário memorizar os registros também em variáveis. No início a variável deve ter um valor de introdução de 0. Ao pressionar agora a tecla 3, a variável tem o valor 3. Depois, quando a seguir pressionar a tecla 5, um valor de 35, e depois a tecla 2 com um valor de 352.



A fechadura com combinação numérica tem dois processos. No processo esquerdo, é atribuído um número a cada tecla através de alguns operadores multiplicação e um operador mais. A tecla 1 o número 2, a tecla 2 o número 2, etc. As teclas fornecem um valor de 1 ou 0 e, um valor de 0 ou X, quando se multiplica o valor com um número fixo X. Como os valores são de 0 quando as teclas não estão a ser pressionadas, é possível adicionar todos os valores, obtendo assim o valor de tecla em forma de número. Logo que uma tecla seja pressionada, é atribuído à variável de introdução 10 vezes o valor anterior, mais o valor da tecla pressionada. A multiplicação por 10 faz avançar o valor, até aqui atual, da variável de introdução uma casa decimal (por exemplo 35 fica 350).

O processo direito aguarda até que depois do código ter sido inserido, seja pressionada a tecla OK do painel de controle. Depois o código da variável de código, com o valor de 352 quando o código foi inserido corretamente, é comparado com a variável de introdução. Se ambos têm o mesmo valor, o magneto de código é ativado, caso contrário, não é ativado. Por fim, a variável de introdução volta para 0. Os registros de variáveis e o código são comparados ao comparar a sua diferença com 0. Também poderíamos ter utilizado um elemento de comparação.



Ao pressionar duas teclas simultaneamente, os seus valores são adicionados um ao outro. Ao pressionar, por exemplo, 3 e 6, o valor é 9. Assim é possível ativar uma fechadura super secreta para qual é às vezes preciso pressionar várias teclas simultaneamente. Reflita que teclas vai pressionar e em que sequência, para que a fechadura abra no caso de um código de 495. Tenha em atenção que o elemento **Aguardar por...** prossiga com o programa quando o valor aumenta e não só quando altera de 0 para 1.



A fechadura com combinação numérica também funciona com códigos de 2 ou 4 caracteres? Se sim, funciona até quantos caracteres e porquê? Como funcionam outros programas de fechaduras com combinação numérica?

6 Ativar módulos de ampliação e várias interfaces

Com uma ROBO Interface ou uma Intelligent Interface já poderá ativar modelos bastante dispendiosos. Mas é provável que alguém possa gostar das coisas um pouco mais complicadas. Caso 8 botões, 4 motores e 4 entradas analógicas não cheguem para você, pode ampliar uma ROBO Interface com até 3 **ROBO I/O-Extensions** ou com a sua Intelligent Interface com um **Extensionmodul**. E para quem isto não é suficiente, ainda existe a possibilidade de ativar várias interfaces no modo online a partir de um programa (cada um deles provavelmente com módulos de ampliação). A quantidade das entradas e saídas é assim, pelo menos no modo online, só limitada pela capacidade do seu PC.

6.1 Módulos de ampliação

Talvez já tenha reparado na lista de seleção em **Interface / Extension** nas janelas de propriedades para elementos de saída e entrada. Aí pode selecionar em que módulo interface ou extension se encontra a entrada ou saída. Se não configurou mais nada (ver próximo capítulo), a lista tem os seguintes registros:

- **IF1:** Esta é a interface principal (nenhum módulo de ampliação)
- **EM1..EM3:** Estes são os módulos de ampliação 1 a 3. Se puder utilizar uma interface inteligente, só poderá utilizar um módulo de ampliação (**EM1**). Mas os módulos de ampliação **EM2** e **EM3** também são indicados ao trabalhar com uma Intelligent Interface, visto que os programas não podem depender do uso de uma ROBO Interface ou uma Intelligent Interface.



Vemos que é muito fácil utilizar módulos de ampliação como **ROBO I/O Extensions**. Para tal, apenas precisa do módulo desejado (Interface ou Extension) nas entradas e saídas.

6.2 Várias Interfaces

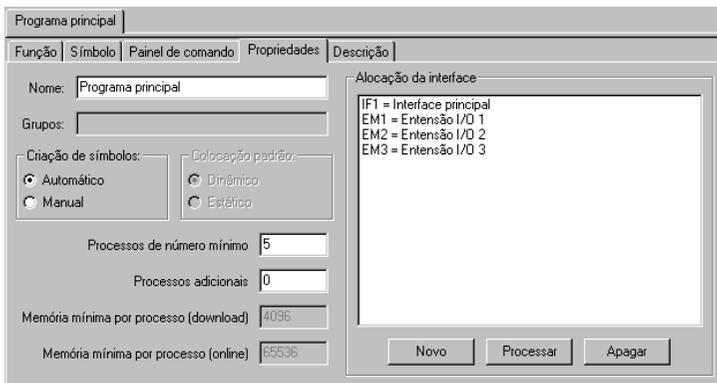
Ativar várias interfaces a partir de um programa, só funciona no modo online. Pode, por exemplo, conectar uma ROBO Interface a COM1, e um segundo a USB. Ambos podem estar equipados com até 3 ROBO I/O-Extensions. Uma outra possibilidade de conexão seria. P. Ex., 2 ROBO Interfaces a USB (ambos com I/O-Extensions) e uma Intelligent Interface a COM1 (também possível com módulo de ampliação). Para poder definir na janela propriedades das entradas e saídas que interface deve ser ativada, é necessário adaptar a alocação da interface.

Enquanto não configurar mais nada, encontrará nas listas de seleção **Interface / Extension** das entradas e saídas os registros **IF1**, **EM1**, **EM2** e **EM3**. Mas também pode alterar ou ampliar a lista. Para isso pode haver várias razões:

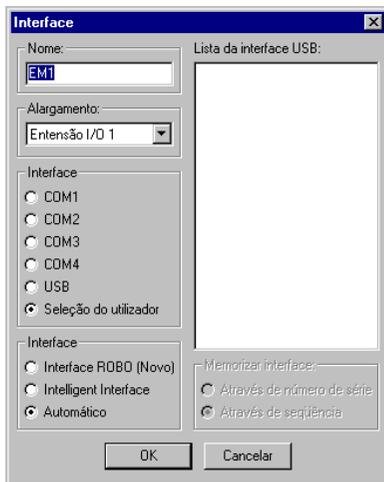
- Para uma melhor visualização, quer dar aos módulos outro nome, que mostra que parte da máquina ou do robô o módulo está a ativar, em vez de IF1 ou EM1.
- Quer trocar dois módulos de ampliação (por exemplo EM1 e EM2) porque funciona melhor com os cabos mas, não altera o seu programa.

- Quer que um programa, feito para um ROBO Interface com mais do que um módulo de ampliação, trabalhe com várias Intelligent Interfaces.
- Quer utilizar no seu programa mais do que uma ROBO Interface ou mais do que uma Intelligent Interface.

Podemos fazer tudo isto ao alterar a **alocação da interface** na janela propriedades do **programa principal**:



Aqui pode ver a que módulos (Interface ou Extension) são atribuídos os nomes **IF1** até **EM3**. Com o Button **Novo** pode adicionar uma nova interface. Se pretende alterar o registro na lista, seleccione este e clique **processar**. Em ambos os casos abre a seguinte janela:



- Em **Nome** pode alterar o nome do módulo. O nome não deve ser muito longo, visto que o espaço para o nome da interface nos símbolos gráficos é muito pequeno. Na maioria das vezes em que altera o nome, também vai ter que alterar o nome do módulo em todos os elementos de entrada e saída que utilizam esse mesmo nome.
- Em **Ampliação** pode ajustar se o nome se refere a uma Interface ou a um dos módulos de ampliação de 1 a 3.
- Em **Interface** pode seleccionar a qual interface a interface está ligada. Se introduzir aqui **Seleção do utilizador**, é utilizada a **Interface seleccionada por você na barra de ferramentas em COM/USB**. É mais fácil utilizar apenas uma interface com vários módulos de ampliação, mas não várias interfaces, porque assim uma outra pessoa também poderá utilizar o seu programa sem alteração. Ao utilizar várias interfaces, você ajustará aqui a interface a qual a respectiva interface se encontra ligada.
- Em **Interface** pode indicar se quer utilizar uma ROBO Interface ou uma Intelligent Interface. Se a interface estiver ligado a uma interface serial, o programa pode reconhecer automaticamente de que interface se trata (Seleção **automática**).



- O lado direito da janela só é importante quando várias ROBO Interfaces estão ligadas ao USB Bus. Ao clicar **USB** em **Interface** pode selecionar em **USB lista interface** uma das interfaces.

Atenção: Se quiser operar várias interfaces no USB-Bus, terá de atribuir a cada interface um número de série próprio: Por norma todas as interfaces são fornecidas com o mesmo número de série para evitar problemas na troca de interfaces. Contudo, o sistema operacional Windows Betriebssystem apenas reconhece interfaces com números de série diferentes. Você pode obter mais informações no capítulo 6.5 *Alterar o número de série da interface ou da versão Firmware* pág. 494.

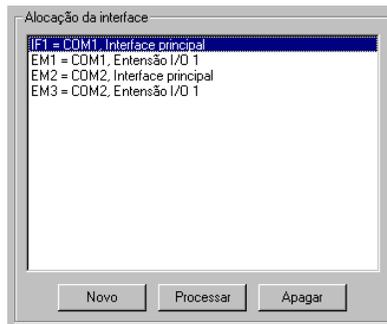
- Em **Memorizar Interface** pode ajustar de que modo o programa memoriza as interfaces. Para isso existem duas possibilidades: Se selecionar **através do número de série**, o programa memoriza o número de série da interface. Mesmo ao adicionar ou deletar outras Interfaces ao USB-Bus, o programa continua podendo encontrar sempre a Interface selecionada através do número de série. No entanto, isso tem a desvantagem do programa passar a trabalhar só com uma interface com o mesmo número de série. Se quiser utilizar uma outra interface com outro número de série, tem de alterar a alocação da interface ou o número de série desta. Para evitar problemas com o número de série, existe uma segunda possibilidade **Através de sequência**. Se selecionar esse ponto, o programa memoriza a sequência da lista em vez do número de série. Isso pode causar confusão ao adicionar ou eliminar interfaces ao USB-Bus mas, o programa continua inalterado com qualquer interface.

6.3 Alocação da interface nos subprogramas

Normalmente quaisquer alocações da interface para o programa são feitas na janela propriedades do programa principal. Mas também pode inserir interfaces nos subprogramas. Depois, pode utilizar, no subprograma, alocações da interface do subprograma e do programa principal. Quando duas alocações têm o mesmo nome, é a alocação no subprograma que tem prioridade. Assim, pode, por exemplo, definir para que o programa principal IF1 possa aceder à interface principal mas que, num subprograma determinado IF1, esteja na posição de um módulo de ampliação. Isso pode ser muito prático quando quer comandar uma maquinaria completa, sendo cada uma das máquinas comandadas por uma interface própria. Pode, assim, desenvolver o comando para cada máquina sob forma de programas independentes, acedendo cada programa principal a IF1. Depois, pode inserir todos os programas principais de máquinas num programa completo sob a forma de um subprograma. A seguir, só precisa de alterar as alocações de interface no programa e não o nome em cada entrada e saída.

6.4 Dicas & Truques

Se quiser executar um programa, desenvolvido para uma ROBO Interface com 3 módulos de ampliação, com 2 Intelligent Interfaces, cada uma com um módulo de ampliação, pode utilizar a coordenação de interface representada. Os módulos de ampliação 2 e 3 são substituídos por mais uma Intelligent Interface com um módulo de ampliação em COM2.

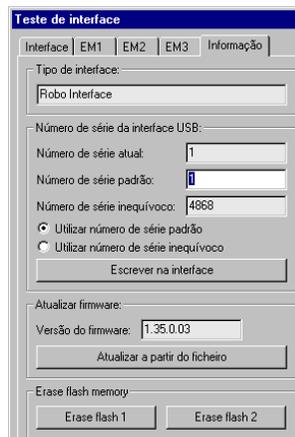


6.5 Alterar o número de série da interface ou da versão Firmware

Todas as ROBO Interfaces e ROBO I/O-Extensions são, por norma, fornecidas com o mesmo número de série. Usar apenas uma interface num computador é o mais prático, visto que deste modo todas as interfaces do computador têm o mesmo aspecto e, assim, não existem problemas na troca de interfaces. Mas, se pretender usar mais do que uma interface num computador através de USB, é necessário alterar, em primeiro lugar, o número de série da interface para que o computador possa distinguir e aceder à interface. Isso não é preciso ao aceder à interface através de várias interfaces seriais.

Para alterar o número de série da interface, deve proceder da seguinte forma:

- Ligue a interface **individualmente** ao USB Bus do computador.
- Clique na barra de ferramentas no botão **COM/USB** e selecione a interface USB.
- Abra a janela de teste interface através do botão **Teste** na barra de ferramentas e mude para a barra **Info**:
- Em **tipo de interface** pode ver o tipo da interface, por exemplo, é indicada **ROBO Interface** ou **ROBO I/O Extension**.
- Em **número de série interface** pode ajustar o número de série que a interface utiliza ao iniciar. Cada interface tem dois números de série instalados, um **número de série padrão** que é 1 enquanto não ajustar mais nada, e um **número de série inequívoco** que não pode ser ajustado e que é para cada interface diferente. A maneira mais simples para utilizar várias interfaces no USB-Bus, consiste em alterar o botão de seleção em cada interface, para **utilizar número de série inequívoco**. Assim, fica garantido que cada interface tenha um número de série próprio e inconfundível. Se utilizar muitas interfaces para um modelo pode-se tornar pouco prático de memorizar todos os números de série. Neste caso seria mais simples ajustar e utilizar o número de série padrão da sua interface para, por exemplo, 1, 2, 3 etc. Depois de ter alterado e selecionado o número de série, ainda tem de pressionar o botão **Escrever na**



interface. Depois de ter alterado o número de série, tem de separar a interface da alimentação de corrente e voltar a conectar.

Atenção: Quando o número de série é alterado, o controlador tem de, eventualmente, voltar a ser instalado; para isso precisa de direitos do administrador para Windows NT, 2000 e XP. Quando alterar o número de série, mas não poderá instalar o controlador porque não tem direitos de administrador, deixará de poder aceder à interface através de USB. Neste caso pode separar a interface da corrente de alimentação e, ao voltar a ligar a corrente, manter o botão **Port** pressionado. Assim, a interface inicia com o número de série 1 e volta a ser reconhecido pelo controlador já instalado. Mas ao fazer isso, o número de série não é alterado permanentemente, ou seja, quando voltar a iniciar sem o botão Port, volta a ter o número de série anterior. Para alterar o número de série permanentemente deve proceder como acima descrito.

- Em **Atualizar Firmware** pode, finalmente, atualizar o programa de controle interno das suas ROBO Interfaces caso a fischertechnik lance no futuro uma nova versão da Interface-Firmware.

7 Vista geral dos elementos do programa

Todos os elementos do programa disponíveis em ROBO Pro estão, a seguir, divididos em grupos e descritos na sequência em que são representados na janela dos elementos.

7.1 Elementos básicos (Level 1)

7.1.1 Início



Um processo num programa começa sempre com um elemento inicial. Se esse elemento de programa faltar, o processo não é efetuado. Quando um programa contém vários processos, cada processo tem de iniciar com um componente inicial. Os processos diferentes são, assim, iniciados simultaneamente.

Um elemento inicial não tem propriedades que você possa alterar. Por isso, ao contrário da maior parte dos elementos, não abre **qualquer** janela propriedades quando clica no elemento com o botão direito do mouse.

7.1.2 Final

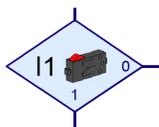


Para finalizar um processo é necessário conectar a saída do último elemento a um elemento final. Um processo também pode ser finalizado em locais diferentes com esse elemento. Também existe a possibilidade de conectar as saídas de diferentes elementos a um único componente final. Mas também pode acontecer

que um processo tenha sido efectuado sob forma de ciclo infinito e, assim, não contenha qualquer elemento final.

Um elemento final não tem propriedades que você possa alterar. Por isso, ao contrário da maior parte dos elementos, não abre **qualquer** janela propriedades quando clica no componente com o botão direito do mouse.

7.1.3 Ramificação digital



Com essa ramificação pode conduzir o processo do programa em duas direções, independentemente do estado de uma das entradas digitais **I1** até **I8**. Quando, por exemplo, um botão na entrada digital se encontra fechado (=1), o programa ramifica-se

com a saída **1**. Quando a entrada se encontra aberta (=0), o programa ramifica-se com a saída **0**.

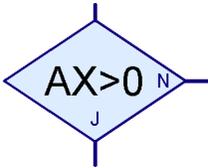
Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

- Com os botões **I1** até **I8** pode inserir que entrada da interface é consultada.
- Em **Interface / Extension** você pode selecionar se quer utilizar a entrada da Interface ou a entrada do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.



- Em **Imagem** pode selecionar uma imagem para sensor conectado na entrada. Nas entradas digitais na maioria das vezes são conectados botões mas também com frequência fototransistores ou contatos Reed.
- Em **Trocar as ligações 1/0** pode trocar a posição das saídas 1 e 0 da ramificação. Normalmente a saída 1 encontra-se em baixo e a saída 0 do lado direito. Mas, torna-se mais prático quando a saída 1 se encontra do lado direito. Pressione **trocar as ligações 1/0** para trocar ambas as ligações logo que feche a janela com OK.

7.1.4 Ramificação analógica



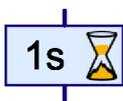
Adicionalmente às entradas digitais, a ROBO Interface tem 6 entradas digitais, 2 entradas de resistência AX e AY, 2 entradas de tensão A1 e A2 como também 2 entradas para sensores de distância D1

e D2. Com esta ramificação pode comparar o valor da entrada analógica com um número fixo e, conforme, se a comparação coincide ou não, ramificar com a saída Sim (J) ou Não (N).

Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

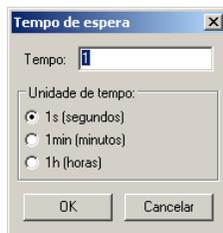
- Em **entrada analógica** pode selecionar que entrada da Interface deve ser consultada. Todas as entradas analógicas fornecem um valor entre 0 e 1023. Pode obter mais informações acerca das entradas analógicas diferentes no capítulo 7.6.2 *Entrada analógica* na página 514.
- Em **Interface / Extensão** você pode selecionar se quer utilizar a entrada da Interface ou a entrada do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Condição** pode selecionar um operador de comparação, como menor (<) ou maior (>), e selecionar o valor de comparação. O valor de comparação deve encontrar-se entre 0 e 1023. Quando inicia um programa com uma ramificação para entradas analógicas no modo online, é indicado o valor analógico atual.
- Em **Trocar as ligações S/N** pode trocar a posição das saídas 1 e 0 da ramificação. Normalmente a saída Sim (J) encontra-se em baixo e a saída Não (N) do lado direito. Mas, torna-se mais prático quando a saída Sim se encontra do lado direito. Pressione **Trocar as ligações S/N** para trocar as ligações S e N logo que feche a janela com OK.

7.1.5 Tempo de espera



Com o elemento **Tempo de espera** pode retardar or um tempo determinado mais execuções de um processo .

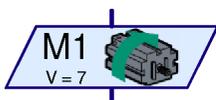
Se clicar com o botão direito do mouse no elemento é indicada a janela Propriedades. Aqui pode inserir o tempo de espera em segundos, minutos ou horas. A zona de ajuste para o tempo de espera é de um milissegundo (isso é milésimo de segundo) até 500 horas (mais ou menos 3 semanas). No entanto, quanto maior o tempo de espera, mais incerta se torna a medição de tempo.



A seguinte lista mostra a precisão de vários tempos de espera:

Tempo de espera	Precisão
Até 30 segundos	1/1000 segundo
Até 5 minutos	1/100 segundo
Até 50 minutos	1/10 segundo
Até 8,3 horas	1 segundo
Até 83 horas	10 segundos
Até 500 horas	1 minuto

7.1.6 Saída do motor



Com o elemento do programa **saída do motor**, conecta-se uma das duas saídas com 2 pólos M1-M4 da interface. As saídas da interface podem tanto ser utilizadas para motores como também, para lâmpadas ou magnetos elétricos. Num motor quer-se, para além da velocidade, também ajustar o sentido de rotação.

Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

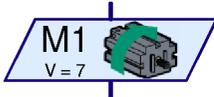
- Em **saída do motor** você pode ajustar que saída, das 4 saídas do motor, **M1** até **M4**, vai ser utilizada.
- Em **Interface / Extension** você pode selecionar se quer utilizar uma saída da interface ou uma saída do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** pode selecionar uma imagem que representa o componente fischertechnik conectado à saída.
- Em **Ação** pode ajustar de que modo a saída deve ser influenciada. Um motor pode ser operado do lado esquerdo ou direito ou, pode ser parado. Caso uma lâmpada seja



conectada à saída do motor (ver dica em saída de lâmpada), essa pode ser ligada e desligada.

- Por fim, também pode indicar uma **velocidade** ou **intensidade** entre 1 e 8. 8 é a velocidade, luminosidade ou força magnética mais elevada, 1 a menor. É obvio que a velocidade não tem de ser indicada para a paragem ou desconexão.

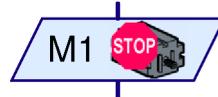
Aqui encontra uma lista com os símbolos para algumas ações e imagens:



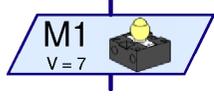
Motor lado direito



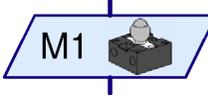
Motor lado esquerdo



Paragem do motor



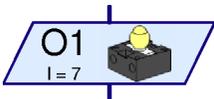
Lâmpada ligada



Lâmpada desligada

Dica: Um motor pode, por vezes, funcionar numa só direção, p. ex. no caso de uma cinta transportadora. Nesse caso pode utilizar uma saída de lâmpada para o motor, de maneira a que uma conexão fique menos gasta.

7.1.7 Saída de lâmpada (Level 2)



Com o elemento do programa **saída de lâmpada** se conecta uma das saídas de 1 pólo 01-08 da interface. As saídas da interface podem ser utilizadas aos pares como saídas do motor (ver em cima) ou individualmente como saídas da lâmpada O1-O8. As saídas da lâmpada ocupam, ao contrário da saída do motor, apenas um pino de ligação. Desta forma, pode ativar separadamente 8 lâmpadas ou válvulas magnéticas. Conecte a outra conexão da lâmpada à tomada da massa (\perp) da interface.

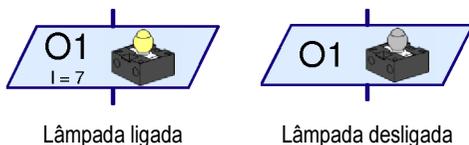
Dica: Se quiser conectar apenas quatro lâmpadas ou motores, pode também utilizar uma saída de motor para lâmpadas. Isto é mais prático porque assim pode conectar diretamente ambas as conexões da lâmpada na saída da interface e não tem de conectar todos os pólos negativos tomada da massa.

Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

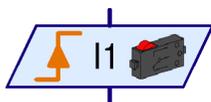
- Em **Saída da lâmpada** pode ajustar qual das oito saídas **O1** a **O8** deve ser utilizada.
- Em **Interface / Extension** você pode selecionar se quer utilizar uma saída da interface ou uma saída do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** pode selecionar uma imagem que representa o componente fischertechnik conectado à saída.
- Em **Ação** pode ajustar de que modo a saída deve ser influenciada. Pode ligar ou desligar uma lâmpada.
- Por fim, também pode indicar ainda uma **intensidade** entre 1 e 8. 8 é a luminosidade ou força magnética mais elevada, 1 a menor. Ao desligar, não precisa indicar a intensidade.



Aqui, estão listados os símbolos para as diferentes ações para a figura **Lâmpada**:



7.1.8 Aguardar pela entrada



O elemento **Aguardar pela entrada** aguarda até que o acesso da Interface tenha um determinado estado, ou se altere para um determinado tipo.

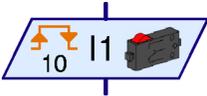
Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

- Em **Aguardar por** pode selecionar que tipo de alteração ou estado se deve aguardar. Se selecionou **1** ou **0** o elemento aguarda até que a entrada esteja fechada (**1**) ou aberta (**0**). Se selecionar **0 -> 1** ou **1 -> 0** o elemento aguarda até o estado de entrada se **alterar** de aberto para fechado (0->1) ou de fechado para aberto (1->0) . Na última possibilidade o elemento aguarda até que a entrada se altera independentemente se passa de aberto para fechado ou de fechado para aberto. No capítulo 3.6 *Outros elementos do programa* na página 462 se explica claramente como você pode construir este elemento com o elemento de ramificação.
- Na **Entrada Digital** você pode ajustar qual das 8 entradas digitais de **I1** a **I8** se deve consultar.



- Em **Interface / Extension** você pode selecionar se quer utilizar a entrada da Interface ou a entrada do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** pode selecionar uma imagem para sensor conectado na entrada. Nas entradas digitais na maioria das vezes são conectados botões mas também com frequência fototransistores ou contatos Reed.

7.1.9 Contador de impulsos



Muitos modelos robóticos da Fischertechnik utilizam também rodas dentadas movidas por impulsos. Estas rodas dentadas acionam um botão, em cada rotação, 4 vezes. Com essas rodas

movidas por impulsos pode ligar um motor numa quantidade definida de rotações, em vez de um determinado tempo. Para isso, tem de contar a quantidade de impulsos na entrada do interface.

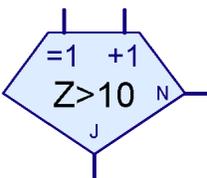


Com este objetivo existe o elemento **contador de pulsos**, que aguarda por uma quantidade configurada de impulsos.

Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

- Em **Tipo de impulso** você pode selecionar que tipo de impulso deve ser contado. Se seleccionar **0 -> 1** (crescente), o elemento aguarda até o estado de entrada se **alterar de aberto para fechado (0->1)** as vezes que você indicou em **Quantidade de impulsos**. Em **1 -> 0** (decrecente) o elemento aguarda até que o estado de entrada se altere de fechado para aberto, como indicado. Com rodas dentadas movidas por impulsos é utilizada muitas vezes a terceira possibilidade. Aqui, o elemento conta não só as alterações **0 -> 1** como também as alterações **1 -> 0**, de forma que a rotação de uma roda dentada movida por impulso seja de 8 impulsos.
- Na **Entrada Digital** pode ajustar qual das 8 entradas digitais de **I1** a **I8** se deve consultar.
- Em **Interface / Extension** você pode selecionar se quer utilizar a entrada da Interface ou a entrada do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** pode selecionar uma imagem para sensor conectado na entrada. Para as entradas digitais são utilizados, na maioria das vezes, botões mas também fototransistores ou contatos Reed.

7.1.10 Ciclo de contagem

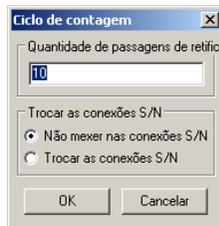


Com o elemento **Ciclo de montagem** pode executar, várias vezes, de forma muito simples, uma determinada parte do programa. O elemento do ciclo de contagem tem um contador incorporado. Quando o ciclo de contagem é ativado em **=1** entrada, o contador é colocado em 1. Quando o ciclo de contagem é ativado em **+1** entrada, o contador é contado em 1. Conforme o contador seja ou não maior do que um valor indicado por si, o ciclo de contagem se ramifica para a saída Sim

(**S**) ou para Não (**N**). Encontra um exemplo deste caso no capítulo 3.6.4 *Ciclo de contagem* na página 464.

Se clicar com o botão direito do mouse no elemento é indicado a janela Propriedades:

- Em **Quantidade de passageiros de ciclo** apresente quantas vezes o ciclo de contagem deve ser terminado na saída Não (**N**), antes da saída Sim (**S**) ser ativada. O valor indicado deve ser positivo.
- Quando pressiona **Trocar as ligações S/N** são trocadas as ligações **S** e **N** logo que feche a janela com OK. Dependendo de onde se encontram as ligações **S** e **N**, a parte do programa que é repetida, encontra-se à direita ou por baixo do ciclo de contagem.



7.2 Subprograma I/O (Level2-3)

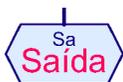
Neste grupo de elementos encontra elementos do programa, que apenas necessita em subprogramas.

7.2.1 Entrada do subprograma (Level 2)



Um subprograma pode ter uma ou mais entradas de subprograma. Através destas entradas o programa principal ou o subprograma superior, transfere a versão ao subprograma. No símbolo verde do subprograma, que é inserido no programa superior, é inserido para cada entrada do subprograma um pin de conexão na parte superior. As conexões no símbolo, têm a mesma sequência (da esquerda para a direita) que as entradas do subprograma no diagrama lógico do subprograma. Se clicar com o botão direito do mouse no elemento é indicada a janela Propriedades. Lá poderá atribuir um nome à entrada, que será depois indicado no símbolo. Pode obter mais informações relativas ao tema Subprogramas no capítulo 4 *Level 2: Trabalhar com os subprogramas* na página 468.

7.2.2 Saída do subprograma (Level 2)



Um subprograma pode ter uma ou mais saídas de subprograma. Através destas saídas o subprograma transfere a versão do programa de volta para o programa principal ou subprograma superior. No símbolo verde do subprograma, que é inserido no programa superior, é inserido para cada entrada do subprograma um pin de conexão na parte inferior. As conexões no símbolo têm a mesma sequência (da esquerda para a direita) que as saídas do subprograma no organigrama do subprograma. Se clicar com o botão direito do mouse no elemento é indicada a janela Propriedades. Lá poderá atribuir um nome à saída, que será depois indicado no elemento. Pode obter mais informações relativas ao tema Subprogramas no capítulo 4 *Level 2: Trabalhar com os subprogramas* na página 468.

7.2.3 Entrada de comando do subprograma (Level 3)



Através deste elemento podem ser ligados subprogramas com elementos de entrada, como p. ex. interruptores no programa principal ou no subprograma superior ou ser fornecidos a partir de lá, com valores de elementos de variáveis, p.ex. coordenadas. No símbolo verde do subprograma, que é inserido no programa superior, é inserido para cada entrada de comando do subprograma um pin

de conexão do lado esquerdo. As conexões no símbolo têm a mesma sequência (de cima para baixo) que as entradas de comando do subprograma no subprograma. Se clicar com o botão direito do mouse no elemento é indicada a janela Propriedades. Lá poderá atribuir um nome à saída, que será depois indicado no elemento. A utilização deste elemento encontra-se detalhadamente descrita no capítulo 5.5 *Entradas de comando para subprogramas* na página 483.

7.2.4 Saída de comando do subprograma (Level 3)



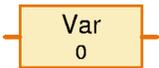
Através deste elemento do programa é possível enviar comandos como p. ex. para a esquerda, para a direita, parar, para os motores ou outros elementos de saída, que se encontram no programa principal ou no subprograma superior.

No símbolo verde do subprograma, que é inserido no programa superior, é inserido para cada saída de comando do subprograma um pin de conexão do lado direito. As conexões no símbolo têm a mesma sequência (de cima para baixo) que as saídas de comando do subprograma no subprograma. Se clicar com o botão direito do mouse no elemento é indicada a janela Propriedades. Lá poderá atribuir um nome à saída, que será depois indicado no elemento. A utilização deste elemento encontra-se detalhadamente descrita no capítulo 5.5 *Entradas de comando para subprogramas* na página 483.

7.3 Variável, lista, ... (Level 3):

Os elementos do programa neste grupo podem memorizar um ou mais valores numéricos. Através disso é possível desenvolver programas que têm uma memória.

7.3.1 Variável, (global)



Uma variável pode memorizar um único valor numérico entre -32767 e 32767. O valor das variáveis é aplicado, conectando na entrada de comando do lado esquerdo um = elemento de comando(ver capítulo 7.4.1 = (*Atribuição*) na página 509). É possível atribuir também a uma variável,

através da janela Propriedades, um valor inicial, que a variável manterá até receber o primeiro comando para alteração do valor.

O ROBO Pro atribui a cada elemento variável com **o mesmo nome e tipo de variável = global** apenas uma única variável. Todas as variáveis globais com o mesmo nome são idênticas e têm sempre o mesmo valor, mesmo quando aparecem em diferentes subprogramas. Quando um destes elementos variáveis é alterado através de um comando, alteram-se também todos os elementos variáveis globais e com o mesmo nome. Também existem variáveis locais (ver capítulo seguinte) com as quais a situação é diferente.

Adicionalmente ao comando =, uma variável também entende os comandos + e -. Se a variável receber, p. ex. o comando + 5, ela adiciona o valor 5 ao seu valor atual. No caso do comando - o valor transmitido com o comando é subtraído ao valor atual da variável.

Atenção:

Se o valor das variáveis exceder a área de valores das variáveis em caso de um comando + ou -, é acrescentado ou subtraído ao valor a variável 65536, de maneira a que o valor volte a ser válido. Uma vez que este é geralmente um comportamento a evitar, deve se certificar, que este não acontece.

Cada vez que o valor das variáveis se altera, envia um comando = com o novo valor a todos os elementos, que se encontram conectados com a saída de comando das variáveis. Se pretender

observar o valor de uma variável, pode conectar uma indicação de painel de comando à saída das variáveis (ver capítulo 7.6.6 *Entrada do painel de comando* na página 518).

São apresentados aqui novamente para uma vista geral, todos os comandos que o elemento variável pode processar:

Comando	Valor	Ação
=	-32767 a 32767	Coloca o valor da variável no valor transmitido pelo comando.
+	-32767 a 32767	Adiciona ao valor actual a variável transmitida pelo comando.
-	-32767 a 32767	Subtrai ao valor actual a variável transmitida pelo comando.

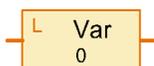
A área de valores sinuosa de -32767 a 32767 resulta do fato dos computadores efetuarem cálculos no sistema binário, e não no sistema decimal, como nós. No sistema binário 32767 é um número inteiro, qualquer coisa como 9999 no sistema decimal. Mas com isso, não necessitamos de nos preocupar, uma vez que o computador faz o cálculo de todos os números do sistema binário para o sistema decimal. Apenas nos valores máximos das variáveis isto se nota, e é possível aperceber-se um pouco desse facto quando, ao calcular, há uma superação de capacidade.

Janela Propriedades para variáveis

- Em **Nome** pode introduzir o nome para a variável.
- Em **Valor Inicial** pode introduzir o valor inicial para a variável. A variável mantém este valor, até receber um novo valor através de um comando =, + ou – .
- O ponto **Tipo de variável** apenas tem interesse para variáveis em subprogramas e será explicado detalhadamente no capítulo que se segue “Variável local” . Em caso de variáveis no programa principal, ambas as configurações têm o mesmo efeito.



7.3.2 Variável local



Todos os elementos de variáveis **globais** com o mesmo nome, utilizam uma e a mesma variável e têm sempre o mesmo valor. Isso é possívelmente o que você espera e o que, geralmente, também é prático. Mas se utilizar variáveis em subprogramas, podem surgir problemas graves. Se o seu programa tiver vários processos paralelos, é possível que um subprograma seja executado várias vezes num determinado momento. Uma situação destas provoca normalmente um caos, se o subprograma utilizar as mesmas variáveis em todos os processos. Por esta razão existem **variáveis locais**. Uma variável local comporta-se quase da mesma forma que uma variável global, com uma diferença: a variável local é válida apenas no subprograma em que é definida. Mesmo quando duas variáveis locais em diferentes subprogramas têm o mesmo nome, estas são variáveis diferentes e independentes. Também quando um subprograma é executado várias vezes paralelamente por vários processos, este tem em cada processo um conjunto independente de variáveis locais. As variáveis locais existem apenas durante o mesmo período de tempo em que o subprograma, no qual são definidas, seja executado. Ao valor inicial são atribuídas, por

isso, variáveis locais não no início do programa, mas sim em cada início do respectivo subprograma. Uma vez que um subprograma tem que fazer normalmente sempre o mesmo, cada vez que é chamado, é muito mais prático que as variáveis sejam colocadas no valor inicial, cada vez que este é chamado. As variáveis não têm, por assim dizer, nenhuma recordação de chamadas anteriores do mesmo subprograma.

No programa principal, as variáveis locais e variáveis globais comportam-se da mesma forma, uma vez que o programa completo e o programa principal são iniciados simultaneamente. As variáveis locais são, no entanto, um pouco mais eficientes na execução do programa. Também devem ser definidos de forma global, anteriormente, elementos de lista, porque a área de memória para variáveis globais é maior do que a para variáveis locais.

7.3.3 Constante

0

Uma constante tem um valor como uma variável, mas o valor não pode ser alterado pelo programa.

Pode ligar uma constante a uma entrada de dados de um símbolo de sub-programa, se o sub-programa deve utilizar sempre o mesmo valor. Mesmo em cálculos com operadores, as constantes são muito práticas. Pode encontrar um exemplo disso no fim do capítulo 5.7 *Operadores* na página 487.



7.3.4 Variável do temporizador

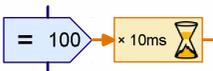
x 10ms



Uma variável de temporizador se comporta essencialmente como uma variável. A diferença existente entre variáveis normais e estáticas também existe nas variáveis do temporizador. A única diferença é que uma variável do temporizador conta até 0 em um ciclo de tempo fixo, o valor salvo.

Desde que o valor do temporizador seja 0, a variável do temporizador permanece inalterável. Se o valor do temporizador ficar negativo através de um comando menos -, o valor no próximo passo de tempo volta a 0.

Você pode determinar a velocidade à qual a variável do temporizador conta, escolhendo entre 1/1000 segundos por passo e 1 minuto por passo na janela Propriedades. Ao fazer isso, você deve ter atenção, pois a precisão do temporizador depende dos passos de tempo ajustados. Se você por exemplo colocar um temporizador em 1 x 10 segundos, o próximo passo de tempo de 10 segundos pode ocorrer passado pouco tempo (p. ex. passado um segundo) ou então só depois de 10 segundos. Assim, a precisão dos temporizadores depende dos passos de tempo ajustados. Por isso, você deve escolher pequenos passos de tempo, por exemplo 10 x 1 segundos ou 100 x 0,1 segundos em vez de 1 x 10 segundos. Você só deve escolher um passo de tempo de um minuto quando o programa tiver de esperar no mínimo uma hora. Depois, não depende de um minuto a mais ou a menos.



A quantidade de passos que devem ser contados para baixo, é destinada pela variável do temporizador normalmente através de um comando = a partir de um elemento de comando. No exemplo retratado são contados para baixo 100 passos cada um com 10ms. Isso corresponde a uma duração de 1000 ms=1 segundo. A precisão

é de 10ms.

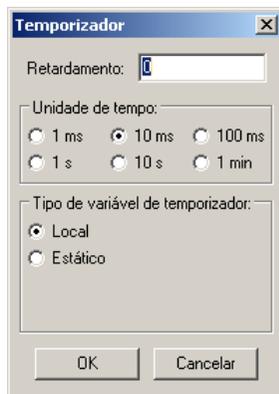
Com as variantes do temporizador é possível solucionar de forma simples tarefas de retardamento e de medição de tempo difíceis. Se por exemplo um robô interromper uma busca após 20 segundos, você pode no início da busca programar uma variável do temporizador de 20 x 1

segundo (ou 200 x 0,1 segundo) e depois consultar frequentemente no programa de busca, se o valor do temporizador ainda é superior a 0. Também em caso de sucesso parcial da busca você pode voltar a programar o temporizador para o valor inicial.

Se quiser medir um tempo, tem de posicionar a variável do temporizador no início num valor positivo, se possível, grande (30000 ou 32767) para sobrar muito tempo até o valor do temporizador ser 0. Se você quiser saber quando tempo passou desde então, você subtrai o valor atual do temporizador ao valor inicial.

Janela Propriedades para variáveis temporizador

- Em **Retardamento** pode determinar o valor inicial para a variável do temporizador. Normalmente se insere 0 e se programa o valor da variável do temporizador com um comando = para o momento correspondente. No entanto, se o temporizador arrancar no início do programa ou de um subprograma, aqui pode ser inserido o valor respectivo.
- Em **Unidade de tempo**, você pode selecionar o tamanho dos passos de tempo, nos quais a variável do temporizador é contada para baixo.
- Em **Tipo de variável de temporizador** pode configurar se o Temporizador é uma variável global ou local (ver capítulo 7.3.2 *Variável local* na página 504).



7.3.5 Lista



O elemento **Lista** corresponde a uma variável, com a qual se pode salvar não só um valor mas também outros valores. A quantidade máxima de valores que podem ser salvados numa lista é determinado na janela Propriedades do elemento.

Na lista atrás, pode anexar valores e deletar valores no final da lista. Além disso, pode alterar, ler ou trocar um valor qualquer na lista, pelo primeiro valor da lista. Introduzir ou deletar um valor no centro ou no início da lista não é possível fazer diretamente. Pode-se escrever um subprograma correspondente que efectue estas funções.

As seguintes funções de uma lista são usadas enviando comandos ao elemento na saída **S** (para escrita). Podem ser enviados os seguintes comando à entrada **S**:

Comando	Valor	Ação
Anexar	-32767 a 32767	Anexa o valor transmitido com o comando no final da lista. A lista é aumentada em um elemento. Quando a lista já tem a dimensão

		máxima, o comando é ignorado.
Deletar 	0 a 32767	Apaga o número de elementos indicados no final da lista. O valor transmitido com o comando é a quantidade de elementos a ser apagados. Quando a quantidade é maior do que a quantidade de elementos na lista, todos os elementos da lista são apagados. Se a quantidade for 0 ou negativa, o comando é ignorado.
Trocar 	0 a 32767	Troca o elemento indicado pelo primeiro elemento da lista. O valor transmitido com o comando é o número do elemento que deve ser trocado.

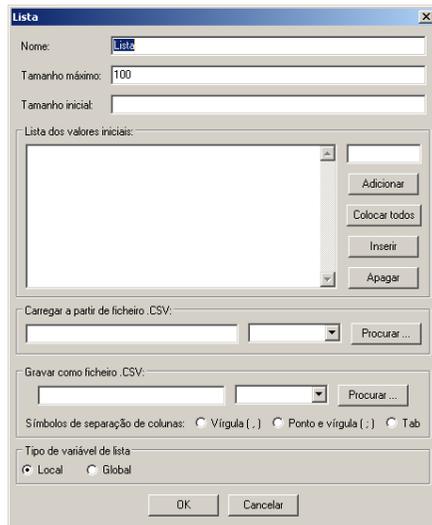
Pela entrada **I** (para índice) pode ser selecionado um determinado elemento da lista. Para isso, se envia à entrada **I** um comando = com o número do elemento desejado. O primeiro elemento tem o número 0. Ao elemento, que foi seleccionado através da entrada **I**, pode-se atribuir um novo número, enviando o novo valor desejado à entrada **S** com um comando =.

O elemento seleccionado através da entrada **I** pode ser consultado através da saída **R** (para localizar). Se se alterar a entrada **I** ou o valor do registro que foi seleccionado através da entrada **I**, a lista envia o valor actual do registro seleccionado aos elementos que estão conectados na saída **R**.

Pela saída **I** se consulta se no índice colocado na entrada **I** é válido. Se **N** for a quantidade de elementos, tem de ser colocado um valor na entrada **I** entre 0 e **N**-1. Se for este o caso, a saída **I** envia um comando = com o valor **N**, caso contrário com valor 0 a todos os elementos conectados.

Janela Propriedades para listas

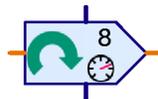
- Em **Tamanho máximo** pode introduzir a quantidade máxima de elementos da lista. Esta quantidade não pode ser ultrapassada por comandos **Anexos**.
- Em **Tamanho inicial** introduza a quantidade de elementos, com os quais a lista deve ser pré ocupada desde o início.
- Em **Lista de valores iniciais** pode introduzir os valores iniciais com os quais a lista tem de ser pré ocupada. Com os botões à direita da lista pode editar a lista.
- Em **Carregar ficheiro de .CSV** pode seleccionar um ficheiro .CSV Excel compatível, do qual a lista assume os seus valores. No campo de seleção no centro pode seleccionar a coluna do ficheiro .CSV, que deve ser utilizada. O ficheiro é carregado imediatamente e indicado na **Lista de valores iniciais**. Quando inicia o programa ou executa um download, o ROBO Pro procura carregar mais uma vez os valores atuais a partir do ficheiro. Se tal não der resultado, são utilizados os valores salvados na lista de valores iniciais.



- Em **Salvar em ficheiro .CSV** pode indicar um ficheiro no qual, o conteúdo da lista após o final do programa, deve ser salvo. Mas só funciona no modo Online- ou Online e apenas para listas estatísticas (ver ponto seguinte). O conteúdo da lista é escrito na coluna selecionada do ficheiro. Em **Símbolos de separação de colunas** pode seleccionar, se as colunas individuais na lista devem ser separadas por vírgulas ou pontos e vírgulas. Em países em que se escreva 0.5 com ponto, é utilizada uma vírgula como símbolo de separação de colunas. Como na Alemanha 0,5 se escreve com vírgula, é utilizado frequentemente um ponto e vírgula como símbolo de separação de colunas. Se tiver problemas na importação de um ficheiro CSV ROBO Pro, por exemplo, no Microsoft Excel, tente um outro símbolo de separação de coluna.
- Em **Tipo de variável de lista** pode configurar se a lista Temporizador é uma variável global ou local (ver capítulo 7.3.2 *Variável local* na página 504). Para grandes listas (tamanho máximo superior a 100 elementos) é recomendado o tipo **Global**, porque para variáveis globais está disponível mais memória do que para variáveis locais.

7.4 Comandos (Level 3)

Todos os elementos do programa neste grupo são elementos de comando. De acordo com a utilização, também podem ser classificados como elementos de mensagem. Se um elemento de comando é executado (ou seja, se o fluxo do programa entra na entrada azul do elemento, em cima), o elemento de comando envia um comando ou uma mensagem ao elemento, que está conectado à saída, à direita. Existem diferentes comandos tais como direita, esquerda ou parar, que têm diferentes efeitos sobre o elemento conectado. Em regra geral, os elementos conectados entendem apenas alguns elementos. No caso dos vários elementos do programa está listado, quais os comandos que eles entendem e qual o efeito dos comandos. A maior parte dos comandos são ainda acompanhados por um valor. No caso de um comando **Direita** se atribui ainda uma velocidade entre 1 e 8. Um comando **Parar** não tem, pelo contrário, qualquer valor adicional.

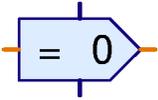


Janela Propriedades para elementos de comando

- Em **Comando** pode seleccionar o comando pretendido de uma lista de todos os comandos possíveis.
- Em **valor** pode introduzir um valor numérico, que deve ser transmitido com o comando. Caso não deva ser transmitido qualquer valor, este campo fica vazio.
- Em **Designação dos valores** pode introduzir uma breve indicação, que é indicado juntamente com o valor no elemento de comando (p.ex. X= ou T=). A indicação deve esclarecer o tipo de valor de que se trata. Isto serve apenas de comentário e não tem qualquer função.
- Em **Entrada de dados para valor de comando** pode introduzir, se o elemento de comando deve ter do lado esquerdo uma entrada de dados cor de laranja, para o valor a ser transmitido. O valor pode, em todos os elementos de comando, ser directamente introduzido no elemento de comando ou lido através de uma entrada de dados do lado esquerdo do elemento de comando. Desta forma, se pode, por exemplo, ativar um motor num circuito de regulação a uma velocidade que se pode alterar.



7.4.1 = (Atribuição)

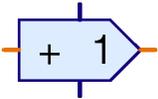


O comando = atribui um valor ao receptor. Normalmente, é utilizado para atribuir um valor às variáveis, variáveis de temporizador, elementos de listas ou saídas de painel de comando.

O comando = - não é, contudo, enviado apenas pelos elementos de comando, mas também por todos os elementos de programa com saídas de dados. Todos os elementos enviam comandos =, quando o valor de uma saída é alterado. Um elemento de entrada digital envia, por exemplo, um comando 1 = quando um botão na entrada é fechado, e um comando 0 =, quando o botão é aberto. Nesse caso, não é, contudo, usado qualquer elemento de comando. Nos elementos de programa com saídas de dados estão integrados elementos de comando =.

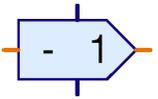
Todas as entradas de dados dos elementos de programa ROBO Pro são capazes de, pelo menos, processar o comando =. O comando = - é, por isso, o comando mais frequentemente usado no ROBO Pro.

7.4.2 + (Mais)



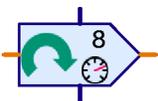
O comando + é enviado a variáveis ou variáveis do temporizador para aumentar o valor das variáveis. Com o comando + pode ser transmitido um valor à escolha que será acrescentado às variáveis. Uma vez que o valor transmitido com o comando também pode ser negativo, o valor da variável pode ser assim reduzido. Ver capítulo 7.3.1 *Variável, (global)* na página 503 e capítulo 7.3.4 *Variável do temporizador* na página 505.

7.4.3 - (Menos)



O comando - é utilizado de forma semelhante ao comando + descrito anteriormente. A única diferença, é que o valor transmitido é subtraído ao valor da variável.

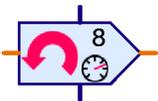
7.4.4 Direita



O comando **Direita** é enviado ao elemento de saída do motor para ligar o motor com sentido de rotação à direita. Ver capítulo 7.6.4 *Saída do motor* na página 516.

O valor é uma velocidade de 1 a 8.

7.4.5 Esquerda



O comando **Esquerda** é enviado ao elemento de saída do motor para ligar o motor com sentido de rotação à esquerda. Ver capítulo 7.6.4 *Saída do motor* na página 516.

O valor é uma velocidade de 1 a 8.

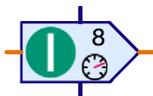
7.4.6 Parar



O comando **Parar** é enviado ao elemento de saída do motor para parar o motor. Ver capítulo 7.6.4 *Saída do motor* na página 516.

Com o comando **Parar** não é transmitido qualquer valor.

7.4.7 Ligar

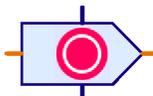


O comando **Ligar** é enviado ao elemento de saída da lâmpada para ligar a lâmpada. Ver capítulo 7.6.5 *Saída da lâmpada* na página 517. Um comando **Ligar** pode também ser enviado aos elementos de saída do motor e corresponde ao comando **Direita**. No caso de motores deveria ser antes utilizado o comando **Direita**, porque a o sentido da rotação pode se

directamente reconhecido.

O valor é a luminosidade ou intensidade de 1 a 8.

7.4.8 Desligar



O comando **Desligar** é enviado aos elementos de saída da lâmpada para desligar a lâmpada. Ver capítulo 7.6.5 *Saída da lâmpada* na página 517. Um comando **Desligar** pode também ser enviado aos elementos de saída do motor e corresponde ao comando **Parar**.

Com o comando **Desligar** não é transmitido qualquer valor.

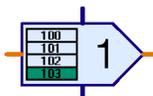
7.4.9 Texto



O comando **Texto** é um comando especial, pois não envia um comando com um número, mas sim com um texto à escolha ao elemento conectado. Mas existe apenas um elemento de programa que pode processar o comando de texto, nomeadamente uma indicação de texto num campo de comando. Para outras informações consulte no capítulo

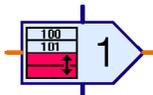
8.1.2 *Indicação de texto* na página 522.

7.4.10 Anexar valor



O comando **Anexar** é um comando especial para elementos da lista. Ver capítulo 7.3.5 *Lista* na página 506. Com o comando é transmitido um valor que é anexado ao fim da lista. Se a lista já estiver completamente cheia, o comando é ignorado.

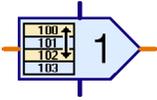
7.4.11 Deletar valor(es)



O comando **Deletar** é um comando especial para elementos da lista. Ver capítulo 7.3.5 *Lista* na página 506. Com o comando pode deletar a quantidade que quiser de elementos no fim de uma lista. A quantidade pretendida é transmitida como valor com o comando. Se o valor transmitido for superior à quantidade de elementos na lista, todos os elementos na lista

são apagados. Para apagar uma lista completa, pode enviar um comando **Delete** com o valor máximo possível de 32767 a um elemento da lista.

7.4.12 Trocar os valores

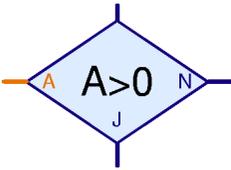


O comando **Trocar** é um comando especial para elementos da lista. Ver capítulo 7.3.5 *Lista* na página 506. Com o comando pode trocar um elemento à escolha de uma lista com o primeiro elemento da lista. O número do elemento com o qual troca o primeiro elemento, é transmitido como valor com o comando. **Importante:** o primeiro elemento de uma lista tem o número 0. Quando o valor transmitido não é um número de elemento válido, o comando é ignorado da lista de elementos.

7.5 Comparações, Aguardar por, ... (Level3)

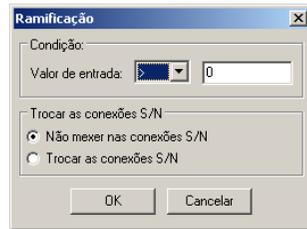
Os elementos do programa neste grupo servem todos para a ramificação do programa ou para o retardamento do processo do programa.

7.5.1 Ramificação (com entrada de dados)



Esta ramificação do programa possui uma entrada de dados cor de laranja **A** do lado esquerdo do elemento. Se lê um valor que vem frequentemente de um elemento de entrada (ver

capítulo 7.6.1 até 7.6.6 a partir da página 514). A entrada de dados **A** pode também ser conectada com as saídas de dados das variáveis, variáveis do temporizador ou operadores (ver capítulo 7.7 *Operadores* na página 519.) O valor à entrada de dados **A** é comparado pelo elemento com um valor fixo, mas de definição livre. Dependendo da comparação coincidir ou não, o elemento ramifica para a saída **S** ou **N**.

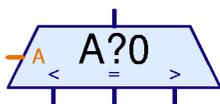


Janela Propriedades para a ramificação

- Em **Condição** pode introduzir um valor no campo direito com o qual o valor de entrada **A** deve ser comparado. Para efeito de comparação estão disponíveis todas as operações de comparação habituais.
- Se selecionar **Trocar as ligações S/N** são trocadas as saídas **S** e **N** logo que feche a janela de propriedades com OK. Para voltar a colocar as ligações S/N na sua posição de saída, pode voltar a trocá-las.

A comparação mais usada é **A > 0**. Isso significa, que o fluxo do programa para a saída **S** se ramifica, quando o valor introduzido na entrada de dados **A** é superior a 0. Desta forma, podem-se avaliar entradas digitais, que forneçam 1 ou 0. Mas também podem-se avaliar as variáveis de temporizador e muitos outros valores através da comparação **A > 0** de forma adequada.

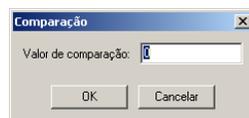
7.5.2 Comparação com constante



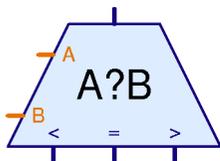
Com o elemento do programa **Comparação com constante** o valor na entrada de dados **A** pode ser comparado com um valor fixo mas livremente determinado. Dependendo do valor introduzido na entrada de dados **A** ser maior, menor ou igual ao valor fixo, este elemento de comparação ramifica para a saída direita, esquerda ou central. Na generalidade é conectada na entrada de dados **A** a saída de uma variável ou de uma lista. O elemento de comparação pode ser substituído através de dois elementos de ramificação. É no entanto compreensível quando apenas um elemento é necessário.

Janela Propriedades para comparação

- Em valor de comparação pode introduzir o valor constante com o qual o valor na entrada **A** será comparado.



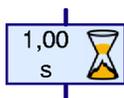
7.5.3 Comparação



Com o elemento do programa **Comparação** podem comparar-se os valores memorizados nas duas entradas de dados **A** e **B**, um com o outro. Conforme **A** é menor que **B**, **A** é maior que **B** ou **A** é igual a **B**, o elemento ramifica para a saída esquerda, direita ou central. A aplicação mais comum é assim a comparação de um valor nominal com um valor real. Conforme a relação entre o valor nominal e o valor real, um motor pode, por exemplo, ser rodado para a esquerda ou para a direita ou ser desligado.

O elemento do programa **Comparação** não possui possibilidades de configuração e, consequentemente, nenhuma janela Propriedades.

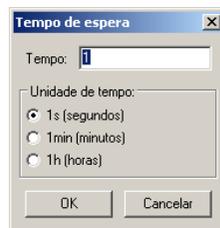
7.5.4 Tempo de espera



Com este elemento pode-se programar um **Tempo de espera** num processo. O tempo de espera começa quando é a vez do elemento em decurso. Logo que o tempo de espera indicado terminar, o processo continua. Veja também o capítulo 3.6.1 *Tempo de espera* na página 22.

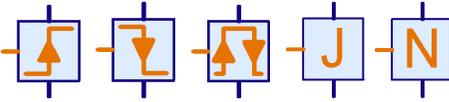
Janela Propriedades para Tempo de Espera:

- Em **Tempo** você pode introduzir o Tempo de Espera. Pode também utilizar números decimais como 1,23.
- Em **Unidade de Tempo** pode selecionar como unidade de tempo segundos, minutos ou horas. A unidade de tempo não tem qualquer influência na precisão do tempo de espera, ao contrário do que acontece com as variáveis do temporizador. Um tempo de espera de 60 segundos e um tempo de espera de 1 minuto têm um comportamento exatamente igual.



No modo de peritos (Level 5) é mostrada uma janela Propriedades ampliada semelhante à janela Propriedades para as variáveis do temporizador.

7.5.5 Aguardar por...

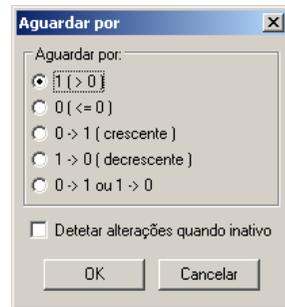


O elemento do programa **Aguardar por...** retarda a execução do programa até que na entrada de dados do elemento seja encontrada uma alteração ou até ocorrer um estado específico. O elemento existe

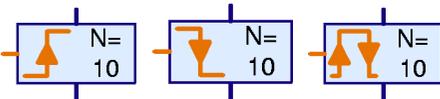
em 5 versões: O elemento esquerdo aguarda até que o valor na entrada aumente. Nesse caso não contam apenas as alterações de 0 para 1, como também qualquer aumento como, por exemplo, de 2 para 3. O segundo elemento aguarda até que o valor na entrada diminua e o elemento central aguarda até que aconteça uma qualquer alteração independentemente em que direção. O terceiro elemento é frequentemente utilizado para rodas dentadas movidas por impulsos. O quarto e o quinto elemento não aguardam uma alteração mas sim que, na entrada, o estado se encontre em Sim (>0) ou em Não (<=0). Quando o respectivo estado já se encontra, o elemento não aguarda. Pelo contrário, os três primeiros elementos aguardam sempre até que seja reconhecida uma alteração na entrada.

Janela Propriedades para Aguardar por alterações

- Em **Tipo de alteração** você pode comutar entre os cinco modos de funcionamento descritos anteriormente.
- Quando o botão **Reconhecer alterações quando inativo** se encontra pressionado, o elemento também deteta alterações que se efetuaram quando não estava na vez do elemento em decurso. Neste caso o elemento memoriza o último valor conhecido. Quando o elemento é novamente executado, a execução do programa continua imediatamente, quando entretanto o valor não tiver sido alterado de modo correto. Deste modo, a possibilidade de perder uma alteração porque o programa se encontra atualmente a executar uma outra tarefa, é mínima.



7.5.6 Contador de impulsos



Este elemento do programa aguarda uma quantidade configurada de impulsos na entrada de dados do lado esquerdo, antes de continuar com a execução do programa.

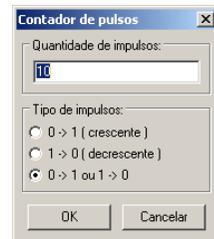
Isso é muito prático para uma tarefa de

posicionamento simples com rodas dentadas movidas por impulsos. Para posicionamentos mais trabalhosos, p. ex. com um valor variável, os sub programas têm de ser utilizados com variáveis.

Janela Propriedades para Contador de impulsos:

- Em **Quantidade de impulsos** você introduz a quantidade de impulsos que se deve aguardar, até a execução do programa continuar.
- Em **Tipo de impulsos** você pode mudar entre os três tipos de impulsos **0-1**, **1-0** ou alterar como preferir.

A possibilidade de reconhecer alterações quando o elemento se encontra inativo, como no elemento simples **Aguardar por...**, não



existe nesse elemento.

7.6 Interface de Entrada/ -Saída, ...

Esse grupo de elementos de programa contém todos os elementos de entrada e de saída. O modo como esses elementos são utilizados encontra-se no capítulo 5 *Level 3: Variáveis, painéis de comando & Co* na página 476.

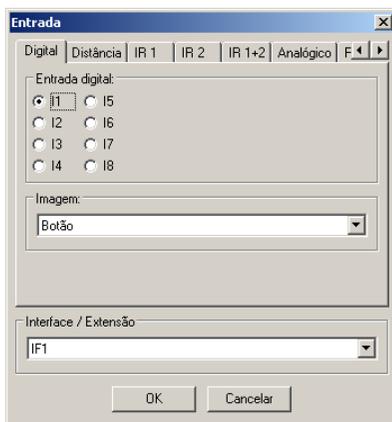
7.6.1 Entrada digital



Através do elemento **Entrada digital** pode consultar o valor das entradas digitais I1 a I8 da interface. Se os plugues pertencentes à entrada na interface estiverem ligando uma com a outra, o elemento de entrada digital fornece à sua ligação laranja um valor de 1, caso contrário um valor de 0.

Janela Propriedades para entradas digitais:

- Em **Entrada Digital** pode selecionar que entrada da interface deve ser utilizada. Você seleciona entradas dos módulos de ampliação em **Interface / Extensão**.
- Em **Imagem** você pode selecionar uma imagem do sensor conectado à entrada. Na maioria dos casos será uma **tecla**. Um **Contato Reed** é um interruptor que reage a campos magnéticos. Um **Fototransistor** também pode ser conectado a uma entrada digital apesar de ser, na verdade, um sensor analógico. Numa entrada digital conectada, você pode utilizar o Fototransistor em conjunto com uma lâmpada de formato lenticular como barreira fotoelétrica que se encontra interrompida (= 0) ou fechada (= 1). Quando, pelo contrário, conecta o Fototransistor a uma *Entrada analógica* pode reconhecer várias matizes de claro e escuro.
- Em **Interface / Extension** você pode selecionar se quer utilizar a entrada da Interface ou a entrada do módulo de ampliação ou de uma segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.



Analisando em pormenor todos os tipos de entradas, existe apenas um único elemento de programa. Através da barra em cima, na janela Propriedades, pode comutar o tipo de entrada a qualquer momento. Isso é muito prático sobretudo para comutar entre interruptor, entradas IR de campos de comando.

7.6.2 Entrada analógica



Através do elemento **Entrada analógica** pode consultar o valor das entradas analógicas. Ao contrário das entradas digitais que podem fornecer apenas o valor 0 ou 1, as entradas analógicas podem distinguir várias matizes. Todas as entradas analógicas fornecem um valor de saída entre 0 e 1023. Na interface ROBO existem, no entanto, vários tipos de entradas analógicas que medem

diferentes dimensões físicas. Existem entradas analógicas para medições de resistência, para medições de tensão e para um sensor especial para medição de distâncias:

Entrada	Tipo de entrada	Área de medição
A1, A2	Entradas de tensão	0-10,23V
AX, AY	Entradas de resistência	0-5,5k Ω
D1, D2	Entradas do sensor de distância	aprox. 0-50cm
AV	Tensão de alimentação	0-10V

Os restantes sensores fischertechnik resistência NTC, Fototransistor e resistência Foto transformam as dimensões de medição (temperatura e intensidade da luz) numa resistência. Por isso você pode conectar esses sensores a entradas **AX** ou **AY**. As entradas de tensão **A1** e **A2** foram elaboradas para todos os sensores que emitam uma tensão entre 0 e 10V.

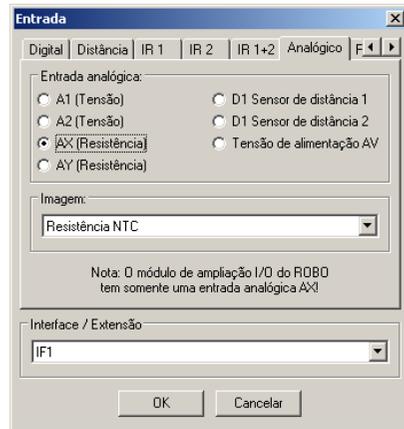
Não existe um plugue na interface ROBO para a entrada **AV**. Está sempre conectado com a tensão de alimentação da interface. Desse modo você pode supervisionar, p. ex., a tensão da bateria, e colocar o seu modelo na posição de saída antes da bateria ficar vazia.

Nas entradas dos sensores de distância **D1** e **D2** Podem ser conectados sensores especiais da fischertechnik capazes de medir a distância a, p. ex., a um obstáculo.

A Intelligent Interface tem apenas duas entradas analógicas, EX e EY. Essas correspondem às entradas AX e AY da interface ROBO. As restantes entradas analógicas não podem ser utilizadas com a Intelligent Interface!

Janela Propriedades para Entradas analógicas:

- Em **Entradas analógicas** você pode selecionar a entrada analógica desejada de acordo com a tabela em cima.
- Em **Imagem** você pode selecionar uma imagem do sensor conectado à entrada.
- Em **Interface / Extension** você pode selecionar se quer utilizar uma entrada da Interface, um módulo de ampliação ou a segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.



Na janela Propriedades para Entradas analógicas é novamente claro que o ROBO Pro utiliza apenas um elemento para todas as entradas e que, através da barra em cima, pode comutar entre todos os tipos de entradas. Para simplificar, já se encontram na janela do elemento, diferentes elementos de entrada à sua escolha.

7.6.3 Entrada IR:



Na ROBO Interface está montado um receptor de infra-vermelhos para o emissor manual da fischertechnik **IR Control Set** Art.-Nº. 30344. O emissor manual de infra-vermelhos não serve apenas para comandar à distância, mas é também muito prático como teclado para o seu modelo. Para o **IR Control Set** existem 2 receptores entre os quais se pode comutar com as teclas **1** e **2** no emissor manual. No ROBO Interface você pode, por isso, programar cada tecla para executar duas funções. Pode comutar entre estas duas ocupações com as teclas de comutação **1** e **2**. Em alternativa as teclas **1** e **2** pode ser utilizadas como teclas completamente normais.

No janelo Propriedades de uma entrada IR você pode comutar com a tecla TAB em cima entre **IR 1**, **IR 2** e **IR 1+2**. Quando tiver selecionado **IR 1**, o elemento de entrada IR fornece apenas um 1, quando a respectiva tecla no emissor está pressionada, e quando o emissor tiver sido regulado através da tecla **1** para a ocupação 1. Quando você selecionar **IR 2** o emissor tem de, pelo contrário, ser regulado através da tecla **2** para a ocupação 2.

No entanto se selecionar **IR 1+2** não importa para que posição o emissor manual está regulado. Depois você também pode utilizar as teclas 1))) e 2))) como entradas.

No elemento do programa é indicada a seleção através de um 1 ou 2 branco em baixo à direita no símbolo do emissor manual. Em **IR 1+2** não é indicado nenhum número no elemento do programa.



No elemento do programa é indicada a seleção através de um 1 ou 2 branco em baixo à direita no símbolo do emissor manual. Em **IR 1+2** não é indicado nenhum número no elemento do programa.

7.6.4 Saída do motor



Através do elemento **Saída do motor** pode-se ativar uma das 4 saídas bipolares do motor de um ROBO Interface ou Intelligent Interface. Uma saída do motor utiliza sempre duas ligações da Interface enquanto uma saída da lâmpada utiliza apenas uma saída. Mais diferenças entre as saídas de motor e as saídas de lâmpada podem ser descobertas nos parágrafos 7.1.6 *Saída do motor* na página 498 e 7.1.7 *Saída de lâmpada*.

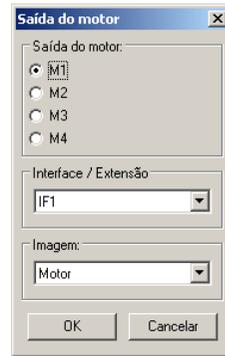
Numa saída do motor tem de ser enviado um comando, através do elemento de comando, para que a saída possa ser ligada. Um elemento do motor pode processar os seguintes comandos:

Comando	Valor	Ação
Direita	1 a 8	O motor gira para a direita com velocidade de 1 a 8
Esquerda	1 a 8	O motor gira para a esquerda com velocidade de 1 a 8
Parar	nenhum	O motor pára
Ligar	1 a 8	Como à direita
Desligar	nenhum	Como parado

=	-8 a 8	Valor -1 a -8: O motor gira para a esquerda Valor 1 a 8: O motor gira para a direita Valor 0: O motor pára
---	--------	--

Janela Propriedades para Elemento do motor:

- Em **Saída do motor** pode selecionar que ligações de saída da Interface devem ser utilizadas. Você seleciona saídas dos módulos de ampliação em **Interface / Extension**.
- Em **Interface / Extension** você pode selecionar se quer utilizar uma saída da Interface, um módulo de ampliação ou a segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** você pode selecionar uma imagem do utilizador conectado à saída. Na maioria dos casos será um **motor**. No entanto você também pode ligar à saída do motor um **eletroímã**, uma **válvula magnética** ou uma **lâmpada**.



7.6.5 Saída da lâmpada



Através do elemento **Saída da lâmpada** pode-se ativar uma das 8 saídas unipolares da lâmpada O1-O8 de uma ROBO Interface ou Intelligent Interface. Uma saída de lâmpada utiliza sempre apenas uma ligação de saída da Interface. A outra saída do utilizador é conectada com a tomada

da massa. Um desses utilizadores só pode ser ligado ou desligado, mas não se pode inverter a sua polaridade. Mais diferenças entre as saídas de motor e as saídas de lâmpada podem ser descobertas nos parágrafos 7.1.6 *Saída do motor* na página 498 e 7.1.7 *Saída de lâmpada*.

Numa saída da lâmpada tem de ser enviado um comando através do elemento de comando para que a saída possa ser ligada. Um elemento de lâmpada pode processar os seguintes comandos:

Comando	Valor	Ação
Ligar	1 a 8	A lâmpada é ligada com uma luminosidade de 1 a 8
Desligar	nenhum	A lâmpada é desligada
=	0 a 8	Valor 1 a 8: A lâmpada é ligada Valor 0: A lâmpada é desligada

Janela Propriedades para Elementos de saída da lâmpada:

- Em **Saída da lâmpada** pode selecionar que ligação de saída da Interface deve ser utilizada. Você seleciona saídas dos módulos de ampliação em **Interface / Extension**.
- Em **Interface / Extension** você pode selecionar se quer utilizar uma saída da Interface, um módulo de ampliação ou a segunda Interface. Você pode obter mais informações no capítulo 6 *Ativar módulos de ampliação e várias interfaces* na página 491.
- Em **Imagem** você pode selecionar uma imagem do utilizador conectado à saída. Na maioria dos casos será uma **lâmpada**. No entanto, você também pode ligar à saída da lâmpada um **eletroimã**, uma **válvula magnética** ou mesmo um **motor**. Um motor ligado a uma saída da lâmpada só pode, no entanto, girar numa direção.



7.6.6 Entrada do painel de comando



ROBO Pro oferece a possibilidade de desenhar os seus próprios painéis de comando para os seus modelos. Você pode obter mais informações no capítulo 8 *Vista geral dos elementos de comando e painéis de comando* na página 521. Desse modo você pode ativar o seu modelo confortavelmente

a partir do computador. No painel de comando existem botões, controles deslizantes e elementos de registro à sua disposição. O estado desse elemento pode ser consultado no programa através do elemento **Painel de comando**. Os botões fornecem um valor de 0 ou 1. Os controles deslizantes fornecem um valor numa área configurada (por norma de 0 a 100).

Os painéis de comando só podem ser utilizados no modo Online. Você pode obter mais informações no capítulo 3.7 *Modo Online ou Download – Qual é a diferença?* na página 464.

Janela Propriedades para Entradas de painéis de comando:

Há um painel de comando para cada programa principal ou subprograma. Por baixo do nome dos programas estão listados os elementos de comando. Se ainda não tiver introduzido qualquer elemento de comando, também não aparecem quaisquer elementos na lista. Tem, por isso, de desenhar primeiro o painel de comando, antes de poder conectar uma entrada de painel de comando com um elemento de comando.

A selecção em **Interface / Extension** é ignorada nas entradas dos painéis de comando, pois neste caso não se tratam de entradas reais num módulo da interface.



7.6.7 Saída do painel de comando



Painel d
Indicação

ROBO Pro oferece a possibilidade de desenhar os seus próprios painéis de comando para os seus modelos. Você pode obter mais informações no capítulo8 *Vista geral dos elementos de comando e painéis de comando* na página 521. Entre os botões e outros elementos de registro para comando do seu modelo, você pode também introduzir elementos de indicação num painel de comando. Nos elementos de indicação você pode por exemplo indicar as coordenadas do eixo de um robô ou o estado de um interruptor fim de curso. Você altera o valor indicado, no qual você introduz um elemento **Saída do painel de comando** no programa e envia ao elemento um comando =, por ex., no qual você conecta uma variável, uma entrada analógica ou um elemento de comando.

Os painéis de comando podem ser utilizados somente em modo Online; voc pode obter mais informações sobre esse assunto no capítulo 3.7 *Modo Online ou Download – Qual é a diferença?* na página 464.

Janela Propriedades para indicações do painel de comando:

Há um painel de comando para cada programa principal ou subprograma. Por baixo do nome dos programas estão listadas as indicações do painel de comando. Se ainda não tiver introduzido qualquer elemento de comando, também não aparecem quaisquer elementos na lista. Tem, por isso, de desenhar primeiro o painel de comando, antes de poder conectar uma entrada de painel de comando com um elemento de comando.



7.7 Operadores

Os elementos de programa deste grupo são os chamados operadores. Os operadores possuem uma ou várias entradas de dados côm-de-laranja. Os valores nas entradas de dados são contactados pelo operador a um valor e entregues na saída do operador através do comando =.

Janela Propriedades para operadores

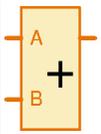
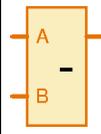
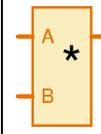
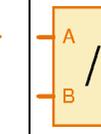
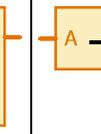
Todos os operadores usam a mesma janela Propriedades. Através da janela Propriedades, pode mudar de um operador para outro.

- Em **Operação** pode configurar de que forma o operador pode conectar as suas entradas. As funções individuais estão descritas nos dois parágrafos seguintes.
- Em **Quantidade de Entradas** você pode ajustar o número de entradas que o operador deve ter.



7.7.1 Operadores aritméticos

O ROBO Pro coloca-lhe à disposição as quatro operações básicas de cálculo. Os símbolos entre as duas entradas têm a seguinte aparência:

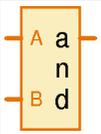
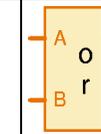
				
Mais	Menos	Multiplicar	Dividir	Menos
$A + B$	$A - B$	$A * B$	A / B	$- A$

Quando o operador **Menos** tem mais de duas entradas, todos os outros valores de entrada são subtraídos ao valor da entrada A. Quando o operador Menos tem apenas uma entrada, o operador inverte o sinal do valor de entrada.

Quando o operador **Menos** tem mais de duas entradas, o valor na entrada A é dividido por todos os outros valores de entrada.

7.7.2 Operadores lógicos

Por exemplo, para a conexão de entradas digitais, existem três operadores lógicos no ROBO Pro:

		
E	Ou	Não
$A > 0$ e $B > 0$	$A > 0$ ou $B > 0$	$A \leq 0$

Os operadores lógicos interpretam um valor superior a 0 na entrada como **sim** ou **verdadeiro** e um valor inferior ou igual a 0 como **não** ou **falso**. Entradas digitais fornecem um valor de 0 ou 1, de forma a que 0 seja interpretado como **falso** e 1 como **verdadeiro**.

O operador **E** envia aos elementos conectados à saída um comando = com valor 1, quando está introduzido em todas as entradas o valor verdadeiro, ou seja, um valor > 0 . Caso contrário, o elemento envia um comando = com valor 0.

O operador **Ou** envia aos elementos conectados à saída um comando = com valor 1, quando está introduzido em pelo menos uma das entradas o valor verdadeiro, ou seja, um valor > 0 . Caso contrário, o elemento envia um comando = com valor 0.

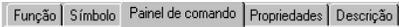
O operador **Não** envia aos elementos conectados à saída um comando = com valor 1, quando está introduzido na entrada o valor falso, ou seja, um valor ≤ 0 . Caso contrário, o elemento envia um comando = com valor 0.

A função dos operadores lógicos também pode ser executada com vários elementos de ramificação. Mas é normalmente mais simples conectar várias entradas com operadores.

8 Vista geral dos elementos de comando e painéis de comando

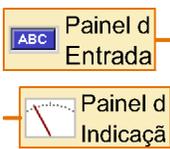
No ROBO Pro podem ser definidos os próprios painéis de comando. Através dos painéis de comando pode controlar modelos complexos. Uma vez que o painel de comando é indicado na tela do PC, os painéis de comando funcionam apenas no modo Online. Voc pode obter mais informações sobre este assunto no capítulo 3.7 *Modo Online ou Download – Qual é a diferença?* na página 464.

Para criar um painel de comando, seleccione **Painel de Comando** na barra de funções:



Pode depois introduzir os elementos de comando no campo cinzento vazio, por baixo. Um painel de comando pertence sempre ao programa principal e ao subprograma, em que você se encontrava durante a criação do painel de comando. Por isso é importante, que seleccione o subprograma correto na barra de subprograma, antes de colocar um painel de controlo. Normalmente se coloca o painel de comando por baixo do **programa principal**.

Nos painéis de comando há indicações e elementos de comando. Através das indicações pode-se, por exemplo, indicar valores de variáveis ou mensagens de texto. Pelo contrário, os elementos de comando funcionam como botões adicionais.



A cada elemento de comando, que você introduz no painel de comando, pertence, no programa, um elemento **Entrada de painel de comando** (para elementos de comando) ou **Saída de painel de comando** (para indicações). Por cima destes elementos de comando, você estabelece a conexão entre o seu programa e o seu painel de comando. Pode encontrá-la no grupo de elementos **Entradas, Saídas**. Dependendo de qual o tipo de elemento de comando que você conecta a este elemento de comando,

é indicado um outro símbolo. Na lista de elementos existem apenas dois elementos: um para indicação e outro para elemento de comando.

8.1 Indicações

As indicações são usadas de forma semelhante às saídas da interface. Pode colocar o valor de uma indicação com um comando =.

8.1.1 Aparelho de medição

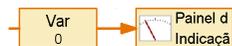


O **aparelho de medição** está afinado de acordo com um instrumento de indicação analógico. É normalmente usado para indicar o valor das entradas analógicas, pode também utilizá-lo para variáveis ou outros elementos de programa.

O aparelho de medição é comandado através de uma saída do painel de comando. Pode encontrar a **saída do painel de comando** no grupo de elementos **Entradas, Saídas**.



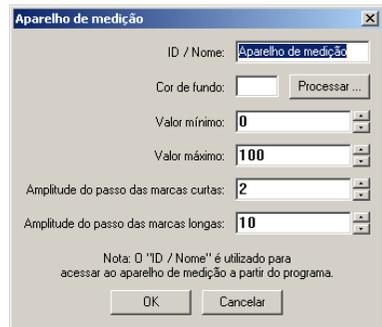
Pode colocar o valor do aparelho de medição, enviando à saída do painel de comando correspondente no programa o comando =.



Quase todos os elementos do programa com saídas de elementos enviam um comando =, quando o seu valor se altera. Entradas analógicas ou variáveis podem, por exemplo, ser conectadas diretamente com a saída do painel de comando.

Janela Propriedades para aparelhos de medição

- EM **ID/Nome** você deve, em primeiro lugar, introduzir um nome para o aparelho de medição. O nome é importante para que você possa diferenciar vários aparelhos de medição no programa.
- Em **cor de fundo** você pode configurar outra cor que não o branco.
- Em **valor mínimo** e **valor máximo** indique os valores, que correspondem à posição do indicador no lado esquerdo e direito da escala. Se um dos valores for menor do que 0 e o outro for maior do que 0, é mostrado um 0 alongado.
- A escala é composta por traços curtos e compridos. A distância dos traços curtos e compridos é introduzida em **distância de marcas curtas / longas**. Quando ambos possuem o mesmo valor, só são visíveis marcas compridas.



8.1.2 Indicação de texto



Numa indicação de texto, você pode indicar valores numéricos, texto ou uma mistura de ambos.

A indicação de texto é comandada pelo programa através da saída do painel de comando. Pode encontrar a **saída do painel de comando** no grupo de elementos **Entradas, Saídas**.

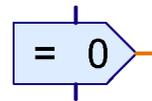


Logo que tenha conectado a saída do painel de comando com uma indicação de texto através da sua janela Propriedades, se altera o símbolo e aparece o nome do painel de comando (por ex. Principal) e a indicação (por ex. Texto).



Pode colocar o texto na indicação de dois modos:

- Você pode colocar o conteúdo da indicação, no qual você envia à respectiva saída do painel de comando um comando =. Isto é muito prático se você quiser utilizar a indicação para indicar o valor de uma variável ou outros elementos do programa, porque a maioria dos elementos do programa enviam automaticamente =-comandos através das suas saídas de dados quando o valor se altera. O comando = substitui apenas os últimos 6 caracteres da indicação. O resto da indicação pode ser preenchido com um texto predefinido. Desta forma, pode colocar um texto de referência relativamente ao valor na visualização. Se a indicação tiver várias linhas, pode também colocar o texto de indicação numa linha própria. No caso de indicações de várias linhas são copiados apenas os últimos 6 caracteres da última linha pelo comando =.
- Com o comando de texto pode colocar arbitrariamente o conteúdo da indicação. O texto comando é um elemento de comando especial, que pode enviar não só um número, como também um texto inteiro através da sua saída. Tal como um elemento habitual de comando, este elemento **Texto** pode também ter uma entrada de dados. Pode integrar no texto, o valor numérico que existe à entrada de dados. Se enviar



a um elemento de indicação vários comandos **Texto**, os textos serão anexados uns aos outros. Deste modo, pode misturar os números e o texto como quiser.

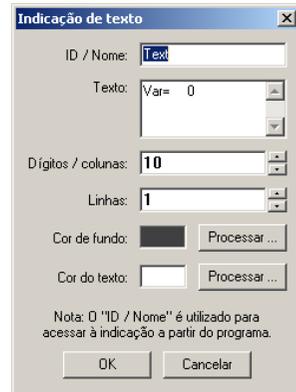
Sinais de comando no comando de texto

No elemento de comando **texto** pode usar os seguintes sinais, para obter efeitos especiais:

Sinal de comando	Efeito
#####	Entrega o valor na entrada de dados como número de 5 dígitos + sinal
##.##	Entrega o valor na entrada de dados como número com 2 casas decimais à direita da vírgula, que são separadas por um ponto .
##,##	Entrega o valor na entrada de dados como número com 2 casas decimais à direita da vírgula, que são separadas por uma vírgula .
\c	Apagar indicação e colocar ponto de introdução no início da indicação

Janela Propriedades para indicações de texto

- Em **ID/Nome** você deve em primeiro lugar introduzir um nome para indicação. O nome é importante para que você possa diferenciar várias indicações no programa.
- Em **Texto** introduzir o conteúdo da indicação. Este conteúdo se mantém até enviar um comando à indicação a partir do programa. Se enviar um comando = à indicação, são apenas copiados os últimos 6 caracteres do conteúdo da indicação. O início do texto se mantém, de forma que ainda possa fazer uma indicação para um número, para saber de que número se trata. No exemplo ilustrado se mantém o texto **Var=**. A indicação tem 10 caracteres, ou seja, se mantém 10-6=4 caracteres.
- Em **Dígitos/Colunas** e em **Linhas** pode introduzir o espaço para caracteres que a indicação deve ter. Numa indicação de várias linhas pode indicar uma nota como **Var=** ou **Visitantes** numa linha própria.
- Em **Cor de fundo** e **Cor do texto** pode alterar o design da cor das indicações. Clique em **Processar ...** para selecionar uma cor ou para definir uma cor própria.



8.1.3 Lâmpada de sinalização



A **lâmpada de sinalização** é o modo mais fácil de indicação. Funciona de forma semelhante ao de um componente de lâmpada fischertechnik, que está ligado a uma saída da interface.

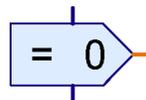
A lâmpada de sinalização é comandada pelo programa através da saída do painel de comando. Pode encontrar a **saída do painel de comando** no grupo de elementos **Entradas, Saídas**.



Logo que tenha conectado a saída do painel de comando com uma lâmpada de sinalização através da sua janela Propriedades, se altera o símbolo e aparece o nome do painel de comando (por ex. Principal) e da lâmpada.

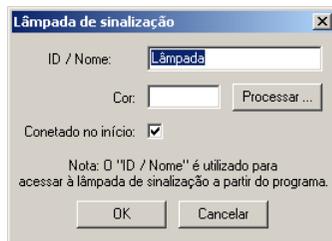


Pode ligar ou desligar a lâmpada, enviando um comando **Ligar** ou **Desligar** à saída do painel de comando correspondente no seu programa, tal como utiliza também para as saídas de lâmpada corretas. Também pode ligar ou desligar a lâmpada de sinalização através de um comando =. Se o valor for superior a 0, a lâmpada é ligada. Se o valor for inferior ou igual a 0, a lâmpada é desligada.



Janela Propriedades para lâmpada de sinalização

- Em **ID/Nome** você deve em primeiro lugar introduzir um nome para a lâmpada de sinalização. O nome é importante para que você possa diferenciar várias lâmpadas de sinalização no programa.
- Em **Cor** você pode alterar a cor da lâmpada de sinalização. Para isso, faça clique no botão **Processar**.
- Se **Ligado no início** estiver com uma cruz, a lâmpada de sinalização está ligada, até que o elemento respectivo do programa receba o primeiro comando. Caso contrário, a lâmpada de sinalização estará desligada no início.



8.2 Elementos de comando

Os elementos de comando são usados de forma semelhante às entradas da interface.

8.2.1 Botão

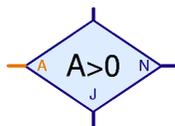


O **botão de elemento de comando** pode ser utilizado como um botão fischertechnik ou interruptor, que está conectado a uma das entradas da interface.

O botão é consultado a partir do programa através de uma **entrada do painel de comando**. Pode encontrar a **entrada do painel de comando** no grupo de elementos **Entradas, Saídas**.

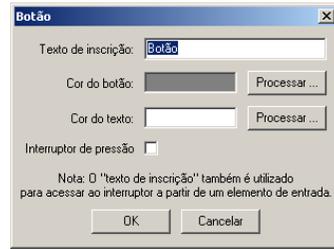


A entrada do painel de comando, pertencente ao botão, pode ser conectada como uma entrada digital da interface a todos os elementos do programa com uma entrada de dados, por exemplo, na **ramificação**. Quando o botão está pressionado, é fornecido como valor 1, caso contrário, como valor 0.



Janela Propriedades para botões

- No **Texto de inscrição** você pode inserir a inscrição para o botão. Este é ao mesmo tempo o nome com o qual se acessa ao botão a partir do programa. No botão não existe um nome adicional/campo ID como nos outros elementos de comando.
- Você pode alterar o design da cor do botão em **Cor do botão** e **Cor do texto**. Para isso, faça clique em **Processar....**
- Se aparecer um visto no interruptor de pressão, o botão não funciona como um botão, mas sim como um interruptor. No primeiro clique no botão, o botão é pressionado para dentro e assim permanece até ao segundo clique. Caso contrário, o botão funciona como botão e salta imediatamente quando é solto.



8.2.2 Regulador

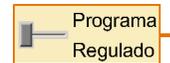


Podemos utilizar o regulador como um potenciômetro, que está conectado a uma entrada analógica da interface. Ao contrário do botão, o regulador pode fornecer não apenas os valores 0 e 1 mas muitos outros valores também, tal como uma entrada analógica. A área de valores pode ser configurada através da janela Propriedades. O regulador pode ser, por exemplo, utilizado para ajustar as rotações do motor entre 1 e 8.

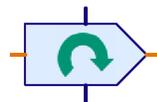
O regulador é consultado a partir do programa através de uma **entrada do painel de comando**. Pode encontrar a **entrada do painel de comando** no grupo de elementos **Entradas, Saídas**.



Logo que tenha conectado a entrada do painel de comando com um regulador, através da sua janela Propriedades, se altera o símbolo e aparece o nome do painel de comando (por ex. Principal) e do regulador.

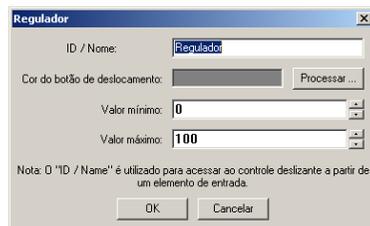


A entrada do painel de comando pertencente ao regulador pode ser conectada a todos os elementos do programa com uma entrada de dados, tal como uma entrada analógica da interface. Muito frequentemente, o regulador é conectado a um elemento de comando com entrada de dados, de forma a que o regulador comande a velocidade de um motor.



Janela Propriedades para regulador

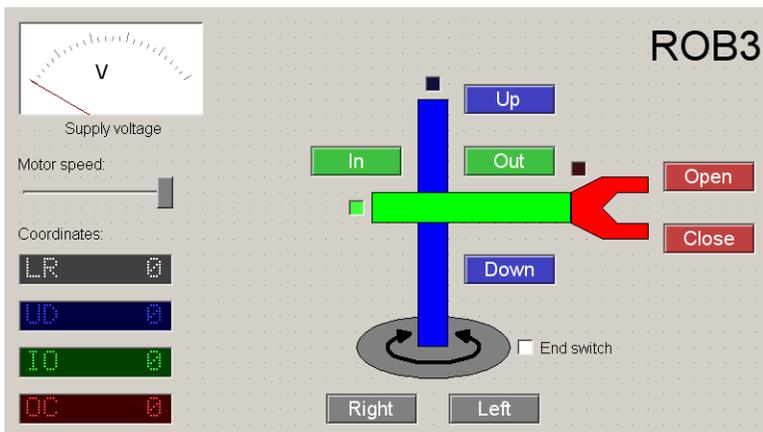
- EM **ID/Nome** você deve em primeiro lugar introduzir um nome para o regulador. O nome é importante para que você possa diferenciar vários reguladores no programa.
- Você pode alterar a cor do botão de deslocamento em **Cor do botão de deslocamento**. Para isso, faça clique em **Processar**.
- Em **Valor mínimo** e **Valor máximo** você insere a área de valores para o regulador. Se você quiser utilizar o regulador para controlar uma velocidade do motor, a área de valores deve estar entre 1 até 8.



9 Funções de desenho

O ROBO Pro dispõe de funções de desenho habituais. Encontra as funções de desenho na janela de grupos de elementos em **Desenhar**. No subgrupo **Formas** pode encontrar ferramentas de desenho para diferentes formas geométricas básicas. No subgrupo **Texto** pode encontrar ferramentas de caracteres para diferentes tamanhos destes. Os outros subgrupos contêm funções para alteração da cor e da espessura de linha.

Através das funções de desenho, pode ilustrar os seus painéis de comando e programas, para destacar a função. Aqui está representado um exemplo de painel de comando desenhado para um robô:



Os botões, indicações de coordenadas e lâmpadas de interruptor de fim de curso são mantidas nas respectivas cores, tal como os eixos individuais no desenho esquemático do robô. Dessa forma se cria um painel de comando muito simples.

A utilização das funções de desenho não deve levantar grandes dificuldades. Por isso, se apresentam a seguir, alguns pontos que possam não estar ainda bem clarificados:

- Objetos gráficos como retângulos e círculos **não** são chamados se pressionar o botão do mouse como acontece em muitos outros programas, terá de clicar duas vezes no mouse, uma vez no canto superior esquerdo e outra vez no canto inferior direito.
- Textos não são trabalhados numa janela de diálogo, mas dim diretamente no ambiente de trabalho. Se acrescentar um novo objeto de texto, surge apenas um quadro azul claro. Pode agora escrever muito facilmente no teclado e o texto escrito surge então, diretamente no ambiente de trabalho. Pode também acrescentar texto da área de transferências com CTRL+V.
- Após ter desenhado um objeto, pode trabalhá-lo deslocando os pequenos pontos de apoio azuis. Existem também pontos de apoio para rodar e distorcer objetos. Um retângulo tem em cima à esquerda dois pontos de apoio. Se deslocar o segundo maior ponto de apoio, pode arredondar os cantos do retângulo. Pode terminar o modo de processamento, se carregar o botão direito do mouse ou se pressionar a tecla **ESC**.
- Se quiser trabalhar à posteriori um objeto, selecione no menu **Desenhar** a função **Processar**. Se clicar num objeto, surge novamente os pontos de apoio azuis claros.

- Muitos objetos têm um ou mais modos de processamento e de desenhar. Pode alternar entre cada um dos modos com a tecla de tabulação no teclado enquanto desenha ou processa um objeto. No caso de um círculo pode selecionar por exemplo, se pretende introduzir dois pontos de canto ou o ponto central e um ponto de canto. No caso de polígonos pode alternar entre o processamento de pontos e funções como “Rodar”. Em objetos de texto pode mudar entre processamento do texto, bem como alteração do tamanho do texto e do ângulo de rotação.
- No menu **Desenhar** existem as funções Objeto **em primeiro plano / em segundo plano**. Com estas funções pode deslocar todos os objetos selecionados (a vermelho) para a frente ou para trás, de forma a cobrir outros objetos ou a serem cobertos por outros.
- Com a função **Captura de imagem** no menu **Desenhar** pode ligar ou desligar a captura de imagem. Deve ter em atenção que a captura está ligada quando está processando o seu programa, uma vez que todos os elementos do programa estão adaptados à captura.
- Em objetos de texto pode alterar a orientação do texto, pressionando a tecla CTRL + uma tecla de 1-9 no bloco numérico. Pode-se fazê-lo se a lâmpada de bloqueio numérico no teclado estiver acesa. Caso contrário, terá de pressionar primeiro a tecla numérica.