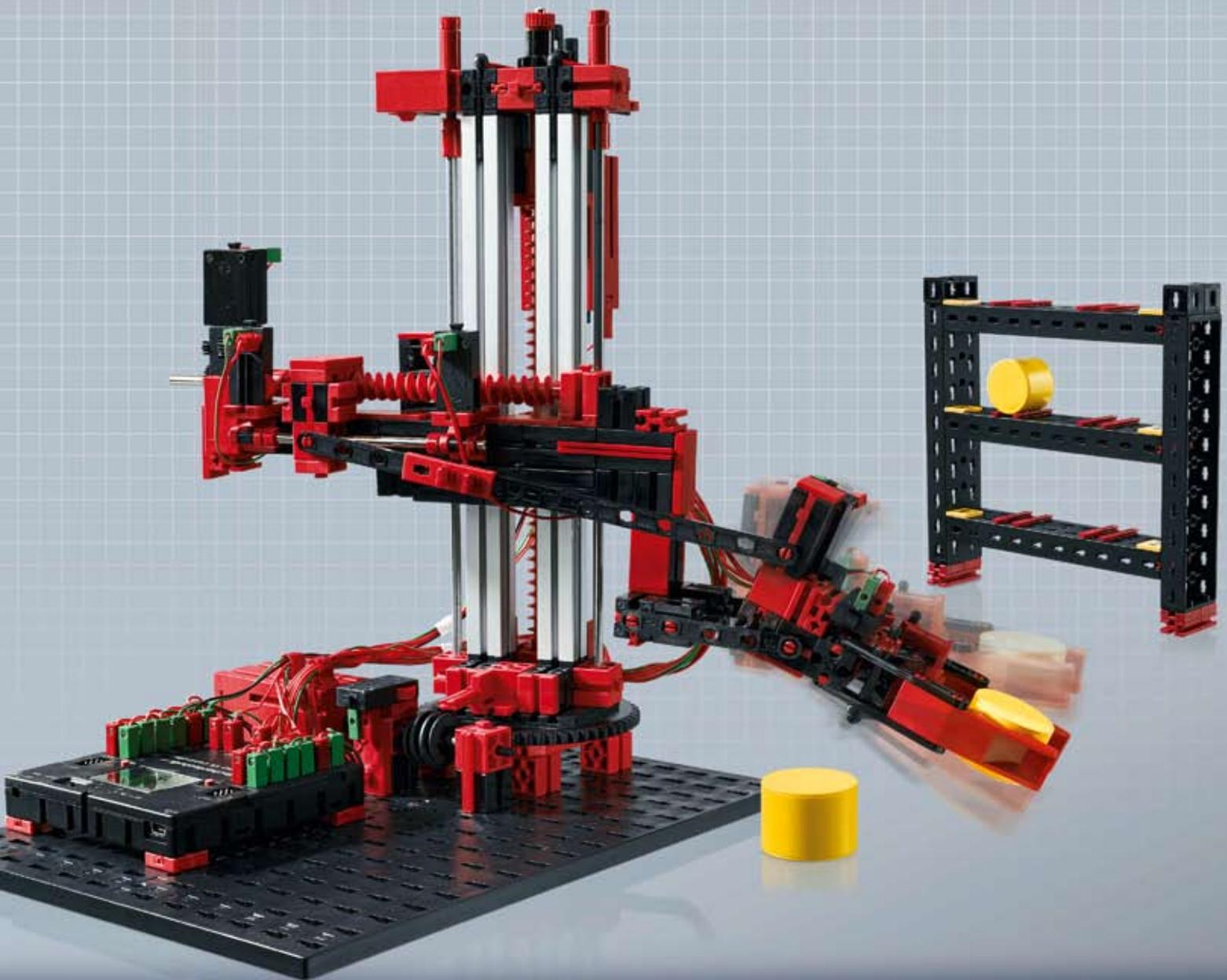




fischertechnik 

COMPUTING



ROBO TX Automation Robots

4 MODELS
4 MODELS

Contents

Welcome to the fischertechnik Computing World	3
About this Activity Booklet	3
Industrial robots	4
Component Explanations	5
The Construction Set Contains All of This	5
Actuators	5
Encoder Motors	5
XS motors	5
Sensors	6
Pushbutton switches	6
As a "closer" (normally open switch)	6
As a "break contact" (normally closed switch)	6
ROBO TX Controller	6
ROBO Pro Software	7
Gripper robot	8
Robot coordinate system	8
High bay storage rack	10
Pivoting gripper	12
3-axis robot	14
Teach-in procedure	14
Quick programming for 3-axis robot	15
Load, start, select control panel	15
Keys on control panel	15
Stop	16
Save	16
Towers of Hanoi	17
Loading teach-in sequence on TX controller	17

Welcome to the fischertechnik Computing World

Hello!

Congratulations on your purchase of the ROBO TX Automation Robot set from fischertechnik. We promise you that your interest will be rewarded. This is because with this construction set you can conduct a lot of interesting experiments and solve exciting tasks.



Reading through this monitor booklet and trying out the experiments and assignments will teach you step for step how various industrial robots can be controlled and programmed with th ROBO TX Controller from fischertechnik.

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

So don't hesitate, simply get started and together we will then proceed to the more comprehensive assignments.

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

Your team from

fischertechnik 

About this Activity Booklet

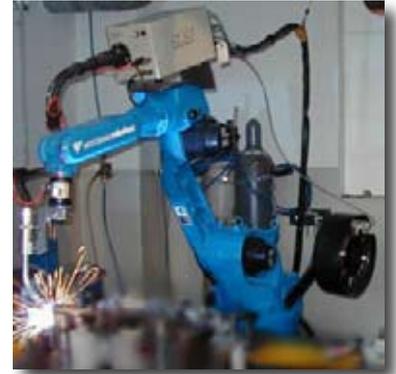
This PDF Activity booklet has a few features not present in the printed booklet. Most are similar to those you may already be familiar with from the Internet. But sometimes they can also do more.

- **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 assignments. 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:\Program Files\ROBOPro\Sample Programs\ROBO TX Automation Robots**.

For testing only.rpp

Industrial robots

An industrial robot is a machine with a universal, freely programmable motion sequence with the purpose of performing highly varying tasks. It can be used for handling, assembling or processing workpieces. Frequently industrial robots are equipped with a gripper to hold workpieces. However other tools can also be integrated to perform certain types of work. As the name says, Industrial robots are used for applications in industry (e.g. production of automobiles). As soon as a robot has been programmed for a task, it can perform the work by itself without manual control.



The inventor of the industrial robot is considered to be George Devol, who registered a patent in the USA in 1954 for a programmable manipulator. Together with Joseph F. Engelberger, Devol founded the world's first robot company, Unimation.

Areas of application

Industrial robots are used in many areas of production for work as:

- **Welding robots**
- **Cutting robots**
- **Measuring robots**
- **Painting robots**
- **Grinding robots**
- **Handling equipment for:**
 - Palleting
 - Stacking
 - Packaging
 - Assembling
 - Equipping machines
 - Removing parts

Component Explanations

The Construction Set Contains All of This

The set contains numerous fischertechnik modules, various motors, pushbuttons and colored assembly instructions for building various models.

After you have unpacked all the modules, it is first necessary to install some components before starting (e.g. cables and plugs). The exact ones are described in the assembly instructions under "Assembly Tips." It is best to do this first.

Actuators

Actuators are all components, which can perform an action. This means when they are connected to an electric current, they become "active" in some form. Usually you can see this directly, e.g. a motor starts turning.

Encoder Motors



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.

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

XS motors



The XS motor is an electric motor, which is exactly as long and high as a fischertechnik building block. In addition, it is very light. This allows it to be installed at locations where there is insufficient space for larger motors. The XS motor is designed for a voltage of nine volts and current input of a maximum of 0.3 amperes.

This is also connected to the outputs M1 to M4 on the ROBO TX Controller.

Sensors

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

Pushbutton switches



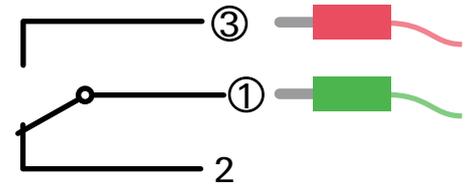
The pushbutton is also called a touch sensor. When the red button is pressed a switch is switched mechanically allowing current to flow from contact 1 (middle contact) to contact 3. Simultaneously the contact between connections 1 and 2 is interrupted. So you can use the pushbutton in two different ways:

As a "closer" (normally open switch):

Contacts 1 and 3 are connected.

Pushbutton switch is pressed: Electricity flows.

Pushbutton switch is not pressed: No electricity flows.

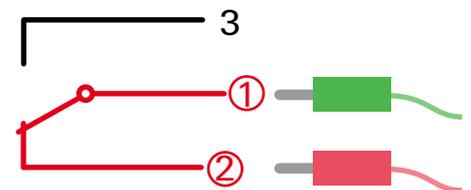


As a "break contact" (normally closed switch):

Contacts 1 and 2 are connected.

Pushbutton switch is pressed: No electricity flows.

Pushbutton switch is not pressed: Electricity flows.



The sensors are connected to the [universal inputs I1 to I8](#) on the ROBO TX Controller.

ROBO TX Controller

The ROBO TX Controller is the most important component in the robot model. It controls the actuators and evaluated information from the sensors. For this purpose the ROBO TX Controller has numerous connections for connection to the components. What components can be connected to what connections and what the functions of the connections are are described in the instruction manual for the ROBO TX Controller.

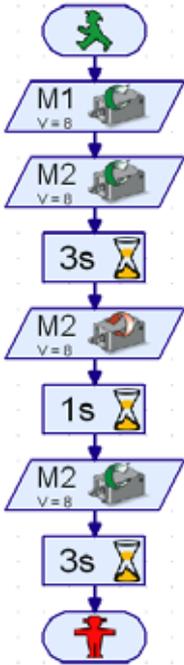


A special treat is the integrated Bluetooth interface. It allows you to connect your PC to the ROBO TX Controller. Or to connect a number of controllers with one another. You can define how the controller interacts with the individual components and what they are to do in detail in the program you write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

ROBO Pro Software

ROBO Pro is a graphical programming surface which allows you to create programs for the ROBO TX Controller.

"Graphical programming surface" means that you don't have to "write" the programs by hand line for line; you can put it together simply with the aid of graphic images. An example of this type of program is shown in the figure at the left.



The exact procedure for creating such a program is described in detail in the [ROBO Pro Help](#) in Chapters 3 and 4. It is best to first read through these chapters in [ROBO Pro Help](#). This will allow you to learn a little about the software, so that you can then start experimenting immediately.

You need the ROBO Pro version 3.1 or higher for the ROBO TX Automation Robot set. If you have an older version of the software, it will be automatically updated when you install the ROBO TX Automation Robot CD.

A Few Tips

Experimenting is most enjoyable when the experiments really work. For this reason it is necessary to observe a few basic rules:

- **Work carefully.**
Take your time and look precisely at the assembly instructions for the model. If you have to look for an error later then this will take much longer.
- **Check the movement of all parts.**
When putting models together continually check to see if the parts, which are to move, move easily.
- **Use the [Interface test](#)**
Before starting to write a program for a model, you should test all of the parts connected to the ROBO TX Controller using the ROBO Pro interface test. How you use these is described in the [ROBO Pro Help](#) in chapter 2.4.

Gripper robot

For your first experiment you can build the "Gripper robot" model using the assembly instructions and wire the electrical elements.

The robot can turn and raise and lower its arm. These are known as the motion axes of a robot.



Robot coordinate system

It is necessary to specify the position of each individual linear axis and the rotary axis of the robot. This then defines the position of the gripper.

The zero position for each axis and the gripper is defined by a limit switch.

Positioning is accomplished by counting the pulses in the encoder motor or on the mechanical pulse sensor.

The table below provides an overview of the various axes on your model:

Action	Axis	Motor	Limit switch	Pulse sensor/encoder
Turning	X	M1	I1	C1
Raising/Lowering	Z	M3	I3	C3
Open/close gripper		M4	I4	C4

Motor direction of rotation

- Counterclockwise: Axis moves in the direction of the limit switch
- Clockwise: Axis moves away from the limit switch

Task 1:

Write a program for a reference run. First the gripper should open. Then the robot should move against the limit switches with each of its axes.



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

Gripper robot_1.rpp

The reference run positions the axes and gripper precisely. We could also speak of the 0 points, from which the motion paths for the motors are calculated.



Task 2:
 Assemble the workpiece for the robot described in the assembly instructions. First complete a reference run with the model. Then the gripper arm should move downward exactly 1175 pulses and gripper the workpiece. Then the robot arm should move up 250 pulses, turn 1000 pulses and then the arm should move back down and release the workpiece. The industrial robot should then move back against its limit switches.

Programing Tips:

Special program elements are required to program the encoder motors.



How you use these is described in the [ROBO Pro Help](#) in chapter 4.4.2 and 11.6.

Before starting with the complete programming it is most practical to make up a flow chart. This could have the following appearance:



The final program can be loaded by clicking the symbol again.

Gripper robot_2.rpp

The robot can also be programmed conveniently using positioning subroutines.

Click the symbol to open an example program, showing how Task 2 can be solved with positioning subroutines.

Gripper robot_2a.rpp

Information on what subroutines 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 3 in ROBO Pro.

Task 3:
 Rewrite the program Gripper robot_2a.rpp so that the robot grips and raises the workpiece, turns and set the workpiece down on a platform.



The design of the platform is given in the assembly instructions.

The example program can be called as usual by clicking on the symbol.

However, before looking, first try to find the solution yourself. Good Luck!

Gripper robot_3.rpp

You will certainly be able to think of many more tasks for your robot. Enjoy programming and trying it out.

High bay storage rack

A high bay storage rack is used in a warehouse system where the goods are stored and retrieved automatically. Large high bay storage racks can be up to 50 meters high and offer space for several thousand pallets.

You can assemble the "high bay storage rack" model and wire the electrical elements as described in the assembly instructions.

In the **warehouse** the goods are stored in high bay storage racks. A lane is present between two rows, in which **rack feeders** run to store and retrieve goods in the racks. The high bay rack system also includes a **receiving station** where the goods are delivered or retrieved.



An overview of the various axes on your model is given below:

Action	Axis	Motor	Limit switch	Pulse sensor/encoder
Move rack feeder	X	M1	I1	C1
Load fork up/down	Z	M3	I3	C3
Load fork forward/back		M2	I2 - Fork at rear I4 - Fork at front	

Motor direction of rotation

- Counterclockwise: Axis moves in direction of limit switch / fork moves toward rear
- Clockwise: Axis moves away from limit switch / fork moves toward front

The built-in buttons I1-I4 serve as limit switches for positioning the system. Buttons I5-I6 are each described in the exercise example.



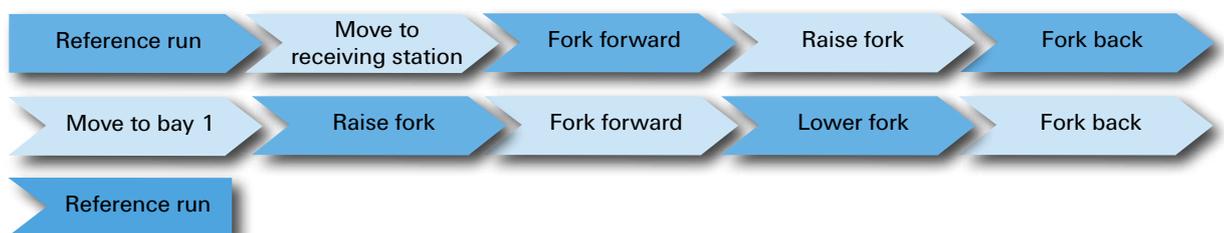
Task 1:

The rack feeder should pick up a barrel at the receiving station and store it in storage bay 1. For this purpose first draw a flow chart for creating the ROBO Pro program.

Programming Tips:

To program the motion paths for the X-axis and Z-axis you can use the same positioning subroutines, you used for the "Gripper robot" model.

A flow chart for this task could look like this:



Create a subroutine for each of the various sub-tasks. You can also use these subroutines for other tasks later.

High rack_1.rpp



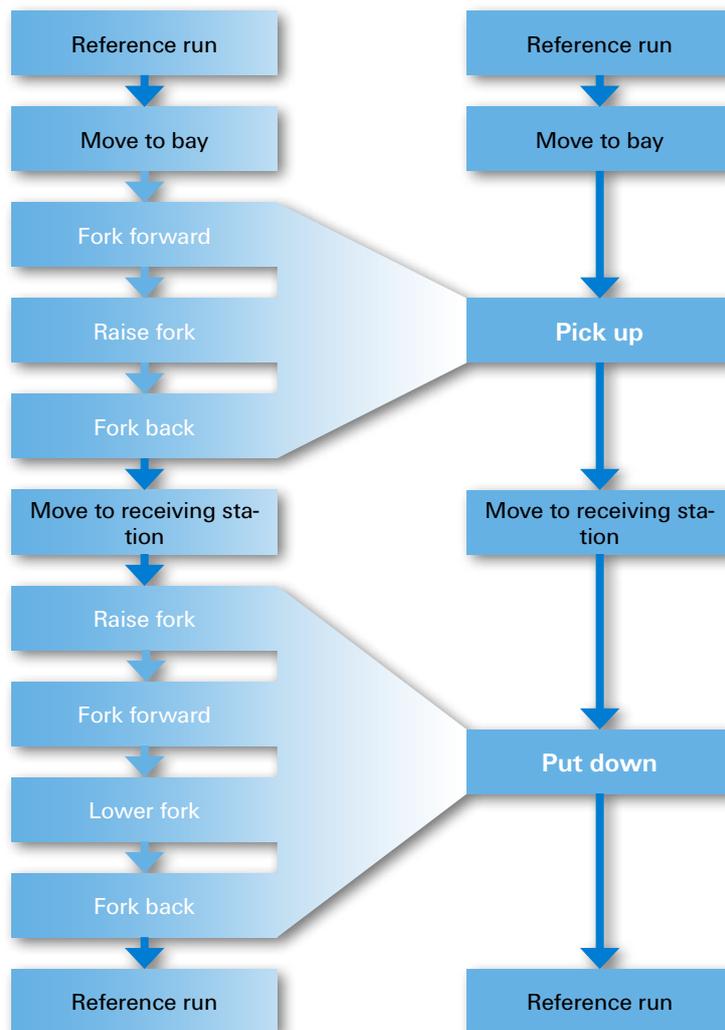
Task 2:
 Expand your program with subroutines to move to the remaining storage bays. Rewrite the program so that a number of barrels can be stored in various storage bays one after the other. Then have the stored barrels retrieved from their storage bays one after another and brought to the receiving station. Naturally during storage you will have to continuously place new barrels in the receiving station and during retrieval remove the barrels from the receiving station to provide room for the next barrel.

Programing Tips:

Placing a barrel in the receiving station is almost the same as placing it in its storage bay. In your flow chart it is only necessary for you to reverse "Move to receiving station" and "Move to bay".

The program can be organized more clearly by combining some sub-tasks into common **subroutines**:

High rack_2.rpp



Task 3:

High rack_3.rpp



Now you can make your high bay storage rack more convenient. Add a slide control to the display on your computer. You can then select the desired storage bay with this slide control. The selected storage bay should be indicated on the display. When button I5 is pressed, a barrel should be picked up at the receiving station and stored in the selected storage bay. When button I6 is pressed, a barrel should be picked up at the selected storage bay and moved to the receiving station.

Information on the ROBO TX Controller display and the sliding control is given in the [ROBO Pro Help](#) in Chapter 11.7.

It is important that you switch to level 3 in ROBO Pro.

Pivoting gripper

The industrial robots introduced up to now all have rigid grippers. Pivoting gripper can rotate around one axis. This allows parts to be turned and pivoted for handling, etc. The figure shows an industrial robot for feeding a machine with sheet metal plates taken from a stack and moved to the processing machine. During this process the plates are turned to a certain angle.

You can assemble the "pivoting gripper" model and wire the electrical elements as described in the assembly instructions.



An overview of the various axes on your model is given below:

Action	Axis	Motor	Limit switch	Pulse sensor/encoder
Turning	X	M1	I1	C1
Raising/Lowering	Z	M3	I3	C3
Pivot gripper		M2	I2 – gripper horizontal I5 – gripper vertical	
Open/close gripper		M4	I4	C4

Motor direction of rotation

- Counterclockwise: Axis moves in direction of limit switch / gripper pivots to horizontal position
- Clockwise: Axis moves away from limit switch / gripper pivots to vertical position

Task 1:
Write a program for a reference run. First the gripper should open and then move to the horizontal position. Then the robot should move against the limit switches with each of its axes.



Programing Tip:
Here you can use the positioning subroutines you already used for the other models.

Pivoting gripper_1.rpp

Place a holder with a horizontal barrel on the left and right sides of your robot. Position a deposit platform in front of the robot. Instructions for building the holders and the platform are given in the assembly instructions.

Task 2:
First complete a reference run with the model. Then the gripper moves to the vertical position. The robot should now move the gripper downward, grip the barrel, pick it up and the gripper should then pivot back to the horizontal position with the barrel. Then the robot should move the gripper over the deposit platform and set the barrel down there.



Then the robot should pick up the second barrel, and set it down on top of the first barrel to form a tower.
Finally the robot moves completes a reference run.

Programing Tip:
Use the positioning subroutines again to solve this problem.

Pivoting gripper_2.rpp

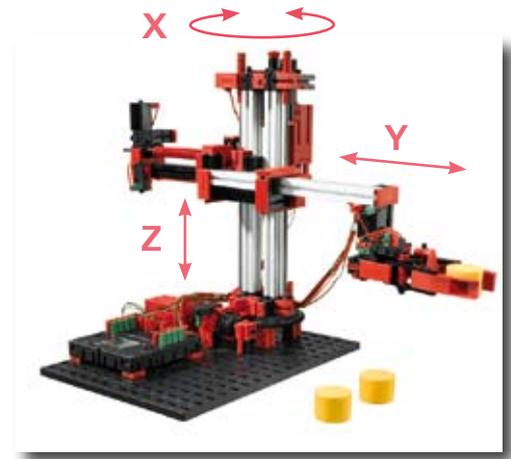
Task 3:
Expand the program in Task 2. The robot should now tear down the tower on the deposit platform and place the barrels back on the holders on the left and right next to the model.



Pivoting gripper_3.rpp

3-axis robot

This model is a 3-axis industrial robot. The robot gripper can move in three different directions. The robots, which have used up to now, were more like specialist, which were particularly good for one certain task. The 3-axis robot, by contrast, is an "all-rounder", capable of being used for highly varying types of work. Letters are used as follows to indicate the three motion axes on this robot. The X direction designates rotation of the robot, the Y direction is used for gripper extension and retraction, raising and lowering are the Z direction.



You can assemble the "3-axis robot" model and wire the electrical elements as described in the assembly instructions.

The motors and buttons installed in the model and the axis designations are illustrated in the table below:

Action	Axis	Motor	Limit switch	Pulse sensor/encoder
Turning	X	M1	I1	C1
Extend/retract gripper	Y	M2	I2	C2
Raising/Lowering	Z	M3	I3	C3
Open/close gripper		M4	I4	C4

Motor direction of rotation

- Counterclockwise: Axis moves in the direction of the limit switch
- Clockwise: Axis moves away from the limit switch

You can solve this problem by creating a program sequence with the aid of the positioning subroutines you already used for the other models. However you will need to write a new positioning subroutine for the Y-axis. For this purpose you can copy the positioning subroutine for the "gripper", rename it Pos Y and adapt it as required.

3-axis robot_1.rpp

However a so-called teach-in program for the 3-axis robot is also included in the example programs for this set. This allows you to program the robot very simply. The meaning of teach-in and how the teach-in program for your robot works are described below.

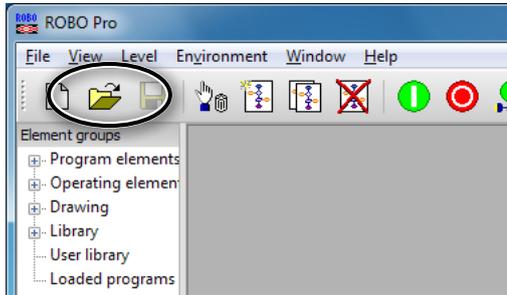
Teach-in procedure

Teach-in programs are used to program industrial robots. Teach-in means you can move the robot to a certain position with a control. When the robot has been moved to the right position, this position is saved. This process is repeated until the entire motion sequence has been saved. The program sequence then consists of the robot combining all saved positions to achieve one motion sequence. The robot can then perform this by itself.

After entering all positions, you can therefore exit the program and everything turns and moves.

Quick programming for 3-axis robot

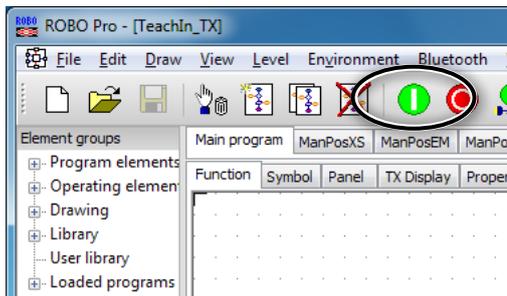
Load, start, select control panel



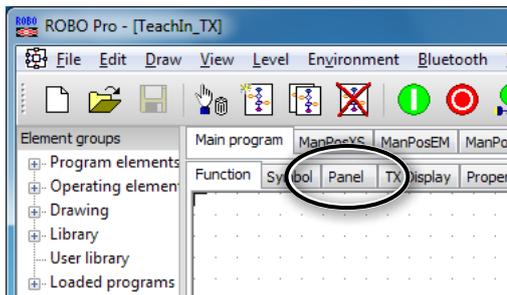
You can load the teach-in program by clicking on the symbol.

TeachIn_TX.rpp

It can also be called under the ROBO Pro example programs:
C:\Program Files\ROBOPro\Sample Programs\ROBO TX Automation Robots\TeachIn_TX.rpp.

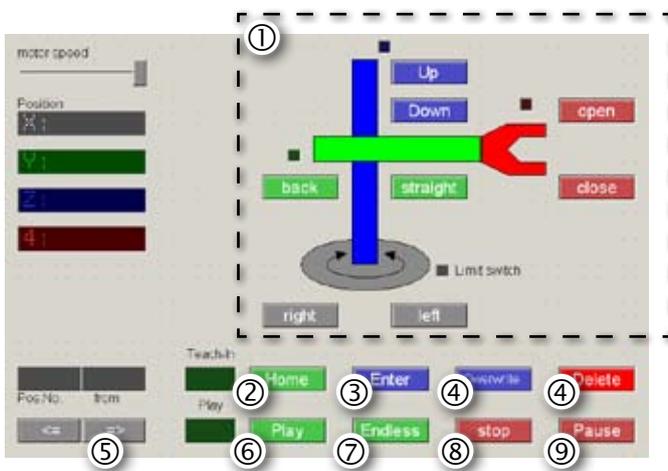


Start the teach-in program.



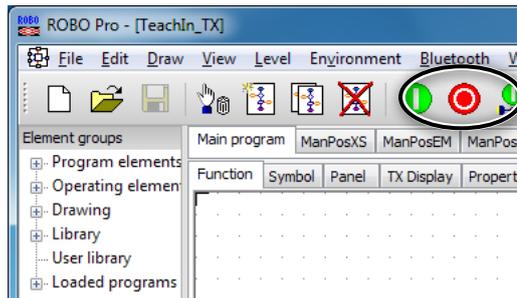
Select the panel to program the robot.

Keys on control panel



- ① Direction keys for controlling robot.
- ② **Home** = Move robot to starting position.
- ③ **Enter** = Save current position.
- ④ **Overwrite/Delete** = Change existing positions.
- ⑤ **Arrow keys** = Jump to previous/next position.
- ⑥ **Play** = Start programmed sequence, robot moves to all entered positions one after another.
- ⑦ **Endless** = Repeat sequence continuously.
- ⑧ **Stop** = Stop sequence.
- ⑨ **Pause** = Interrupt sequence and continue next time key is pressed.

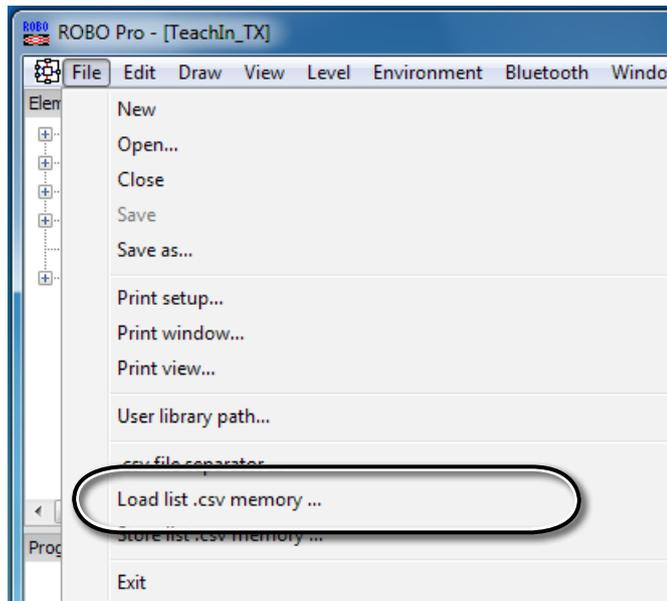
Stop



Stop teach-in program.

Save

Save programmed positions as table in a .csv file before closing teach-in program. You can then reload them each time the teach-in program is opened. If you close the program before saving the positions, they are deleted.



You can solve the following task with the aid of the teach-in program:

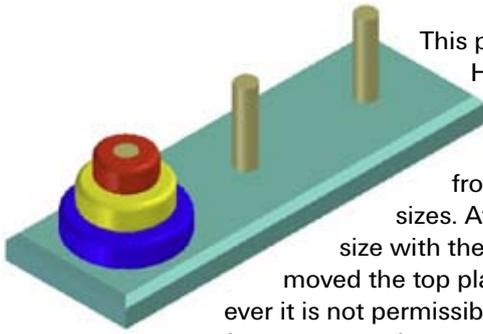
Task 1:
 The robot should pick up a barrel with its gripper and put it back down at a different position.



Task 2:
 The robot should stack three barrels to form a tower. Then it should remove the barrels one by one from the tower and rebuild the tower at a different position.



Towers of Hanoi



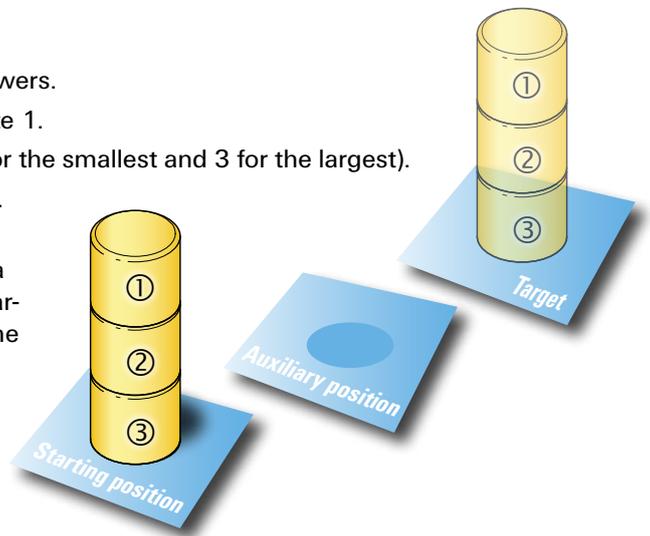
This puzzle was invented in 1883 by the French mathematician Édouard Lucas. He made up the following story:

"A monk in a cloister in Hanoi was assigned the task of moving 64 plates from one tower to another. There were 3 towers and 64 plates, all of different sizes. At the beginning all of the plates were in one tower arranged according to size with the largest plate at the bottom and the smallest at the top. Each time a plate is moved the top plate in any desired stack can be moved to one of the other two towers. However it is not permissible for a smaller plate to already be present there. This means that the plates are always arranged according to size in each tower at all times.

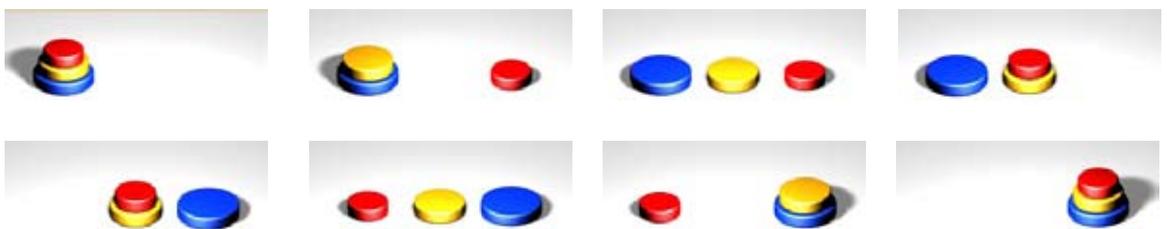
To make this puzzle a little simpler, use 3 barrels instead of 64 plates. Since the barrels are all the same size, you can label them with stickers with the numbers 1–3.

Here are the rules:

- There are three construction sites for barrel towers.
- A tower with three barrels 1–3 is located at site 1.
- The barrels have different "sizes" (1 standing for the smallest and 3 for the largest).
- The tower must be moves from site 1 to site 3.
- It is only permissible to move one barrel - the top one – at a time. It is never permissible for a "larger" barrel to be located above a smaller barrel (e.g. the 2 should NEVER be above the 1, the 3 NEVER above the 2).



The solution for 3 barrels is shown in the figures below.



Loading teach-in sequence on TX controller

The program Teach-Player_TX.rpp allows you to load a sequence created with the teach-in program on the TX controller, so that it can be run without the PC. First load the saved .csv file in the Teach-In-Player. Then load the program on the ROBO TX Controller. The sequences is then performed automatically in download mode.

Teach-Player_TX.rpp