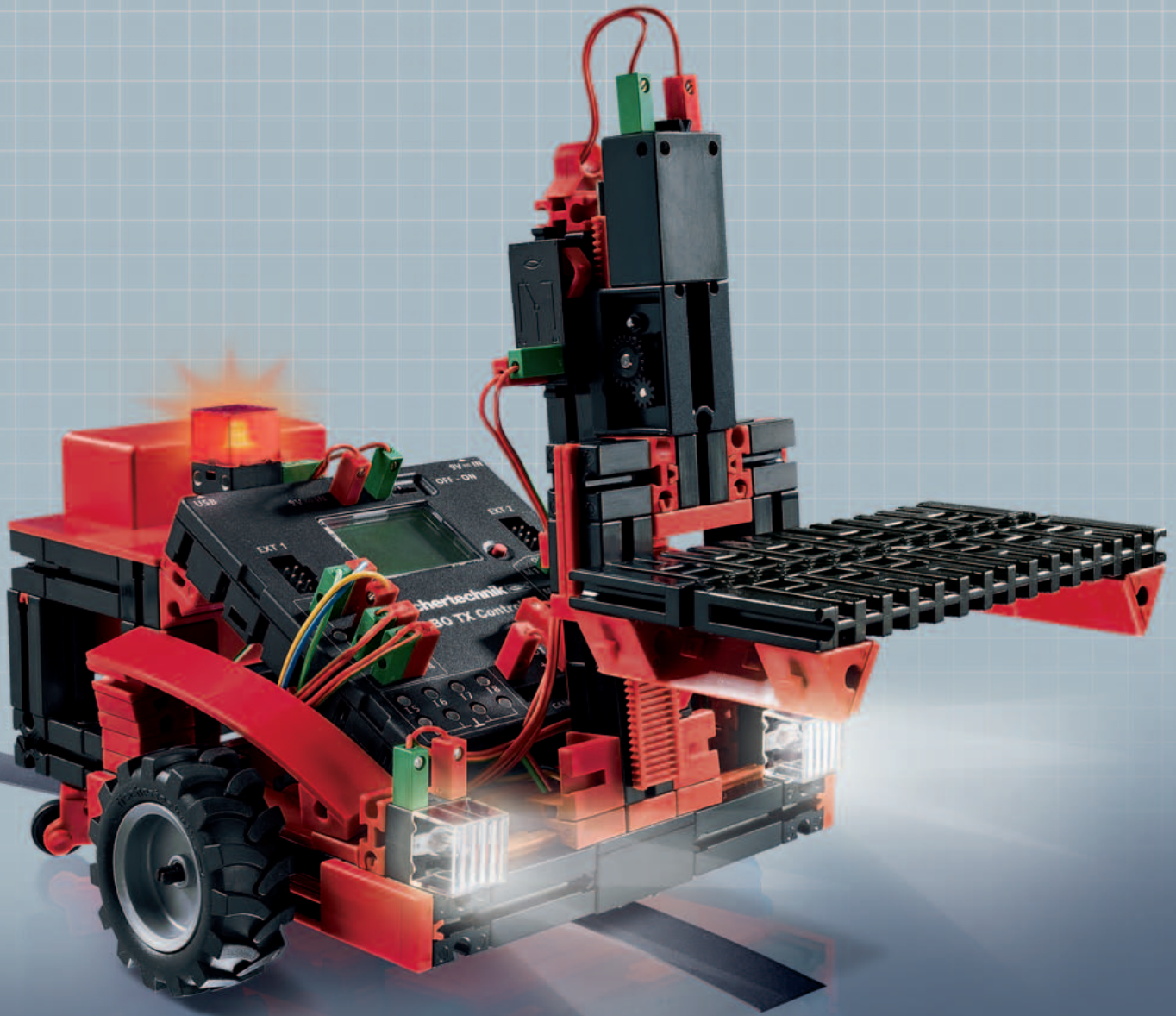


fischertechnik 

COMPUTING

Folheto



ROBO TX Training Lab

11 MODELS
MODEL 2

Conteúdo

Bem-vindo ao ROBO TX Training Lab	3
Algumas informações gerais	3
Eletricidade	3
Sobre este folheto	3
Robôs, pessoas artificiais?	4
Computing, (quase) tudo automático	4
Explicação sobre os componentes	4
Motores do codificador	5
Motor XS	5
Lâmpada incandescente	5
Lâmpada de lente	5
Fototransistor	5
Sensor de rastreamento	6
Botão de pressão	6
Sensor térmico (CNT)	6
Algumas dicas	7
Primeiro passo	7
Modelos para iniciantes	8
O secador de mãos	8
O semáforo	9
O elevador	10
A máquina lava-louça	10
Regulagem da temperatura	11
Robôs - o próximo desafio	12
Modelo básico	12
O detector de pistas	13
O corta-grama	14
O robô jogador de futebol	15
O robô medidor	16
A empilhadeira de forquilha	17
Se não der certo no começo...	19
E como prosseguir?	20

Bem-vindo ao ROBO TX Training Lab

Olá!

Estamos contentes com a sua decisão de comprar o módulo ROBO TX Training Lab da fischertechnik. E prometemos que o seu interesse será recompensado. Porque com esse módulo, você pode realizar uma vasta quantidade de experimentos e tarefas interessantes.

Ao ler esse manual de tela e testar os experimentos e tarefas, você aprenderá, passo a passo, como programar e controlar máquinas e robôs simples e complexos com o ROBO TX Controller da fischertechnik.

Como sempre ocorre durante o aprendizado, não é possível começar pelas tarefas mais complexas, mesmo que sejam obviamente mais interessantes que as mais simples. Por isso organizamos os experimentos e tarefas neste manual de modo que você aprenda algo novo com cada tarefa para poder utilizar na tarefa seguinte.

Portanto não tenha medo, começaremos de modo simples e nos prepararemos juntos para os grandes robôs.

Agora, desejamo-lhes boa sorte e sucesso nos experimentos com o ROBO TX Training Lab.

A sua equipe da

fischertechnik



Algumas informações gerais

Antes de lidarmos com o módulo fluentemente, você precisa saber de algumas coisas. Os componentes com os quais trabalharemos apesar de serem muito robustos, se não forem manuseados corretamente, podem sofrer avarias, sob certas circunstâncias.

Eletricidade

Como você certamente sabe, vários componentes do ROBO TX Training Lab funcionam com eletricidade. E com coisas que têm a ver com eletricidade, deve-se prestar bastante atenção para não cometer erros. Consulte o manual de montagem com muito cuidado quando se tratar de cabeamento de componentes elétricos.

Sob nenhuma hipótese você poderá conectar o pólo negativo com o positivo, o que resultaria em curto-circuito. Nesse caso o ROBO TX Controller e a bateria podem sofrer danos.

Eletricidade e eletrônica são temas tão interessantes quanto robótica (portanto o assunto de que trata este módulo), e existe um módulo da fischertechnik que se ocupa especialmente desse tema. Então se você se interessar por isso, você se divertirá com o módulo PROFI E-Tech tanto quanto nós nos divertimos com o ROBO TX Training Lab.



Sobre este folheto

Esse folheto em PDF tem algumas funções que não existem em um folheto impresso e que você talvez já conheça da Internet.

Links internos do suplemento

Se em algum lugar no texto for mencionado algo que estiver melhor explicado em outro local do suplemento (p.ex. os componentes), o texto aparecerá em azul escuro e sublinhado. Você pode clicar no texto e automaticamente será levado à página onde a explicação se encontra. Isso se chama "referência cruzada".

Informação de fundo

Em parte neste suplemento existem termos ou palavras estrangeiras cuja explicação pode ser útil. Esses termos estão escritos em verde e sublinhados. Se você passar sobre o texto o ponteiro do mouse, aparecerá um campo com uma explicação.

Links externos ao suplemento

Para certos links você precisa de uma conexão à internet (por exemplo, para o site da fischertechnik) ou um ROBO Pro instalado (para consultar a ajuda online da ROBO Pro). Esses links aparecem em azul claro e sublinhados.

Imagens

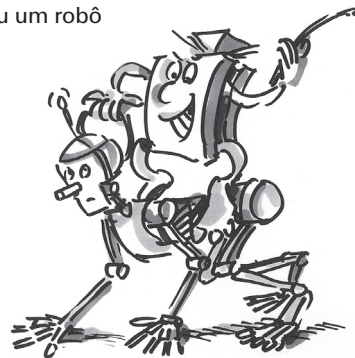
Uma imagem diz mais que mil palavras. Você certamente já ouviu essa frase várias vezes. E como ela tem muito de verdade, você pode, encostando nas palavras marrons e sublinhadas, adicionar uma imagem na qual você pode reconhecer o que o texto quer dizer.

Robôs, pessoas artificiais?

O que vem primeiro à sua cabeça quando você escuta a palavra "robô"? Você já viu um robô alguma vez? No cinema ou na televisão? Ou quem sabe em pessoa?

Existem inúmeros tipos diferentes de robôs. Alguns se parecem com um ser humano, outros consistem meramente de um ou mais braços. O que exatamente faz de um robô um robô?

No dicionário, temos: "Robôs são máquinas estacionárias ou móveis que realizam tarefas determinadas conforme um certo programa".



Computing, (quase) tudo automático

Robôs são, portanto, máquinas controladas por um programa. E esse controle de máquinas (ou, em nosso caso, modelos) é chamado de "computing".

Com o "ROBO TX Training Lab" você pode adentrar esse tema de modo espetacular. Porque o módulo contém tudo o que você precisa para construir e controlar diversas máquinas.

Os programas para controlar os modelos podem ser instalados no PC com ajuda do software ROBO Pro e em seguida ser transmitidos ao ROBO TX Controller por conexão USB ou Bluetooth. O Controller "controla" e guia o modelo de acordo com a programação que você criou.

Explicação sobre os componentes

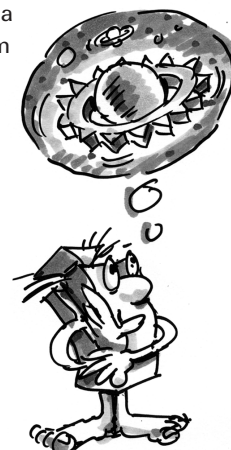
Estão inclusos no módulo

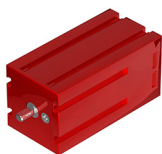
Primeiramente, você encontrará um grande número de módulos da fischertechnik, tais como: motor, lâmpadas e sensores, bem como um manual de montagem colorido para a construção de diversos modelos.

Quando você tiver desembalado todos os módulos, deverá, primeiramente, montar alguns componentes, antes que possa iniciar (p.ex. cabo e plugue). Quais são estes exatamente, está descrito no manual de montagem, nas "Dicas de montagem". Cuide disso logo no início.

Atuadores

São designados atuadores todos os componentes que podem realizar uma ação. Isso significa que se conectamos tais componentes a uma corrente elétrica, de algum modo eles se tornam "ativos". Geralmente isso pode ser visto diretamente. Um motor roda, uma lâmpada acende, etc.





Motores do codificador

Como acionamento para nossos robôs, usamos os dois motores de codificador localizados no módulo. À primeira vista, eles são motores elétricos normais, feitos para uma voltagem de 9 volts e uma amperagem de, no máximo, 0,5 amperes.

Porém, os motores do codificador podem fazer muito mais coisas: Além da conexão para a alimentação elétrica do motor, eles têm ainda uma entrada para um cabo tri-polar, com o qual pode-se avaliar o movimento giratório do motor com a ajuda do assim-chamado codificador.

O codificador funciona similarmente ao tacômetro de uma bicicleta. Um ímã (na bicicleta, ele fica geralmente em um dos aros) a cada rotação se aproxima de um sensor (na bicicleta, geralmente no garfo da roda), o que faz com que o sensor gere um impulso. Esses impulsos podem ser contados e, por exemplo, com o tacômetro, multiplicados pelo perímetro do pneu. Assim podemos obter a distância percorrida.

O codificador nos motores de codificadores da fischertechnik geram 3 impulsos por rotação do cardã do motor. E já que os motores de codificador têm adicionalmente uma engrenagem com uma [relação de transmissão de 25:1](#) (diz-se "25 por 1"), uma rotação do cardã vinda da engrenagem corresponde a 75 impulsos do codificador.

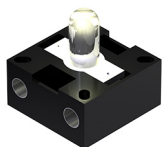


Motor XS

O motor XS é um motor elétrico com o mesmo comprimento e a mesma altura que um elemento da fischertechnik. Além disso, ele é muito leve. Assim você pode instalá-lo em locais onde não há espaço para motores grandes.

Ambas as engrenagens, que também estão no módulo, cabem exatamente no motor XS.

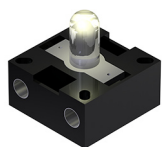
O motor XS foi projetado para uma voltagem de 9 volts e um consumo elétrico máximo de 0,3 amperes.



Lâmpada incandescente

O módulo contém duas lâmpadas incandescentes. Elas podem ser usadas de diversos modos. Por exemplo, como sinalizadores em um semáforo ou como luzes piscantes em um robô.

As lâmpadas incandescentes devem ser usadas com 9 volts e ter um consumo de 0,1 amperes.



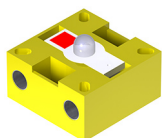
Lâmpada de lente

Nesta lâmpada encontra-se inserida uma lente, que foca a luz. Ela se parece muito com a lâmpada esférica. Você deve prestar atenção para não confundi-las. Para uma melhor diferenciação, o soquete de inserção desta lâmpada é cinza, enquanto a lâmpada esférica possui um soquete branco. A lâmpada de lente é necessária para a construção de uma [célula fotoelétrica](#).

As lâmpadas de lente, como as incandescentes, devem ser usadas com 9 volts e ter um consumo de 0,15 amperes.

Sensores

Sensores são os contrapontos dos atuadores, em certa medida. Porque eles não realizam ações, mas apenas reagem a certas situações e acontecimentos. Um botão de pressão reage, por exemplo, a um "aperto de botão", deixando passar eletricidade ou interrompendo-a. Um sensor térmico reage à temperatura em seu ambiente.



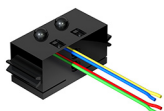
Fototransistor

O fototransistor também é denominado "sensor de luminosidade". Ele é um "sensor" que reage à luminosidade.

No caso da [célula fotoelétrica](#) ele é o contraponto da lâmpada de lente. No caso de alta luminosidade, ou seja, quando o transistor for irradiado pela lâmpada de lente, ele conduz corrente elétrica. Se o raio luminoso for interrompido, o transistor não conduz corrente elétrica.

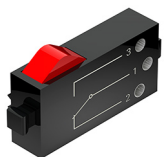
Atenção:

Quando da conexão do fototransistor na alimentação de corrente, prestar atenção à polaridade correta. O pólo positivo deve ser conectado à marcação vermelha no fototransistor.



Sensor de rastreamento

O sensor de rastreamento a **infravermelhos** é um sensor digital para a identificação de uma faixa preta sobre um substrato branco a distâncias de 5 a 30 mm. Ele consiste de dois elementos de emissão e dois de recepção. Como conexão você precisa de duas entradas digitais e uma voltagem de 9 volts (pólo negativo e positivo) no [ROBO TX Controller](#).



Botão de pressão

O botão de pressão é também denominado sensor de contato. Quando do acionamento do botão vermelho, será comutado mecanicamente um interruptor, irá fluir corrente entre os contatos 1 (contato central) e 3. Simultaneamente, o contato entre as conexões 1 e 2 será interrompido. Assim, você poderá utilizar o botão de pressão de duas maneiras diferentes:

Como "dispositivo de fechamento":

Serão conectados os contatos 1 e 3.

Botão de pressão apertado: Passa corrente.

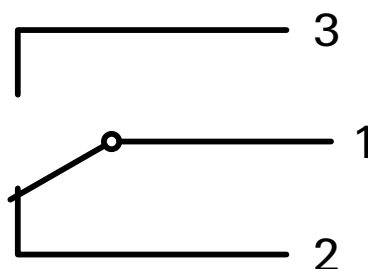
Botão de pressão não apertado: não passa nenhuma corrente.

Como "dispositivo de abertura":

Serão conectados os contatos 1 e 2.

Botão de pressão apertado: Não passa nenhuma corrente.

Botão de pressão não apertado: Passa corrente.



Sensor térmico (CNT)

No caso teste componente, trata-se de um sensor térmico, com o qual se pode medir temperatura. A 20°C, a sua **resistência elétrica** é de 1,5kΩ (quilo-Ohm). CNT significa Coeficiente Negativo de Temperatura. Isto significa, simplesmente, que o valor da resistência diminui com o aumento da temperatura.

As informações, que os sensores nos fornecem (p.ex. claro-escuro, apertado-não apertado, valor de temperatura), como veremos mais tarde, podem ser encaminhadas através do [ROBO TX Controller](#) para o PC e, auxiliados pelo software, p.ex., programadas em um motor, de maneira que ele ligue um ventilador tão logo uma célula fotoelétrica tenha sido interrompida.



Software ROBO Pro 2.x

ROBO Pro é uma interface gráfica de programação com a qual você pode compilar programas para o [ROBO TX Controller](#).

"Interface gráfica de programação" significa que você não precisa "escrever" o programa linha por linha, ao invés disso pode formar tudo através de símbolos gráficos de modo simples. Um exemplo de tal programa se encontra na figura à direita.

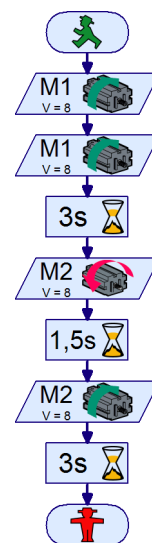
Como exatamente compilar tal programa está descrito em detalhes na [Ajuda Online ROBO Pro](#) nos capítulos 3 e 4.

A instalação do ROBO Pro e do driver para o ROBO TX Controller está descrita nas instruções de instalação encontradas no módulo.

O software se encontra no mesmo CD que este folheto.

Se você já tiver acabado a instalação do ROBO Pro, você pode iniciar o ROBO Pro imediatamente, já que em seguida usaremos da ajuda online do software.

É melhor você terminar de ler agora os dois primeiros capítulos da [Ajuda Online ROBO Pro](#). Assim você passa a conhecer o software um pouco para poder passar em seguida aos experimentos.





ROBO TX Controller

O ROBO TX Controller é o coração desse módulo de computação. Ele controla os [atuadores](#) e avalia as informações dos [sensores](#).

Para essa tarefa, o ROBO TX Controller dispõe de diversas conexões às quais você pode conectar os componentes. Quais componentes podem ser conectados a quais conexões e quais são as funções das conexões está descrito nas instruções de operação do ROBO TX Controller.

Uma funcionalidade especial é a interface Bluetooth integrada. Com ela, você pode conectar o seu PC ao ROBO TX Controller sem cabo. Ou diversos Controllers ao PC e uns com os outros.

Como o Controller lida com os diversos componentes e o que eles devem fazer especificamente é estabelecido por você através do programa que escreve no [software ROBO Pro](#).



Alimentação de corrente (não inclusa)

Já que, como você sabe, vários dos componentes do ROBO TX Training Lab funcionam a eletricidade, você certamente precisa de uma fonte elétrica.

O mais adequado para esse propósito é o Accu Set (conjunto de baterias) da fischertechnik. Ele não vem incluso no ROBO TX Training Lab.

Algumas dicas

Experimentar é geralmente divertido, se os experimentos funcionam. Por isso você deve prestar atenção a certas regras básicas ao construir os modelos:

- **Trabalhar com cuidado**

Não se apresse e consulte o manual de montagem do modelo com exatidão. Se for necessário procurar um erro, isso demora muito mais.

- **Verificar a mobilidade de todas as peças**

Ao montar, verificar sempre se as peças que devem se mover podem ser movidas com facilidade.

- **Usar o [teste de interface](#)**

Antes de começar a escrever um programa para um modelo, você deve testar todas as peças conectadas ao ROBO TX Controller com ajuda do teste de interface do ROBO Pro. Como exatamente isso funciona está descrito na [Ajuda Online ROBO Pro](#) no capítulo 2.4.



Primeiro passo

Pois então. Após todos os preparativos e informações, você pode finalmente começar.

Já que você vai trabalhar com o [software ROBO Pro](#), além dos componentes da fischertechnik para os experimentos, primeiro você deve se familiarizar com a criação de programas. E já que isso está explicado de modo claríssimo nos capítulos 3 e 4 da [Ajuda Online ROBO Pro](#), é melhor você prosseguir explorando cuidadosamente esses capítulos.

Aqui também vale a dica: Não se apresse e se ocupe com isso, logo em seguida você terá mais prazer com os modelos.



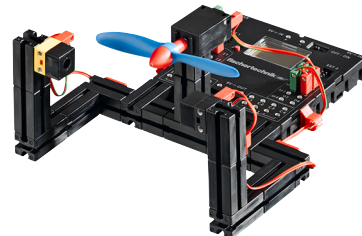
Modelos para iniciantes

Após ter lido completamente os capítulos 3 e 4 da [Ajuda Online ROBO Pro](#) você já poderá programar alguns modelos do módulo. Por isso, também gostaríamos de iniciar imediatamente. Sempre que você terminar de construir e cabear um modelo, verifique, com ajuda do [teste de interface](#), se no [ROBO TX Controller](#) todas as entradas e saídas estão conectadas corretamente e os [sensores](#), [motores](#) e [lâmpadas](#) funcionam corretamente.

O secador de mãos

Na sua escola foram instalados nos banheiros, junto às pias, novos secadores de mãos. Estes estão equipados com uma [célula fotoelétrica](#) através da qual se pode ligar e desligar a ventoinha.

- Construa primeiramente o modelo como está descrito no manual de montagem.



Tarefa 1:

- O secador de mãos deverá ser programado de maneira que, tão logo a [célula fotoelétrica](#) seja interrompida, a ventoinha seja ligada e, após 5 segundos, desligada novamente.

Dicas de programação:

- Ligue, no decurso do programa, primeiramente, a lâmpada para a [célula fotoelétrica](#) na [saída M2](#).
- A seguir, espere um segundo, para que o [fototransistor](#) tenha tempo para reagir à luz. Somente então a [célula fotoelétrica](#) funciona corretamente.
- A seguir, consulte o [fototransistor](#) na [entrada I1](#). Se o valor for "1" ([célula fotoelétrica](#) não interrompida), a entrada deverá ser continuamente consultada num loop.
- Tão logo o valor fique "0" ([célula fotoelétrica](#) interrompida), ligue o motor M1 e desligue-o novamente após 5 segundos.
- A seguir, o [fototransistor](#) deverá ser novamente consultado, etc.

Inicie o seu programa com o [botão Start](#) e verifique se ele funciona como desejado. Se funcionar, você está no melhor caminho para ser um programador ROBO Pro profissional.

Se ainda não funciona, tente descobrir onde está o problema:

- Com o [teste de interface](#) você pode verificar se todas as entradas e saídas funcionam e estão corretamente conectadas.
- Enquanto o programa corre, você pode acompanhar o decurso do programa, baseando-se nos componentes marcados em vermelho. Assim você pode identificar rapidamente onde se insinuou um erro.
- Por último, você pode comparar o seu programa com o programa pronto de exemplo que você pode abrir através do símbolo à direita:

Após ter vencido esta barreira, vamos modificar um pouco a proposição da tarefa:



Tarefa 2:

- O diretor da escola, que deseja poupar energia, não está satisfeito, pois o secador de mãos continua a funcionar por um determinado tempo mesmo com as mãos já secas. Ele solicita que você estruture o programa, de maneira que a ventoinha desligue logo que as mãos sejam retiradas debaixo do secador. Isto não é nenhum problema para você, é?

Dicas de programação:

- Como no primeiro programa, você consulta com uma derivação o [fototransistor](#) I1. Se o valor for "0", você liga o motor M1; se o valor for "1", você desliga o motor M1, etc.
- Também para esta tarefa existe, para um caso de emergência, um programa pronto:

O semáforo

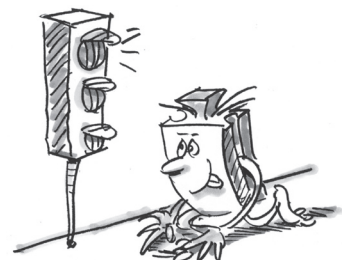
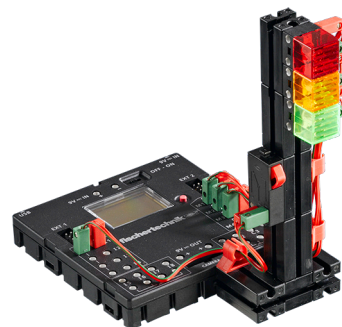
Na frente da sua casa foi instalado um semáforo. Como o montador da empresa de semáforos está sobrecarregado de trabalho, você oferece a ele a possibilidade de assumir a programação do controle do semáforo.

O montador lhe explica como o controle deve funcionar. Mas primeiro, monte o modelo.



Tarefa 1:

- O semáforo deve, em seguida, estar em verde. Se o [botão de pressão I1](#) for apertado por um pedestre, o semáforo deve comutar 3 segundos mais tarde para amarelo e, após outros 4 segundos, para vermelho. A fase em vermelho deve durar 10 segundos, a fase vermelho-amarelo subsequente, 3 segundos. Após isso, deve retornar ao verde.



Dicas de programação:

- As diferentes lâmpadas pertencem às seguintes saídas da interface:
 - Vermelho – M1
 - Amarelo – M2
 - Verde – M3
- Ligar as lâmpadas uma após a outra, de maneira que ocorra a seqüência desejada.
- O programa pronto pode ser carregado de novo clicando na imagem à direita:



Tarefa 2:

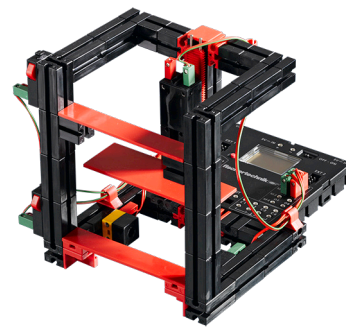
- No dia seguinte, o montador da empresa de semáforos lhe telefona. Ele esqueceu de dizer que na caixa de circuitos, na calçada, se encontra um interruptor I2, que deve comutar o semáforo para luz pisca-pisca amarela, logo que seja acionado. Você assegura ao montador que vai integrar imediatamente no seu programa esta função.

Dicas de programação:

- Afixe no seu modelo de semáforo um segundo [botão de pressão](#) e conecte-o à [entrada I2](#).
- Consulte com uma outra derivação a [entrada I2](#). Se o [botão de pressão I2](#) for apertado, o decurso é derivado para luz pisca-pisca. Caso contrário, o controle do semáforo decorre como na tarefa 1.
- A luz pisca-pisca é obtida através do ligar e desligar a [lâmpada M2](#) em intervalos de 0,5 segundo. Para isso, utilize um subprograma. Você pode ler como criar um subprograma no capítulo 4 da [Ajuda Online ROBO Pro](#).
- O programa de exemplo se encontra, como de costume, clicando no símbolo. Tente, porém, antes de verificar o programa exemplo, obter por si mesmo a solução. Muito sucesso!

O elevador

O seu vizinho instalou um elevador de cargas na oficina dele, de modo que ele não precisa mais carregar objetos pesados escada acima até o primeiro andar. Agora ele precisa apenas de um controle, que você programará para ele com o maior prazer.



Tarefa 1:

- Programe o elevador de modo que no início ele vá para baixo até sua posição de saída. Nessa posição, a [célula fotoelétrica](#) para I3 está interrompida. Se um dos dois [botões de pressão](#) for pressionado (I1 no térreo ou I2 no primeiro andar), o elevador deve ir até o outro andar.

Dicas de programação:

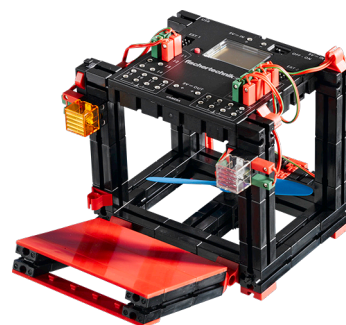
- Se a [célula fotoelétrica](#) estiver interrompida, o elevador está em baixo, se não, você pressupõe que o elevador está no primeiro andar.
- Que distância ele percorre de baixo para cima pode ser controlado durante o período em que o motor está ligado.
- Mesmo se você conseguir realizar isso sem ajuda, para todos os casos, temos uma sugestão de solução mais uma vez:

Antes que você parta para a próxima tarefa de programação, você deve abrir a [Ajuda Online ROBO Pro](#) novamente. Estude o capítulo 5 cuidadosamente. Comute o ROBO Pro para o nível 3. De pouco em pouco, as tarefas de programação vão se tornando mais exigentes. Utilizamos entradas analógicas, elementos de comando, operadores e variáveis. Mas quando você ler a ajuda online do ROBO Pro com atenção, será mais fácil, posteriormente, tratar destes assuntos.

A máquina lava-louça

Em seguida, você quer partir para a programação de uma máquina lava-louça. A lava-louça deve ter as seguintes funções:

- [Botão de pressão](#) para ligar e desligar (I1)
- [Botão de pressão](#) que detecta se a porta está fechada (I2)
- Função de lavagem (hélice em M1)
- Função de secagem (lâmpada [vermelha](#) em M2)
- Indicação de que a máquina está ligada (lâmpada [laranja](#) em M3)
- Indicação do nível operacional em que a máquina se encontra (lâmpada [transparente](#) em M4)
 - Piscando rápido: A máquina está lavando
 - Piscando devagar: A máquina está secando
 - Luz permanente: A máquina já terminou a tarefa



Tarefa 1:

- Compile um programa de lavagem que só começa se a porta estiver fechada e o botão start I1 for acionado. Primeiro ocorre a lavagem, depois a secagem. O nível operacional deverá ser indicado através das duas lâmpadas em M3 e M4.

Dicas de programação:

- Mais uma vez, nossa solução sugerida:


Tarefa 2:

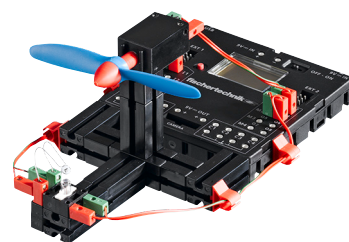
- Se a porta for aberta, o processo de lavagem deverá ser interrompido. Após o fechamento da porta, o programa prossegue a partir do ponto em que foi interrompido.
- Adicionalmente, o nível operacional da máquina lava-louça é mostrado no display do [ROBO TX Controller](#).

Dicas de programação:

- É verdade que essa tarefa é difícil. Se você conseguir resolvê-la sozinho: Ótimo! Se você não tiver progresso, apesar de muito esforço, não se preocupe. Consulte nossa solução sugerida.

Regulagem da temperatura

Foi instalado na sua casa um novo equipamento de ar condicionado. Naturalmente, você perguntou ao instalador imediatamente como funciona a regulagem de temperatura. Ele explicou prontamente que o sensor de temperatura mede continuamente a temperatura actual. Tão logo seja ultrapassado um valor limite superior, a refrigeração liga. Do contrário, se um valor limite inferior não for atingido, a refrigeração desliga e o aquecimento liga. Assim, auxiliado pelo modelo "Regulagem da temperatura", você quer programar um [circuito de regulagem](#) como este. Monte, primeiramente, o modelo.

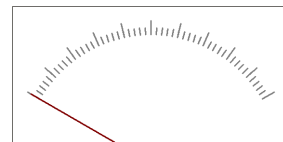

Tarefa 1:

O aquecimento será simulado pela [lâmpada de lente](#) M2. Como "agregado de refrigeração", a ventoinha atua na [saída M1](#). Para a medição de temperatura, utilizamos a [resistência CNT](#) na [entrada I2](#).

- Programe o modelo de maneira que, acima de uma determinada temperatura, o aquecimento desliga e a ventoinha liga. Este deve refrigerar até que um valor limite mínimo seja atingido. Logo após, a ventoinha deverá ser desligada e o aquecimento ligado.
- O valor atual da entrada analógica deve ser emitido para um medidor e uma indicação textual no ROBO Pro, bem como no display do [ROBO TX Controller](#).

Dicas de programação:

- **Observe:** O valor de resistência da [resistência CNT](#) abaixa com o aumento da temperatura. O valor limite superior da temperatura é, assim, o menor valor de I2. Neste valor limite, a ventoinha deve ligar. O valor limite inferior da temperatura é o maior valor de I2. Neste valor limite, o aquecimento deve ligar.
- Com o [teste de interface](#) você pode encontrar qual valor I2 possui sob temperatura ambiente. Para isso ligue a lâmpada M2 e observe o quanto o valor cai. A seguir, ligue a ventoinha e descubra o quanto o valor aumenta. De maneira correspondente, selecione os valores limite para o aquecimento e a refrigeração.
- Indique o valor da entrada analógica no seu programa com uma indicação textual e/ou com um aparelho de medição (veja também a [Ajuda Online ROBO Pro](#), capítulo 8.1).



Temp. = 0

- Clicando no símbolo à direita, você pode abrir o programa pronto.

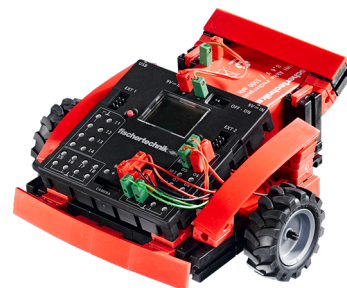


Robôs - o próximo desafio

Modelo básico

Com esse modelo, queremos descobrir juntos como controlar um robô em movimento. Como fazê-lo se mexer, como funciona a direção, e é possível melhorar ainda mais sua precisão? Essas perguntas são respondidas com ajuda das tarefas neste capítulo.

Mas primeiro, você deve construir o robô. A descrição se encontra no manual de montagem, como sempre.



- Construa o robô como descrito no manual de montagem.
- Não tenha pressa ao montar o robô. Consulte os desenhos e o cabeamento no manual de montagem com cuidado. Se você conectar os componentes com o [ROBO TX Controller](#) diferentemente do descrito no manual de montagem, o robô se comportará diferente do que você espera.
- Após a construção, verifique todos os componentes conectados ao [ROBO TX Controller](#) através do [teste de interface](#) do [software ROBO Pro](#). Se você girar os motores para a esquerda, o robô deve ir para a frente.



Tarefa 1: Simplesmente avante (nível 1)

- Faça o robô se mover por 3 segundos reto para frente (**não sobre a mesa, perigo de queda!**), e em seguida mais 3 segundos reto para trás.
 - *O robô realmente voltou ao seu ponto de partida?*
- Repita o programa várias vezes e observe se o robô se movimenta reto para frente e para trás.

Dicas de programação:

- Mesmo se a tarefa não for problema para você, eis nossa sugestão:

A direção

Mesmo se for divertido observar o robô se mover reto, fica um pouco monótono. Portanto ele deve aprender a fazer curvas agora. Como? Bem simples:



Tarefa 2: Fazendo curvas (nível 1)

- Deixe o robô andar mais uma vez 3 segundos em linha reta (ambos os motores rodam com a mesma velocidade), então altere por 1 segundo a direção de rotação do motor direito (M1) e deixe o robô em seguida mais 3 segundos andar em linha reta (ambos os motores com a mesma velocidade e a mesma direção).
- Descubra quanto tempo leva com os motores rodando em direções diferentes para o robô girar 90°.

Dicas de programação:

- Altere o tempo de espera depois do elemento de controle no qual a direção do segundo motor é alterada.
- O programa pronto pode ser carregado de novo clicando no símbolo à direita:



Tarefa 3: Alcançando uma figura (nível 2)

- Agora que você já sabe quanto tempo é necessário de alteração da direção de rotação do motor para o robô virar para a esquerda ou para a direita, programe o robô de modo que ele se mova num quadrado e em seguida volte para o seu ponto de partida.
 - *Verifique com uma marcação se o robô realmente volta para o seu ponto de partida.*

Dicas de programação:

- Para a "virada da esquina", você pode compilar um sub-programa. O seu programa principal fica mais claro assim.
- Você certamente já tem em mente a solução para essa tarefa. Mas em qualquer caso, aqui vai nossa solução sugerida mais uma vez:

Sempre a mesma coisa, mas mesmo assim diferente?

Como você certamente notou, a [precisão de repetição](#) do robô sempre pode ser melhorada. Mesmo se ele realiza a mesma tarefa diversas vezes, o resultado nem sempre é o mesmo. Isso tem diversos motivos. Um deles é que os dois motores não rodam exatamente com a mesma velocidade. Por exemplo, a engrenagem de um motor podem ter menor mobilidade do que a do outro. E já que ambos os motores operam com a mesma voltagem (9 volts), um motor roda mais devagar que o outro mesmo assim. Já que nós operamos nosso robô além dos tempos de espera, talvez uma roda tenha girado mais que a outra durante esse tempo.

A solução seria, portanto, fazer com que os dois motores rodassem com a mesma velocidade, exatamente. E isso é perfeitamente possível com os [motores do codificador](#).



Tarefa 4: Usando os motores do codificador

- Repita as últimas três tarefas e no lugar dos [elementos de saída do motor](#) e de [tempo de espera](#) use os [elementos do motor do codificador](#). Como estes se usam está descrito na [Ajuda Online ROBO Pro](#) no capítulo 11.6.

Dica de programação:

- Com o [elemento do motor do codificador](#) você pode acionar os dois motores ao mesmo tempo com um elemento do programa. Através do [campo de informação de distância](#) você garante que cada motor roda exatamente o quanto deve.
- As soluções sugeridas podem ser abertas de novo clicando no símbolo à direita:

Para contar os impulsos nas [entradas de contador C1-C4](#) rápidas, você não precisa de mais nenhum elemento de programa em [ROBO Pro](#). Ao motor M1 é alocada internamente e automaticamente a entrada de contador C1, M2 pertence a C2, etc.

Aviso:

Se um modelo, apesar do uso dos [elementos do motor do codificador](#) não andar reto, a causa pode estar no próprio modelo. Se, por exemplo, uma porca de centro que suporta a força do eixo para as rodas não for aparafusada muito bem, o eixo gira e o modelo faz uma curva apesar de os motores girarem com a mesma velocidade. Portanto, aperte as porcas muito bem.

O detector de pistas



O seu robô agora pode andar em linha reta e fazer curvas. E até agora ele se comporta exatamente como você o programou.

Mas um robô deve ser capaz de reagir independentemente quando possível. Por isso vamos dar a ele algo para reagir agora: Uma linha preta como marca no chão.

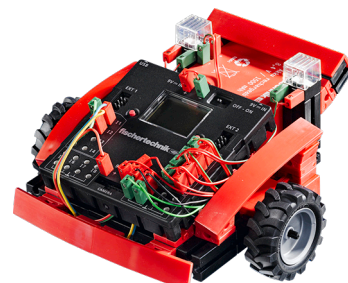
O objetivo é que o robô procure a linha preta e ande ao longo dela.

Mas uma coisa de cada vez. Primeiramente você precisa transformar o modelo básico em detector de pistas. Para fazer isso,

consulte o manual de montagem.

Após trocar o modelo, você tem que verificar com o [teste de interface](#) se todos os componentes estão conectados corretamente no [ROBO TX Controller](#) e se eles funcionam. Você pode testar o [sensor de rastreamento](#) segurando-o sobre a pista preta do percurso e movimentando-o de lado. Os sinais nas entradas aos quais o [sensor de rastreamento](#) está conectado devem mudar com isso.

Pense em ajustar as entradas no [teste de interface](#) para "digital 10V (sensor de rastreamento)".




Tarefa 1: Reconhecendo uma pista (nível 2)

- Programe o robô para que ele siga uma pista preta reta sobre a qual ele foi posto. Se ele perder a pista ou se ela terminar, o robô deve ficar parado e as duas lâmpadas devem piscar três vezes cada. Para essa tarefa, use o percurso 1a do módulo.

Dicas de programação:

- Primeiro você pede as duas entradas do [sensor de rastreamento](#). Se as duas entradas receberem o sinal "0", o robô fica sobre a pista preta. Você pode soltá-lo.
- A função de piscar pode ser programada em um sub-programa.
- Use novamente o [elemento do motor do codificador](#) para movimentos em linha reta mas sem informar a distância.
- Você pode encontrar o programa pronto em:

O seu robô agora pode reagir. Contudo, essa função ainda é bastante limitada. Seria melhor se o robô, ao invés de parar, pudesse corrigir sua direção para poder seguir a pista.


Tarefa 2: Seguindo uma pista (nível 2)

- Expanda o seu programa com a função de fazer o robô corrigir sua direção e seguir a pista se ele sair da pista. Teste essa tarefa primeiro com o percurso 1a, então com o percurso 1b.

Dicas de programação:

- Existem várias possibilidades de corrigir a direção. Você pode parar um motor e deixar o outro rodar, ou fazer um motor rodar contra a direção da movimentação. Descubra qual método é mais adequado.
- Eis a nossa solução sugerida:

Então. Agora o robô pode andar sobre um "trilho" ótico. A desvantagem é apenas que é preciso primeiro colocá-lo sobre o trilho. Nós queremos mudar isso. O robô deve ser capaz de procurar sua pista sozinho.


Tarefa 3: Achando a pista e então seguindo-a (nível 2)

- Escreva um sub-programa "procurar" com o qual o robô procura por uma pista se ele não encontrar uma no início do programa. Para tal, o robô deve ser capaz de girar uma vez no círculo primeiro. Se ele não encontrar uma pista, ele deve andar em linha reta por um pedaço. Assim que o robô reconhecer uma pista, ele deve segui-la. Senão a busca deve começar novamente da frente. Se ele girou 10 vezes no círculo sem encontrar uma pista, ele deve ficar parado e piscar 3 vezes.

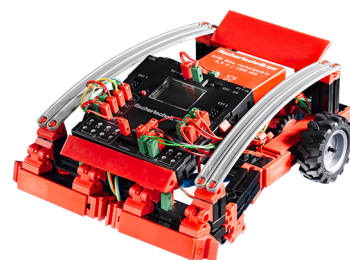
Dicas de programação:

- Se você ainda não estiver na pista certa, eis nossa solução sugerida:

O corta-grama

Robôs sabem cortar grama? É claro que sim. É preciso apenas dizer-lhes como reagir diante de obstáculos e onde o gramado acaba. E então já é possível delegar essa tarefa ao robô e passar a tarde na piscina.

Mas antes que cheguemos a isso, você precisa primeiro montar o robô corta-grama conforme o manual de montagem. Em seguida, você deve verificar mais uma vez, com o [teste de interface](#) se tudo está funcionando como deve.



Então você pode partir para a programação. O gramado é simbolizado pela superfície branca de nosso estádio de futebol Percurso 1b, delimitado por um canteiro preto. Esse canteiro não pode ser ultrapassado pelo corta-grama (porque ele cairia sobre a pista de atletismo, por exemplo). Se aparecer um obstáculo sobre o gramado, o robô deve desviar chegar a ele. Além disso o corta-grama deve ser desligado se for encontrado um obstáculo.



Tarefa 1: Reconhecendo e desviando de limites e obstáculos (nível 2)

- Programe o corta-grama para que ele ande em linha reta para frente a partir de sua posição inicial (dentro da limitação) até encontrar um obstáculo ou alcançar o limite do gramado (linha preta).
- Se o corta-grama se deparar com um obstáculo (pára-choques na frente), ele deve parar imediatamente, desligar o aparador, andar um pouco para trás, girar para a esquerda, e andar de novo para frente ligando o aparador novamente. Essa função deve ser incluída no sub-programa "desviar".
- Se o corta-grama alcançar o limite do gramado, ele também deve parar imediatamente e iniciar o sub-programa "desviar".

Dicas de programação:

- Nossa solução proposta:

Dependendo do tamanho do seu "gramado", podem aparecer os seguintes problemas: dependendo de qual duração da "virada" for configurada, o robô só anda ao longo da borda ou sempre fica na mesma área que o "gramado". Por isso o robô deve se comportar sempre um pouco diferente ao desviar.



Tarefa 2: O aleatório (nível 3)

- Altere o programa do corta-grama para que ele gire a um ângulo diferente a cada vez que desvia. Assim ele desviará uma hora mais, outra hora menos. Além disso, ele deve desviar para a esquerda ao reconhecer um obstáculo com o pára-choques direito, e vice-versa.

Dicas de programação:

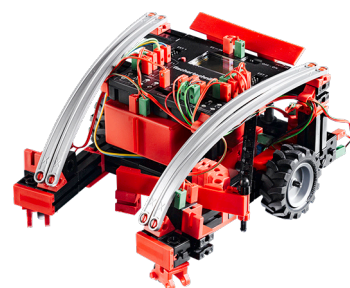
- Para poder tirar vantagem das variáveis em ROBO Pro, você precisa ajustar o software para o "nível 3".
- Para tal você precisa de um [gerador aleatório](#). Você pode compilá-lo se você contar o valor de uma variável de um [loop de contagem](#) sempre de "0" sobre um certo valor. E no sub-programa "desviar" você configura um "esperar por..." conforme a ordem para rodar o corta-grama. Elemento que espera que a variável vire "0". Já que a variável provavelmente tem um outro valor cada vez que for consultada, sempre demora um intervalo diferente para que volte a "0". E assim se produzem tempos diferentes ao rodar o robô.
- É verdade que isso não é tão simples assim. Se você não encontrar a solução após pensar muito, temos naturalmente uma solução proposta:

O robô jogador de futebol

Já ouviu falar da Copa dos Robôs? Trata-se de um campeonato mundial de futebol para robôs. Ela é realizada todo ano em um país diferente. Existem diversas ligas onde se encontram vários tipos de robôs. Mais informações podem ser encontradas, por exemplo, no website da Copa dos Robôs <http://www.robocup.org>.

No manual de montagem se encontra uma sugestão para a construção de um robô jogador de futebol. Ele é tão móvel quanto nossos outros robôs, mas possui ainda uma célula fotoelétrica para reconhecer uma bola e um "mecanismo de chute". Construa-o logo e, em seguida, vamos programá-lo e treiná-lo com alguns truques de jogo. E como sempre, você deve primeiro verificar o funcionamento do modelo com o [teste de interface](#) antes de começar com a programação.

Você pode usar, por exemplo, uma bola de tênis como bola (não inclusa no módulo). Dependendo de que tipo de bola você usar, é necessário talvez ajustar o mecanismo de chute um pouco.




Tarefa 1: "Daí ele dominou a bola e chuta ..." (nível 2)

- No primeiro passo, o nosso craque eletrônico deve aprender a reconhecer a bola e reagir a ela. Programe-o para que ele chute a bola assim que ela foi reconhecida pela [célula fotoelétrica](#). Experimente um pouco com a "velocidade de chute". Se necessário, uma pausa curta entre "reconhecer" e "chutar" pode gerar uma melhoria.
- Então ele deve aprender o "chute de pênalti". Ponha uma bola sobre o ponto de pênalti do percurso 1a. Ponha o robô no começo da linha preta. Agora ele deve começar ao longo da linha e assim que ele reconhecer a bola, deve chutá-la em direção ao gol. No final da linha, ele deve parar e virar.

Dicas de programação:

- Como com o [secador de mãos](#) você precisa esperar um segundo com a [célula fotoelétrica](#) após o ligamento da [lâmpada de lente](#) antes de consultar o [fototransistor](#).
- Não é fácil ser treinador. Se o seu jogador robótico não lhe escutar, talvez você possa convencê-lo com a nossa sugestão de programa:

Já que um verdadeiro craque precisa ser capaz de mais do que pênaltis, queremos aumentar as capacidades de nosso robô jogador um pouco mais.


Tarefa 2: Quem procura, acha. (nível 2)

- O robô jogador de futebol deve perambular dentro do estádio (percurso 1b) sem passar da linha de delimitação. Se ele encontrar uma bola, ele deve chutá-la, obviamente em direção ao gol.

Dicas de programação:

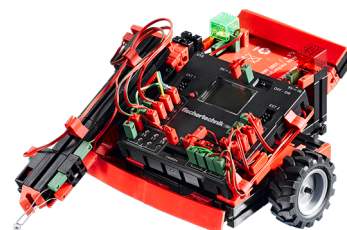
- A maior parte das funções para esse programa você já programou no [corta-grama](#). Portanto você pode usar o programa do corta-grama, salvando-o com outro nome e expandi-lo com as funções do robô jogador de futebol.
- Provavelmente você não vai precisar dela, mas mesmo assim temos uma sugestão de solução:

O robô medidor

É mais quente embaixo de sua cama do que sob a escrivaninha? E que temperatura tem a chama de uma vela? E pode-se resfriar um quarto com um cubo de gelo?

Para responder esta (e mais) perguntas, você pode usar o robô medidor. Ele vem equipado com um [sensor de temperatura \(CNT\)](#) e pode medir e exibir a temperatura de diversos locais. Adicionalmente, o robô medidor tem um [sensor de rastreamento](#) de modo que você pode ensiná-lo um percurso através de uma linha preta.

Construa o robô medidor como descrito no manual de montagem e teste seu funcionamento com o [teste de interface](#).


Tarefa 1: Controle e medição de temperatura (nível 3)

- Pegue o programa do [detector de pistas](#) e expanda-o com um sub-programa para controlar o braço medidor. O robô deve andar sobre a pista do percurso 1b e medir a temperatura em certos intervalos. A temperatura deve ser exibida no display do [ROBO TX Controller](#).

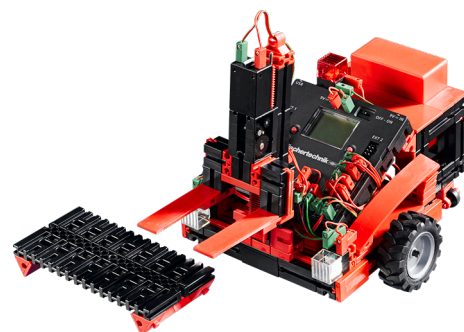
Dicas de programação:

- Para exibir o valor da resistência do [sensor de temperatura](#) você precisa de uma [entrada analógica](#), de uma [saída do campo de comando](#) e de um [elemento de comando de "exibição"](#).
- Na entrada analógica, o valor de resistência do [sensor de temperatura](#) é exibido, e não a temperatura em si. Para converter esse valor para uma indicação de temperatura você pode utilizar um sub-programa.
- A nossa solução proposta completa para essa tarefa se encontra aqui:

A empilhadeira de forquilha

Uma área na qual os robôs são empregados com muita aceitação é a logística. Portanto em qualquer lugar onde as coisas precisam ser movimentadas de "A" a "B", como se diz.

Tais tarefas de transporte podem ser realizadas com o modelo de empilhadeira de forquilha. Construa o modelo como descrito no manual de montagem. Em seguida verifique com o [teste de interface](#) se todos os componentes estão funcionando corretamente.


Tarefa 1: Para cima e para baixo (nível 3)

- Escreva um sub-programa para a função "levantar" e um para "baixar".
- Já que durante a viagem a forquilha não pode estar nem muito para cima nem muito para baixo, você ainda precisa de um sub-programa "posição de viagem".

Dicas de programação:

- Ao "levantar", a forquilha deve ser elevada para cima (o motor M3 roda para a esquerda) até que o interruptor final superior I4 seja acionado.
- Ao "abaixar", a forquilha deve ser baixada (o motor M3 roda para a direita) até que a engrenagem no interruptor final inferior I3 seja ultrapassado e o interruptor esteja aberto de novo. Para tal você deve utilizar um [elemento "esperar por"](#) e ajustar para "1 -> 0 (caindo)".
- Para a "posição de viagem", a forquilha deve ser colocada para cima até que a forquilha se encontre logo acima do interruptor final inferior.
- Como sempre temos uma proposta de solução:

Depois que a mecânica passar a funcionar perfeitamente, gostaríamos de usá-la. Finalmente, uma empilhadeira de forquilha deve poder transportar coisas, além de poder movimentar sua forquilha.


Tarefa 2: De "A" a "B" (nível 3)

- No módulo existe um percurso 2a para a empilhadeira de forquilha. Ele deve começar a partir de sua posição inicial, pegar um pallet que está no campo "A", transportá-lo para o campo "B" através da pista e descarregá-lo ali.

Dicas de programação:

- Programe a empilhadeira de forquilha de modo que ela carregue ou descarregue o pallet sempre que a pista terminar. Para cada um desses dois processos, use um sub-programa.
- Deixe a empilhadeira andar devagar de modo que ela não saia da pista de modo algum acidentalmente. Senão ela pensará que chegou ao final e deve carregar ou descarregar o pallet.
- Nossa solução proposta:



Tarefa 3: Reconhecendo bifurcações

Agora a coisa fica interessante. A empilhadeira deve reconhecer uma bifurcação. Use o percurso 2b para isso.

- A empilhadeira começa de sua posição inicial. Ela deve ir ao campo "A", pegar o pallet ali e transportá-lo ao campo "B". Em seguida ela deve voltar à sua posição inicial. Enquanto ela segue a pista, seus faróis devem estar ligados. Se ela virar ou andar de marcha-ré, deve acender a lâmpada vermelha.

Dicas de programação:

- A interrupção da pista pode ser reconhecida como uma bifurcação pela empilhadeira. Quando ela chegar a esse ponto, você pode decidir como ela vai continuar: Ela pode andar em linha reta até encontrar a pista, fazer uma curva à esquerda e assim procurar a pista ou virar à direita até que encontre a pista novamente. Compile um sub-programa para cada um desses 3 casos.
- E para encontrar a pista ao fazer curvas para a esquerda ou para a direita, o robô precisa andar um pequeno pedaço em linha reta primeiro. Senão ele gira até passar da pista.
- Se você perder a pista ao programar, consulte a nossa proposta:



Tarefa 4: Transportando sem fim

- Expanda o programa da tarefa 3 de modo que a empilhadeira faça uma pequena pausa após voltar à sua posição inicial e então pegue o pallet do campo "B", leve ao campo "A" e volte ao início. O processo inteiro deve ser repetido pela empilhadeira agora como na esteira de montagem.

Dicas de programação:

- Produza um loop infinito ao retornar uma linha do último elemento de programa para o início do programa.
- Mesmo se a tarefa não for problema para você, eis nossa sugestão:



Se não der certo no começo...

... geralmente existe um motivo simples para isso. Mas nem sempre é fácil encontrá-lo. Portanto gostaríamos de dar-lhe algumas dicas neste local quanto a causas de erros.

Teste de Interface

Uma dica aqui novamente: Verifique o funcionamento de cada componente com ajuda do [teste de interface](#) em [ROBO Pro](#).

Cabos e cabeamento

Se um componente elétrico não funcionar de jeito nenhum, verifique o cabo com o qual você o conectou ao [ROBO TX Controller](#). Conecte então com o cabo a [bateria](#) e uma [lâmpada incandescente](#). Se a lâmpada acender, o cabo deve estar em ordem.

Plugues instalados de modo errado (p.ex. um plugue verde com um cabo vermelho) também podem ser causas de erros.

Verifique se "+" e "-" estão conectados corretamente. Compare o seu modelo com as imagens do manual de montagem.

Mau contato

Um componente que funciona intermitentemente e então pára de funcionar deve estar com mau contato em algum ponto de seu cabeamento.

As causas principais são:

- **Plugues soltos**

Se os plugues dos cabos estiverem soltos demais, e portanto não estão bem encaixados, podem não estar com contato suficiente. Nesse caso, você pode usar a chave-de-fenda para desembaraçar as molas de contato diante dos plugues relevantes **com cuidado**. Mas apenas de leve, para que o plugue volte a ficar fixo no encaixe se for conectado.

- **Mau contato entre cabo e plugue**

Verifique também o contato entre as extremidades [desencapadas](#) do cabo no plugue e no plugue em si. Possivelmente basta apertar um pouco mais os parafusos no plugue.

Curtos-circuitos

Ocorre um curto-circuito quando uma conexão positiva encosta em uma negativa. Tanto a [bateria](#) quanto o [ROBO TX Controller](#) são equipados com fusíveis, de modo que um curto-circuito não os danifique. Eles desligam a eletricidade por um curto período. O seu modelo deixa então de funcionar, naturalmente.

A causa de um curto-circuito pode ser um erro no cabeamento ou parafusos mal apertados nos plugues. Eles podem se tocar se os plugues estiverem inseridos de acordo, produzindo um curto-circuito. Portanto você sempre deve apertar os parafusos completamente e encaixar o plugue de modo que os parafusos não possam se tocar.

Alimentação elétrica

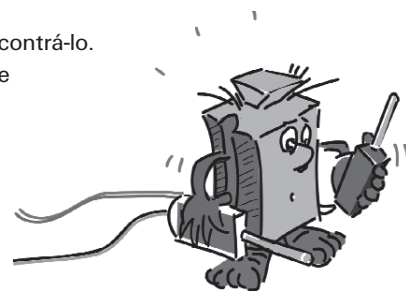
Pequenos blecautes ou motores muito lentos geralmente indicam [bateria](#) descarregada. Nesse caso, você deve carregar a bateria com o carregador fornecido. Se o LED vermelho no carregador parar de piscar e ficar permanente, a bateria está plenamente carregada.

Erro no programa

Mesmo que ninguém goste de admitir: todo mundo comete erros. E especialmente em programas complexos, sempre pode ocorrer um erro.

Se você verificar por si mesmo o modelo e eliminar todos os erros, e o seu modelo mesmo assim não fizer o que você quer, então você deve verificar o seu próprio programa. Vá peça por peça e veja se você encontra o erro.

No modo online, portanto se o [ROBO TX Controller](#) estiver conectado com o PC você pode seguir o programa na tela enquanto ele corre. O elemento ativo do programa é destacado, de modo que você sempre pode ver em que local está o programa e onde aparece o erro.



O último coringa

Se apesar de tudo você não encontrar o erro, ainda existem duas possibilidades para obter ajuda:

- **Coringa por e-mail**

Você pode enviar um e-mail para nós na fischertechnik descrevendo o seu problema.

O endereço é info@fischertechnik.de.

- **Coringa do público**

Você também pode nos visitar na internet no endereço <http://www.fischertechnik.de>. Lá você encontra, entre outras coisas, um fórum, onde você pode achar ajuda. Além disso você pode se tornar membro do fanclub da fischertechnik gratuitamente.

E como prosseguir?

Isso é tudo? É claro que não. Os experimentos e modelos que você passou a conhecer e testou neste folheto são apenas o começo. Por assim dizer, são os seus primeiros passos no gigantesco e excitante tema de "computing".

O que lhe mostramos aqui é apenas uma pequena fração das possibilidades que você tem com o [ROBO TX Controller](#) e com os componentes da fischertechnik. E a partir de agora você mesmo é requisitado. Você pode dar rédeas à sua imaginação e começar a construir à vontade.

Se você ainda não tem uma idéia de um modelo completo próprio, dê uma olhada nos modelos deste folheto. Talvez algum deles lhe chame a atenção e lhe dê idéias para fazer um modelo diferente. Ou você pode alterar a função de um modelo.

Por exemplo, no caso da [empilhadeira de forquilha](#), um lápis poderia ser instalado no lugar da forquilha, que pode ser levantado e abaixado e pintar sobre um pedaço de papel enquanto o robô passa por cima. Você então poderia não apenas criar, como também desenhar figuras com o movimento. E assim a empilhadeira se torna uma máquina de desenhar.

E se amigos seus também possuírem um [ROBO TX Controller](#), tudo fica ainda mais interessante. Já que com a interface Bluetooth, não só o seu PC pode se comunicar com o [ROBO TX Controller](#), mas também outros controladores entre si. Assim, por exemplo, vocês podem programar 2 robôs para reagirem um ao outro. Ou para dançarem um com o outro. No capítulo 7 da [Ajuda Online ROBO Pro](#) você encontra informações interessantes sobre esse tema.

O seu computador tem uma interface Bluetooth? Caso tenha, você pode conectar o [ROBO TX Controller](#) ao computador através do Bluetooth ao invés de um cabo USB. Se não tiver, você pode comprar um stick USB de Bluetooth e assim conectar o [ROBO TX Controller](#) sem cabos ao PC. Para saber como proceder, consulte as instruções de operação do [ROBO TX Controllers](#) e o site <http://www.fischertechnik.de>.

Então, o que você está esperando? Comece! Invente e experimente. E não deixe se abalar por pequenos contratempos. Para experimentar, é necessário antes de tudo paciência e perseverança. A recompensa é um modelo funcional depois de todo o esforço.

Desejamos muita diversão a você ao testar as suas próprias idéias.

