



fischertechnik 

COMPUTING

Begleitheft

Activity booklet

Manual d'accompagnement

Begeleidend boekje

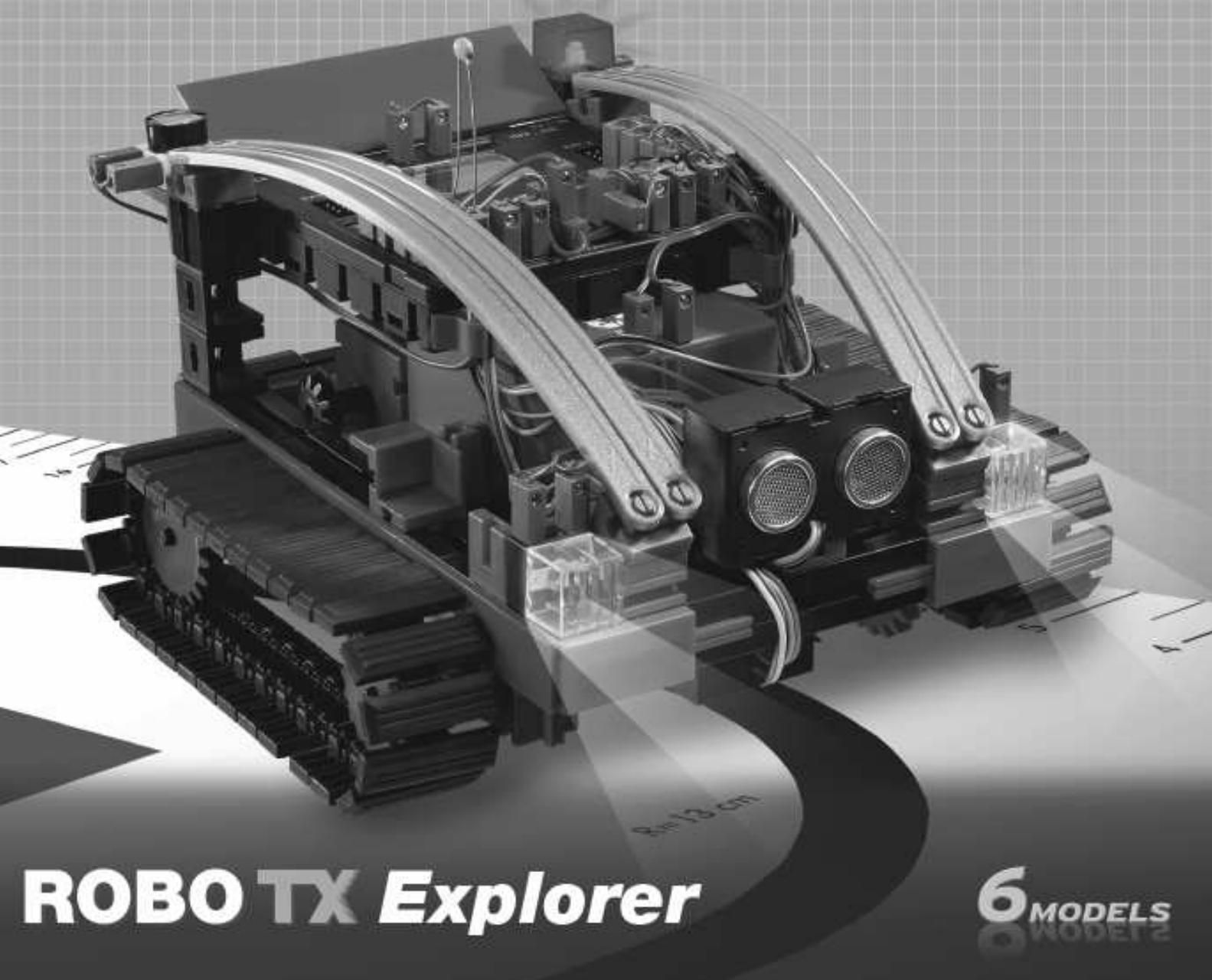
Cuaderno adjunto

Folheto

Libretto delle attività

Сопроводительная инструкция

附带说明书



ROBO TX Explorer

6 MODELS
MODELS

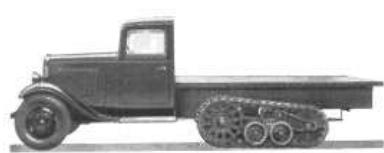
Contents



Vehicles with Crawler Drive	p. 2
The Steering	p. 2
Explorer Model from fischertechnik	p. 3
Actuators	p. 3
Sensors	p. 4
ROBO TX Controller	p. 5
Power Supply	p. 5
Software ROBO Pro	p. 5
Preliminary Considerations	p. 6
Basics of the ROBO TX Controller	p. 6
The Basic Model	p. 7
Basic Program	p. 7
Encoder Motors	p. 8
Subprograms	p. 8
Autonomous Tracked Vehicles	p. 9
Trail Searcher	p. 9
Tunnel Robot	p. 12
Color Detector	p. 13
Explorer	p. 14
RoboCupJunior – Rescue Robot	p. 16
Important Tips	p. 16

Vehicles with Crawler Drive

■ Why are autonomous vehicles with a crawler drive needed? The invention of the crawler drive for vehicles was necessary in order to be able to travel over very difficult terrain. The crawler drive was used in areas such as the desert where a wheel drive could not travel. For example, in the First World War, the first trucks and tanks with a crawler drive were built and employed.



Depending on the terrain, the vehicles could be converted from a wheel drive to crawler drive.

Tracked vehicles were also used in the civilian area. As you can see from the pictures, wheel-driven vehicles were actually always the basis for tracked vehicles.



But it didn't take long to determine a weak point: The steerable front wheels. Therefore, the crawler drive was expanded and used for all axles.

The Steering



■ But then how did the steering work? Very simple, through the slowing or speeding up of one of the two tracks. If a turn to the right was to be made, the right track was slowed down by using a control stick, one per track. When this was done, this track moved slower and thus the vehicle moved to the right.



Even today, of course with the state of the art technology, you still find many vehicles, which use the crawler drive. These range from small excavators to the giants for the open pit mining of brown coal.



In the Cheops Pyramid in Egypt, a minirobot was used to help find the trail of more secrets.



The researchers sent the robot, which was about the size of a toy locomotive, through a dark narrow shaft. It led upwards from a chamber in the center of the pyramid, which was 4500 years old, and ended in front of a mysterious stone door.

■ Explore unknown areas, measure distances, follow trails, show directions of travel by means of blinking signals, recognize colors, measure temperatures, avoid obstacles without touching them, recognize day and night, turn headlights on and off automatically and trigger an alarm etc. The ROBO TX Explorer sensors can do all of this and lots more. More specifically this includes: The NTC resistor, the photoresistor, the ultrasonic distance sensor, the optical color sensor and the specially developed infrared trail sensor. Thanks to two encoder motors and the crawler drive even almost impassable terrain can be explored and traveled on. With the rescue robot, which is contained as a model, the construction set provides the ideal basis for participation in the RoboCupJunior.

Now, before you get started, you should become familiar with the most important components. These are described in the following:

Encoder Motor

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.

However, the encoder motors can do even more than this: 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.

Thus, 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.

The encoder motors are connected to the outputs M1 to M4 on the ROBO TX Controller. The encoder signals are read in through the inputs C1 to C4.

Buzzer

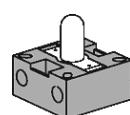
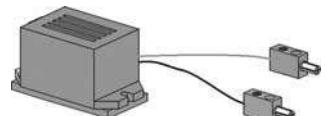
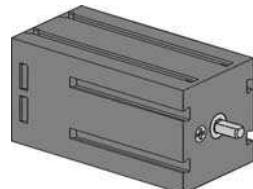
The buzzer is used, for example, to acoustically report obstacles or colors, which are identified. This is also connected to the outputs M1 to M4.

Bulb Lamp

This is a bulb for a voltage of nine volts. This can be used as a status signal for the direction of travel or simply for illumination. This is connected to the outputs M1 to M4.

Explorer Model from fischertechnik

Actuators



Sensors

■ Sensors are so to speak the counterparts to the actuators. This is because they do not perform any actions, but react to certain situations and events.

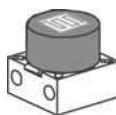
The sensors are connected to the universal inputs I1 to I8 on the ROBO TX Controller.

NTC Resistor (Negative Temperature Coefficient)



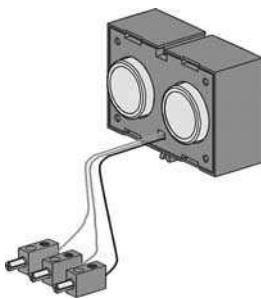
Here, this is a component, with which you can measure various temperatures. You can also call this a heat sensor. At about 20 degrees, the NTC resistor has a value of 1.5 kilohms. When the temperature increases, the resistance value decreases. This information is available in ROBO Pro as a numerical value.

Photoresistor



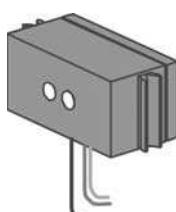
The LDR 03, which is an analog brightness sensor, reacts to daylight and changes its resistance value due to this. This is an indicator for the brightness of the light.

Ultrasonic Distance Sensor



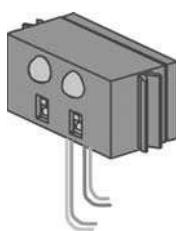
A distance sensor is a technical component, which has the capability to measure the distance between itself and an object. Distance sensors work with light, infrared radiation, radio waves and **ultrasonic sound** and use various measuring methods. Sound is propagated as a wave. An echo is reflected back to the source of the ultrasonic sound and the echo is then received as a signal and evaluated. The time difference between the transmission and reception of the signal provides information about the distance between the obstacle and the sensor. The range of the sensor is up to four meters. The numerical value, which is shown, corresponds to the distance in centimeters.

Optical Color Sensor



Color sensors are mostly used in automation technology. When this is done, for example, the color or the color imprint is to be examined to insure that the correct components are installed. The fischertechnik color sensor transmits red light, which is reflected with a different strength from different colored surfaces. The quantity of the reflected light is measured by the phototransistor and outputted as a voltage value between 0 V and 10 V. The measured value is dependent on the ambient brightness and the distance of the sensor from the colored surface. You can read in this value and process this as a numerical value from zero to 10,000 in your program.

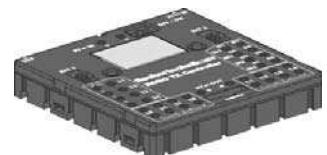
Trail Sensor



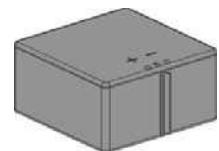
The **IR trail sensor** is a digital infrared sensor for the identification of a black trail on a white background at a distance of five to 30 mm. It consists of two transmission and two receiver elements. You need two universal inputs and the nine volt voltage output for the connection.

- The most important component for building a tracked vehicle is the ROBO TX Controller, which is installed as a fixed element in the various models. You can connect your sensors and actuators to this as needed. You can find the basic wiring in the enclosed assembly instructions.

ROBO TX Controller



- The ROBO TX Explorer models are autonomous vehicles, which move freely in the area. Therefore, you use the fischertechnik rechargeable battery set as the power supply.

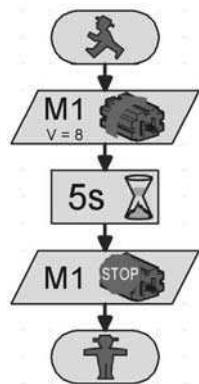


- ROBO Pro is a simple graphic programming interface, with which you can write your programs. The advantage is that you don't have to learn any programming language. Actually, you can start right now.

You need the ROBO Pro version 2.1.4.2 for the construction set, ROBO TX Explorer. If you have an older version of the software, you can update this at no charge. Download this with the help menu in ROBO Pro-Download New Version-or at:

www.fischertechnik.de/robopro/update.html.

Software ROBO Pro



Preliminary Considerations

Basics of the ROBO TX Controller

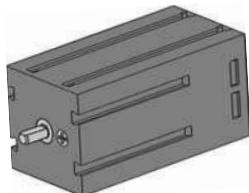
As for all other fischertechnik robots, with the ROBO TX Explorer, you will be introduced step-by-step to the fascination of technology and programing. You start with a simple model and work your way forwards to increasingly more comprehensive systems with fascinating possibilities. The primary focus for all models is the precise construction and precise placing in operation.

Before you start with the individual models, you should become familiar with the Controller with the help of some experiments. If you have problems, you can also find information in the ROBO Pro help.

After you have installed the software, you can connect the Controller using the cable, which is supplied, to your PC. Now, start the ROBO Pro program and open the window for the interface test with the Test button.



Encoder Motor



Connect the connections of the encoder motor with the connection M1. Click with the mouse on the selection „Counterclockwise“ or „Clockwise.“ The motor starts with its maximum speed. By operating the control, you can adjust the RPMs. Use „Stop“ to stop the action.



Photoresistor



Connect the photoresistor, which is provided, to the connection I1 and set the input to „Analog 5 kOhm (NTC, . . .).“

Change the luminous intensity of the photoresistor by slowly covering with a black strip of paper. What happens? You will see that this changes the numerical value of the input.

The interface test is explained very well in chapter 2 of the ROBO Pro help. You can also find help if there are problems between your computer and the Controller and the software – take a look, it's worth it!

■ Now you can get started. Now that you are familiar with the ROBO TX Controller and the programming, you can now solve the first task. First, the basic model is to be built using the assembly instructions.



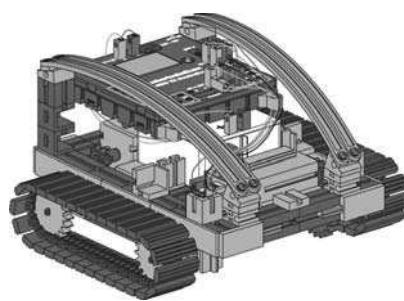
Task 1 – ROBO Pro Level 1

Your tracked vehicle is to travel six seconds straight ahead and then it is to turn to the right for three seconds and then stop.

We want to give you some support with your first program. First, click on the button „New File.” Your program starts with a little green traffic light figure for program start.



The Basic Model



After this, you need two motor symbols. Place the first symbol under program start so that the connection is automatically drawn. Move the mouse to the motor symbol and turn on the properties window (right mouse button). There, you set the motor output to „M1” and for action the direction of rotation „Counterclockwise.” Then confirm with OK. Insert the second motor symbol in the same way and repeat the action for the motor output „M2.”

Next, the program is to wait for a certain time. For this, use the waiting time symbol. You place this under the second motor and adjust the time to six seconds.

Following this, the tracked vehicle is to turn for three seconds. To do this, you again insert two motor symbols for M1 and M2. M1 is to rotate counterclockwise and M2 clockwise. Since both motors are to work for three seconds, now insert the waiting time symbol and set the time value to three seconds.

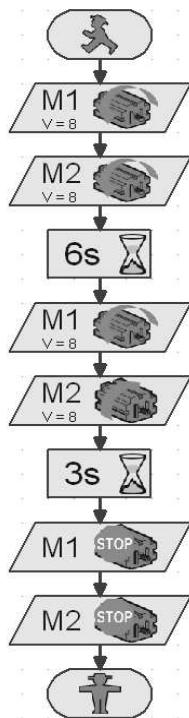
After this, you have to stop both motors. This is done as before the rotation by inserting both motor symbols and the parameter setting „Stop.”

At the end, you must then insert the symbol for the end of the program, the „little red traffic light figure.” Now, your first program is finished and you can store it. Then test it in the online mode. To do this, click on the „Start” button.

If you did everything right, then you can load the program onto the Controller. To do this, click on the „Download” button. Accept the setting of the download window. Right after the download, the model starts. Unfortunately, it is still connected with the USB cable. Load the program once again, but activate the „Start program using button on Interface.” Once the program has been transferred, you can pull the cable out. To start the program, press the left selection button on the Controller.

You can find the finished program at:

C:\Programs\ROBOPro\Sample Programs\ROBO TX Explorer\Basic_Model_1_TX.rpp



Start program in online mode.

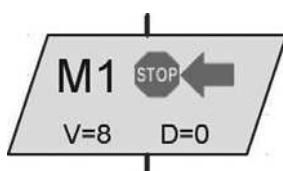


Download program to ROBO TX Controller.

Encoder Motors

As you have certainly noticed, your model does not travel exactly straight ahead. 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 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.



Encoder motor element

Task 2 – ROBO Pro Level 1

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

Find out how long you have to let the motors run in different directions so that the robot turns 90°. For this purpose, change the distance information in the encoder motor element, with which the robot is turned.



You can find the finished program at:

C:\Programs\ROBOPro\Sample Programs\ROBO TX Explorer\Basic_Model_2_TX.rpp

Subprograms



Generate new subprogram.



Copy current subprogram.



Delete current subprogram.

For the solution to the next task, subprograms are needed. For this, read chapter 4.1 in the ROBO Pro help. It is important that you switch to **level 2** in ROBO Pro.

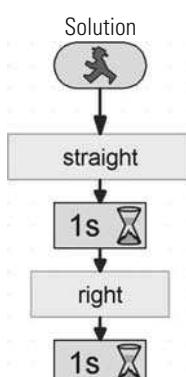


Task 3 – ROBO Pro Level 2

Your tracked vehicle is to move around a square. Use the same parameters as in the second task. Make a subprogram for each direction of travel.

First, make the subprogram „straight“ (see ROBO Pro help chapter 4). Mark the program parts and copy them to the cache. Following this, prepare the subprogram „left“ and „right“. Insert the program parts for „straight“ into both from the cache and change the parameters accordingly.

As a little bit of help, we show you a part of the task. The following table is to show you at a glance how you have to program the motors for the directions of travel.



Direction of Travel	Direction of Rotation Motor 1	Direction of Rotation Motor 2
Forwards	Counterclockwise	Counterclockwise
Backwards	Clockwise	Clockwise
Right	Counterclockwise	Clockwise
Left	Clockwise	Counterclockwise
Stop	Stop	Stop

Using this table, all motors in the sample programs can be programmed.

Finished Program:

C:\Programs\ROBOPro\Sample Programs\ROBO TX Explorer\Basic_Model_3_TX.rpp

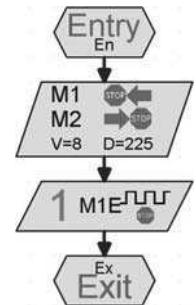
■ After you have experimented with the basic model enough, your robot is now to react to various signals from an external source.

In order to allow your tracked vehicle to identify its environment and to complete certain tasks, you must equip it with sensors. The following model suggestions show you different variations of tracked vehicles with various sensors. This is to provide the capability to identify different distances traveled, light and colors or even heat sources or distances. You can find the individual programs in the directory: C:\Programs\ROBOPro\Sample Programs\ROBO TX Explorer\

Of course you have seen factory buildings, which have no human beings in them, in TV movies where the transport vehicles in the buildings travel as if steered by an invisible hand. In some cases, such systems are controlled by data lines, which are embedded in the floor, or by markings on the floor to indicate the path.

The basis of your programming is to be that the robot travels along a black line.

Before you start with the programming, first build the trail searcher using the assembly instructions. You can find a demonstration course with an imprinted black line in the construction set. The line, which is to be followed by the trail searcher, is to be straight at first.



Subprogram „right“

Autonomous Tracked Vehicles

Trail Searcher





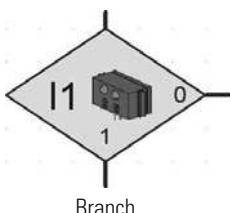
■ Now, how is the model to function?

The robot is to find a black line on a white background and then follow this line. To make this possible, you have the IR trail sensor installed in your model. The module transmits a light in the infrared range to the pavement substrate. Depending on the substrate, this is reflected and measured by phototransistors. For your programming, this means: A bright and/or white substrate reflects the light and you receive the value 1. With a black substrate, the light is not reflected and you receive the value 0. If both transistors have the value 0, then your robot has found the road (the black line) and must now follow this road.



Task 1 – ROBO Pro Level 2

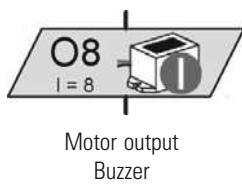
Your tracked vehicle is to be placed on a straight black trail and then travel along this. If it loses the trail or it ends, it is to stop and honk three times.



A Few Little Tips

Use the interface test to check the trail identification by the sensor. Don't forget to set the inputs to „Digital 10 V (Trail sensor).“ If the black-white identification does not function properly, then interfering light sources such as the sun could be the cause. If necessary, the sensor must be positioned somewhat closer to the trail or be shielded with a building board.

Finished Program: **Trail_searcher_1_TX.rpp**

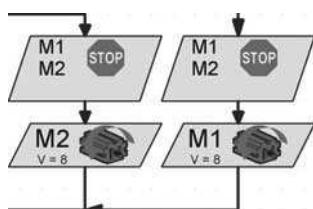


■ Of course, you will not be satisfied with your first solution because your robot can certainly travel a short stretch along the line. However, because it cannot yet readjust, it leaves the marking and stops and sends you a signal indicating this.



Task 2 – ROBO Pro Level 2

Expand your main program in the interrogation branches of the trail sensor so that the robot recognizes when it is not traveling exactly on the trail. Then, it is to correct its direction of travel accordingly. You can find a tip in the left program section.



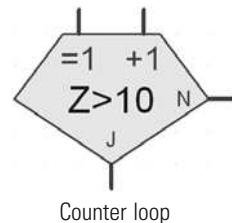
Now it looks better already. Your robot remains precisely on the prescribed trail. In an industrial building, other robots would now remove the load, which was transported, at the end of the trail or would give the robot a new load. This load could then be transported back to the starting point of the trip.

Finished Program: **Trail_searcher_2_TX.rpp**



Task 3 – ROBO Pro Level 2

Your robot traveled along a certain black line in the previous tasks. For this task, it is to search for a line. To do this, it is to turn in a circle one time. If it doesn't find a trail when it does this, it is to travel a short distance straight ahead and then search again. If your robot finds a trail, then it is to follow the trail. If the trail ends or it loses the trail, then it is to start the search again. After it has moved in a circle 10 times without finding a trail, it is to stop and honk three times.



Tip

Think about the first tasks in the the basic model. Here, the robot was supposed to turn 90 degrees. You did this with an encoder motor element. This technology can be of use to you here as well.

For the trail search, prepare your own subprogram with the name „track-seeking”. In the diagram, you see how we suggest this be done.

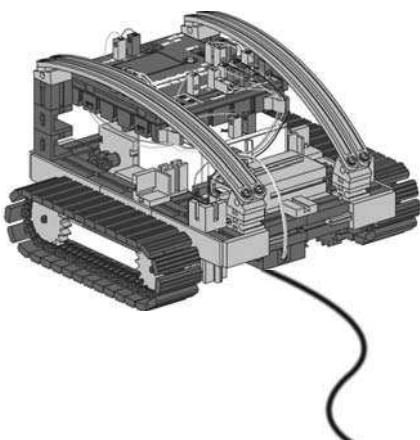
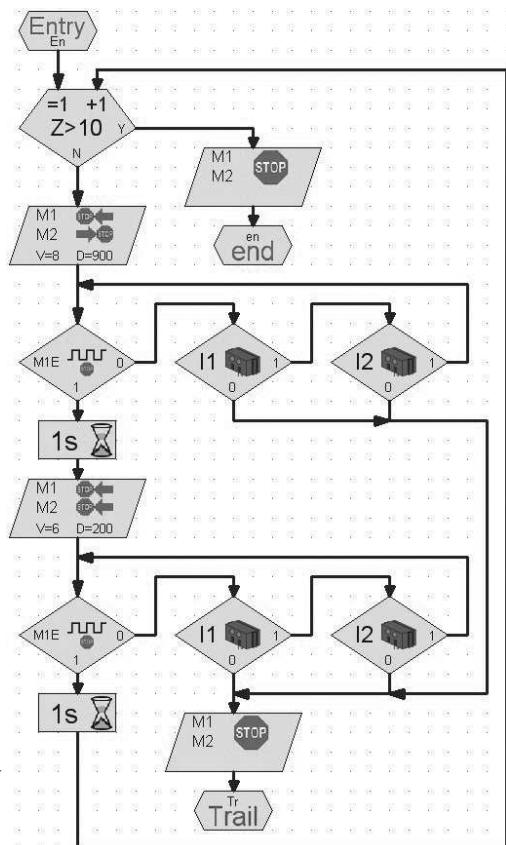
Finished Program: [Trail_searcher_3_TX.rpp](#)

The previous routes were always straight. However, routes sometimes also have curves. You can find these, for example, in industrial facilities, in which material or workpieces must be transported from one machine to the next one.

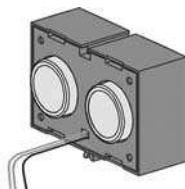
Task 4 – ROBO Pro Level 2

The demonstration course contains various curves with differing radiuses.

Also experiment with various speeds for M1 and M2 for your circuit. What settings allow the robot to complete the course the fastest? Record the results in a small table.



Tunnel Robot, Distance and Temperature Sensor



■ The sensor for distance measurement and the NTC resistor offer you additional possibilities to expand your model to a professional robot.

Measuring distances and temperatures and possibly initiating rescue measures. What do you think, where could robots with such capabilities be used? Of course, you can think of various areas, where they could be used. We want to examine the area of fire protection and fire fighting in auto and train tunnels.



The task for such a robot is to cautiously approach the source of the fire, measure temperatures in the tunnel and to report data to a control center. In most cases, the robots are equipped with mobile extinguishing equipment, which can be used depending on the conditions.

Now here as well conscientiously build the model, „tunnel robot,“ according to the assembly instructions.

Task 1- ROBO Pro Level 2

Similar to the trail searcher, which travels along a line, your robot is to travel a certain route along a wall at a certain distance of about 20 centimeters.



Finished Program: **Tunnel_1_TX.rpp**

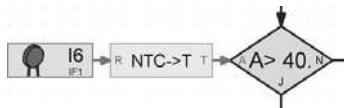
■ Let's examine a fire extinguisher robot once again for the next task. In order that it can travel along the wall like your robot, it uses distance sensors. However, in order to identify the source of the fire, it employs heat sensors. This heat sensor is the NTC resistor for your model. The physical characteristic of this component is that the resistance value decreases when the temperature increases. You can try out this change again with the interface test. Connect the NTC resistor to the connection I6. Place a heat source near the NTC and observe the numerical value of the input. Don't forget to set the input to „Analog 5 kOhm (NTC, . . .).“

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 temperature, you can use the subprogram „NTC->T“ (see Tunnel_2_TX.rpp).

Task 2 – ROBO Pro Level 2

Expand the program, which allows the robot to travel along the tunnel wall. In addition, measure the current temperature. If this increases to a certain value then your robot is to stop and transmit a warning signal using the buzzer. At the same time as the buzzer, the red indicator light is to blink.

Following this simulated extinguishing action, your robot is to turn and return to the starting point.



Finished Program: **Tunnel_2_TX.rpp**

Tip

Since your robot only has one distance sensor, it needs a second wall, along which it can travel, to return to the starting point.

If you have another motor and a propeller in your fischertechnik collection, you can also integrate the extinguishing of the fire into your program.

■ You will get to know the color sensor as an additional sensor. The picture on the side gives you an example for industrial use. As you can see, colored cans, which were placed in the wrong row, are being sorted out.

The light, which is reflected from the material being examined, is received, digitalized and processed using a computer and software. The task of the sensor is to identify the various colors and to transmit the measurement data to the ROBO TX Controller.

The color sensor is installed in the color detector model. This is connected with the black wire to I4, the red wire to plus (+) and the green wire to the ground (⊥). For the first test programs, you are to use the colored surfaces imprinted on the course.

Task 1 – ROBO Pro Level 2

First, check the values, which the interface puts out for the various colors in the interface test. Use black and white in addition to the three colors prescribed.

Make a small table and enter the values, which you have measured. Also observe the changes when the distance to the colored surface or the ambient light changes.

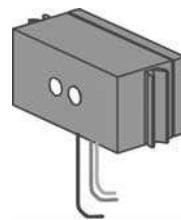
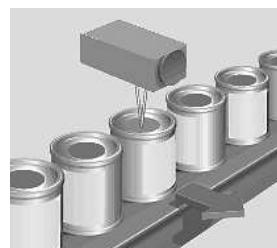
Task 2 – ROBO Pro Level 2

Write a small program, with which the sensor can identify the green colored area. If the measured value is within the prescribed range of values, the buzzer is to be activated for one second. Following this, the program jumps to start.

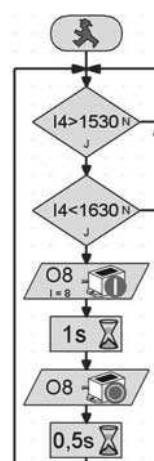
Finished Program: **Color_detector_2_TX.rpp**

Tip

For the next task, you need the three indicator lights with different colored luminous caps, which are already installed in the model.

Color Detector

Color	Value
White	
Black	
Blue	
Red	
Green	





Task 3 – ROBO Pro Level 3

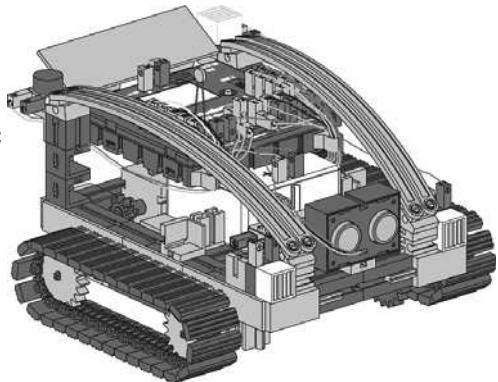
Write a program, which allows your robot to travel straight ahead for a certain distance. There are three colored areas on this route. If the sensor detects a color, the robot is to stop for three seconds. During this time, it turns on the indicator lights with the corresponding color and emits an acoustic signal with the buzzer. Following this, it travels to the next area and repeats its work. Following this, it travels to the last area, reports the result and stops there.

Finished Program: **Color_detector_3_TX.rpp**

Explorer Model

■ The „Explorer“ model contains all actuators and sensors, which are needed for an autonomous robotic vehicle.

Now, you have no limits for the solution of simple and difficult tasks. In the preceding construction stages, you have only used one sensor in most cases to become familiar with the possible areas of employment.



Task 1 – ROBO Pro Level 2

Program your robot so that it moves towards an existing obstacle on its trip. At a distance of about 60 centimeters, it is to reduce its speed. At a distance of 40 centimeters, it is to stop. If the obstacle moves closer to your robot, the robot is to move backwards slowly at a distance of 20 centimeters and at 10 centimeters, it is to move backwards fast.

Finished Program: **Explorer_1_TX.rpp**

Task 2 – ROBO Pro Level 2

Now, your robot is going on its exploration trip. Prepare a program for the use of two sensors: trail sensor and distance sensor. First, the robot is to follow the black line on the demonstration course. Place an obstacle on the route. The robot is to stop about 10 centimeters in front of the obstacle and travel one centimeter backwards. Following this, it is to turn around and follow the trail in the other direction.



Finished Program: **Explorer_2_TX.rpp**



Task 3 – ROBO Pro Level 2

The program from task 2 is to be expanded with three sensors: the color detector, the temperature sensor and the photoresistor for the measurement of brightness.

There are various colored areas along the trail. The robot reports these with various acoustic signals. If the ambient temperature becomes too high during the trip, the red indicator light is to blink. As soon as the room becomes dark, your robot turns on its two headlights. If it gets bright again, the headlights are turned off.

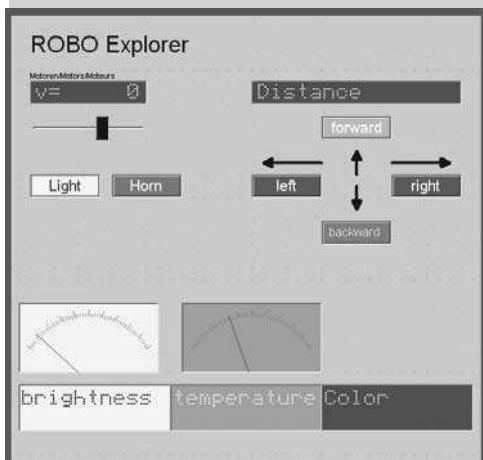
Finished Program: **Explorer_3_TX.rpp**

- Our Explorer can also be programed as a remote controlled robot to explore unknown worlds. To do this, the ROBO TX Controller is connected with the computer using the Bluetooth radio interface.



Task 4 – ROBO Pro Level 3

For this task, your computer is the control center for a Mars expedition. The mission is to transmit the measured values of a Martian landscape to the ground station. The control of your robot is generated in ROBO Pro in the operator's console (see ROBO Pro help chapter 9).



Your robot is to be programed so that it transmits measured values for the ground color, temperature, brightness and obstacles. The robot is to be controlled by hand using the operator's console of the main program in ROBO Pro.

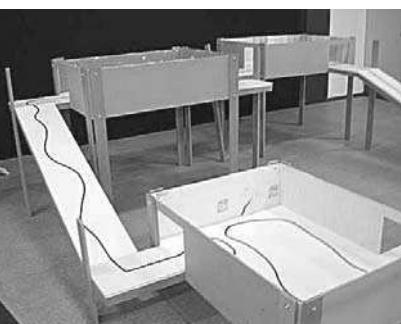
Finished Program: **Explorer_4_TX.rpp**

Mars – Start to the Fourth Planet



RoboCup Junior

Rescue Robot



Now, you have performed all tasks, become familiar with the construction of a robot and the programing technology, so now you could harvest the fruits of your labor and participate in the Rescue League of the RoboCupJunior with the „rescue robot“ model.

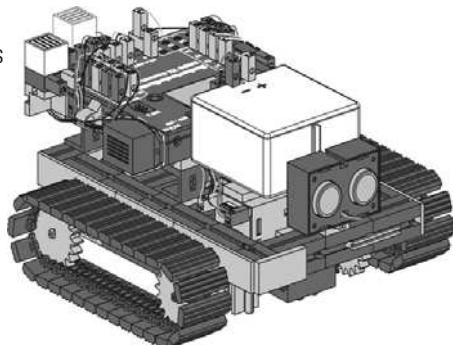
■ RoboCupJunior is a worldwide, project oriented education initiative, which promotes regional, national and international robot events for young people. The goal is to introduce children and young people to robots and their application.

Using the tasks, which you can obtain from the Internet at

<http://rcj.robocup.org>

it will certainly be easy for you to program your rescue robot for this competition.

The picture on the left shows you a course, which your robot must traverse. On this route, it must complete various tasks, for example, traveling along a line, searching for figures of various colors on the floor or travel through a door and so forth. Wouldn't that be something for you?



Important Tips

■ The fun with the subject of robotics can be lost very fast if the robot doesn't function like you want it to function.

Often, simple means can be used to identify and eliminate errors.

Cables

Here, you must proceed precisely. First, the cables are cut to the prescribed length and then the ends are to be stripped of insulation and connected firmly with the plugs. Check for proper functioning using a luminous stone (38216), with ball plug-in light (37869) and the rechargeable battery pack.

Power Supply

Often, a rechargeable battery, which is almost totally discharged, is the cause of erroneous behavior by your ROBO TX Explorer. If the voltage drops below five volts, the ROBO TX Controller shuts off automatically. Improper functioning can also occur even if the rechargeable battery is not so low. But then the rechargeable battery must still be charged.



Programming

When all mechanical problems have been solved and the robot is still not quite right, this is often due to errors in programing. Here, ROBO Pro offers you the online mode, which allows you to follow the program flow on the monitor. In most cases here, you can find the small program errors, which have crept in.