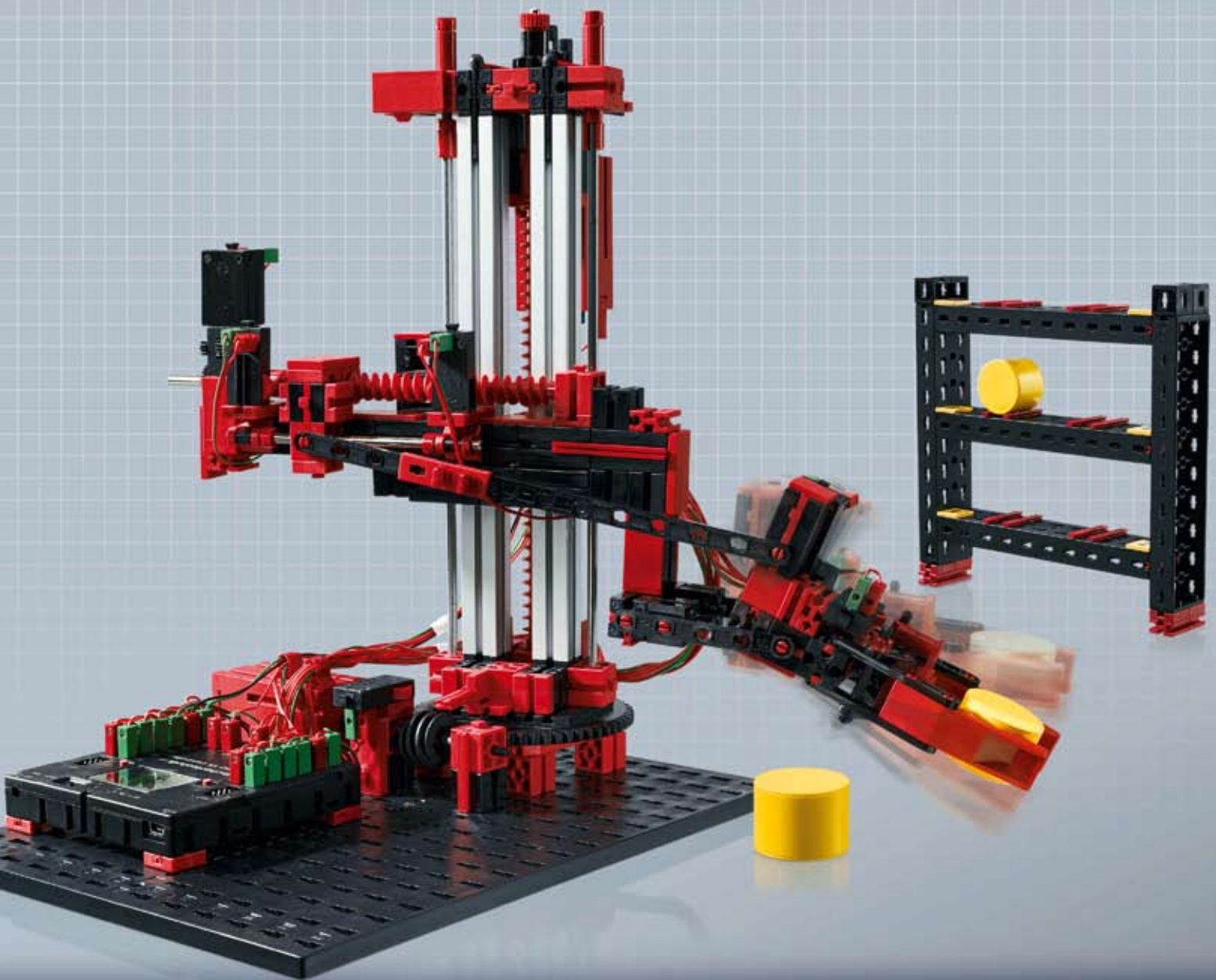




fischertechnik 

COMPUTING

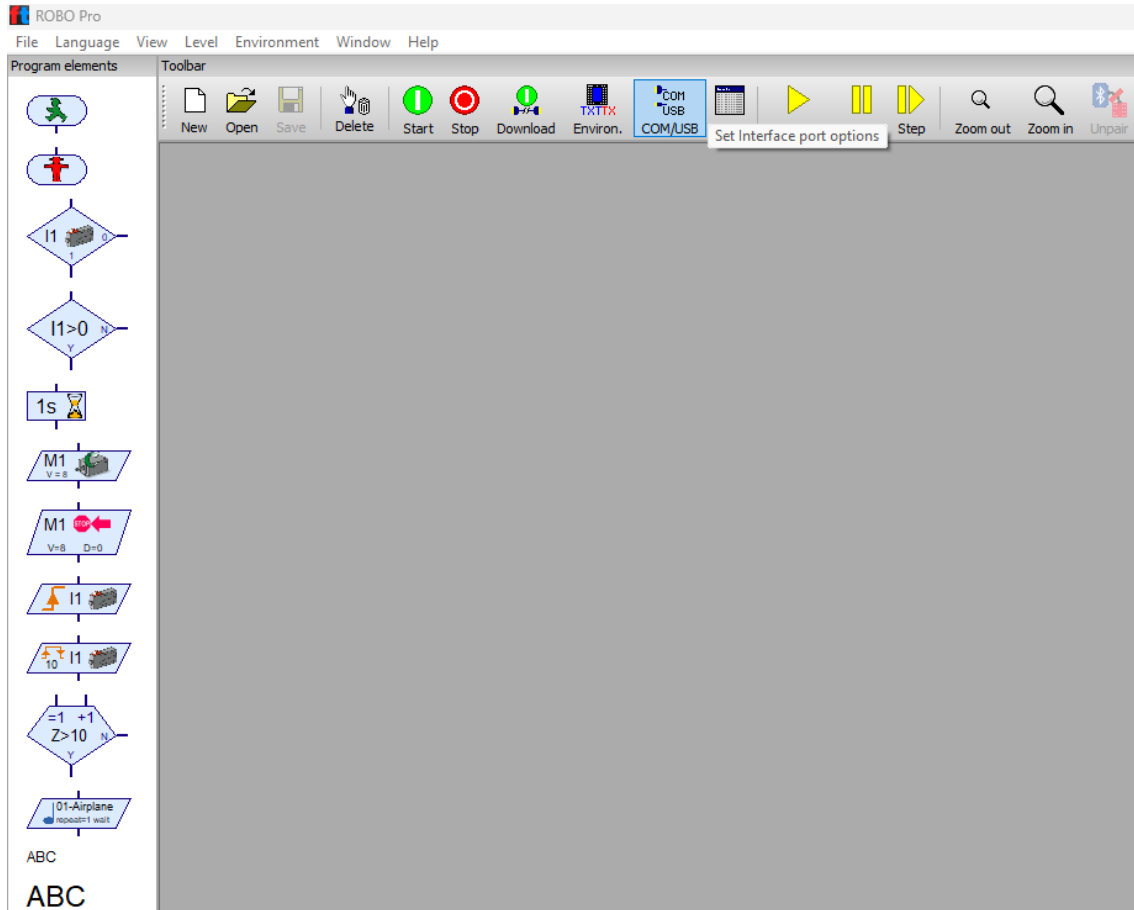


ROBO TX Automation Robots

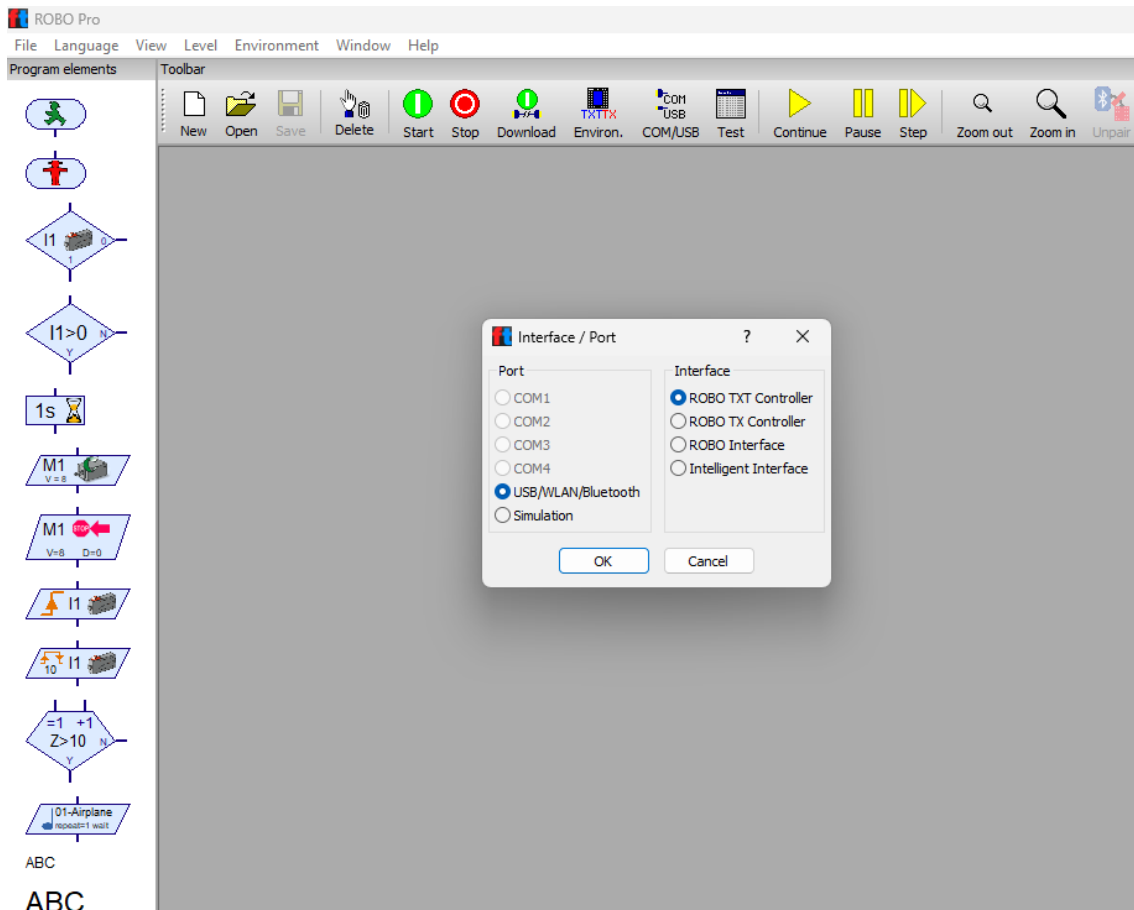
4 MODELS

Configuration Robo-Pro-1.1 - TXT Controller

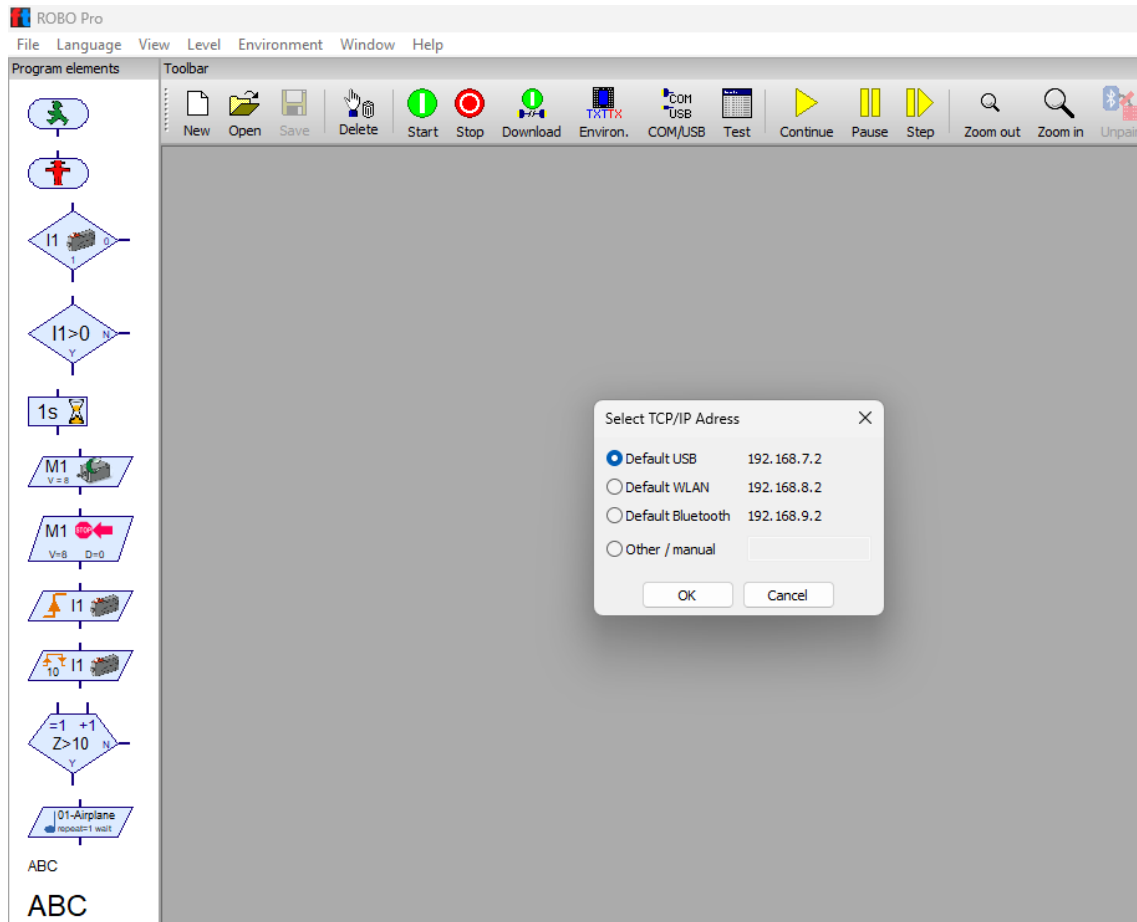
1: Select the communication port.



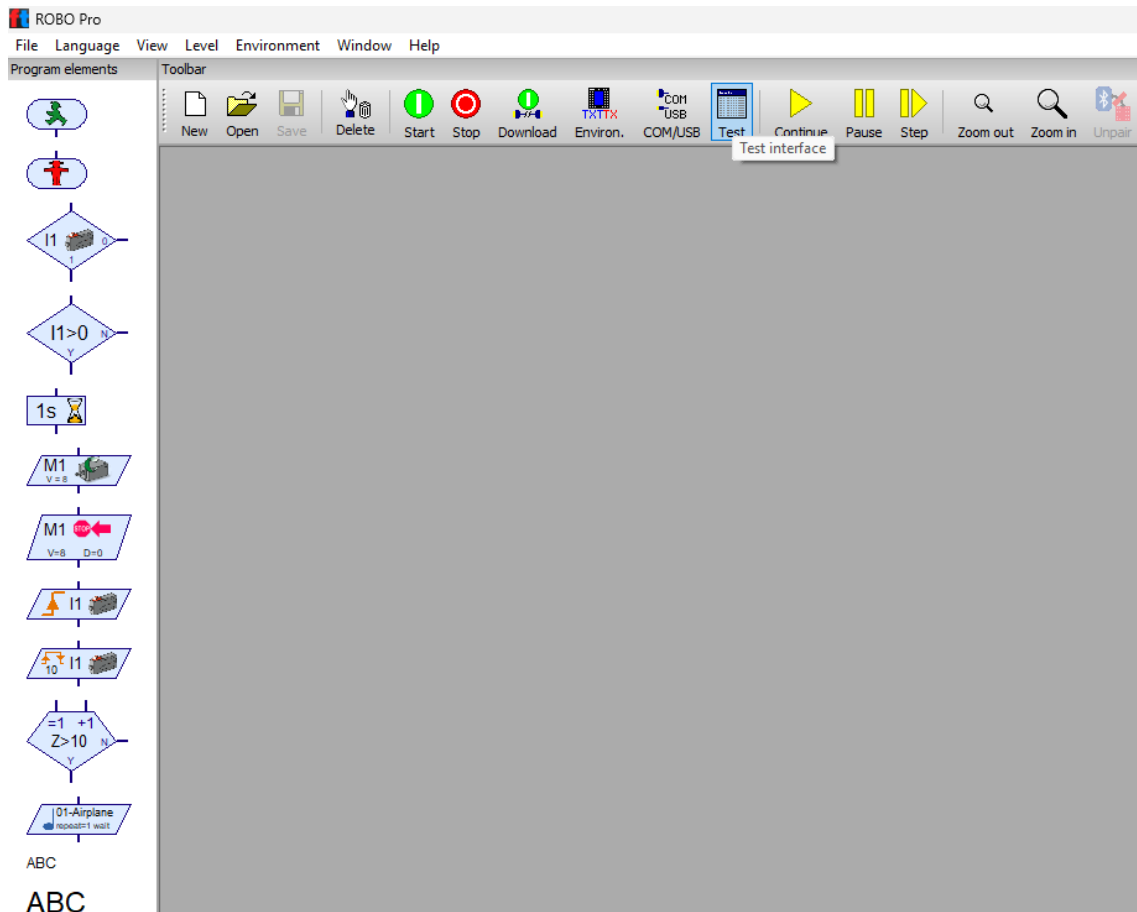
2. on Interface/port select USB with interface: Robo TXT Controller, and OK



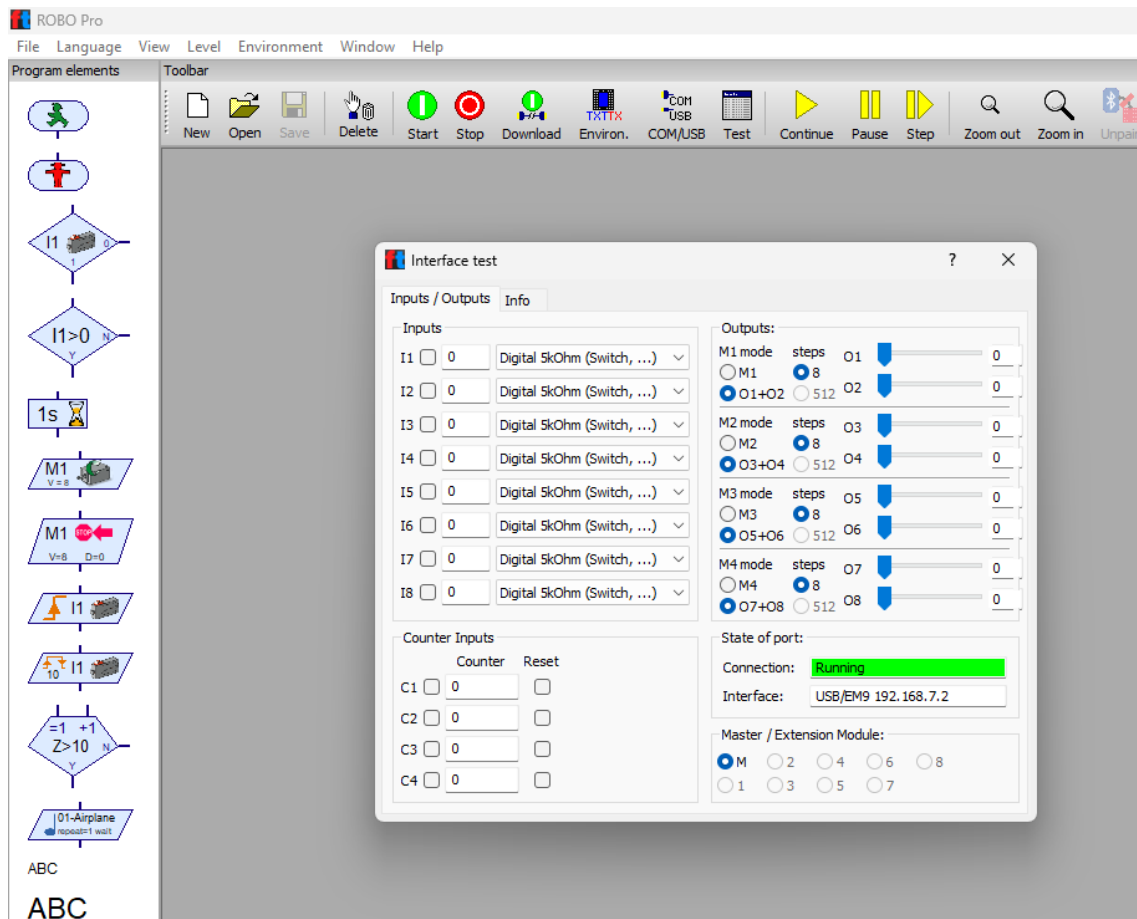
3. Select Default USB port, or in other: type the IP number for TCP/IP protocol.



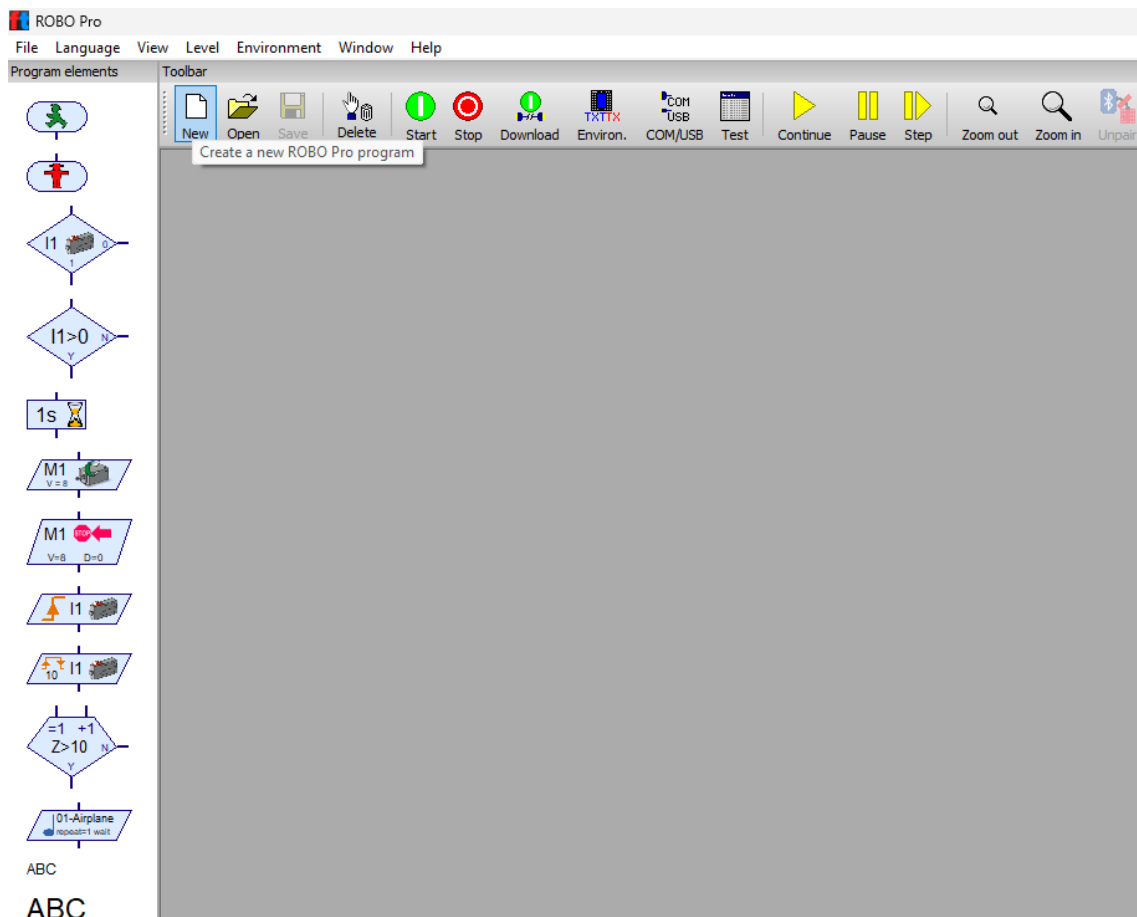
4. Check in Test interface to conform the communication with the controller.



5. if the controller is online, is possible to active outputs & look the input ports.



6. Now the system is ready to create a new program.



7. One example of program and sequence is present as follow.

The screenshot displays the ROBO Pro software interface. The main workspace shows a program sequence starting with a green robot icon, followed by a decision diamond labeled 'C1D'. The sequence then proceeds through a motor block 'O3', a delay block '1s', and another motor block 'O3'. An 'Interface test' window is open, showing a table of digital inputs and outputs. The 'State of port' is indicated as 'Running'.

Inputs	Outputs
11 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	M1 mode steps O1 <input type="checkbox"/> 0
12 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	<input checked="" type="radio"/> M1 <input checked="" type="radio"/> 8 O2 <input type="checkbox"/> 0
13 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	<input type="radio"/> O1+O2 <input type="checkbox"/> 512 O3 <input type="checkbox"/> 0
14 <input checked="" type="checkbox"/> 1 Digital SKOhm (Switch, ...)	M2 mode steps O4 <input type="checkbox"/> 0
15 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	<input type="radio"/> M2 <input checked="" type="radio"/> 8 O5+O6 <input type="checkbox"/> 512 O6 <input type="checkbox"/> 0
16 <input checked="" type="checkbox"/> 1 Digital SKOhm (Switch, ...)	M3 mode steps O7 <input type="checkbox"/> 0
17 <input checked="" type="checkbox"/> 1 Digital SKOhm (Switch, ...)	<input type="radio"/> M3 <input checked="" type="radio"/> 8 O7+O8 <input type="checkbox"/> 512 O8 <input type="checkbox"/> 0
18 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	M4 mode steps O7 <input type="checkbox"/> 0
	<input type="radio"/> M4 <input checked="" type="radio"/> 8 O7+O8 <input type="checkbox"/> 512 O8 <input type="checkbox"/> 0

Counter Inputs:

Counter	Reset
C1 <input type="checkbox"/> 3	<input type="checkbox"/>
C2 <input type="checkbox"/> 0	<input type="checkbox"/>
C3 <input type="checkbox"/> 0	<input type="checkbox"/>
C4 <input type="checkbox"/> 0	<input type="checkbox"/>

State of port: **Running**
Connection: USB/EM9 192.168.7.2
Interface: USB/EM9 192.168.7.2
Master / Extension Module:
 M 2 4 6 8
 1 3 5 7

8. Finally, the program can be executed online or downloaded to the controller.

The screenshot displays the ROBO Pro software interface. The main workspace shows a program sequence starting with a green robot icon, followed by a decision diamond labeled 'C1D'. The sequence then proceeds through a motor block 'O4', a delay block '1s', a motor block 'O4', a motor block 'O3', a delay block '1s', and a motor block 'O3'. An 'Interface test' window is open, showing a table of digital inputs and outputs. The 'State of port' is indicated as 'Running'. A 'Download' dialog box is also open, showing options for memory area and start program.

Inputs	Outputs
11 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	M1 mode steps O1 <input type="checkbox"/> 0
12 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	<input checked="" type="radio"/> M1 <input checked="" type="radio"/> 8 O2 <input type="checkbox"/> 0
13 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	<input type="radio"/> O1+O2 <input type="checkbox"/> 512 O3 <input type="checkbox"/> 0
14 <input checked="" type="checkbox"/> 1 Digital SKOhm (Switch, ...)	M2 mode steps O4 <input type="checkbox"/> 0
15 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	<input type="radio"/> M2 <input checked="" type="radio"/> 8 O5+O6 <input type="checkbox"/> 512 O6 <input type="checkbox"/> 0
16 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	M3 mode steps O7 <input type="checkbox"/> 0
17 <input checked="" type="checkbox"/> 1 Digital SKOhm (Switch, ...)	<input type="radio"/> M3 <input checked="" type="radio"/> 8 O7+O8 <input type="checkbox"/> 512 O8 <input type="checkbox"/> 0
18 <input type="checkbox"/> 0 Digital SKOhm (Switch, ...)	M4 mode steps O7 <input type="checkbox"/> 0
	<input type="radio"/> M4 <input checked="" type="radio"/> 8 O7+O8 <input type="checkbox"/> 512 O8 <input type="checkbox"/> 0

Counter Inputs:

Counter	Reset
C1 <input type="checkbox"/> 0	<input type="checkbox"/>
C2 <input type="checkbox"/> 0	<input type="checkbox"/>
C3 <input type="checkbox"/> 0	<input type="checkbox"/>
C4 <input type="checkbox"/> 0	<input type="checkbox"/>

State of port: **Running**
Connection: USB/EM9 192.168.7.2
Interface: USB/EM9 192.168.7.2
Master / Extension Module:
 M 2 4 6 8
 1 3 5 7

Download dialog box options:

- Memory area:
 - RAM (contents lost when powered down)
 - Flash (retained through power down)
- Start program:
 - Start program after download
 - Start program using button on Interface
- Options:
 - Download all Bluetooth programs (0)

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



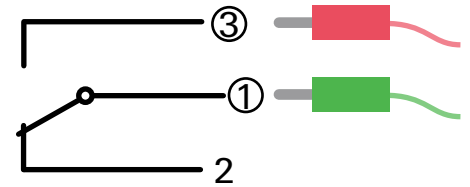
The pushbutton is also called a touch sensor. When the red button is pressed a switch is switched mechanically allowing current to flow from contact 1 (middle contact) to contact 3. Simultaneously the contact between connections 1 and 2 is interrupted. So you can use the pushbutton in two different ways:

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.

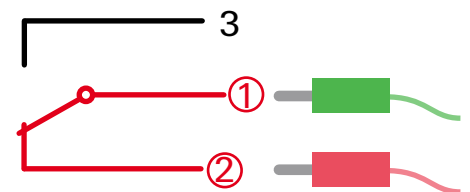


As a "break contact" (normally closed switch):

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.

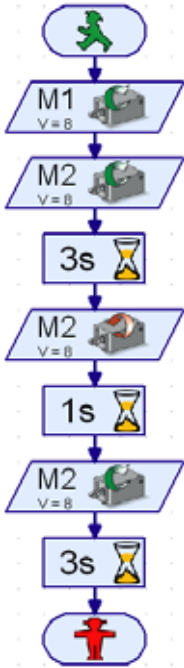


A special treat is the integrated Bluetooth interface. It allows you to connect your PC to the ROBO TX Controller. Or to connect a number of controllers with one another. You can define how the controller interacts with the individual components and what they are to do in detail in the program you write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

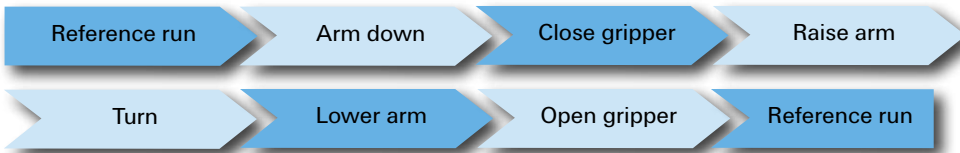
Programing Tips:

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.



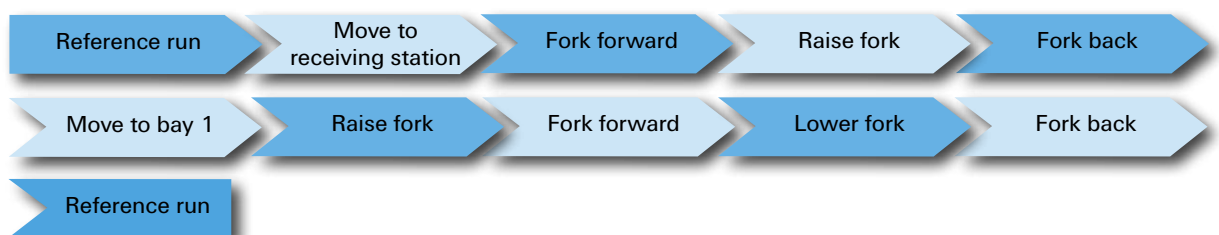
Task 1:

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



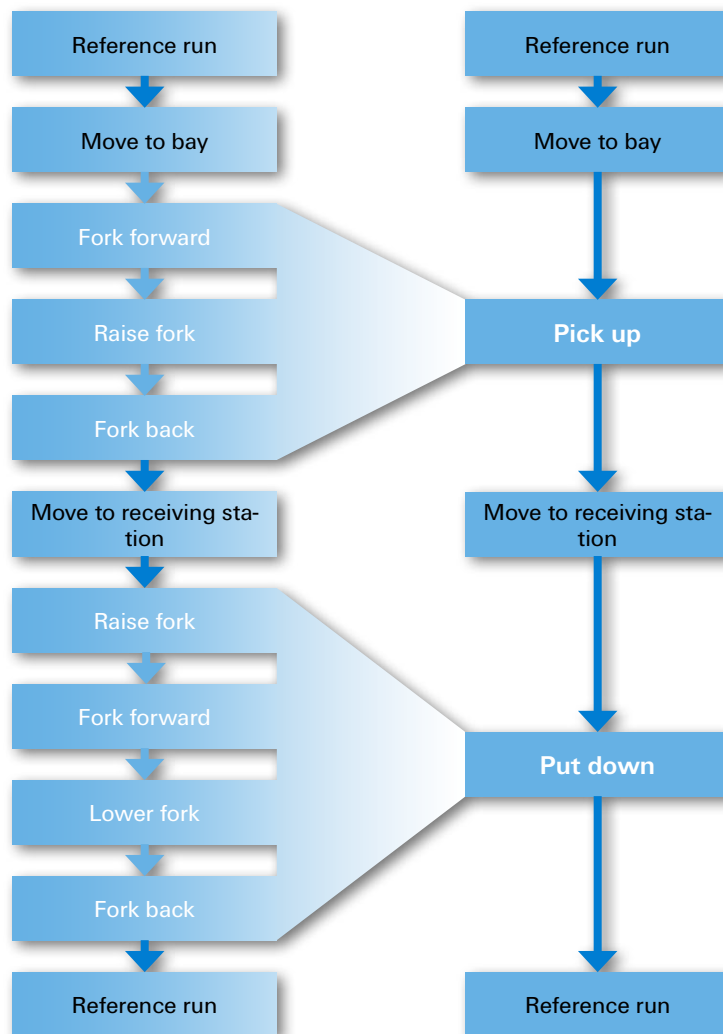
Task 2:
 Expand your program with subroutines to move to the remaining storage bays. Rewrite the program so that a number of barrels can be stored in various storage bays one after the other. Then have the stored barrels retrieved from their storage bays one after another and brought to the receiving 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

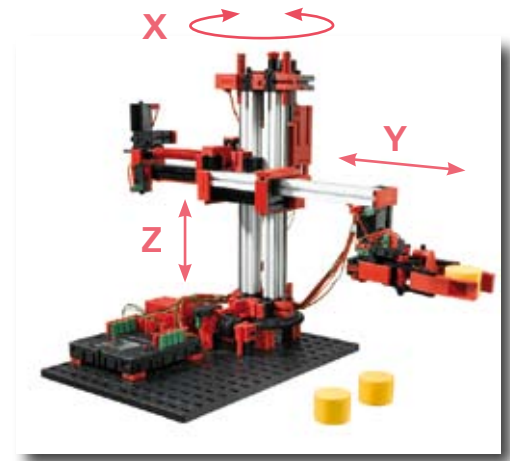
Task 3:
Expand the program in Task 2. The robot should now tear down the tower on the deposit platform and place the barrels back on the holders on the left and right next to the model.



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.

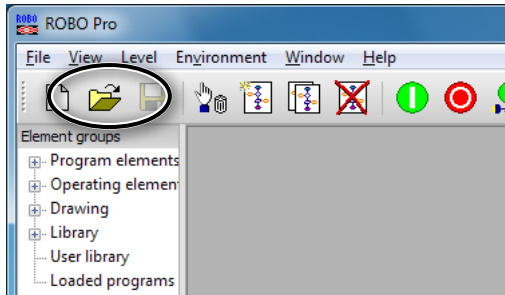
Teach-in procedure

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

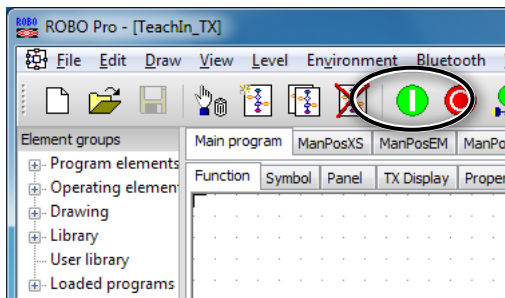
Load, start, select control panel



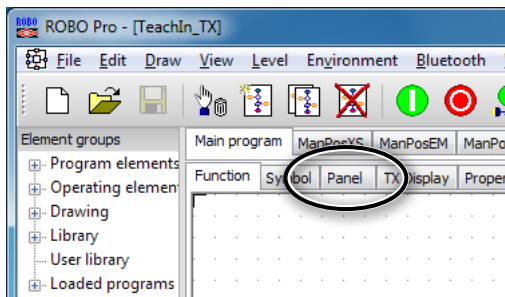
You can load the teach-in program by clicking on the symbol.

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.

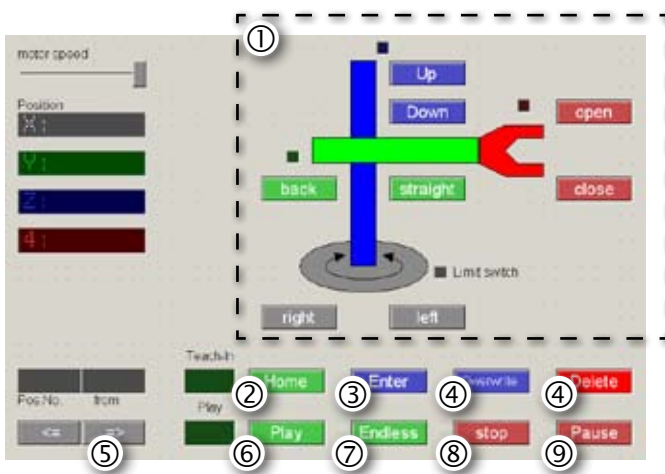


Start the teach-in program.



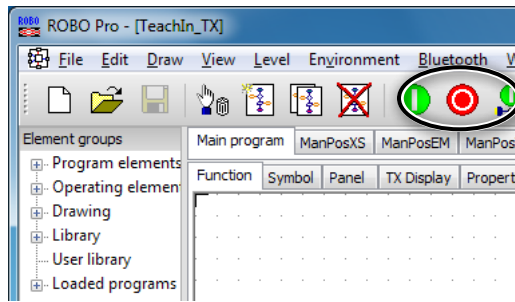
Select the panel to program the robot.

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.

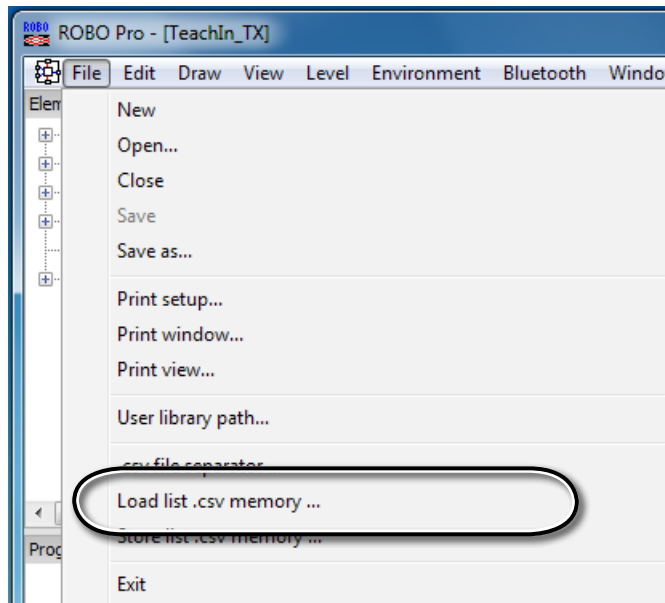
Stop



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:

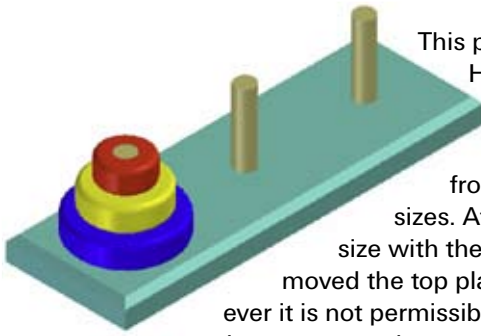
Task 1:
 The robot should pick up a barrel with its gripper and put it back down at a different position.



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



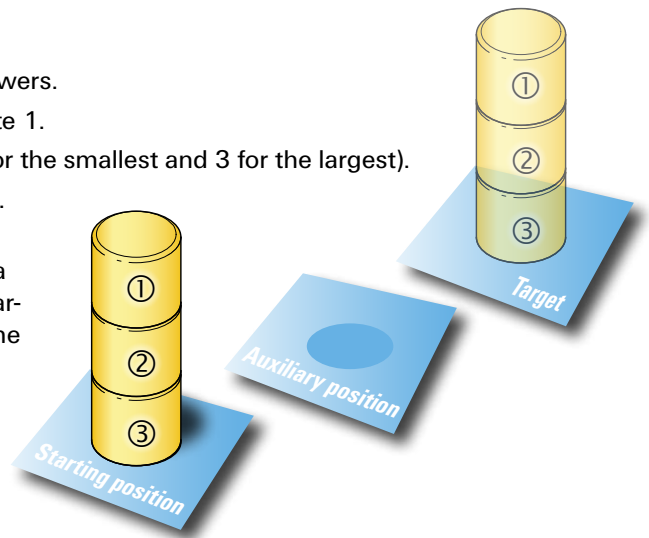
This puzzle was invented in 1883 by the French mathematician Édouard Lucas. He made up the following story:

"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.

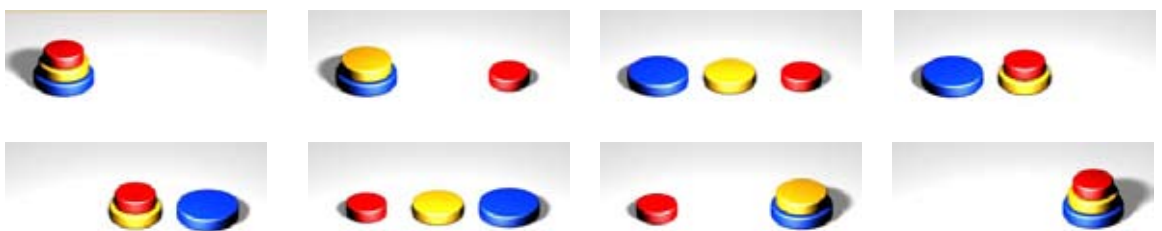
To make this puzzle a little simpler, use 3 barrels instead of 64 plates. Since the barrels are all the same size, you can label them with stickers with the numbers 1–3.

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 moved 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

ÍNDICE

1	Introducción – Controlar los modelos de fischertechnik mediante ROBO Pro	439
1.1	Instalación de ROBO Pro.....	439
1.2	Instalación del controlador USB para la ROBO-Interface.....	440
1.3	Primeros pasos.....	443
2	Antes de la programación, un breve test del hardware	447
2.1	Conexión de la interfase al ordenador.....	447
2.2	Para que la conexión sea correcta: las configuraciones de la interfase.....	447
2.3	Si la conexión es incorrecta: ¿no hay conexión a la interfase!.....	449
2.4	Todo funciona: prueba de interfase ..	449
3	Nivel 1: Tu primer programa de control	451
3.1	Crear un nuevo programa.....	451
3.2	Los elementos del programa de control.....	452
3.3	Insertar, desplazar y modificar elementos de programa.....	452
3.4	Interconectar los elementos de programa.....	455
3.5	Probar el primer programa de control.....	456
3.6	Más elementos de programa.....	458
3.6.1	Tiempo de espera.....	458
3.6.2	Esperar a entrada.....	459
3.6.3	Contador de impulsos.....	459
3.6.4	Bucle contador.....	460
3.7	Modo online o modo de descarga. ¿Cuál es la diferencia?.....	460
3.8	Consejos y trucos.....	463
4	Nivel 2: Trabajar con subprogramas	464
4.1	Tu primer subprograma.....	465
4.2	La biblioteca del subprograma.....	469
4.2.1	Utilizar la biblioteca.....	469
4.2.2	Utilizar la propia biblioteca.....	470
4.3	Editar símbolos de subprogramas.....	471
4.4	Tango.....	471
4.5	Tango 2: comunicación a través de ROBO RF Data Link.....	478
4.5.1	Configuración de la frecuencia y el número de llamada por radio.....	482
4.5.2	Requisitos para la comunicación por radio ation.....	483
4.5.3	Selección de interfaces con el botón de puertos (COM/USB).....	483
5	Nivel 3: Variables, paneles de control & cía	485
5.1	Variables e instrucciones.....	485
5.2	Variables y varios procesos.....	487
5.3	Paneles de control.....	487
5.4	Temporizador.....	491
5.5	Entradas de comando para subprogramas.....	492
5.6	Listados (Arrays).....	495
5.7	Operadores.....	497
6	Nivel 4: comandos definidos por el usuario	501
6.1	Procesamiento de comandos en un proceso.....	501
6.2	El filtro de comandos.....	502
6.3	Enviar cualquier comando a subprogramas.....	504
7	Módulos de extensión y gestión de más de una interfase	505
7.1	Módulos de extensión.....	505
7.2	Múltiples interfaces.....	505
7.3	Asignaciones de interfase en subprogramas.....	507
7.4	Consejos & Trucos.....	508
7.5	Cambiar el número de serie de la interfase o la versión de firmware.....	508
8	Los elementos de programa	510
8.1	Elementos básicos (Nivel 1).....	510
8.1.1	Inicio.....	510
8.1.2	Fin.....	510
8.1.3	Bifurcación digital.....	511
8.1.4	Bifurcación analógica.....	511
8.1.5	Tiempo de espera.....	512
8.1.6	Salida de motor.....	512
8.1.7	Salida de lámpara (Nivel 2).....	513
8.1.8	Esperar a entrada.....	515
8.1.9	Contador de impulsos.....	516
8.1.10	Bucle contador.....	516
8.2	Enviar y recibir (Niveles 2-4).....	517
8.2.1	Emisor (Nivel 2).....	517
8.2.2	Receptor / Bifurcación al recibir el comando (Nivel 2).....	518
8.2.3	Receptor (Nivel 3).....	519

8.2.4	Esperar comando (Nivel 4)	520	9.1	Indicaciones.....	541
8.2.5	Filtro de comandos (Nivel 4)	521	9.1.1	Aparato de medición	541
8.3	Subprograma I/O (Nivel 2-3)	521	9.1.2	Mensaje de texto	542
8.3.1	Entrada de subprograma (Nivel 2).....	521	9.1.3	Lámpara indicadora.....	543
8.3.2	Salida de subprograma (Nivel 2).....	521	9.2	Elementos de mando.....	544
8.3.3	Entrada de comando de subprograma (Nivel 3).....	522	9.2.1	Botón	544
8.3.4	Salida de comando de subprograma (Nivel 3).....	522	9.2.2	Regulador.....	545
8.4	Variable, Listado, ... (Nivel 3)	522	10	Funciones de dibujo.....	546
8.4.1	Variable (global).....	522			
8.4.2	Variable local.....	523			
8.4.3	Constante.....	524			
8.4.4	Variable de temporizador	524			
8.4.5	Listado	525			
8.5	Instrucciones (Nivel 3).....	527			
8.5.1	= (Asignar)	528			
8.5.2	+ (Más)	528			
8.5.3	- (Menos)	529			
8.5.4	Derecha	529			
8.5.5	Izquierda	529			
8.5.6	Detener	529			
8.5.7	ON	529			
8.5.8	OFF	529			
8.5.9	Texto	530			
8.5.10	Agregar valor.....	530			
8.5.11	Eliminar valor(es).....	530			
8.5.12	Intercambiar valores.....	530			
8.6	Compara, Esperar a, ... (Nivel 3).....	530			
8.6.1	Bifurcación (con entrada de datos)	531			
8.6.2	Comparación con valor fijo	531			
8.6.3	Comparación	532			
8.6.4	Tiempo de espera	532			
8.6.5	Esperar a	532			
8.6.6	Contador de impulsos	533			
8.7	Entradas / salidas de la interfase, ...	533			
8.7.1	Entrada digital	533			
8.7.2	Entrada analógica	534			
8.7.3	Entrada IR	535			
8.7.4	Salida de motor	536			
8.7.5	Salida de lámpara	537			
8.7.6	Entrada del panel de control	538			
8.7.7	Salida del panel de control.....	539			
8.8	Operadores.....	539			
8.8.1	Operadores aritméticos.....	539			
8.8.2	Operadores lógicos.....	540			
9	Los elementos de control y paneles de control	541			

1 Introducción – Controlar los modelos de fischertechnik mediante ROBO Pro

Seguramente te has preguntado alguna vez cómo funciona un robot cuando, movido como por arte de magia, realiza alguna tarea determinada. No solamente con los robots propiamente dichos: en muchos otros campos se utiliza la técnica de control y automatización; también en fischertechnik. En el capítulo siguiente diseñaremos juntos un programa de control para una puerta automática de garaje y con ello aprenderemos cómo podemos solucionar y probar tales tareas de control con la ayuda del Software ROBO Pro para Windows. Al hacerlo comprobaremos que el manejo del ROBO Pro es muy sencillo. Como aprenderemos más adelante, en la superficie gráfica pueden crearse los programas de control (más concretamente los diagramas de operaciones primero y los diagramas de flujo de datos después) casi exclusivamente mediante el uso del ratón.

Para que puedas activar tus modelos de fischertechnik a través del ordenador, necesitarás, además del software de control ROBO Pro, una interfase que actúa como nexo de unión entre ordenador y modelo. Transforma las instrucciones del software de tal manera que puedan, por ejemplo, activarse los motores y procesarse las señales de los sensores. fischertechnik ofrece la ROBO Interface Art. No. 93293 y la interfase inteligente Art. No. 30402, más antigua. Puedes utilizar cualquiera de ellas con la ROBO Pro, aunque la ROBO Pro solamente soporta el modo "Online" de la interfase inteligente. ROBO Pro ya no soporta la antigua interfase paralela Art. No. 30520.

Añadamos algunas palabras más sobre la estructura de este manual. El manual se divide en dos partes; la primera comprende desde el capítulo 1 hasta el capítulo 4, y describe los procedimientos básicos de programación con la ROBO Pro. Contiene una gran cantidad de información y conocimientos generales sobre programación en general y sobre la utilización del software ROBO Pro en particular.

La segunda parte comprende desde el capítulo 5 hasta el 7 e introduce las funciones de los programas avanzados.

Los capítulos 8 y siguientes son más bien un texto de consulta. Si al terminar la primera parte te hallas ya familiarizado con el manejo de la ROBO Pro y buscas información adicional sobre algún tema concreto, encontrarás en esos capítulos explicaciones detalladas sobre los elementos individuales de los programas.

Y sin más dilación, ¡manos a la obra! Seguramente ya tienes mucha curiosidad por descubrir qué posibilidades te ofrece el software ROBO Pro para programar tus modelos de fischertechnik. ¡Que te diviertas!

1.1 Instalación de ROBO Pro

Los requisitos mínimos para la instalación de ROBO Pro son los siguientes:

- Un PC compatible IBM con procesador Pentium a 600 MHz como mínimo, 32 MB RAM y unos 20 MB de memoria libre en el disco duro
- Un monitor y una tarjeta gráfica con resolución mínima de 1024x768 píxeles. En monitores con tubo de imagen la frecuencia de refresco de imagen debería ser como mínimo de 85 Hz para obtener una imagen sin parpadeo. Las pantallas planas TFT ofrecen siempre una imagen sin parpadeo, de modo que la frecuencia de refresco no tiene importancia en este tipo de pantallas.

- Microsoft Windows 95, 98, ME, NT4.0, 2000 o XP
- Un puerto USB libre o un puerto RS232 libre de COM1 a COM4 para la conexión de la ROBO Interface Art. No. 3293, o un puerto RS232 de COM1 a COM4 para la conexión de la interfase inteligente Art. No. 30402 más antigua

El primer paso es, claro está, iniciar el ordenador y esperar hasta que el sistema operativo (Windows) se haya cargado completamente. La ROBO Interface no se debe conectar al ordenador hasta que la instalación haya concluido con éxito. Se procede a introducir el CD de instalación en la unidad del CD-ROM. El programa de instalación del CD se inicia automáticamente.

- En la primera ventana de bienvenida del programa de instalación, haz clic en **Continuar**.

La segunda ventana, **Indicaciones importantes**, contiene información actualizada importante sobre la instalación del programa o sobre el programa en sí. Aquí también haz clic en **Continuar**.

- En la tercera ventana, **Condiciones de la licencia**, se refleja el contrato de licencia para ROBO Pro. Debes aceptar este contrato con **Sí** antes de poder acceder a la siguiente ventana haciendo clic en **Continuar**.
- En la siguiente ventana, **Datos del usuario**, debes introducir tu nombre.
- En la ventana **Tipo de instalación** puedes elegir entre la **Instalación Exprés e Instalación definida por el usuario**. En la instalación definida por el usuario se te permite excluir algunos componentes de la instalación. Si estás instalando una nueva versión de ROBO Pro sobre una versión anterior y has cambiado algunos de los programas de ejemplo de la versión antigua, puedes excluir los ejemplos originales en la instalación definida por el usuario. En caso contrario **se sobrescriben sin aviso** los programas de ejemplo cambiados durante la instalación. Si eliges la instalación definida por el usuario y haces clic en **Continuar**, aparece una nueva ventana en la que podrás elegir los componentes.
- En la ventana **Directorio de instalación** puedes elegir la carpeta o la ruta de carpeta donde quieras que se instale el programa ROBO Pro. La ruta por defecto es C:\Archivos de programa\ROBO Pro, pero puedes introducir cualquier otro directorio.
- Si en la última ventana haces clic sobre **Finalizar**, se ejecuta la instalación. Al terminar la instalación, que normalmente tarda sólo pocos segundos, el programa indica si la instalación se ha completado con éxito. En caso de que haya problemas, aparece un aviso de error que debería ayudarte a solucionar el problema.

1.2 Instalación del controlador USB para la ROBO-Interface

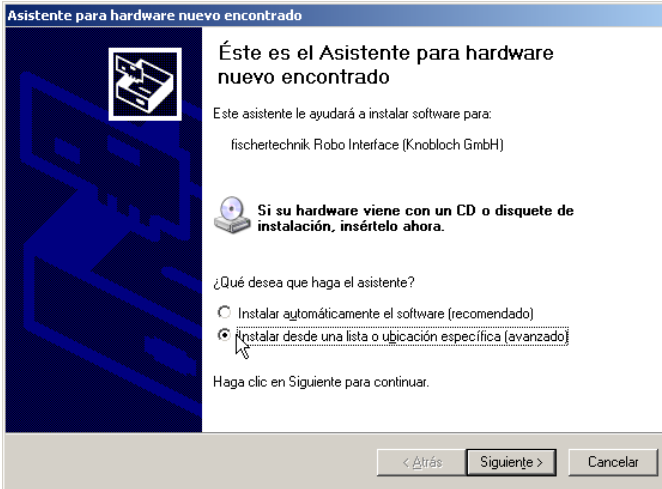
Este paso es solamente necesario si la nueva ROBO-Interface se va a conectar al puerto USB. La ROBO-Interface puede también conectarse a un puerto serie COM1-COM4. Para la interfase inteligente anterior este paso no es necesario, dado que se conecta exclusivamente al puerto serie. Las versiones de Windows más antiguas (Windows 95 y Windows NT 4.0) no soportan el puerto USB, de modo que si se utiliza Windows 95 o NT 4.0 la ROBO-Interface únicamente puede conectarse a través de un puerto serie, en cuyo caso tampoco se necesita instalar el controlador.

Indicación importante para la instalación en Windows 2000 y Windows XP:

Solamente puede instalar el controlador USB un usuario que cuente con derechos de administrador. Si el programa de instalación avisa de que no es posible instalar el controlador

USB, debes pedir a tu administrador que te instale el controlador o instalar ROBO Pro sin él, pero en este último caso sólo podrás conectar la interfase a través del puerto serie (más lento).

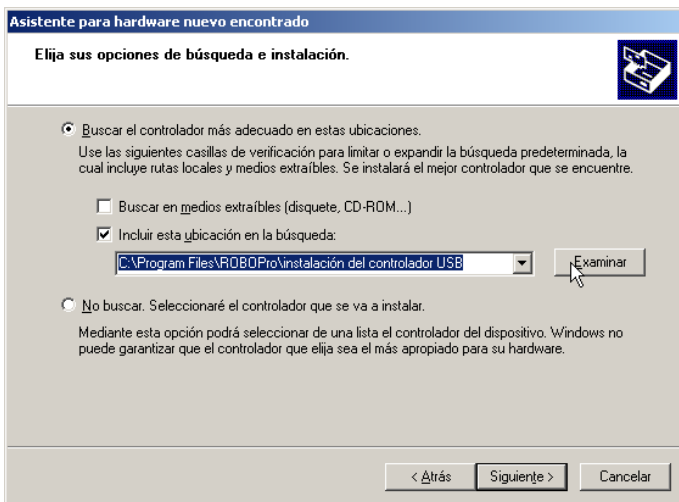
Para instalar el controlador USB, conecta primero la ROBO-Interface al ordenador mediante un cable USB y enchúfalo a la alimentación. Windows reconoce automáticamente que la interfase está conectada y muestra la siguiente ventana:



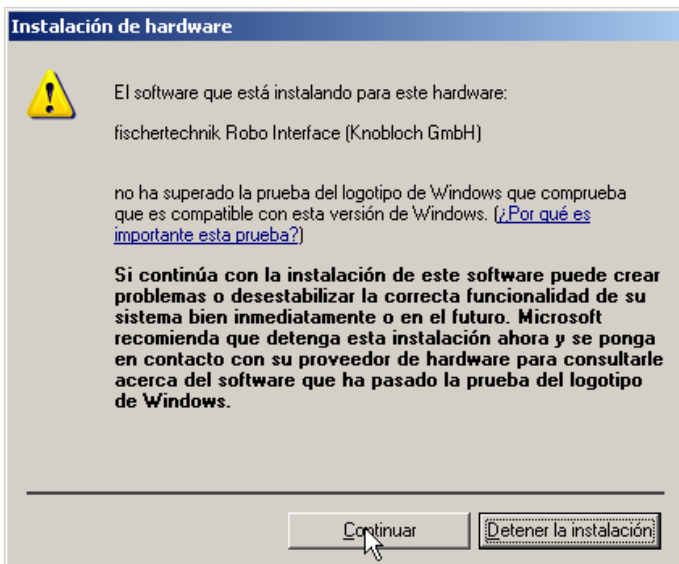
Atención: el aspecto de la ventana puede variar según se use uno u otro sistema operativo.

Aquí debes elegir **Instalar software desde una lista o desde una fuente determinada** y hacer clic en **Continuar**.

En la siguiente ventana debes desactivar **Examinar unidades extraíbles** y activar **Examinar también las siguientes fuentes**. Después haz clic sobre **Examinar** y elige el subdirectorio **Instalación del controlador USB** en el directorio en el cual está instalada la ROBO Pro (el directorio por defecto es C:\Archivos de programa\ROBOPro):

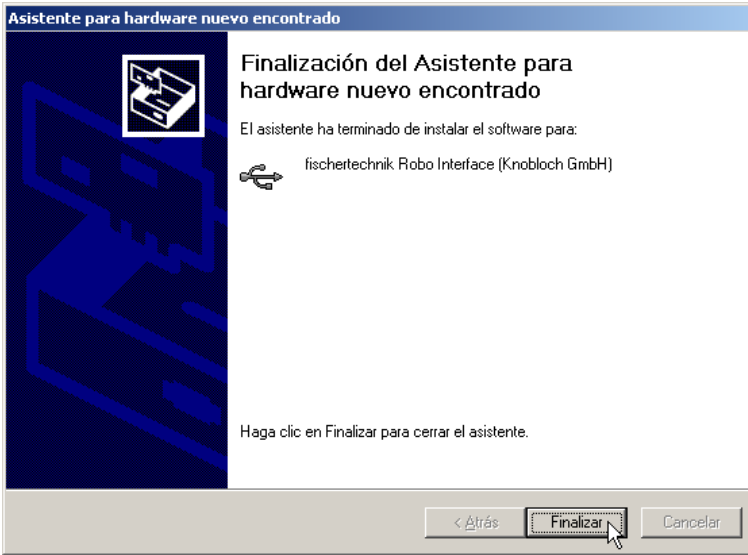


Al hacer clic en **Continuar**, es posible que Windows XP muestre el siguiente aviso:



El controlador USB tendrá que ser comprobado por Microsoft. En cuanto haya terminado la comprobación, Microsoft registra el controlador, así que este aviso ya no aparecerá más. Para instalar el controlador, haz clic en **Continuar la instalación**.

Finalmente aparece la siguiente pantalla:



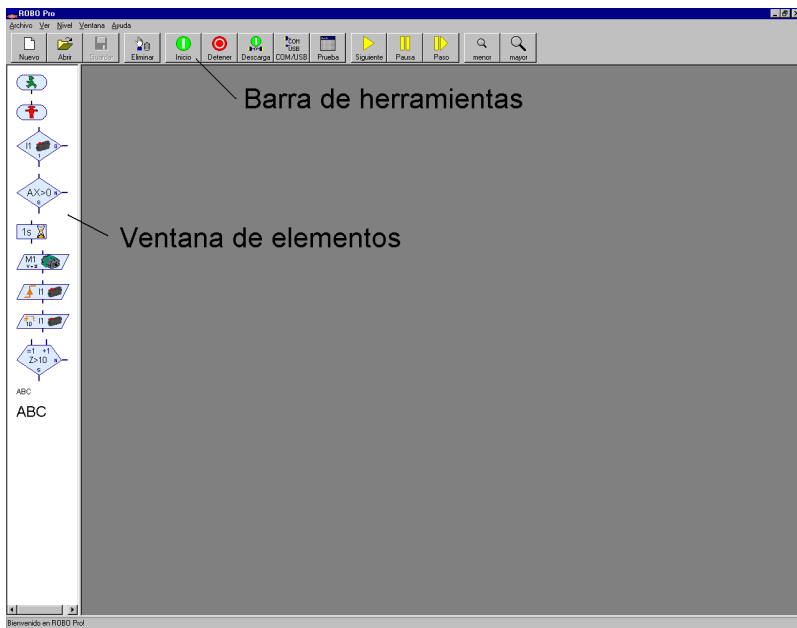
Haz clic sobre **Finalizar** para concluir la instalación del controlador USB.

1.3 Primeros pasos

¿Tienes ya ganas de empezar? Pues venga, inicia el programa ROBO Pro. Para hacerlo, haz clic en el botón de Inicio de la barra de tareas y a continuación elige **Programas** o **Todos los programas** y **ROBO Pro**. En esta carpeta del menú del Inicio encontrarás las siguientes entradas::



La entrada Desinstalar permite desinstalar ROBO Pro. La entrada Ayuda abre el archivo de ayuda para ROBO Pro. La entrada ROBO Pro abre el programa de ROBO Pro. Debes elegir la entrada **ROBO Pro** para iniciar el software.

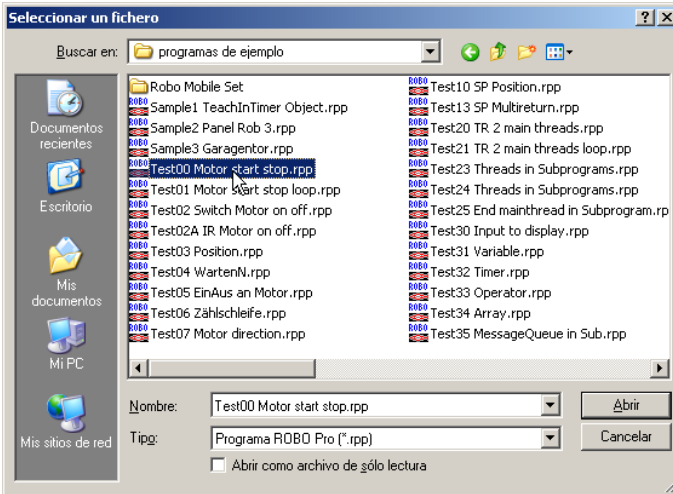


La ventana tiene en la parte superior una barra de menús y una barra de herramientas con distintos botones de instrucción y en la parte izquierda una ventana con elementos del programa. Si en la columna lateral izquierda ves dos ventanas superpuestas, es que la ROBO Pro no está ajustada en **Nivel 1**. Para adaptar la funcionalidad de ROBO Pro a los conocimientos acumulativos del usuario, es posible ajustar ROBO Pro desde al Nivel 1 para principiantes hasta el Nivel 5 para expertos. En este momento debes comprobar en el menú **Nivel** si el **Nivel 1: Principiantes** está marcado. Si no es así, debes marcarlo.

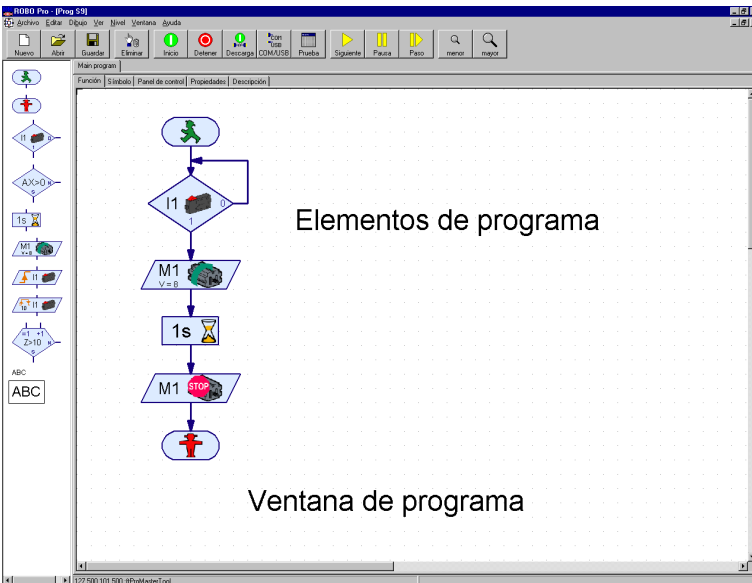
Ahora tienes la posibilidad de crear un nuevo archivo de programa o de abrir un archivo de programa ya existente. Crearemos un nuevo archivo de programa en el capítulo 3, cuando generemos nuestro primer programa de control. Para familiarizarnos con la superficie de control, abriremos ahora uno de los dos programas de ejemplo ya existentes. Para ello debes hacer clic en el menú **Archivo** sobre la entrada **Abrir** o utilizar el botón **Abrir** de la barra de herramientas. Los archivos de ejemplo se encuentran en el directorio **C:\Archivos de programa\ROBO Pro\Programas de ejemplo**.



Abrir



Abre el archivo **Test00 Motor start Stop.rpp**:



Aquí puedes ver qué aspecto tiene un programa sencillo de ROBO Pro. Con los elementos de programa de la ventana de elementos se crean durante la programación los diagramas de operaciones de los programas de control en la ventana de programación, para después comprobarlos y probarlos mediante una interfase fischertechnik que habremos conectado al ordenador. ¡Pero tranquilo! En los próximos capítulos aprenderemos paso a paso a programar. Cuando te hayas hecho ya una idea de cómo es la superficie de control, cierra de nuevo el archivo de programa mediante la instrucción **Salir** del menú **Archivo**. A la pregunta de si quieres guardar el archivo, puedes contestar con **No**.

2 Antes de la programación, un breve test del hardware

Para que podamos probar los programas de control que más adelante crearemos, la interfase debe estar (naturalmente) conectada al ordenador. Pero en función de la interfase utilizada (ROBO-Interface Art. No. 93293 o interfase inteligente Art. No. 30402, más antigua) deberá ajustarse también el software y comprobarse la conexión. Todo eso lo haremos en esta sección.

2.1 Conexión de la interfase al ordenador

Esto no debería representar ningún problema. El cable de conexión suministrado con la interfase se conecta a ella y a un puerto del ordenador:

- En la ROBO-Interface (Art. No. 93293) puede conectarse a un puerto USB o a un puerto serie (de COM1 a COM4).
- En la interfase inteligente (Art. No. 30402) debe conectarse a un puerto serie (de COM1 a COM4).

Las conexiones de estos puertos se encuentran normalmente en la parte trasera del ordenador. La posición exacta de los puertos está descrita detalladamente en el manual del PC, y ahí puedes consultarla. También es frecuente que el puerto USB se encuentre en la parte delantera del ordenador. No olvides alimentar la interfase (aparato alimentador o acumulador). Las conexiones individuales de la interfase están descritas detalladamente en las instrucciones de uso de cada una.

2.2 Para que la conexión sea correcta: las configuraciones de la interfase

Para que la conexión del ordenador y la interfase funcione correctamente, debe ajustarse la interfase utilizada en ROBO Pro. Para ello inicia ROBO Pro seleccionando la opción **ROBO Pro** desde el menú de Inicio, abriendo primero **Programas** o **Todos los programas** y después **ROBO Pro**. Después haz clic en el botón **COM/USB** de la barra de herramientas. Aparecerá la siguiente ventana:

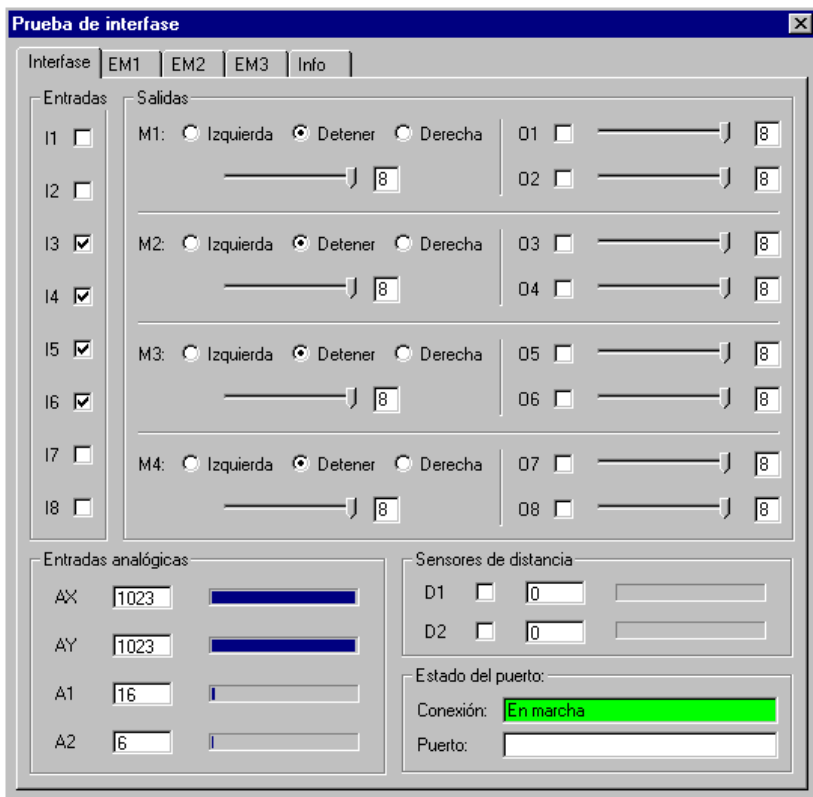


Aquí puedes elegir tanto el puerto como el tipo de interfase. La interfase inteligente soporta, como se dijo, solamente puertos en serie COM1–COM4.



Prueba

Una vez efectuados las configuraciones necesarias, debes cerrar la ventana con OK. A continuación abre la ventana de prueba de la interfase con el botón **Prueba** de la barra de herramientas:



Se abre una pantalla que muestra las entradas y las salidas presentes en la interfase. La barra de la esquina inferior derecha indica el estado de conexión desde el ordenador a la interfase:

- **Conexión a la interfase OK** confirma una conexión correcta a la interfase.
- **No hay conexión a la interfase** indica que la conexión no se ha ajustado correctamente y que el ordenador no pudo establecer una conexión con la interfase, en cuyo caso la barra aparece en color rojo.

Para poder cambiar las configuraciones de la interfase o de la conexión, debes cerrar la ventana de prueba (con la X de la esquina superior derecha) y, como se describió anteriormente, elegir otro puerto u otro tipo de interfase mediante el botón COM/USB en la barra de herramientas.

Si con eso consigues ajustar la conexión del ordenador con la interfase y en la ventana de **Prueba** aparece la barra verde, puedes saltarte tranquilamente la siguiente sección.

Si no, quizás te ayuden los consejos del siguiente párrafo..

2.3 Si la conexión es incorrecta: ¡no hay conexión a la interfase!

Si en la ROBO o la interfase inteligente apareciera el aviso **No hay conexión a la interfase** a pesar de haber ajustado correctamente el puerto COM en serie (según lo explicado antes), deberías comprobar los siguientes puntos. Quizás te haga falta preguntar a alguien que entienda de ordenadores:

- **Alimentación de corriente:**

¿Le llega una correcta alimentación a la interfase? Si para la alimentación de corriente utilizas pilas o acumuladores, existe la posibilidad de que los acumuladores estén vacíos y no suministren tensión suficiente. Si la tensión de la batería baja por debajo de 6 V, el procesador de la interfase ROBO deja de funcionar. En este caso el diodo luminoso estará encendido permanentemente o no se encenderá en absoluto. Si la tensión está bien, parpadean algunos de los diodos luminosos en verde. En la interfase inteligente más antigua no se puede descubrir tan fácilmente mediante los diodos luminosos si la tensión es suficiente para el procesador. En caso de que la tensión sea demasiado baja, debes recargar los acumuladores o utilizar pilas nuevas, o por el contrario comprobar la interfase, si posible, con un aparato alimentador.

- **¿Funciona el puerto?**

Puedes averiguarlo probando en el puerto otro aparato en serie, como por ejemplo un módem externo.

- ¿Hay algún conflicto con otro controlador de dispositivos del mismo puerto (por ejemplo un módem)? Posiblemente deberá desactivarse este controlador (consulta el manual de Windows o el manual del dispositivo).
- Solamente para Windows NT/2000/XP y la interfase inteligente más antigua: si durante el arranque del PC ya hay una interfase inteligente más antigua conectada al ordenador y la alimentación de corriente, Windows NT cambiará al modo de descarga. Para reestablecer la conexión al PC, debes simplemente interrumpir un momento la alimentación de la interfase. En la nueva ROBO Interface esto no sucede.
- Si sigues sin poder establecer una conexión a la interfase, es probable que la interfase o el cable de conexión estén defectuosos. En este caso debes dirigirte al servicio técnico de fischertechnik (encontrarás la dirección en el menú: ? / **Acerca de**).

2.4 Todo funciona: prueba de interfase

Una vez que la conexión esté ajustada correctamente, procedemos a comprobar la interfase misma y los modelos conectados a ella. Como ya hemos mencionado, la ventana de prueba muestra las diferentes entradas y salidas de la interfase:



Prueba

- **Entradas digitales I1-I8**

I1-I8 son las entradas digitales de la Interfase. A ellas se conectan los llamados sensores. Las entradas digitales solamente aceptan los valores 0 y 1, o Sí o No. Hay varios componentes que se pueden conectar a las entradas digitales como sensores: interruptores (minipulsadores), fototransistores (sensores luminosos) o contactos Reed (sensores magnéticos).

Puedes comprobar la función de estas entradas conectando a la interfase (por ejemplo a I1) un minipulsador (37783) (utiliza los contactos 1 y 3 del pulsador). Al pulsar el pulsador aparece un ganchito en el indicador del I1. Si has conectado el controlador por otro sitio (contactos 1 y 2), aparece enseguida el ganchito y desaparece pulsando una tecla

- **Salidas de motor M1–M4**
M1 – M4 son las salidas de la interfase. Aquí se conectan los llamados actuadores, como por ejemplo motores, electroimanes o lámparas. Las 4 salidas de motor pueden controlarse tanto en velocidad en 8 niveles como en dirección. Para el control de la velocidad se usa el regulador deslizante, con la indicación en valor numérico de la velocidad al lado del regulador. Si quieres comprobar una salida, debes conectar un motor en ella, por ejemplo en la M1.
- **Salidas de lámpara O1–O8**
Las salidas de motor pueden utilizarse también como un par de salidas individuales. Con ello pueden activarse lámparas, pero también motores que se muevan solamente en una dirección (por ejemplo una cinta transportadora). Si quieres comprobar una de estas salidas, debes conectar una conexión de lámpara en la salida, por ejemplo O1. La otra conexión de la lámpara debes conectarla con la hembra de la interfase (\perp).
- **Entradas analógicas AX–AY**
Las entradas analógicas AX y AY miden la resistencia del sensor conectado. A ellas pueden conectarse las resistencias NTC para la medición de temperatura, potenciómetro, fotorresistencias o fototransistores.
- **Entradas analógicas A1–A2**
Estas dos entradas miden la tensión de 0–10V.
- **Sensores de distancia D1–D2**
A las entradas para los sensores de distancia D1 y D2 pueden conectarse solamente sensores de distancia especiales. Los sensores de distancia D1 y D2 se presentan como entradas digitales y también como entradas analógicas.
- **Módulos de extensión EM1–EM3**
Siempre y cuando haya módulos de extensión conectados a la interfase (hasta tres I/O-Extension Art. No. 93294 en la ROBO Interface, como máximo un módulo de extensión Art. No. 16554 en la interfase inteligente), pueden controlarse conmutando a través del registro en el margen superior a los módulos de extensión.

3 Nivel 1: Tu primer programa de control

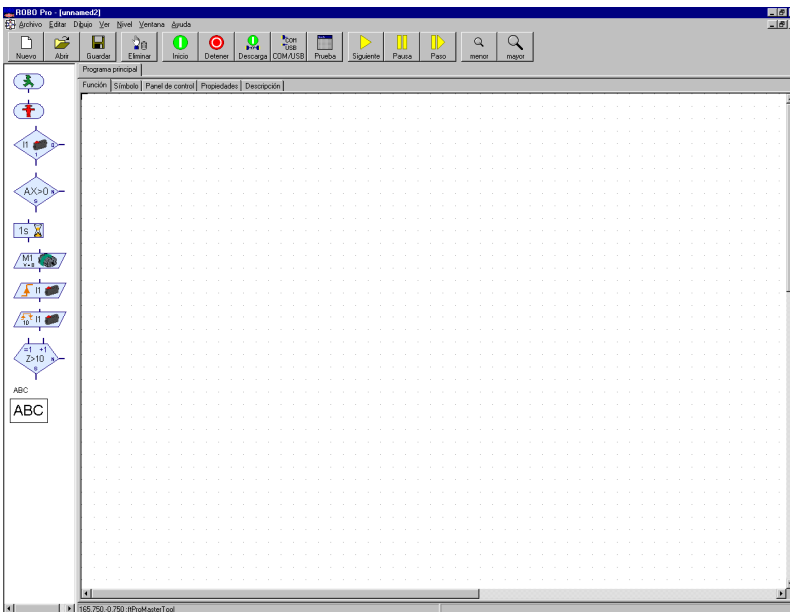
Una vez comprobado el hardware (es decir, la interfase y los pulsadores y motores conectados) en el último capítulo, queremos empezar ya con la programación. ¿Pero qué significa "programar"? Imagina que tenemos conectado a nuestra interfase un robot, por ejemplo. Lo que ocurre es que este robot es tan poco inteligente que no funciona por sí mismo. Por suerte nosotros somos algo más inteligentes, y podemos decirle al robot exactamente qué es lo que tiene que hacer. ¿Cómo? Recuerda lo que pasó cuando en el último capítulo pusimos la salida de motor M1 a "la izquierda" mediante la tecla del ratón. Exacto: conectamos el motor. Si este motor moviera, por ejemplo, la pinza de nuestro robot, lo que habríamos hecho sería, ni más ni menos, decirle al robot "¡Agarra el objeto!". Pero no queremos accionar cada paso manualmente, sino que el robot lo haga "automáticamente". Para ello debemos ir guardando cada paso de la rutina de tal manera que el robot pueda realizarlos uno tras otro. En otras palabras, debemos crear un programa que controle el robot en nuestro lugar. En lenguaje técnico esto se llama, lógicamente, un programa de control.

3.1 Crear un nuevo programa

Con el software ROBO Pro tenemos a nuestra disposición una herramienta estupenda para crear un programa de control de estas características y probarlo mediante una interfase conectada. No te preocupes: por ahora no queremos programar un robot. De momento nos conformamos con tareas de control sencillas. Para ello debemos crear un nuevo programa. En la barra de herramientas encontrarás la opción "Nuevo". Si haces clic sobre ella con el botón izquierdo del ratón, verás que se crea un nuevo programa vacío:



Nuevo



En ese momento verás una superficie de dibujo grande y blanca, en la que introducirás enseguida tu primer programa. Si en el margen izquierdo ves dos ventanas solapadas, debes cambiar en el menú **Nivel** a **Nivel 1: Principiantes**.

3.2 Los elementos del programa de control

Ahora podemos ponernos ya manos a la obra y crear nuestro primer programa de control. Queremos hacerlo mediante un ejemplo concreto:

Descripción de la función:

Imagínate una puerta de garaje de apertura automática. ¡Quizás incluso tengas una en casa! Si llegas en coche al garaje, basta con pulsar la tecla del mando a distancia y la puerta del garaje se abre, accionada por un motor. El motor debe estar en marcha hasta que la puerta del garaje quede completamente abierta.

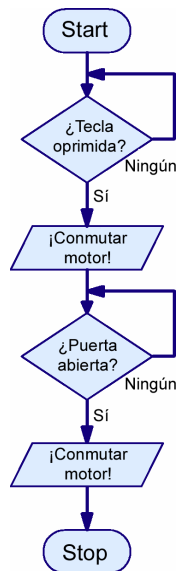
Describir un proceso de control con palabras resulta bastante complicado y no muy ilustrativo, y por esta razón se utilizan los llamados **diagramas de operaciones**, de control o de flujo para describir las sucesivas acciones de ejecución y las condiciones que deben cumplirse para que se ejecuten estas acciones. Por ejemplo, en nuestro caso la condición para que se ejecute la acción "Conectar el motor" es que se presione el pulsador. Interpretar un diagrama de operaciones de este tipo es bastante sencillo: simplemente hay que ir siguiendo las flechas. El diagrama de flujo da como resultado el funcionamiento exacto del control: cada paso puede ejecutarse solamente en el orden indicado por las flechas, nunca de otra forma. En caso contrario no necesitaríamos hacer todo este trabajo, ¿verdad?

Con ayuda de nuestro software ROBO Pro, ahora podemos dibujar este diagrama de operaciones detalladamente y con ello crear el **programa de control** para el hardware conectado (interfase, motores, pulsadores, etc.). Del resto se encarga el software, igual que ocurre en las grandes aplicaciones industriales. De ahí que ahora podamos concentrarnos totalmente en la creación del diagrama de flujo.

El diagrama de operaciones se construye a partir de los elementos de programa. ¿Otra palabreja más? ¡Qué va, no es para tanto! En ROBO Pro los elementos individuales con los que se crea el diagrama de control se llaman elementos de programa. La acción "conectar el motor" no quiere decir otra cosa que la instrucción de que la interfase debe efectivamente encender este motor, que está conectado en la interfase. Los elementos de programa disponibles se encuentran en la ventana de elementos del margen izquierdo.

3.3 Insertar, desplazar y modificar elementos de programa

Lo que debemos hacer ahora es crear, a partir de los elementos de programa que se encuentran en la ventana de elementos, el diagrama de operaciones de nuestro control de puerta del garaje. Todos los elementos de programa disponibles pueden tomarse en la ventana de elementos e insertarse en la ventana de programa.

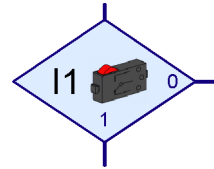


Insertar elementos de programa

Para hacerlo, lleva el ratón hasta el icono del elemento de programa deseado y haz clic sobre él con el botón izquierdo. A continuación lleva el ratón a la posición deseada dentro de la ventana de programa (es la superficie grande y blanca) y haz clic de nuevo. También puedes arrastrar el elemento de programa desde la ventana de elementos a la ventana de programa manteniendo presionada la tecla del ratón. Un programa siempre empieza con un elemento de inicio. El elemento de inicio es el elemento redondeado con un hombrecillo verde tipo semáforo. Lo mejor es que lo pruebes tu mismo con este elemento de programa: haz clic con el botón izquierdo del ratón sobre el elemento de inicio en la ventana de elementos, mueve el ratón hasta la ventana de programa y haz clic otra vez con el botón izquierdo del ratón.



El siguiente paso del diagrama de flujo es un elemento que consulta una entrada y, según el estado de la entrada, bifurca hacia una u otra ruta del programa. Haz clic en la ventana de elementos sobre el elemento ilustrado a la derecha y después lleva el ratón hasta debajo del elemento de inicio insertado anteriormente. Si la entrada superior del elemento de bifurcación está uno o dos puntos reticulares por debajo de la salida del elemento de inicio, aparecerá una línea de unión en la ventana de programa. Si haces clic de nuevo con el botón izquierdo del ratón, se inserta el elemento de bifurcación y se une automáticamente con el elemento de inicio.



Desplazar elementos de programa y grupos

Los elementos de programa pueden desplazarse a la posición deseada, incluso después de insertados, si los arrastras mientras mantienes presionada la tecla del ratón. Si quieres desplazar varios elementos juntos, puedes enmarcarlos con el ratón. Para ello debes hacer clic con el botón izquierdo del ratón en un área **vacía**, mantener el botón pulsado y arrastrar el puntero para abrir un rectángulo dentro del que queden enmarcados los elementos deseados. Entonces los elementos del rectángulo se verán con un borde rojo. Si desplazas uno de los elementos rojos con el botón izquierdo del ratón, se desplazan todos los elementos rojos. También puedes ir marcando todos los elementos en rojo haciendo clic sobre ellos con el botón izquierdo del ratón mientras mantienes pulsada la tecla de mayúsculas. Si con el botón izquierdo del ratón haces clic en un área vacía, todos los elementos marcados en rojo vuelven otra vez a su aspecto normal.

Copiar elementos de programa y grupos

Para copiar elementos de programa hay dos posibilidades. Puedes proceder de la misma forma que para desplazarlos, pero pulsando antes la tecla **CTRL**. De esta forma los elementos no se desplazan, sino que se copian. Lo que ocurre es que con esta función solamente puedes copiar los elementos dentro de un mismo programa. Si quieres copiar elementos de un programa a otro, puedes utilizar el **Portapapeles** de Windows. Primero selecciona los elementos que quieras copiar, tal y como fue descrito en la sección anterior (al explicar cómo desplazar elementos). Si una vez seleccionados los elementos pulsas **CTRL+C** o accedes a la opción de menú **Edición / Copiar**, todos los elementos seleccionados se copiarán en el portapapeles de Windows. Seguidamente ya puedes pasar a otro programa y reinserter los elementos mediante **CTRL+V** o **Edición / Pegar**. Incluso puedes insertar varias veces los elementos copiados. Si lo que quieres es mover elementos de un programa a otro, en vez de **CTRL+C** o **Edición / Copiar** debes utilizar la función **CTRL+X** o **Edición / Cortar**.



Eliminar

Eliminar elementos y función deshacer

Eliminar elementos resulta también muy fácil. Para eliminar todos los elementos marcados en rojo (ver párrafo anterior), pulsa la tecla suprimir (**Supr**). También puedes eliminar elementos individuales mediante la función de eliminación. Para ello haz primero clic sobre el botón de la barra de herramientas que se ilustra al lado de estas líneas y después sobre el elemento que quieres eliminar. No tengas miedo de probar: siempre podrás dibujar de nuevo el elemento borrado. Para recuperarlo también puedes utilizar la función **Deshacer** del menú **Edición**. Esta opción del menú te permite deshacer todos los cambios realizados en el programa.

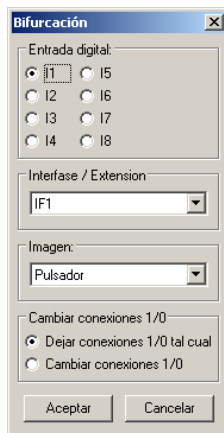
Editar propiedades de elementos de programa

Si con el botón **derecho** del ratón haces clic sobre un elemento de programa de los mostrados en la ventana de programa, aparece una ventana de diálogo en la que puedes cambiar las propiedades del elemento. La imagen de la derecha ilustra la ventana de propiedades de un elemento de bifurcación.

- Con los botones **11** a **18** puedes indicar, qué entrada de la interfase se debe consultar.

La selección **Interfase / Extensión** se explicará en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.

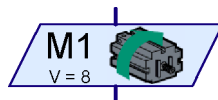
- En **Imagen** puedes seleccionar una imagen para el sensor conectado en la entrada. Las entradas digitales en la mayoría de los casos son utilizadas con pulsadores, pero es frecuente que se utilicen también con fototransistores o contactos Reed.
- En **Cambiar conexiones 1/0** puedes cambiar la posición de las salidas 1 y 0 de la bifurcación. Normalmente la salida 1 se encuentra abajo y la salida 0 a la derecha, pero muchas veces es más práctico que la salida 1 esté a la derecha. Si haces clic sobre **Cambiar conexiones 1/0**, se cambiarán las conexiones 1 y 0 en cuanto cierres la ventana con OK.



Nota: Si conectas un minipulsador como cierre en la conexión 1 y 3 del pulsador, la bifurcación del programa irá a la rama 1 si el pulsador está presionado y a la rama 0 si no lo está.

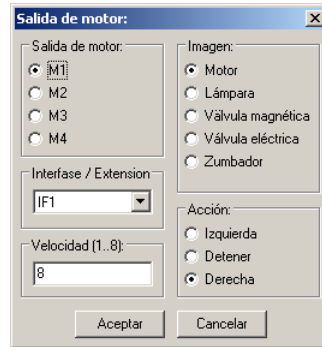
Si conectas un minipulsador como apertura en la conexión 1 y 2 del pulsador, la bifurcación del programa irá a la rama 1 si el pulsador está presionado y a la rama 0 si no lo está.

El siguiente elemento de programa en nuestro control de puerta de garaje es un elemento de motor. Debes introducirlo en el programa igual que los dos elementos anteriores, es decir, debajo del elemento de bifurcación. Lo mejor es que vuelvas a situar el elemento de forma que se conecte automáticamente con el elemento superior.



Con el elemento de motor puedes conectar o desconectar tanto un motor como una lámpara o un electroimán. Igual que antes, puedes abrir la ventana de propiedades del elemento de motor con un clic del botón derecho del ratón sobre el elemento.

- Con los botones **M1** a **M4** puedes elegir qué salida de la interfase debe activarse.
- En **Imagen** puedes seleccionar una imagen que muestre el componente de fischertechnik conectado en la salida.
- La selección **Interfase / Extensión** se explicará en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Acción** puedes seleccionar de qué manera se actuará en la salida. Puedes arrancar o parar un motor mediante sentido de giro a la izquierda o a la derecha, o conectar o desconectar una lámpara.
- En **Velocidad/Intensidad** puedes ajustar la velocidad con la que debe girar el motor o la intensidad con la que debe encenderse la luz de la lámpara. Los valores posibles van desde 1 a 8.

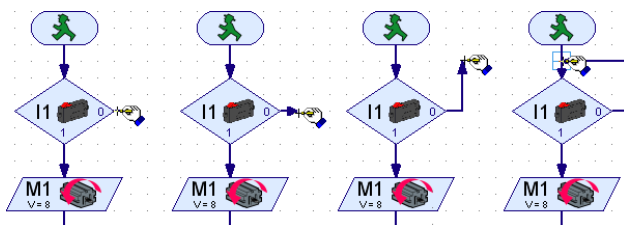
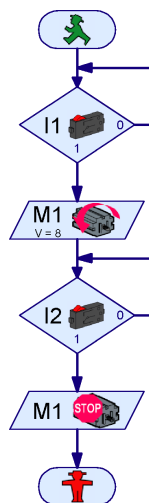


Para nuestro diagrama de programa necesitamos la instrucción **Motor M1 a la izquierda con velocidad 8**.

3.4 Interconectar los elementos de programa

Una vez que ya sabes cómo introducir elementos en un programa de control, podemos empezar a completar nuestro programa de control. Piensa en la descripción de la función del control de la puerta de garaje. ¿No falta algo? Correcto: hemos conectado el motor mediante la pulsación de una tecla, ¡pero después de abrir la puerta, el motor debe desconectarse automáticamente! En la práctica esto puede hacerse con un interruptor llamado de fin de carrera. Es un pulsador que se instala en la puerta del garaje de tal manera que se acciona en el momento en que el motor ha abierto la puerta totalmente, y durante la conexión del motor puede usarse esta señal para desconectarlo de nuevo. Para la consulta del interruptor de fin de carrera podemos utilizar otra vez el elemento de bifurcación.

Vamos, entonces, a introducir en el programa de control otro elemento de bifurcación, para que consulte el interruptor de fin de carrera de la entrada I2. No te olvides de hacer clic con el botón derecho del ratón sobre el elemento y de cambiar la entrada a I2. En cuanto la puerta de garaje esté abierta y el interruptor de fin de carrera esté presionado, el motor debe pararse de nuevo. Esto se consigue mediante un elemento de motor. Primero utilizas el mismo elemento que para la conexión del motor. Si haces clic con el botón derecho del ratón sobre él, puedes cambiar la función del elemento a **Detener motor**. El programa se termina con un elemento de fin. Tu programa debería ahora parecerse al de la ilustración de la derecha. Si has situado unos elementos directamente debajo de otros siempre con una distancia entre sí de uno o dos puntos reticulares, la mayoría de las entradas y salidas estarán ya interconectadas mediante flechas de flujo del programa. Pero la salida de No (N) de las dos bifurcaciones aún no estará conectada. Mientras no se presiona el pulsador de la entrada I1, el programa tiene que retroceder y consultar el interruptor de nuevo. Para dibujar esta línea, debes ir haciendo clic con el ratón sucesivamente sobre las posiciones indicadas en la imagen siguiente.



Nota: Si una línea no está conectada correctamente con una conexión o con otra línea, se mostrará un rectángulo verde en la punta de la flecha. En este caso debes establecer la conexión desplazando la línea o eliminando y redibujando. En caso contrario el proceso del programa no funcionará en esta posición.

Eliminar líneas de flujo del programa

La eliminación de líneas funciona igual que la eliminación de elementos de programa. Simplemente debes hacer clic con el botón izquierdo del ratón sobre la línea, para hacer que se marque en rojo. A continuación pulsas la tecla de suprimir (**Supr**) del teclado para borrar la línea. También puedes seleccionar varias líneas manteniendo pulsada la tecla de mayúsculas (es la tecla que cambia de mayúsculas a minúsculas y viceversa) y después haciendo clic sucesivamente sobre las líneas con el botón izquierdo del ratón. Además también puedes generar un marco alrededor de estas líneas para marcar varias a la vez y eliminar así de una vez todas las líneas marcadas en rojo con la tecla **Supr**.

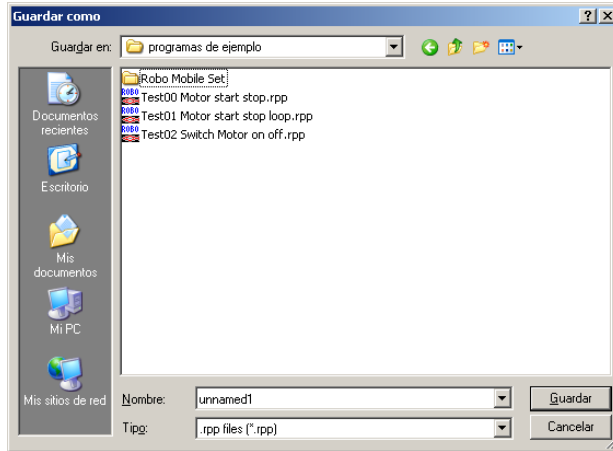
3.5 Probar el primer programa de control

Para poder probar nuestro primer programa de control, deberás construir un pequeño modelo. Es suficiente con que conectes un pulsador en la interfase en I1 y I2 y un motor en M1.

Nota: La conexión de la interfase al PC y su ajuste se efectuó ya en el capítulo anterior, donde puede consultarse si es necesario.

Antes de probar el programa de control debes guardar el archivo del programa en el disco duro del ordenador. Haz clic con el ratón sobre la instrucción **Guardar como** del menú **Archivo**. Enseguida aparece la siguiente ventana de diálogo:

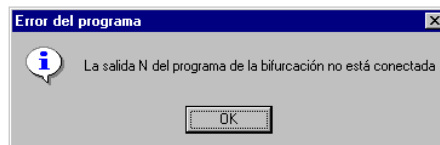
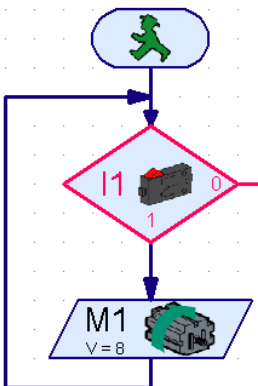
Selecciona en "Guardar como" el directorio en el cual quieres guardar el programa. Introduce en "Nombre de archivo" un nombre que todavía no hayas utilizado, por ejemplo PUERTA DE GARAJE, y confirma haciendo clic con el botón izquierdo del ratón sobre "Guardar".



Para probar el programa haz clic sobre el botón de Inicio, a la izquierda de la barra de herramientas. Lo primero que hace la ROBO Pro es comprobar si todos los elementos del programa están interconectados correctamente. En caso de que algún elemento no esté interconectado correctamente o que haya cualquier otra cosa incorrecta, se marcará con rojo y se mostrará un aviso de error que describe el problema. Si por ejemplo has olvidado conectar la salida de No (N) de la bifurcación del programa, aparece lo siguiente:



Inicio

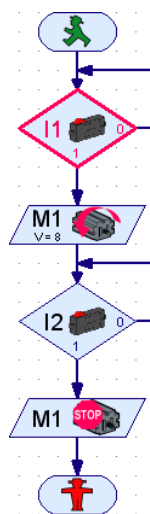


Si has recibido un aviso de error, debes primero corregir el error avisado. De otra manera no se iniciará el programa.

Nota: Encontrarás una explicación más detallada de este modo de servicio y del modo de descarga en el capítulo 3.7, página 460.

El primer elemento de bifurcación se ve marcado en rojo. Esto indica que el proceso está esperando un suceso en este elemento de programa: debe presionarse el pulsador de I1 que abre la puerta de garaje. Mientras no se presione el pulsador, el programa bifurca hacia la salida de No (N) y desde ahí retrocede al inicio de la bifurcación. Presiona ahora el pulsador que está conectado en la entrada I1 de la interfase. Con ello se ha cumplido la condición para continuar y se conecta el motor. El proceso está esperando en el próximo paso a que se presione el interruptor de fin de carrera de la entrada I2. En cuanto hayas accionado el interruptor de fin de carrera de I2, el programa bifurcará hacia la segunda salida del motor y lo desconectará. Finalmente el programa llega al final y aparece un aviso de que el programa se ha finalizado.

¿Ha funcionado todo bien? Si es así, ¡enhorabuena! Has creado y probado tu primer programa de control. Si no ha funcionado, no te desanimes: comprueba otra vez todo, puesto que seguramente se deba a que se haya colado un error en alguna parte. Todos los programadores cometen errores de vez en cuando y equivocarse es la mejor forma de aprender. Así que ¡ánimo!



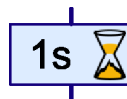
3.6 Más elementos de programa

Si has probado tu primer programa de control en un modelo auténtico de puerta de garaje, la puerta estará abierta. ¿Pero cómo cerrarla de nuevo? Es evidente que podríamos iniciar el motor volviendo a presionar un pulsador, pero ahora queremos probar otra solución y aprovechar para familiarizarnos con un nuevo elemento de programa. Para ello, lo primero que haremos será guardar el programa con un nuevo nombre (ya que más adelante necesitaremos volver a usar el diagrama de operaciones con el que hemos trabajado hasta ahora). Utiliza para ello la opción **Guardar como...** del menú **Archivo** e introduce un nuevo nombre de archivo que no hayas utilizado aún.

3.6.1 Tiempo de espera

Antes de que podamos ampliar el programa de operaciones, debes eliminar la conexión entre “Desconectar motor” y “Fin de programa” y desplazar el elemento de fin hacia abajo. Ahora ya puedes insertar los nuevos elementos de programa entre estos dos puntos. Vamos a hacer que la puerta de garaje se cierre automáticamente pasados 10 segundos. Para hacerlo puedes utilizar el

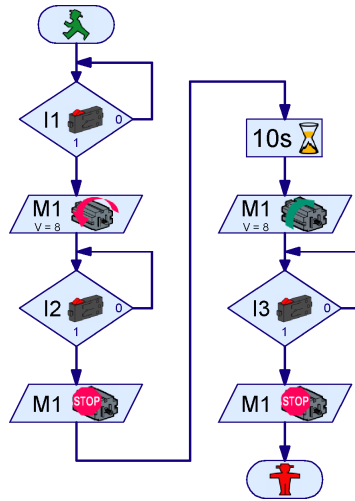
elemento de programa **Tiempo de espera** que ves a la derecha. Puedes configurar el tiempo de espera a tu gusto dentro de un amplio margen, haciendo clic como de costumbre con el botón derecho del ratón sobre el elemento. Introduce el tiempo de espera deseado: 10 segundos. Para cerrar la puerta, el motor debe (naturalmente) moverse en dirección contraria, es decir, a la derecha. Se desconecta el motor mediante otro interruptor de fin de carrera conectado en I3.



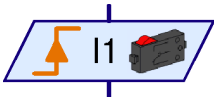


Inicio

Tu diagrama de operaciones ya terminado debería parecerse más o menos a la ilustración de la derecha. Hemos desplazado los nuevos elementos de programa a la derecha para que se vean mejor. Si has comprobado que en el diagrama de operaciones ya no hay errores, puedes probar el control ampliado de la puerta de garaje como de costumbre con el botón de **Inicio**. Accionando el pulsador de I1 se conecta el motor, que se desconecta de nuevo accionando el pulsador de I2. Hemos abierto la puerta de garaje. El elemento de programa de tiempo de espera de 10 segundos, que es nuestro tiempo de espera configurado, se ve marcado en rojo. A continuación se conectará el motor con dirección de giro inversa hasta que se accione el pulsador de I3. Prueba un par de veces más cambiando el tiempo de espera.



3.6.2 Esperar a entrada



Aparte del elemento tiempo de espera hay dos elementos más cuya función consiste en aguardar a que ocurra algo para poder continuar la ejecución del programa. El elemento **Esperar a entrada** (a la izquierda) espera hasta que una entrada de la interfase adopte un determinado estado o

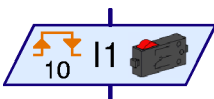
hasta que varíe de determinada manera. Existen 5 variaciones de este elemento:

Símbolo					
Esperar a	Entrada=1 (cerrado)	Entrada=0 (abierto)	Cambio 0-1 (de cerrado a abierto)	Cambio 1-0 (de abierto a cerrado)	Cualquier cambio (1-0 ó 0-1)
Misma función pero con bifurcación					



Para ello podemos utilizar también una combinación de elementos de bifurcación, pero con el elemento de **Esperar a entrada** es más fácil y se entiende mejor.

3.6.3 Contador de impulsos

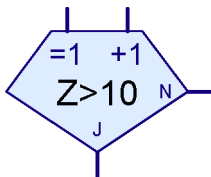


Muchos modelos de robot de Fischertechnik utilizan también ruedas dentadas de impulsos. Estas ruedas dentadas accionan un pulsador 4 veces por cada giro. Con esas ruedas dentadas puede conectarse un motor ya no en un momento determinado, sino conforme a un número de revoluciones exactamente definido. Para ello debe contarse el

número de impulsos recibidos en una entrada de la interfase. Para este fin está el elemento **Con-**

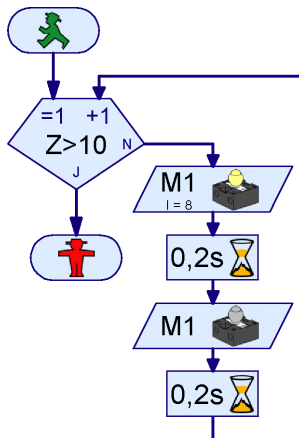
tador de impulsos ilustrado a la izquierda, cuya función es esperar a que se registre un número regulable de impulsos. En este elemento también puedes configurar si debe entenderse como impulso cualquier cambio o solamente cambios de 0 a 1 o de 1 a 0. Las ruedas de impulsos esperan generalmente cualquier cambio, de modo que con 4 dientes se obtiene una definición de 8 pasos por revolución.

3.6.4 Bucle contador



El elemento de bucle contador te permite ejecutar varias veces una parte determinada del programa de una manera muy fácil. El programa que vemos en la ilustración, por ejemplo, puede encender y apagar 10 veces una lámpara conectada en **M1**. El

elemento de bucle contador tiene un contador incorporado; si se accede al bucle contador por la entrada =1, el contador se pone a 1. Pero si por el contrario se accede al bucle contador por la entrada +1, se añade un 1 al contador. Dependiendo de si el contador registra o no registra un número mayor a un valor prefijado por ti, el bucle contador bifurca hacia la salida Sí (**S**) o No (**N**). Así, se utiliza la salida Sí cuando se ha pasado el bucle tantas veces como habías prefijado tú en el valor del contador. Si por el contrario se necesitan más pasos de bucle, el bucle contador bifurca hacia la salida No. Igual que en el elemento de bifurcación, puedes intercambiar la salida de Sí y No desde la ventana de propiedades.



3.7 Modo online o modo de descarga.

¿Cuál es la diferencia?



Inicio

Hasta ahora hemos probado nuestros programas de control en el llamado **modo online**. En este modo puedes seguir el avance de los programas en la pantalla, dado que se van marcando en rojo los correspondientes elementos activos. Este modo se utiliza para entender los programas o para buscar errores en ellos.



Pausa

En el modo online también puedes detener el programa y hacer que continúe después pulsando el botón de **Pausa**. Esto resulta muy práctico si quieres examinar alguna parte de tu modelo sin detener el programa completamente. La función de pausa también te puede resultar muy útil si intentas comprender el desarrollo de un programa.



Paso a paso

Con el botón de **Paso a paso** puedes ejecutar el programa en pasos individuales, elemento por elemento. Cada vez que pulsas el botón, el programa saltará al siguiente elemento de programa. Si ejecutas un elemento de **Tiempo de espera** o un elemento de **Esperar a**, naturalmente puede pasar un rato hasta que el programa salte al siguiente elemento.



Descarga

Si tienes una ROBO Interface (no una interfase inteligente), en vez del modo online puedes usar también el **modo de descarga**. En el modo online se ejecutan los programas desde tu ordenador. El ordenador envía instrucciones de control (como "conectar motor") a la interfase; para que ella los reciba, debe estar conectada al ordenador mientras se ejecuta el programa. Pero en el modo de descarga, el programa lo ejecuta la propia interfase. El ordenador almacena el programa en la

interfase y en ese momento puedes desconectar ya ordenador e interfase. A partir de ese momento la interfase puede ejecutar el programa de control independientemente del ordenador. El modo de descarga es importante, por ejemplo, en la programación de robots móviles, para los que un cable de conexión entre el PC y el robot sería un estorbo. No obstante, los programas de control deberían probarse primero en modo online, dado que así resulta más fácil detectar los posibles errores. Después puede transferirse el programa ya probado a la ROBO Interface mediante la funcionalidad de descarga. En la ROBO Interface puede prescindirse del cable, si es molesto, y utilizar la conexión de radio **ROBO RF Data Link**, Art. No. 93295. Con ello el modelo será absolutamente móvil también en el modo online.

Pero el modo online tiene también ventajas frente al modo de descarga. Un ordenador tiene mucha más memoria en comparación con la interfase y trabaja mucho más rápido, lo que resulta muy conveniente con los programas grandes. Además, en el modo online pueden activarse varias interfaces en paralelo, incluso combinando ROBO Interfaces e interfaces inteligentes.

Resumen de los dos modos de servicio:

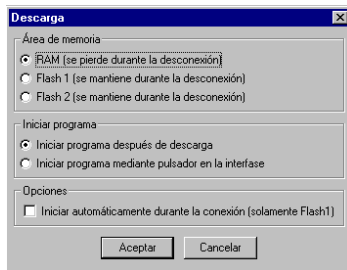
Modo de servicio	Ventajas	Desventajas
Online	<ul style="list-style-type: none"> • Puede seguirse la ejecución del programa en la pantalla • Los programas (incluso los más grandes) se ejecutan con gran rapidez • Pueden activarse varias interfaces en paralelo • Soporta la interfase inteligente más antigua • Pueden utilizarse paneles de control • Puede detenerse y continuarse el programa. 	<ul style="list-style-type: none"> • Ordenador e interfase deben mantenerse conectados entre sí
Descarga	<ul style="list-style-type: none"> • Pueden separarse ordenador e interfase después de la descarga • 	<ul style="list-style-type: none"> • No soporta la interfase inteligente más antigua • No puede seguirse la ejecución del programa en la pantalla. • Puede activarse el programa hasta un máximo de 3 módulos de extensión



Descarga

Utilizar el modo de descarga

Así pues, si tienes la nueva ROBO Interface, puedes transferir tu control de puerta de garaje a la interfase haciendo clic en el botón de **Descarga** de la barra de herramientas. Primeramente se mostrará la ventana de diálogo ilustrada a la derecha. La ROBO Interface dispone de varias ubicaciones de registros de programa, un área de **RAM** (Random Access Memory) y dos áreas de **Flash**. La RAM conserva los programas solamente mientras esté conectada a la red eléctrica o los acumuladores le proporcionen energía. Por el contrario un programa guardado en Flash queda almacenado en la interfase durante muchos años, incluso sin corriente. Esto no significa que no puedas sobrescribir los programas guardados en Flash en un momento dado, pero la descarga a RAM funciona claramente más rápido y por ello es recomendada para fines de prueba.



En las dos áreas de Flash puedes almacenar dos programas distintos, por ejemplo dos maneras de comportamiento distintas para un robot móvil. Después puedes seleccionar, iniciar y detener los dos programas con la tecla **Prog** de la interfase. Si la opción **Iniciar programa después de la descarga** está activada, se iniciará el programa inmediatamente después de la descarga, como su nombre indica. Mientras que se ejecute el programa, en la ROBO Interface parpadeará en verde bien el LED **Prog 1** (programa cargado en Flash 1) o bien **Prog 2** (programa cargado en Flash 2) al lado de la tecla **Prog**. Si se ha cargado un programa en la RAM, parpadearán ambos LED. Para detener el programa debes pulsar la tecla **Prog**, con lo cual el LED quedará encendido permanentemente. Para cambiar del programa 1 al programa 2, mantén pulsada la tecla Prog hasta que se ilumina el LED del programa 1 ó 2. Para iniciar el programa, pulsa de nuevo la tecla.

En los robots móviles es más práctica la opción **Iniciar programa desde el pulsador de la interfase**. Si no dispones de un RF Data Link, tendrás que quitar el cable antes de que el programa ponga el robot en movimiento. En este caso debes seleccionar primero el programa deseado con la tecla Prog de la interfase e iniciar el programa pulsando de nuevo.

Si está activada la última opción **Iniciar automáticamente durante la conexión**, se iniciará automáticamente el programa 1 de Flash tan pronto como la interfase reciba alimentación de corriente. De esta forma puedes, por ejemplo, colocar un temporizador en la fuente de alimentación de la interfase de modo que el programa se inicie cada día a la misma hora. Así no tendrás que dejar conectada la interfase ni iniciar el programa con la tecla Prog después de cada conexión.

Notas:

Si se carga un programa en o se ejecuta desde Flash, se perderán los programas cargados en la RAM, porque los programas de Flash utilizan igualmente la memoria RAM.

En el manual de la interfase encontrarás una descripción más detallada de las funciones de la ROBO Interface.

3.8 Consejos y trucos

Modificar las líneas de unión

Si desplazas elementos, la ROBO Pro tratará de adaptar las líneas de unión de manera razonable. En caso de que no te guste una línea adaptada, puedes modificar las líneas de unión fácilmente haciendo clic con el botón izquierdo del ratón sobre la línea y desplazándola manteniendo pulsado el botón del ratón. Dependiendo de en qué punto de la línea se encuentre el ratón, se desplazará un ángulo o un borde de la línea. La distinta modalidad se mostrará mediante los distintos punteros del ratón:



Si el ratón se encuentra sobre una línea vertical de unión, puedes desplazar toda la línea vertical con la tecla izquierda del ratón pulsada.



Si el ratón se encuentra sobre una línea horizontal de unión, puedes desplazar toda la línea horizontal con la tecla izquierda del ratón pulsada.



Si el ratón se encuentra sobre una línea diagonal de unión, se insertará un nuevo punto en la línea de unión si pulsas la tecla izquierda del ratón. Debes mantener pulsada la tecla izquierda del ratón y soltarla sólo cuando el ratón se encuentre donde quieres tener el nuevo punto.

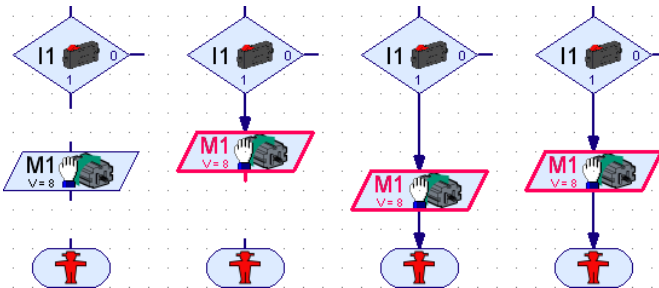


Si el ratón se encuentra cerca de un punto angular o un punto final de una línea de unión, puedes desplazar el punto con la tecla izquierda del ratón pulsada. Puedes arrastrar un punto final de línea unida solamente a otra unión adecuada de un elemento de programa.

En este caso se une el punto final de la línea de unión con este punto de unión. En caso contrario el punto no se desplazará.

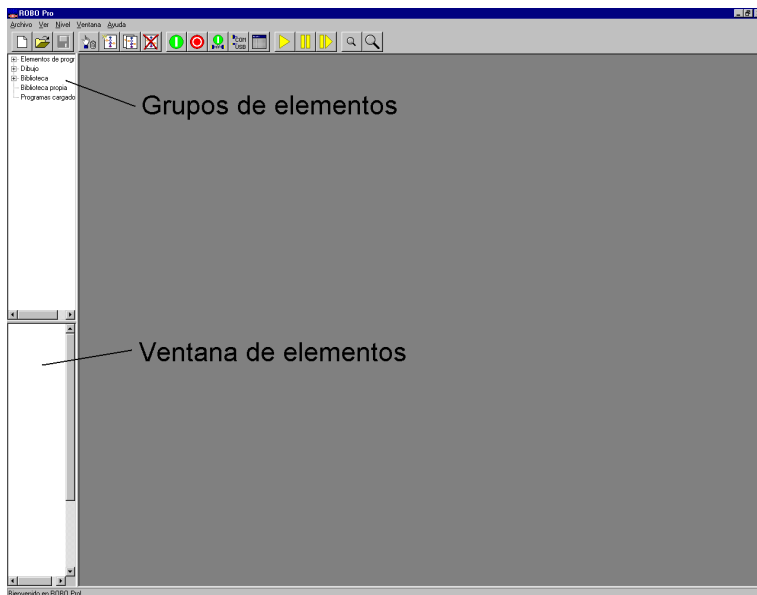
Otra manera de crear líneas de unión

También pueden crearse líneas de unión desplazando de los elementos de programa. Si desplazas un elemento de programa de tal forma que la entrada quede uno o dos puntos reticulares por debajo de la salida de otro, se creará una línea de unión entre ambos elementos. Esto también ocurre si una salida es desplazada por encima de una entrada. Después puedes desplazar el elemento de programa a su posición final o dibujar más uniones para las entradas y salidas restantes:



4 Nivel 2: Trabajar con subprogramas

Después de haber creado y probado con éxito tu primer programa de control, estarás preparado para proceder con el nivel 2 de ROBO Pro. Elige en el menú **Nivel** la opción **Nivel 2: Subprogramas**. Seguramente notarás enseguida la diferencia: la ventana de elementos ya no está y en su lugar puedes ver ahora, en el margen izquierdo, dos ventanas solapadas:



¡Pero no te preocupes! La ventana de elementos sigue estando ahí, lo que ocurre es que está vacía. En el nivel 2 hay más elementos de programa, así que sería poco práctico meterlos todos en una sola ventana. Por eso los elementos a partir del nivel 2 se han recogido en grupos. Los elementos se organizan en grupos de forma muy parecida a la que se usa en el disco duro del ordenador para organizar los archivos en carpetas. Si seleccionas un grupo de los que aparecen en el margen izquierdo de la ventana superior, se desplegarán en la ventana inferior todos los elementos de este grupo. Puedes encontrar los elementos del nivel 1 en el grupo de **Elementos de programa / Elementos básicos**. Dado que la ventana de elementos se ha quedado en la mitad del tamaño original, para mostrar los elementos inferiores debes hacer uso de la barra de desplazamiento que encontrarás a la derecha de la ventana de elementos.

Pero vayamos ahora al tema que nos interesa realmente: los subprogramas. Es cierto que los diagramas de operaciones que hemos creado hasta ahora no son tan amplios que nos hagan perder la visión de conjunto, pero, como seguramente imaginarás, es muy fácil que esto ocurra en proyectos más grandes con diagramas de operaciones más amplios. De repente nos encontramos con que la hoja de trabajo está llena de componentes, tenemos líneas de unión por todas partes y continuamente nos vemos obligados usar la barra de desplazamiento para movernos por la pantalla. "¿Pero dónde estaría tal o cual salida?", nos preguntamos. Resumiendo: ¡caos a la vista! ¿Qué podemos hacer? ¿Ya no existe ninguna posibilidad de poner un poco de orden en todo esto? ¡Claro que sí! ¡Utilizar un **subprograma**!

4.1 Tu primer subprograma

Un subprograma se parece mucho a los programas que has conocido hasta ahora. Para examinarlo más detalladamente, vamos a empezar por crear un nuevo programa y dentro de él crear un nuevo subprograma vacío. Para hacerlo, pulsa sobre programa **Nuevo** y después sobre el botón **Nuevo SP** de la barra de herramientas. Se abre una ventana que te permite introducir el nombre y la descripción del subprograma.

El nombre no debería ser demasiado largo (aproximadamente de 8 a 10 caracteres), porque si no el símbolo del subprograma quedaría demasiado grande. Naturalmente siempre estarás a tiempo de cambiar más adelante todo lo que introduzcas ahora.

En cuanto cierras la ventana **Nuevo subprograma** con **OK**, aparece un nuevo subprograma en la barra de subprogramas:

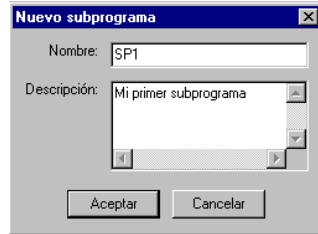


Puedes moverte en cualquier momento del programa principal al subprograma haciendo clic sobre el nombre pertinente de la barra de subprogramas. Dado que ambos programas aún están vacíos, todavía no se ve ninguna diferencia.

Vamos ahora a estructurar en subprogramas el control de la puerta de garaje del capítulo anterior (vea sección 3.6 *Más elementos de programa*, página 458). El programa consta de cuatro unidades de función:

- Esperar hasta que el pulsador I1 esté pulsado
- Abrir puerta
- Esperar 10 segundos
- Cerrar puerta

Ahora queremos separar los procesos de abrir y cerrar en dos subprogramas. Después puedes llamar a los dos subprogramas desde el programa principal con un solo símbolo. El pulsador I1 de espera y el tiempo de espera de 10 segundos se mantienen en el programa principal, dado que constan de un solo elemento cada uno y no supondría mucha diferencia hacerles un subprograma. Acabas de crear un nuevo programa con un subprograma dentro, llamado **Subprograma 1**; mejor sería darles los nombres **Abrir** y **Cerrar** a los dos subprogramas. Para cambiar el nombre del subprograma ya creado, selecciona el subprograma 1 en la barra de subprogramas, si no está ya seleccionado.

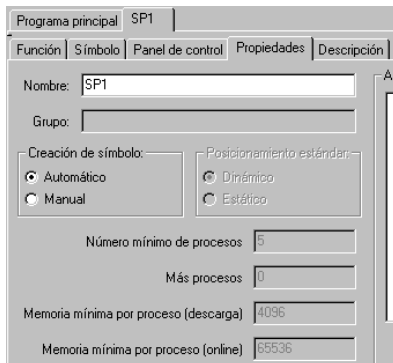


A continuación, pasa a la ventana de propiedades del subprograma en cuestión haciendo clic sobre **Propiedades**. Aquí puedes cambiar el nombre de **SP 1** a **Abrir**. Casi todos los demás campos pueden cambiarse solamente en el nivel avanzado o incluso en el nivel experto. Explicaremos el punto **Crear símbolo** más adelante.

Si en la barra de función haces clic sobre la descripción, podrás cambiar la que habías introducido antes, aunque "Mi primer subprograma" sigue siendo una buena opción.

Ve a la barra de función y haz clic sobre **Función** para poder programar la función del subprograma.

Ahora ves de nuevo la ventana de programa en la que habías insertado los elementos de programa cuando creaste tu primer programa ROBO Pro en el capítulo anterior. Fíjate en que esté seleccionado el subprograma **Abrir** en la barra de subprogramas:

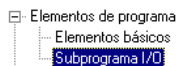


¿Estás preparado para programar tu primer subprograma? Muy bien: ¡vamos allá! Bueno, pero ¿cómo empieza realmente un subprograma? ¡Buena pregunta! Hasta ahora habías empezado todos los programas principales con el elemento de inicio. Pues bien: un subprograma empieza con un elemento parecido: la entrada de subprograma. Se llama así al elemento porque, a través de este elemento, el control del programa pasa del programa principal al subprograma. Aquí no puedes utilizar un elemento de inicio, porque no queremos iniciar un nuevo proceso.





	Elemento de inicio	Inicia un nuevo proceso independiente
	Entrada de subprograma	Aquí el control del programa pasa del programa principal al subprograma

Encontrarás la entrada de subprograma en la ventana de grupos de elementos (en **Subprograma I/O**). Coloca la entrada de subprograma en la parte de arriba de la ventana del subprograma **Abrir**. Aunque puedes ponerle al elemento de entrada de subprograma otro nombre que no sea **Entrada**, solamente tendrás que cambiarlo si más adelante vas a querer escribir un subprograma con varias entradas.



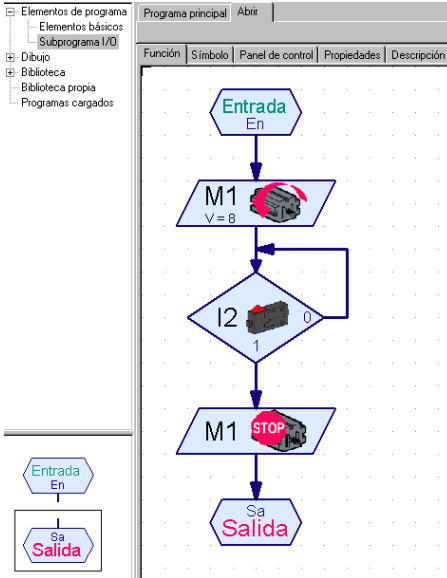
A partir de aquí, el desarrollo de la sección inicial es igual que si se tratase de un programa principal como los que hemos hecho hasta ahora: conectas el motor M1 con sentido de giro a la izquierda, esperas hasta que el pulsador de la entrada I2 esté cerrado y después desconectas de nuevo el motor.

Para finalizar el subprograma, utilizas una salida de subprograma. La diferencia entre la salida de subprograma y el elemento de detener es la misma que existe entre la entrada de subprograma y el inicio de proceso.

	Elemento de detener	Finaliza la ejecución de programa de un proceso independiente
	Salida de subprograma	Aquí se devuelve el control del programa, que se había pasado al subprograma, al programa principal.

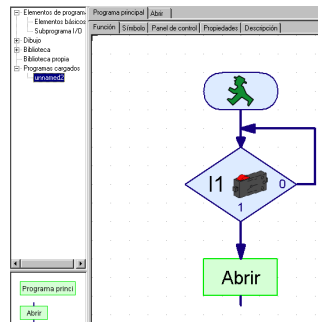


Tu subprograma terminado debería ahora tener más o menos este aspecto:



Verifica que de verdad hayas registrado el subprograma como **Abrir** y no como **Programa principal**. Ve ahora a la barra de subprogramas para pasar de nuevo desde **Abrir** a **Programa principal**. Verás la ventana de programa del programa principal, que sigue estando vacía. En el programa principal inserta, como siempre, un elemento de inicio (¡no una entrada de subprograma!). La consulta del pulsador de **I1**, que debe abrir la puerta de garaje, la ejecutarás como de costumbre en el programa principal.

Ahora puedes insertar tu nuevo subprograma como un elemento de programa habitual en tu programa principal (u otro subprograma). Lo encontrarás en la ventana de grupos de elementos, en **Programas cargados** y con el nombre de archivo de tu programa. Si todavía no has guardado el archivo, su nombre será **nombre1**. Si has cargado más archivos de programa, puedes seleccionar en la ventana de selección también los subprogramas que pertenecen a otros archivos. Esto te permite utilizar fácilmente subprogramas de otro archivo.



En el grupo de elementos **Programas cargados / sin nombre1** verás dos símbolos de subprograma en verde. El primero (de nombre **Programa principal**) es el símbolo del programa

principal. Aunque es menos frecuente, es posible utilizarlo también como subprograma, por ejemplo si estás controlando todo un parque de maquinaria y habías desarrollado previamente los controles para cada máquina individualmente como programas principales. El segundo símbolo (de nombre **Abrir**) es el símbolo de tu nuevo subprograma. **Abrir** es el nombre que habías introducido en propiedades. Inserta ahora el símbolo del subprograma en tu programa principal (tal y como estás acostumbrado a hacer con los elementos de programa habituales). ¡Así de simple!

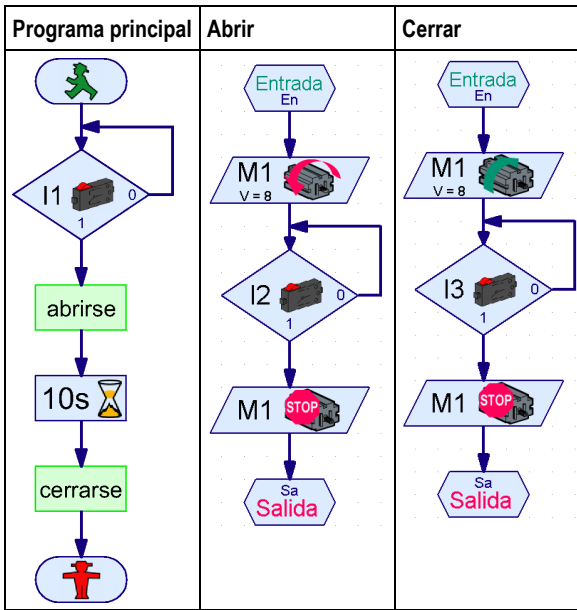


Nuevo SP

Si quieres, puedes concluir el programa principal con un elemento de detener y proceder a probarlo. La puerta ya se abre con el pulsador I1, pero todavía nos falta programar la parte de cerrar. Para hacerlo, vamos a crear otro subprograma. Pulsa el botón de **Nuevo SP** en la barra de herramientas e introduce en la ventana **Nuevo subprograma** el nombre **Cerrar**. No necesitas introducir una descripción (aunque no estaría mal, para saber después cuál es la función para la que está pensado el subprograma).

Introduce ahora en la ventana de programa del subprograma **Cerrar** el programa que cierra la puerta de garaje. Empiezas otra vez con una entrada de subprograma. El motor **M1** debe girar primero a la derecha. Tan pronto como el pulsador de límite de I3 esté cerrado, debe detenerse el motor **M1**. Se finaliza el subprograma de nuevo con una salida de subprograma.

Ahora debes pasar al programa principal mediante la barra de subprogramas. Si antes has conectado tu programa principal con un elemento de detener para poder probarlo, ahora es el momento de borrar ese elemento. Una vez que la puerta de garaje esté abierta, debe mantenerse así durante 10 segundos antes de cerrarse de nuevo. Después de un tiempo de espera de 10 segundos, introduce desde el grupo de elementos **Programas cargados / sin nombre1** el símbolo **Cerrar** del subprograma. El programa principal y los dos subprogramas deberían más o menos tener este aspecto:

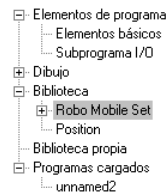


El programa se inicia en el elemento de inicio del **programa principal**. Después el programa espera hasta que se active el pulsador **I1**. Para conseguirlo también podrías utilizar el elemento **Esperar a entrada** (vea sección 8.1.8 *Esperar a entrada*, página 515). Después de haber activado el pulsador **I1**, el programa principal se encuentra con la llamada al subprograma **Abrir**. La ejecución del programa pasa con ello a la entrada de subprograma del subprograma **Abrir**. El subprograma **Abrir** abre la puerta de garaje y llega después a su salida de subprograma. En este punto el programa bifurca de nuevo hacia el programa principal. Después del final del subprograma **Abrir** se produce una espera de 10 segundos en el programa principal. A continuación la ejecución del programa cambia al subprograma **Cerrar**, que cierra la puerta de garaje de nuevo. Después del regreso desde el subprograma **Cerrar**, el programa principal se encuentra con un elemento de detener, con lo que se finaliza el programa.



4.2 La biblioteca del subprograma

Puedes copiar subprogramas de un archivo a otro de un modo bastante sencillo cargando ambos archivos y valiéndote después del grupo de elementos **Programas cargados** para insertar después un subprograma de un archivo dentro de otro archivo. Para copiar subprogramas utilizados frecuentemente hay un sistema todavía más sencillo: la **biblioteca**. ROBO Pro contiene una biblioteca de subprogramas terminados que puedes reutilizar fácilmente. Además, puedes crear tu propia biblioteca personal, en la que depositarás los subprogramas de tu creación que utilices con frecuencia.



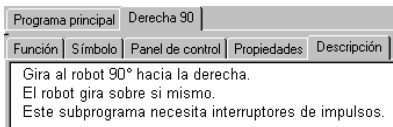
4.2.1 Utilizar la biblioteca

Para empezar, la **biblioteca** está dividida en dos grupos principales. En el grupo **Kits** encontrarás los subprogramas que puedes utilizar específicamente con los modelos de determinados kits de

construcción modular. En el grupo **General** encontrarás subprogramas que se pueden utilizar con todos los modelos posibles. Eso sí: la mayoría de los subprogramas del grupo **General** necesitan técnicas del nivel 3, que explicaremos en el próximo capítulo.

Para cada kit de la línea Computing, como por ejemplo el ROBO Mobile Set, hay un subgrupo independiente en el grupo **Kits**. Este subgrupo contiene a veces subdivisiones adicionales correspondientes a los modelos cuya construcción se explica en las instrucciones de montaje. Si seleccionas un kit o uno de los modelos, se mostrarán en la ventana de elementos los subprogramas terminados correspondientes a ese kit o modelo.

Si posas el ratón sobre un símbolo de subprograma, se muestra una breve descripción. Si insertas un subprograma en tu programa, puedes leer una descripción exacta seleccionando el subprograma en la barra de subprogramas y yendo después a la barra de función para hacer clic sobre **Descripción**:



Atención: Si insertas un subprograma desde la biblioteca, se insertarán en parte más subprogramas que son utilizados por él. Puedes eliminar todos los subprogramas seleccionando la función **Deshacer** del menú **Edición**.

4.2.2 Utilizar la propia biblioteca

Cuando lles un tiempo trabajando con ROBO Pro, seguramente tendrás ya una serie de subprogramas de tu creación que utilizarás a menudo. Para ahorrarte el tener que buscar y cargar cada vez el archivo correspondiente, puedes crear tu propia biblioteca personal de subprogramas, que funciona igual que la biblioteca predefinida. La biblioteca personal consta de uno o varios archivos ROBO Pro, guardados todos en una carpeta. Para cada archivo de esta carpeta se mostrará un grupo propio en la selección de grupos.

Puedes indicar en el menú **Archivo**, opción **Directorio de biblioteca personal**, en qué carpeta quieres guardar tu propia biblioteca. El directorio usual para la propia biblioteca es C:\Programas\ROBOPro\Biblioteca personal. Si tienes tu propio directorio de usuario en el ordenador, recomendamos que crees en él una carpeta propia y la utilices para este fin.

Consejo: Al principio puedes indicar en **Directorio de biblioteca personal** la carpeta en la que también guardarás tus programas de ROBO Pro. Así tendrás acceso rápido a todos los subprogramas de todos los archivos de tu carpeta de trabajo.

Organizar tu propia biblioteca

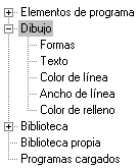
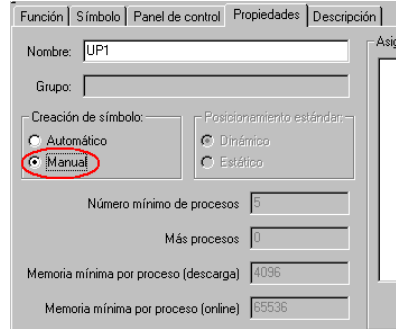
En ROBO Pro no hay ninguna función especial, para modificar una biblioteca. A pesar de ello es bastante fácil. Si quieres añadir subprogramas a un grupo de bibliotecas o eliminarlos de él, debes empezar por cargar el archivo correspondiente, que encontrarás en el directorio que hayas configurado como **Directorio de biblioteca personal**. Entonces podrás, por ejemplo, cargar un segundo archivo y arrastrar un subprograma del grupo **Programas cargados** al programa principal de la biblioteca. En una biblioteca el programa principal no es un programa de verdad, sino simplemente una colección de todos los subprogramas de la biblioteca. El propio programa

principal no se mostrará en la ventana de elementos. Naturalmente también puedes borrar o modificar subprogramas de una biblioteca.

Si has modificado y guardado un archivo de biblioteca, debes seleccionar en el menú **Archivo** la opción **Actualizar biblioteca personal**. Con ello se actualizará la lista de archivos en la ventana de grupos.

4.3 Editar símbolos de subprogramas

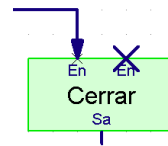
Como hemos visto en la sección anterior, ROBO Pro crea automáticamente símbolos verdes de subprogramas para tus subprogramas. Pero también puedes dibujar tus propios símbolos para indicar más claramente cuál es la función de los subprogramas de creación propia. Para hacerlo debes ir a la ventana de propiedades del subprograma y cambiar la opción de creación de símbolo de automático a manual. A continuación puedes pasar desde la barra de función de **Propiedades** a **Símbolo** y editar ahí el símbolo del subprograma. Encontrarás las funciones de dibujo en la ventana de grupos de elementos, en **Dibujo**.



En **Dibujo / Formas** encontrarás todos los elementos gráficos usuales (rectángulo, círculo, elipse, polígono etc). En **Dibujo / Texto** encontrarás objetos de texto en distintos tamaños de fuente. En los otros grupos encontrarás herramientas para cambiar el color y características similares de los elementos seleccionados. La utilización exacta de las funciones de dibujo se explicará en el capítulo 10 *Funciones de dibujo* página 546. Fíjate también en las funciones que aparecen en el menú principal, en **Dibujo**.

También puedes desplazar las conexiones del subprograma, pero no puedes borrarlas ni añadir nuevas conexiones. En el símbolo del subprograma hay siempre una conexión para cada entrada o salida del subprograma. Los elementos de conexión se crearán automáticamente aunque hayas marcado la opción de creación de símbolo manual.

Tan pronto como hayas salido de la ventana de edición de símbolo, todas las llamadas del subprograma del programa principal o de otros subprogramas se adaptarán en consecuencia. Ten en cuenta que si habías desplazado las conexiones de un subprograma, puede que se haya creado cierto caos en las llamadas, si las conexiones ya estaban hechas. Si es así, puede que los puntos finales de las líneas de unión ya no lleguen a la conexión correcta, lo que se indicará por medio de una cruz en el punto final de la línea o en la conexión (ver imagen). Por regla general es suficiente hacer clic con el botón izquierdo del ratón sobre cualquier parte de la línea de unión: la línea se trazará de nuevo automáticamente. Sin embargo, en los subprogramas con muchas uniones es posible que además tengas que editar la línea.



4.4 Tango

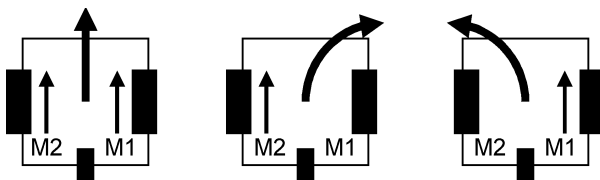
Hasta ahora te hemos presentado programitas muy sencillos, así que tal vez ardas ya en impaciencia por conocer nuevos elementos de programa y nuevas posibilidades. Pero antes de

ocuparnos de las variables y otras cuestiones más difíciles en el siguiente capítulo, queremos repasar en este punto todas las cosas que se pueden hacer con los elementos de programa de Nivel 2, que no son pocas. ¿Qué te parecería, por ejemplo, hacer que tu robot móvil baile el tango? A los que tengáis más de nerd^Σ os interesará saber que el tango se baila con un compás de 2/4. La figura básica comprende 8 pasos en 3 compases. Para el hombre, la figura es la siguiente:

- un paso lento hacia delante con el pie izquierdo (1/4)
- un paso lento hacia delante con el pie derecho (1/4)
- el paso de "la cunita", que dura 4/8. Para hacerlo casi no se mueven los pies: más bien se desplaza el peso del cuerpo del uno al otro. En primer lugar se desplaza el peso al pie izquierdo (atrás) en 1/8, después se desplaza en 1/8 al pie delantero (delante), y otra vez se vuelve a desplazar durante otro 1/8 al pie izquierdo (atrás). Para terminar el paso de la cunita se hace una pausa de 1/8.
- A continuación llegan tres pasos rápidos: En primer lugar se da un pequeño paso hacia atrás con el pie derecho, de manera que vuelve a posicionarse al lado del izquierdo. Después se da un paso lateral con el pie izquierdo. Finalmente se vuelve a colocar el pie derecho al lado del izquierdo. Cada uno de estos tres pasos duran 1/8 y se cierran con una pausa, también de 1/8.

Para las señoras, la figura es idéntica pero invertida, es decir, en vez de a la derecha se mueve a la izquierda, y en vez de hacia adelante se mueve hacia atrás. Se repite la figura completa hasta que se termine la música, se llegue a la pared de la habitación o aparezca el aburrimiento. En los dos últimos casos conviene consultar con un profesor de baile.

Pero en fin, volvamos al tema de la robótica. Tal vez tengas un kit fischertechnik de robots móviles. Los robots de estos kits suelen tener dos ruedas impulsoras con un motor independiente cada una. La dirección se controla igual que en los vehículos de cadena: cuando los dos motores giran en la misma dirección, el robot se desplaza en línea recta; cuando uno de los motores está parado, en cambio, el robot describe una curva.



Como es natural, estos robots también puedes desplazarse marcha atrás, tanto en línea recta como en curvo. Cuando los dos motores giran en direcciones opuestas, el robot rota sobre sí mismo. Cada una de las ruedas motrices está conectada con un modulador de impulsos. Por norma general el modulador emite 8 semiimpulsos por cada giro de la rueda, consistentes en pasos de abierto a cerrado o viceversa.

Vamos a tratar de traducir los pasos del tango a giros de las ruedas. El compás 1/4 debe coincidir con un giro de la rueda u 8 semiimpulsos, de manera que tenemos lo siguiente:

^Σ "Nerd" es una palabra inglesa que denota a una persona que no sale de casa sin la calculadora y que prefiere hablar con el profesor de matemáticas que salir por ahí de marcha.

- Rueda izquierda: 8 semiimpulsos hacia adelante (por norma general, motor M2 a la izquierda).
- Rueda derecha: 8 semiimpulsos hacia adelante (por norma general, motor M1 a la izquierda).

Ahora llega el paso de la cunita, pero, como todos sabemos, el robot no puede hacer lo de mover el cuerpo sin desplazar los "pies". También el paso lateral del tercer compás es bastante complicado para un robot. Así las cosas, lo que vamos a hacer es un pequeño giro hacia la izquierda en el segundo compás y un pequeño avance hacia adelante en el tercer compás para simular el paso lateral. Para el segundo compás tenemos:

- Rueda izquierda: 4 semiimpulsos hacia atrás (por norma general, motor M2 a la derecha).
- Rueda derecha: 4 semiimpulsos hacia adelante.
- Rueda izquierda: 4 semiimpulsos hacia atrás.

Tanto en el movimiento de izquierda hacia atrás como el de derecha hacia adelante, el robot gira hacia la izquierda. En el tercer compás hacemos lo siguiente:

- Rueda derecha: 4 semiimpulsos hacia atrás.
- Desplazamiento recto 4 semiimpulsos hacia atrás.
- Rueda derecha hacia atrás y rueda izquierda hacia adelante durante 4 semiimpulsos.

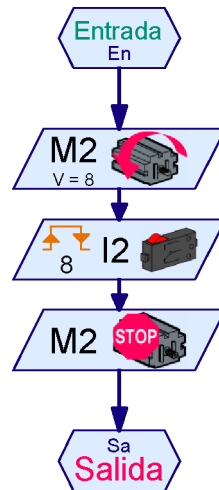
De este modo volvemos a girar el robot un poquito hacia la derecha, hacemos que avance en recto (que será en dirección adelante- izquierda) para simular el paso a la izquierda, y luego volvemos a girar el robot.

Trata de programar estas figuras de baile en ROBOPro. Lo mejor es comenzar creando subprogramas para cada uno de los pasos. Encontrarás ya diseñado el subprograma del primer paso, "rueda izquierda: 8 semiimpulsos hacia adelante". Por norma general el motor de accionamiento de la rueda derecha está conectado a la salida de interfase M2, y el correspondiente modulador de impulsos a la entrada de interfase I2, de manera que el giro a la izquierda se traduce en un avance.

Para el primer paso pones a girar el motor M2 hacia la izquierda (a máxima velocidad) y esperas 8 semiimpulsos en la salida I2. Esperar 8 semiimpulsos significa contar tanto los flancos 0 a 1 como los 1 a 0. Puedes seleccionar elementos en la ventana de propiedades del contador de impulsos. En cuanto hayan pasado los 8 impulsos, desconectas el motor M2 y se termina el subprograma. A este subprograma puedes darle el nombre de "Izquierda 1/4", por ejemplo.

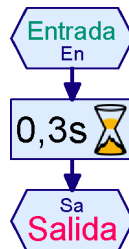
Para el resto de los pasos necesitarás diseñar los siguientes programas:

- Derecha 1/4 (igual que el Izquierda 1/4 pero con M1 e I1 en lugar de M2 e I2)
- Izquierda 1/8R (igual que el Izquierda 1/4, pero con 4 en lugar de 8 semiimpulsos y con desplazamiento hacia atrás, es decir, con el motor girando a la derecha)
- Derecha 1/8 (igual que el Derecha 1/4, pero con 4 en lugar de 8 semiimpulsos)



- Derecha 1/8 D (igual que el Derecha 1/8, pero hacia atrás, es decir, con el motor girando a la derecha)

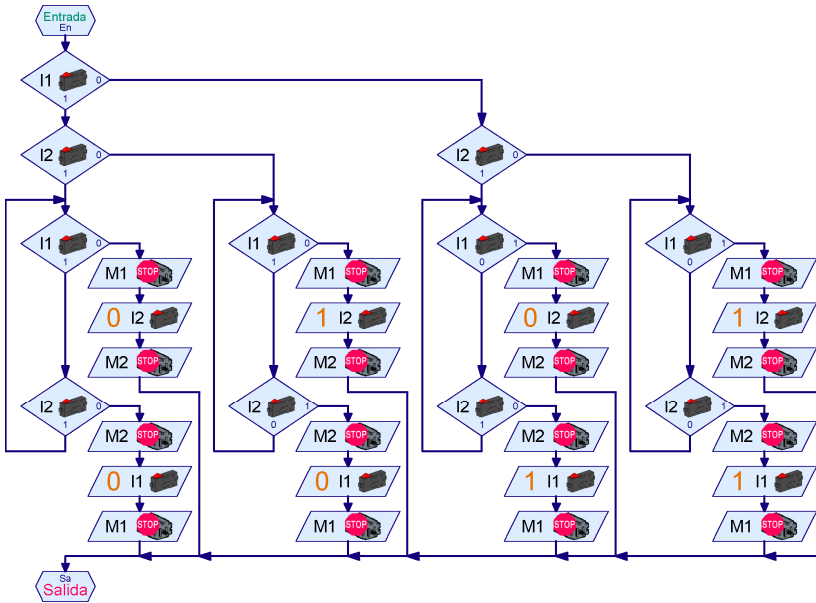
La pausa de 1/8 no puedes fijarla, claro está, con el contador de impulsos, porque en las pausas no se mueve ninguna de las ruedas. En lugar del contador nos valemos de un retardo. En los modelos estándar del ROBO Mobile Set, 4 semiimpulsos corresponden a unos 0,3 segundos, pero en función de la transmisión y del motor puede ser que en tu modelo en concreto varíen los valores. Para la pausa de 1/8 crea también un subprograma. Este subprograma contiene un único elemento de programa (además de la entrada y la salida), pero en realidad tú necesitas que la pausa sea doble. Si creas un subprograma para ello verás que puedes modificar fácilmente el período de pausa.



A lo mejor estás pensando que también deberíamos utilizar un retardo para los pasos, en lugar de optar por el contador de impulsos. Ahí no habría problema para adaptar el tiempo de pausa y del paso. La ventaja sería, no obstante, que los motores derecho e izquierdo no girarían a la misma velocidad, de manera que el robot no podría bailar unas figuras reproducibles. Al utilizar el contador de modulador de impulsos, en cambio, garantizamos que las dos ruedas se muevan siempre exactamente el mismo trecho, incluso cuando el acumulador esté bajo o cuando una de las ruedas sea un poco más pesada de mover que la otra.

Ahora sólo nos falta el subprograma para el paso 1/8 hacia adelante y el giro sobre sí mismo de 1/8. En realidad es lo mismo que con los otros pasos, sólo que en lugar de un motor encendemos los dos. La única cuestión es cómo hacerlo con los moduladores de impulsos. La opción de poner dos contadores de impulsos, uno detrás del otro, no es válida, porque el programa esperaría primero los 4 semiimpulsos de un motor y a continuación los 4 semiimpulsos del otro. Si esperaríamos a uno de los dos motores estaríamos más cerca de lo que queremos hacer, pero entonces tendríamos un problema si los dos motores no girasen a exactamente la misma velocidad. La mejor solución es encender los dos motores y a continuación esperar a que cambie uno de los moduladores de impulsos. En ese momento se detiene inmediatamente el motor cuyo modulador de impulsos haya cambiado y se espera a que cambie también el segundo modulador para parar también el segundo motor. Es bastante complicado, por desgracia, porque no se sabe si de entrada el modulador está abierto o cerrado. Como hay dos moduladores de impulsos, en total tenemos cuatro posibilidades. Por suerte para esta función ya tenemos un programa preconfigurado en la biblioteca. Crea el subprograma "Recto 1/8" y añade el subprograma "PasoSinc" que encontrarás en la biblioteca homónima de la carpeta "ROBO Mobile Set". Si no te ubicas, consulta *La biblioteca del subprograma* del capítulo 4.2, página 469.

Los que quieren saber cómo funciona todo encontrarán en el mismo sitio una breve explicación del subprograma "PasoSinc". Si te conformas con un simple vistazo del subprograma puedes saltarte tranquilamente los siguientes párrafos y el correspondiente ejercicio. Es perfectamente correcto utilizar un subprograma aunque no se entienda al detalle cómo funciona: lo importante es saber lo que hace.



El subprograma pregunta en primer lugar cuál es el estado de los moduladores de impulsos I1 e I2. Dependiendo de si I1 y I2 están en 0 o en 1, el subprograma salta a una de cuatro vías. La vía más a la izquierda es para el caso de que tanto I1 como I2 tengan el valor 1. En ese caso el subprograma debe esperar, por lógica, que o bien I1 o bien I2 tengan el valor 0. Esto se hace en el bucle con los dos elementos de bifurcación. Mientras I1 y I2 tengan el valor 1, el programa se ejecuta en círculo, pero en el momento que las dos entradas tengan valor 0 se detiene de inmediato el motor en cuestión. El subprograma espera entonces con un elemento de "Esperar en entrada" hasta que la segunda entrada sea 0, y detiene entonces el segundo motor. El bucle en el que espera a las dos entradas es necesario porque no se sabe cuál de los dos motores gira más rápido, y en consecuencia se ignora también cuál de los dos moduladores de impulsos cambia antes. Las otras 3 vías del programa funcionan exactamente igual, sólo que parten de un estado de salida diferente y esperan, como es natural, a su correspondiente estado final. En la segunda vía desde la izquierda, por ejemplo, partimos de I1=1 e I2=0, como puedes comprobar si vas siguiendo el camino por los dos primeros elementos de bifurcación. La segunda vía espera, por lo tanto, que I1 pase a 0 y que I2 pase a 1. En caso de que quieras diseñar el programa tú mismo, tienes que adaptarte con total precisión a los valores de partida del modulador de impulsos de cada vía, y esperar los valores que correspondan en cada caso.

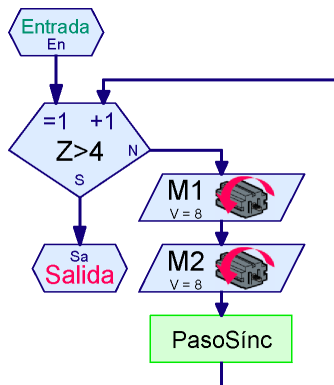


Si ya has echado un vistazo a la información sobre variables contenida en los siguientes capítulos puede intentar crear un subprograma que haga lo mismo, pero con variables. Resulta más fácil, porque al principio puedes almacenar el valor de los dos moduladores de impulsos en dos variables y a partir de ahí sólo necesitarás una única vía de programa, en la que puedes comparar el valor actual de las entradas con los valores de las variables.



Volvamos por fin al tango: el objetivo del subprograma PasoSínc era crear un subprograma de nombre "Recto 1/8" que avanzase en línea recta 4 medios pasos. Si se conectan los motores M1 y M2 y se ejecuta a continuación el subprograma PasoSínc, los motores darán medio paso y se detendrán. Esto significa que tiene que hacerlo 4 veces: para conseguirlo, lo mejor es optar por un elemento de bucle.

Si eres muy meticulouso puede que te ocupes de que los motores se detengan al final del subprograma de PasoSínc y vuelvan a funcionar a continuación. En el motor que gire más despacio habrá una brevisima pausa entre la desconexión y la conexión, necesaria para que las velocidades de los dos motores se ajusten mutuamente. A los motores no les supone ningún problema. De hecho, la propia interfase regula la velocidad de los motores a base de encenderlos y apagarlos constantemente. A este sistema se le llama MDI (modulación de duración de impulsos). En el motor más rápido, el encendido y apagado es tan rápido que el motor ni siquiera lo percibe. En el subprograma de PasoSínc puede prescindirse de apagar el segundo motor y hacer que los dos se apaguen en cuando termine el bucle. A la hora de programa suele haber distintas formas de llegar al mismo objetivo.



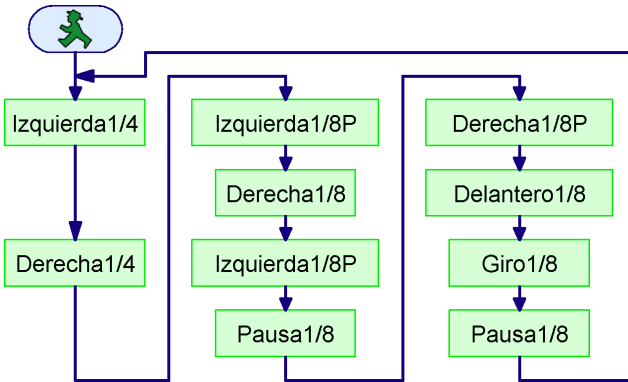
Comprueba ahora si verdaderamente el robot avanza mejor en línea recta con el subprograma PasoSínc que cuando simplemente enciendes los dos motores durante un número determinado de impulsos.

El último subprograma que necesitamos es el que haga que el robot gire sobre sí mismo durante 4 medios pasos. Lo interesante es que para ello puede utilizar el mismísimo subprograma "PasoSínc" que ya empleaste para "Recto 1/8". Es así porque el subprograma "PasoSínc" simplemente detiene los motores, con independencia del sentido de giro. Simplemente encientes el motor M1 en el bucle con giro a la derecha en lugar de a la izquierda, sin más complicación. Los moduladores de impulsos no tienen en cuenta el sentido de giro. Tanto da que los motores giren hacia la izquierda o hacia la derecha: los moduladores de impulsos pasan siempre de 0 a 1 y de 1 a 0. Para crear el subprograma "Giro 1/8", por tanto, simplemente tienes que copiar el programa "Recto 1/8" y modificar el sentido de giro.

Una vez que tienes todos los subprogramas, puedes proceder a montar el programa principal. Tampoco aquí encontrarás demasiadas complicaciones. En la siguiente página verás cómo podría ser el programa principal.



El programa principal que te presentamos hace que el robot baile el tango en un bucle sin fin. Intenta utilizar un bucle contador para ejecutar la secuencia de tango 5 veces seguidas en un bucle. Para ello copia el contenido del programa principal con Edición / Copiar y Edición / Pegar en un nuevo subprograma y añádele un subprograma de entrada y de salida. A continuación ejecuta este subprograma 5 veces en el bucle.

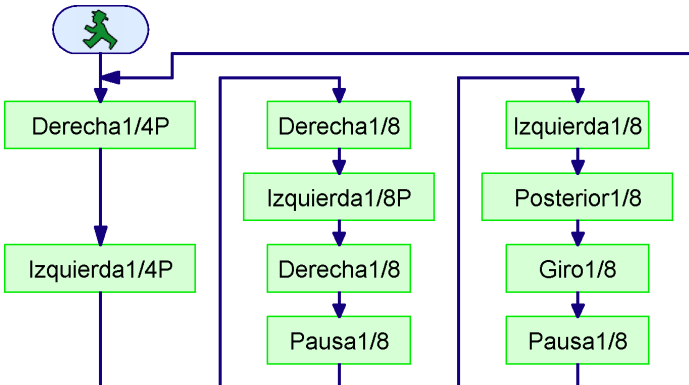


En la carpeta de instalación ROBOPro encontrarás el programa de tango ya completo:

Programas de ejemplo\Manual\TangoSolo.rpp

Si tienes un robot adecuado, prueba el programa que hayas diseñado tú o el que ya viene pre-diseñado.

Pero ten en cuenta una cosa: está muy bien y todo lo que quieras, pero para bailar hacen falta dos. Por suerte no es difícil diseñar un programa con los pasos de la mujer, para que un robot baile con el otro. Lo único que tienes que hacer es intercambiar izquierda/derecha y adelante/atrás. Primero carga el programa de los pasos de baile masculinos y grábalo con un nuevo nombre, por ejemplo **TangoSoloMujer.rpp**. Modifica a continuación el subprograma, por ejemplo **Izquierda 1/4** pasa a ser **Derecha 1/4 P**. Para ello tienes que pasar el M2 a M1 y cambiar el sentido de giro de izquierda a derecha. El nombre del subprograma puedes cambiarlo haciendo clic en la pestaña de **Propiedades** y poniendo el nuevo. El nombre también se cambia automáticamente en el punto del programa principal en el que se le llama al subprograma en cuestión.

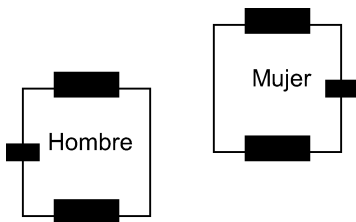


El subprograma **Giro 1/8** no cambia. ¿Se te ocurre por qué? Cambia en el **Giro 1/8** la izquierda por la derecha (M1 y M2 del subprograma) y el desplazamiento hacia adelante por un desplazamiento hacia atrás (motor izquierda/derecha) y compara el subprograma original con el modificado.



Si tienes dos robots móviles, carga en uno de ellos el programa **TangoSolo.rpp** y en el otro el programa **TangoSoloMujer.rpp**. Si sólo tienes un robot, tal vez podrías probar con alguien que

tenga el suyo. Al hacer la descarga debes indicar que el programa se va a iniciar con el botón de la interfase. Cuando los dos robots tengan su correspondiente programa, colócalos frente a frente tal y como se muestra en la ilustración, selecciona el programa con el botón PROG de la interfase (1x o varias largas repetidas) y enciende los dos robots al mismo tiempo con una breve presión sobre los botones PROG de ambas interfaces.

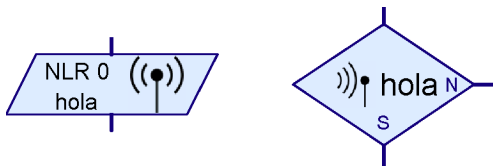


Si has encendido los dos robots más o menos al mismo tiempo, los dos bailarán el tango al compás durante un rato. Como los motores y los acumuladores no son idénticos, lo que ocurrirá será que los motores no girarán exactamente a la misma velocidad, de manera que antes o después los robots acabarán por descompasarse patentemente. Para saber cómo conseguir que los dos robots sigan el compás durante más tiempo, lee la siguiente sección.

4.5 Tango 2: comunicación a través de ROBO RF Data Link

Para no perder el compás, los dos robots tanguistas tienen que "hablar entre sí", cosa que harán con ayuda del ROBO RF Data Link. El ROBO RF Data Link consiste en dos módulos de radio. El módulo de radio de la interfase se encuentra montado directamente como pletina en la ROBO Interface; el módulo de radio del PC de carcasa roja se conecta al PC por el puerto USB. Si tienes un RF Data Link seguramente lo hayas estado usando hasta ahora para radiocontrolar tus robots, es decir, para manejarlos sin necesidad de cables. Pero el RF Data Link tiene muchas más posibilidades: gracias a él dos robots pueden intercambiarse mensajes y comunicarse de este modo entre sí.

En el Nivel 2 los **elementos de programa** contienen un subgrupo de **Enviar, Recibir** conformado por dos elementos. El elemento de la izquierda es el emisor; el de la derecha, el receptor.



El elemento de envío de la imagen

envía el mensaje **hola** a la ROBO Interface con el Número de llamada por radio 1, que se abrevia como FRN 1. El número de llamada por radio es una especie de número de teléfono que indica a qué interfase se envía el mensaje. Encontrarás más información sobre este tema en la siguiente sección, 4.5.1 *Configuración de la frecuencia y el número de llamada por radio*, que empieza en la página 482.

El receptor de la derecha de la imagen funciona como una bifurcación de programa: al recibir el mensaje de **hola**, el elemento se pasa a la bifurcación del sí, **S**; en caso contrario pasa a la salida no, **N**.

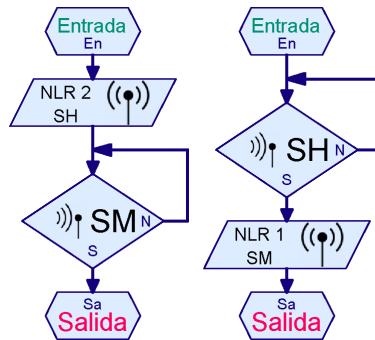
Si tomamos en el elemento de emisión, se llama desde un programa a la interfase con el número de llamada de radio (número de teléfono) 2. A continuación envía el mensaje **hola** a la interfase con el número de llamada por radio 1, porque es lo que se indica como destinatario. La interfase

que posee el número de llamada por radio 1 se da cuenta de que ha recibido el mensaje **hola**. La próxima vez que un elemento de recepción pregunte a la interfase si ha recibido un mensaje de **Hola**, la respuesta será que sí; a partir de ahí contestará que no hasta que vuelva a recibir otro. El receptor no es capaz de distinguir qué interfase le ha enviado el mensaje†. Para permitirle hacer esa distinción tienes que enviarle mensajes diferentes.

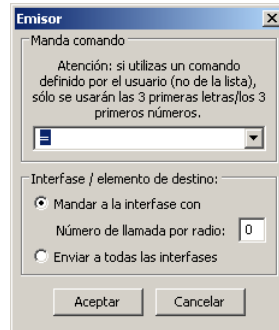
El mensaje es la palabra que tú elijas (por ejemplo, **hola**). Lo que ocurre es que por razones técnicas sólo se tienen en cuenta las tres primeras cifras o letras del mensaje. Si quieres puedes poner más de tres letras, pero "hola", "holita" y "Holanda", por ejemplo, darán los tres el mismo mensaje, porque empiezan todos por "hol". Las mayúsculas y minúsculas y los caracteres especiales (espacio en blanco, signos de admiración e interrogación y similares) tampoco se distinguen. Por ejemplo, XY! y XY? se entenderán como el mismo mensaje. Los números sí que se distinguen, de forma que XY1 no se confunde con XY2.

Hola
=
Holita

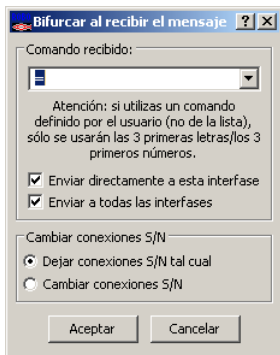
La sincronización de los dos robots tanguistas por medio del elemento de emisión y recepción no es excesivamente complicada. Al principio del ciclo de pasos uno de los robots envía un mensaje de "¿Bailamos?" y el otro robot contesta "Sí, bailemos": enseguida se ponen a ello. Como los mensajes no pueden ser tan largos, vamos a usar la abreviatura SH para "Sincronización hombre" y SM para "Sincronización mujer". A la derecha se muestran dos subprogramas de sincronización. El de la izquierda lo ejecuta el robot "hombre" con número de llamada por radio 1, que a continuación envía un mensaje de SH a la "mujer", que es el robot con número de llamada por radio 2. Entonces el "hombre" espera recibir a su vez un mensaje de la "mujer".



A la derecha se muestra la ventana de propiedades del elemento de emisión. Con el **comando de enviar** puedes elegir un comando (un mensaje) de la lista o indicar el tuyo propio. En **Interfase de destino** puede seleccionar si el comando se manda a una interfase con un número de llamada por radio en concreto o a todas las interfasas.



† A partir del Nivel 4 existen los grupos de recepción para estos casos.



A la izquierda se muestra la ventana de propiedades del elemento de recepción. Al igual que con el emisor, puedes seleccionar un comando (un mensaje). A continuación debes escoger si el receptor ha de reaccionar solamente a los comandos que se envíen directamente a la interfase (es decir, con un número de llamada por radio específico) o a todas las órdenes que se envíen a todas las interfases. Puedes escoger las dos opciones al mismo tiempo. Finalmente puedes intercambiar las conexiones de sí y no en cada elemento de bifurcación.

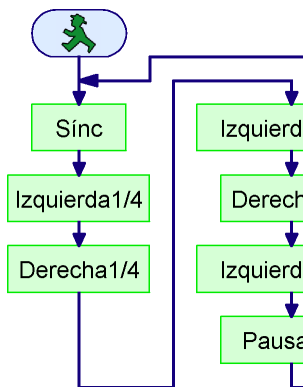
Hasta ahora hemos hablado de "mensajes": En la ventana de propiedades del emisor y del receptor (y más adelante, en el Nivel 3) utilizaremos el término "comando". Desde el punto de vista de la transmisión de datos es lo mismo. Que un mensaje sea o no un comando depende de la interpretación y no de la forma de transmisión. En el Nivel 3 tendrás muchas oportunidades de trabajar con mensajes que son comandos para controlar un motor, por ejemplo, o variables. Por esa razón en ROBOPro usamos siempre y en general el término "comando".

Añade el programa **TangoSolo.rpp** y **TangoSoloMujer.rpp** sendos subprogramas de los presentados en la página anterior. Da al subprograma el nombre de Sinc. Naturalmente también puedes diseñar el subprograma de manera que sea la "mujer" quien "saque a bailar" al "hombre". El subprograma se llama en el programa principal como se muestra en la imagen de la derecha al inicio del ciclo de pasos. Encontrarás un programa ya confeccionado en la carpeta de instalación de ROBOPro.

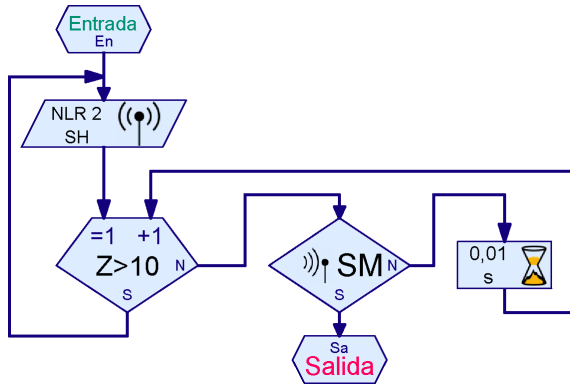
Programas de ejemplo \ Manual\TangoSincHombre.rpp

Programas de ejemplo \ Manual\TangoSincMujer.rpp

Carga cada programa en el correspondiente robot e iniciales. Comprobarás que los robots sólo se ponen a bailar si se inicia primero el programa del robot "mujer". Es así porque el "hombre" empieza enviando un mensaje de **SH**, que la "mujer" espera. Si el "hombre" se inicia antes, el mensaje se mandará al vacío y la "mujer" esperará en vano para siempre. Si por el contrario la "mujer" ya está preparada y esperando el mensaje de **SH** cuando se inicia el "hombre", no habrá problema alguno. Todo esto resulta bastante práctico, como estarás pensando, especialmente cuando uno ya no se acuerda de cuál robot es cuál. ¿Qué se puede hacer para remediarlo?



Muy sencillo: decirle al "hombre" que repita la invitación para bailar tantas veces como sea necesario hasta que la "mujer" responda. Cada vez que el "hombre" envía el mensaje, espera un poquito para comprobar si la "mujer" le manda respuesta. En el ejemplo de la derecha el "hombre" manda el mensaje **SH** y espera en un bucle de 10 veces 0,01 segundos (un total de 1/10 segundos) para ver si la "mujer" le contesta. Si la "mujer" no responde en ese tiempo, el "hombre" vuelve a enviar su mensaje de **SH**.



Quizás te estás preguntando por qué el "hombre" no puede simplemente mandar la invitación **SH** y comprobar inmediatamente si ha recibido la respuesta de **SM**. No es posible porque la transmisión del mensaje del "hombre" a la "mujer" tarda entre 1/100 y 2/100 segundos, a lo que hay que sumar otro tanto: el tiempo que tarda en llegar la respuesta. Incluso si el programa de la "mujer" ya está en ejecución, nos encontramos, entonces, con que desde el envío hasta que llega la respuesta pueden transcurrir 4/100 segundos. En este período de tiempo el "hombre" podría mandar una gran cantidad de mensajes de **SH** que habría que transmitir a la "mujer". Para la primera sincronización esto no tiene importancia, pero lo cierto es que todos esos mensajes superfluos de **SH** se quedarían almacenados en el programa de mujer del elemento de recepción. Cuando se empezase el siguiente ciclo, la "mujer" ya habría recibido un mensaje de **SH** antes de que el "hombre" hubiese mandado nada más. Si así fuese, la "mujer" ya no esperaría recibir más del "hombre". Por esta razón es imprescindible que el "hombre" espere un tiempo suficiente antes de repetir la invitación.



Naturalmente el "hombre" también podría seguir esperando más de 1/10 segundos antes de comprobar si hay respuesta o no. La ventaja de una espera en bucle de 10 veces 1/100 segundos es que el "hombre" puede proseguir con el programa casi inmediatamente si se da el caso de que efectivamente recibe la respuesta de la "mujer".

Comprueba que el programa funciona mejor si en **TangoSyncHombre.rpp** modificas el subprograma **Sync** tal y como te explicamos antes. Hacemos que los robots inicien su programa en cuando el otro se inicia, independientemente de cuál sea el que se encendió primero.

Si dejas que los robots bailen hasta que se vayan agotando los acumuladores, la sincronización por ciclo de pasos pierde su efectividad. Los robots se sincronizan al principio de cada ciclo de pasos, pero cuando uno de los acumuladores llega a su límite se descompasan evidentemente. En ese caso es mejor diseñar una sincronización adicional tras cada paso. Como de todas formas está claro que los dos programas están funcionando, puedes prescindir de la repetición. Para que la sincronización inicial y la sincronización de pasos no se interfieran, tiene que utilizar dos subprogramas distintos, **Sync1** y **Sync2**, que manden mensajes diferentes (por ejemplo **SH1**, **SM1** por un lado y **SH2**, **SM2** por el otro). Encontrarás los programas ya confeccionados en la carpeta de instalación de ROBOPro, en



Programas de ejemplo ManualTangoHombre.rpp

Programas de ejemplo ManualTangoMujer.rpp

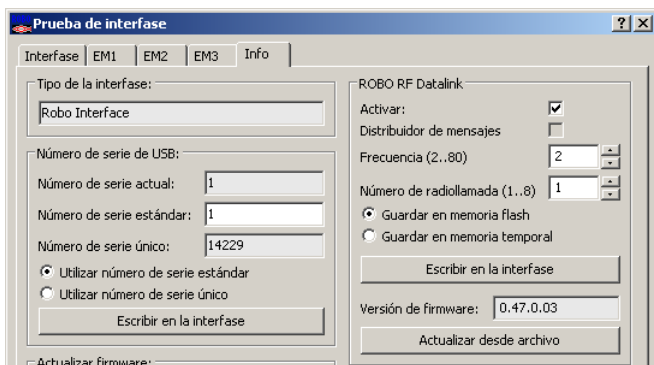
En la misma carpeta verás el programa **TangoMonitorMensajes.rpp**. Si tienes una tercera interfase con módulo de ROBO RF Data Link, puedes ejecutar este programa en modo online mientras bailan los dos robots. El programa muestra en la pantalla los mensajes que se han enviado. El programa utiliza elementos de Nivel 3, así que en este momento no necesitas entender cómo funciona.

4.5.1 Configuración de la frecuencia y el número de llamada por radio



Prueba de interfaces

Cada ROBO Interface tiene asignado un número de llamada por radio (de 1 a 8) y una frecuencia; puedes configurar estos datos en la ventana de **info** de la prueba de interfase. En la imagen de la página siguiente verás la configuración del ROBO RF Data Link, en la parte superior izquierda. Todos los módulos de RF Data Link que hayan de intercambiarse mensajes deben compartir la misma **frecuencia**. La frecuencia se indica con un número entre 2 y 80. Puedes modificarla si es que en un mismo espacio (por ejemplo en un aula o en un concurso) tienes varios robots que deben comunicarse entre sí pero formando grupos distintos. Cada grupo tiene su propia frecuencia, común a todos los robots pero diferente de la frecuencia de otros grupos. También puedes modificar la frecuencia si el RF Data Link no funciona bien en la que estás usando. Muchos sistemas de radio (por ejemplo las redes inalámbricas de ordenadores) utilizan el mismo rango de frecuencia (2,4 GHz) que el ROBO RF Data Link. Si el RF Data Link está interferido por otros sistemas de radiocomunicación, un simple cambio de frecuencia puede resolver el problema. Eso sí: ten en cuenta que entonces deberás modificar la frecuencia de todos los módulos de RF Data Link y el módulo de radio del PC, puesto que todos los dispositivos del grupo deben utilizar obligatoriamente la misma frecuencia.



Todas las ROBO Interface con RF Data Link configuradas en la misma frecuencia deben tener asignado un **número de llamada por radio** entre 1 y 8. El número 0 está reservado para el módulo del PC del RF Data Link, la "caja roja". Dicho de otro modo, pueden comunicarse un máximo de 8 interfaces con módulo de radio integrado y un módulo de PC RF Data Link. El número de llamada por radio es, por así decirlo, el número de teléfono de cada interfase en la red de radiocomunicación. A la hora de asignar los números de 1 a 8 a las 8 interfaces puedes proceder como prefieras.

La marca de **activar** está puesta casi siempre, pero si quieres puedes desactivar el módulo de radio de una interfase cuando no la vas a utilizar y quieres ahorrar corriente sin necesidad de desmontarlo del sistema que has compuesto.

Cuando hayas completado todos los ajustes de configuración puedes guardarlos en la interfase con el botón de **Escribir en la interfase**. Por norma general debes **guardar** la configuración **en la memoria flash**, en la cual se quedarán aunque apagues la interfase. Si tan sólo quieres hacer una breve prueba, puedes **guardar** la configuración **en la memoria temporal**.

De la **versión de firmware**, es decir, la versión del programa interno de control del RF Data Link, puedes despreocuparte. ROBOPro te avisará automáticamente de que has de actualizar el firmware cuando llegue el momento.

Para permitir la comunicación entres los dos robots tanguistas empezamos por conectar uno de los dos robots al puerto USB del PC y abrimos la ventana de prueba de la interfase. Puede ser que antes tengas que apretar el botón COM-USB y seleccionar la interfase en cuestión. Cambia de la ventana de prueba de interfase a la pestaña de info y configura en ella la **frecuencia 2** y el **número de llamada por radio 1**; guarda la configuración apretando el botón de **escribir en la interfase**. Cierra a continuación la ventana de prueba de interfase, conecta el otro robot al puerto USB y configúralo con **frecuencia 2** y **número de llamada por radio 2**.



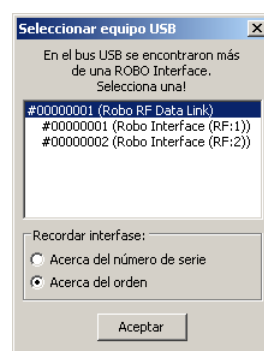
4.5.2 Requisitos para la comunicación por radio ation

Incluso en el caso de que dos interfases puedan intercambiar mensajes por módulos RF Data Link (como ocurre con los dos robots tanguistas), tendremos que utilizar el módulo rojo de radio del PC. Este módulo actúa como "centralita" de todos los demás, razón por la cual debe estar conectado al puerto USB de un PC. El PC tiene que estar encendido y no puede pasar a modo de ahorro de energía ni de hibernación o suspensión, porque de lo contrario el módulo de radio PC se quedaría sin alimentación. El módulo de radio del PC tiene que estar configurado con la misma frecuencia que los módulos de radio de las interfases. El número de llamada por radio del módulo del PC está preconfigurado como 0 y no puede modificarse.

Para configurar la frecuencia del módulo de radio del PC, conéctalo al puerto USB: podrás modificar la frecuencia en la pestaña de info de la ventana de prueba de interfase, igual que con los módulos de radio de las interfases. En caso de que el módulo de radio del PC no sea capaz de comunicar con ninguna interfase con módulo de radio integrado (porque no coincida la frecuencia, por ejemplo), te saldrá un mensaje de error al abrir la ventana de prueba de interfase. Este mensaje de error indica únicamente que no se ha encontrado ninguna interfase y que por lo tanto no hay disponibles entradas ni salidas, lo que no impide, no obstante, que puedas hacer igualmente los ajustes de configuración en la ventana de info.

4.5.3 Selección de interfases con el botón de puertos (COM/USB)

Hasta ahora seguramente has trabajado con una sola ROBO Interface. Desde el momento en que conectas al PC varias ROBO Interfaces o un módulo de radio de PC que puede comunicar por radio con más de una ROBO Interface, surge la pregunta de a cuál de las interfases debe conectarse cuando se inicia un programa en modo online, cuando se procede a descargar un programa o cuando se abre la ventana de prueba de interfase. Si presionas el botón de opciones de puerto (botón COM-USB), aparecen las posibilidades de selección. Si seleccionas USB y ROBOPro encuentra más de una interfase en el bus USB o en la red de radio, se muestra la ventana de selección de la derecha. En este ejemplo el módulo de radio del PC de un RF Data Link está conectado al puerto USB del PC. El



módulo de radio del PC ha encontrado por radio dos ROBO Interfaces con RF Data Link, correspondientes a los números 1 y 2. En esta ventana puedes seleccionar cuál de las dos interfases pueden utilizarse para operaciones futuras. Por norma general en este punto seleccionas una de las interfases y no el ROBO RF Data Link.



Si seleccionas el RF Data Link, como en la imagen, ROBOPro se conecta a la interfase con el número de llamada por radio más bajo (en el ejemplo, el 1). Hay, no obstante, una diferencia muy importante: si se selecciona el RF Data Link en sí, la configuración que aparece en la pestaña info de la venta de prueba de interfase pasa al propio módulo de radio del PC y no al módulo de radio de la interfase conectada por radio. Esto quiere decir que puedes, por ejemplo, modificar la frecuencia de los dos módulos de radio de la interfase y del PC por radio sin tener que conectar las interfases al PC con un cable USB. Para ello seleccionas primero con el botón COM/USB una de las interfases, cambias la frecuencia desde la ventana de prueba de interfase y vuelves a cerrarla. Si vuelves a presionar el botón COM/USB, la interfase a la que has cambiado la frecuencia ya no aparece en la lista, porque el módulo de radio del PC ha dejado de poder comunicarse con ella. Selecciona en este punto la segunda interfase y modifícale la frecuencia. Si vuelves a apretar el botón COM/USB, lo más probable es que ya no se muestre ningún menú de selección, porque el módulo de radio del PC ha pasado a ser la única unidad con la que se puede comunicar. En caso de que hayas conectado otras interfases directamente al USB y aparezca el menú de selección, selecciona el módulo de radio del PC. Al abrir la prueba de interfase aparecerá un mensaje de error avisando de que no se ha encontrado ninguna interfase. No te preocupes: tú querías cambiar la configuración del Data Link, nada más. Si modificas la frecuencia del módulo de radio del PC y vuelves a apretar el botón COM/USB, se mostrarán de nuevo el módulo de radio del PC y las dos interfases, porque ahora tienen las tres la misma frecuencia.



¿Por qué debes cambiar primero las frecuencias de las interfases y sólo después la del módulo del PC? Prueba a ver qué pasa si lo haces al revés.

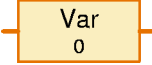
Puede ocurrir, por cierto, que las modificaciones tarden un par de segundos en surtir efecto. Si el menú de selección del botón COM/USB no se corresponde con la configuración actual, simplemente apriétalo de nuevo. En caso de que dejes de poder comunicar por radio con una interfase, lo mejor es que la conectes directamente al puerto USB del PC y repases la configuración del RF Data Link de la interfase.

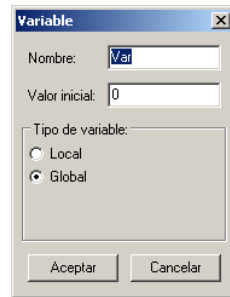
5 Nivel 3: Variables, paneles de control & cía

¡No te olvides de cambiar la ROBO Pro a **Nivel 3** (o superior) en el menú **Nivel!**

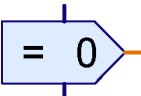
Imagínate que dentro de un museo descubres, en una sala lateral hasta ahora sin investigar, una máquina fascinante, cuya construcción quieres imitar a toda costa con los materiales fischertechnik. Pero estudiando la máquina pierdes la noción del tiempo y no te das cuenta que todos los otros visitantes están ya saliendo del museo. Hasta que ha pasado la hora de cierre del museo, no acabas de estudiar la máquina lo suficiente para poder reconstruirla. Pero lamentablemente debes pasar una noche desagradable y solitaria en el museo antes de poder ponerte manos a la obra. Para que esto no te suceda otra vez, ofreces al director del museo programar un contador de visitantes que cuente todas las personas que entren y salgan y que encienda un piloto de luz roja mientras quede algún visitante dentro del museo. ¿Pero cómo hacerlo? ¿Cómo puedes contar algo con ROBO Pro? La respuesta es: con **variables**.

5.1 Variables e instrucciones

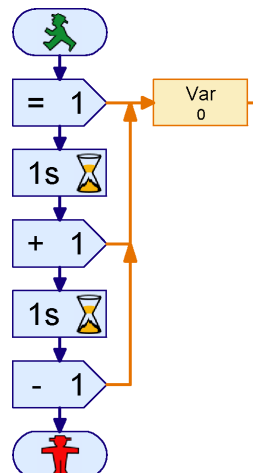
 Una variable es un elemento que puede retener un número. En la ventana de propiedades de las variables estableces un **nombre**, que debe indicar el tipo de valor numérico que se guardará en la variable. En **Valor inicial** puedes indicar qué número debe guardarse en la variable al inicio del programa. La opción **tipo de variable** se explicará en la sección 8.4.2 *Variable local* página 523



Puedes cambiar el número guardado enviando instrucciones a la variable. Una variable entiende 3 instrucciones distintas: =, + y -. La instrucción = sustituye el número guardado por un nuevo número. Las instrucciones + y - suman o restan algo del número guardado.

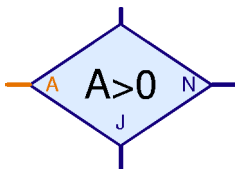
 Las instrucciones se envían a la variable con un **elemento de comando**. El elemento de comando tiene, como la mayoría de los otros elementos de programa, una entrada de programa azul por arriba y una salida de programa azul por abajo. Pero a la derecha tiene algo totalmente nuevo: una conexión de color **naranja**. Se trata de una salida de comando. Siempre que se ejecute el elemento de comando, por esta salida se envía una instrucción a todos los elementos conectados. La variable tiene en la parte izquierda la entrada de comando correspondiente. Si unes la salida de comando con la entrada de comando, la ROBO Pro dibujará, en vez de las típicas uniones en azul, una línea naranja. Mediante las líneas naranjas los elementos de programa pueden enviar instrucciones o mensajes y así intercambiar información.

El programa de la derecha envía primero a la variable **Var** una instrucción = 1. Una instrucción consta normalmente de una instrucción propiamente dicha (como =) y de un valor (como 1). La



instrucción = 1 pone la variable a 1. Después de un segundo, el programa envía a la variable la instrucción + 1. La variable entonces sumará 1 a su valor actual y adquirirá el valor 2. Después de otro segundo, el programa enviará una instrucción - 1, con lo cual la variable tendrá de nuevo el valor 1.

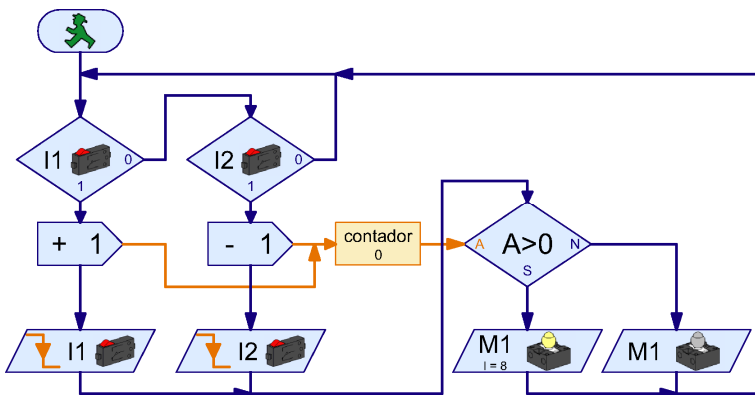
Intenta dibujar este sencillo programa en ROBO Pro. Encontrarás los elementos de comando en el grupo **Instrucciones**, la variable en el grupo **Variable, Temporizador**.... Si ejecutas el programa en modo online, verás cómo va cambiando el valor de las variables.



Quizás ahora pensarás: pues muy bien, ahora puedo ver el valor de las variables. ¿Pero qué hago con ello? Muy sencillo: la variable tiene a la derecha una conexión naranja mediante la cual envía mensajes con su valor actual a todos los elementos conectados. ROBO Pro contiene una serie de elementos que cuentan con una entrada naranja a la izquierda, entrada ésta que podrás unir con la salida de la variable. Así encontrarás, por ejemplo, en el grupo **Bifurcación, Esperar**... un elemento de bifurcación Sí / No que no

consulta directamente una entrada, sino que puede consultar cualquier valor, entre otros el valor de una variable.

Con ello puede programarse el contador de visitantes para el museo de la siguiente forma:

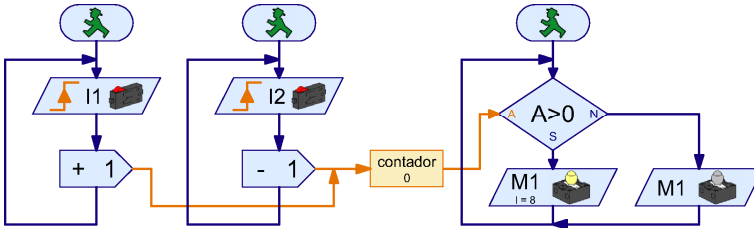


El torno de la entrada acciona el pulsador de I1 y el torno de la salida acciona el pulsador de I2. En cuanto se pulsa I1, el programa envía a la variable **Contador** una instrucción + 1. A continuación el programa espera hasta que se suelte el pulsador de I1. Con el pulsador de salida de I2 ocurre lo mismo, sólo que aquí se envía a la variable **Contador** una instrucción - 1. Cada vez que se modifique el contador, se controla el valor indicado. Si la variable **Contador** tiene un valor > 0, se iluminará el piloto en rojo de M1; en caso contrario, se apagará.

Copia ahora el programa anterior y pruébalo. En cuanto pulses el pulsador de I1 y lo sueltes de nuevo, se iluminará el piloto de M1. Si accionas el pulsador de I2, se apagará de nuevo. Si accionas I1 varias veces, debes accionar I2 el mismo número de veces para que la lámpara piloto se desconecte de nuevo. Intenta también averiguar qué sucede si primero vienen 5 visitantes, después se van 2 y más adelante vienen otros 3. ¿Cuántas veces debes accionar ahora el pulsador de I2 para que se apague el piloto?

5.2 Variables y varios procesos

Quizás habrás notado durante la prueba del contador de visitantes que surgen problemas si se activan los pulsadores I1 y I2 al mismo tiempo. Mientras uno de ellos está pulsado, el programa no puede reaccionar al otro. Pero dado que puede haber visitantes entrando y saliendo al mismo tiempo por el turno correspondiente, esto llevará a errores de conteo. Pues bien: puedes eliminar este error utilizando varios procesos paralelos. Hasta ahora todos los programas tenían siempre un solo elemento de inicio, pero también se pueden utilizar varios a la vez. De este modo, todas las operaciones que tengan su propio elemento de inicio se procesarán en paralelo. De ahí que los expertos hablen de **procesos de ejecución paralela**. Esta técnica te permite modificar el programa contador de visitantes de la siguiente manera:



Ahora se utilizarán procesos independientes para I1 e I2. Si se activa el pulsador de I1, el proceso de I2 no se verá afectado y podrá continuar controlando el pulsador correspondiente. Para consultar los valores del contador y para conectar y desconectar el piloto se utilizará igualmente un proceso específico.



Como puedes ver, no es difícil servirse de una variable para varios procesos. Por un lado, puedes enviar instrucciones a una variable desde varios procesos; por otro, puedes utilizar el valor de una variable en varios procesos. Por eso las variables también sirven para intercambiar información entre procesos.

El director del museo está tan entusiasmado con tu fantástico contador de visitantes que enseguida te pide que le soluciones otro problema: el museo ha abierto una nueva exposición pero, como todos los visitantes quieren contemplar esta novedad, hay tanto gentío que es imposible ver nada. Por eso el director quiere limitar a 10 el número de visitantes simultáneos de la exposición. El director ha colocado un tornillo en cada entrada y cada salida de la exposición. El tornillo de la entrada puede bloquearse electrónicamente. Ahora lo único que necesita es un experto programador. ¡Y ese eres tú!



Intenta desarrollar con ROBO Pro el programa descrito. Funciona básicamente como el contador de visitantes. Simularás el bloqueo electrónico de la entrada con una lámpara roja conectada a M1 que se enciende en el momento en que haya 10 personas visitando la exposición.

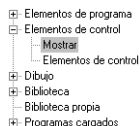
5.3 Paneles de control

Una vez que has solucionado el problema de la exposición, el director de museo tiene otro encargo para ti: quiere saber cuántas personas han visitado su museo en un día. Crear un programa que cuente a los visitantes ya no supone ninguna dificultad para ti. ¿Pero cómo puedes mostrar ese valor numérico? Naturalmente podrías ejecutar el programa en modo online y enseñarle al director del museo en qué variable tiene que buscar el valor. Pero para una persona no muy ducha en informática como es el director del museo esto resulta bastante complicado. ¡Tiene que haber un sistema más fácil!

Para estos casos ROBO Pro pone a tu disposición sus paneles de control. Un panel de control es una página individual en la que puedes dibujar mensajes y botones de instrucción. Carga tu programa de contador de visitantes y pasa en la barra de función a **Panel de control**.

Función Símbolo Panel de control Propiedades Descripción

El panel de control es, de primeras, una superficie vacía de color gris. En esta superficie colocas los mensajes y los elementos de mando, que encontrarás en la ventana de grupos de elementos, en la opción **Elementos de control**. Entre los elementos de mando encontrarás botones pulsadores, reguladores etc. Entre los mensajes encontrarás mensajes de texto, lámparas de alarma e indicaciones con punteros giratorios.



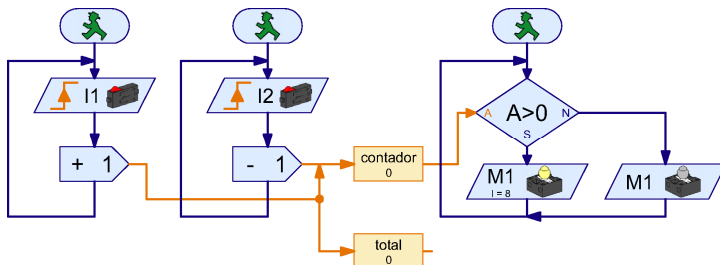
Atención: Un panel de control es parte de un subprograma. Si tienes subprogramas, debes tener cuidado de crear el panel de control en **Programa principal** y no en otro subprograma. Más adelante, cuando seas experto, podrás crear varios paneles de control.

Si has dibujado un panel de control y de repente este panel desaparece, seguramente es porque has seleccionado un subprograma en la barra de subprogramas. Pasa de nuevo a **Programa principal** y seguramente verás de nuevo tu panel de control.

Var: 0

Para el contador de visitantes tomas un **mensaje de texto** (da igual el color) de la ventana de elementos **Elementos de control / Mensajes** y la colocas en el panel de control. En este mensaje debe indicarse ahora el número de visitantes del museo.

Pero primero tienes que añadir una segunda variable a tu programa, que contará el número de visitantes en la entrada sin restar los visitantes en la salida. Para ello vuelves a pasar en la barra de función a **Función** y añades la variable **Total** de la siguiente manera:



Como ves, un elemento de comando puede usarse también para enviar una instrucción simultáneamente a dos variables. La variable **Total** no contiene las instrucciones - 1, dado que las líneas naranjas transmiten las instrucciones solamente en el sentido de la flecha. Las instrucciones + 1, en cambio, se transmiten a ambas variables. Pero aquí solamente lo indicamos como ejemplo. Por regla general es más sencillo y más claro utilizar un segundo elemento de comando.

Consejo: Cuando se bifurcan las líneas naranjas, a menudo es más práctico dibujar las líneas de fin a principio. Así, si en el ejemplo superior quieres dibujar la línea hacia la variable **Total**, haz clic primero sobre la entrada de la variable **Total** y después ve trazando la línea hacia atrás hasta llegar al punto de bifurcación. Si por el contrario quieres empezar una línea

naranja en otra línea naranja ya existente, debes hacer doble clic con el botón izquierdo del ratón sobre el punto en el que debe empezar la nueva línea.



Bien: ya tienes un mensaje de texto en el panel de control y una variable que quieres mostrar en ese mensaje. ¿Cómo se unen ahora estos dos elementos? Dado que el mensaje de texto y la variable están en dos páginas distintas, no puedes unirlos con una línea. Por eso existe un elemento especial que transmite el valor que debe mostrarse en el panel de control al mensaje correspondiente. El elemento que ves arriba, **Salida del panel de control**, se halla al final del grupo **Entradas, Salidas**. Inserta una de esas salidas de panel de control en tu programa, al lado de la variable **Total**, y une la conexión derecha de la variable con la conexión de la **salida del panel de control**.

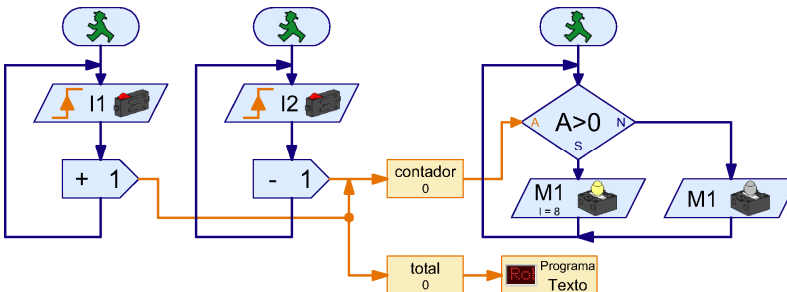
Dado que por regla general tendrás más de un mensaje en el mismo panel de control, todavía te resta indicarle a la salida del panel de control a cuál de los mensajes debe enviar los valores de la variable. Puedes hacerlo fácilmente desde la ventana de propiedades del elemento. Si haces clic con el botón derecho del ratón sobre la salida del panel de control, verás un menú de selección que lista todos los mensajes que han sido insertados en el panel de control. Como cada subprograma puede tener su propio panel de control, los paneles están clasificados por subprogramas. En nuestro ejemplo no hay ningún subprograma, sino solamente el programa principal. Debajo del programa principal hay un mensaje llamado **Texto**. Elígelo y haz clic en OK.



En cuanto hayas unido la salida del panel de control con un mensaje, el símbolo y la leyenda cambiarán como corresponde. La salida del panel de control que utilizamos establece una unión con el mensaje de texto de nombre **Texto** del (sub)programa **PRINCIPAL**.



Cuando hayas insertado la salida del panel de control y la hayas unido con el mensaje de texto, tu programa tendrá este aspecto:



¡Pruébalo enseguida! Al iniciar el programa en modo online, el mensaje mostrará en el panel de control el número de visitantes que han pasado por el torno de la entrada.

Nota: Si quieres utilizar más de un mensaje en un panel de control, es importante dar un nombre distinto a cada uno para poder distinguirlos en el enlace con el programa. Para ello haz clic con el botón derecho del ratón sobre el mensaje que tengas en el panel de control. Ahí puedes introducir un nombre (en **ID / Nombre**). Si después enlazas una salida del panel de control con el mensaje, aparecerá este nombre en la ventana de selección de la salida del panel de control. Dado que por ahora solamente tenemos un mensaje, el nombre no es importante y mantendremos el nombre de **Texto**.

El programa aún no es perfecto: falta todavía un interruptor que ponga a cero el contador. Pero para generarlo no queremos utilizar un pulsador corriente, sino un botón que podamos pulsar desde dentro del panel de control.

Botón

Encontrarás este botón de instrucción en la ventana de elementos, en el grupo **Elementos de control / elementos de mando**. Ve a la barra de función para pasar a **Panel de control** e inserta un botón al lado del mensaje de texto de tu panel de control. La inscripción **Botón** no es muy adecuada, pero puede cambiarse fácilmente desde su ventana de propiedades. Haz clic con el botón derecho del ratón sobre el botón, introduce en **Texto de leyenda** las palabras que quieras (por ejemplo 0000) y confirma con **OK**.

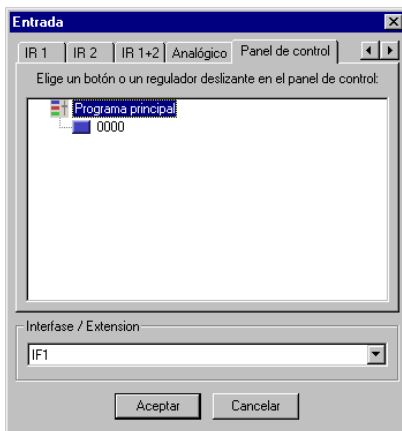


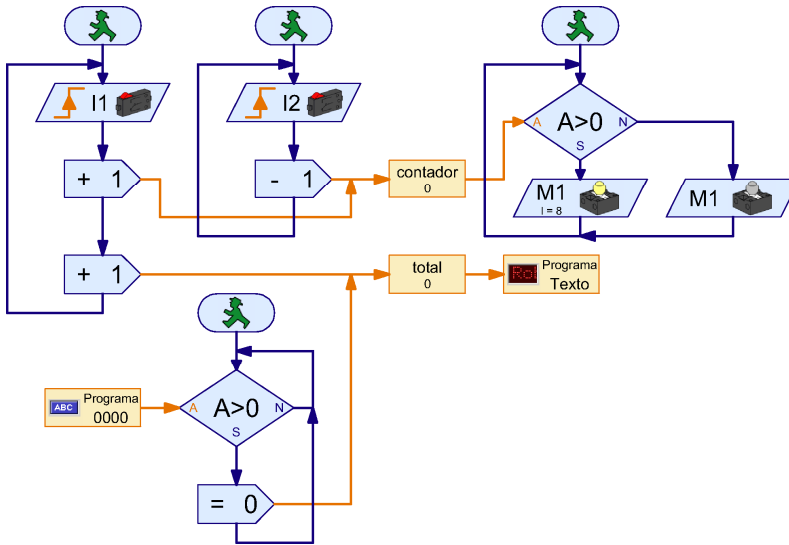
Igual que con el mensaje de texto, necesitaremos también un elemento de programa que una el botón con el desarrollo del programa. Por eso debes pasarte de nuevo a **Función en la barra de función**. Encontrarás en la ventana de elementos, en el grupo **Entradas, Salidas**, el elemento **Entrada del panel de control**, cuya imagen ves aquí. Colócalo en el programa por debajo de lo que llevamos hasta ahora.

Ahora solamente te queda enlazar la entrada del panel de control con el botón. Para ello haz clic con el botón derecho del ratón sobre el elemento entrada del panel de control. Los elementos de mando están ordenados, igual que los mensajes, por subprogramas, dado que cada subprograma puede tener su propio panel de control. Ahora elige el botón **0000** y confirma con **OK**.

Quizás habrás notado que este elemento puede configurarse (gracias a la barra de pestañas de la ventana de propiedades) a todos los tipos posibles de entradas. Pero esto lo explicaremos más adelante, dentro de dos secciones, en *Entradas de datos para subprogramas*.

Para consultar el valor que devuelve la entrada del panel de control utilizamos un elemento de bifurcación. Es el mismo elemento que ya habías utilizado para consultar la variable. El programa terminado y puesto a cero tiene ahora este aspecto:





Mientras que el botón **0000** esté pulsado, se enviará al contador total una instrucción **= 0**, que pone el contador precisamente a 0.

5.4 Temporizador

Después de tus éxitos, el director ya no sabe qué hacer sin ti y te nombra asesor informático del museo. Un cargo de esta entidad es un gran honor, como es natural, pero también conlleva mucho trabajo: en el museo hay muchos modelos que se mueven mediante la pulsación de un botón. El caso es que muchas de las personas que van a ver el museo pulsan los botones tantas veces que los modelos se calientan y acaban continuamente en el taller. Ahora el director quiere que los modelos estén en marcha cuantas veces se pulse el botón, pero como máximo durante 30 segundos cada vez. Una vez que el modelo haya estado en marcha, debe hacerse una pausa de 15 segundos antes de poder reconectarlo de nuevo.

Pues no es tan difícil, pensarás: unos tiempos de espera, unas bifurcaciones de programa y listo. ¡Inténtalo! No tardarás en descubrir que no es tan fácil. Y no lo es por dos razones:

- Durante el intervalo de 30 segundos, el programa debe consultar al botón para averiguar si el botón ha sido soltado ya antes del transcurso de los 30 segundos. Vale, es cierto: esto puede solucionarse con dos procesos que se desarrollen al mismo tiempo (ver sección 5.2 *Variables y varios procesos*, página 487).
- Si un visitante soltara el botón después de 5 segundos y lo pulsara de nuevo después de 15 segundos, el tiempo de espera de 30 segundos tendría que iniciarse de nuevo. Pero el tiempo de espera transcurrido sería solamente de $5 + 15 = 20$ segundos, y con ello aún estaría activo. Ni siquiera con procesos de desarrollo en paralelo se puede reiniciar un tiempo de espera. Quizás funcionaría con dos tiempos de espera en tres procesos de inicio alternativo, pero eso ya te causaría unos quebraderos de cabeza espantosos.



¿No hay nada más fácil? Claro que sí: las **Variables de temporizador** o, abreviando, un **Temporizador**. En primer lugar, un temporizador

funciona como una variable normal y corriente. El temporizador memoriza un número y tú puedes modificar el número con las instrucciones =, + y -. La peculiaridad del temporizador es que cuenta, por sí mismo y a intervalos variables, desde un número hacia atrás hasta llegar a 0. El intervalo de cada paso de conteo puedes configurarlo entre una milésima de segundo y un minuto. Con los temporizadores se pueden solucionar muchos problemas de control de tiempo de una manera más elegante que con los tiempos de espera. ¿Empiezas a intuir ya cómo podrías solucionar el problema con un temporizador?

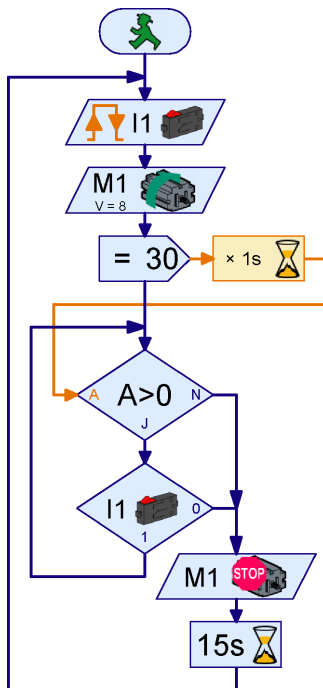


Exacto: en cuanto el visitante pulse el pulsador de I1, enciendes el modelo y pones el temporizador con una instrucción = a 30 x 1 segundo = 30 segundos. Después consultas en un bucle si ha transcurrido el tiempo de 30 segundos y si se ha soltado el pulsador de I1. Si se cumple uno de los dos criterios de interrupción, detienes el modelo y esperas 15 segundos. Después, vuelta a empezar.



Tienes razón: los programas son cada vez más exigentes.

Pero intenta solucionar esta propuesta: desarrolla un programa que desempeñe la misma función con tiempos de espera en vez de temporizadores. **Atención: ¡Esta es una tarea muy difícil y está pensada solamente para aquellos a quienes les gusta romperse la cabeza con un enigma! Los demás pueden saltarse el resto de la sección.** Para este trabajo hay dos principios de solución: puedes utilizar dos tiempos de espera que iniciarás alternativamente en procesos propios. Dado que hay un tiempo muerto de 15 segundos, uno de los dos tiempos de espera habrá transcurrido como mucho después de la segunda vuelta, así que puede iniciarse de nuevo. Otra alternativa sería reconstruir un temporizador con una variable corriente y un elemento **tiempo de espera** con un tiempo de espera muy cortito, por ejemplo de un segundo.

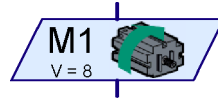


5.5 Entradas de comando para subprogramas

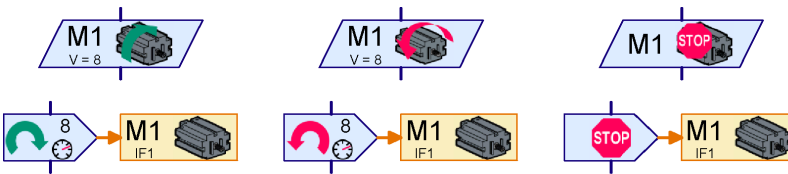
Como siempre tu programa funciona de maravilla y fischertechnik está encantado, porque todos los modelos del museo están equipados con la ROBO Interface. Pero por desgracia, como en todas las instituciones oficiales del mundo, el museo tiene un presupuesto muy limitado. Por eso el director quiere arreglarse con el menor número posible de interfases. Al fin y al cabo, una ROBO Interface tiene cuatro salidas de motor y también entradas suficientes para activar cuatro modelos. Dado que la mayoría de los modelos sólo pueden girar en una dirección, puedes llegar a activar hasta 8 modelos mediante las salidas unipolares O1 a O8.

Como es natural, esto ahorrará mucho dinero al director del museo. Para conseguirlo tendrás que copiar el programa 7 veces y adaptar las entradas y salidas en cada copia. ¿O no es así? ¿No se podría hacer también con subprogramas?

Sí que se podría, pero hay un problema: si en un subprograma utilizas las consultas habituales al pulsador y los elementos de motor usuales del grupo **Elementos básicos**, cada llamada del subprograma consultará el mismo pulsador y activará los mismos motores. Esto se debe a que, por ejemplo, en un elemento de salida de motor la instrucción de control para el motor (derecha, izquierda o detener) y el número de salida de motor (M1... M8) forman una unidad. Pero dado que solamente existe un subprograma, tiene siempre también el mismo motor. Si en una llamada de subprograma cambiaras el número de la salida de motor, este número se cambiaría también en todas las llamadas existentes del subprograma. Así que deberías copiar de nuevo el subprograma 7 veces, dar un nombre distinto a cada subprograma y adaptar las entradas y salidas a mano en todas partes.

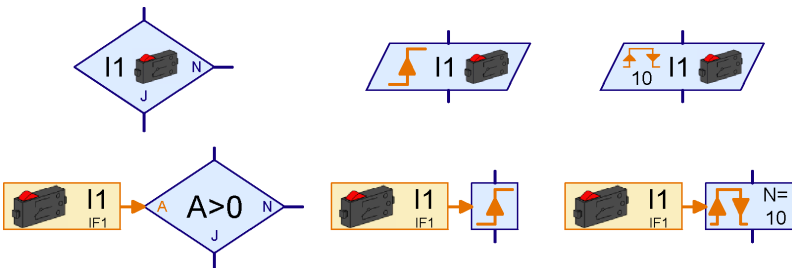


Pero se puede solucionar este problema de una manera mucho más elegante. El truco consiste en separar las instrucciones de control de los símbolos de motor. De este modo se podrán colocar las instrucciones de control (izquierda, derecha, detener) en el subprograma y los elementos de motor en el programa principal. Después usas un elemento de comando (que ya vimos con las variables) en el desarrollo del subprograma para enviar las instrucciones izquierda, derecha o detener al programa principal, donde después las podrás transmitir a los distintos motores. Para el motor hay un elemento que se limita a representar un motor en general, sin fijar qué es lo que debe hacer. Este elemento tiene una entrada a la que puedes enviar instrucciones. Los elementos del grupo de elementos básicos puedes sustituirlos por un elemento de comando y un elemento de motor de la siguiente forma:



En la línea superior verás un elemento de motor del grupo **Elementos básicos**. En la segunda línea está ilustrada la combinación de un elemento de comando del grupo **Instrucciones** y un elemento de motor del grupo **Entradas, Salidas**, que tiene exactamente el mismo efecto. Realmente los elementos superiores no son más que abreviaciones o simplificaciones de las combinaciones de la línea inferior. Ambas envían una instrucción derecha, izquierda o detener al motor **M1**.

Lo mismo funciona también para la consulta de pulsadores:



En la línea superior verás de nuevo elementos del grupo **Elementos básicos**. En la línea inferior encontrarás una combinación de una entrada digital y un elemento del grupo **Bifurcación, Espe-**

rar... Encontrarás el elemento naranja Entrada digital igual que el elemento de motor, en el grupo **Entradas, Salidas**.

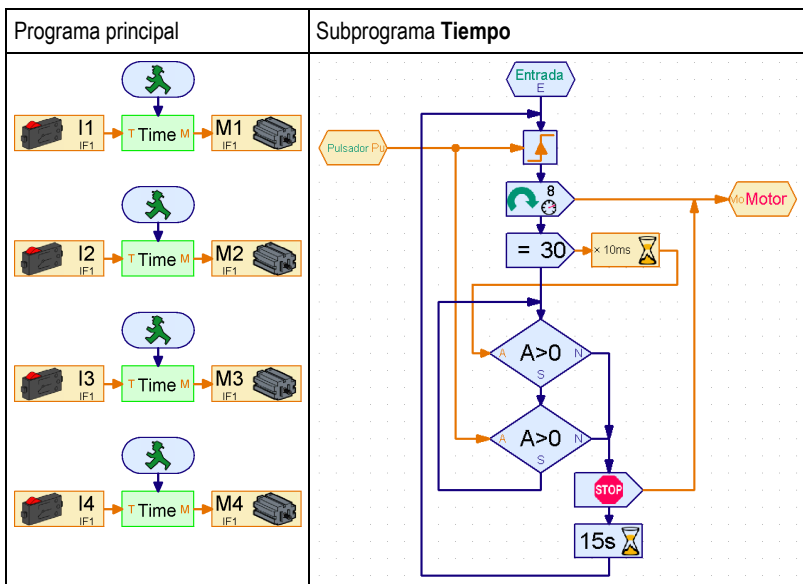
Con este truco puedes separar la lógica de un desarrollo de programa y las entradas y salidas. Pero todavía falta algo. Si los elementos de motor y de pulsadores deben estar en el programa principal y las instrucciones en un subprograma, debería haber un modo de unir los pulsadores y los elementos de motor con el subprograma. Encontrarás los elementos de conexión necesarios para hacerlo en el grupo **Subprograma I/O**.

Entrada En — Mediante una entrada de comando puedes enviar instrucciones desde el exterior a un subprograma. El elemento de entrada digital (pulsador) envía a través de la línea naranja su nuevo valor siempre que varíe el estado de la entrada (con lo que se llama una "instrucción ="). En el cuadro de diálogo del elemento puedes darle nombre a la entrada.

— **Salida Sa** Mediante una salida de comando puedes enviar instrucciones desde un subprograma hacia el exterior. Así podrás, por ejemplo, enviar las instrucciones izquierda, derecha y detener desde un subprograma a un motor. También en este elemento puedes introducir un nombre en el cuadro de diálogo.

diálogo.

Ya lo tienes todo articulado para proceder con tu **temporizador múltiple de modelo con subprogramas**:



- El subprograma Tiempo es prácticamente igual que el programa de la sección anterior, pero los elementos **Esperar a entrada digital I1** del inicio y del bucle han sido sustituidos por los elementos **Esperar a** con conexiones de datos para líneas naranjas del grupo **Bifurcación, Esperar...** Ambos están unidos con la entrada de comando del subprograma **Pulsador**. Los dos elementos de motor del inicio y del final del programa han sido sustituidos por elementos de comando. Ambos envían sus instrucciones a la salida de comando del subprograma **Motor**.

El subprograma **Tiempo** es consultado cuatro veces en el **Programa principal**. La entrada de comando del subprograma **Pulsador** ha creado automáticamente la conexión naranja **P** en el símbolo verde del subprograma de la parte izquierda. A través de la salida de comando del subprograma **Motor** se ha producido la conexión **M** en la parte derecha. La conexión **P** del símbolo del subprograma se une con el pulsador (de **I1** a **I4**) **correspondiente**. En la conexión **M** se conectará el motor **M1** a **M4** **que corresponda**. De esta manera cada consulta del subprograma **Tiempo** consultará otro pulsador y activará otro motor.

Intenta copiar el subprograma superior y el programa principal y pruébalos. Primero debes dibujar el subprograma, puesto que de otra manera no podrás insertar el subprograma en el programa principal. Si tienes problemas con el subprograma, puedes consultar otra vez el capítulo 4 *Nivel 2: Trabajar con subprogramas*, página 464.

Nota: encontrarás más información sobre las entradas de comando en la sección 6.3 *Enviar cualquier comando a subprogramas* de la página 504.

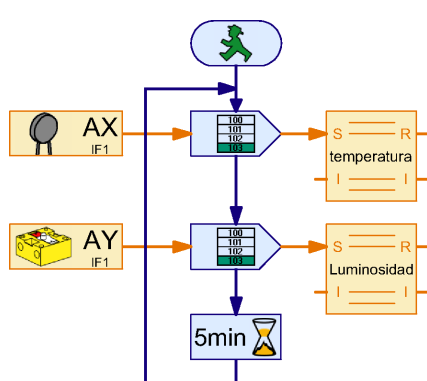
5.6 Listados (Arrays)

Después de que todas las estructuras de ensayo del museo han sido equipadas con tu control reductor de gastos, el director del museo no tarda mucho en comunicarte su siguiente problema: en una sala que alberga piezas de exposición antiguas y muy valiosas han sucedido últimamente oscilaciones peligrosas de temperatura. Tu sospecha es que tiene que ver con la radiación solar. Para demostrar esta relación, quieres construir un aparato que grabe la luminosidad y la temperatura. Sabemos que la ROBO Interface tiene varias entradas analógicas y también sabes ya cómo guardar valores mediante las variables. Así que todo eso no te supondrá ningún problema. ¿O sí? Para grabar dos valores cada 5 minutos durante 12 horas se necesitan ¡288 variables! Saldría un programa gigante y complejísimo. ¿Quizás se podría simplificar con subprogramas? Pues sí, pero hay una manera mucho mejor: el elemento **Listado** (los programadores lo llaman "Array").

Dentro de un listado puede guardarse no solamente un valor, sino toda una lista de valores. Lo normal es que al principio el listado esté vacío. Si envías una instrucción **Agregar** a la entrada de datos situada en la parte superior izquierda (con la denominación **S**), se agregará el valor indicado en este elemento de comando al final del listado. Puedes configurar la longitud máxima del listado con un valor de entre 1 y 32767 desde la ventana de propiedades del elemento **Listado**. Con ello, generar el programa para grabar temperatura y luminosidad será muy fácil:

En la entrada analógica **AX** está conectado el sensor de temperatura; en la entrada analógica **AY**, el sensor de luminosidad. El programa leerá ambos valores en un ciclo de 5 minutos y los agregará mediante la instrucción **Agregar** a cada listado.

Nota: Al insertar el elemento de comando debes activar en la ventana de propiedades la opción **Entrada de datos para valor de instrucciones**. Entonces aparecerá una entrada de datos a la izquierda en el elemento de comando, a la que podrás



conectar la entrada analógica.

Para probar el programa sería útil reducir el tiempo del ciclo de 5 minutos a unos segundos.

Seguramente te preguntarás ahora cómo podrás leer en el listado los valores guardados. Para ello hay dos posibilidades: puedes leer los valores como si se tratase de una variable normal y corriente y procesarlas a continuación en tu programa. Dado que el listado contiene varios elementos, eliges primero en la entrada de datos izquierda con la denominación I el número de elementos que quieras leer. Entonces saldrá el valor que tiene ese elemento en la salida de datos D de la parte derecha.

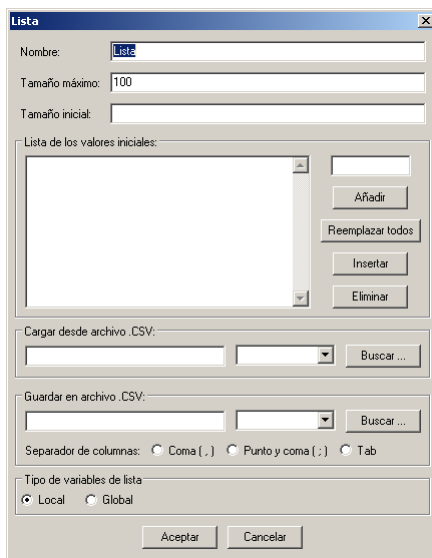
Pero ROBO Pro también puede guardar todos los valores del listado en un archivo de tu ordenador que más adelante podrás procesar (por ejemplo en Excel). Dado que en el caso presente solamente deseas mirar y comparar las luminosidades y temperaturas grabadas, seguramente esta opción sería más práctica. ROBO Pro guarda los valores en lo que se llama un **archivo CSV** (comma separated values = valores separados por comas). Los archivos CSV son archivos de texto que contienen una o más columnas con una serie de datos cada una. Así, puedes guardar varias series de mediciones (como temperatura y luminosidad) en las distintas columnas de un archivo CSV. Las columnas están separadas por comas. En los países en los que 0,5 se escribe con coma y no 0.5 con punto (por ejemplo, en España), también es frecuente que se utilice como signo de separación de las columnas también un **punto y coma (;)**. Si esto te causa

problemas a la hora de exportar archivos CSV de ROBO Pro y (por ejemplo) Microsoft Excel, puedes cambiar en la ventana de propiedades del listado el **signo de separación de columnas**.

Puedes configurar el nombre del archivo CSV y la columna en la cual debe guardarse el contenido de un listado en la ventana de propiedades del listado, en **Guardar archivo CSV**. Se guardarán los datos si sales del programa en modo online o si seleccionas en el menú **Archivo** la opción **Guardar archivo CSV** mientras el programa esté en marcha (modo online o descarga). En el modo de descarga puedes separar la ROBO Interface para grabar datos del PC y después conectarla de nuevo para guardar.

Una vez que hayas ejecutado el programa de anterior en modo online, puedes abrir el archivo de datos .CSV, creado con ROBO Pro, en Microsoft Excel u otra hoja de cálculo. Si no tienes un programa de este tipo, también puedes utilizar el editor de Windows (Bloc de notas.exe), que encontrarás casi siempre en el menú de Inicio de Windows, en Accesorios.

Mientras el programa se ejecuta en modo online, puedes ver los datos en un listado haciendo clic con el botón derecho del ratón sobre el elemento del listado.

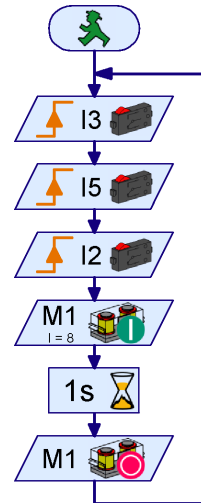
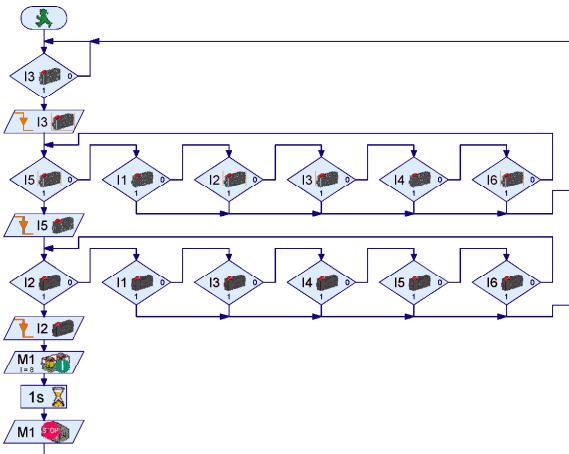


5.7 Operadores

El programa que has creado para grabar luminosidad y temperatura ha funcionado bien, pero la grabación muestra que la temperatura de la sala de exposición del museo no tiene nada que ver con el sol. Resulta que unos visitantes han confundido el climatizador de la sala de exposición con un control de un modelo y se ha dedicado a darle al botón del aparato sin parar. ¡No es de extrañar que la temperatura de la sala de exposición se haya vuelto loca!

Pero este problema puede solucionarse fácilmente: colocando con un candado de combinación electrónico. El candado de combinación debe tener un teclado con teclas de 1 a 6. Si se introducen correctamente 3 cifras seguidas, el candado con teclado debe desbloquear mediante un imán la tapa del climatizador.

A primera vista ese candado con teclado es muy sencillo: el programa espera hasta que se hayan pulsado las teclas correctas en el orden correcto. A la derecha puede verse ese tipo de programa para la combinación 3-5-2. A segunda vista, sin embargo, este programa tiene un problema: el candado puede descifrarse fácilmente pulsando 3 veces seguidas todas las teclas de 1 a 6, puesto que de este modo se habrá pulsado sea como sea la tecla correcta. Como ya dijo Albert Einstein: "Hay que hacer las cosas lo más fáciles posibles, pero no más". Así que el programa no solamente debe consultar si se pulsan las teclas correctas, sino también si se ha pulsado una tecla incorrecta. Entonces el programa tendrá este aspecto:



Este programa abre el candado solamente si se pulsan las teclas 3 5 2 pulsar en el medio ninguna otra tecla. Por ejemplo: si se pulsa el 3, el programa espera a que se suelte la tecla. Si a continuación se pulsa cualquier otra tecla en vez de la tecla 5, el

programa se reinicia. De este modo el programa funciona correctamente, pero no es ni fácil ni claro. Además es bastante difícil cambiar el código. Pero no te preocupes: se puede arreglar de una manera sencilla y correcta: con **Operadores**. Hay diferentes tipos de operadores. Los puedes encontrar en **Elementos de programa**, en el grupo **Operadores**. Para el candado de combinaciones necesitaremos primero un **Operador O**.



¿Funcionará también el candado de combinaciones para códigos de 2 a 4 cifras? Si es así, ¿hasta qué número de cifras funcionará y por qué? ¿Qué pasa con los otros programas de candado de combinaciones?

6 Nivel 4: comandos definidos por el usuarionds

¡No te olvides de cambiar la ROBO Pro a **Nivel 4** (o superior) en el menú **Nivel!**

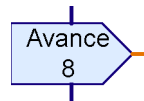
En el Nivel 3 has explorado en detalle cómo procesar datos con ayuda de comandos y, por ejemplo, controlar motores. Para ello ha empleado únicamente comandos predeterminados, como la orden = o las instrucciones de derecha, izquierda o detención. En el Nivel 2, concretamente en la sección 4.5 *Tango 2: comunicación a través de ROBO RF Data Link* (que vimos en la página 478) ya has comprobado cómo puedes utilizar comandos ideados por ti mismo. En el Nivel 4 estas dos ideas, el envío de comandos por conexiones naranjas y la utilización de comandos propios, se combinan entre sí.

6.1 Procesamiento de comandos en un proceso

Seguramente ya habrás diseñado algún programa que controla un robot propulsado por dos ruedas o un vehículo de cadena. Independientemente de que quieras que el vehículo se desplace a la izquierda, a la derecha, hacia atrás o hacia adelante, siempre tienes que dar a los dos motores un comando de derecha, izquierda o parada. Además tienes que fijarte constantemente en cuál es el motor que mueve la rueda derecha y cuál el que mueve la rueda izquierda, además de discurrir si el motor debe girar hacia la derecha o hacia la izquierda para que el vehículo avance. Pero una mente inteligente como la tuya estaría mejor ocupada en otros asuntos, inventando ideas geniales, por ejemplo, sin necesidad de ocuparse de semejantes nimiedades.

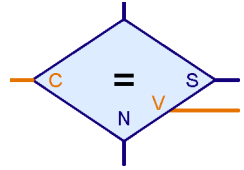
Como ya sabemos, una solución a este problema es diseñar un subprograma para cada operación. Lo que ocurre es que sería mucho más elegante si pudiésemos diseñar un subprograma que como salida de motor tenga una entrada de comando a la que tan sólo haya que enviar comandos como avance, retroceso, izquierda, derecha y parada, y que se encargue solo de que los dos motores se muevan como deben para conseguir ese efecto.

Seguramente estarás pensando que en el elemento de comando de ROBOPro efectivamente hay comandos de izquierda y derecha, pero no de avance ni de retroceso. Es cierto, pero también lo es que nuestros queridos elementos de comando de toda la vida nos pueden dar muchas sorpresas... Te proponemos una cosa: diseñes un programa nuevo, que lleves el elemento de comando que te apetezca al programa principal y que en la ventana de propiedades indiques simplemente "Avance" como comando. Y comprobarás que... ¡funciona!



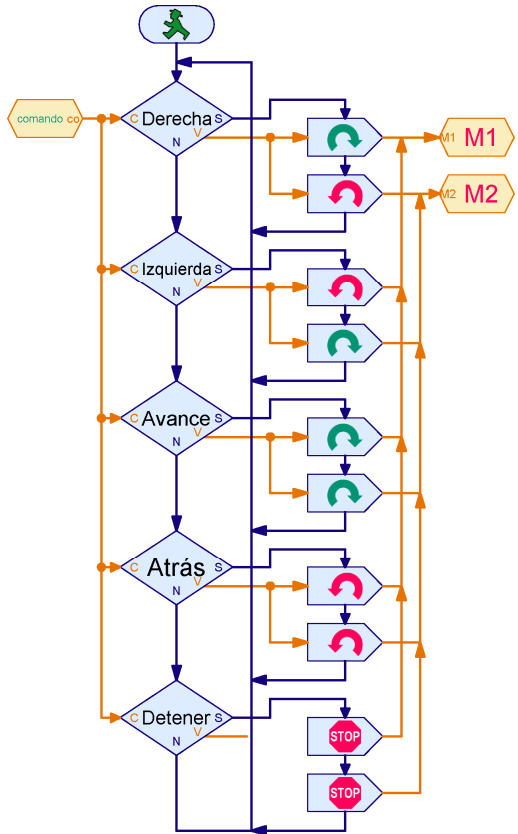
La siguiente pregunta es: ¿qué se hace con semejante elemento de comando? Porque de hecho no hay ningún elemento que pueda procesarlo. Si por ejemplo envías el comando de avance a una salida de motor y tratas de iniciar el programa, ROBOPro te dará un mensaje: "Ninguna entrada conectada puede procesar el comando Avance". A partir del Nivel 4 hay dos elementos nuevos, nada llamativos pero extremadamente rentables, que pueden procesar los comandos que quieras: el elemento "Esperar comando" y el filtro de comandos. Los encontrarás a ambos al final del grupo de elementos **Enviar, recibir**.

La tarea de diseñar un subprograma que nos permita controlar un robot de 2 ruedas con los comandos avance, retroceso, derecha e izquierda se soluciona con estos dos elementos. Para empezar vamos a probar con el elemento "Esperar comando". A este elemento le puedes enviar el comando que quieras por la entrada de comando **C**. Lo que ocurre es que el elemento espera siempre un comando específico que puedes determinar. Cuando el elemento ha recibido el comando en cuestión, el flujo de programa pasa a la salida **S**; de lo contrario, pasa a la salida **N**. Como ya sabes, en ROBOPro los comandos constan de un nombre y un número, el valor de comando. Cuando el elemento "Esperar comando" ha recibido el comando que estaba esperando, el valor de comando está disponible en la salida **V**.



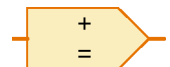
El subprograma deseado es muy sencillo de hacer. En un bucle sin fin se consultan cada uno de los 5 posibles comandos de control con un elemento "Esperar comando". En el momento en que se ha recibido el correspondiente comando, el valor de comando de transmite a los comandos derecha e izquierda que se envían a las salidas de motor M1 y M2. Si por ejemplo el subprograma recibe un comando de avance con el valor 8, el valor 8 de la salida E del filtro de comandos se transmite a dos elementos de comando que a continuación envían a los dos motores un comando de derecha con este valor como dato de velocidad. En estos casos, dicho sea de paso, no es demasiado práctico utilizar un único elemento de comando de derecha que el comando envíe a ambos motores. En este tipo de construcciones suele ser muy difícil distinguir las líneas de comando de los motores, de manera que muchas veces se mandan también comandos a los dos motores.

En el ejemplo la salida E del filtro de comandos de parada no está conectada porque los elementos de comando de parada no necesitan ningún valor.



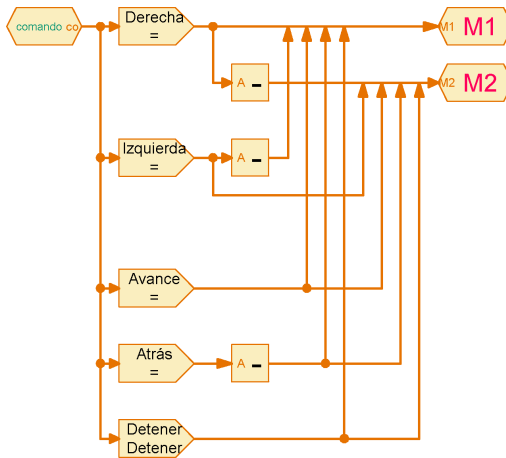
6.2 El filtro de comandos

La tarea de la sección anterior puede resolverse con ayuda del filtro de comando. El filtro de comandos permite "rebautizar", por así decirlo, los comandos. Si a la entrada izquierda se envía un comando determinado, el



elemento envía al elemento conectado a la salida derecha otro comando, pero con el mismo valor que el recibido en la entrada. De este modo puede hacerse a partir de un comando =, por ejemplo, un comando de control de motor como derecha o izquierda. En especial puede usarse el Filtro de comandos para transformar los comandos propios que se desee en comandos estándar de ROBOPro, para poder lanzar acciones con los comandos propios.

En la imagen de la derecha verás cómo se puede estructurar el subprograma de control de un robot de 2 ruedas con el filtro de comandos. El filtro de comandos de arriba de todo transforma, por ejemplo, el comando de **derecha** en un comando =. Las salidas de motos pueden procesar los comandos = con un valor de -8 a 8. Como en el giro a la derecha del modelo los dos modelos giran en sentidos contrarios, el valor del comando = para el motor **M2** debe hacerse negativo con un operador -. En el comando de izquierda, por el contrario, el valor que debe ser negativo es el del motor **M1**. El avance y el retroceso son más sencillos, porque ambos motores giran en el mismo sentido.



Esto significa que no es imprescindible cambiar los comandos con el filtro. El último filtro de comandos tiene el comando de **detener** como comando tanto de entrada como de salida. Con este elemento los comandos de parada se transmiten directamente a los motores. De todas formas sí necesitarás un filtro de comandos, para que otros comandos (como el de **derecha** o el de **izquierda**) no se envíen directamente a las salidas de motor.

La gran ventaja del filtro de filtro de comando respecto del elemento de "Esperar comando" que vimos en la sección anterior es que no se necesita ningún proceso. Esto ahorra memoria y permite que el procesamiento tenga lugar de inmediato, sin tener que esperar al próximo cambio de proceso.



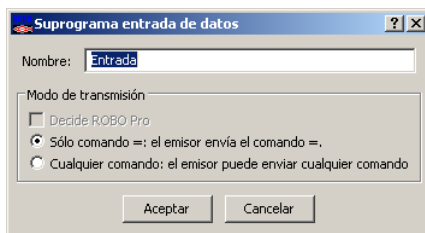
Al aplicar el filtro de comando de esta manera tienes que asegurarte de que no has confundido las líneas de datos de los dos motores. Lo más fácil es que todas las flechas apunten a una única línea de datos, tal y como se ilustra en la imagen anterior. A veces también resulta más fácil utilizar dos filtros de comandos para un comando de manera que tengamos dos salidas separadas.

En el programa anterior los robots giran sobre sí mismos con el comando de **derecha**. Trata de modificarlo de manera que el M1 se mueva y el M2 se quede quieto. Tendrás que usar dos filtros para el comando de derecha. Uno transforma el comando en un comando = y el otro en un comando de detención. En el comando de parada se hace caso omiso del valor de comando.



6.3 Enviar cualquier comando a subprogramas

En la sección, 5.5 *Entrada de comando de subprograma (Nivel 3)* (página 492) te presentamos las entradas de comando para subprogramas. Sin embargo, en ese momento sólo conectabas elementos digitales o analógicos a las entradas de comando. Dichos elementos envían siempre un comando = cuando varía el valor de la entrada. Si quieres enviar otro comando distinto a una entrada de comando de un subprograma, tendrás que indicarlo en la ventana de propiedades de la entrada de comando. A partir del Nivel 4 se añade en la ventana de propiedades una nueva opción de **modo de transmisión**.



Si aquí seleccionas **Sólo comandos =**, sólo podrás enviar comandos = a la correspondiente entrada de llamada de subprograma. Además, el último comando = se **repite automáticamente** cuando se inicia el subprograma. De otro modo la entrada de subprograma no tendría el valor adecuado en el momento de inicio. Imagínate que en la entrada de subprograma hay conectado un elemento de entrada digital. Estos elementos sólo envían comandos cuando varía el valor de la entrada de la interfase. Si la entrada está conectada, el elemento de entrada digital envía un comando = 1. Cuando se inicia el subprograma después de haberse enviado el comando, es importante que el comando se mande de nuevo una vez iniciado el subprograma. Si no se hiciese así, la entrada tendría el valor que no es hasta que el elemento conectado a la entrada de subprograma volviese a cambiar de valor.

Este mecanismo automático de envío, no obstante, también puede dar sus problemas, razón por la cual en la mayoría de comandos no conviene introducirlo. Por ejemplo, si envías un comando de inicio o de +1 a un subprograma, por norma general no querrás que lo repita. Por ello no se envían los comandos por duplicado si seleccionas la opción **Cualquier comando**.

Al iniciarse el subprograma no se repite el comando ni siquiera cuando envías **un comando =** a una entrada de **Cualquier comando**. Puede ocurrir que el valor que transmite la entrada de subprograma **no coincida** con el valor efectivo de la salida.

7 Módulos de extensión y gestión de más de una interfase

Con una ROBO Interface o una interfase inteligente pueden controlarse modelos ya bastante sofisticados, pero quizás a alguno de vosotros os apetezca un reto algo más complicado. Si 8 pulsadores, 4 motores y 4 entradas analógicas no son suficientes para ti, puedes ampliar tu ROBO Interface con un máximo de 3 **ROBO I/O-Extensions** o tu interfase inteligente con un **módulo de extensión**. Y para aquellos a quienes esto todavía no les parece suficiente, existe aún la posibilidad de trabajar en modo online con varias interfasas a la vez (y posiblemente cada interfase tenga sus propios módulos de extensión) desde un solo programa. El número de entradas y salidas estará limitado únicamente (al menos en el modo online) por la potencia de tu PC.

7.1 Módulos de extensión

Quizás ya te habrás fijado en el menú desplegable de nombre **Interfase / Extensión** que aparece en la ventana de propiedades de los elementos de entrada y de salida. Se trata del lugar donde puedes seleccionar en qué interfase o módulo de extensión se encontrará la entrada o la salida. A no ser que hayas configurado otra cosa (ver próxima sección), el menú desplegable tendrá las siguientes entradas:

- **IF1:** Es la interfase en sí (no un módulo de extensión)
- **EM1...EM3:** Son los módulos de extensión de 1 a 3. Si utilizas una interfase inteligente, puedes utilizar solamente un módulo de extensión (**EM1**), pero de todos modos se mostrarán también los módulos de extensión **EM2** y **EM3**, dado que el software es el mismo independientemente de si utilizas una ROBO Interface o una interfase inteligente.



Así pues es muy fácil utilizar módulos de extensión como **ROBO I/O Extensions**. Lo único que tienes que hacer es seleccionar en las entradas y salidas el módulo deseado (interfase o extensión).

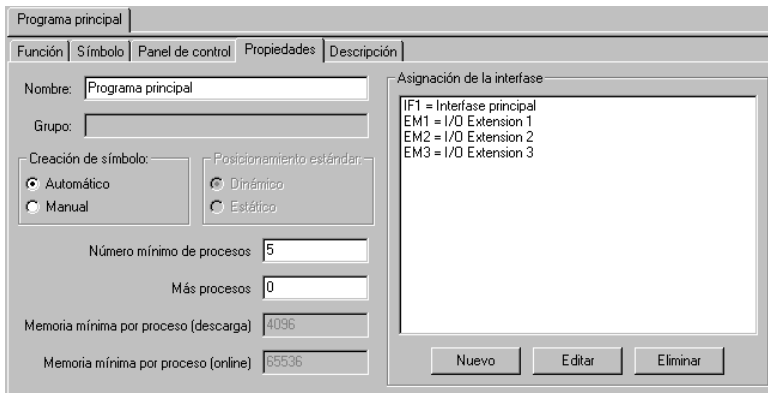
7.2 Múltiples interfasas

Si quieres activar más de una interfase desde un programa, comprobarás que es posible hacerlo solamente en modo online. Por ejemplo, puedes conectar una ROBO Interface en el puerto COM1 y otra en el puerto USB. Ambas pueden estar equipadas con hasta 3 módulos ROBO I/O-Extension. Otra posibilidad de combinación sería, por ejemplo, 2 ROBO Interfaces en el USB (las dos con sus I/O-Extensions) y una interfase inteligente en el puerto COM1 (preferiblemente también con módulo de extensión). Para que puedas definir en la ventana de propiedades de entradas y salidas a cuál de las interfasas debe dirigirse, tienes que adaptar la asignación de interfase.

Mientras no configures otra cosa, en el menú desplegable de **Interfase / Extensión** encontrarás las entradas **IF1**, **EM1**, **EM2** y **EM3**. Pero también puedes cambiar o ampliar el listado. Podría haber varias razones para ello:

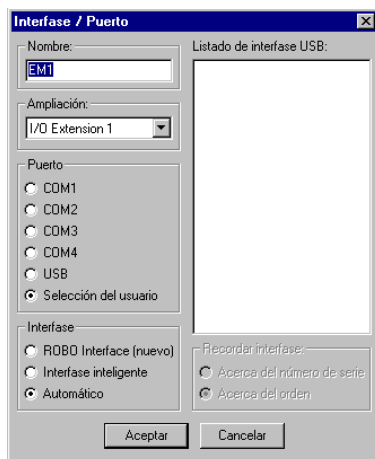
- Para obtener una mejor visión de conjunto, quieres que en vez de IF1 o EM1 los módulos tengan un nombre que haga referencia a la pieza de la máquina o del robot que controlan.
- Quieres intercambiar dos módulos de extensión (por ejemplo EM1 y EM2) porque va mejor para los cables, pero no quieres cambiar el programa.
- Quieres ejecutar con varias interfases inteligentes un programa que ha sido escrito para una ROBO Interface equipada con más de un módulo de extensión.
- Quieres utilizar en tu programa más de una ROBO Interface o más de una interfase inteligente.

Todo esto puedes hacerlo muy fácilmente cambiando la **asignación de interfase** en la ventana de propiedades del **programa principal**:



Aquí puedes ver qué módulos (interfase o extensión) se corresponden con los nombres **IF1** a **EM3**. Con el botón **Nuevo** puedes añadir una nueva interfase. Si quieres cambiar una entrada del menú, simplemente la eliges y haces clic en **Edición**. En ambos casos se mostrará la siguiente ventana:

- En **Nombre** puedes cambiar la denominación del módulo. El nombre no debería ser demasiado largo, dado que el espacio disponible en los símbolos gráficos es muy pequeño. Si cambias el nombre, casi siempre deberás cambiar también el nombre del módulo en todos los elementos de entrada y salida que hagan referencia a él.
- En **Extensión** puedes indicar si el nombre se refiere a una interfase o a un módulo de extensión 1 a 3.
- En **Puerto** puedes seleccionar en qué puerto estará conectada la interfase. Si indicas aquí **Selección del usuario**, se utilizará la interfase que hubieses seleccionado en la barra de herramientas en **COM/USB**. Mientras quieras utilizar solamente una interfase con varios módulos de extensión y no más, esto será lo más fácil, dado que de esta manera otras



personas podrán utilizar tu programa sin cambios. Si utilizas varias interfases, debes configurar aquí el puerto en el cual estará conectada cada una.

- En **Interfase** puedes indicar si quieres utilizar una ROBO Interface o una interfase inteligente. Si la interfase está conectada con el puerto serie, el programa podrá reconocer automáticamente de qué interfase se trata (selección **automática**).
- La parte derecha de la ventana solamente es importante si tienes varias ROBO Interface conectadas en el bus USB. Si en **Puerto** haces clic en **USB**, puedes seleccionar en Listado de interfases **USB una de las interfases**.

Atención: Si quieres trabajar con varias interfases en el bus USB, primero tienes que asignar un número de serie propio a cada interfase. Por defecto todas las interfases vienen con el mismo número de serie para evitar problemas a la hora de cambiarlas, pero el sistema operativo Windows reconoce solamente interfases con números de serie distintos. Para saber más sobre esto, consulta la sección 7.5 *Cambiar el número de serie de la interfase o versión de firmware*, página 508.

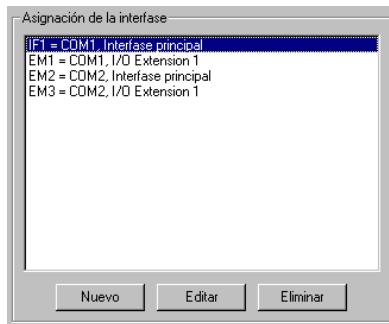
- En **Recordar interfase** puedes configurar el modo en que el programa recordará la interfase seleccionada. Existen dos posibilidades: si seleccionas **Por número de serie**, el programa guardará el número de serie de la interfase. Así, si conectas y eliminas varias interfases en el bus USB, el programa siempre podrá encontrar la interfase seleccionada mediante el número de serie. La desventaja es, sin embargo, que a partir de entonces el programa solamente funcionará con la interfase que tenga el número de serie en cuestión. Si quieres utilizar una interfase con otro número de serie, debes cambiar la asignación de interfase o el número de serie de la interfase. Para evitar problemas con los números de serie, existe la segunda posibilidad: **Por orden**. Si marcas esta opción, el programa guardará en el listado el orden y no el número de serie. Este sistema podría llevar a confusión si añades o eliminas interfases en el bus USB, pero lo positivo es que el programa se ejecutará sin cambios con cualquier interfase.

7.3 Asignaciones de interfase en subprogramas

Normalmente todas las asignaciones de interfase de tu programa se hacen desde la ventana de propiedades del programa principal, pero también puedes introducirlas desde los subprogramas. Si es así, podrás utilizar las asignaciones de interfase tanto del subprograma como del programa principal. Si dos asignaciones tienen el mismo nombre, tendrá prioridad la asignación del subprograma. Así podrás, por ejemplo, definir que IF1 acceda a la interfase principal en el programa principal, pero que en un determinado subprograma IF1 corresponda a un módulo de extensión. Esto resulta muy práctico si quieres controlar todo un parque de maquinaria en el que cada máquina va a estar controlada por una interfase propia. Entonces podrás desarrollar los controles para cada máquina en principio como programas independientes y hacer que cada programa principal acceda a IF1. Más adelante podrás insertar todos los programas principales de las máquinas en un programa completo como subprogramas. En el programa completo necesitarás cambiar solamente las asignaciones de interfase, pero no los nombres en cada entrada y cada salida.

7.4 Consejos & Trucos

Si quieres ejecutar en 2 interfases inteligentes equipadas con un módulo de extensión cada una un programa que ha sido desarrollado para una ROBO Interface equipada con 3 módulos de extensión, puedes utilizar la asignación de interfase que ves en la imagen. Con ello los módulos de extensión 2 y 3 son sustituidos por otra interfase inteligente con módulo de extensión en COM2.

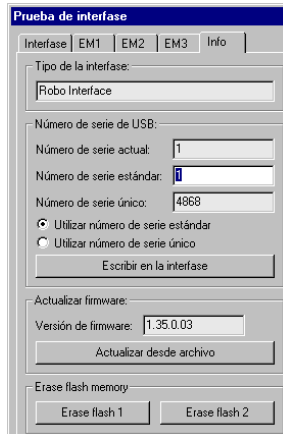


7.5 Cambiar el número de serie de la interfase o la versión de firmware

Por defecto todas las ROBO Interfaces y ROBO I/O-Extensions se suministran con el mismo número de serie. Mientras quieras utilizar solamente una interfase en el ordenador, este sistema resulta más práctico, dado que de esta manera todas las interfases tendrán el mismo aspecto para el ordenador y no habrá problemas a la hora de cambiarlas. Pero si quieres activar más de una interfase en el ordenador por puerto USB, debes cambiar previamente el número de serie de la interfase para que el ordenador pueda distinguir las interfases y dirigirse a ellas. Esto no es necesario, por el contrario, en caso de que te dirijas a las interfases a través de varios puertos en serie.

Para cambiar el número de serie de una interfase, debes proceder como sigue:

- Conecta la interfase **individualmente** en el bus USB del ordenador.
- Pulsa en la barra de herramientas el botón **COM/USB** y selecciona el puerto USB.
- Abre la ventana de prueba de interfase con el botón **Prueba** en la barra de herramientas y cambia a la pestaña **Info**:
- En **Tipo de interfase** se mostrará el tipo de la interfase, por ejemplo **ROBO Interface** o **ROBO I/O Extension**.
- En **Número de serie de USB** puedes configurar el número de serie que utilizará la interfase en el inicio. Cada interfase tiene incorporados dos números de serie, un **número de serie estándar** (que es 1 mientras no configures otra cosa) y un **número de serie único**, que no podrás cambiar y que será distinto en cada interfase. La manera más fácil de utilizar varias interfases en el bus USB pasa por marcar en todas las interfases la opción **Utilizar número de serie único**. Queda entonces garantizado que cada interfase tendrá un número de serie único e inconfundible. Pero si quieres utilizar muchas interfases para un modelo, podría ser muy poco práctico recordar tener todos los números de serie. En este caso es más fácil configurar los números de serie estándar de tu interfase (por ejemplo, 1, 2, 3



etc.) y utilizar estos. Después de que hayas cambiado o seleccionado el número de serie, debes pulsar el botón **Escribir en la interfase**. Tras cualquier modificación en el número de serie, debes desconectar la interfase de la alimentación de corriente y conectarla de nuevo.

Atención: Al cambiar el número de serie es posible que haya que reinstalar el controlador, por lo que necesitarás derechos de administrador si estás trabajando en Windows NT, 2000 o XP. Si cambias un número de serie pero después no puedes reinstalar el controlador porque no tienes derechos de administrador, te será imposible acceder a la interfase a través del USB. Si esto ocurre, desconecta la interfase de la alimentación y vuelve a conectarla mientras pulsas el botón **Port**. La interfase se iniciará entonces con el número de serie 1 y volverá a ser reconocida por el controlador ya instalado. Eso sí: esto no cambia el número de serie permanentemente, de modo que, la próxima vez que se reinicie, el número de serie estará configurado como antes a no ser que vuelvas a pulsar **Port**. Para cambiar el número de serie permanentemente, debes proceder como se explica arriba.

- Finalmente, **Actualizar firmware** te permite actualizar el programa de control interno de tu ROBO Interface en caso de que fischertechnik ofreciera una nueva versión del firmware de Interface.

8 Los elementos de programa

A continuación se describen todos los elementos de programa disponibles en ROBO Pro, estructurados en grupos de elementos y presentados en el orden en el que están ilustrados en la ventana de elementos.

8.1 Elementos básicos (Nivel 1)

8.1.1 Inicio



simultáneamente.

Un proceso de un programa empezará siempre con un elemento de inicio. Si este elemento de programa falta al principio, el proceso no podrá ejecutarse. Si un programa contiene varios procesos, cada uno de ellos deberá empezar con su propio componente de inicio. De ese modo los distintos procesos se iniciarán

Un elemento de inicio no tiene propiedades susceptibles de ser modificadas. Por eso, al contrario de lo que ocurre con la mayoría de los elementos, **no** se abrirá ninguna ventana de propiedades al hacer clic con el botón derecho del ratón en el elemento.

8.1.2 Fin

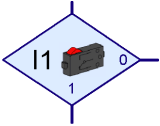


ejecute como bucle sin fin y que no tenga ningún elemento final.

Si quieres salir de un proceso, debes unir la salida del último elemento con el elemento final. También te permite este elemento salir de un proceso en distintos puntos. Existe la posibilidad de unir las salidas de distintos elementos con un único componente final, pero también puede suceder que un proceso se

Un elemento final no tiene propiedades susceptibles de ser modificadas. Por eso **no** se abre, al contrario de lo que pasa con la mayoría de los elementos, ninguna ventana de propiedades al hacer clic con el botón derecho del ratón en el componente.

8.1.3 Bifurcación digital



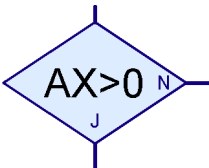
Con esta bifurcación puedes dirigir la ejecución del programa en una de dos direcciones posibles en función del estado de una de las entradas digitales **I1** a **I8**. Si por ejemplo un pulsador está cerrado (en la entrada digital (=1), el programa bifurcará hacia la salida 1. Si por el

contrario la entrada está abierta (=0), el programa bifurcará hacia la salida **0**.

Al hacer clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades:

- En los botones **I1** a **I8** puedes introducir cuál de las entradas de la interfase debe consultarse.
- En **Interfase / Extensión** puedes seleccionar si quieres utilizar una entrada de la interfase o una entrada del módulo de extensión o de una segunda interfase. Encontrarás más información sobre este tema en el capítulo 7 *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Imagen** puedes seleccionar una imagen para el sensor conectado en la entrada. En las entradas digitales se conectan casi siempre pulsadores, pero frecuentemente también fototransistores o contactos Reed.
- En **Cambiar conexiones 1/0** puedes cambiar la posición de las salidas 1 y 0 de la bifurcación. Normalmente la salida 1 se encuentra abajo y la salida 0 a la derecha, pero a menudo resulta más práctico poner la salida 1 a la derecha. Pulsa en **Cambiar conexiones 1/0** y las dos conexiones se intercambiarán en cuanto cierres la ventana con OK.

8.1.4 Bifurcación analógica



Además de las entradas digitales, la ROBO Interface tiene 6 entradas analógicas, 2 entradas de resistencia AX y AY, 2 entradas de tensión A1 y A2 y 2 entradas para sensores de distancia D1 y D2.

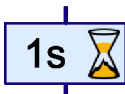
Con esta bifurcación puedes comparar el valor de una entrada analógica con un número fijo y, dependiendo si la comparación devuelve o no una igualdad, bifurcar hacia la salida Sí (S) o No (N).

Al hacer clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades:

- **Entrada analógica** te permite escoger cuál de las entradas de la interfase debe consultarse. Todas las entradas analógicas proporcionan un valor entre 0 y 1023. Para obtener más información sobre las distintas entradas analógicas, consulta la sección 8.7.2 *Entrada analógica*, página 534.

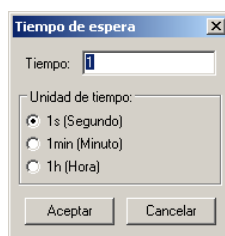
- En **Interfase / Extensión** puedes seleccionar si quieres utilizar una entrada de la interfase o una entrada del módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7 *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Condición** puedes seleccionar un operador de comparación como menor (<) o mayor (>) e introducir el valor de comparación. El valor de comparación debe hallarse entre 0 y 1023. Si inicias un programa con una bifurcación para entradas analógicas en modo online, se mostrará el valor analógico actual.
- En **Cambiar conexiones S/N** puedes cambiar la posición de las salidas 1 y 0 de la bifurcación. Normalmente la salida Sí (S) se encuentra abajo y la salida No (N) a la derecha, pero a menudo resulta más práctico poner la salida Sí a la derecha. Pulsa en **Cambiar conexiones S/N** y las conexiones S y N se cambiarán en cuanto cierres la ventana con OK.

8.1.5 Tiempo de espera



Con el elemento **Tiempo de espera** puedes retardar la posterior ejecución de un programa durante un intervalo de tiempo regulable.

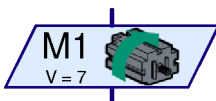
Al hacer clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades. Aquí puedes introducir el tiempo de espera en segundos, minutos u horas. La gama de ajuste para el tiempo de espera abarca desde un milisegundo (una milésima de segundo) hasta 500 horas (casi 3 semanas). Pero cuanto más largo es el tiempo de espera, tanto más imprecisa será la medición de tiempo.



La siguiente tabla indica qué precisión corresponde a cada uno de los distintos tiempos de espera:

Tiempo de espera	Precisión
Hasta 30 segundos	1/1000 segundo
Hasta 5 minutos	1/100 segundo
Hasta 50 minutos	1/10 segundo
Hasta 8,3 horas	1 segundo
Hasta 83 horas	10 segundos
Hasta 500 horas	1 minuto

8.1.6 Salida de motor



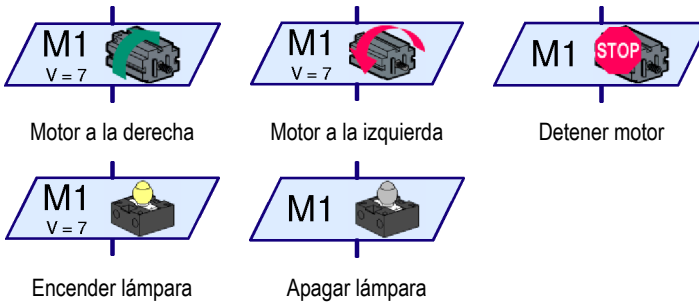
Con el elemento de programa **Salida de motor** se conecta una de las salidas bipolares M1-M4 de la interfase. Las salidas de la interfase pueden utilizarse tanto para motores como para lámparas o electroimanes. En un motor necesitamos poder ajustar no sólo la velocidad, sino también el sentido de giro.

Al hacer clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades:

- En **Salida de motor** puedes ajustar cuál de las cuatro salidas de motor **M1** a **M4** debe utilizarse.
- En **Interfase / Extensión** puedes elegir si quieres utilizar una salida de la interfase o una salida del módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Imagen** puedes seleccionar una imagen que representa el componente de fischertechnik conectado en la salida.
- En **Acción** configuras cómo debe actuarse sobre la salida. Puedes poner en marcha un motor en sentido de giro izquierdo o derecho. Si conectas una lámpara en una salida de motor (ver consejo en salida de lámpara) puedes encenderla o apagarla.
- Finalmente puedes indicar también una **Velocidad** o una **Intensidad** entre 1 y 8. 8 es el valor mayor de velocidad, luminosidad o fuerza magnética; 1 es el valor menor. Durante las funciones de detener o desconectar naturalmente no necesitas indicar una velocidad.

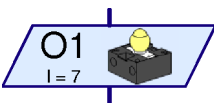


Tienes aquí una lista de los símbolos de algunas acciones e imágenes:



Consejo: A veces un motor se acciona en una única dirección; es el caso, por ejemplo, de una cinta transportadora. Si es así puedes utilizar una salida de lámpara para el motor, lo que te permite ahorrar una conexión.

8.1.7 Salida de lámpara (Nivel 2)



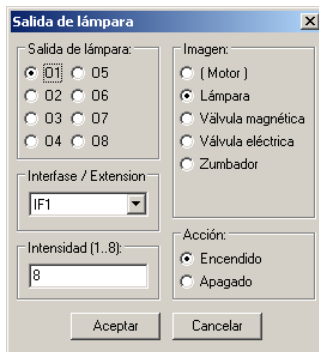
Con el elemento de programa **Salida de lámpara** se conecta una salida unipolar O1-O8 de la interfase. Las salidas de la interfase pueden utilizarse bien en pareja como salidas de motor (ver arriba), bien individualmente como salidas de lámpara O1-O8. Las salidas de lámpara ocupan, al contrario que la salida de motor, solamente una

clavija de conexión. Con ello puedes activar 8 lámparas o válvulas magnéticas por separado. Debes unir la otra conexión de la lámpara con la hembrilla de masa (\perp) de la interfase.

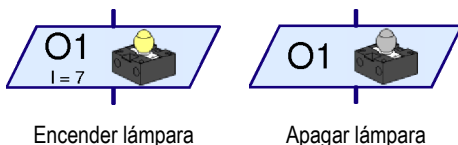
Consejo: Si quieres conectar solamente cuatro lámparas o motores, puedes utilizar una salida de motor también para las lámparas. Resulta más práctico, dado que de esta manera podrás conectar ambas conexiones de la lámpara directamente a la salida de la interfase y no tendrás que unir todos los polos negativos con la hembrilla de masa.

Al hacer clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades:

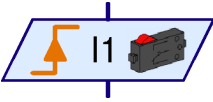
- En **Salida de lámpara** puedes configurar cuál de las ocho salidas **O1** a **O8** debe utilizarse.
- En **Interfase / Extensión** puedes seleccionar si quieres utilizar una salida de la interfase o una salida del módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Imagen** puedes seleccionar una imagen que representa el componente fischertechnik conectado en la salida.
- **Acción** te permite configurar cómo debe actuarse en la salida. Por ejemplo, encender y apagar una lámpara.
- Finalmente puedes indicar también una **Intensidad** entre 1 y 8. 8 es el número mayor para la luminosidad o la fuerza magnética, 1 es el número menor. Si vas a desconectar, naturalmente no necesitarás indicar una intensidad.



Tienes aquí una lista de símbolos de las distintas acciones que se pueden ejercer sobre la lámpara:

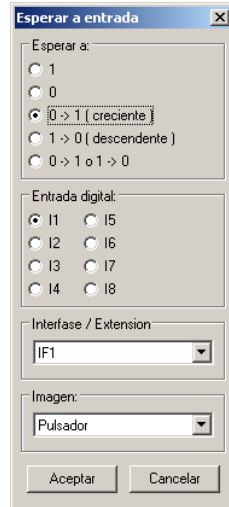


8.1.8 Esperar a entrada



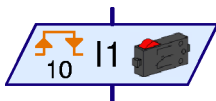
El elemento **Esperar a entrada** esperará hasta que una entrada de la interfase adopte un estado determinado o hasta que varíe de un modo determinado.

Haciendo clic en el elemento con el botón derecho del ratón se mostrará la ventana de propiedades:



- En **Esperar a** puedes seleccionar a qué modificación o a qué estado se debe esperar. Si seleccionas **1** ó **0**, el elemento esperará hasta que la entrada esté cerrada (1) o abierta (0). Si seleccionas **0 -> 1** ó **1 -> 0**, el elemento esperará hasta que el estado de la entrada **cambie** de abierto a cerrado (0->1) o de cerrado a abierto (1->0). Con la última opción el elemento esperará hasta que la entrada cambie, independientemente de que sea de abierta a cerrada o al contrario. En la sección 3.6 *Más elementos de programa*, página 458 se explicará con más detenimiento cómo reconstruir este elemento con el elemento bifurcación.
- En **Entrada digital** puedes configurar cuál de las 8 entradas digitales de **I1** a **I8** debe consultarse.
- En **Interfase / Extensión** puedes seleccionar si quieres utilizar una entrada de la interfase o una entrada del módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Imagen** puedes seleccionar el sensor conectado en la entrada. En las entradas digitales se conectarán casi siempre pulsadores, pero frecuentemente también fototransistores o contactos Reed.

8.1.9 Contador de impulsos



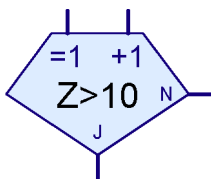
Muchos modelos de robot de Fischertechnik utilizan también las llamadas ruedas dentadas de impulsos. Estas ruedas dentadas accionan un pulsador 4 veces por revolución. Con esas ruedas dentadas de impulsos puede hacerse funcionar un motor durante un tiempo de revoluciones exactamente definido en lugar de un tiempo determinado. Para ello debe contarse el número de los impulsos en una entrada de la interfase. Para este fin existe el elemento **Contador de impulsos**, que espera un número regulable de impulsos.



Haciendo clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades:

- En **Tipo de impulso** puedes seleccionar de qué modo deben contarse los impulsos. Si seleccionas **0 -> 1** (ascendente), el elemento espera hasta que el estado de la entrada haya cambiado de abierto a cerrado (0->1) tantas veces como hayas indicado en **Número de impulsos**. En **1 -> 0** (descendente) el elemento espera hasta que el estado de la entrada haya cambiado de cerrado a abierto tantas veces como se haya indicado. Pero con las ruedas dentadas de impulsos es muy usual utilizar la tercera opción: el elemento cuenta tanto los cambios **0 -> 1** como los cambios **1 -> 0**, de modo que se cuentan 8 impulsos por cada giro de una rueda dentada de impulsos.
- En **Entrada digital** puedes configurar cuál de las 8 entradas digitales de **I1** a **I8** debe consultarse.
- En **Interfase / Extensión** puedes seleccionar si quieres utilizar una entrada de la interfase o una entrada del módulo de extensión o una segunda interfase. Encontrarás más información en el capítulo 7 *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Imagen** puedes seleccionar una imagen para el sensor conectado en la entrada. Para entradas digitales se utilizan casi siempre pulsadores, pero frecuentemente también fototransistores o contactos Reed.

8.1.10 Bucle contador

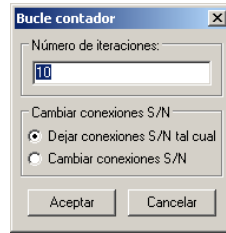


El elemento **Bucle contador** te permite ejecutar varias veces una determinada parte de un programa. El elemento lleva un contador incorporado. Si se accede al bucle contador a través de la entrada **=1**, el contador se pondrá en 1. Si por el contrario se accede al bucle contador a través de la entrada **+1**, el contador aumentará en 1. Dependiendo si el valor del contador es o no es mayor que el valor fijado por ti, el bucle contador bifurcará hacia la salida Sí (**S**) o hacia No (**N**) respectivamente. Encontrarás un ejemplo en la sección 3.6.4

Bucle contador, página 460

Haciendo clic con el botón derecho del ratón en el elemento se mostrará la ventana de propiedades:

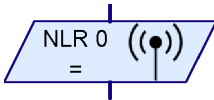
- En **Número de ciclos** debes indicar cuántas veces debe salirse del bucle contador a través de la salida No (N) antes de activarse la salida Sí (S). El valor indicado debe ser positivo.
- Haciendo clic en **Cambiar conexiones S/N**, se cambiarán las conexiones de **S** y **N** en cuanto hayas cerrado la ventana con OK. Dependiendo el sitio de la conexión **S** o **N**, la parte del programa que se repetirá estará a la derecha o debajo del bucle contador.



8.2 Enviar y recibir (Niveles 2-4)

En este grupo de elementos encontrarás elementos de programa que puedes utilizar para enviar recibir mensajes por ROBO RF Data Link o el puerto serie de la ROBO Interface.

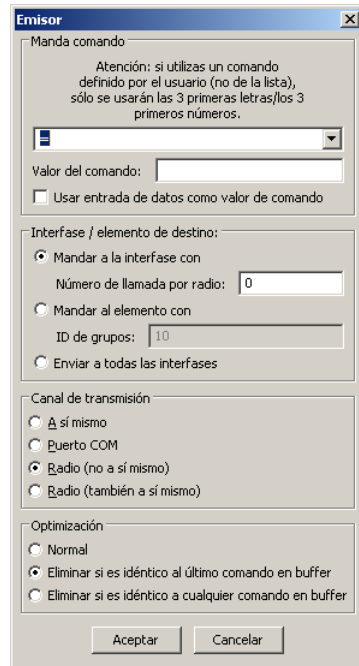
8.2.1 Emisor (Nivel 2)



Con el emisor puedes enviar un comando (o en general un mensaje) a otra interfase por medio del ROBO RF Data Link. Así, por ejemplo, puedes hacer que varios robots se comuniquen entre sí. Encontrarás una introducción al tema y un ejemplo detallado en la sección 4.5 *Tango 2: comunicación a través de ROBO RF Data Link* (página 478).

Ventana de propiedades del elemento de emisión

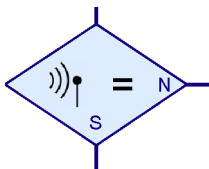
- En **Manda comando** puedes indicar el comando y, a partir del Nivel 3, su valor. Los comandos constan de un nombre y un valor de comando. El valor de comando puede estar determinado por una entrada de datos. Para los nombres de comando que no se hayan seleccionado desde el menú sólo se tienen en cuenta las primeras 3 letras o los primeros 3 números. Si quieres puedes poner más de tres letras, pero "hola", "holita" y "Holanda", por ejemplo, darán los tres el mismo mensaje, porque empiezan todos por "hol". Las mayúsculas y minúsculas y los caracteres especiales (espacio en blanco, signos de admiración e interrogación y similares) tampoco se distinguen. XY! y XY? se entenderán como el mismo mensaje. Los números sí que se distinguen, de forma que XY1 no se confunde con XY2.
- En **Interfase / elemento de destino** puedes seleccionar a qué interfase o elemento de programa ha de enviarse el comando. En la mayoría de los casos querrás enviar el comando a una interfase con un **número de llamada por radio** en concreto o a **todas las interfases**.



A partir del Nivel 4 existe además la posibilidad de utilizar como dirección de destinatario un **grupo** de entre 10 y 255. Los comandos a un grupo no se envían a una interfase en concreto con un número de llamada por radio específico, sino a los elementos de recepción en los que se haya indicado el mismo número de grupo. De este modo puedes, por ejemplo, diferenciar a quién se le manda un mensaje a base de utilizar un grupo distinto para cada emisor. Los números de grupo empiezan en 10, porque los números de 0 a 9 ya están reservados. Sobre los números de grupo reservados sabrás más al hablar del elemento de recepción.

- En **Canal de transmisión** configuras cómo se ha de enviar el mensaje, desde un punto de vista puramente técnico. En el Nivel 2 esta opción no está disponible, porque se transmite siempre por **Radio (no a sí mismo)**. Por radio quiere decir que el mensaje se manda por un RF Data Link. Si se selecciona **Radio (también a sí mismo)**, el mensaje se envía también a la interfase que ha emitido el mensaje. Para que funcione, en **Interfase / elemento de destino** tiene que estar seleccionado un destinatario con el que la interfase emisora quiera conectar, por ejemplo, **todas las demás interfaces** o un **grupo** de ellas. También puedes hacer que un mensaje se envíe únicamente a la interfase emisora (**A sí mismo**). Esta función es útil, por ejemplo, para comunicar distintos procesos entre sí. A partir del Nivel 4 el emisor también soporta el envío de comandos por el **puerto COM** de la ROBO Interface. Para ello debes conectar dos interfaces con un cable null modem.
- En **optimización** (a partir del Nivel 4) puedes configurar si los comandos idénticos se deben enviar o no por repetido. Con muchos comandos no importa que se manden una sola vez o varias. Sin optimización, por ejemplo, un comando que se envía varias veces seguidas en un bucle puede llenar el buffer de transmisión del RF Data Link, impidiendo que se envíen los demás comandos pendientes. Por esta razón es interesante eliminar los comandos idénticos. Por norma general sólo interesa borrar un comando si el **último comando del buffer es idéntico**. Si en este modo envías, por poner un ejemplo, 2 comandos de inicio y a continuación otros 2 de parada, sólo se transmitirán uno de cada. Sin embargo, si mandas inicio, parada, inicio y parada, es decir, los mismos comandos repetidos pero intercalados con otros, se transmitirán tal cual, sin eliminar nada. No obstante también puedes indicar, si quieres, que se borren los comandos cuando coinciden con otros comandos que ya están en el buffer, independientemente de su posición. Con algunos comandos la optimización no presenta ventajas, de manera que conviene usar el modo **normal**. Encontramos un ejemplo en el comando de **Añadir**, con el que se pueden añadir elementos a una lista: a fin de cuentas, en una relación sí que es importante saber si un ítem está presente 1 ó 2 veces. En el Nivel 2 está siempre seleccionada la opción **Eliminar si es idéntico al último comando en buffer**.

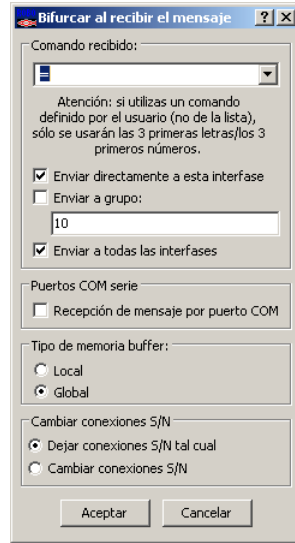
8.2.2 Receptor / Bifurcación al recibir el comando (Nivel 2)



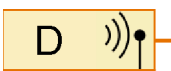
Este elemento es el contrario del elemento de emisión anterior. En función de si un determinado comando ha sido o no recibido, el elemento pasa a la salida S o a la salida N.

Ventana de propiedades del elemento de recepción

- En **Comando recibido** se indica el comando que debe recibir el receptor. Como ya se ha explicado al hablar del emisor, solamente se tendrán en cuenta las 3 primeras letras o números. A continuación debes escoger si el receptor ha de reaccionar solamente a los comandos que se envíen directamente a la interfase (es decir, con un número de llamada por radio específico) o a todas las órdenes que se envíen a todas las interfases. Puedes escoger las dos opciones al mismo tiempo. Como ya se ha explicado al hablar del emisor, a partir del Nivel 4 también se puede hacer el envío a un grupo determinado. Se trata de mensajes que han de ser recibidos por todos los elementos de recepción a los que se ha asociado con el grupo. Pueden usarse grupos de 10 a 255. Los grupos 0 a 8 corresponden a los números de llamada por radio 0 a 8. El grupo 9 está reservado para el envío a todas las interfases. Al enviar mensajes no importa si se mandan al grupo 1 o al número de llamada por radio 1. Al recibir, en cambio, no se puede indicar ningún número de llamada por radio, porque cada interfase tiene el suyo. Al indicar en el destinatario un grupo de recepción de 1 a 8, si pueden recibirse mensajes que en realidad se idearon para otra interfase. Los grupos de recepción 10 o menos no pueden utilizarse hasta el Nivel 5.
- En **Puertos COM serie** (a partir del Nivel 4) puedes indicar que el elemento también reciba mensajes del puerto COM. Hablamos de una configuración global que activa o desactiva en su totalidad el puerto COM. Si en un programa un emisor o receptor en concreto utiliza el puerto COM, todos los elementos de recepción podrán recibir mensajes del puerto COM.
- En **Cambiar conexiones S/N** puedes cambiar la posición de las salidas S y N de la bifurcación. Normalmente la salida Sí (S) se encuentra abajo y la salida No (N) a la derecha, pero a menudo resulta más práctico poner la salida Sí a la derecha. Presiona en **Cambiar conexiones S/N** y verás que se intercambian las salidas S y N en cuando salgas de la ventana pulsando Aceptar.



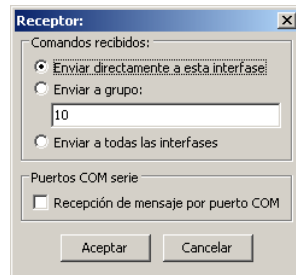
8.2.3 Receptor (Nivel 3)



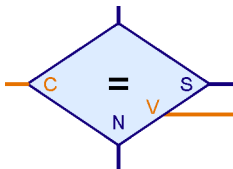
El elemento de recepción descrito en la sección anterior está concebido primordialmente para el Nivel 2, puesto que puede recibir comandos, pero no valores de comando. El elemento de recepción de Nivel 3 recibe comandos con valor. En este elemento no tienes que indicar un comando que quieras que llegue al elemento de recepción, puesto que lo que hace es enviar simplemente todos los comandos recibidos al elemento conectado a la salida.

Ventana de propiedades del elemento de recepción

- En **Comando recibido** escoges si el receptor ha de reaccionar solamente a los comandos que se envíen directamente a la interfase (es decir, con un número de llamada por radio específico) o a todas las órdenes que se envíen a todas las interfaces. A partir del Nivel 4 también puedes seleccionar un grupo determinado. Sobre grupos de recepción te contamos más en las secciones anteriores *Emisor (Nivel 2)* y *Receptor / Bifurcación al recibir el comando (Nivel 2)*. A diferencia de lo que ocurre con el receptor de Nivel 2, en el Nivel 3 solamente puedes escoger una opción. Lo que sí puedes hacer es conmutar las salidas de dos o más elementos de recepción con distintas opciones, si quieres que los comandos se reciban con distintos destinatarios. En especial puedes cambiar receptores con distintos grupos.
- En **Puertos COM serie** (a partir del Nivel 4) puedes indicar que el elemento también reciba mensajes del puerto COM. Sobre esto encontrarás información en la sección anterior, *Receptor / Bifurcación al recibir el comando (Nivel 2)*.



8.2.4 Esperar comando (Nivel 4)

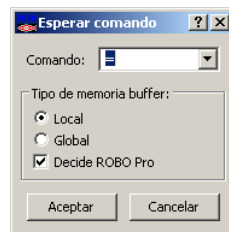


El elemento "Esperar comando" se utiliza igual que elemento *Receptor / Bifurcación al recibir el comando (Nivel 2)* para esperar a un comando. Lo que ocurre es que no espera comandos que se envíen por ROBO RF Data Link o por otros puertos, sino a comandos que se envían a la entrada de comando por el lado izquierdo. Cuando se conecta en él un elemento *Receptor (Nivel 3)*, se obtiene un receptor de Nivel 2. De todos modos este elemento tiene además

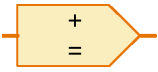
una salida de datos en el lado derecho. Cada vez que se recibe un comando y el flujo de programa se dirige a la salida S, en la salida de valor de comando E está disponible el valor numérico enviado junto con el comando. Como las salidas E y S van juntas, en este elemento no se pueden cambiar las salidas S y N. Encontrarás un ejemplo en la sección 6.1 (página 501).

Ventana de propiedades del elemento "Esperar comando"

- En **Comando** seleccionas el comando al que debe esperar el elemento. También puedes indicar un comando propio, pero recuerda que sólo se tendrán en cuenta las 3 primeras letras o números. Consulta la descripción de **Manda comando** que encontrarás en la sección 8.2.1 *Emisor (Nivel 2)* de la página 517.
- La configuración **Dimensiones del buffer** de comando no se muestra hasta el Nivel 5. Como el receptor de Nivel 2, este elemento guarda los comandos que van entrando. Puesto que el elemento de Esperar comando debe acumular tanto el comando como su valor, el número máximo de comandos guardables está limitado. Para la mayoría de las aplicaciones llega de sobra el valor estándar de 4 comandos, puesto que la mayoría de los programas son capaces de procesar con esta agilidad.



8.2.5 Filtro de comandos (Nivel 4)

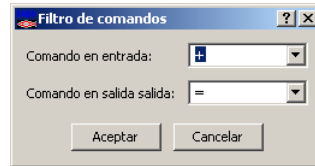


Con el filtro de comandos pueden etiquetarse comandos. Si a la entrada izquierda se envía un comando determinado, el elemento envía al elemento conectado a la salida derecha otro comando, pero con el mismo valor que el recibido en la entrada. De este modo puede hacerse a partir de un comando

=, por ejemplo, un comando de control de motor como derecha o izquierda. Encontrarás un ejemplo en la sección 6.2 (página 502).

Ventana de propiedades del elemento de filtro de comandos

El elemento de filtro de comandos sólo tiene dos configuraciones: el comando que espera en la entrada y el comando en el que lo transforma antes de enviarlo a la salida. También puedes indicar un comando propio, pero recuerda que sólo se tendrán en cuenta las 3 primeras letras o números. Consulta la descripción de **Manda comando** que encontrarás en la sección 8.2.1 *Emisor (Nivel 2)* de la página 517.



8.3 Subprograma I/O (Nivel 2-3)

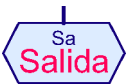
En este grupo de elementos encontrarás elementos de programa que necesitarás únicamente para generar subprogramas.

8.3.1 Entrada de subprograma (Nivel 2)



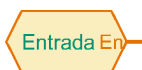
Un subprograma puede tener una o varias entradas de subprograma. Por medio de estas entradas, el programa principal o un subprograma de orden superior transfiere la ejecución del proceso al subprograma. En el símbolo verde del subprograma que se introduce en el programa de orden superior, habrá una clavija de conexión para cada entrada de subprograma en la parte superior. Las conexiones del símbolo tienen el mismo orden (de izquierda a derecha) que las entradas de subprograma en el plano de funciones del subprograma. Al hacer clic con el botón derecho del ratón en el elemento, se mostrará la ventana de propiedades. En ella puedes darle a la entrada un nombre que después verás en el símbolo. Encontrarás más información sobre los subprogramas en el capítulo 4: *¡Socorro, me estoy perdiendo! - Trabajar con subprogramas*, página 464.

8.3.2 Salida de subprograma (Nivel 2)



Un subprograma puede tener una o varias salidas de subprograma. Por medio de estas salidas el subprograma devuelve la ejecución del programa al programa principal o al subprograma de orden superior. En el símbolo verde del subprograma introducido en el programa de orden superior, habrá una clavija de conexión para cada salida de subprograma en la parte inferior. Las conexiones del símbolo tienen el mismo orden (de izquierda a derecha) que las salidas de subprograma del diagrama de operaciones del subprograma. Al hacer clic con el botón derecho del ratón en el elemento, se mostrará la ventana de propiedades. En ella podrás darle a la salida un nombre que después verás en el elemento. Encontrarás más información sobre subprogramas en el capítulo 4: *¡Socorro, me estoy perdiendo! - Trabajar con subprogramas* página 464.

8.3.3 Entrada de comando de subprograma (Nivel 3)



Mediante este elemento pueden unirse los subprogramas con elementos de entrada (como por ejemplo interruptores del programa principal o del subprograma de orden superior) o suministrarles valores de elementos de variables (por ejemplo coordenadas). En el símbolo verde del subprograma que es introducido en el programa de orden superior, habrá una clavija de conexión para cada entrada de comando de subprograma en la parte izquierda. Las conexiones del símbolo tienen el mismo orden (de arriba a abajo) que las entradas de comando de subprograma del subprograma. Al hacer clic con el botón derecho del ratón en el elemento, se mostrará la ventana de propiedades. En ella puedes darle a la salida un nombre que después verás en el elemento. La utilización de este elemento se explica detalladamente en la sección 5.4 *Entrada de datos para subprogramas*, página 491.

8.3.4 Salida de comando de subprograma (Nivel 3)

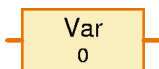


Mediante este elemento de programa pueden enviarse comandos (como por ejemplo izquierda, derecha, detener) a motores u otros elementos de salida que se encuentren en el programa principal o en el subprograma de orden superior. En el símbolo verde del subprograma introducido en el programa de orden superior, habrá una clavija de conexión para cada salida de comando de subprograma en la parte derecha. Las conexiones del símbolo tienen el mismo orden (de arriba a abajo) que las salidas de comando de subprograma del subprograma. Al hacer clic con el botón derecho del ratón en el elemento, se mostrará la ventana de propiedades. En ella puedes darle a la salida un nombre que después verás en el elemento. La utilización de este elemento se explica detalladamente en la sección 5.4 *Entrada de datos para subprogramas*, página 491.

8.4 Variable, Listado, ... (Nivel 3)

Los elementos de programa de este grupo pueden almacenar uno o más valores numéricos. Con ello pueden desarrollarse programas con capacidad de recordar.

8.4.1 Variable (global)



Una variable puede almacenar un valor numérico único entre -32767 y 32767. El valor de la variable se establece conectando en la entrada de comando de la parte izquierda un elemento de comando = (ver sección 8.5.1 = (*Asignar*) página 528). También puede dársele un valor inicial a una variable desde la ventana de propiedades; la variable mantendrá ese valor hasta que obtenga el primer comando que le indique que lo cambie.

ROBO Pro establece una única variable para todos los elementos de variable con el **mismo nombre** y **tipo de variable = global**. Todas las variables globales con el mismo nombre son idénticas y tienen siempre el mismo valor, aunque aparezcan en distintos subprogramas. Si se cambia uno de estos elementos de variable con un comando, cambiarán también todos los otros elementos de variable globales con el mismo nombre. También existen variables locales (ver próxima sección) donde no es así.

Además del comando =, una variable entiende también los comandos + y -. Si una variable recibe el comando + 5, por ejemplo, sumará el valor 5 a su valor actual. Con el comando - se restará del valor actual de la variable el valor transmitido mediante el comando.

Atención:

Si después de obedecer un comando + ó – el valor de la variable se sale del rango de valores de la variable, se le sumará o se le restará 65.536 para que el valor vuelva a ser válido. Dado que este comportamiento por regla general no es el deseado, deberías tener cuidado de que esto no suceda.

Cada vez que cambia el valor de la variable, ella envía un comando = con el nuevo valor a todos los elementos que estén conectados en la salida de comando de la variable. Si quieres observar el valor de una variable, puedes conectar en la salida de la variable un campo de mensaje para que se muestre en el panel de control (ver sección 8.7.6 *Entrada del panel de control* página 538).

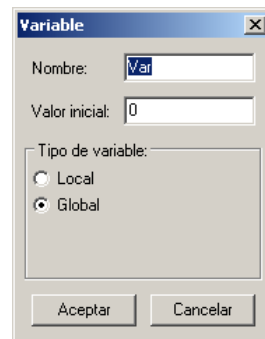
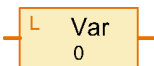
Aquí tienes un resumen de todos los comandos que pueden modificar el elemento de variable:

Comando	Valor	Acción
=	-32767 a 32767	Iguala el valor de la variable al valor transmitido mediante el comando.
+	-32767 a 32767	Suma al valor actual de la variable el valor transmitido mediante el comando.
-	-32767 a 32767	Resta del valor actual de la variable el valor transmitido mediante el comando.

Que el rango de valores de la variable se encuentre entre dos números tan poco redondos como -32.767 y 32.767 se debe a que el ordenador cuenta en sistema binario y no en sistema decimal, como hacemos nosotros. En el sistema binario el número 32.767 es un número más "lógico" para ser fijado como límite, algo así como si fuese 9.999 en el sistema decimal. Pero nosotros podemos despreocuparnos: el ordenador se encarga de convertir todos los números del sistema binario al sistema decimal. Solamente tenemos conciencia de que usa un sistema de cálculo diferente al ver los valores máximos de las variables y si durante el procesamiento se produce un desbordamiento.

Ventana de propiedades para variables

- En **Nombre** puedes introducir un nombre para la variable.
- En **Valor inicial** puedes introducir el valor inicial de la variable. La variable mantiene este valor hasta que recibe un nuevo valor mediante el comando =, + ó –.
- El punto **Tipo de variable** es interesante solamente para variables de subprogramas y se explicará más detalladamente en la siguiente sección, Variable local. Con las variables de programa principal, ambas configuraciones tienen el mismo efecto.

**8.4.2 Variable local**

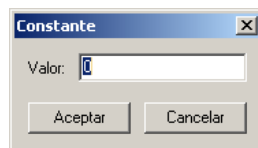
Todos los elementos de variable **globales** con el mismo nombre utilizan la misma variable y tienen siempre el mismo valor. Probablemente esto es lo que esperabas y también lo que en general resulta más práctico. Pero si utilizas variables en subprogramas, este sistema podría provocar grandes problemas. Si tu programa tiene varios procesos paralelos, un subprograma puede ejecutarse

varias veces durante un período de tiempo. Tal situación lleva generalmente a confusión si el subprograma utiliza en todos los procesos las mismas variables. Por esta razón existen las **Variables locales**. Una variable local se comporta de manera casi idéntica a una variable global, con una diferencia: la variable local es válida solamente en el subprograma para el cual esté definida. Incluso si dos variables locales de distintos subprogramas tienen el mismo nombre, seguirán siendo variables distintas e independientes. Aunque se ejecute un subprograma de varios procesos varias veces en paralelo, el subprograma tendrá en cada proceso una parte independiente con sus variables locales. Las variables locales existen solamente mientras se ejecute el subprograma en el cual estén definidas. Por eso el valor inicial es asignado a las variables locales en cada inicio del subprograma correspondiente y no durante el inicio del programa. Dado que por regla general un subprograma debe hacer siempre lo mismo cada vez que lo llamen a ejecución, es mucho más práctico poner las variables al valor inicial en cada llamada. Las variables locales, por decirlo así, no tienen memoria de llamadas anteriores del mismo subprograma.


En el programa principal las variables locales y las variables globales se comportan exactamente igual, dado que el programa completo y el programa principal se inician simultáneamente. Las variables locales, eso sí, son algo más eficaces en la ejecución del programa. En cambio, es preferible definir los elementos de listado globalmente, porque el área de memoria para variables globales es mayor que para variables locales.

8.4.3 Constante

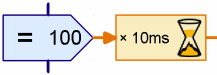
0 — Al igual que una variable, una constante tiene un valor, pero este valor **no** puede cambiarse mediante el programa. Puedes, por ejemplo, unir una constante con una entrada de datos de un símbolo del subprograma si el subprograma debe utilizar siempre el mismo valor. Las constantes también resultan muy prácticas al trabajar con operadores. Encontrarás un ejemplo ilustrativo al final de la sección 5.7 *Operadores*, página 497.



8.4.4 Variable de temporizador

x 10ms  — Una variable de temporizador se comporta básicamente igual que una variable. En las variables de temporizador también existe la distinción entre variables normales y variables estáticas. La única diferencia es que una variable de temporizador va bajando el valor memorizado según un ritmo de intervalos fijos hasta llegar a 0. En cuanto el valor de temporizador es 0, se detiene la variable de temporizador. Si el valor de temporizador se convierte en negativo mediante un comando de resta, el valor en el próximo ciclo será de nuevo 0.

Puedes configurar la velocidad con la cual la variable de temporizador cuenta hacia atrás entre un milisegundo por paso y 1 minuto por paso desde la ventana de propiedades. Pero deberías tener en cuenta que la exactitud del temporizador depende de los ciclos configurados. Si por ejemplo pones un temporizador a 1×10 segundos, el próximo paso de temporizador de 10 segundos puede efectuarse un poco más tarde (un poquito después del segundo, por ejemplo) o hasta que hayan pasado 10 segundos. La exactitud de los temporizadores es siempre tan exacta como los ciclos configurados. Por eso deberías seleccionar ciclos cortos, por ejemplo 10×1 segundos ó $100 \times 0,1$ segundos en vez de 1×10 segundos. No selecciones un ciclo de un minuto a no ser que el programa deba esperar como mínimo una hora. Entonces no importarán un par de minutos arriba o abajo.



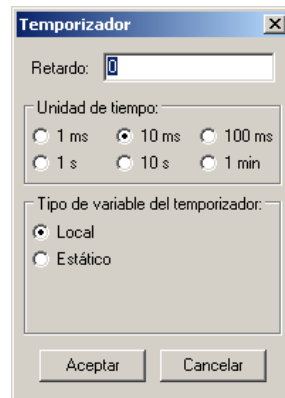
El número de pasos que se contarán hacia atrás es asignado a la variable de temporizador generalmente a través de un comando = por parte de un elemento de comando. En el ejemplo ilustrado se cuentan hacia atrás 100 pasos por cada 10ms. Esto corresponde a una duración de 1000 ms=1 segundo. La exactitud es de 10ms.

Con las variables de temporizador pueden solucionarse de una manera fácil también los problemas de medición de tiempo y de retardo. Si un robot por ejemplo debe cancelar una búsqueda después de 20 segundos, puedes poner al inicio de la búsqueda una variable de temporizador de 20 x 1 segundo (ó 200 x 0,1 segundos) y después consultar en el programa de búsqueda si el valor de temporizador es todavía mayor que 0. También puedes poner el temporizador de nuevo en el valor inicial en caso de éxitos parciales de la búsqueda.

Si quieres medir un tiempo, debes ajustar la variable de temporizador al principio a un valor positivo lo más alto posible (30.000 ó 32.767) para que tarde mucho tiempo en llegar a 0. Si quieres saber cuánto tiempo habrá transcurrido entonces, debes restar el valor actual del temporizador del valor inicial.

Ventana de propiedades para variables de temporizador

- En **Retardo** puedes fijar el valor inicial para la variable de temporizador. Normalmente se introduce aquí 0 y se pone el valor de la variable de temporizador con un comando = al momento adecuado. Pero si el temporizador debe ponerse en marcha durante el inicio del programa o de un subprograma, puedes introducir aquí el valor correspondiente.
- En **Unidad de tiempo** puedes seleccionar el tamaño de los ciclos en los que la variable de temporizador va bajando hacia atrás.
- En **Tipo de variable de temporizador** puedes configurar si el temporizador es una variable global o una variable local (ver sección 8.4.2 *Variable local*, página 523).






8.4.5 Listado



El elemento **Listado** corresponde a una variable que puede memorizar no solamente un valor, sino varios. El número máximo de valores que pueden guardarse en un listado se fija en la ventana de propiedades del elemento.

Puedes añadir valores en la parte inicial del listado o eliminarlos de la parte final. Además puedes cambiar o leer cualquier valor del listado e intercambiar cualquier valor del listado con el valor de la primera posición. No se puede insertar o borrar directamente un valor a la mitad o al principio del listado, pero se puede escribir un subprograma que ejecute esta función.

Se utilizan las siguientes funciones de listado enviando comandos al elemento en la entrada **ES** (de escribir). Pueden enviarse los siguientes comandos a la entrada **ES**:

Comando	Valor	Acción
Añadir 	-32767 a 32767	Añade el valor transmitido con el comando al final del listado. El listado aumentará en un elemento. Si el listado ya tiene el tamaño máximo, se hará caso omiso del comando.
Eliminar 	0 a 32767	Borra el número indicado de elementos al final del listado. El valor transmitido con el comando es el número de los elementos que se desean borrar. Si el número es mayor que el número de los elementos en el listado, se borrarán todos los elementos del listado. Si el número es 0 ó negativo, se hará caso omiso del comando.
Intercambiar 	0 a 32767	Intercambia el elemento indicado con el primer elemento del listado. El valor transmitido con el comando es el número de elementos que se desea intercambiar.

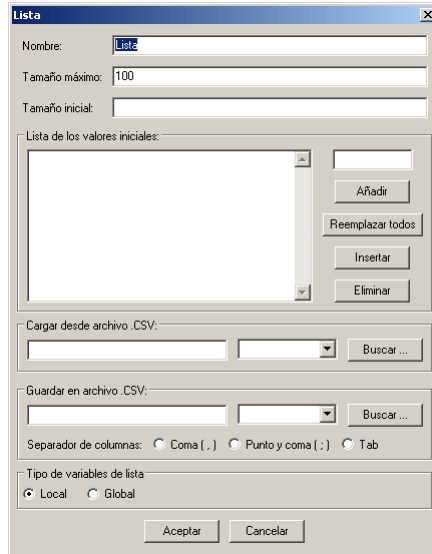
A través de la entrada **I** (de índice) puede seleccionarse un determinado elemento del listado. Para ello envías un comando = a la entrada **I** con el número deseado de elementos. El primer elemento tendrá el número 0. Puedes asignar un nuevo valor al elemento seleccionado mediante la entrada **I** enviando el nuevo valor deseado con el comando = a la entrada **ES**.

El elemento seleccionado mediante de la entrada **I** puede consultarse mediante la salida **LR** (de lectura de retorno). Si se cambiara la entrada **I** o el valor de la entrada de datos seleccionada a través de la entrada **I**, el listado devolverá el valor actual de la entrada de datos seleccionada a los elementos conectados en la salida **LR**.

A través de la salida **I** puede consultarse si el índice creado en la entrada **I** será válido. Si **N** es el número de los elementos, debe crearse en la entrada **I** un valor entre 0 y **N**-1. Si eso es el caso, la salida **I** enviará un comando = con valor **N**; si no es así, enviará el comando con valor 0 a todos los elementos conectados.

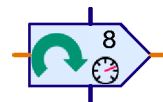
Ventana de propiedades para listados

- En **Tamaño máximo** puedes introducir el número máximo de elementos del listado. Este tamaño no puede ser sobrepasado por los comandos **Añadir**.
- En **Tamaño inicial** se introduce el número de elementos con los que el listado debe estar ocupado previamente al inicio.
- En **Listado de valores iniciales** puedes introducir los valores iniciales con los cuales el listado debe estar ocupado previamente. Con los botones a la derecha (junto al listado) puedes editar el listado.
- En **Cargar desde archivo .CSV** puedes seleccionar un archivo .CSV compatible con Excel desde el cual el listado tomará sus valores. En el campo central de selección puedes seleccionar la columna del archivo .CSV pertinente. El archivo se carga inmediatamente y se muestra en **Listado de valores iniciales**. Si inicias el programa o ejecutas una descarga, ROBO Pro intentará de volver a cargar los valores actuales desde el archivo. Si no lo consigue, se utilizarán los valores guardados en el listado de valores iniciales.
- En **Guardar en archivo .CSV** puedes indicar un archivo en el cual debe guardarse el contenido del listado después de finalizar el programa. Solamente funcionará, sin embargo, en modo online y solamente para listados estáticos (ver próximo punto). El contenido del listado se escribe en la columna del archivo seleccionada. En **Carácter de separación de columna** puedes seleccionar si debe separarse cada columna del listado mediante coma o punto y coma. En los países en los que se escribe 0.5 con punto, se utiliza normalmente una coma como carácter de separación de columna. Dado que en España 0,5 se escribe con coma, se utiliza también frecuentemente el punto y coma como carácter de separación de columna. Si surgen problemas durante la importación de un archivo CSV de ROBO Pro CSV, por ejemplo en Microsoft Excel, intenta utilizar otro carácter de separación de columnas.
- En **Tipo de variable de listado** puedes configurar si el listado temporizador es una variable global o una variable local (ver sección 8.4.2 *Variable local*, página 523). Para listados grandes (tamaño máximo más de 100 elementos) es recomendable el tipo **Global**, porque se ofrece más memoria para variables globales que para variables locales.



8.5 Instrucciones (Nivel 3)

Todos los elementos de programa de este grupo son elementos de comando. De acuerdo con su aplicación, también se denominan elementos de mensaje. Cuando se ejecuta un elemento de comando (esto es, cuando el flujo del programa se introduce en la entrada azul del elemento), el elemento de comando envía un comando o un mensaje al elemento conectado en la salida de la derecha. Los distintos comandos (como derecha, izquierda o detener) tienen distintos efectos sobre el elemento conectado. Generalmente los elementos conectados entienden un

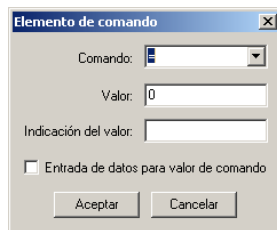


Generalmente los elementos conectados entienden un

repertorio muy limitado de comandos, por eso en los distintos elementos de programa hay un listado de las instrucciones que entiende cada uno y de los efectos que ejercen sobre ellos. La mayoría de los comandos van además acompañados por un valor. En un comando como puede ser **Derecha**, por ejemplo, se indica además una velocidad entre 1 y 8. El comando **Detener**, por el contrario, no tiene un valor añadido.

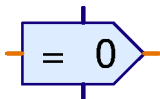
Ventana de propiedades para elementos de comando

- En **Comando** puedes seleccionar el comando deseado en un listado que muestra todos los comandos posibles.
- En **Valor** se indica el valor numérico que debe transmitirse con el comando. Si no debe transmitirse ningún valor, este campo quedará vacío.
- En **Denominación de valor** puedes introducir un corto texto indicativo que se mostrará junto con el valor en el elemento de comando (por ejemplo, X= ó T=). La indicación se concibe como una aclaración del tipo de valor de que se trata, pero es algo únicamente informativo y no tiene función alguna.



En **Entrada de datos para valor de comando** puedes indicar si el elemento de comando debe tener en su parte izquierda una entrada de datos naranja para el valor que se va a transmitir. En todos los elementos de comando puede introducirse el valor directamente en el elemento de comando o hacer que se lea a través de una entrada de datos colocada en la parte izquierda del elemento de comando. De este modo puede accionarse, por ejemplo, el motor de un círculo de regulación con una velocidad modificable.

8.5.1 = (Asignar)

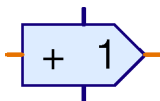


El comando = asigna un valor al receptor. Generalmente se utiliza para asignar un valor a variables, variables de temporizador, elementos de listado o salidas de panel de control.

Pero el comando = no es enviado solamente por elementos de comando, sino por todos los elementos de programa con salidas de datos. Todos los elementos envían comandos = si se cambia el valor de una salida. Por ejemplo, un elemento de entrada digital envía un comando = 1 cuando se cierra un pulsador de la entrada, y un comando = 0 cuando se abre el pulsador. Pero en estos casos no se utiliza un elemento de comando; en realidad, podría decirse que estos elementos de programa con salidas de datos ya llevan incorporados los elementos de comando =.

Todas las entradas de datos de elementos de programa ROBO Pro pueden editar como mínimo el comando =. Así que el comando = es el comando más utilizado en ROBO Pro.

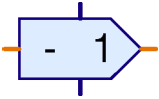
8.5.2 + (Más)



El comando + se envía a variables o variables de temporizador para aumentar el valor de la variable. Con el comando + puede transmitirse cualquier valor, que se añadirá a la variable. Dado que el valor transmitido con el comando también puede ser negativo, el valor comando también sirve para disminuir la variable. Ver sección 8.4.1 *Variable*, página 522 y sección

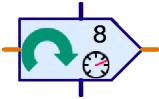
8.4.4 *Variable de temporizador*, página 524.

8.5.3 – (Menos)



El comando **-** se utiliza de forma parecida al comando **+** que se acaba de describir. La única diferencia es que el valor transmitido con el comando se resta del valor de la variable.

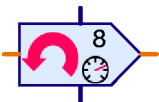
8.5.4 Derecha



El comando **Derecha** se envía a elementos de salida de motor para conectar el motor con sentido de giro a la derecha. Ver sección 8.7.4 *Salida de motor*, página 536.

El valor es una velocidad de 1 a 8.

8.5.5 Izquierda



El comando **Izquierda** se envía a elementos de salida de motor para conectar el motor con sentido de giro a la izquierda. Ver sección 8.7.4 *Salida de motor*, página 536.

El valor es una velocidad de 1 a 8.

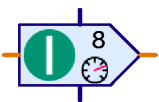
8.5.6 Detener



El comando **Detener** se envía a elementos de salida de motor para detener el motor. Ver sección 8.7.4 *Salida de motor*, página 536.

Con el comando **detener** no se transmite valor alguno.

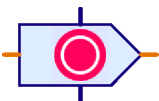
8.5.7 ON



El comando **ON** se envía a elementos de salida de lámpara para encenderla. Ver sección 8.7.5 *Salida de lámpara*, página 537. Un comando **ON** también puede enviarse a elementos de salida de motor. Corresponde al comando **derecha**. Para motores, sin embargo, sería mejor utilizar el comando **derecha**, dado que así el sentido de giro es directamente reconocible.

El valor es la luminosidad o intensidad de la luz, de 1 a 8.

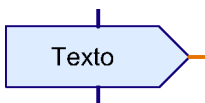
8.5.8 OFF



El comando **OFF** se envía a elementos de salida de lámpara para apagarla. Ver sección 8.7.5 *Salida de lámpara*, página 537. El comando **OFF** puede también enviarse a elementos de salida de motor. Corresponde al elemento **detener**.

Con el comando **OFF** no se transmite valor alguno.

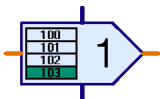
8.5.9 Texto



El comando **Texto** es un comando especial, dado que no envía ninguna instrucción con ningún número, sino un texto, al elemento conectado. Pero solamente hay un elemento de programa que puede trabajar con el comando texto: un mensaje de texto en un panel de control. Encontrarás más información en la sección 9.1.2 *Mensaje de texto*

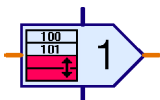
página 542.

8.5.10 Agregar valor



El comando **Agregar** es un comando especial para elementos de listado. Ver sección 8.4.5 *Listado* página 525. Con el comando se transmite un valor, que es agregado al final del listado. Si el listado ya está lleno, se hará caso omiso del comando.

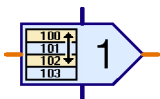
8.5.11 Eliminar valor(es)



El comando **Eliminar** es un comando especial para elementos de listado. Ver sección 8.4.5 *Listado* página 525. Con el comando puede borrarse cualquier número de elementos del final de un listado. La cantidad deseada se transmite como valor con el comando. Si el valor transmitido es mayor que el número de elementos del listado, se borrarán todos los elementos.

Para borrar un listado completamente, puede enviarse un comando **Eliminar** con el valor máximo posible de 32.767 a un elemento de listado.

8.5.12 Intercambiar valores



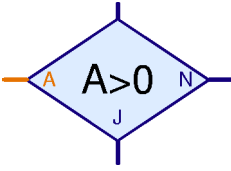
El comando **Intercambiar** es un comando especial para elementos de listado. Ver sección 8.4.5 *Listado* página 525. Con el comando puede intercambiarse cualquier elemento de un listado con el primer elemento del listado. El número del elemento con el que se intercambiará el primer elemento se transmite como valor con el comando. **Importante:** el primer

elemento de un listado tiene el número 0. Si el valor transmitido no tiene ningún número válido de elemento, se hará caso omiso del comando.

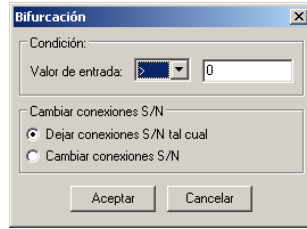
8.6 Compara, Esperar a, ... (Nivel 3)

Los elementos de programa de este grupo sirven para la bifurcación del programa o para el retardo de su desarrollo.

8.6.1 Bifurcación (con entrada de datos)



Esta bifurcación de programa tiene una entrada de datos naranja **A** a la izquierda del elemento. Mediante esta entrada se leerá un valor que frecuentemente procede de un elemento de entrada (ver secciones de 8.7.1 a 8.7.6 a partir de la página 533). Pero la entrada de datos **A** puede unirse también con las salidas de datos de variables, variables de temporizador o operadores (ver sección 8.8 *Operadores*, página 539). El elemento compara el valor de la entrada de datos **A** con un valor fijo, pero libremente definible. Dependiendo si la comparación devuelve una igualdad o no, el elemento bifurcará hacia la salida **S** o **N**.

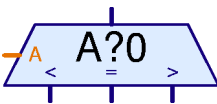


Ventana de propiedades para la bifurcación

- En **Condición** introduces en el campo derecho el valor con el cual debe compararse el valor de entrada **A**. Para la comparación están disponibles todas las operaciones de comparación habituales.
- Si seleccionas **Intercambiar conexiones S/N**, se intercambiarán las salidas **S** y **N** en cuanto cierres la ventana de propiedades con OK. Para llevar las conexiones S/N de nuevo a su posición de salida, puedes intercambiarlas otra vez.

La comparación más utilizada es **A > 0**. Significa que el flujo de programa bifurca hacia la salida **S** si el valor creado en la entrada de datos **A** es mayor que 0. Con ello pueden evaluarse, por ejemplo, las entradas digitales que suministran 1 ó 0. Pero la comparación **A > 0** también puede evaluar de manera práctica las variables de temporizador y muchos más valores.

8.6.2 Comparación con valor fijo

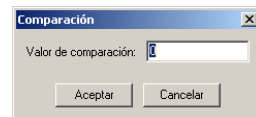


Con el elemento de programa **Comparación con valor fijo** puede compararse el valor de la entrada de datos **A** con un valor fijo, pero libremente definible. Dependiendo de si el valor creado en la entrada de datos **A** es mayor, menor o igual al valor fijo, este elemento de comparación bifurcará hacia la salida derecha, izquierda o central.

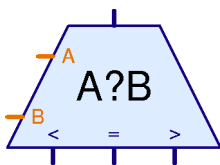
Generalmente se conecta en la entrada de datos **A** de la salida de una variable o un listado. El elemento de comparación puede sustituirse por dos elementos de bifurcación. Sin embargo en muchos casos resulta más claro, si es que se necesita solamente un elemento.

Ventana de propiedades para comparación

- En **Valor de comparación** puedes introducir el valor constante con el cual debe compararse el valor de la entrada **A**.



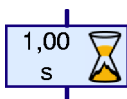
8.6.3 Comparación



Con el elemento de programa **Comparación** pueden compararse los valores creados en las dos entradas de datos A y B. Dependiendo de si **A** es menor que **B**, de si **A** es mayor que **B** o de si **A** es igual a **B**, el elemento bifurcará hacia la salida derecha, izquierda o central. La utilización más frecuente para ello es la comparación de un valor nominal con un valor real. Puede hacerse, por ejemplo, que un motor gire a la izquierda, a la derecha o se detenga en función de cuál sea el valor real respecto del valor nominal.

El elemento de programa **Comparación** no tiene ninguna opción de configuración y por ello tampoco ninguna ventana de propiedades.

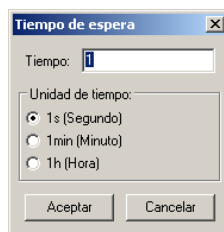
8.6.4 Tiempo de espera



Con este elemento puede programarse un **Tiempo de espera** en una operación. El tiempo de espera empieza en el momento en que llegue el turno del elemento dentro de la operación. En cuanto haya terminado el tiempo de espera introducido, continuará la operación. Ver también sección 3.6.1 *Tiempo de espera* página Fehler! Textmarke nicht definiert.

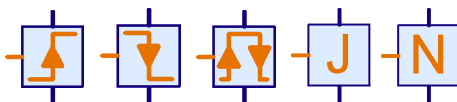
Ventana de propiedades para tiempo de espera:

- En **Tiempo** puedes introducir el tiempo de espera. Se aceptan también números decimales, como 1,23, por ejemplo.
- En **Unidad de tiempo** se selecciona como unidad de tiempo el segundo, el minuto o la hora. La unidad de tiempo, a diferencia de lo que ocurre con las variables de temporizador, no tiene influencia sobre la exactitud del tiempo de espera. Un tiempo de espera de 60 segundos y un tiempo de espera de 1 minuto se comportan exactamente igual.



En el modo de expertos (Nivel 5) se mostrará una ventana de propiedades ampliada, que es más parecida a la ventana de propiedades para variables de temporizador.

8.6.5 Esperar a...

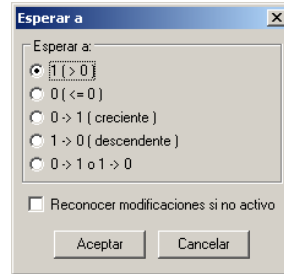


El elemento de programa **Esperar a...** retarda la ejecución del programa hasta que se haya efectuado un cambio en la entrada de datos del elemento o hasta que haya sucedido un estado determinado.

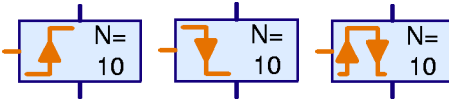
Existe el elemento en cinco tipos: el elemento de la izquierda espera hasta que el valor en la entrada haya aumentado. Este aumento incluye no solamente los cambios de 0 a 1, sino cualquier incremento (por ejemplo, también de 2 a 3). El segundo elemento espera hasta que el valor en la entrada se haya reducido. Y el elemento del centro espera a cualquier cambio, independientemente de en qué sentido sea. El tercer elemento se utiliza a menudo para trabajar con las ruedas dentadas de impulso. Los elementos cuarto y quinto no esperan a un cambio, sino que aguardan a que la entrada adopte el estado Sí (>0) o No (<=0). Si ya está en el estado correspondiente, el elemento no esperará. Por el contrario los tres primeros elementos esperan siempre hasta que se reconozca un cambio en la entrada.

Ventana de propiedades para esperar a cambio

- En **Modo de cambio** puedes elegir una de las cinco funciones descritas arriba.
- Si se halla marcado el botón **Reconocer cambios en estado inactivo**, el elemento reconocerá también los cambios que hayan ocurrido cuando no era el turno del elemento en cuestión en la operación. En este caso el elemento memoriza el último valor conocido. Si el elemento se ejecuta de nuevo, el programa continúa enseguida si es entretanto el valor ha cambiado al estado pertinente. De esta manera se reduce la probabilidad de que un cambio pase desapercibido porque el programa esté haciendo otra cosa.



8.6.6 Contador de impulsos



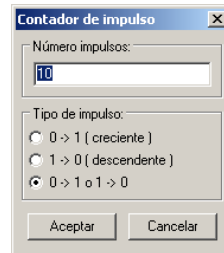
Este elemento de programa espera a que se haya registrado un número determinable de impulsos en la entrada de datos de la parte izquierda antes de continuar con la ejecución del programa.

Es bastante práctico para tareas simples de posicionamiento con ruedas dentadas de impulso. Para posicionamientos más complejos, por ejemplo con un valor variable, deben utilizarse subprogramas con variables.

Ventana de propiedades para contador de impulso

- En **Número de impulsos** introduces el número de impulsos al que se debe esperar antes de que continúe la ejecución del programa.
- En **Tipo de impulso** puedes escoger entre los tres tipos de impulso **0-1**, **1-0** o **cualquier cambio**.

La posibilidad de reconocer cambios cuando el elemento no esté activo (que comentamos en el elemento simple **Esperar a...**) no existe en este elemento.



8.7 Entradas / salidas de la interfase, ...

Este grupo de elementos de programa contiene todos los elementos de entrada y salida. Como utilizar estos elementos se explicará en capítulo 5 *Nivel 3: variables, paneles de control & cia*, página 485.

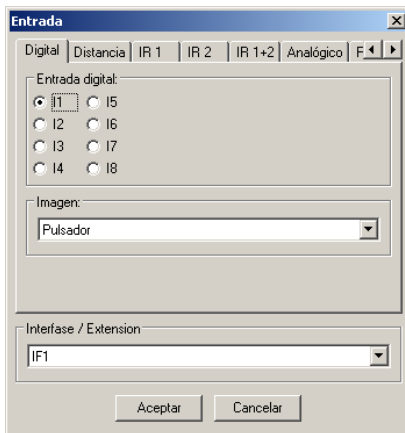
8.7.1 Entrada digital



Con el elemento **Entrada digital** puede consultarse el valor de una de las entradas digitales I1 a I8 de la interfase. Si las dos hembrillas pertenecientes a la entrada estén conectadas con la interfase, el elemento de entrada digital suministrará a su conexión naranja un valor de 1; en caso contrario, le enviará un valor de 0.

Ventana de propiedades para entradas digitales:

- En **Entrada digital** puedes seleccionar cuál de las entradas de la interfase debe utilizarse. Las entradas de módulos de extensión debes seleccionarlas en **Interfase / Extensión**.
- En **Imagen** puedes seleccionar una imagen del sensor conectado en la entrada. En la mayoría de los casos será un **pulsador**. Un **contacto Reed** es un interruptor que reacciona a campos magnéticos. También puede conectarse un **fototransistor** en una entrada digital, aunque realmente es un sensor analógico. Si lo conectas a una entrada digital, puedes utilizar el fototransistor junto con una bombilla lenticular como barrera de luz, cuyos estados son interrumpida (= 0) o intacta (= 1). Si por el contrario conectas el fototransistor a una **Entrada analógica**, podrás distinguir muchas matizaciones en la gama que va del claro al oscuro.
- En **Interfase / Extensión** puedes escoger si quieres utilizar una entrada de la interfase o una entrada de un módulo de extensión o una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.



Bien mirado existe solamente un único elemento de programa para todos los tipos de entradas. A través de las pestañas de la parte superior de la ventana de propiedades siempre te será posible cambiar el tipo de entrada. Esto es especialmente práctico para cambiar entre interruptores, entradas IR y entradas de panel de control.

8.7.2 Entrada analógica



Con el elemento **Entrada analógica** puede consultarse el valor de una de las entradas analógicas. Al contrario que las entradas digitales, que solamente pueden suministrar los valores 0 ó 1, las entradas analógicas pueden distinguir matices muy finos. Todas las entradas analógicas suministran un valor de salida de entre 0 y 1023. En la ROBO Interface, sin embargo, existen varios tipos de entradas analógicas que miden distintas magnitudes físicas. Hay entradas analógicas para mediciones de resistencia, para mediciones de tensión y para un sensor especial de medición de distancia:

Entrada	Tipo de entrada	Rango de medición
A1, A2	Entradas de tensión	0-10,23V
AX, AY	Entradas de resistencia	0-5,5kΩ
D1, D2	Entradas de sensor de distancia	Aprox. 0-50cm
AV	Tensión de alimentación	0-10V

Los sensores habituales de resistencia NTC, fototransistor y fotorresistencia convierten la magnitud que se quiere medir (temperatura o intensidad luminosa) en una resistencia. Por eso debes conectar estos sensores a las entradas **AX** o **AY**. Las entradas de tensión **A1** y **A2** están concebidas para todos los sensores que emitan una tensión entre 0 y 10V.

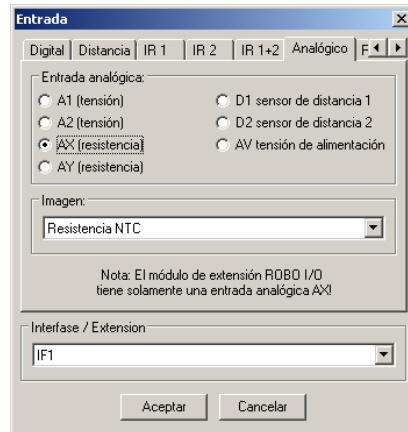
Para la entrada **AV** no existe una hembra en la ROBO Interface. Está siempre conectada con la tensión de alimentación de la interfase. De esta manera puedes, por ejemplo, controlar la tensión de acumulador y hacer que tu modelo vuelva a la posición de salida antes de que el acumulador se vacíe.

A las entradas de sensor de distancia D1 y D2 se conectan unos sensores especiales de fischer-technik que son capaces de medir la distancia que los separa de un obstáculo, por ejemplo.

La interfase inteligente tiene solamente dos entradas analógicas, EX y EY. Estas corresponden a las entradas AX y AY de la ROBO Interface. ¡Las otras entradas analógicas no pueden utilizarse con la interfase inteligente!

Ventana de propiedades para entradas analógicas:

- En **Entrada analógica** puedes seleccionar la entrada analógica deseada de las mostradas en la tabla anterior.
- En **Imagen** puedes seleccionar el sensor que se conecta a la entrada.
- **Interfase / Extensión** te permite escoger si quieres utilizar una entrada de la interfase, una entrada del módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.



En la ventana de propiedades para entradas analógicas verás de nuevo claramente que ROBO Pro utiliza solamente un único elemento para todas las entradas, entre las que podemos movernos gracias a las pestañas. Aún así, para hacerlo más simple, la ventana de elementos pone a tu disposición distintos elementos de entrada a tu elección.

8.7.3 Entrada IR

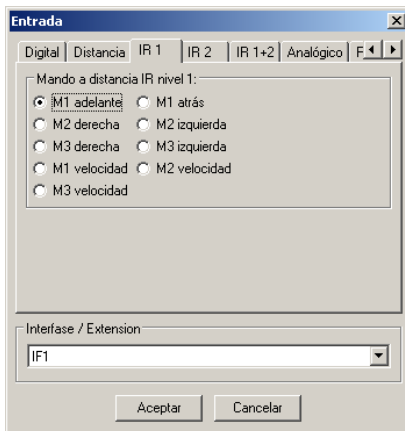


En la ROBO Interface viene incorporado un receptor de infrarrojos para el emisor manual de **IR Control Set** fischertechnik Art. N° 30344. El emisor manual de infrarrojos es muy práctica no solamente como control remoto, sino en general como teclado para manejar los modelos. Hay dos receptores para el **IR Control Set**; puedes pasar de uno a otro con las teclas **1** y **2** del emisor manual. Como consecuencia, en la ROBO Interface tienes la posibilidad de asignar dos funciones diferentes a cada uno de los botones del emisor manual, cambiando de una a otra con el conmutador. Si quieres también puedes utilizar los botones **1** y **2** como teclas normales.

En la ventana de propiedades de una entrada IR puedes ir cambiando, en la barra de pestañas de la parte de arriba, entre **IR 1**, **IR 2** y **IR 1+2**. Si has seleccionado **IR 1**, entonces el elemento de entrada IR devolverá un 1 solamente si el pulsador correspondiente del emisor está pulsado y si el emisor ha sido configurado previamente con el botón 1 para la función 1. Si seleccionas **IR 2**, la configuración del emisor debe contener la tecla **2** asignada a la función 2.

Pero si seleccionas **IR 1+2**, da igual como esté configurado el emisor manual: podrás utilizar las teclas 1))) y 2))) también como entradas.

En el elemento de programa se mostrará tu elección con un 1 ó un 2 blanco en la parte inferior derecha, en el símbolo del emisor manual. Con **IR 1+2** no se mostrará ningún número en el elemento de programa.



8.7.4 Salida de motor



Con el elemento **Salida de motor** controlarse una de las 4 salidas de motor bipolares de una ROBO Interface o de una interfase inteligente. Una salida de motor utiliza siempre dos conexiones de la interfase, mientras la salida de lámpara utiliza solamente una conexión. Encontrarás más información sobre la diferencia entre salida de motor y salida de lámpara en las secciones 8.1.6 *Salida de motor* página 512 y 8.1.7 *Salida de lámpara*.

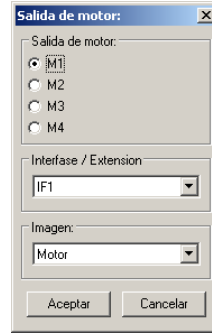
Debe enviarse un comando por medio de un elemento de comando a una salida de motor para que se conecte la salida. Un elemento de motor puede procesar los siguientes comandos:

Comando	Valor	Acción
Derecha	1 a 8	El motor gira hacia la derecha con velocidad de 1 a 8
Izquierda	1 a 8	El motor gira hacia la izquierda con velocidad de 1 a 8
Detener	Ninguno	El motor se detiene
ON	1 a 8	Como Derecha
OFF	Ninguno	Como Detener
=	-8 a 8	Valor -1 a -8: El motor gira hacia la izquierda Valor 1 a 8: El motor gira hacia la derecha Valor 0: El motor se detiene

Ventana de propiedades para elementos de motor:

- En **Salida de motor** puedes seleccionar cuál de las conexiones de salida de la interfase debe utilizarse. Puedes escoger las salidas de módulos de extensión en **Interfase / Extensión**.
- En **Interfase / Extensión** puedes elegir si quieres utilizar una salida de la interfase, una salida de un módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.

En **Imagen** puedes seleccionar una imagen del dispositivo consumidor conectado a la salida. En la mayoría de los casos será un **motor**, pero también puedes conectar un **electroimán**, una **válvula magnética** o una **lámpara** a la salida de motor.



8.7.5 Salida de lámpara



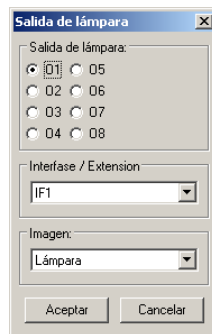
Con el elemento **Salida de lámpara** puede controlarse una de las 8 salidas de lámpara unipolares O1-O8 de una ROBO Interface o de una interfase inteligente. Una salida de lámpara siempre utiliza solamente una conexión de salida de la interfase. La otra conexión del dispositivo consumidor se conectará con una hembra de masa. Un dispositivo consumidor conectado de esta forma puede solamente conectarse o desconectarse, pero no conmutarse. Para más información sobre la diferencia entre una salida de motor y una salida de lámpara, consulta las secciones 8.1.6 *Salida de motor* página 512 y 8.1.7 *Salida de lámpara*.

Debe enviarse un comando con un elemento de comando a un elemento de lámpara, que conectará la salida. Un elemento de lámpara puede procesar los siguientes comandos:

Comando	Valor	Acción
ON	1 a 8	La lámpara se enciende con una luminosidad de 1 a 8
OFF	Ninguno	La lámpara se apaga
=	0 a 8	Valor 1 a 8: la lámpara se enciende Valor 0: La lámpara se apaga

Ventana de propiedades para elementos de salida de lámpara:

- En **Salida de lámpara** puedes seleccionar qué conexión de salida de la interfase debe utilizarse. Debes seleccionar las salidas de módulos de extensión en **Interfase / Extensión**.
- En **Interfase / Extensión** puedes escoger si quieres utilizar una salida de la interfase, una salida de un módulo de extensión o de una segunda interfase. Encontrarás más información en el capítulo 7. *Módulos de extensión y gestión de más de una interfase*, página 505.
- En **Imagen** puedes seleccionar una imagen del dispositivo consumidor conectado en la salida. En la mayoría de los casos será una **lámpara**, pero también puedes conectar un **electroimán**, una **válvula magnética** e incluso un **motor** en una salida de lámpara. Ahora bien: un motor conectado a una salida de lámpara solamente puede girar en una dirección.



8.7.6 Entrada del panel de control



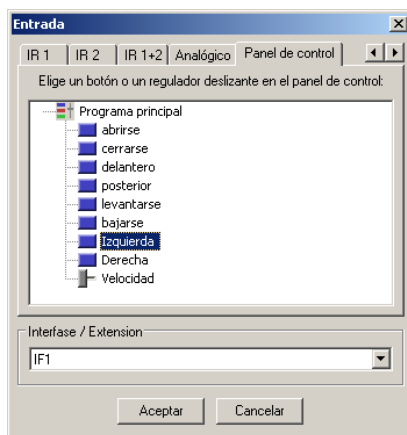
ROBO Pro te ofrece la posibilidad de dibujar paneles de control propios para tus modelos. Encontrarás más información en el capítulo 9 *Los elementos de control y paneles de control* página 541. De esta manera puedes controlar tus modelos cómodamente desde tu ordenador. En el panel de control tienes botones pulsadores, reguladores deslizantes y elementos de entrada a tu disposición. El estado de estos elementos puede consultarse en el programa a través del elemento **Entrada de panel de control**. Los botones pulsadores suministran un valor de 0 ó 1. Los reguladores deslizantes suministran un valor dentro de un rango configurable (por defecto, de 0 a 100).

Los paneles de control pueden utilizarse solamente en el modo online. Encontrarás más información en la sección 3.7 *Modo online o modo de descarga. ¿Cuál es la diferencia?* página 460.

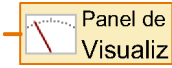
Ventana de propiedades para entradas de panel de control:

Cada programa principal o subprograma tiene su panel de control. Bajo el nombre de los programas están listados los elementos de control. Si todavía no has creado ningún elemento de control, tampoco aparecerá ningún elemento en el listado. Así que antes de poder unir una entrada del panel de control con un elemento de control tendrás que dibujar un panel de control.

La selección que se efectúe en **Interfase / Extensión** no se tendrá en cuenta en las entradas del panel de control, dado que no se trata de entradas reales de un módulo de interfase.



8.7.7 Salida del panel de control



Panel de Visualiz

ROBO Pro te ofrece la posibilidad de dibujar paneles de control propios para tus modelos. Encontrarás más información en el capítulo 9 *Los elementos de control y paneles de control* página 541. Aparte de los botones pulsadores y otros elementos de entrada que te permiten controlar tu modelo, en un panel de control también puedes insertar elementos de mensaje. En los elementos de mensaje puedes, por ejemplo, mostrar las coordenadas en los ejes de un robot o el estado de un interruptor de fin de carrera. Puedes cambiar el valor indicado insertando un elemento **Salida de panel de control** en tu programa y enviando un comando = al elemento (por ejemplo, conectando una variable, una entrada analógica o un elemento de comando en el elemento).

Los paneles de control pueden utilizarse solamente en modo online. Encontrarás más información en la sección 3.7 *Modo online o modo de descarga. ¿Cuál es la diferencia?* página 460.

Ventana de propiedades para indicación del panel de control:

Cada programa principal o subprograma tiene su panel de control. Bajo el nombre de los programas están listadas las indicaciones del panel de control. Si todavía no has creado ningún elemento de control, tampoco aparecerá ningún elemento en el listado. Así que antes de poder unir una entrada del panel de control con un elemento de control tendrás que dibujar un panel de control.



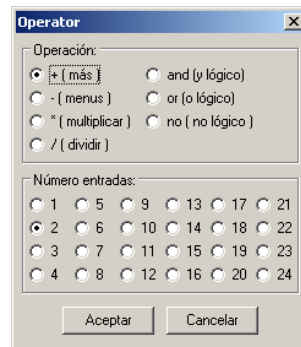
8.8 Operadores

Los elementos de programa de este grupo son los llamados operadores. Los operadores tienen una o varias entradas de datos de color naranja. Los valores de las entradas de datos son enlazados a un valor por el operador y en la salida del operador son emitidos por el comando =.

Ventana de propiedades para operadores

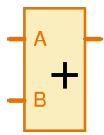
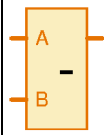
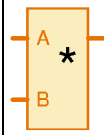
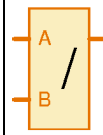
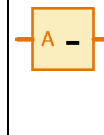
Todos los operadores utilizan la misma ventana de propiedades. La ventana de propiedades también sirve para transformar un operador en otro.

- En **Operación** puedes configurar el modo en que el operador debe enlazar sus entradas. En las dos siguientes secciones se explicará cada función.
- En **Número entradas** puedes configurar cuántas entradas debe tener el operador.



8.8.1 Operadores aritméticos

ROBO Pro pone a tu disposición las cuatro reglas aritméticas en forma de operadores. Los símbolos en dos entradas tienen el siguiente aspecto:

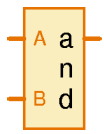
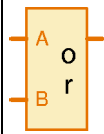
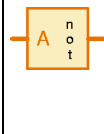
				
Más	Menos	Por	Entre	Menos
$A + B$	$A - B$	$A * B$	A / B	$- A$

Si el operador **Menos** tiene más de dos entradas, se restarán todos los demás valores de entrada del valor de la entrada A. Si el operador **Menos** tiene solamente una entrada, el operador invierte el signo del valor de la entrada.

Si el operador **Entre** tiene más de dos entradas, se dividirá el valor de la entrada A por todos los demás valores de entrada.

8.8.2 Operadores lógicos

En ROBO Pro existen tres operadores lógicos para enlazar entradas digitales por ejemplo:

		
Y	O	No
$A > 0 \text{ y } B > 0$	$A > 0 \text{ o } B > 0$	$A \leq 0$

Los operadores lógicos interpretan los valores mayores que cero de la entrada como **sí** o **verdadero** y los valores menores o iguales a cero como **no** o **falso**. Las entradas digitales suministran un valor de 0 ó 1, de modo que 0 es interpretado como **falso** y 1 como **verdadero**.

El operador **Y** envía un comando = con valor 1 a los elementos conectados en la salida si el valor registrado en todas las entradas es verdadero, es decir, si hay un valor > 0 . En caso contrario, el elemento enviará un comando = con valor 0.

El operador **O** envía un comando = con valor 1 a los elementos conectados en la salida si el valor registrado es verdadero en una de las entradas como mínimo, es decir, si hay un valor > 0 . En caso contrario, el elemento enviará un comando = con valor 0.

El operador **No** envía un comando = con valor 1 a los elementos conectados en la salida si el valor en la entrada es falso, es decir, si hay un valor ≤ 0 . En caso contrario, el elemento enviará un comando = con valor 0.

La función de los operadores lógicos puede reproducirse también con varios elementos de bifurcación, pero a menudo es más práctico enlazar varias entradas con operadores.

9 Los elementos de control y paneles de control

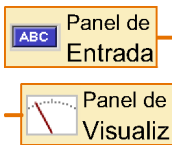
En ROBO Pro pueden definirse algunos paneles de control que permiten controlar modelos complejos. Dado que el panel de control se muestra en la pantalla del PC, los paneles de control funcionan solamente en modo online. Ver sección 3.7 *Modo online o modo de descarga. ¿Cuál es la diferencia?* página 460.

Para crear un panel de control, debes seleccionar **Panel de control** en la barra de funciones:



En el campo gris vacío que se ve debajo puedes proceder a insertar elementos de control. Un panel de control pertenece siempre al programa principal o al subprograma en el que te encontrabas en el momento en que creaste el panel de control. Por ello es importante seleccionar en la barra de subprograma el subprograma correcto antes de crear un panel de control. Casi siempre se creará el panel de control bajo el **Programa principal**.

Los paneles de control contienen mensajes y elementos de mando. Con las indicaciones puedes, por ejemplo, mostrar valores con variables y mensajes de texto. Los elementos de mando, por su parte, funcionan como pulsadores añadidos o entradas analógicas.



A cada elemento de control insertado en el panel de control corresponde un elemento **Entrada de panel de control** (para elementos de mando) o **Salida de panel de control** (para indicaciones) del programa. A través de estos elementos de programa establecerás la conexión entre tu programa y tu panel de control. Los encontrarás en el grupo de elementos **Entradas, Salidas**. Dependiendo del tipo de elemento de mando con el que quieras unir este elemento de programa, se mostrará un símbolo u otro, pero en el

listado de elementos solamente hay dos elementos: uno para indicaciones y otro para elementos de mando.

9.1 Indicaciones

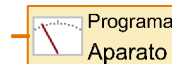
Las indicaciones se utilizan de modo similar a las salidas de interfase. Puedes poner el valor de una indicación con el comando =.

9.1.1 Aparato de medición



El **Aparato de medición** es parecido a un instrumento de aguja analógico. Se utiliza casi siempre para mostrar el valor de entradas analógicas, pero también puedes utilizarlo con variables u otros elementos de programa.

El aparato de medición se controla desde el programa a través de una salida de panel de control. Encontrarás la **salida de panel de control** en el grupo de elementos **Entradas, Salidas**.

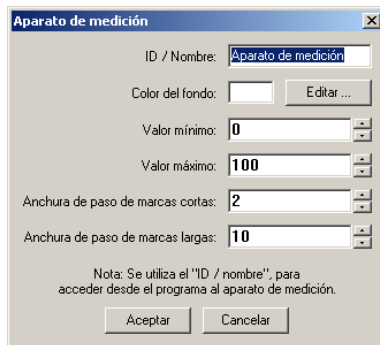


Para fijar el valor del aparato de medición, envía un comando = a la salida del panel de control correspondiente del programa. Casi todos los elementos de programa con salidas de datos envían un comando = cuando cambia su valor. Puedes, por ejemplo, unir entradas analógicas o variables directamente con la salida del panel de control.



Ventana de propiedades para aparatos de medición

- En **ID / Nombre** deberás primero introducir un nombre para el aparato de medición. El nombre es importante para poder distinguir los distintos aparatos de medición de tu programa.
- En **Color de fondo** puedes configurar un color que no sea blanco.
- En **Valor mínimo** y **Valor máximo** indicas los valores que corresponden a la posición de la aguja en el margen izquierdo o derecho de la escala. Si el valor es menor que 0 y el otro valor es mayor que 0, se dibujará una línea 0 especialmente larga.
- La escala consta de líneas largas y cortas. La distancia entre las líneas largas y las cortas se introduce en **Ancho de paso marcas cortas / largas**. Si ambos tienen el mismo valor, se mostrarán solamente las marcas largas.



9.1.2 Mensaje de texto



Un mensaje de texto te sirve para mostrar valores numéricos, texto o ambos.

El mensaje de texto se controla desde el programa mediante una salida de panel de control. Encontrarás la **Salida de panel de control** en el grupo de elementos **Entradas, Salidas**.

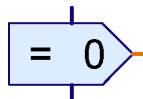


En cuanto hayas unido la salida del panel de control a través de su ventana de propiedades con un mensaje de texto, el símbolo cambiará y aparecerá el nombre del panel de control (por ejemplo, Principal) y el mensaje (por ejemplo, Texto).



Para fijar el texto del mensaje puedes proceder de dos maneras:

- Enviando a la salida del panel de control correspondiente un comando =. Esto resulta muy práctico si quieres utilizar un mensaje para mostrar el valor de una variable o de otro elemento de programa, porque la mayoría de los elementos de programa envían automáticamente comandos = a través de sus salidas de datos cuando cambia el valor. El comando = sobrescribe solamente los 6 últimos caracteres del mensaje. Puedes completar el resto del mensaje con un texto prefijado. De esta manera puedes hacer que el valor vaya acompañado de un texto de indicación en el mensaje. Si el mensaje tiene varias líneas, también puedes poner un texto de indicación en su propia línea. En los mensajes consistentes en varias líneas se sobrescriben solamente los 6 últimos caracteres de la última línea de un comando =.
- Con el comando de texto, que te permite poner el contenido del mensaje a tu gusto. El comando de texto es un elemento de comando especial, que puede enviar no solamente un número sino todo un texto a través de su salida. Al igual que un elemento de comando normal, el elemento de comando **Texto** también puede tener una entrada de datos. Entonces podrás incorporar en



el texto el valor numérico que se encuentra en la entrada de datos. Si envías varios comandos de **texto** a un elemento de mensaje, los textos se juntarán. De esta manera puedes combinar números y textos a tu gusto.

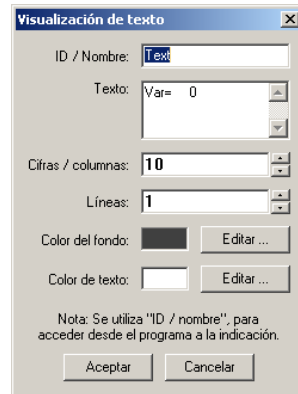
Caracteres de control en comandos de texto

En el elemento de comando **Texto** puedes utilizar los siguientes caracteres para conseguir efectos especiales:

Caracteres de control	Efecto
#####	Emite el valor de la entrada de datos en forma de un número de 5 cifras + signo
##.##	Emite el valor de la entrada de datos en forma de número de 2 decimales, con punto de separación.
##,##	Emite el valor de la entrada de datos en forma de número de 2 decimales, con coma de separación.
\c	Borra el mensaje y pone el punto de inserción al principio del mensaje

Ventana de propiedades para mensajes de texto

- En **ID / Nombre** deberías introducir antes de nada un nombre para el mensaje. El nombre es importante para poder distinguir varios mensajes del programa.
- En **Texto** introduces el contenido del mensaje. Este contenido se mantiene hasta que envías un comando = al mensaje, se sobrescriben solamente los 6 últimos caracteres. El principio del texto se mantendrá, de modo que se te permite indicar junto al número de qué tipo de número se trata. En el ejemplo ilustrado se mantiene el texto **Var=**. El mensaje tiene 10 caracteres, así que se mantendrán $10-6=4$ caracteres.
- En **Cifras / Columnas** y en **Líneas** puedes introducir cuántos caracteres quieres que ocupe el mensaje. En un mensaje con varias líneas puedes mostrar una indicación como **Var=** o **Visitante** en una línea propia.
- En **Color de fondo** y **Color de texto** puedes cambiar el diseño de color del mensaje. Haz clic en **Editar...** para seleccionar un color o definir un color propio.

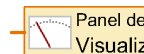


9.1.3 Lámpara indicadora



La **Lámpara indicadora** es el tipo más simple de mensaje. Funciona de manera parecida a un componente de lámpara de fischertechnik conectado a una salida de la interfase.

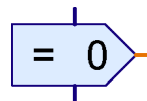
La lámpara indicadora se controla desde el programa a través de una salida de panel de control. Encontrarás la **Salida de panel de control** en el grupo de elementos **Entradas, Salidas**.



En cuanto hayas unido la salida del panel de control a través de su ventana de propiedades con una lámpara indicadora, cambiará el símbolo y aparecerá el nombre del panel de control (por ejemplo Principal) y de la lámpara.

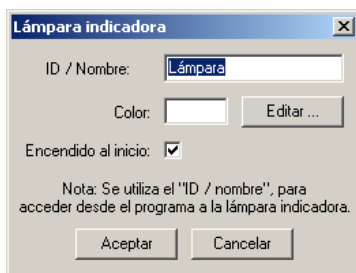


Puedes encender o desconectar la lámpara enviando a la salida de panel de control correspondiente del programa un comando **ON** u **OFF**, igual que con salidas de lámpara verdaderas. También puedes conectar y desconectar la lámpara indicadora con un comando **=**. Si el valor es mayor que 0, se encenderá la lámpara. Si el valor es menor o igual 0, se apagará la lámpara.



Ventana de propiedades para lámparas indicadoras

- En **ID / Nombre** deberías introducir antes de nada un nombre para la lámpara indicadora. El nombre es importante para poder distinguir varias lámparas indicadoras del mismo programa.
- En **Color** puedes cambiar el color de la lámpara indicadora. Para ello haz clic en el botón **Editar**.
- Si está marcado **ON al inicio**, la lámpara indicadora estará encendida hasta que el elemento de programa correspondiente reciba el primer comando. En caso contrario la lámpara indicadora estará apagada al principio.



9.2 Elementos de mando

Los elementos de mando se utilizan de manera similar a las entradas de interfase.

9.2.1 Botón

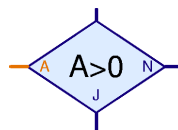
Botón

El **Botón elemento de mando** puede utilizarse como pulsador o interruptor de fischertechnik conectado en una de las entradas de la interfase.

El botón es consultado desde el programa por medio de una **Entrada de panel de control**. Encontrarás la **Entrada de panel de control** en el grupo de elementos **Entradas, Salidas**.

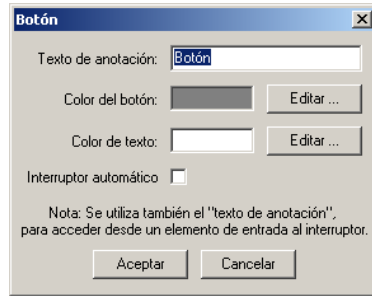


La entrada de panel de control perteneciente al botón puede conectarse como una entrada digital de la interfase a todos los elementos de programa que dispongan de entrada de datos; por ejemplo, a la **Bifurcación**. Si el botón está pulsado, suministra un 1 como valor; en caso contrario, el valor que devuelve es 0.



Ventana de propiedades para botones

- En **Texto de anotación** puedes introducir la leyenda del botón, que al mismo tiempo es el nombre con el que se accede a él desde el programa. En el botón no existe el campo de nombre/ID añadido que sí existía en los otros elementos de control.
- **Color del botón** y **Color de texto** te permite cambiar el diseño de color del botón. Para ello haz clic en **Editar....**
- Si aparece un ganchito en el interruptor automático, el botón no se acciona como un pulsador sino como un interruptor. Haciendo clic en el botón, el botón es presionado hacia dentro y se quedará así hasta el segundo clic. En caso contrario el botón se accionará como pulsador y al soltarlo saltará de nuevo hacia fuera.



9.2.2 Regulador

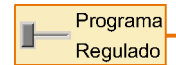


Puedes utilizar el regulador como un potenciómetro conectado en una entrada analógica de la interfase. Al contrario que al botón, el regulador no solamente puede suministrar los valores 0 y 1, sino varios valores igual que si fuese una entrada analógica. El rango de valor puede configurarse desde la ventana de propiedades. El regulador puede utilizarse, por ejemplo, para configurar la revolución del motor de 1 a 8.

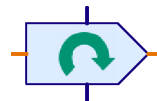
El regulador es consultado desde el programa por medio de una entrada de panel de control. Encontrarás la **Entrada de panel de control** en el grupo de elementos **Entradas, Salidas**.



En cuanto hayas unido la entrada del panel de control desde la ventana de propiedades con un regulador, cambiará el símbolo y aparecerá el nombre del panel de control (por ejemplo Principal) y del regulador.

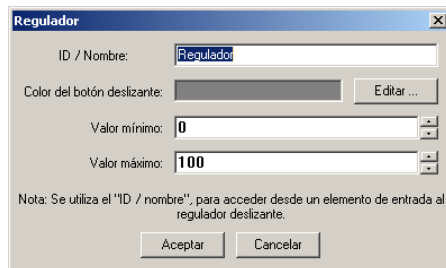


La entrada del panel de control perteneciente al regulador puede conectarse como una entrada analógica de la interfase a todos los elementos de programa que dispongan de una salida de datos. Muy a menudo el regulador se conecta a un elemento de comando con entrada de datos para que controle la velocidad de un motor.



Ventana de propiedades para regulador

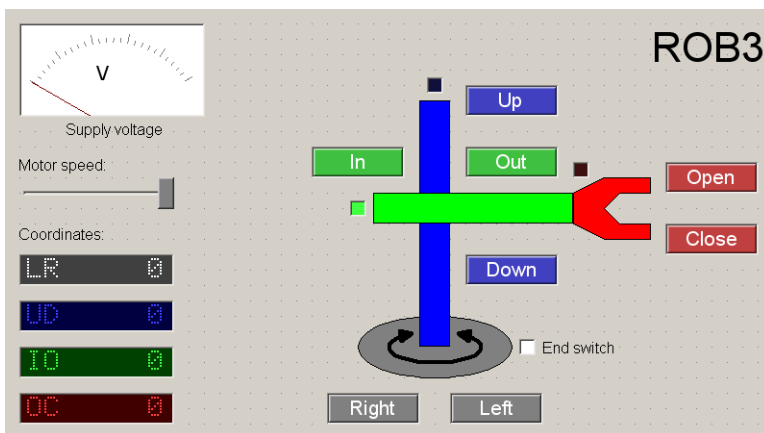
- En **ID / Nombre** deberías introducir antes de nada un nombre para el regulador. El nombre es importante para poder distinguir varios reguladores del mismo programa.
- En **Color de botón deslizable** puedes cambiar el color del botón deslizable. Para ello haz clic en **Editar**.
- En **Valor mínimo** y **Valor máximo** introduces el área de valor para el regulador. Si quieres utilizar el regulador para el control de una velocidad de motor, el rango de valor debería estar entre 1 y 8.



10 Funciones de dibujo

ROBO Pro dispone de las tradicionales utilidades de dibujo. Las encontrarás en la ventana de grupos de elementos, en **Dibujo**. El subgrupo **Formas** contiene herramientas de dibujo para trabajar con diferentes formas geométricas básicas. En el subgrupo **Texto** encontrarás herramientas de dibujo de texto para diferentes tamaños de fuente. Los otros subgrupos contienen funciones para cambiar el color y para el grosor de las líneas.

Las utilidades de dibujo te permiten ilustrar tus paneles de control y tus programas para resaltar funciones. Aquí, por ejemplo, te mostramos un panel de control dibujado para un robot:



Los botones, indicaciones de coordenadas y lámparas de interruptores de fin de carrera se dejan en el mismo color que los ejes individuales del dibujo esquemático del robot. De esta manera obtenemos un panel de control bastante claro.

La utilización de las utilidades de dibujo no debería plantear grandes problemas, por lo que a continuación nos limitaremos a exponer solamente algunas de las cuestiones que pueden ser menos evidentes:

- Al contrario de lo que ocurre en muchos otros programas, los objetos gráficos, como por ejemplo rectángulos y círculos, **no** se arrastran pulsando el botón del ratón, sino con dos clics del ratón: uno en la esquina superior izquierda y otro en la esquina inferior derecha.
- El texto **no** se edita en una ventana de diálogo, sino directamente en el área de trabajo. Si insertas un nuevo objeto de texto, lo que único que aparece en el primer momento es un marco en color azul claro. Es el momento de escribir sencillamente desde el teclado; verás el texto escrito directamente en el área de trabajo. También puedes insertar texto desde el portapapeles con CTRL+V.
- Tras haber dibujado un objeto, puedes editarlo desplazando los puntos de anclaje, que son los puntitos azules del objeto. También existen puntos de anclaje para girar y transformar los objetos. Un rectángulo tiene dos puntos de anclaje en la parte superior izquierda; desplazar el punto de anclaje más pequeño te permite redondear los cantos del rectángulo. Para salir del modo de edición, pulsa el botón derecho del ratón o la tecla **ESC**.

- Si posteriormente quieres editar un objeto, debes seleccionar en el menú **Dibujo** la función **Editar**. Si después haces clic en un objeto, aparecerán de nuevo los puntos de anclaje en azul claro.
- Muchos objetos tienen dos o más modos de edición y de dibujo. Puedes cambiar de un modo a otro con la tecla de tabulación mientras estás dibujando o editando un objeto. En un círculo puedes, por ejemplo, seleccionar si quieres trabajar con dos puntos angulares o con el punto central y un punto angular. En los polígonos puedes cambiar entre edición de punto y funciones como "girar". En objetos de texto puedes conmutar entre edición de texto como modificación del tamaño de texto y del ángulo de giro.
- En el menú **Dibujo** también se hallan las funciones **Traer al frente / Enviar al fondo**. Con estas funciones puedes traer todos los objetos seleccionados (marcados en rojo) al frente o enviarlos al fondo, de forma que tapen otros objetos o queden tapados por ellos respectivamente.
- La utilidad **Cuadrícula** del menú **Dibujo** te permite mostrar o esconder la cuadrícula de dibujo. Pero cuidate de que la cuadrícula esté activada si te dispones a editar tu programa, dado que todos los elementos de programa están adaptados a ella.
- Al trabajar con objetos de texto puedes cambiar la alineación del texto pulsando CTRL más una de las teclas de 1-9 del bloque numérico. Pero ojo: sólo funcionará si la luz de BloqNum del teclado está encendida. Si no es así, antes de nada debes pulsar la tecla BloqNum.