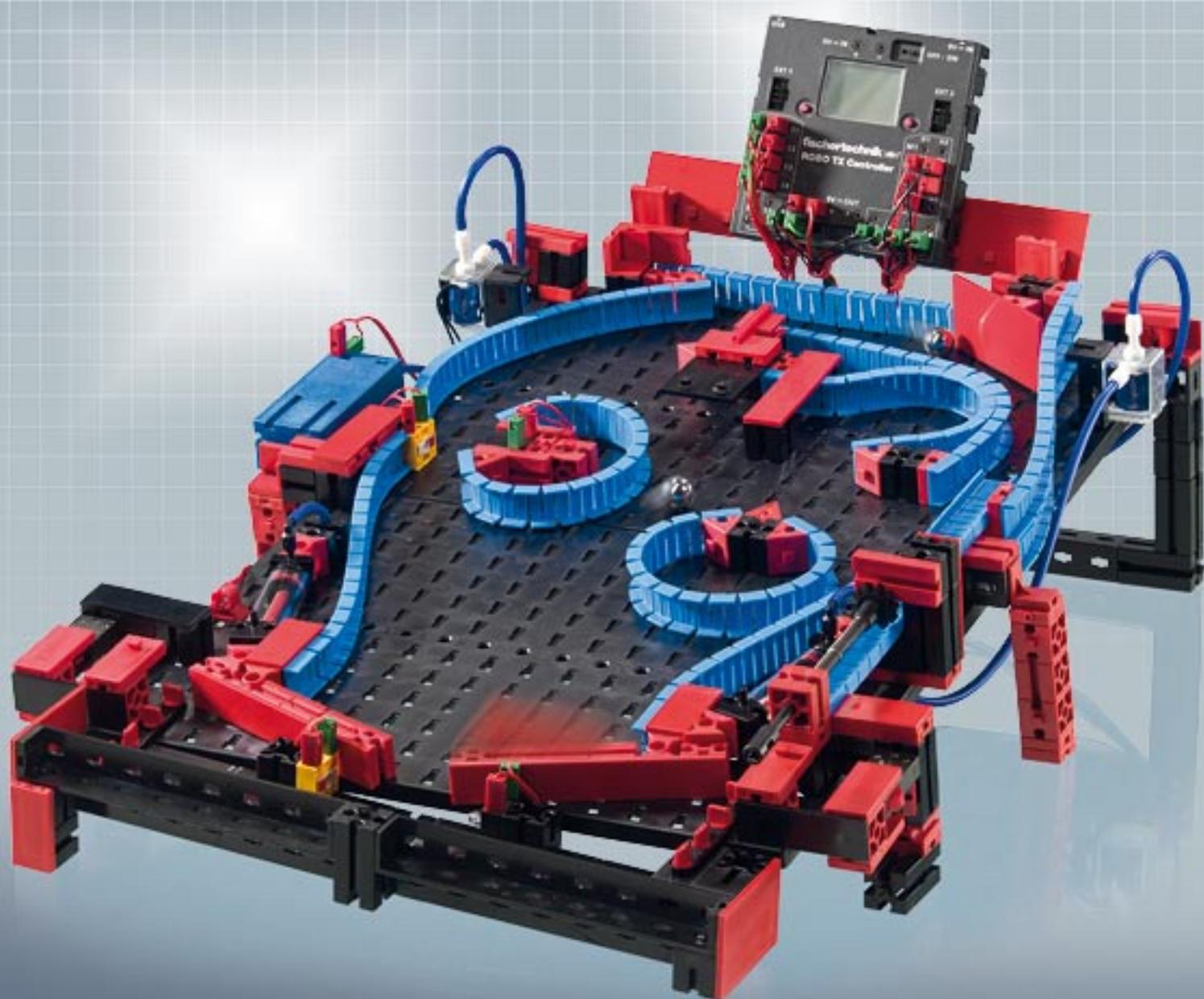




**fischertechnik** 

**COMPUTING**



**ROBO TX** *ElectroPneumatic*

**4** MODELS  
MODELS

## Contents

<b>Welcome to the fischertechnik Computing World</b>	<b>3</b>
About this Activity Booklet	3
<b>History</b>	<b>4</b>
<b>Principles of pneumatics</b>	<b>4</b>
Producing motion with air	4
Generating and storing compressed air – diaphragm pump as compressor	5
Switching compressed air – solenoid valves	6
Interaction of electrical and pneumatic circuits	6
ROBO Pro control software	7
ROBO TX Controller	7
<b>Compressed air motor</b>	<b>8</b>
Programming	9
<b>Color sorting robot</b>	<b>10</b>
Sensors	10
Sensors and actuators	11
Color recognition subprogram	12
<b>Ball obstacle course with vacuum picker arm</b>	<b>13</b>
Variables	14
<b>Pinball machine</b>	<b>15</b>

## Welcome to the fischertechnik Computing World

**Hello!**

Congratulations on your purchase of the ROBO TX ElectroPneumatic construction set from fischertechnik. We promise you that your interest will be rewarded. Because with this construction set you can conduct many interesting experiments and solve exciting tasks.



This digital activity booklet accompanies you step for step as you try out your fischertechnik models. It contains important tips and valuable additional information for conversion and optimization. At the end you will be able to control and program various electro-pneumatic models with the ROBO TX Controller. We start with simpler things initially to ensure you have fun right from the very beginning. With the knowledge you gain you can then meet the challenge of the next task - and so forth - step for step.

So don't be timid, we will plunge into the fischertechnik Computing World together and then go on to more complex tasks.

Now we wish you a great deal of fun and success experimenting with the ROBO TX ElectroPneumatic set.

Your team from

**fischertechnik** 

### About this Activity Booklet

This PDF Activity booklet has a few features not present in the printed booklet. Most are similar to those you may already be familiar with from the Internet. But sometimes they can also do more.

- **Purple text**  
This shows you information on the term itself when you **roll over it** with the mouse.
- **Underlined blue text:**  
This actuates a function when clicked - for example starting the [ROBO Pro help](#).
- **ROBO Pro Symbol:**  
This is always located in the vicinity of tasks. This makes sense, because as soon as you click on it a suitable example program opens with a possible solution.  
All example programs are listed under **C:\Programs\ROBOPro\Example programs\ROBO TX Electro-Pneumatic**.

test only.rpp

## History



Compressed air is one of the oldest forms of energy. Nearly 2500 years ago soldiers built military equipment, which used compressed air to shoot projectiles such as balls or spears.

Ctesibius from Alexandria in Egypt, (\* 296 B.C. in Alexandria, † 228 B.C.), was a Greek engineer, inventor and mathematician, who lived during the first half of the third century before Christ and built weapons operated by compressed air.

It is therefore no wonder that the word "**Pneumatic**" was taken from the Greek word "**pneuma**", which means "**air**".



The first compressor was a bellows. In blacksmith shops in the middle ages and far later, right up to today's modern industrial era, bellows have been used to increase the temperature of a fire.

Today pneumatics play an essential role in modern industry. Pneumatically driven machines and automation equipment can be found everywhere. For example, on assembly lines various parts are put together to form assemblies and the function is checked; goods are sorted and packed.

## Principles of pneumatics



Air can be used for different purposes in technology. For example wind drives gigantic windmills to generate electrical power. Pneumatics uses air to generate motion and transfer forces.

You are certainly familiar with at least one pneumatic tool - the air pump for your bicycle tires. It is designed using the physical and technical principles of a cylinder, as introduced in this construction set, for example using a compressor to generate compressed air.

### Producing motion with air

A number of pneumatic cylinders are included in the ElectroPneumatic construction set. You need one of them for the first experiment.

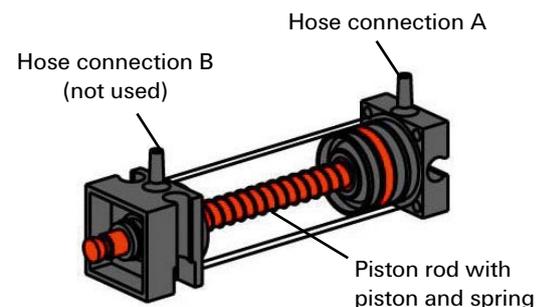
#### Pneumatic cylinder from fischertechnik

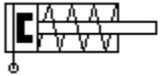
The piston rod with piston can move and is sealed along the cylinder wall by gaskets.

If you blow air into the cylinder through connection A, the piston moves.



Industrial cylinder





Circuit diagram of single acting cylinder

The air moves this cylinder in one direction only. It is returned to its initial position by the force of a spring. Such cylinders are called "single acting cylinders".

**Note:**

The connection you use to move out (extend) the piston is designated "A"; the piston is pulled back (retracted) with the aid of a spring.

**Air can be compressed**

Anyone working with pneumatic equipment today should know something about the physical properties of air. You can use a little test for this purpose:

Pull the red piston rod in the cylinder all the way out. Then hold connection A shut with your finger. Now release the piston rod. What can you observe?

The piston rod is pulled back only a short distance by the spring.

**Result:**

The air in the cylinder is pressed together (compressed) preventing the piston rod from moving. The more air compressed, the greater the air pressure in the cylinder. This pressure can be measured with a pressure gage or also calculated. The unit of measure for pressure is "bars" or "Pascals"

You can remember the following equation for this:

**Pressure = force/area or  $p = F/A$**

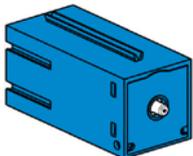
From this equation you can see that the pressure depends on the force exerted on a round surface in the cylinder.



Pressure gage for measuring air pressure

**Generating and storing compressed air – diaphragm pump as compressor**

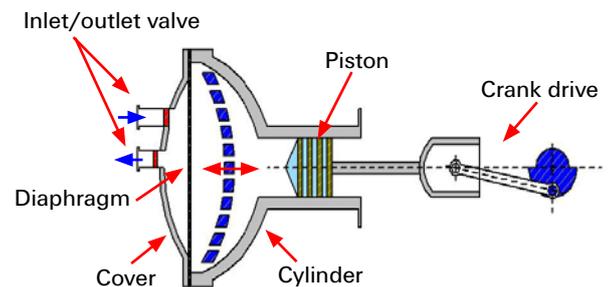
The diaphragm pump included in the construction set supplies the compressed air required for you to control the individual models. In industrial circles this is known as the compressed air source.



Compressor

**Mode of operation:**

A diaphragm pump consists of two chambers separated by a diaphragm (membrane). In one chamber the resilient diaphragm is moved up and down by a piston or cam. During the downward stroke the diaphragm is pulled back and air is pulled into the second chamber through the inlet valve. When the piston moves up, the diaphragm presses the air out of the pump head through the outlet valve.

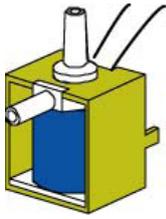


**Note:**

The pressure generated by the compressor is approx. 0.7 to 0.8 bars. The diaphragm pump is maintenance-free.



Circuit diagram of compressed air source



3/2-way valve

### Switching compressed air – solenoid valves

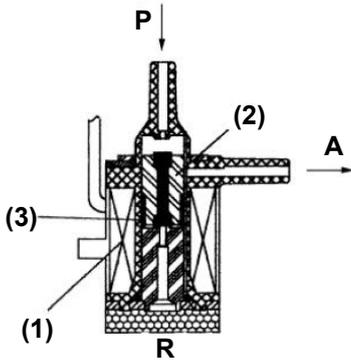
In pneumatics the purpose of a valve is to control the flow of air to the pneumatic cylinder so that the cylinder either extends or retracts. The valve can be actuated either by hand, pneumatically or electromagnetically as on your technical models.

Technical data on valve: 3/2-way valve, 9V DC/130 mA

3/2-way means that the valve has three connections and two switching states.

**Note:**

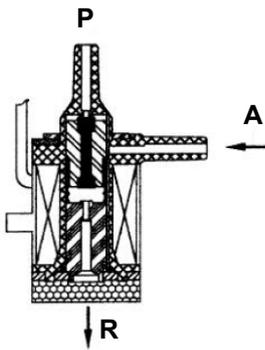
When connecting the valve to the power source or to the ROBO TX Controller ensure that the polarity is correct.



**Brief technical explanation:**

Applying a voltage to the coil (1) creates a magnetic field which pulls down core (2). The valve opens and the air flows from connection "P" through connection "A" to the cylinder. When voltage is not applied, the core is pressed upward by the spring (3) and the valve is closed.

When the valve is closed, connection "A" is connected to the vent "R". This is important, to allow the air to escape from the cylinder.



The connections are always designated as follows in pneumatics:

**P = Compressed air connection**

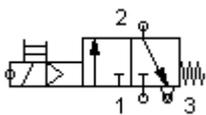
**A = Connection to cylinder**

**R = Vent**

### Interaction of electrical and pneumatic circuits

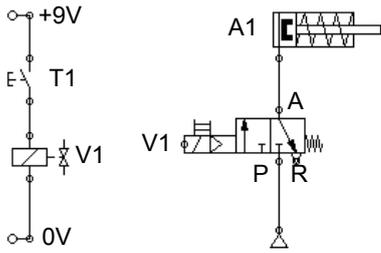
**Task:**

Actuate a solenoid valve to extend a **single acting cylinder**. The cylinder should extend when the operator closes a switch. As long as the switch is closed, the cylinder should remain extended. When the switch is returned to the initial position, the pressure of the spring should cause the cylinder to retract.



Circuit diagram of a 3/2-way valve

Engineers frequently use symbols to show such tasks. One circuit diagram shows the electrical part and one the pneumatic part or stage.



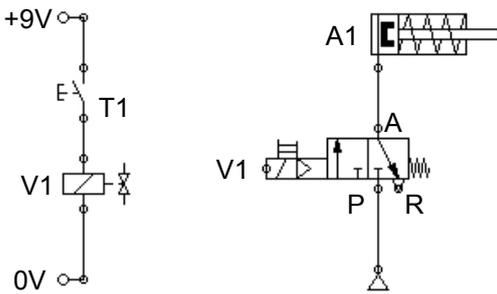
Circuit diagram - electrical, pneumatic stage

In the illustration the electrical part is on the left and the pneumatic stage on the right. The electrical part consists of a +9V power source, a pushbutton and the valve coil (electromagnet). The pneumatic stage consists of the compressed air source, the valve and the cylinder.

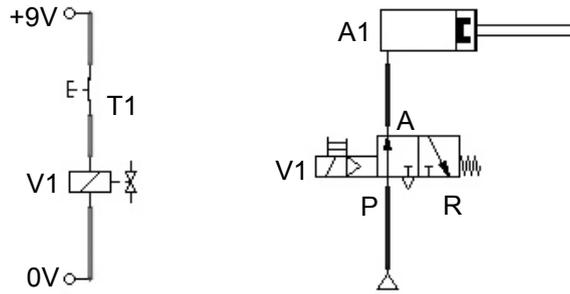
**Note:**

Since the magnetic coil and valve form one unit, they are shown using the same symbol. This clearly shows that the coil and valve belong together.

The two figures below show the circuit in the non-actuated state, in the figure on the right with the button pressed. The figure on the right clearly shows the flow of electricity as well as air.

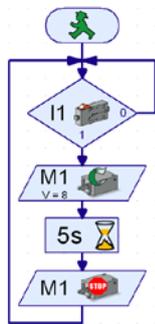


Circuit diagram - non-actuated position



Circuit diagram - button pressed

**ROBO Pro control software**



Program example with symbolic commands

**Control logic with ROBO Pro software and ROBO TX Controller**

In addition to its mechanical construction, the unit requires the control logic, software for the PC and an interface ( ROBO TX Controller) to convert the software commands into signals which the machine can execute.

The ROBO Pro control software has a simple graphical programming interface which allows you to create programs without having to learn a programming language.

**For the ROBO TX ElectroPneumatic construction set you need ROBO Pro version 3.1.3 or higher. If you have an older version of the software, it will be automatically updated when you install the ROBO TX ElectroPneumatic CD.**

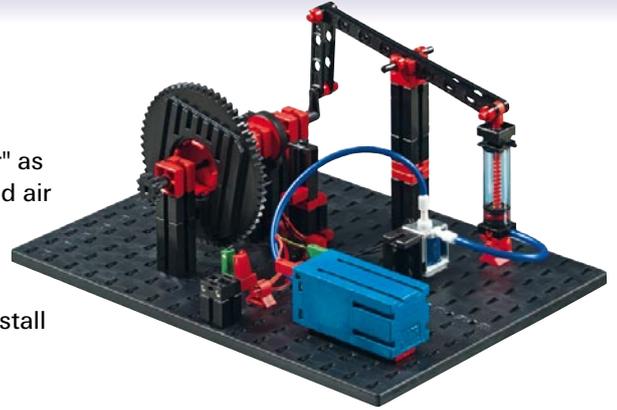
**ROBO TX Controller**

The ROBO TX Controller is the most important component in the model. It controls the actuators (motors, lights & valves) and evaluates information from the sensors. For this purpose the ROBO TX Controller has numerous connections for connection to the components. The instruction manual for the ROBO TX Controller describes which components can be connected to which connections and the functions of the connections.



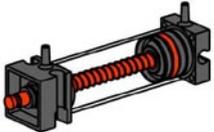
A special feature is the integrated Bluetooth interface. It allows you to complete a wireless link between your PC and the ROBO TX Controller. Or to connect a number of controllers with one another. You can define how the controller interacts with the individual components and what they are to do in detail in the program you write with the ROBO Pro software.

## Compressed air motor



3/2-way valve

For your first model, build the "compressed air motor" as described in the assembly instructions. A compressed air or pneumatic motor functions similar to a steam engine. It has a cylinder, a piston, an intake and an outlet. However, compressed air serves as the "propellant" instead of hot steam. For the first task install the button so that you can press it with your hand.



Pneumatic cylinder

### Technical information on pushbutton.

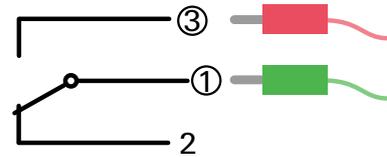
The pushbutton has three connections or terminals. Depending on the application you can use the pushbutton ....

... as a "normally open switch":

By connecting terminals 1 and 3.

When the pushbutton is pressed: Electricity flows.

When the pushbutton is not pressed: Electricity does not flow

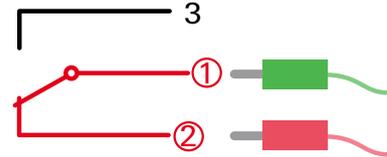


... or as a "normally closed switch":

By connecting terminals 1 and 2.

When the pushbutton is pressed: No electricity flows.

When the pushbutton is not pressed: Electricity flows.



Pushbutton

Wire the electrical components as described in the assembly instructions in circuit diagram A.



### Task 1: Manual control with pushbutton.

Press the button and observe the model. What makes the wheel turn and how?



As you can see, each time you press the button the solenoid valve switches and the cylinder extends. This causes the wheel to rotate half a turn. When you release the button, the wheel rotates another half turn. This is accomplished by the return spring in the pneumatic cylinder.

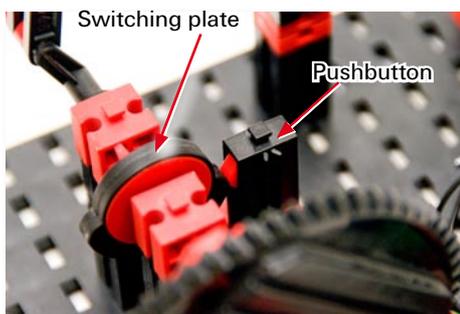


Switching plate

### Task 2: Control with switching plate.



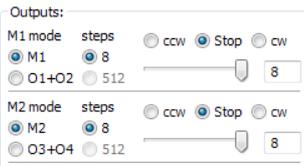
Install the pushbutton to the installation position specified. Instead of pressing the button manually, it is now actuated by a switching plate. How will this affect the model?



It is necessary to switch the valve on and off at the right time. Then the wheel will rotate continuously.

**Important!** Adjust the switching plate so that the pushbutton switches the valve exactly when the crank is in the top-most or bottom-most position.

For the next task install the ROBO TX Controller in the model and wire the electrical components as described in circuit B in the assembly instructions.



**Task 3: Testing the model with the ROBO TX Controller**

Connect the ROBO TX Controller to the power supply and switch it on. Connect the ROBO LT Controller to the PC. Then start the ROBO Pro software. Activate the "Test" button. The operating display appears to test the controller and the connected sensors and actuators. Click with the mouse pointer on Output M1 - right and then Output M2 - right. Observe what happens at Input I1.

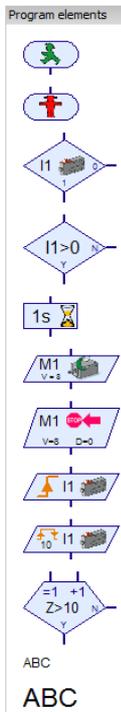


The compressor at Output M1 starts running and produces compressed air for the cylinder. If M2 is switched on, the solenoid valve is actuated and the piston in the cylinder extends. At Input I1 a check set when the connected pushbutton is closed.

**Programming**



New program

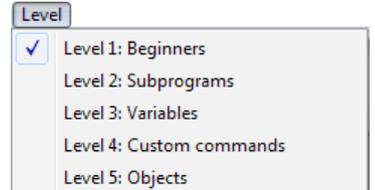


**Task 4: Program control with ROBO Pro - Level 1**

ROBO TX Controller takes over actions in task 3 using program. Here the switching state of pushbutton (I1) is checked and the information, "closed/open" used to control the valve and cylinder.



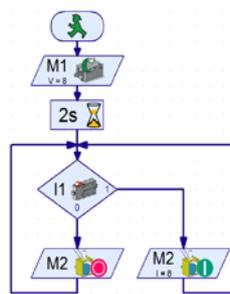
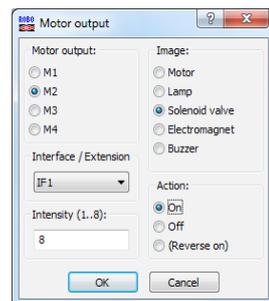
The operating display can be cleared with the "New" button. In the "Level" selection window switch to level 1.



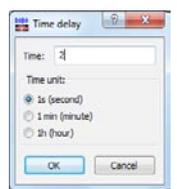
All the commands required for this task are present in the "Element group" selection window. However the [ROBO Pro help](#) is also very valuable here.

Each program always starts with the "Little green traffic light man". Then the individual program commands such as "Motor on" or "Time delay" can be entered. The symbols can be moved to the operating display with the mouse using the drag and drop feature. Information on the commands used is given in [ROBO Pro help](#) Chapter 3.

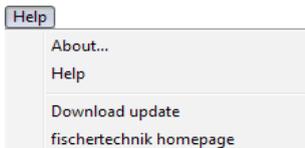
Use the program from the following program structure:



Click with the right mouse button on the desired symbol; an interactive window appears in which you can make various settings, e.g. set the time, actuator, etc.



**Important!** You can get help under the menu point



or by clicking the right mouse button on the program element in the "Element window"

You can call a finished example program for this task with this symbol.

compressed air motor.rpp



Start program in online mode



Stop all programs in progress



Download program for ROBO TX Controller

After finishing the program, you can start it with the button "Start program in online mode". The individual program steps are then performed. Since you have programmed a continuous loop, it is also necessary to stop it, when you are finished. For this purpose use the button "Stop all running programs". It may be necessary to readjust the switching plate when operating the compressed air motor with the ROBO TX Controller, to get the motor to run "smoothly". You can do this by trial and error.

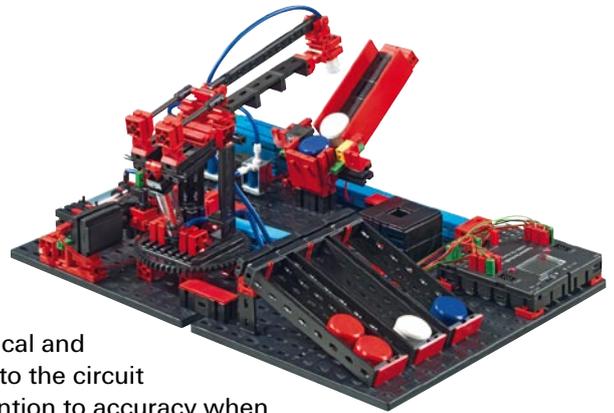
It is possible to transfer the program to the ROBO TX Controller. This can be accomplished with the "Download" button. The interactive window appears to enter the various parameters.



Start program: The program can be started immediately after transfer or after pressing a button. Information is given in [ROBO Pro help](#) Chapter 3.7.

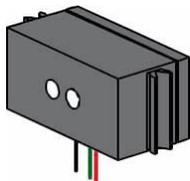
## Color sorting robot

The color sorting robot model is designed to sort parts automatically according to their color. Build the model according to the assembly instructions and wire the electrical and pneumatic components according to the circuit diagram. During assembly pay attention to accuracy when installing the parts, connecting the hoses and wiring the electrical components. This eliminates the need for trouble-shooting when you put the model into operation.



## Sensors

With this model you become familiar with new components such as those used in industrial equipment. This includes the vacuum picker arm with suction cup, the two-cylinder vacuum generator, the optical color sensor and the light barrier with phototransistor and light source.



### Optical color sensor

Color sensors are used frequently in automation technology. This is done, for example, to examine the color or the color imprint to ensure that the correct components are installed. The fischertechnik color sensor transmits red light, which is reflected with different strength from different colored surfaces. The quantity of reflected light is measured by the phototransistor and output as a voltage between 0 V and 10 V. A type of "darkroom" is built into the sensor in this model to prevent excessive light scatter. An opening is provided for the sensor. The part to be measured can be positioned on top of this opening.

Connect the color sensor to I3 using with the black and green cable and to + with the red cable (see circuit diagram in assembly instructions).

Color	Value
White	
Red	
Blue	

**Task 1 - Defining color values:**

First check the values output by the ROBO TX Controller for the parts in the interface test (white, red, blue). Use the interface test in ROBO Pro for this purpose. Set input I3, connected to the color sensor, to analog 10V (color sensor).

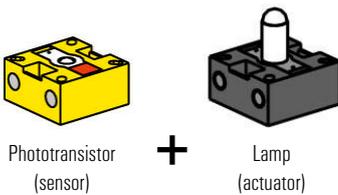


Make a small table and enter the values you have measured. Also observe the changes when the distance to the colored surface or the ambient light changes.

In addition a light barrier is used for the next task. This consists of a light-sensitive sensor (phototransistor) and a lens tip lamp (actuator) as light source.

**Sensors and actuators**

**Light barrier**

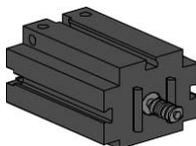


**Phototransistor**

You can also call the phototransistor a "brightness sensor." This is a "sensor" that reacts to brightness. In a light barrier it is the counter-part of 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! Pay attention to the polarity when connecting the phototransistor to the power supply. Red = positive.

**Lens tip lamp**

This is an incandescent bulb with built-in lens to focus the light. It is required to build a light barrier. The lens tip lamp can be recognized by its gray base.

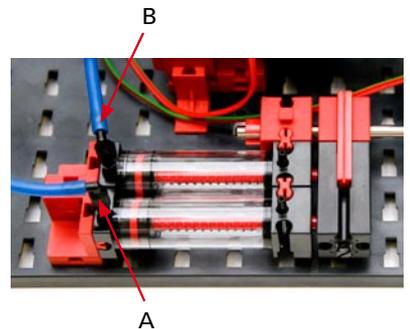


**Electric motor**

The direct current motor converts electrical energy into mechanical energy. This results in the rotational movement of the motor. A motor also has a gearbox. With this gearbox you can reduce the speed of the motor and simultaneously increase the **torque** (motor pulling force).

**Vacuum pump**

You need a vacuum pump for the model to produce the vacuum to pick up the parts. Since the vacuum pumps used for industrial applications are very expensive, you can use a simpler solution for producing a vacuum. You need two cylinders with connected piston rods. Connect connection A to the compressor over the solenoid valve. Connection B leads to the suction cup. When the valve is actuated, both pistons are pushed forward. If the suction cup is positioned on a part, the air is sucked in by the two pistons producing a vacuum.



The bellows type suction cup has a lifting function and can be used on flat as well as slightly arched surfaces.

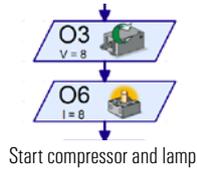
**Task 2 - Control program - ROBO Pro Level 2**

A light barrier is located at the end of the chute to recognize whether a part is present. If so, the gripper arm picks up the part and places it on the color sensor for recognition of the color. The sensor determines the color.



The part is then placed in the proper box.

Box 1 = white, Box 2 = red and Box 3 = blue.



Start compressor and lamp



Button connect to C1 as pulse counter

**Tip:**

For programming consider the sorting sequence. Here is a little help:

- Compressor switches on - Output O3 also for lens tip lamp
- Model starts after pressure builds up
- The robot arm turns until limit switch I1 is switched; motor turns: counterclockwise
- Light barrier check - is part present?
- Part picked up and transported to color recognition - pulse counter check C1D
- Color checked by color sensor I3
- Continued transport and separation according to color - pulse counter C1D
- Return to check light barrier - limit switch I1

You can call a finished example program for this task with this symbol.

color sorting robot.rpp

**Color recognition subprogram**

In addition to the main program you can also create subprograms in ROBO Pro. These serve to keep your program structure clear - once written, subprograms can also be copied into other applications.



Subprogram



Input subprogram



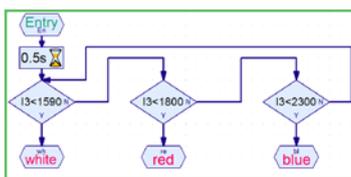
Output subprogram

Information on what subprograms are and how they can be used is given in the [ROBO Pro help](#) in Chapter 4. It is important that you switch to level 2 or higher in ROBO Pro.

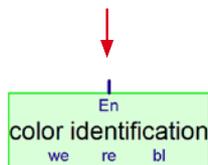
The "color recognition" subprogram provides an example. As defined in the task, the color is to be recognized and the part then stored according to color. The jump to the "color recognition" subprogram is shown by a green box in the main program.

**Important!** To incorporate a subprogram into a main program, it is first necessary to write the subprogram.

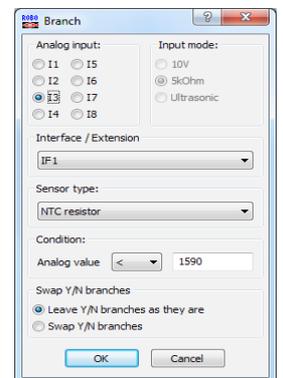
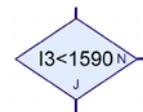
It may be necessary to adapt the values in the color recognition subprogram to the values you determined in task 1, so that the color recognition functions properly.



Color recognition subprogram

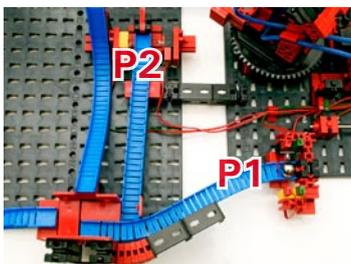
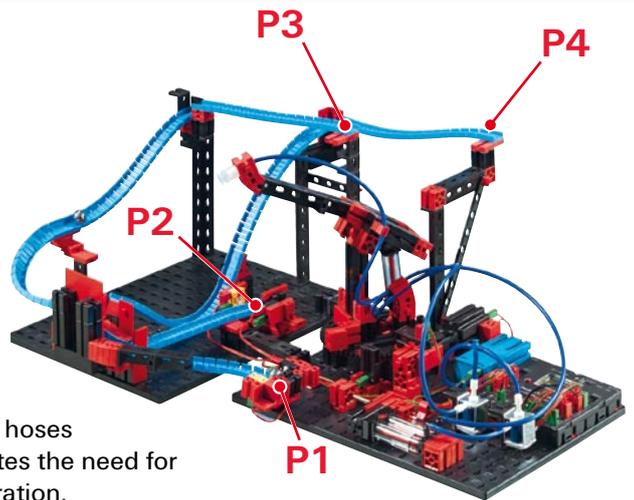


Symbol for color recognition subprogram



## Ball obstacle course with vacuum picker arm

With this model the vacuum picker arm automatically returns balls from the end to the start of an obstacle course. Build the model according to the assembly instructions and wire the electrical and pneumatic components according to the circuit diagram. During assembly pay attention to accuracy when installing the parts, connecting the hoses and wiring the electrical components. This eliminates the need for trouble-shooting when you put the model into operation.



Ends of obstacle courses

### Task 1 - Return a ball - ROBO Pro Level 2

A ball is to be returned at one of the two ends of obstacle course P1 or P2 by the vacuum picker arm. The two light barriers are checked to determine where the ball is located. After picking up, the ball is transported to position 3 on the obstacle course and put down. The ball rolls to the gate where it is guided to one of the two final paths. The program is intended to run as a continuous loop.



### Tips

As with the previous programming tasks, it is necessary to consider the sequence you want. Also consider which parts of the program will need to be written in a subprogram.

- Switch on the compressor and lens tip lamps for the light barriers (2 sec. time delay)
- Start the model and position the robot arm at the start position - Check sensor I1, motor direction of rotation: Counterclockwise
- Check the two light barriers
- Pick up the balls
- Transport to set-down location P3 and set down balls - pulse counter
- Return to light barrier check to see if new balls are present

You can call a finished example program for this task with this symbol.

ball obstacle course\_1.rpp

### Task 2 - Both obstacle courses - ROBO Pro Level 2

The light barrier check tells the robot where a ball is located so that the robot can transport the ball to the start of the track. If the ball is at the end of the front track (P1), it should be put down at position P3. If the ball is at the end of the rear track (P2), it should be put down at position P4.



You can call a finished example program for this task with this symbol.

ball obstacle course\_2.rpp

**Tips**

The position of the gripper arm can be defined with the aid of the "pulse counter" command. The values required can be entered in the interactive window.

If the ideal position is between two pulses, you can also adjust the start or end position of the balls by moving one or more of the building blocks so that the gripper picks up or puts down the ball reliably.

Information on the commands used is given in [ROBO Pro help](#) Chapters 3. and 8.1.10.

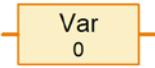
**Variables**

**Task 3 - Two balls in obstacle course - ROBO Pro Level 3**

Now two balls roll through the obstacle course. Two conditions should be fulfilled to ensure that the robot distributes the balls uniformly on the two tracks:

- Balls arriving at the bottom end points (P1 or P2) should be transported alternately to the upper starting points (P3 or P4).
- If balls are present simultaneously at both end points (P1 and P2), they should also be picked up alternately.

You can meet these requirements by supplementing the program.

**Tips**

So-called variables help to solve this problem. Information on what variables are and how they can be used is given in the [ROBO Pro help](#) in Chapter 5. It is important that you switch to level 3 in ROBO Pro.

Here is the solution:

Define a variable (Pos1 – Pos4) for each of the 4 positions P1, P2, P3 and P4. When a position is reached, set the value for the associated variable to 1 to help you remember the position moved to last.

Example:

<b>First ball:</b>	<b>Variable</b>	<b>Value</b>	<b>Variable</b>	<b>Value</b>
Ball present at P1	Pos1 =	1	Pos2 =	0
Ball moved to P3	Pos3 =	1	Pos4 =	0

**Second ball:**

Ball present at P1 and P2. Since Pos1=1, the robot should now go to P2. Then Pos1 is set to 0 and Pos2 to 1. Since the ball was set down at P3 during the first operation (Pos3 = 1), the ball is now transported to P4. Then Pos1 is set to 0 and Pos2 to 1.

Too complicated? You can call a finished example program for this task with this symbol.

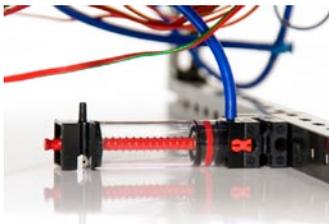
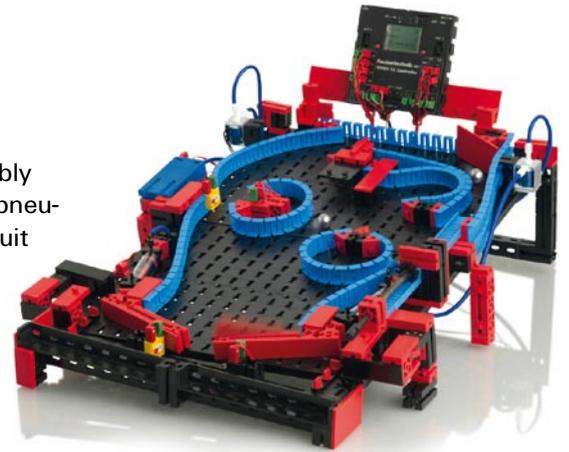
ball obstacle course\_3.rpp

Attempt to understand how the principle works by looking at the final sequence.

## Pinball machine

Now you can play pinball!

Build the model according to the assembly instructions and wire the electrical and pneumatic components according to the circuit diagram. During assembly pay attention to accuracy when installing the parts, connecting the hoses and wiring the electrical components. This eliminates the need for trouble-shooting when you put the model into operation.



Cylinder as air reservoir

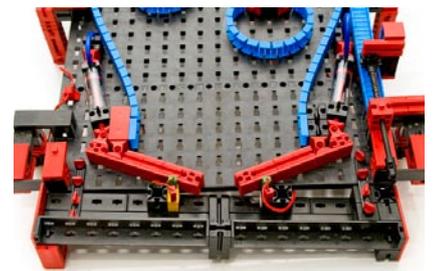
- Before starting to program the model, here is further technical information on the cylinder below the model as well as additional actuators and sensors. The cylinder is used as a storage reservoir for air, to ensure that enough air is available to move the cylinders when the flipper buttons are pressed quickly one after the other.

- The ball can be shot into the playing field by a spring mechanism on the right side of the model. This works according to the same principle as a real pinball machine.



Ball launching pad

- The two buttons at the left and right lower edges serve for electrical control of the two valves for the built-in cylinders. These control the two flipper arms. A light barrier detects when a ball drops through the middle gap or runs past an incorrectly positioned flipper arm. This light barrier also serves later for counting the balls planned for one game.

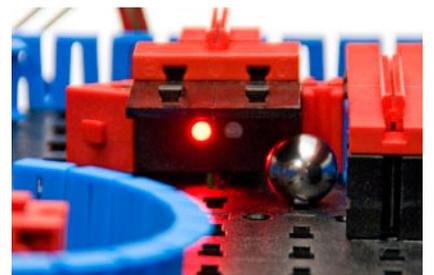


Pinball control

- Another light barrier and the color sensor serve to keep the score. The light barrier records when the ball passes through and the color sensor measures the distance between the color sensor and ball. This is explained in greater detail later in the program explanation.



Light barrier for detecting passage



Color sensor for proximity measurement

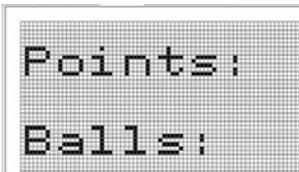
- The ROBO TX Controller display serves to indicate the score and number of balls remaining. The display can be controlled with the two red buttons.



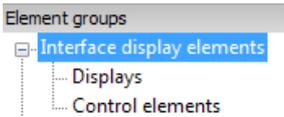
Measured value logging and output

Main program	Left flipper	Right flipper	Light barrier	Star
Function	Symbol	Panel	TX Display	Properties
				Descrip

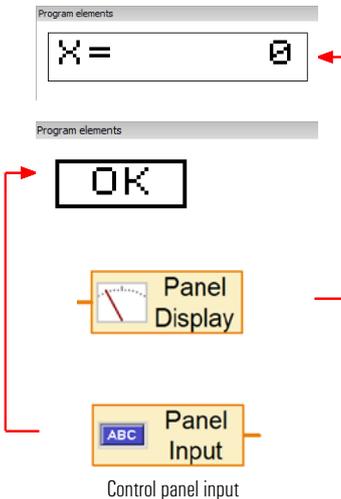
TX display program call



Defined TX display



TX display element group



Control panel input

### Task 1 - Pinball control program - ROBO Pro Level 3

A ball is shot into the game field from a launching pad. The ball then runs through the obstacle course set up. When it passes the light barrier or comes close to the color sensor, points are given. The points are shown on the TX display. Every player has three balls. The round is completed when all balls have been played. The game can be started again by pressing the red button on the left (OK button).



You already know all the commands required for this task. New is the fact that the data is shown on the ROBO TX Controller display.

You can call a finished example program for this task with the symbol.

pinball machine.rpp

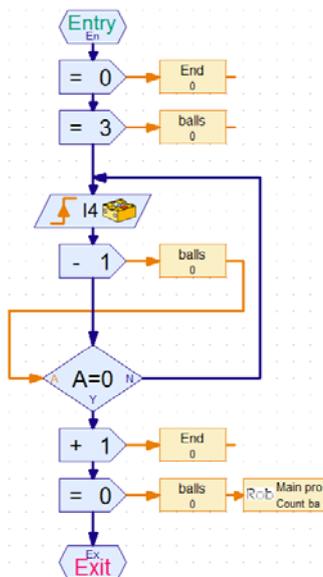
Information on the commands and how they can be used is given in the [ROBO Pro help](#) in Chapters 9 and 11.

#### Tip:

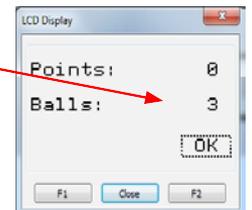
For programming consider the game sequence. Here is a little help:

- Switch on compressor 03 - light outputs 06 and 07 (for light barriers)
- Shoot in ball - manually
- Check flipper buttons I1 left and I2 right, actuate valves at 04 and 05
- Check and evaluate color sensor at I8
- Check and evaluate light barriers at I3 and I4
- Output points earned and balls played on display
- End of game, reset outputs 03, 06, and 07 to zero
- Start new game with OK button (right display button)

The "Count balls" subprogram illustrated is intended to show you the interrelationship between the program and the TX display.



At the start of the game the value 3 appears on the display indicating that 3 balls are available. When the light barrier at I4 detects a ball, the value shown on the display is reduced by 1.



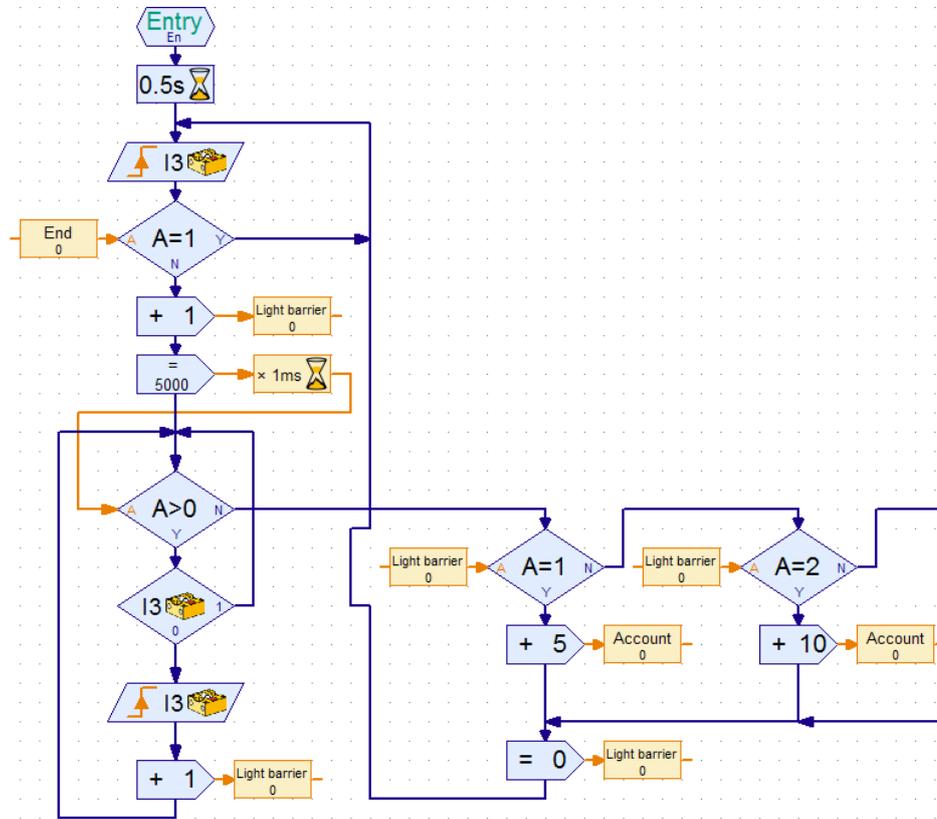
After all balls have been played, the display indicates zero. The game is finished.



If you want to stop the program loaded on the TX controller completely, it is necessary to press both display buttons simultaneously.

Keeping score:

You can use various systems to assign points. In the program used as an example the program waits 5 seconds after the light barrier is interrupted for the first time and simultaneously counts how many times the light barrier is interrupted during this time. The more times you quickly shoot the ball through the light barrier, the more points you get.



Light barrier subprogram

The color sensor allows you to get more points, when the ball passes closer to the sensor.  
 Do you have any other ideas? Use your imagination.  
 Enjoy programming and playing with your pinball machine.

Your team from