1 Introduction – controlling fischertechnik models with ROBO Pro

You must have asked yourself at some time how it works when robots carry out their allotted tasks as if controlled by an invisible hand. But it's not just with actual robots, but in many other fields as well, that we encounter control and automation technology. Including fischertechnik. By the next chapter but one, we will be designing a little control program for an automatic garage door together, and in doing so we'll learn how control problems like this can be solved and tested with the help of ROBO Pro software for Windows. ROBO Pro is also very simple to operate. Control programs, or more precisely flow charts and later data flow charts, as we shall learn, can be created on the graphical user interface, almost exclusively using the mouse.

In order to be able to control your fischertechnik models through your PC, you will need, as well as the ROBO Pro control software, an Interface to connect the computer with the model. It transforms the software commands so that, for example, motors can be controlled and sensor signals can be processed. The ROBO TX Controller (item number 500995), the earlier ROBO Interface (item number 93293) and the Intelligent Interface (item number 30402) are available from fischertechnik. You can use any of these Interfaces with ROBO Pro. But ROBO Pro only supports the online mode of the Intelligent Interface. ROBO Pro no longer supports the very early parallel Interface (item number 30520).

A few words about the layout of this manual. It is divided into two parts. The first part, from Chapter 1 to Chapter 4, describes the basic procedure for programming with ROBO Pro. This gives you a lot of information and background knowledge about programming in general and about how to use the ROBO Pro software.

The second part consists of Chapters 5 to 7, and gives an introduction to the functions needed for more advanced programs.

Chapters 8 onwards are more for reference. So when you're familiar with the operation of ROBO Pro after reading the first part and you need very specific information, here is where you will find comprehensive explanations of the individual program elements.

> If you are already familiar with ROBO Pro and only want to find out what new features were added with the ROBO TX Controller, you should read only chapters 11 through 13 of the manual.

So let's go! You must already be itching to know what possibilities ROBO Pro gives you for programming your fischertechnik models. Have fun!

1.1 Installation of ROBO Pro

System requirements for installing ROBO Pro are:

- an IBM-compatible PC with Pentium II processor with a clock speed of at least 500 MHz, 64 MB RAM and ca. 40 MB free disk-space on the hard drive
- a monitor and a graphics card with a resolution of at least 1024x768 pixels. With CRT monitors the refresh rate should be at least 85 Hz to maintain a flicker-free image. TFT flat screens provide a flicker-free image at any refresh rate, so that the refresh rate is not critical with TFT flat screens.

- Microsoft Windows, Version Windows XP or Vista
- A free USB interface to connect the ROBO TX Controller. To connect the ROBO Interface you need a free USB interface or a free RS232 interface (COM1 to COM4).

First of all, of course, you must start the computer and wait until the operating system (Windows) has finished loading. The ROBO Interface should only be connected to the computer after successful installation. Insert the installation CD into the CD-ROM drive. The installation program on the CD will then be started automatically.

- In the first Welcome window of the installation program you push the **Next** button.
- The second window, **Important Notes**, contains important up-to-date notes about installing the program or about the program itself. Here too, you click on the **Next** button.
- The third window, **License Agreement**, displays the ROBO Pro licensing contract. You must click **Yes** to accept the agreement before you can proceed to the next window with **Next**.
- In the next window, **User Details**, please enter your name.
- The next window, **Installation Type**, allows you to choose between **Express Installation** and **Customized Installation**. With customized installation, you can choose to leave out individual components of the installation. If you are installing a new version of ROBO Pro over an older version, and you have modified some of the sample programs in the older version, you can exclude the sample programs from the customized installation. If you don't do this, the modified sample programs will be ***overwritten without warning***. If you select customized installation and press **Next**, an additional window, allowing you to select the components, will appear.
- In the **Target directory** window you can select the folder or directory path where you want the ROBO Pro program installed. This will normally be the path C:\Program Files\ROBOPro. However, you can also enter another directory.
- When you push the **Finish** button in the last window, the installation is performed. As soon as the installation is finished – this normally only takes a few seconds – the program announces successful installation. If there are problems, an error message is displayed, which should help you to solve the problem.

1.2 Installing the USB driver

This step is required if the ROBO TX Controller or the ROBO Interface is to be connected to the USB port. The ROBO Interface can also be connected to one of the serial ports COM1-COM4. The earlier Windows versions Windows 95 and Windows NT4.0 don't support USB ports. With Windows 95 or NT 4.0, the ROBO Interface can only be connected via the serial port. There is no need to install a driver in this case.

Important note for the installation under Windows 2000, XP and Vista:

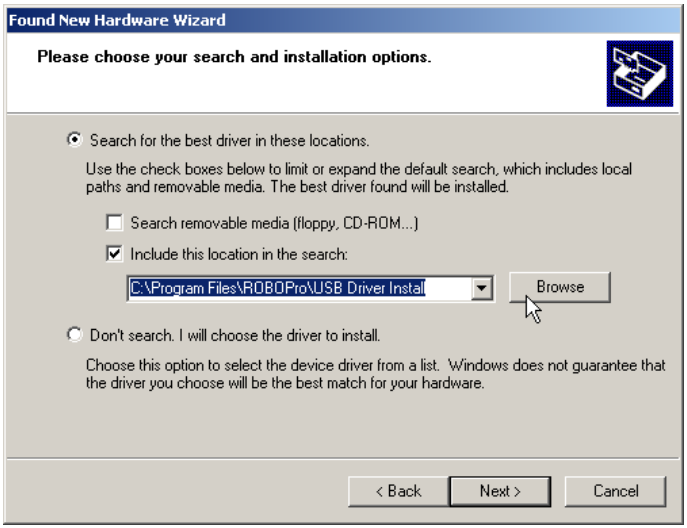
The USB driver can only be installed by a user with PC systems administrator privileges. Should the installation program advise you that you are not permitted to install the USB driver, you must either ask your system administrator to install your driver or install ROBO Pro without this driver.

In order to install the USB driver, you must first connect the ROBO TX Controller or the ROBO Interface with a USB cable to your computer and supply it with power. Windows recognizes automatically that the Interface is connected and displays the following window:

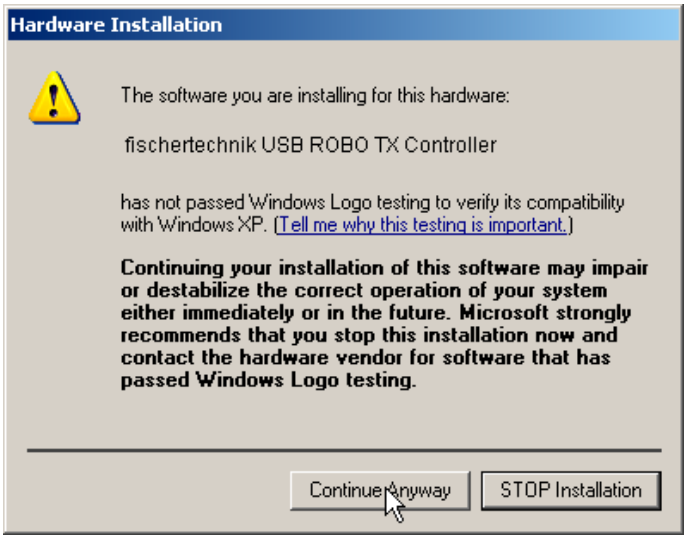
Depending on the operating system, the appearance of the window can be somewhat different from the illustration above!

Here you must select **Install software from a list or specific source** and press **Next**.

In the next window you deactivate **Search removable media** and activate **Also search following sources**. Then you click **Search** and select the sub-directory ***USB Driver Installation*** in the directory in which ROBO Pro is installed (the standard directory is C:\Program Files\ROBOPro\). For the ROBO TX Controller you first select the sub-directory ***TXController,*** for the ROBO Interface the sub-directory ***ROBOInterface,*** and then the sub-directory containing the driver for your operating system, for example ***WinXP***.



Under Windows XP, you may see the following message after pressing **Next**:



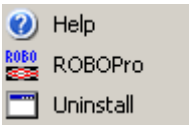
The USB driver is still being tested by Microsoft. Once testing is completed the driver will be approved by Microsoft, so that this notice no longer appears. In order to install the driver, press **Proceed with installation**.

Finally, the following message will appear:

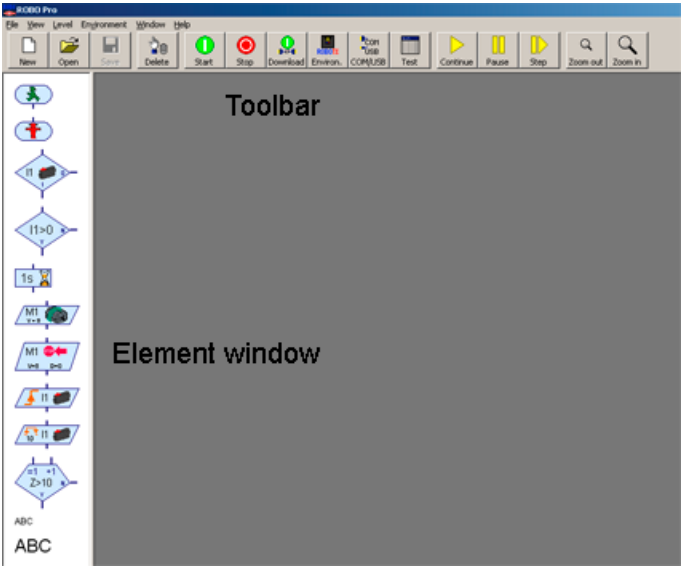
Press **Finish** to complete USB driver installation.

1.3 First Steps

Curious? Then simply start the program ROBO Pro. To do this, you click on the Start button on the task bar and then select **Programs** or **All programs** and **ROBO Pro**. In this folder of the Start menu you will find the following entries:



The Uninstall entry allows you to uninstall ROBO Pro. The Help entry opens the ROBO Pro Help file, and the ROBO Pro entry opens the ROBO Pro program. Now select the entry **ROBO Pro** to launch the program.

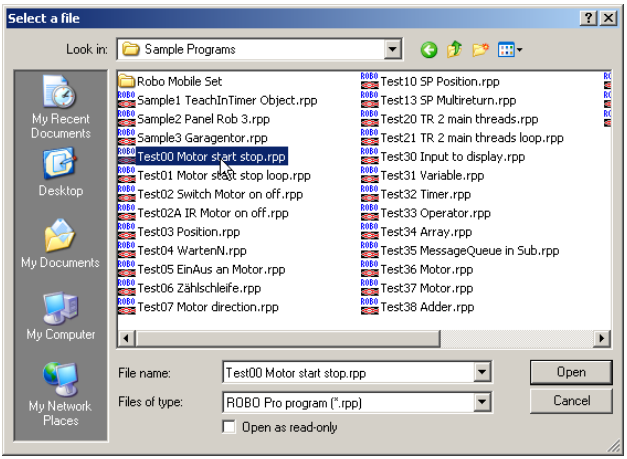


The window has a menu bar and toolbar with various operating buttons above as well as a window on the left-hand side with program elements. If you see two stacked windows in the left margin, ROBO Pro is not set on **Level 1**. To allow the functionality of ROBO Pro to match your growing knowledge, you can set ROBO Pro from Level 1 for beginners up to Level 5 for experts. Look in the **Level** menu to see whether there is a checkmark by **Level 1: Beginners**. If not, please switch to level 1.

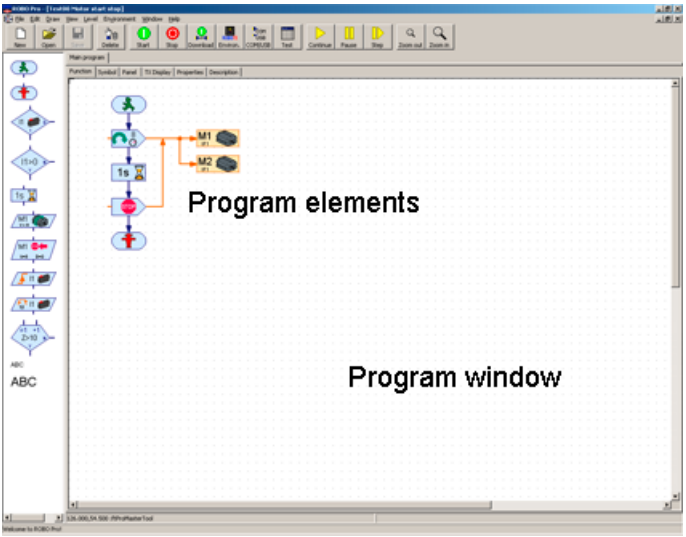
ROBO Pro is configured to use the ROBO TX Controller as interface. You can see this by the presence of the button ROBO TX in the toolbar. In *Chapter 11.2 Environment* you learn how you can switch to the earlier ROBO Interface and what you need to pay attention to.


Open

Now you may either create a new program file or open an already existing program file. We do not intend to create a new program file until Chapter 3, when we will write our first control program. To familiarize ourselves with the user interface, we shall open an already existing sample program. To do this, you click the entry ***Open*** in the ***File*** menu, or use the ***Open*** button in the toolbar. The sample files are found in the folder ***C:\Program Files\ROBOPro\Sample Programs***.



Open the file **\Level3\Motor start stop.rpp**:



Here you can see what a simple ROBO Pro program looks like. In programming, control-program flow charts are created in the program window using program elements from the element window. The finished flow charts can then be checked before being tested using a connected fischertechnik Interface. But not too fast: we shall learn programming step-by-step in the following chapters! Having thus gained your first impression of the user interface, you close the program file using the

Close command in the **File** menu. You can answer **No** to the question of whether you want to save the file.

2 A quick hardware test before programming

Clearly, the Interface must be connected to the PC for us to be able to test the programs we will later create. But, depending on the Interface used (ROBO TX Controller or ROBO–Interface), appropriate interface connection settings must also be made and tested. We sill do this in the coming chapter.

2.1 Connecting the Interface to the PC

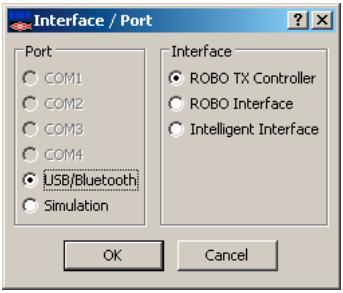
This should not be a great problem. The connecting cable supplied with the Interface is connected to the Interface and to a port on the PC:

- With the ROBO TX Controller a USB port can be used
- With the ROBO Interface (item number 93293) either a USB port or one of the serial ports COM1 to COM4 can be used.

The connections for these ports are normally found on the back of the computer. The exact placement of the various connections is described precisely in the user manual of your PC; please look it up there. USB connections are also often found on the front of a PC. Don't forget to give the Interface a power supply (mains unit or battery). The individual connections of the Interface are described in detail in the user manual of the respective equipment.

2.2 Getting the right connection – Interface settings

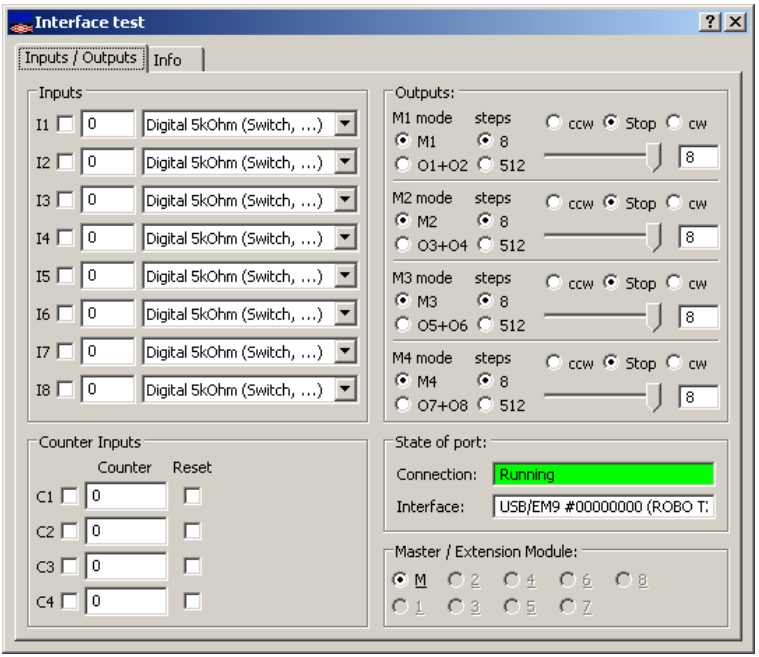
For the connection between the Interface and the PC to function correctly, ROBO Pro must be configured for the Interface currently in use. To do this, start ROBO Pro using the **ROBO Pro** entry on the Start menu under ***Programs*** or ***All programs*** and ***ROBO Pro***. Then push the **COM/USB** on the toolbar. The following window will appear:



Here you can select the port as well as the Interface type.


Test

Once you have selected the appropriate settings, close the window with **OK**. Now open the Interface test window with the **Test** button on the toolbar.



It shows the inputs and outputs available on the Interface. The green bar in the lower left of the window displays the connection status of the PC to the Interface:

- **Connection: Running** confirms correct connection to the Interface
- **Connection: Stopped** indicates that the connection has not been correctly set up and the PC was unable to establish a connection to the Interface. In this case, the bar will appear red.

To be able to change the Interface or connection settings, you must close the Test window (with the **X** in the upper right) and select another port or another Interface type as previously described, via the **COM/USB** button in the toolbar.

If you have been able to set up the connection between PC and Interface as described and the green bar appears, you will be relieved to know you can skip the next section.

If not, perhaps the tips in the next section can help you out.

2.3 Wrong connection: no connection to the Interface!?

If you get the message **Stopped** with your interface despite having correctly set the port (see above), you should check the following points. For this purpose, you may need to get advice from a computer expert:

- ***Power supply:***
  Does the Interface have an appropriate power supply? If you are using disposable or rechargeable batteries as power supply, the possibility arises that these are flat and no longer

supply sufficient voltage. If the battery voltage falls below 6 V, the ROBO TX Controller's processor may stop working. In this case the display will stop showing any information. If the voltage is too low, you must replace or, where appropriate, recharge the batteries, or, if possible, test the Interface with a mains power supply.

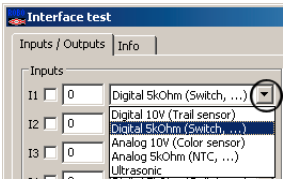
- ***Has the USB driver been installed correctly?***
  You can find this out by checking in the Device Manager in the Windows Control Panel whether the entry fischertechnik USB ROBO TX Controller appears under connections (COM and LPT) and functions properly. Should this entry not appear, install the USB driver again. If an error appears, uninstall the driver (click on the respective entry with your right mouse button) and install it once again.
- Is there a conflict with another device driver on the same port (e.g. a modem)? This driver may need to be deactivated (see Windows or device handbooks).
- If you still can't establish a connection to the Interface, then probably either the Interface or the connection cable is faulty. In this case, you should consult fischertechnik Service (Address: see menu: **"?"** / **Information about**).

2.4 Is everything working – the Interface test


Test

Once the connection has been correctly set up, you can use the Interface test to test the ROBO TX Controller and the models connected to it. The test window displays the various inputs and outputs of the Controller:

- ***Universal inputs I1—I8***
  I1—I8 are the universal inputs of the ROBO TX Controller. This is where different types of sensors can be connected. There are digital and analog sensors. You set the universal inputs depending on the type of sensor you would like to connect.



  - ***Digital sensors*** can only assume the states 0 and 1, or Yes and No. By default, both universal inputs are set to the input type Digital 5kOhm. Switches (mini pushbutton-switches), as well as phototransistors (light sensors) or reed-switches (magnetic sensors), can be connected to these digital inputs.
    You can check the functioning of these inputs by connecting a mini-sensor (item number 37783) to the Interface, e.g. to I1 (use contacts 1 and 3 on the switch). As soon as you press the button, a check-mark appears in the display of I1. If you have connected the switch the other way around (contacts 1 and 2), the check-mark will appear straight away and disappear when you press the button.
- The setting **Digital 10V** is used fort he infrared trail sensor.
- The setting **Analog 10V** can be used for the color sensor or to measure voltages between 0 and 10V such as the supply voltage of the battery pack. The voltage is displayed in mV (millivolt).
- **Analog 5kOhm** is used for the NTC resistor to measure temperatures and for the photoresistor to measure light. Here the reading is displayed Ohm (Ω = electrical resistance).
- The setting **Distance** is used for the ultrasound distance sensor (for the ROBO TX Controller only the version TX of the distance sensor with 3 pin connection cable, item number 133009, can be used).

- ***Counter inputs C1-C4***
  These inputs allow you to count fast pulses with frequencies of up to 1000 pulses per second. You can also use them as digital inputs for buttons (not suitable for the trail sensor). If you connect a button to this input, every push of the button (=pulse) will increase the value of the counter by 1. This allows you, for example, to let a robot travel a specific distance.
- ***Motor outputs M1—M4***
  M1 – M4 are the outputs from the Interface. This is where what are called actuators are connected. These can be, e.g., motors, electromagnets or lamps. The 4 motor outputs can be controlled in speed and in direction. Speed is controlled using the slide control. You can choose between a coarse resolution with 8 different steps of speed or a fine resolution with 512 steps. The program elements in levels 1 and 2 only use the coarse resolution, but starting with level 3, there are elements which allow you to use the fine resolution. The speed is displayed next to the slider control as a number. If you would like to test an output, you connect a motor to an output, e.g. M1.
- ***Lamp outputs O1—O8***
  Each motor output can alternatively be used as a pair of individual outputs. These can be used to control not only lamps, but also motors which only need to move in one direction (e.g. for a conveyor belt). If you would like to test one of these outputs, you connect one lamp contact to the output, e.g. O1. You connect the other lamp contact with one of the ground sockets of the R (⊥).
- ***Extension modules***
  The ROBO TX Controller connected to the PC via the USB port (=master) can take up to 8 additional ROBO TX Controller as extensions (see manual ROBO TX Controller). These buttons allow you to select which of the connected devices you would like to access with the test window.

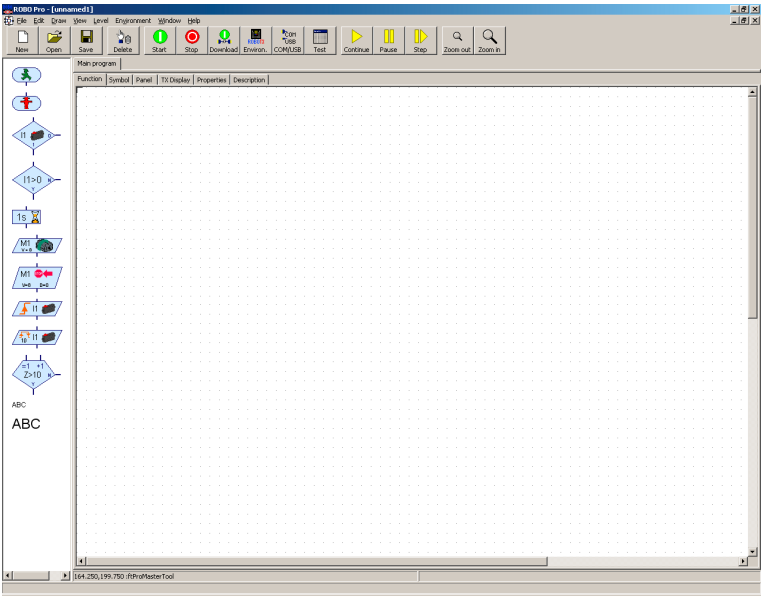
3 Level 1: Your first control program

After testing the hardware, that is the Interface and the switches and motors connected to it, in Chapter 1, we'll now get down to programming. But what does "programming" actually mean? Well, just imagine that, for example, a robot is connected to our Interface. But this robot is so stupid that it can't do anything on its own. Luckily, we're a bit smarter than that. We can tell the robot exactly what to do. How? Well, what happened in the last chapter when we used the mouse button to set the motor output M1 on "left"? Right, we switched the motor on. If, for example, this motor were to drive the gripping claw of our robot, we would have done nothing else than to say to the robot: "Grip the object!" But now we don't want to initiate every step by hand; rather the robot should do this automatically. To achieve this, we must store the individual steps to be carried out, so that the robot can work through them one after another, i.e., we must create a program, which will control the robot on our behalf. Logically enough, the technical term for this is a control program.

3.1 Creating a new program


New

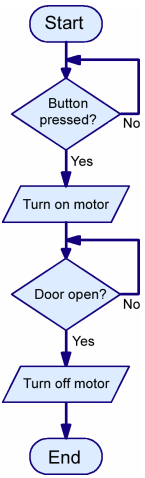
The ROBO Pro software gives us a great tool to design these control programs and to test them with the aid of a connected Interface. Don't worry: we're not about to program the robot straight away. We shall content ourselves initially with simple control tasks. To do this we must create a new program. In the toolbar you will find the entry **New**. If you left-click on it with your mouse, a new, empty program is created.



Now you see a large white drawing surface, in which you will enter your first program. If you see two stacked windows in the left margin, please switch to **Level 1: Beginners** in the **Level** menu.

3.2 The elements of a control program

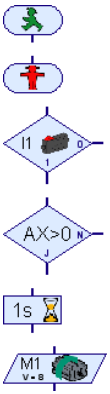
Now we can set about creating our first control program. We shall do this on the basis of a concrete example:



Functional description:

Imagine a garage door that can be opened automatically. Maybe you've even got one at home! You arrive at the garage in your car and, with the push of a button on the transmitter, the door, driven by a motor, is opened. The motor must keep running until the garage door is completely opened.

Words are a rather cumbersome and not very graphic way to describe a control program. So what we call ***flow charts*** are used to represent the sequence of actions to be performed and the conditions that need to be fulfilled for these actions. In the case of our control system, the condition for the action "switching on motor" is that the button be pressed. It is easy to read one of these flow charts: just follow the arrows step-by-step! These show exactly how the control system works – the individual steps can only be carried out in the order given by the arrows, never in any other way. Otherwise it wouldn't be worth going to all the trouble, would it?



Using our ROBO Pro software, we can now draw precisely this flow chart and in so doing create the ***control program*** for the connected hardware (Interface, motors, switches, etc.). The software does the rest, which, as it happens, is just the way it is with large industrial applications too! So we can concentrate fully on the creation of the flow chart.

You put the flow chart together from program elements. Another new concept? Don't worry! In ROBO Pro the individual elements that are put together to form a flow chart are called program elements. The action "switch on motor" means just that: the Interface should actually switch on the motor that is connected to it! You will find the available program elements in the element window on the left-hand side.

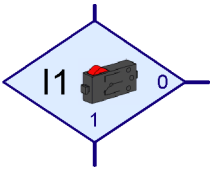
3.3 Inserting, moving and modifying program elements

Now it's a matter of creating a flow chart for our garage door control system from the program elements contained in the element window. All available program elements can be fetched from the element window and inserted in the program window.

Inserting program elements.

You move the mouse onto the symbol for the desired program element and left-click on it once. Then you move the mouse into the program window (that's the large white area) and click once again. You can also drag the program element into the Program window while holding down the mouse button. A program always begins with a Start element. The Start element is the rounded element with the little green GO man. It would be best to try this out straight away with this program element: Left-click once on the Start element in the element window, move the mouse up into the program window and once there left-click once more.

The next element in the program flow chart queries an input and then branches to one path or another depending on its state. In the element window, click on the element depicted right and then move the mouse below the previously inserted Start element. If the upper input of the Branch element is one or two grid points below the exit of the Start element, a connecting line will appear in the program window. If you left-click again, the Branch element is inserted and automatically connected with the Start element.



Moving program elements and groups

A program element can be moved to the desired position after insertion while holding down the left mouse button. If you want to move several elements as a group, you can start by drawing a frame around the elements while holding down the left mouse button. To do this you have to left-click in an ***empty*** zone, keep the button pressed and use the mouse to draw a rectangle containing the desired elements. The elements in the rectangle are now displayed with a red border. If you now move one of the red elements with the left mouse button, all the red elements are moved. You can also mark individual elements red by left clicking on them while holding down the shift key (i.e. the upper/lower case key). If you left click in an empty zone, all the red-marked elements will be displayed normally again.

Copying program elements and groups

Copying program elements and groups can be done in two ways. You can proceed exactly as for moving, except that you press the **CTRL** key on the keyboard before moving the elements. In this way the elements are not moved, but copied. However, with this function you can only copy elements within a program. If you want to copy elements from one program to another, you can use the Windows ***clipboard***. First select some elements, as described in the previous section in the case of moving elements. If you now hit **CTRL+C** on the keyboard or click on **Copy** on the **Edit** menu, all the selected elements will be copied onto the Windows clipboard. Now you can change over to another program and re-insert the elements there with **CTRL+V** or **Edit / Paste**. Once elements are copied, you can also paste them in several times. If you want to move elements from one program to another, you can use **CTRL+X** or **Edit / Cut** function at the beginning instead of **CTRL+C** or **Edit / Copy**.

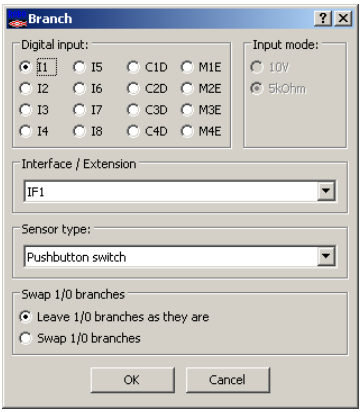
Deleting elements and Undo function


Delete

It is quite simple to delete elements. You can delete all the elements marked in red (see previous section) by pressing the “delete” key (**Del**) on the keyboard. You can also delete individual elements with the Delete function. To do this, first click on the button in the toolbar like the one illustrated and then on the element you want to delete. Try it out now. Then you can redraw the deleted element. But you can also retrieve the deleted element using the **Undo** function in the **Edit** menu. By using this menu item you can undo any changes to the program.

Editing program element properties

If you ***right*** click on a program element in the program window, there will appear a dialog window, in which you can change the element's properties. The Properties window for a Branch element is illustrated on the right.

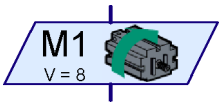


- Buttons **I1** to **I8** allow you to enter which of the Interface's inputs is to be queried. The inputs C1D-C4D correspond to counter inputs if you use them as digital inputs. We will deal with the inputs M1E-M4E later.
- The additional inputs C1D-C4D and M1E-M4E are covered in the reference section in 8.1.3 *Digital Branch* on page 61.
- The selection **Interface / Extension** is not explained until Chapter 7 *Controlling several Interfaces* on page 56.
- Under **Sensor type** you can select the sensor connected to the input. Digital inputs are mostly used with push-button sensors, but often also with phototransistors or reed-contact switches. Selecting the sensor automatically sets the required input type for the universal inputs I1-I8 of the ROBO TX Controller.
- Under **Interchange 1/0 connections** you can interchange the positions of the 1 and 0 exits of the Branch element. Normally the 1 exit is below and the 0 exit is on the right. But sometimes it's more practical to have the 1 exit on the right. Press on **Interchange 1/0 connections** and the 1 and 0 connections will be changed over as soon as you close the window with **OK**.

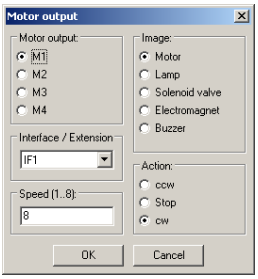
Hint: If you connect a mini-sensor as a "closer", using connections 1 and 3 of the switch, the program branches to the "1" branch if the switch is depressed, and otherwise to the "0" branch.

If you connect a mini-sensor as an "opener", using connections 1 and 2 of the switch, the program branches to the "1" branch if the switch is depressed, and otherwise to the "0" branch.

The next element in our garage door control system is a Motor element. Insert it into the program as you did with the previous two elements, this time under the Branch element. It is best to place the element in such a way that that it is automatically connected to the element above.



The Motor element allows you to switch on or off either a motor, or a lamp or an electromagnet. Again, you open the Properties window for the Motor element by right-clicking on the element.



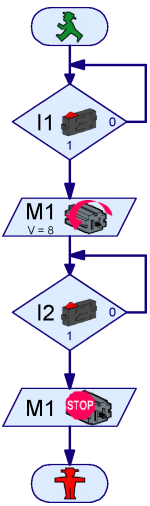
- You can choose which of the Interface's outputs to control by means of buttons **M1** to **M4**.
- Under **Image** you can choose an image to represent the fischertechnik component connected to the output.
- We will deal with the selection **Interface / Extension** when we get to Chapter 77 *Controlling several Interfaces* on page 56.
- Under Action you can select how the output is to be affected. You can start a motor with direction left (counterclockwise) or right (clockwise) or stop it. You can switch a lamp on or off.
- Under **Speed/Intensity** you can set the speed at which a motor is to operate, or how brightly a lamp should glow. Possible values are 1 to 8.

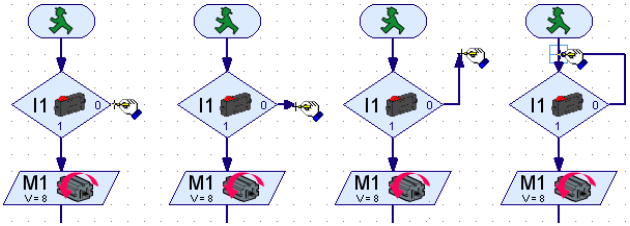
For our flow chart we need the command ***Motor M1 left with speed 8***.

3.4 Linking program elements

Now that you know how to insert elements into a control program, we can get on with the job of completing our control program. Think back to the functional description of the garage door control system: is there still something missing? Right: we may have turned the motor on by pushing the button, but once the door is opened, the motor must be automatically switched off again! In practice, this is done with the so-called "end switch". This is a sensor fitted to the garage door in such a way that it is operated the moment the motor has fully opened the door. As in the case of switching on the motor, this signal can be used to switch it off again. To query the end switch we can use the Branch element again.

So insert another Branch element into the program, one which will check the end switch on input I2. Don't forget to left-click on the element and to set the input to I2. As soon as the garage door is open and the end switch has been pressed, the motor should stop again. This will be achieved using a Motor element. Start with the same element we used to switch on the motor. If you right-click on the element, you can change the function of the element to **Stop motor**. The program is finished off with an End element. Your program should now look almost like the illustration on the right. If you have placed the elements under one another with a separation of one or two grid points, most of the entries and exits will be connected with program flow arrows. But the No (N) exit of the two Branch elements is not yet connected. As long as the switch on input I1 has not been pressed, the program should go back and query the switch again. To draw this line, click with the mouse successively on the places shown in the diagram below.





Hint: It a line should ever not correctly joined to a connection or another line, this will be indicated by a green rectangle at the point of the arrow. In this case you have to create the connection by shifting the line or by deleting it and drawing it again. Otherwise the program flow will not work at this point.

Deleting program flow lines

Deleting lines works exactly like deleting program elements. Simply left-click on the line, so that it gets marked in red. Now click on the delete (**Del**) key on the keyboard to delete the line. You can also select several lines, if you hold down the shift key (that's the key for shifting between upper and lower case) and then left-click on the lines in succession. Apart from this, you can also mark several lines by drawing a frame around them. Now you can delete all the red-marked lines at once by pressing the **Del** key.

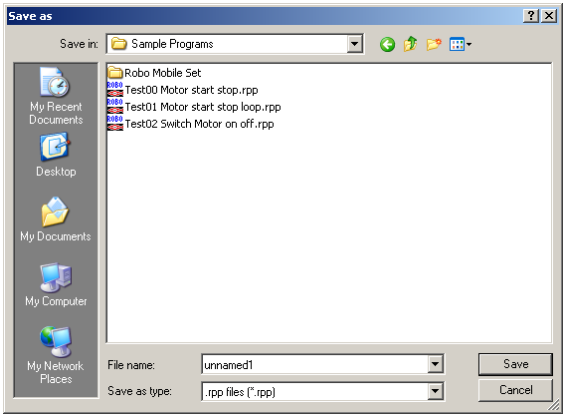
3.5 Testing your first control program

To test our first control program, you should build a little model. To do this, it is enough to connect a switch to I1 and to I2 on the Interface, as well as a motor to M1.

Note: Connecting the Interface to the PC and establishing Interface settings has already been covered in the previous chapter, which you can refer back to for details.

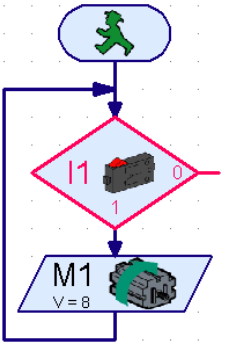
Before testing the program, you should save the program file on the hard drive of your computer. Click on the command **Save as** on the **File** menu. The following dialog window will then appear.

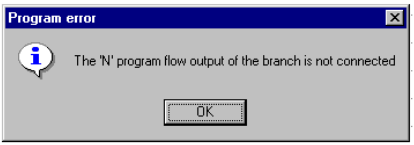
Under “Save in”, choose the directory in which you want to save the file. Under “Filename”, enter a name not yet in use, e.g. GARAGE DOOR and confirm by left-clicking on “Save”.



Start

To test the program, push the start button (shown left) in the toolbar. First, ROBO Pro will test whether all the program elements are properly connected. Should an element not be correctly connected or something else not be in order, it is marked in red, and an error message is displayed describing what is not right. If, for example, you have forgotten to connect the No (N) exit of a program branch, it will look like this:

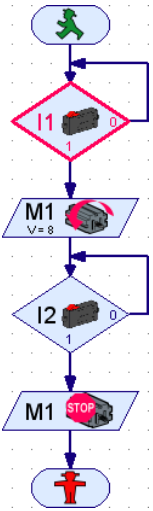




If you have received an error message, you must first of all correct the reported error. If you do not, the program will not be started.

Note: You will find a full explanation of this mode of operation and of “Download Operation” in Section 3.7, on page 24.

The first Branch element will be marked in red. This shows that the program is waiting at this element for an event, namely the pressing of the button on I1, which is supposed to open the garage door. As long as the switch on input I1 has not been pressed, the program takes the No (N) alternative of the branch and goes from there back to the beginning of the branch again. Now press the switch connected to input I1 of the Interface. This fulfils the condition for proceeding, and the motor is switched on. In the next step, the program waits for the end switch on input I2 to be pressed. As soon as you operate the end switch on I2, the program branches to the second Motor element and switches the motor off again. Finally the program arrives at the program end. A message will appear saying that the program has been terminated.

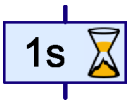


Did everything work? Congratulations! That means you've created and tested your first control program. If it doesn't work properly—don't give up, just check through everything carefully again; there must be a mistake hidden in there somewhere. Every programmer makes mistakes, and making mistakes is the best way to learn. So keep your chin up!

3.6 Other program elements

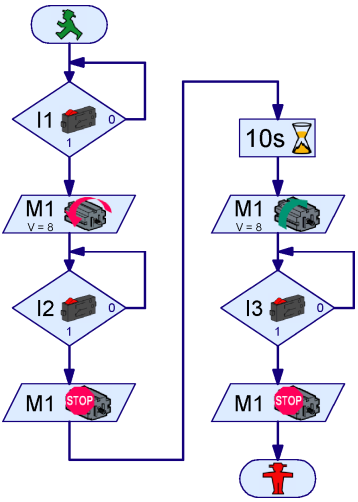
If you have tried your first control program on a real model garage door, the door will now be open. How can we close it again? Of course we can start the motor again by pushing a button! But we want to try another solution, and learn about a new program element in the process. To do this, you start by saving the program under a new name (we will need the current flow chart again later). Use the menu item **Save as ...** in the **File** menu to do this, entering an as yet unused filename.

3.6.1 Time delay

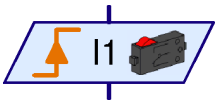


Before we can extend the flow chart, you have to delete the connection between "switch off motor" and "Program end" and shift the End element down. Now you can insert the new program elements between these two elements. The garage door is to be closed automatically after a period of 10 seconds. To do this you can use the **Time delay** program element illustrated right. Within a broad range, you can set the waiting time as you wish, as usual by right-clicking on the element. Enter the desired time delay of 10 seconds. To close the garage door, the motor must of course go the other way, that is, to the right clockwise). The motor is turned off by another end switch on I3.

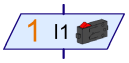
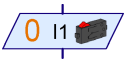
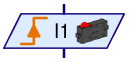
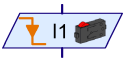
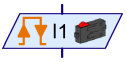
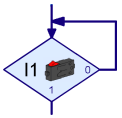
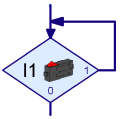
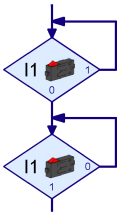
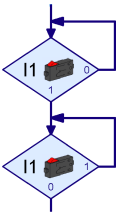
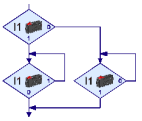
The finished flow chart should look roughly as presented on the right. For the sake of presentation, the new program elements have been moved to the right. Once there are no more mistakes in the flow chart, you can test the extended garage door control system as usual with the **Start** button. The motor is switched on by operating the switch on I1, and switched off again by operating I2. This is how the garage door is opened. Now the Time delay program element has a red border for 10 seconds, that is the delay time we set. Then the motor is switched on to turning the other direction until the switch on I3 is operated. You should also try changing the delay time.



3.6.2 Wait for input

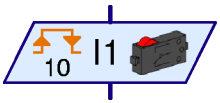


Alongside the Time delay element there are another two elements that wait for something before allowing the program to proceed. The **Wait for Input element**, depicted left, waits until one of the Interface's inputs is in a particular state of has changed in a particular way. There are 5 variants of this element.

Symbol					
Wait for	Input=1 (closed)	Input=0 (open)	Change 0-1 (open to closed)	Change 1-0 (closed to open)	Any change (1-0 or 0-1)
Same function using Branch alone					

A combination of Branch elements could be used instead, but the **Wait for Input** element makes things simpler and easier to understand.

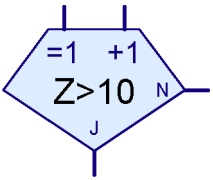
3.6.3 Pulse counter



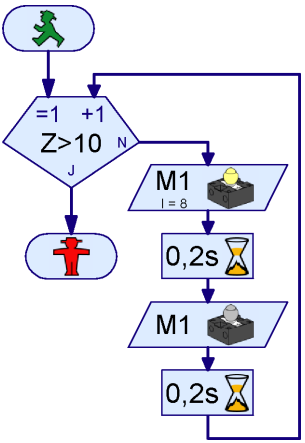
Many fischertechnik model robots also use pulse wheels. These gear wheels operate a switch four times for every revolution. With these pulse wheels you can turn a motor on for a precisely defined number of revolutions rather than for a given time. To do this, you need to count the number of pulses at an input of the Interface. For this purpose there is the ***Pulse counter element***, depicted left, which waits for a user-definable number of pulses. In the case of this element, too, you can set whether any alterations or only 0-1 or only 1-0

changes are regarded as pulses. With pulse wheels, one normally waits for changes in either direction, so that a resolution of 8 steps per revolution is obtained with 4 gear teeth.

3.6.4 Counter loop



With the Counter Loop element you can very easily have a specific part of the program executed several times. The program illustrated, for example, turns a lamp on M1 on and off again 10 times. The Counter Loop element has a built-in counter. If the counter loop is entered via the **=1** entry, the counter is set to 1. If the counter loop is entered via the **+1** entry, 1 is added to the counter. According to whether the counter is greater than a value you have prescribed, the counter loop branches to the Yes (Y) or No (N) exit. So the Yes exit is used when the loop has been traversed as many times as you specified in the counter value. If further passes through the loop are needed, on the other hand, the counter loop branches to the No exit. As in the case of the Branch element, you can also swap the Yes and No exits through the property window.



3.7 Online and download operation—what's the difference?


Start

So far we have tested our control programs in what is called **online operation**. In this way you were able to follow the progress of the program on the screen, because the currently active element was marked in red on the screen. You use online operation to understand programs or to look for errors in programs.


Pause

In online operation you can also stop the program and continue it again by pressing the **Pause** button. This is very practical if you want to investigate something about your model without stopping the program altogether. Also, if you are trying to understand the way a program runs, the Pause function can be very helpful.


Step

With the **Step** button, you can execute the program in individual steps, element by element. Every time you press the Step button, the program goes to the next program element. If you execute a **Time Delay** or **Wait for** element, it can of course take a while for the program to get to the next element.


Download

For your ROBO TX Controller you can also use **download operation** instead of online operation. In online operation programs are executed by your computer. In this mode, it sends control commands such as "switch on motor" to the Interface. For this, the Interface needs to be connected to the computer for as long as the program is running. On the other hand, in download operation the program is executed by the Interface itself. Your computer stores the program in the ROBO TX Controller. As soon as this has been done, the connection between the computer and the Interface can be broken. Now the Interface can execute the control program independently of the computer. Download operation is important for example in programming mobile robots, for which a connecting cable between PC and robot would be very cumbersome. Even so, control programs should initially be tested in online operation, as possible errors are more easily found here. Once fully

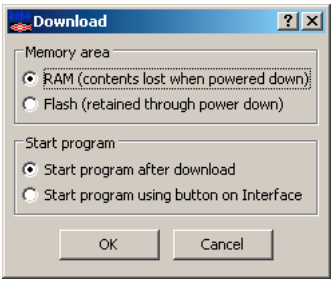
tested, the program can be downloaded onto the ROBO TX Controller. The problematic USB cable can be replaced by a Bluetooth connection. In that way the model has unrestricted mobility even in online operation (see manual ROBO TX Controller).

But online operation also has advantages compared with download operation. In comparison with the Interface, a computer has much more working memory, and can calculate much faster. This is an advantage with large programs. Also, during online operation a ROBO TX Controller and a ROBO Interface can be controlled simultaneously from a program.

An overview of the two modes of operation

Mode	Advantage	Disadvantage
Online	Program execution can be followed on screen Execution, even of large programs, is very fast Simultaneous control of a ROBO TX Controller and a ROBO Interface possible The earlier Intelligent Interface is supported Panels can be used The program can be stopped and continued	Computer and Interface must remain connected
Download	Computer and Interface can be separated after download	The earlier Intelligent Interface is not supported Program execution cannot be followed on screen

Using download mode



If you have the ROBO TX Controller or the ROBO Interface, you can transfer the garage door control system to the interface by means of the **Download** button. First the dialog window on the left is displayed. The ROBO Interface has several program storage areas, a ***RAM*** (Random Access Memory) area and two ***Flash*** memory areas. A program in RAM is lost as soon as you disconnect the Interface from the power supply or the battery pack is discharged. A program stored in Flash memory, on the other hand, will remain stored in the Interface, even without power, for years. Of course you can nevertheless overwrite programs in Flash memory at any time. Download to RAM, however, is distinctly faster, and is therefore recommended for testing purposes.

You can store multiple programs, for example multiple behavior modes for a mobile robot, in the Flash memory. You can select, start and stop the multiple programs by using the display and the selection keys of the ROBO TX Controller. If the ***Start program after download*** option is active, the program is started immediately after download. To stop the program you press the left selection key on the TX Controller.

For mobile robots, the option **Start program with key on Interface** makes more sense. This is because, if you don't have a Bluetooth interface, you still have to unplug the USB cable before your program sets the robot in motion. In this case, you start the downloaded program by using the left selection key of the TX Controller.

By using the function Autostart of the ROBO TX Controller, a program is started automatically as soon as the Interface is supplied with power. In this way, you can for example you can supply the Interface with power via a mains adapter with a time switch, and start the program every day at the same time. Then you don't have to either leave the Interface permanently switched on or start the program with the selection key every time you switch it on.

Note:

You can also find a comprehensive description of the functions of the ROBO TX Controller in the accompanying operating manual.

3.8 Tips and Tricks

Altering connection lines

If you shift an element, ROBO Pro will try to adjust the connecting lines in a reasonable way. Should you not like an adjusted line, you can easily change the connecting lines by left-clicking on the line and moving it while holding the key down. According to where the mouse is placed on the line, a corner or an edge of the line is moved. This is displayed by different mouse-cursors:

If the mouse is positioned over a vertical connection line, you can move the whole vertical line while holding down the left mouse key.

If the mouse is positioned over a horizontal connection line, you can move the whole horizontal line while holding down the left mouse key.

If the mouse is positioned over an oblique connection line, a new point is inserted into the connecting line when you left-click. You have to hold the left mouse key down, not releasing it until the mouse is positioned where the new point is to be placed.

If the mouse is positioned near an end point or a corner of a connecting line, you can move this point while holding down the left mouse key. You can only move a connected line endpoint to another suitable program element connection. In this case the endpoint of the connecting line will be linked to this connecting line. Otherwise, the point will not be moved.

A different approach to connecting lines

Connecting lines can also be created by moving program elements. If you move a program element so that its entry is one or two grid points below the exit of another, a connecting line between the elements is created. This also applies to an exit that is moved over an entry. After that, you can move the program element to its final position or draw further links for the remaining entries and exits.