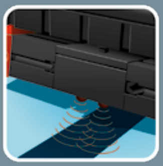
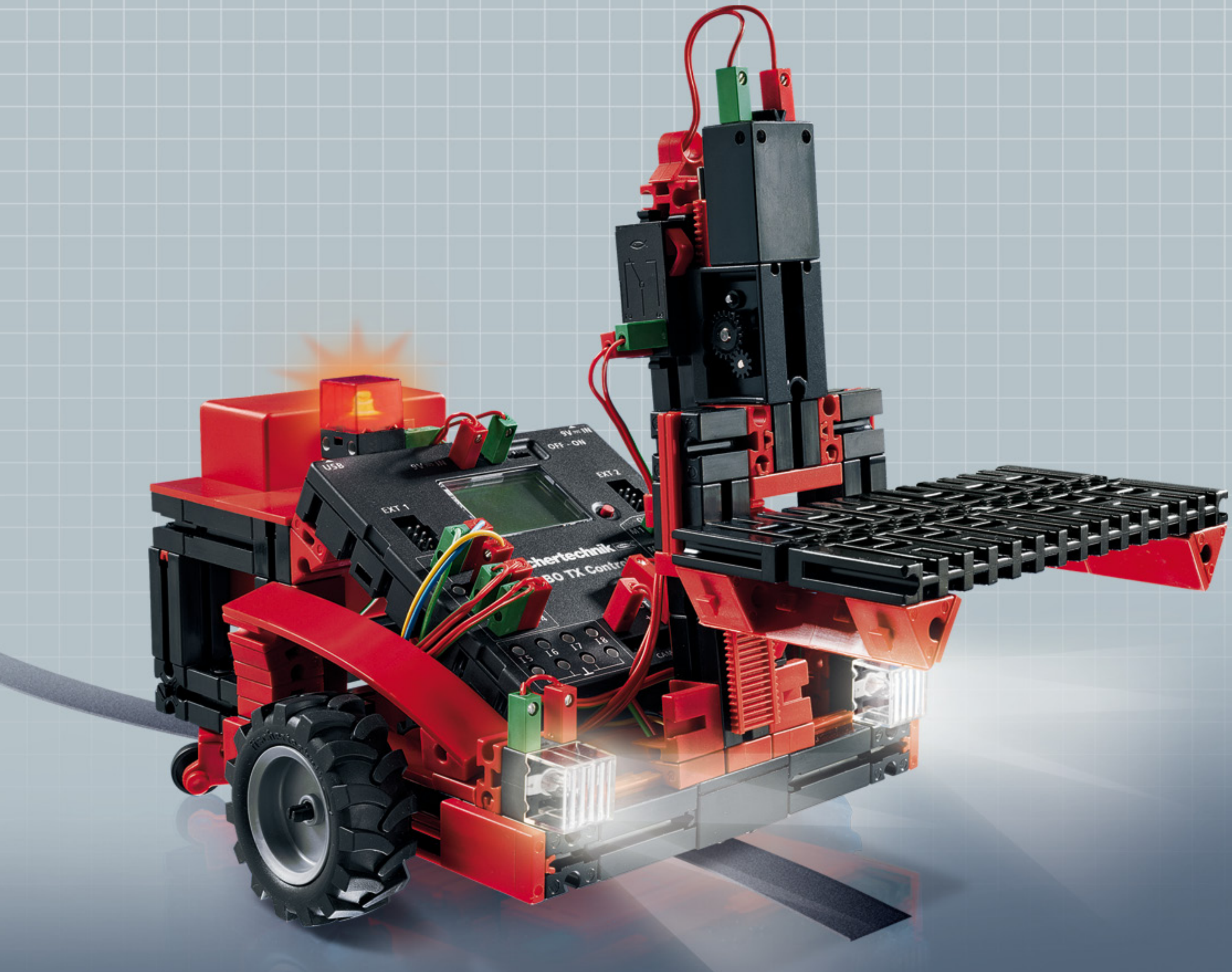


fischertechnik®

COMPUTING

Activity booklet



ROBO TX Training Lab

11 MODELS

Contents

Welcome to the ROBO TX Training Lab

Hello!

We are happy that you have chosen the construction set "ROBO TX Training Lab" from fischertechnik. And we promise you that your interest will be rewarded. This is because with this construction set you can conduct a lot of interesting experiments and solve exciting tasks.

When you read this screen booklet and try the experiments and tasks, you will learn step-by-step how you can control and program simple and also complicated machines and robots using the ROBO TX Controller from fischertechnik.

Since learning is just this way, you can't immediately start with the most difficult things even if they are naturally often a little bit more interesting than the somewhat more simple ones. This is why we have structured the experiments and tasks in this booklet so that you learn something different with every new task and can then use this for the next task.

So don't worry, we start with small things and then we work together to progress to the big robots.

We hope you have a lot of fun and success now with the experimentation with the ROBO TX Training Lab.

Your team from

fischertechnik

Some General Information

Before we really get started with the construction set, you still need to know a few things. The components, which we will work with, are of course very robust, but if you don't handle them correctly, they can be damaged under certain circumstances.

Electricity

As you certainly know, a lot of the components in the ROBO TX Training Lab use electric power. And for things, which have to do with electricity, you must be particularly careful not to make any mistakes. That is why you should always read the assembly instructions very carefully when wiring the electrical components.

Do not connect the positive and negative poles with each other in any case, which means do not short-circuit. If this is done then this can damage the ROBO TX Controller or even the rechargeable battery.

Electricity and electronics are just as interesting topics as is robotics (which is just what this construction set is about) and there is a construction set from fischertechnik, which deals with these subjects in particular. So if you are interested in this then you will also have just as much fun with the "PROFI E-Tech" construction set as with the ROBO TX Training Lab.

About this Activity Booklet

This pdf activity booklet has a few functions, which you do not find in a printed booklet and which you may already know from the Internet.

Links within the Booklet

When something is mentioned somewhere in the text, which is explained in more detail at another point in this booklet (for example, components), then this text is in dark blue and underlined. You can click on the text and thus move automatically to the page, which contains the explanation. This is called a "cross reference."

Background Infos

In some cases in this booklet, there are terms or foreign words, which may require explanation. These terms are in green and underlined. If you touch the text with the mouse pointer then a field appears with an explanation.

Link Outside of this Booklet

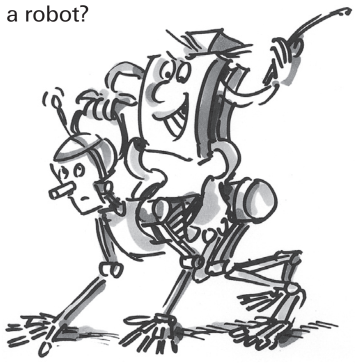
You need an Internet connection for a few links (for example, for the fischertechnik Web site), or an installed ROBO Pro (for connection to the ROBO Pro online help). These links are in bright blue and underlined.

Pictures

A picture is worth a 1000 words. You have certainly heard this sentence already. And because this certainly contains a lot of truth, you can display a picture by touching the words in brown and are underlined and the picture shows you what is meant in the text.

The robot, the artificial human being?

What do you first think about when you hear the word "robot"? Have you ever seen a robot? In a movie or on television? Or perhaps a real one?



There are innumerable different types of robots. Some robots look a bit like a human, but others only consist of one or several arms. So, what exactly makes a robot a robot?

The dictionary states: "Robots are stationary or mobile machines, which perform set tasks according to a certain program."

Computing, (Almost) Everything Automatic

Thus, robots are machines, which are controlled by a program. And we call this control of machines (or in our case models) "computing."

The "ROBO TX Training Lab" provides you with a wonderful start to learn about this subject. This is because the construction set contains everything that you need to build and control many different machines.

You can create the programs for the control of the models on a PC with the help of the software, ROBO Pro, and then transfer them to the ROBO TX Controller using the USB or Bluetooth connection. The Controller then "controls" and steers the model according to the programing, which you have prepared.

Component Explanations

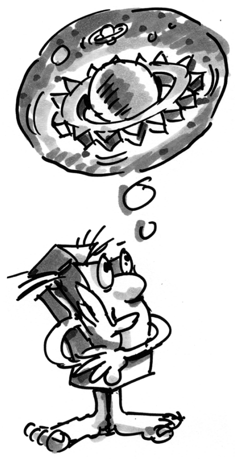
The Construction Set Contains All of This

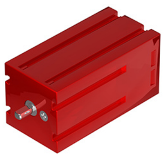
First, you will find several fischertechnik building blocks, as well as motors, indicator lights and sensors and colored assembly instructions for the building of various models.

After you have unpacked all of the building blocks, you have to assemble some components first such as cables and plugs before you can really get started. The exact ones are described in the assembly instructions under "Assembly Tips." It is best to do this first.

Actuators

Actuators are all components, which can perform an action. This means that they become "active" in some form when they are connected to electric power. In most cases you can see this directly. A motor runs, an indicator light is illuminated and so forth.





Encoder Motors

We use the two encoder motors, which are contained in the construction set, as the drive for our robots. At first glance, these are normal electric motors, which are designed for a voltage of nine volts and current input of a maximum of 0.5 amperes.

But the encoder motors can do still more: In addition to the connection for the power supply for the motor, they have another jack socket for a three-pin connection cable and through this and with the aid of what is called the encoder, the rotational movement of the motor can be evaluated.

The encoder functions similar to the speedometer on a bicycle. A magnet (in most cases for a bicycle located on one of the spokes) passes by a sensor (attached to the fork of the bicycle in most cases) with each revolution and the sensor generates a pulse due to this. These pulses can be counted, and, for example, multiplied with the circumference of the tire for the speedometer. In this way, the distance traveled is obtained.

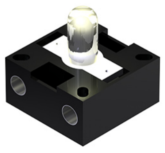
The encoders on the fischertechnik encoder motors generate three pulses with each revolution of the motor shaft. And because the encoder motors also have a gearbox with a transmission ratio of 25:1 (meaning "25 to 1"), then one revolution of the shaft, which comes out of the gearbox, corresponds to 75 pulses of the encoder.

XS Motor

The XS motor is an electric motor, which is exactly as long and high as a fischertechnik building block. In addition, it is very light. Due to this, you can install it at points, which do not have enough space for the big motors.

Both gearboxes, which are also included in the construction set, fit exactly on the XS motor.

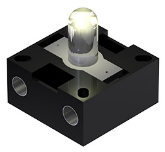
The XS motor is designed for a supply voltage of nine volts and a current consumption of a maximum of 0.3 amperes.



Bulb

Two bulbs are contained in the construction set. They can be used very multifariously, for example, as signal lights for a traffic light or as a blinking light on a robot.

The bulbs are designed for a voltage of nine volts and consume about 0.1 amperes of current.



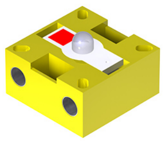
Lens Tip Lamp

A lens is a part of this lamp and this focuses the light. It looks very similar to the bulb lamp. You must be careful that you don't mistake one for the other. To make it easier to see the difference, the plug base for this lamp is gray, but the bulb lamp has a white plug base. You need the lens tip lamp to build a light barrier.

The lens tip lamp is, as are the bulbs, designed for a voltage of nine volts and consumes about 0.15 amperes of current.

Sensors

Sensors are so to speak the counterpart to the actuators. This is because they do not perform any actions, but react to certain situations and events. For example, a pushbutton, reacts to the "pushing of the button" by allowing an electric current to pass or by interrupting it. A heat sensor reacts to the temperature in its surroundings.



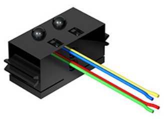
Phototransistor

You can also call the phototransistor a "brightness sensor." This is a "sensor" that reacts to brightness.

For a light barrier, this is the counterpart to the lens tip lamp. When there is a high degree of brightness, that is when the transistor receives light from the lens tip lamp, it conducts electricity. If the light beam is interrupted, the transistor does not conduct any electricity.

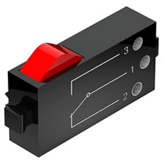
Caution!

When connecting the phototransistor to the power supply, you must pay special attention to the right polarity. The positive pole must be connected to the red marking on the phototransistor.



Trail Sensor

The infrared trail sensor is a digital sensor for the identification of a black trail on a white background with a distance of five to 30 mm. It consists of two transmission and receiver elements. For connection, you need two digital inputs and the nine volt voltage supply (positive and negative pole) on the ROBO TX Controller.



Pushbutton

The pushbutton is also called a touch sensor. When you press the red button, this mechanically throws a switch and electricity flows between the contacts 1 (center contact) and 3. At the same time, the contact between the connections 1 and 2 is interrupted. So you can use the pushbutton in two different ways:

As a "closer"

Contacts 1 and 3 are connected.

Press the pushbutton: electricity flows.

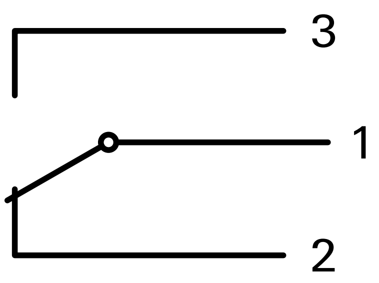
When the pushbutton is not pressed: no electricity flows.

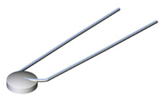
As a "break contact"

Contacts 1 and 2 are connected.

Press the pushbutton: no electricity flows.

When the pushbutton is not pressed: electricity flows.





Heat Sensor (NTC)

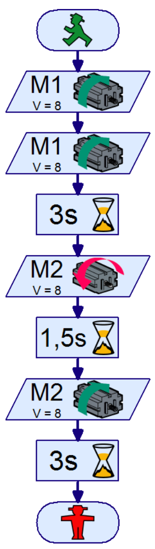
This component is a heat sensor that you can use to measure the temperatures. At 20°C, its electrical resistance is 1.5kΩ (kiloohms). NTC stands for Negative Temperature Coefficient. This simply means that the resistance value decreases when the temperature increases.

The information, which the sensors provide us, for example, bright-dark, pressed-not pressed and temperature value, can, as we will see later, be transmitted through the ROBO TX Controller to a PC and then with the help of the software, for example, be used to program a motor in such a way that a ventilating fan blows as soon as a light barrier is interrupted.

Software ROBO Pro 2.x

ROBO Pro is a graphic programing interface, with which you can create the programs for the ROBO TX Controller.

"Graphic programing interface" means that you don't have to "write" the programs line for line by hand, but that you can simply compile them visually with the help of graphic symbols. You can find an example for such a program in the picture to the right.



How you can precisely create such a program is described in detail in the ROBO Pro online help in chapters 3 and 4.

The installation of ROBO Pro and the driver for the ROBO TX Controller is described in the installation instructions, which are contained in the construction set.

The software is on the same CD as this activity booklet.

When you are finished with the installation of ROBO Pro, you can then start ROBO Pro right away because the next thing we need is the online help of the software.

The best thing for you to do now is to read through the first two chapters of the ROBO Pro online help. When you do this, you learn a little bit about the software at the same time so that following this we can start directly with the experimentation.

ROBO TX Controller

The ROBO TX Controller is the heart of this computing construction set. This is because it controls the actuators and evaluates the information from the sensors.

For this task, the ROBO TX Controller has numerous connections, to which you can connect the components. What components can be connected to what connections and what the functions of the connections are are described in the instruction manual for the ROBO TX Controller.

A special treat is the integrated Bluetooth interface. Using this, you can connect your PC to the ROBO TX Controller without a cable. Or you can also connect several Controllers with a PC and with each other.

You determine how the Controller handles the individual components and what these are to do in detail with the program, which you write in the software ROBO Pro.

Power Supply (Not Included)

Because as you well know, many of the components in the ROBO TX Training Lab need electricity to function, you naturally need a power supply as well.

The fischertechnik rechargeable battery set is best suited for this. This is not included in the ROBO TX Training Lab.

A Few Tips

Experimenting makes the most fun when the experiments also work. This is why you should follow a few basic rules when constructing the models.

- **Work carefully.**

  Take your time and look precisely at the assembly instructions for the model. If you have to look for an error later then this will take much longer.

- **Check the movement of all parts.**

  When putting models together continually check to see if the parts, which are to move, move easily.

- **Use the interface test.**

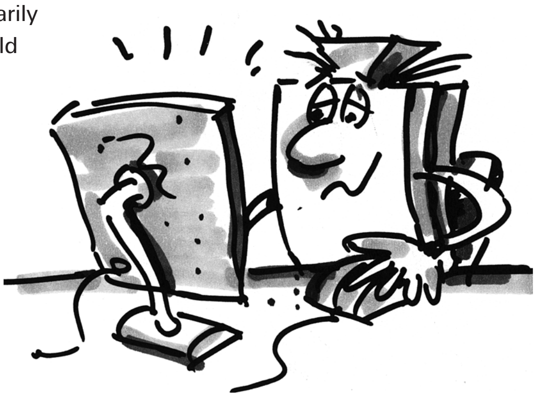
  Before you start to write a program for a model, you should test all parts, which are connected to the ROBO TX Controller, using the interface test from ROBO Pro. How this works precisely is explained in the ROBO Pro online help in chapter 2.4.

First Steps

So. After all of the preparations and information, you can now finally get started.

Since in addition to the fischertechnik components, you will primarily work with the ROBO Pro software when experimenting, you should first become familiar with it and learn how you can create a program. And because this is explained in a really excellent way in the chapters 3 and 4 of the ROBO Pro online help the best thing for you to do is to continue and carefully work through this chapter.

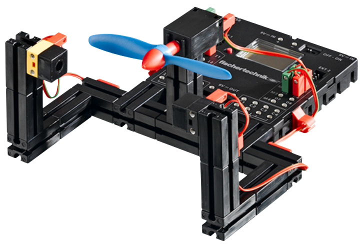
The following tip also applies here: Take your time and concentrate and then you will have that much more fun with the models.



Starter Models

After you have read through chapters 3 and 4 of the ROBO Pro online help, you can now program some models from the construction set. So, that's why we want to start right away. When you have completed building a model and connected the wires, always make a check using the interface test to determine if all outputs and inputs are correctly connected to the ROBO TX Controller and sensors, motors and indicator lights work properly.

The Hand Dryer



In your school in the bathroom, new hand dryers were installed beside the wash basins. These have a light barrier, which allows you to turn on and shut off the blower.

- First, build the model as described in the assembly instructions.

Task 1

- The hand dryer is to be programed so that as soon as the light barrier is interrupted, the blower is turned on and then after five seconds it is turned off.

Programing Tips

- First, in the program sequence, turn on the indicator light for the light barrier at output M2.
- After this, wait a second so that the phototransistor has time to react to the light. Only then will the light barrier work properly.
- Then interrogate the phototransistor at input I1. If the value is "1" (light barrier not interrupted) then the input is to be interrogated continuously in a loop.
- As soon as the value changes to "0" (light barrier interrupted), turn on the motor, M1, and then turn it off after five seconds.
- Following this, the phototransistor is to be interrogated again and so forth.

Start your program with the start button and check to see if it is working as desired. If it does work right, then you're on the best path to become a professional ROBO Pro programer.

If it doesn't work, try to find out why.

- With the interface test, you can check to see if all inputs and outputs are working and correctly connected.
- While the program is running, you can follow the program sequence with the red building blocks. This allows you to quickly see where an error has crept in.
- Finally, you can compare your program with the finished sample program, which you can call up with the symbol on the right.

After you have taken this hurdle, we want to change the task a bit:

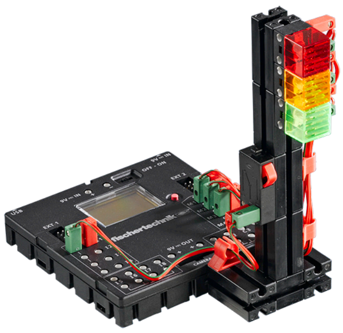
Task 2

- The school principal, who is always interested in saving energy, doesn't like it if the hand dryer continues to run for a time although your hands are already dry. He asks you to structure the program so that the blower is shut off as soon as your hands are pulled back. That's not a problem for you, or?

Programing Tips

- As in the first program, you interrogate the phototransistor I1 with a branch. If the value is "0," then turn the motor, M1, on and when the value is "1" then turn the motor, M1, off and so forth.
- For this task, there is also a complete program just in case:

The Traffic Light

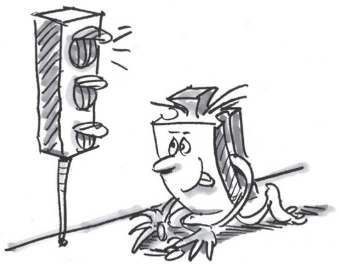


A traffic light was put up in front of your house. Since the electrician from the traffic light company is under a lot of time pressure, you offer to do the programing for the control of the traffic light for him.

The man explains to you how the control is supposed to work. But first, build the model.

Task 1

- First, the traffic light is to be green. If the pushbutton, I1, is pressed by a pedestrian then the traffic light is to change to yellow three seconds later and after an additional four seconds to red. The red phase is to last 10 seconds and the following red-yellow phase three seconds. Then it is to turn green again.



Programing Tips

- The various indicator lights belong to the following interface outputs:
  - Red – M1
  - Yellow – M2
  - Green – M3
- Turn the indicator lights on and off one after the other so that the desired sequence is achieved.
- You can load the finished program by clicking on the picture on the right.

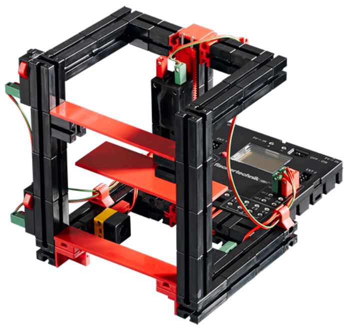
Task 2

- On the next day, the electrician from the traffic light company calls you up. He forgot to tell you that in the switch box on the sidewalk, there is a switch, I2, which switches the traffic light to blinking yellow when it is activated. You promise the electrician that you will integrate this function into your program quickly.

Programing Tips

- Attach a second pushbutton to your traffic light model and connect it to the input, I2.
- Using an additional branch, interrogate the input, I2. If the pushbutton, I2, is pressed, the sequence branches to the blinking light. Otherwise, the control of the traffic light runs as in task 1.
- You get the blinking light by turning the indicator light, M2, on and off with a time interval of 0.5 seconds. Use a subprogram to do this. You can find out how to make a subprogram in chapter 4 of the ROBO Pro online help.
- You can find the sample program as usual by clicking on the symbol. But before you look, try to find the solution on your own. Good Luck!

The Elevator



Your neighbor has installed a freight elevator in his shop so that he doesn't have to carry his heavy containers up the steps to the second floor anymore. Now, the only thing he needs is a control, which you will naturally gladly program for him.

Task 1

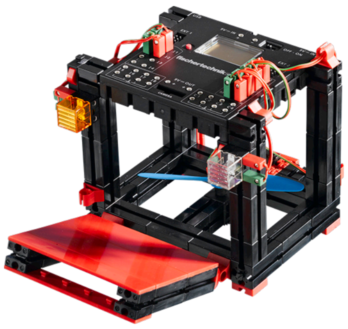
- Program the elevator so that at the beginning it moves down to its starting position. In this position, the light barrier at I3 is interrupted. When one of the two pushbuttons (I1 on the ground floor or I2 on the second floor) is pressed, the elevator is to travel to the other floor.

Programing Tips

- When the light barrier is interrupted, the elevator is located at the bottom and when it is not interrupted then you assume that the elevator is located on the second floor.
- You control how far it travels from the bottom upwards with the time, during which the motor is turned on.
- Although you can certainly do this without help, we again have a suggested solution just in case.

Before you try the next programing tasks, you should open the ROBO Pro online help. Work through chapter 5 carefully there. Switch to level 3 in ROBO Pro. The programing tasks are slowly becoming more demanding. We will use analog inputs, operational controls, operators and variables. However, if you read the ROBO Pro online help attentively, you will find it easier to use these later.

The Rinsing-Drying Machine



The next thing you want to do is to try programing a rinsing-drying machine. The rinsing-drying machine is to have the following functions:

- Pushbutton for turning on and off (I1).
- Pushbutton, which detects if the door is closed (I2).
- Rinsing function (propeller on M1)
- Drying function (red indicator light on M2)
- Display that the machine is turned on (orange indicator light on M3).
- Display, which shows the operating status of the machine (transparent indicator light on M4).
  - Blinking fast: machine rinsing
  - Blinking slowly: machine drying
  - Steady light: Machine is finished.

Task 1

- Create a rinsing program, which first starts when the door is closed and the start button, I1, is pushed. First rinsing is to be done and then drying. The operating status is to be shown by the two indicator lights on M3 and M4.

Programing Tips

- Here, is our suggested solution.

Task 2

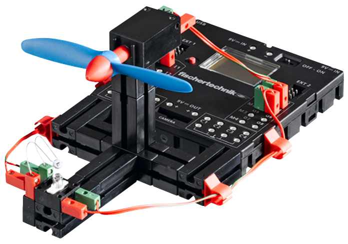
- If the door is opened, the rinsing operation is to be interrupted. After the door is closed, the program continues at that point, at which it was interrupted.
- In addition, show the operating status of the rinsing-drying machine on the display of the ROBO TX Controller.

Programing Tips

- Admittedly, this task is a bit difficult. If you can solve it on your own, great! But if you can't solve it even after thinking a lot, no problem. Just take a look at our suggested solution.

Temperature Control

At home where you live, a new air conditioning system was installed. Of course, you immediately asked the plumber how the temperature control works. He was happy to explain to you that a temperature probe continually measures the temperature. As soon as a maximum value is exceeded, the air conditioning is turned on. However, on the other hand, if the temperature is below the minimum value, the air conditioning is turned off and the heating is turned on. Now, using the "temperature control" model, you also want to try to program such a control circuit. But first, build the model.



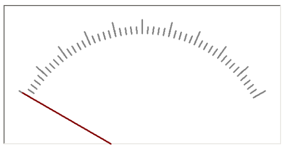
Task 1

The heating is simulated by the lens tip lamp, M2. The blower on output, M1, serves as the "cooling unit." To measure the temperature we will use the NTC resistor on the input, I2.

- Program the model so that above a certain temperature value the heating is shut off and the blower is turned on. This is to cool until a minimum value is reached. Then the blower is to be shut off and the heating is to be turned on.
- The current value of the analog input is to be shown on a measuring device and a text display in ROBO Pro and on the display of the ROBO TX Controller.

Programing Tips

- **Please note!** The resistance value of the NTC resistor decreases with increasing temperature. Therefore, the maximum temperature value is the smallest value for I2. At this limit value, the blower is to be turned on. The lower temperature limit is the biggest value for I2. At this limit value, the heating is to be turned on.
- With the interface test, you can find out what value I2 has at room temperature. Turn on the indicator light M2 and observe how much the value decreases. Now turn on the blower and find out how much the value increases. Based on this, select the limit values for heating and cooling.
- Display the value of the analog input in your program with the display of a text and/or with a measuring device (see ROBO Pro online help chapter 8.1).

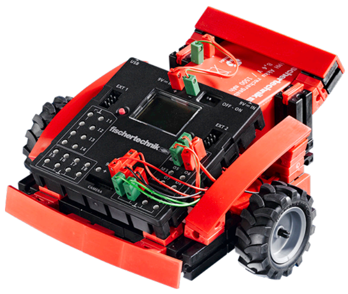


```
Temp.=       0
```

- By clicking on the symbol on the right, you can open the finished program.

Robots – The Next Challenge

Basic Model



With this model, we want to find out together how you can control a moving robot. How do you get it to move, how does the steering work and can you perhaps improve its precision? We will answer these questions with the help of the tasks in this chapter.

But first of course, you have to put the robot together. As always, you will find the description in the assembly instructions.

- Put the robot together as is described in the assembly instructions.
- Take your time putting it together. Look closely at the drawings in the assembly instructions and the wiring as well. If you connect the components differently with the ROBO TX Controller than is described in the assembly instructions then the robot may possibly behave differently than you expect.
- After you have put it together, check all of the components, which are connected to the ROBO TX Controller, using the interface test of the ROBO Pro software. If you allow the motors to turn counterclockwise then the robot must move forwards.

Task 1: Just Straight Ahead (Level 1)

- Let the robot move three seconds straight ahead (**not on the table, danger of falling off the table!**) and then move straight back for three seconds.
  - *Did the robot really come back to its starting point?*
- Repeat the program several times and observe if the robot really moves precisely straight ahead and back.

Programing Tips

- Even if the task is certainly no problem for you, here is our suggestion:

The Steering

Even if it is fun to watch the robot move straight ahead, it is still somewhat monotonous. That's why, it is now to learn to move in a curve. And how is that done? Very simple:

Task 2: Let's Make a Curve. (Level 1)

- Let the robot move three seconds straight ahead (both motors turning at the same speed) then change the direction of rotation of the right motor (M1) for one second and then let the robot move straight ahead for three seconds (meaning both motors at the same speed in the same direction).
- Find out how long you have to let the motors run in different directions so that the robot turns 90°.

Programing Tips

- For this, change the waiting time after the control, in which the direction of the second motor is changed.
- You can open the finished program as usual by clicking on the symbol on the right.

Task 3: Moving in a Figure (Level 2)

- Now that you know how long you must reverse the direction of rotation of a motor so that the robot turns to the right or left, program the robot so that it moves in a rectangle and then arrives back at its starting position.
  - *Make a mark to check if the robot really moves precisely to its starting position.*

Programing Tips

- You can create a subprogram for "going around a corner." This allows the main program to remain clearer.
- You certainly already have the solution for the task in your head. But just in case, here is a suggestion from us again:

Always the same, but not exactly?

As you have certainly noticed, the repeating accuracy of the robot is still capable of improvement. Even if it carries out the same task several times, the result is not always the same. This is due to various reasons. One of these is that both motors do not rotate at exactly the same speed. For example, the gearbox for one motor can run with more difficulty than that of the other motor. And because both motors are operated with the same voltage (nine volts) then one motor rotates slower than the other one. Since in the past, we have controlled our robot through waiting times, perhaps one wheel has rotated farther during this time than the other one.

Thus, the solution would be to have both motors rotate at exactly the same speed. And it is precisely this that can be very simply done with the encoder motors.

Task 4: Use Encoder Motors

- Repeat the last three tasks and instead of the normal motor output and waiting time elements use the encoder motor elements. How you use these is described in the ROBO Pro online help in chapter 11.6.

Programing Tip

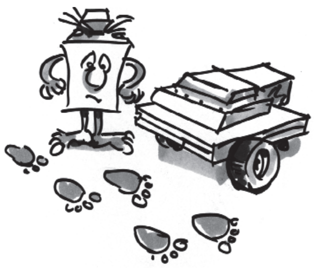
- With the encoder motor element, you can control both motors at the same time with a program item. Using the distance, you insure that each motor only rotates as far as it is supposed to.
- You can open the solution suggestions as usual by clicking on the symbols on the right.

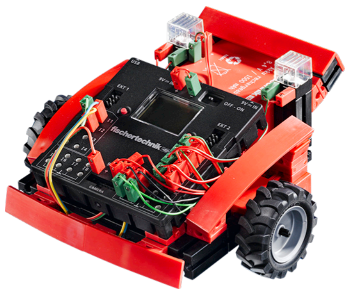
For counting the pulses at the fast counting inputs, C1-C4, you do not need an additional program item in ROBO Pro. The counting input, C1, is automatically assigned internally to the motor, M1, M2 belongs to C2 and so forth.

Note!

If your model does not move straight ahead in spite of the use of the encoder motor elements then the cause may be in the model itself. For example, if a hub nut, which transmits the force from the axle to the wheels, is not tightened enough then the axle slips and the model moves in a curve although the motors rotate at the same speed. So tighten the hub nuts very firmly.

The Trail Searcher





Your robot can now move straight ahead and turn. And in the past, it only makes this precisely as you prescribe for it with the program.

However, a robot is supposed to actually be able to react as independently as possible. That's why we now want to give it something that it can react to: a black line as a marking on the floor.

The goal is that the robot searches for the black line and moves along it.

But one thing at a time. First, you must modify the basic model so it becomes the trail searcher. How this is done is found naturally in the assembly instructions.

After you have finished modifying the model, you should use the interface test to check if all components are correctly connected to the ROBO TX Controller and function properly. You can test the trail sensor by holding it over the black trail of the course and moving it sideways. The signals at the inputs, to which the trail sensor is connected, should change due to this.

Don't forget to set the inputs in the interface test to "digital 10V (trail sensor)."

Task 1: Detecting a Trail (Level 2)

- Program the robot so that it follows a straight black trail when it is placed on this. If it loses the trail or it ends, the robot is to stop and both indicator lights are to blink three times. For this task, use the course 1a from the construction set.

Programing Tips

- First, interrogate both inputs of the trail sensor. If both inputs receive the signal "0" the robot is standing on the black trail. You can let it start.
- Put the blink function in a subprogram.
- For moving straight ahead, use the encoder motor element again, but without entering any distance.
- You can find the finished program here:

Your robot can now react. However, the function is rather limited. It would be better if the robot would correct its direction to follow the trail instead of stopping.

Task 2: Follow the Trail (Level 2)

- Expand your program with the function, which correspondingly corrects the direction of the robot when it leaves the trail and allows it to follow the trail. Try this task first with course 1a then with course 1b.

Programing Tips

- There are several possibilities for correcting the direction. You can stop one motor and let the other one continue to run or have one motor rotate opposite to the direction of movement. Experiment to determine which method is better suited.
- Here, is our suggested solution:

So! Now the robot can move on an optical "rail." The disadvantage is only that you first have to set it on the rail. We want to change this. The robot is now supposed to search for its trail on its own.

Task 3: Finding the Trail and then Following It (Level 2)

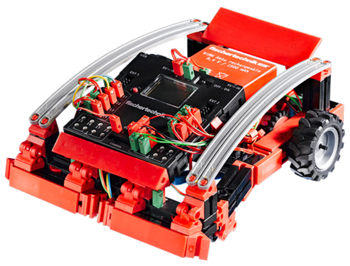
- Write a subprogram "search," which allows the robot to search for a trail if it does not find one when the program starts. For this purpose, the robot is to first move in a circle for one time. If it does not find a trail when it does this then it is to move straight ahead for a short distance. As soon as the robot detects a trail, it is to follow this. Otherwise, the search is to start from the beginning again. After it has moved in a circle 10 times without finding a trail, it is to stop and blink three times.

Programing Tips

- In case you are not exactly on the right trail, here is our suggested solution:

The Lawn Mower

Robots can mow lawns? Well of course. You only have to tell them how they are to handle obstacles and where the lawn stops. And then you can leave the work to the robot and spend the afternoon in the outside swimming pool.



However, before this is possible, you must put the lawn mower robot together according to the assembly instructions. After this, you should make a check with the interface test to determine if everything works like it is supposed to.

Then you can get started with the programing. The lawn is symbolized by the white area of our soccer stadium course 1b, which is bordered by a black edge. The lawn mower is not to pass over this edge, because otherwise, for example, it will end up on the synthetic running track. If an obstacle appears on the lawn, the robot is to avoid it when it runs into it. In addition, the cutter bar is to be turned off when an obstacle is detected.

Task 1: Detecting and Avoiding Borders and Obstacles (Level 2)

- Program the lawn mower so that it moves straight ahead from its starting position (within the boundary) until it hits an obstacle or reaches the boundary for the lawn (black line).
- If the lawn mower hits an obstacle (front bumpers), it is to stop immediately, stop the cutter bar, move back a bit, rotate to the left and then move forward and turn the cutter bar back on. Put this function in the subprogram "avoid."
- If the lawn mower hits the lawn boundary, it is also to stop immediately and start the subprogram "avoid."

Programing Tips

- Our suggested solution:

Depending on the size of your "lawn," the following problem could arise: Depending on how you set the duration of the "turn," the robot will move only along the edge or will possibly always remain in the same area of the "lawn." That's why, the robot is to always behave a bit differently when making an avoidance move.

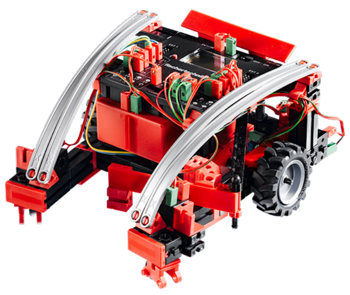
Task 2: The Random Occurrence (Level 3)

- Change the program for the lawn mower so that the lawn mower takes a different angle for rotation every time it takes an avoidance action. Thus, it is to rotate sometimes more and sometimes less. In addition, it is to go to the left to avoid an obstacle, which it detects with its right bumper and conversely for the left bumper.

Programing Tips

- In order to be able to use variables in ROBO Pro, you must set the software to "level 3."
- For this, you need a random number generator. You can create this, for example, by always increasing the value of a variable using a counter loop from "0" to a certain value. And in the subprogram "avoid," you set a "wait for . . ." element after the command for the rotation of the lawn mower that waits until the variable becomes "0." Since each time, the variable will probably have a different value when it is interrogated, the time until it returns to "0" will always be different. And due to this, the times for the rotation of the robot will be different.
- Admittedly, this is not exactly easy. In case you do not arrive at the solution in spite of a lot of thinking, there is of course again a suggested solution from us:

The Soccer Robot



Have you ever heard about the Robo-Cup? This is the Soccer World Cup for robots. It takes place each year in a different country. There are various leagues, in which various robot types are grouped. You can find more information on this, for example, on the Robo-Cup home page at http://www.robocup.org.

You can find a suggestion for the construction of a soccer robot in the assembly instructions. It is just as agile as our other robots, but in addition it has a light barrier to detect a ball and a "kicking mechanism." Put it together and then we will get started with the programing of it and "training" it with a few game tricks. As is usual, you should check the functions of the model first with the interface test before you start with the programing.

You can use, for example, a tennis ball for the ball (not included in the construction set). Depending on what you use for a ball, you must perhaps adjust the kicking mechanism somewhat.

Task 1: "There, he got the ball and he shoots . . ." (Level 2)

- In the first step, our electronic ball wizard is to learn to detect the ball and react to this. Program it so that it shoots the ball as soon as it is detected by the light barrier. Experiment a little with the "shooting speed." A short pause between the "detection" and "shooting" can possibly also lead to an improvement.
- Then the robot is to learn the "penalty shootout." Place a ball on the penalty spot of course 1a. Set the robot at the start of the black line. Now, it is to make a run-up on the line and as soon as it detects the ball it is to shoot it at the goal. At the end of the line, the robot is to stop and turn.

Programing Tips

- As for the hand dryer, for the light barrier you should wait a second after the lens tip lamp is turned on before you interrogate the phototransistor.
- Being a trainer is not easy. If your robot player does not listen to you, perhaps you can convince it with our program suggestion.

But since a real wizard with the ball must be able to do more than just the penalty shootout, we want to expand the capabilities of our soccer robot a bit.

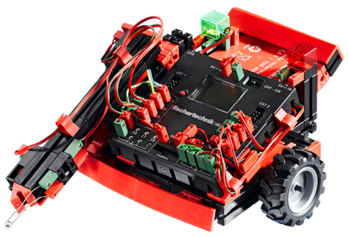
Task 2: Seek and ye shall find. (Level 2)

- The soccer robot is to now move around the stadium (course 1b) and not cross the borderline when it does this. If it finds the ball, it is to shoot the ball, of course into the goal if possible.

Programing Tips

- You have already programed most of the functions for this program for the lawn mower. Thus, you can use the lawn mower program, store it under a different name and expand it with the functions for the soccer robot.
- You probably won't need it, but in spite of this there is again a suggested solution here:

The Measuring Robot



Is it warmer under your bed than under the desk? And how hot is a candle flame? And can you cool your room with an ice cube?

You can find answers to such questions and other questions with the help of the measuring robot. It is equipped with a temperature sensor (NTC) and can measure and display the temperature at various points. The measuring robot also has a trail sensor so that you can prescribe a route for it with a black line.

First, put the measuring robot together as described in the assembly instructions and check its functions with the interface test.

Task 1: Control and Temperature Measurement (Level 3)

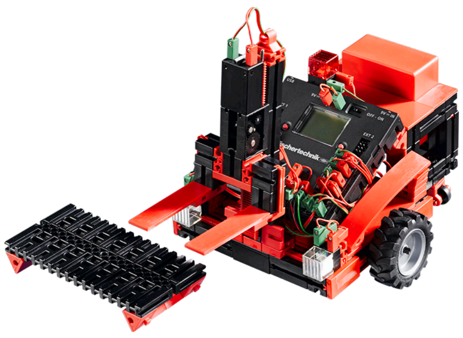
- Take the program for the trail searcher and expand it with a subprogram for the control of the measuring arm. The robot is to move on the trail of course 1b and measure the temperature at certain time intervals.

  The temperature is to be shown on the display of the ROBO TX Controller.

Programing Tips

- To display the resistance value of the temperature probe, you are to use an analog input, a control panel output and a "display" operational control.
- The resistance value of the temperature probe is displayed at the analog input, not the temperature itself. In order to convert this value into a display of the temperature, you can use a subprogram.
- You can find our complete suggested solution for this task here:

The Forklift



An area where robots are used a lot in industry is logistics. So they are used everywhere where things have to be moved from "A" to "B" as the saying goes.

You can make a great imitation of such transportation tasks with the forklift model. Put the model together as is described in the assembly instructions. Then, use the interface test to check if the components all work properly.

Task 1: Up and Down (Level 3)

- Write a subprogram for each of the functions "raise" and "lower."
- Since, when the forklift is moving, the fork is not to be too far up or down, you need a subprogram "travel position."

Programing Tips

- For "raise," the fork is to be moved up (motor M3 rotates counterclockwise) until the upper limit switch, I4, is activated.
- For "lower," the fork is to be moved downward (motor M3 rotates clockwise) until the gearbox on the lower limit switch, I3, is passed and the switch is open. For this purpose, you are to use a "wait-for" element and set it to "1 -> 0 (falling)."
- For the "travel position" the fork is to be moved upward until the fork is just above the lower limit switch.
- As always, there is a suggestion from us for the solution for the task.

Now that the mechanics are working quite well, we also want to use them. Well, a forklift is supposed to transport things and not just move its fork.

Task 2: From "A" to "B" (Level 3)

- The course 2a for the forklift is in the construction set. It is to start from the start position, pick up a pallet, which is on field "A," transport it on the trail to field "B" and drop it off there.

Programing Tips

- Program the forklift so that when the trail ends it always picks up or drops off the pallet. Use a subprogram for each of these two actions.
- Let the forklift move slowly so that it does not unintentionally lose the trail in any case. Otherwise, it thinks that it has reached the end and must pick up or drop off the pallet.
- Our suggested solution:

Task 3: Detecting Junctions

Now it is getting exciting. The forklift is to detect a junction. For this purpose, use course 2b.

- The forklift starts at its starting position. It is to move to field "A," pick up the pallet there and transport it to field "B." Then, it is to move back to its starting position. When it is following the trail, its headlights are to be turned on. If it rotates or moves backwards, the red indicator light is to be turned on.

Programing Tips

- The forklift can detect the interruption of the trail as a junction. When it reaches this point, you can decide how it is to move further. It can move straight ahead until it finds the trail, turn left and when doing this search for the trail or turn right until it finds the trail again. Create a subprogram for each of the three cases.
- In order to find the trail again after turning left or right, the robot must first move a bit straight ahead. Otherwise, it will rotate and pass by the trail.
- If you lose the trail when programing, just take a look at our suggestion:

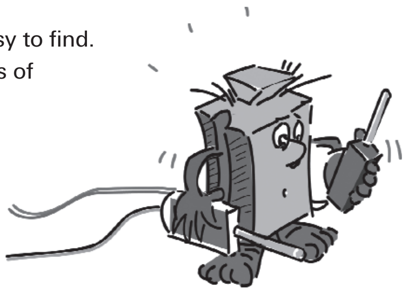
Task 4: Transporting without an End

- Expand the program from task 3 so that the forklift takes a short break after it returns to its starting position and then picks up the pallet from field "B" and brings it back to "A" and then moves to the starting position. Now, the forklift is to continually repeat the entire sequence like on an assembly line.

Programing Tips

- Generate an endless loop by making a line from the last program item to the start of the program.
- Even if this task certainly presents no problem for you, here is our suggested solution:

If it doesn't work right away . . .



. . . then this is due to a simple reason in most cases. But, this is not always easy to find. That's why, we want to give you some information here about possible sources of errors.

Interface Test

Once again here the advice: Check the functioning of the individual components with the help of the interface test in ROBO Pro.

Cables and Wiring

If an electrical component does not work at all, check the cable, with which you connected it to the ROBO TX Controller. To do this, connect the cable with the rechargeable battery and a bulb. If the bulb lights up then the cable should be okay.

Also, incorrectly installed plugs, for example, a green plug on a red cable, can be an error source.

Also check that "+" and "-" are correctly connected. For this purpose, compare your model with the pictures in the assembly instructions.

Loose Connection

A component, which works sometimes and doesn't work at other times, probably has a loose connection somewhere in the wiring for the component.

The most frequent causes for this are:

- **Loose Plugs**

  When the plugs for the cables are too loose, meaning are wobbly in the jack sockets, then they do not have sufficient contact. In this case, you can use a screwdriver to **carefully** bend the contact springs at the front on the plugs concerned apart. Just a bit, so that the plugs are firmly seated in the jack sockets when you plug them in.

- **Poor Contact between Cable and Plug**

  Also check the contact between the stripped cable ends in the plug and the plug itself. It may possibly be sufficient to tighten the screws in the plug a bit more.

Short Circuits

You have a short circuit when a positive and negative connection are touching each other. Both the rechargeable battery and the ROBO TX Controller have a fuse built in so that they are not damaged by a short circuit. They simply switch off the power supply for a while. However, your model will naturally not work anymore either.

The cause for a short circuit can be either a mistake in the wiring or screws, which have not been tightened enough, in the plugs. They can touch each other when the plugs are plugged in correspondingly and thus cause a short circuit. That's why you should always completely screw in the screws and plug in the plugs so that the screws cannot touch each other.

Power Supply

Short interruptions or motors, which run slow, indicate in most cases a discharged rechargeable battery. In this case, you should charge the rechargeable battery with the enclosed battery charger. When the red LED stops blinking on the battery charger and remains steady then the rechargeable battery is fully charged.

Errors in the Program

Even if no one likes to admit it: Everyone makes mistakes. And especially with more complex programs, an error can creep in quite quickly.

So, when you have checked everything on your model yourself and have eliminated all errors and your model doesn't do what you want it to do in spite of this then you should also check your program. Go through it piece by piece and check to see if you can find the error.

In the online mode, meaning when the ROBO TX Controller is connected with the PC, you can also follow the program on the screen while it is running. The particular active program item is highlighted so that you can always see the point where the program is and where the error occurs.

The Last Sources of Help

If in spite of all this, you have not found the error, there are still two possibilities to obtain help:

- **E-mail Help**

  You can send us an e-mail at fischertechnik and describe your problem to us.
  The e-mail address is info@fischertechnik.de.

- **Public Help**

  You can also visit us in the Internet at http://www.fischertechnik.de. There is a forum there, among other things, where you can certainly find help. In addition, you can also become a member of the fischertechnik Fan Club at no charge.

And what else is there to do?

Was that everything? No, of course not. The experiments and models, with which you have become familiar in this booklet and have tried out, are only the beginning. So to speak your "first attempts to walk" in the gigantic and exciting theme of "computing."

What we have shown you here is only a tiny part of the possibilities, which you have with the ROBO TX Controller and the fischertechnik components. And now it is your turn. You can give your fantasy free rein and simply build what you feel like.

If you don't have any idea for your own complete model then just take a look at the models in this booklet. Perhaps something will occur to you about what you would do differently for a model. Or you can change the function of a model.

For example, you could attach a pen to the forklift instead of the fork and then raise it and lower it and use it to draw on a big piece of paper when the robot moves across it. You can then not only move in figures, but also draw them. And so the forklift has been turned into a drawing machine.

And if friends of yours also have a ROBO TX Controller then it will be even more interesting. This is because, using the Bluetooth interface, not only your PC can communicate with the ROBO TX Controller, but several controllers can communicate with each other. Then, for example, all of you could program two robots, which react to one another. Or dance with each other. In chapter 7 of the ROBO Pro online help, you will find some interesting information on this topic.

Does your computer have a Bluetooth interface? If so, then you can connect the ROBO TX Controller with the computer using Bluetooth instead of a USB cable. If not, then you can buy yourself a USB Bluetooth stick and use this to connect the ROBO TX Controller without a cable with your PC. You can read how to do this in the instruction manual for the ROBO TX Controller and at http://www.fischertechnik.de.

So, what are you waiting for? Get started! Invent and experiment! And don't be bothered by little setbacks. Patience and perseverance are primary factors for experimenting. The reward after this is a functioning model.

We hope you have lots of fun trying out your own ideas.

fischertechnik GmbH · Weinhalde 14–18 · D-72178 Waldachtal
Tel.: +49 7443 12 4369 · Fax.: +49 7443 12 4591 · E-Mail: info@fischertechnik.de · http://www.fischertechnik.de
Illustrations: Bernd Skoda design · Layout: Kurt Wandelt GmbH
11/2009 · Created in Germany · Technische Änderungen vorbehalten · Subject to technical modification