

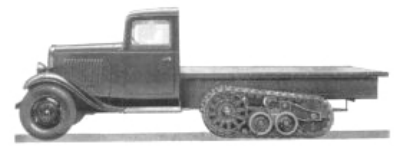
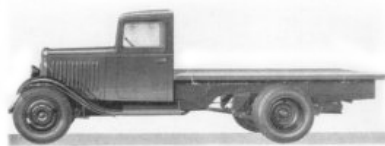
Contents



Vehicles with Crawler Drive	p. 18
The Steering	p. 18
Explorer Model from fischertechnik	p. 19
Actors	p. 19
Sensors	p. 19
ROBO Interface.	p. 21
Power Supply	p. 21
Software ROBO Pro	p. 21
Interface Expansions	p. 21
Preliminary Considerations	p. 22
Basics of the Interface	p. 22
The Basic Model	p. 23
Basic Program	p. 23
Pulse Control	p. 24
Subprograms	p. 24
Autonomous Tracked Vehicles	p. 25
Trail Searcher	p. 25
Tunnel Robot	p. 28
Color Detector	p. 29
Explorer	p. 30
RoboCupJunior – Rescue Robot	p. 32
Important Tips	p. 32

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 wheeled vehicles could not travel or at least only with extreme difficulty. 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 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 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 curve to the right was to be made, the right track was slowed down by using the 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 track-powered vehicles. These range from small excavators to the gargantuans for the mining of brown coal in surface mining.



In the Cheops Pyramid in Egypt, a minirobot was to be 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 driving directions 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 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 power 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 RoboCup-Junior.

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

Power Motor

In order to drive and steer the tracks and thus your models, you need two power motors. These are connected on the interface to the outputs M1 to M4.

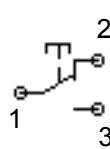
Domed Lamp

This is a light bulb for a voltage of 9 V. This can be used as an indicator signal for the direction of travel or simply for illumination. This is, as is the power motor, connected to the interface outputs M1 to M4.

Buzzer

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

Push Button Switch



Push button switches are in the category of contact sensors. If you push the red button, then a contact is mechanically moved in the casing and electrical current flows between the contacts 1 and 3. At the same time the switch circuit between the contact points 1 and 2 is interrupted.

Push button switches or switches are used in two different ways:

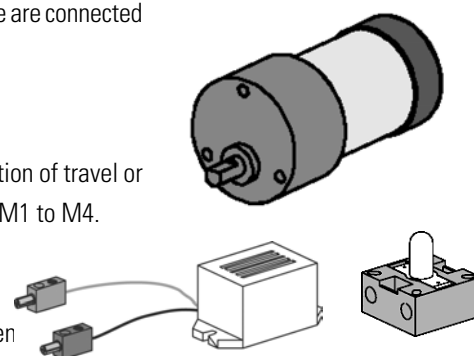
Push button switch as a „**normally open contact.**“

Both circuit diagrams show you the test setup. A voltage source (9 V) is connected to contact 1 of the push button switch, a lamp to contact 3 of the push button switch and to the negative pole of the voltage source.

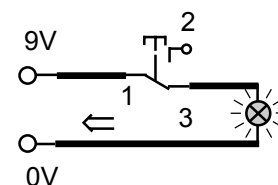
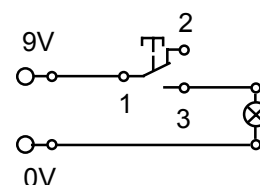
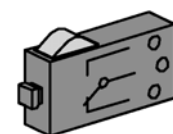
If the push button switch is not pushed then the lamp is off. If you push the push button switch then the circuit through contact 1 and contact 3 is closed, and the lamp lights up.

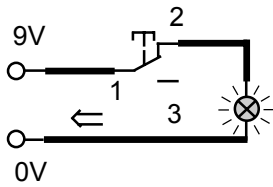
Explorer Model from fischertechnik

Actors



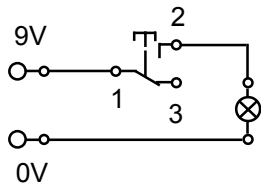
Sensors





Push button switch as a „**normally closed contact**.“

Both circuit diagrams here also show you how this works. Contact 1 is connected with the voltage source. Contact 2 with the lamp and the lamp with the negative pole of the voltage source.



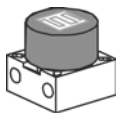
The lamp lights up. If you operate the switch now, the electrical current is interrupted and the lamp goes off.

In your models, you use the push button switch together with the pulse wheel as a counting push button switch.



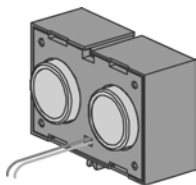
NTC Resistor(Negative Temperature Coefficient)

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 kOhms. When the temperature increases, the resistance value decreases. This information can be read into the interface at the analog inputs AX and AY and is available to ROBO Pro as a numerical value from 0 to 1023.



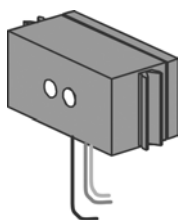
Photoresistor

The LDR 03, which is an analog brightness sensor for the inputs AX and AY (resistance measurement), reacts to daylight and changes its resistance value when it does this. This is an indicator for the brightness of the light.



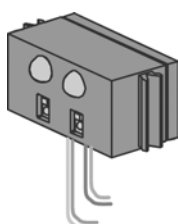
Ultrasonic Distance Sensor

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. D1 and D2 are available as connections on the interface. 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 engineering. 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 through the inputs A1 and A2 and process the numerical value from 0 to 1000 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. For connection, you need two digital inputs and the 9 V voltage supply (plus and minus).

■ The most important component for building a tracked vehicle is the ROBO Interface, which is installed as a fixed element in the various models. You can connect your sensors and actors to this as needed. You can find the basic wiring in the attached construction instructions.

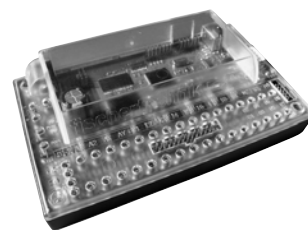
■ The ROBO Explorer models are autonomous vehicles, which move freely in the area. Therefore, you use the fischertechnik rechargeable battery set as the power supply.

■ 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.

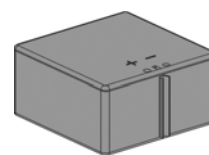
You need the ROBO Pro Version 1.2.1.30 for the construction set, ROBO Explorer. If you have an older version of the software, you can update this at no charge. Download this with the help menu in ROBO Pro—download new version—or at:

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

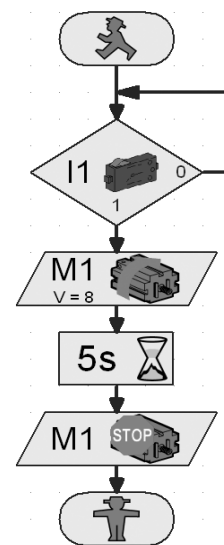
ROBO Interface



Power Supply



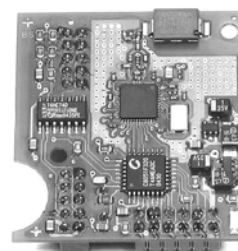
Software ROBO Pro



ROBO RF Data Link

The radio interface, ROBO RF Data Link, replaces the interface cable between the PC and the interface. The data are entered into the interface by means of radio transmission. The advantage is that the cable, which is often bothersome, is no longer needed. An additional advantage is that you can operate programs over radio in the online mode. In this way, you can find errors easier than in the download mode. And this method also has another advantage, mobile robots can be controlled online through the monitor and can transmit measured values to the computer without any cable. If several robots are equipped with a data link, they can even exchange data between each other, so to say „talk“ with each other.

Interface Expansions



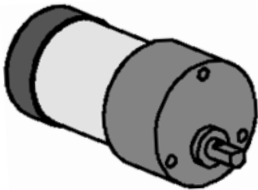
Preliminary Considerations

■ As for all other fischertechnik robots, with the ROBO Explorer, you will be introduced step-by-step to the fascination of technology and programming. 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.

Basics of the Interface

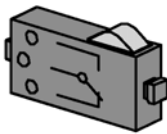
■ Before you start with the individual models, you should become familiar with the interface with the help of some experiments. For this purpose, put the handbook for the ROBO Pro software by your side so that you can look in the handbook if you have problems.

After you have installed the software, you can connect the interface using the cable, which is supplied, to your PC. Now start the program, ROBO Pro, and open the „Interface test.“



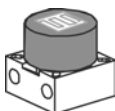
Power Motor

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



Push Button Switch

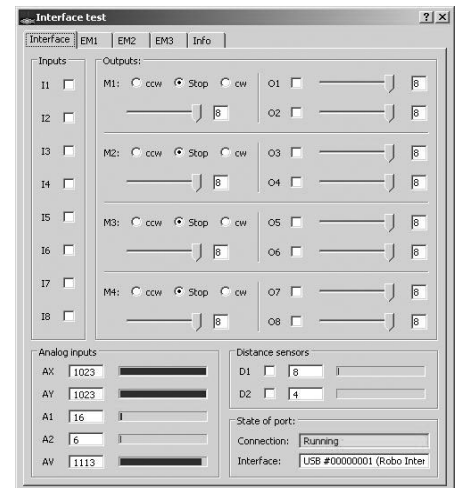
Now, connect a push button switch (normally open contact) to the digital input I1 and observe what happens in the test display when you push the push button switch. An optical display (check mark) shows you the status of the push button switch. Connect the push button switch as a normally closed contact and watch the result. First, the status display for „Switch Closed“ appears. If you push the switch, the information displayed disappears.



Photoresistor

Connect the photoresistor, which was supplied to you, to the connection AX or AY. Change the luminous intensity of the photoresistor by slowly covering with a black strip of paper. What happens? You see the blue light bar and the numerical value for the input used changes.

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



■ Now, let's get started. Now that you have found out how the interface and the programming work, you can solve the first task. First, the basic model is to be built using the construction 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 left 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.



After this, you need two motor symbols. Place the first symbol under program start so that the connection is automatically made. 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, „Left.“ 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 to the left and M2 to the right. 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 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 „red little 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.

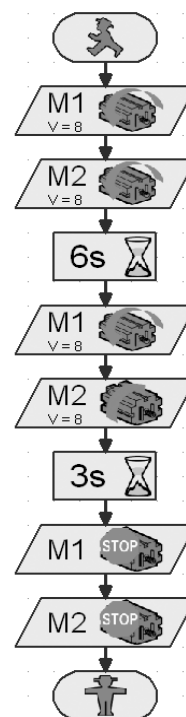
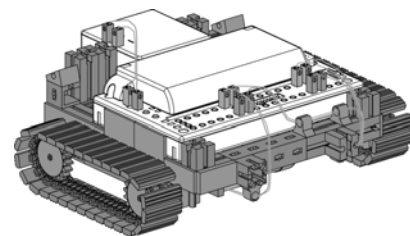
If you did everything right, then you can load the program onto the interface. To do this, click on the button „Download the Program to the Interface.“ Accept the setting of the download window. Right after the download, the model starts. Unfortunately, it is still connected with the USB cable. Load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

You can find the finished program at:

C:\Programs\Robopro\SamplePrograms\Robo_Explorer\Basic_Model_1.rpp.

That was really to simple for you. But, as we already said, you should move forward slowly. For this purpose, the task will be expanded somewhat.

The Basic Model

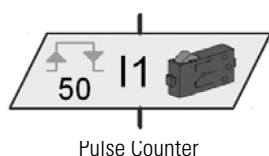
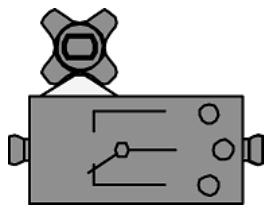


Start program in online mode



Download program to ROBO interface

Pulse Control



■ The basic model already contains two sensors (push button switches) for measuring distances and one pulse wheel for each, which is connected with the motor axle. This pulse wheel operates the switch four times for each revolution. Check the model with „Interface test“ for proper functioning.

Task 2 - ROBO Pro Level 1

Your tracked vehicle is to move 50 pulses forward. Following this, it is to travel 11 pulses to the left and then stop. You can find information about the pulse counter in the handbook in chapter 3.6.3.



Solution

First turn both motors on. Then add the pulse counter to these. In the properties change the number of pulses to 50. Again, insert two motor symbols. M1 continues to rotate to the „ccw,“ switch M2 to „cw.“ This turning is to last for 11 pulses. Finally, you must again insert two motor symbols and add „Stop.“ Transfer the program and test it on your vehicle.

When in the properties window for the pulse counter, you set the pulse type to „0->1“ or „1->0,“ you get eight instead of four pulses per rotation of the pulse wheel, which means the number of the pulses per distance traveled doubles and the precision of the measuring of the distance increases.

You can find the finished program at:

C:\Programs\Robopro\SamplePrograms\Robo_Explorer\Basic_Model_2.rpp.

Subprograms



Generate new subprogram



Copy current subprogram



Delete current subprogram

■ When you were making the program, you certainly noticed that you had to set the motors for every change of direction or stop. As a result, big programs become complicated very fast and the search for errors often becomes a test of patience.

Here, ROBO Pro offers „subprograms,“ which you can work with, as an elegant solution. Read all of chapter 4.1 in the handbook on this subject. It is important that you switch to level 2 in ROBO Pro.



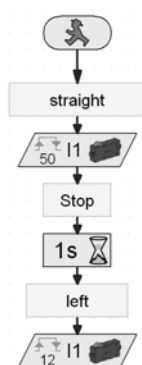
Task 3 - ROBO Pro Level 2

Your tracked vehicle is to move around a square. As for the first task, use the parameters 50 and 11. Make a subprogram for each direction of travel and for the Stop command.

First, make the subprogram „Forwards“ (see ROBO Prof Handbook chapter 4). Mark the program parts and copy them to the cache. Following this, prepare the subprogram, „Left“ and „Stop.“ Insert the program parts for „Forwards“ into both from the cache and change the parameters accordingly. For the turn, use a lower speed of rotation. 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.

Solution



Direction of Travel	Direction of Rotation Motor 1	Direction of Rotation Motor 2
Forwards	CCW	CCW
Backwards	CW	CW
Left	CCW	CW
Right	CW	CCW
Stop	Stop	Stop

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

Finished Program:

C:\Programs\Robopro\SamplePrograms\Robo_Explorer\Basic_Model_3.rpp

Task 4 - ROBO Pro Level 3

The robot does not travel in a precise square. Check the following questions: What is the cause? How can the result be improved?



Tip:

Synchronize the motors M1 and M2 using the push button switches I1 and I2 so that the model travels exactly straight ahead. For this purpose, we have supplied a finished program, „forward_sync“ for you. Here you only have to enter the number of pulses as a constant in the orange subprogram input (see Basic_Model_4.rpp). You can also enter the number of pulses for the angle of rotation in exactly the same way in the subprogram, „Turn_sync.“

Finished Program:

C:\Programs\Robopro\SamplePrograms\Robo_Explorer\Basic_Model_4.rpp

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

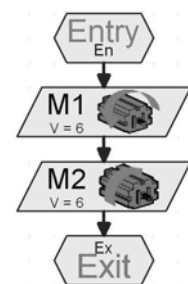
In order to allow your tracked vehicle to identify its environment and to complete certain tasks, you must equip it with sensors. The following model suggestions show you different variations of tracked vehicles with various sensors. This is to provide the capability to identify different paths, light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

C:\Programs\Robopro\SamplePrograms\Robo_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 the hand of a ghost. 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 construction instructions. You can find an experiment course with a printed black line in the construction set. The line, which is to be followed by the trail searcher, is to be straight at first.



Subprogram, „Left“

Autonomous Tracked Vehicles

Trail Searcher





■ Now, how is the model to function?

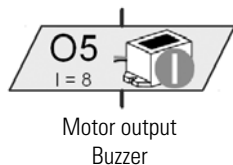
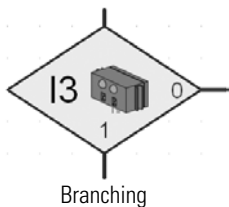
The robot is to find a black line on a white background and then follow this line. To make this possible, you have installed an IR trail sensor in your model. The module transmits a light in the infrared range to the pavement surface. Depending on the background, this is reflected and measured by phototransistors.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



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 if the trail ends, it is to stop and transmit an acoustic signal for one second.

Subdivide the entire program into a main program for the sensor interrogation and a subprogram for the travel straight ahead, one for the buzzer signal and one for stop.

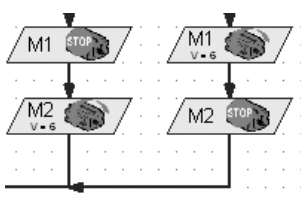


A Few Little Tips:

Use the interface test to check the trail identification by the 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.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 phototransistors so that the robot recognizes when it is not traveling exactly on the trail. Then, it is to correct the direction of travel accordingly. You can find a tip in the left program section.

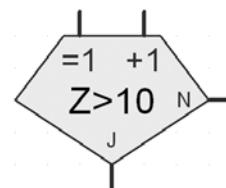


Now it looks better already. Your robot remains precisely on the prescribed trail. In an industrial building, other robots would remove the load, which was transported, at the end of the trail or would give the robot a new load. This load could then be transported back to the starting point of the trip.

Finished Program: **Trail_searcher_2.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.



Counter loop

Tip:

Think about the first tasks with the basic model. Here, the robot was supposed to turn 90 degrees. You did this with a distance traveled control. This technology can be of use to you here as well.

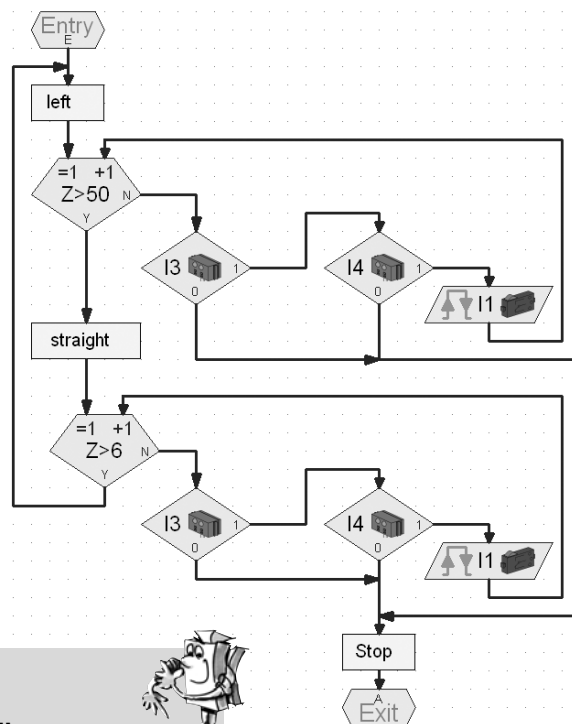
For the trail search, prepare your own subprogram with the name „track-seeking.“ In the diagram, you see how we suggest this be done.

Write another subprogram with the name, left. You have already created forwards and stop.

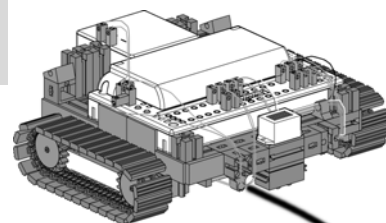
Thus, the main program and all subprograms for the trail searcher are finished. Now, your task is to integrate these into the program so that task 3 can be completed.

Finished Program: **Trail_searcher_3.rpp**

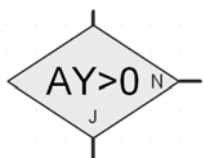
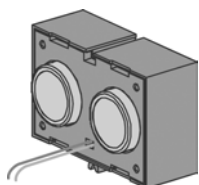
The previous routes were always straight. However, routes sometimes also have curves. You can find these, for example, in industrial facilities, in which material or workpieces must be transported from one machine to the next one.

**Task 4 - ROBO Pro Level 2**

The experiment course contains various curves with differening radiuses. 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 initiating possible 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 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 conscientiously build the model, „tunnel robot,“ according to the construction instructions as well. The distance sensor is connected to the connections D1.

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.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, AY. Hold a heat source close to the NTC and observe the blue bar for AY.

Task 2 ROBO Pro Level 2

Expand the program, which allows the robot to travel along the tunnel wall. As an addition, measure the current heat with the AY connection. If this increases to a certain value then the 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.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 a 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, the 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 interface.

The color sensor is installed in the color detector model. This is connected with the black wire to A1, the red wire to plus and the green wire to minus. 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

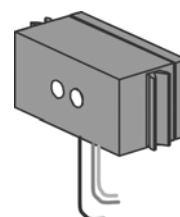
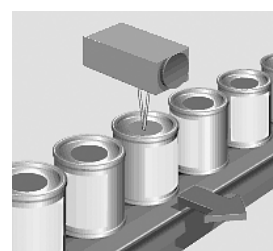
Write a small program, with which the sensor can identify the green colored area. If the measured value is within the prescribed range of values, the buzzer is to be activated for one second. Following this, the program jumps to start.

Finished Program: **Color_detector_2.rpp**

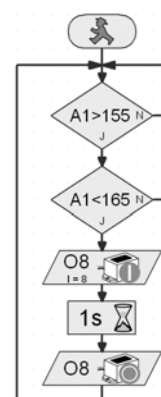
Tip:

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

Color Detector



Color	Value
White	
Black	
Blue	
Red	
Green	





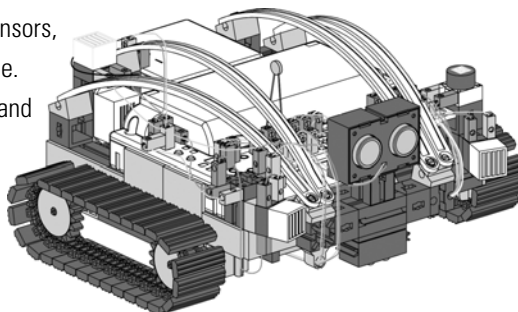
Task 3 - ROBO Pro Level 3

Write a program which allows your robot to travel straight ahead on a certain route. 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 lamp 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 remains there.

Finished Program: **Color_detector_3.rpp**

Explorer Model

■ The „Explorer“ model contains all actors 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 travels up to an existing obstacle on its trip. At a distance of about 60 centimeters, it is to reduce its speed by half. At a distance of 40 centimeters, it is to stop. If the obstacle gets closer to your robot, the robot is to travel slowly at a distance of 20 centimeters and at 10 centimeters, it is to travel backwards fast.

Finished Program: **Explorer_1.rpp**

Task 2 - ROBO Pro Level 2

Now, your robot is going on an 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 experiment 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.rpp**

Task 3 - ROBO Pro Level 2

The program from task 2 is to be expanded with three sensors: the color sensor, 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 indication light is to blink. As soon as the room becomes dark, the robot turns on its two headlights. If it gets bright again, the headlights are turned off.



Finished Program: **Explorer_3.rpp**

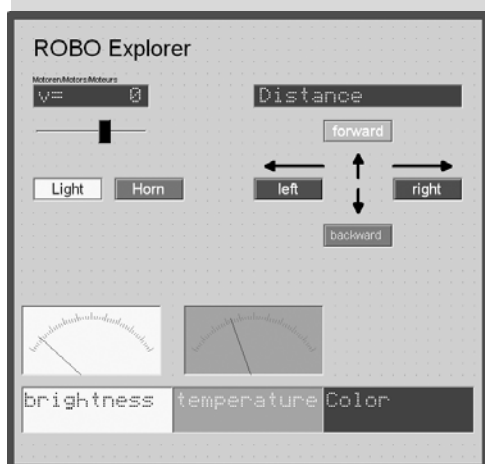
■ In 2003, Mars was closer to the Earth than almost ever before. The Europeans and Americans exploited this to send up their own exploration vehicles. Their mission: To prove life.

Our explorer can also be programed as a remote controlled robot to explore unknown worlds. The RF data link is needed additionally for this purpose.

Mars - Start to the Fourth Planet

**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 Panel (see handbook chapter 8).

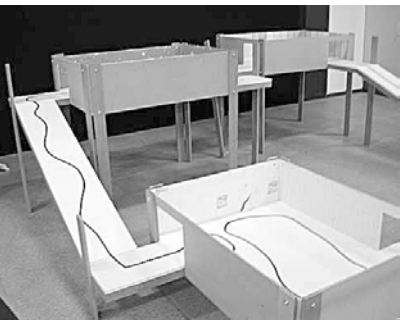


Your robot is to be programed so that it transmits measured values for the ground color, temperature, brightness and obstacles. The robot is to be controlled by hand through the "Panel" of the main program in ROBO Pro.

Finished Program: **Explorer_4.rpp**



RoboCup Junior Rescue Robot



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

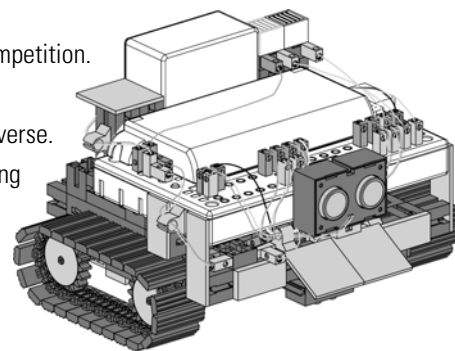
■ RoboCupJunior is a worldwide, project oriented education initiative, which promotes regional, national and international robot events for young people. The goal is to introduce children and young people to robots and their application.

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

www.robocupjunior.de,

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 various figures 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 with plugs. Check for proper functioning using a plug in light holder (38216), a 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 Explorer. If the voltage goes below 5 V, this is indicated by a red LED on the interface. At the same time, the interface is automatically turned off. Improper functioning can also occur even if the rechargeable battery is not so low. But then the rechargeable must still be charged.

Programing

When all mechanical problems have been solved and the robot is still not quite right, this is often due to errors in programming. 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.

