INTRODUCTION A

CREATING AN APPLICATION B

SIMULATION C

UTILITIES D

EXAMPLES E

APPENDIX F

Section		Page
	Preface	**P/1**
1	**Introduction and installation**	**1/1**
1.1	General	1/1
1.1.1	Functions offered by SIMACTEL software	1/1
1.2	Required configuration	1/1
1.3	Checking the hardware	1/2
1.4	Connections	1/3
1.5	Software installation	1/3
1.5.1	Preliminary operations	1/3
1.5.2	Installation procedure	1/3
1.5.3	Operations following installation	1/4
2	**Using SIMACTEL**	**2/1**
2.1	Simulation methodology	2/1

Section		Page
3	**Accessing SIMACTEL software**	**3/1**
3.1	From the X-TEL software workshop	3/1
3.2	From XTEL-BROWSER	3/2
3.3	SIMACTEL main window	3/2

Software environment

SIMACTEL runs under MINI XTEL-BASE or XTEL-BASE (version • V52). Certain precautions should be taken when selecting configuration parameters for WIN-OS/2 sessions, depending on which version of OS/2 is used (OS/2 2.1, WARP Red or WARP Blue). Further details can be found in section 1 of part A : Introduction and installation.

1.1 General

This document describes the installation and use of SIMACTEL application simulation software, reference TXT L SIM V6, on one of the following terminals equipped with an OS/2 operating system and the X-TEL software workshop :
- FTX 507 terminal,
- IBM PS/2 or IBM PC compatible microcomputer.

This design office software is used to develop a virtual representation of a machine or process. It then uses this representation to validate and test PLC programs.

1.1.1 Functions offered by SIMACTEL software

A PC equipped with SIMACTEL software connected to the PLC replaces all equipment upstream of the PLC, such as the I/O cards, the electrical interface and the process itself. A model configured using SIMACTEL has the same characteristics as the installation and thus reacts in exactly the same way in relation to the PLC.

To do this, SIMACTEL software has the following functions :
- representation of a machine giving a full description of its mechanical and electrical parts,
- validation of this representation by simulation in offline mode,
- control of the “virtual machine” in online mode,
- display and analysis of the state of the process,
- location of faults on the application for debugging the PLC program.

1.2 Required configuration

SIMACTEL software requires the following minimum hardware configuration :
- PS/2 or IBM PC compatible computer with OS/2 operating system version 2.1 or version 3.0 “Warp” (the “Warp” version is recommended).
- a VGA screen (a high resolution screen is recommended),
- an 80386 or higher processor and an 80387 coprocessor (a 486 DX is recommended),
- 16 Mb of RAM,
- a hard disk with 20 Mb of free space,
- a mouse.

Important :

- The XTEL (or Mini-XTEL) software workshop, version • 5.6, must be installed on the terminal.
- A SIMACTEL application can only be connected to TSX or PMX processors with software version 5.0 or later (see the label on the processor).
- EPSON and HP Laser Jet compatible printers are necessary for printing trend diagrams of movements. To print machine and model dossiers, use your usual printer.

SIMACTEL operates perfectly well in full screen or in window mode under WARP V3.0 (BLUE version).

However, some problems connected with emulating WINDOWS under OS/2 2.1 or WARP red can arise :

- in terms of the screen refresh
- or the system may freeze if mouse operations are performed too quickly.

These problems can be solved :
• either by installing a FIXPACK, index • 10, (supplied by IBM) under WARP red
• or by using **full screen** mode for SIMACTEL.

These parameters are set at OS/2 **System Setup** group level, by clicking on the **WIN-OS/2 Setup** icon, and selecting **WIN-OS/2 Full Screen** as the WINDOWS activation parameter.
Also, for problems encountered when switching between sessions : it is advisable to set the **VIDEO_SWITCH_NOTIFICATION** parameter to OFF.

1.3 Checking the material supplied

SIMACTEL software comprises :

- a set of 3" 1/2 SIMACTEL software floppy disks,
- a set of 3" 1/2 XTEL update B floppy disks,
- a software protection key,
- a licence agreement,
- this manual.

1.4 Connections

This sections assumes that all connections specific to the terminal (monitor, keyboard, mouse, printer, key support, etc) have been made, and this only explains how to install the software key.
This is done by inserting the software key in the free slot in the key support.

Note :
This software key contains the access right, which is essential for gaining access to SIMACTEL software. The Key Manager tool, supplied with X-TEL BASE, is used to transfer this right to the master key, thus concentrating all the rights in a single key (master key) and freeing a slot in the key support. Refer to the X-TEL documentation for the relevant installation procedures.

1.5 Software installation

1.5.1 Preliminary operations

Before installing SIMACTEL software on the hard disk, it is advisable to :

- read the licence certificate and warranty concerning restrictions on copying and installing the software,
- make a copy of the original installation floppy disks and use these backup copies to work with, so as to any avoid accidental damage to the originals.

Important :

The floppy disks (3" 1/2) are supplied write-protected. Do not change the position of the tab.

1.5.2 Installation procedure

The following operations must be carried out before installing SIMACTEL software :

1. Check that the version of the X-TEL or MINI X-TEL software workshop installed on the terminal is at least 5.6 and that the C update is also installed (refer to the relevant documentation). If so, proceed with the installation of SIMACTEL. If this is not the case, the software workshop must be installed first (refer to the relevant documentation).
2. Close all current sessions (refer to the relevant documentation).

Installing SIMACTEL software

- open an OS/2 full screen session,
- insert the first **SIMACTEL software** floppy disk in the drive,
- enter the drive identifier (a: or b:), then confirm with "Enter",

A

- at the new prompt ([A:\] or [B:\] for example), enter the install command,
- follow the procedure displayed on the screen for installing all the**SIMACTEL software** floppy disks,
- once installation is complete, check the configuration and confirm with "Enter",
- remove the floppy disk from the drive and return to X-TEL by pressing "Ctrl" and "Esc" simultaneously.

1.5.3 Operations following installation

Configuring the operating system

As with all software running under WINDOWS (or WIN-OS/2), SIMACTEL uses the system default parameters found in the **WIN-OS/2 Setup** tool (which can be accessed via the OS/2 **System Setup** group).

If any problems arise when launching SIMACTEL, check that the following parameters are set correctly :

- WIN RUN MODE	set to 3.1 Enhanced Compatibility
- DOS FILES	set to 30 (at least)
- DPMI DOS_API	set to AUTO
- DPMI MEMORY_LIMIT	set to 10 (at least)

Character font set used by SIMACTEL

SIMACTEL requires **Courier New** to be installed on the terminal (at WINDOWS or WIN-OS/2 level). If this font is not installed, the software signals an error and cannot start up. If this occurs, the font should be added at Program Manager level (WINDOWS or WIN-OS/2). Refer to WINDOWS-OS/2 documentation for instructions on how to add fonts.

Briefly, the procedure is as follows :

From the **Program Manager** group (WINDOWS or WIN-OS/2), select the **Control Panel** icon, and then the **Fonts** icon. The list of fonts installed on the terminal is displayed, as is the **Add** button. Click on this button, and then select the SYSTEM subdirectory from the directory in which WINDOWS is installed. If the **Courier New Normal (True Type)** font is available in the list displayed, simply add it.

If the font is not displayed (ie. it is not present on the hard disk) repeat the operation using SIMACTEL installation floppy disk n°1 and selecting drive A.

2.1 Simulation methodology

The methodology suggested here is designed to assist the user in developing a SIMACTEL application, and then using it as effectively as possible to test PLC programs. To do this, it is useful to perform a number of operations in chronological order at various levels (station and tools, PL7-3, and SIMACTEL).

1. Describe all possible "movements" of the application.

Section 2, part B

- Using all the documents produced by the machine or installation designers (functional description, cycle diagrams, list of sensors/actuators, flow charts, layout diagrams, etc), identify all the possible basic movements of the application : mechanical movements, movement of parts, changes to certain values (for example : level within a vat), and - by extension - communication between the PLC and the subassemblies of the application viewed as a "black box" (the "movements" correspond to the transcription of data exchange datagrams rather than the control of actuators/reading of sensors).

- In SIMACTEL, create the axes required for describing the movements identified : **Axes** window, **Create** button.

Section 2.2, part B

- For each axis, describe its name, length, type (linear, rotary), and the associated sensors.

Section 2.3, part B

- For each sensor, indicate its position on the axis and its logic behavior. The **"Special"** button is used to describe the behavior of the sensor during movements along the axis.

- For a given axis, display the behavior of the sensors along the axis using the **Display** button in the **Axis Sensors : ...** window to check the behavior descriptions.

- Special cases :

- simple linear axes with a "start limit" sensor and an "end limit" sensor can be created quickly using the **Standard** button in the **Axes** window**.**
- as with any other element of the application, one or more axes can be incorporated immediately by instantiating a standard model.

• For each axis, describe the associated movements by selecting the Movements button in the Axis Configuration window.	Section 2.4, part B
• Describe each movement by its direction of change, speed and a Boolean equation giving the activation conditions according to the selected design (associated actuator or preactuator, presence of power supply).	
• Deal separately with the modelling of the flow of parts in materials handling installations. It may in this case be necessary to create Boolean variables (Variables button in the Axis Configuration window) in order to describe the logic of the change in "parts present" using Boolean equations involving axis positions or memory functions.	Section 2.5, part B
• During the modelling process, perform a "static" check of the application model (for example : edit the list of elements of each type, variables "upstream" or "downstream" of each axis, status of the list of the "undefined" variables, ie. those variables for which the behavior has not yet been defined in the model, etc).	
• Similarly, consistency check functions exist in SIMACTEL and can be activated via the Check button in the Edit Axis window.	Section 2.8, part B
2. Describe the electrical subassemblies of the application.	Part B
• Once the machine electrical wiring diagrams have been validated, it is possible in SIMACTEL to describe all the electrical and electromechanical components which act as the interface between the control PLC and the movements of the application.	
• Describe the I/O configuration (remote and in-rack discrete I/O) : declaration of the I/O cards and the TBX modules for the remote I/O (**Create I/O Configuration** window).	Section 4, part B
• Edit the I/O created in this way : assign a mnemonic and a comment to the outputs; assign a mnemonic, a comment and a Boolean equation to the inputs, the equation indicating the conditions under which the input will change to 1, depending on the associated wiring.	

- Two file import functions are available for retrieving the description of the I/O : — Section 4.3, part B
  - one, accessible via the **SDBASE** File button, is for reading the mnemonics and comments from an .SCY file generated by **SDBASE**.
  - the other, accessible via the **Neutral File** button, is for importing the I/O from the standard file generated by **X-TEL-CAD** (or certain electrical CAD tools).
- Describe the hard-wired logic by entering the relays and the power supply terms in the **Relays** window. Each relay can be on-delay or off-delay to reproduce the contact switching time. — Section 3, part B

  Various functions are available to help with entering the associated Boolean equations : **Edit Help, Dictionary**, etc, buttons (these functions are available for all equations entered in SIMACTEL).
- Describe the conventional operator panels wired to the I/O terminal block by declaring the various elements (pushbuttons, indicator lamps, switches, etc) : **Edit Panel** window. — Section 5, part B
- Access to undefined variables enables the user to check the consistency of the model regularly or to detect any data entry errors. — Section 2.7, part B

3. Describe the application view(s) — Section 6, part B

- A main view is created, without a title, which will then become the background of the simulation interface, for displaying the main axes and variables of the model : **Create View** window**.**
- Other views, each with a title, can be added to provide greater detail of the machine subassemblies.

4. Validate the model obtained

- Launch the SIMACTEL station tool and switch to “offline mode” (Simulation menu from the main **SIMACTEL** window). — Section 3, part C
- Once the model of the machine has been loaded, the simulation interface is used to access all control and analysis functions : access to operator panels, views, external variables, simulation traces, etc.

- This validation corresponds to the basic tests carried out during the initial power-up operations of a real installation. This can be sufficient for simple machines (with only a few SIMACTEL axes) and if the movements involved are not particularly correlated.
- It is highly advisable to validate the global behavior of the model, which also enables the overall design to be checked in collaboration with the mechanical designer. To do this, a “Chart” function is used to describe the sequence of movements (functional description) and to automatically animate the model, while remaining in offline mode (not connected to the PLC) :
- Describe the animation chart (**Machine Description** window, **Chart** menu). Here the operation of the installation is reproduced (for example : machining cycle of a machine) using the documents supplied by the mechanical designer : Mechanical Grafcet chart, trend diagram of movements, or the literal description).

  Section 3.1, part C

  Describe the chart(s) required : initial step, steps in the chart, transitions, links between the elements to determine the overall structure. Then describe each step and each transition (**Edit** button, having selected the step or transition to be described).
- Since this is a question of substituting the control system in sequencing the movements of the machine, the step actions can be on the electrical contacts, preactuators or actuators of the model, while the transition conditions can retrieve sensors, operator panel buttons or any other variable from the application.
- Next return to the simulation environment, by selecting **offline mode**, having first selected the animation Grafcet chart described above.
- The simulation can then proceed via continuous communication between the animation Grafcet chart and the model of the application. It is possible, during simulation, to display and interrogate the animation Grafcet chart which appears in the list of views.
- Use the control tools to animate the simulation : **panels**, **external**, **contexts**, **scenarios**
- Use the SIMACTEL operating modes (step by step mode in particular) to advance operation of the model progressively.

  Section 2, part C

- Use the display tools (**views**, **pages**) and the analysis tools (**examination**, **activity**, **evolutions**, **traps**) for debugging the model.
- Access the **Description** function for any "online" modification made to the model which does not alter the current mechanical description.
- Use the **Trace** function to save any changes made to the machine and to represent these in the form of a trend diagram, so that the behavior of the model can be validated definitively (if possible in collaboration with the mechanical designer).
- At this stage, the SIMACTEL model obtained must correspond exactly to the real machine currently being developed. It can then be used to initiate debugging of the PL7.3 application as soon as the coding and test phases have been completed by the control systems engineer.

5. Print out a dossier of the model of the application in the required format.

Section 2, part D

- Click on the **Tools** button in the SIMACTEL configuration interface, then select **Print Selected Machine** from the **Utilities** window.
- Select the elements to be printed from those listed in the **Print on <default printer>** window. Check the **Open SIMPRINT** box to use a different print format from the default format or to set the print options.
- In SIMPRINT, modify the appearance of the printed dossier by changing the print options, which can be accessed from the **Format** menu, and by modifying the content of the fields in the main application window.
- Check the content of the dossier by displaying it using the **Print Preview** command in the **File** menu**.**
- Print the documents.
- Save the modified print format to be able to use it for other print dossiers.

6. Prepare the simulation platform

- The terminal equipped with X-TEL, and SIMACTEL should be connected to the PLC in accordance with the communication driver associated with the station (via

Section 2.4, part C of X-TEL V5 software workshop

the terminal port or UNITELWAY port : on a TSX SCM 21 module or UNITELWAY port integrated in the CPU, type XX.425).

Note
FIPWAY, ETHWAY and MAPWAY communication supports are also accepted. Refer to the relevant manuals for their installation.

7. Configure the SIMACTEL OFB in the PL7.3 application

Sections 6.1, 6.7 and 6.11, part B of PL7-3 V5 Operating Modes

- Launch PL7.3.
- Access OFB configuration.
- Select the OFB from the SIMACTEL family.
- Instantiate the SIMACTEL OFB (once).
- Reconfigure the PL7.3 application.

8. Transfer the PL7.3 application to the PLC

Section 14.2, part G of PL7-3 V5 Operating Modes

- Launch the **Transfer** station tool.
- Select Disk -> PLC Station.../Global Transfer from the Transfer menu.
- Once the transfer is complete, quit the **Transfer** station tool.

9. Adjust the SIMACTEL OFB

Section 10.3 part F of PL7-3 V5 Operating Modes

- Access the **Constant Mode** screen by selecting **Constant** in the **Mode Selection** screen.
- Select **OFB** followed by SIMACTEL OFB to access the internal constants of the OFB.
- Use the **MODIFY** command to adjust the WATCHDOG constant which is used to configure the time the OFB waits before sending the historic detailing the changes in state of the outputs to SIMACTEL.
- The simulation is ready to start.

10. Launch the simulation

- Launch the simulation by selecting the SIMACTEL station tool, then online mode from the simulation menu.

11. Carry out the tests to perform a complete debug of the PL7.3 application

- Use all the tools available in SIMACTEL to optimize productivity and the relevance of the tests (**Contexts**, **Scenarios**, **Fault** to trigger faults on the components of the application,**Trace**, **Force**, etc). It is highly advisable to prepare a test plan and test files beforehand to ensure that the PL7.3 application debug operation is exhaustive.
- Access the simulation “Server” to follow the changes of the I/O during simulation. The server is a WINDOWS application launched automatically with SIMACTEL, which manages communication between the OFB and SIMACTEL. Access the various display menu functions to show a historic of the I/O changes, or a dynamic view of the racks and TBX modules used in the application.
- Confirm acceptance of the PL7.3 application once all functions have been tested and validated. Use the **Trace** function as required to edit the trend diagrams in the main operating modes.

A

3.1 From the X-TEL software workshop

The access procedure is exactly the same as that used for all other Station Tools under X-TEL.

SIMACTEL software is accessed by opening the corresponding window. This is done as follows :

Procedure	**Mouse**	**Keyboard**
1. Open the Desktop Manager window	double-click on the icon	<Ctrl><Esc> < > or <¬> <Enter>
2. Open the Telemecanique group	double-click on Telemecanique	< > or <¬> <Enter>
3. Open the User window	double-click on X-TEL	< > or <¬> <Enter>
4. Enter the user parameters (this opens the Volumes window)	click in the data entry field, xxx, OK	<¬>, xxx, <Enter>
5. Open a volume	double-click on the volume icon	<Alt> <∅> Window <¬> <¬> Volumes <Enter> <Tab> <F2>
6. Open a project	double-click on the project icon	same as Volumes <¬> Projects
7. Open a station	double-click on the station icon	same as Volumes <¬> Stations
8. Select New from the Define menu in the Station Tools window (if the Simactel icon is not already available among the Station Tools)	click on Define menu, New	<Alt> <∅> Window <¬> <¬><Station Tools> <Enter> <Alt> <∅><Define> <¬><New> <Enter>
9. Select SIMACTEL	double-click on simactel	<¬><simactel> <Enter>
10. Open the SIMACTEL window.	double-click on the Simactel icon	same as Volumes <¬> Station Tools

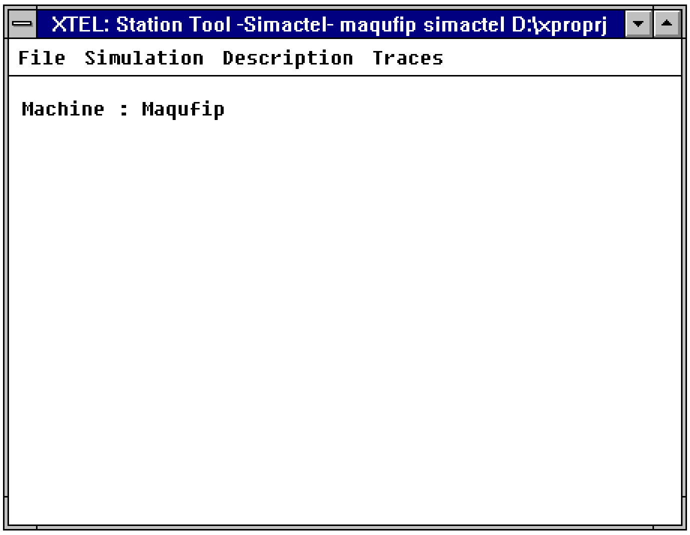
3.2 From XTEL-BROWSER

The Browser is a V5 tool which can be accessed from the Telemecanique Group window or from the Display menu in all X-TEL primary windows. It is used to display the hierarchy of the elements in the software workshop and to access the various levels quickly : to open all the windows in the software workshop or launch a function or tool. To launch the Simactel Station Tool from the Browser, it must already exist at station level. Then proceed as follows :

Procedure	Mouse	Keyboard
1. Open the Desktop Manager window	double-click on the icon	<Ctrl><Esc> < > or <¬> <Enter>
2. Open the Telemecanique group	double-click on Telemecanique	< > or <¬> <Enter>
3. Open the Xtel Browser window	double-click on Xtel Browser	< > or <¬> <Enter>
4. Enter the user parameters, (this opens the Browser)	click in the data entry field, xxx, OK	<¬>, xxx, <Enter>
5. Use the Browser to find the Simactel station tool for the required station	click on the items	<∅> simactel
6. Open the SIMACTEL window.	double-click on the simactel item	<Enter>

3.3 SIMACTEL main window

Once SIMACTEL has been launched, the main window provides access to the various software functions.



The menu line of the main window comprises the following 6 items :

- **MACHINE :**

  This accesses the list of existing machines.

- **CHART :**

  This item can be accessed once a machine has been selected.
  As for MACHINE, a window appears with the list of existing Grafcet chart files.

- **SIMULATION :**

  This button activates a menu which is used to launch the simulation in one of two available modes : “offline” or “online”.

- **TRACE :**

  Accesses the representation in the form of a trend diagram of the evolution of variables during simulation.

- **DESCRIPTION :**

  Used to switch to the Description environment.

- **END :**

  As the name suggests.

A

1.1 Introduction

The principle behind SIMACTEL consists of modelling the application independently of any assumption relating to the operation of the control system. This makes it possible to test both the correct operation of an installation and also the various operating modes, or the behavior of the PLC program in the event of any faults in the application (sensor break, for example).

The first step in setting up a simulation using SIMACTEL software is the creation of a representation of the application. To do this, the software comprises a full description interface, which is explained in the following sections. This is used to describe the following elements :

• the axes and other elements representing the mechanical part,

• the relay interface,

• the PLC I/O and terminal block,

• the operator panels,

• the views,

• the models.

1.2 Modelling an automated system application

Some simulation methods use Grafcet charts to describe a “dual” model representing the expected operation of the application.
This method is particularly well used in Grafcet chart CAD software packages where it offers the advantage of representing the evolutions of the application using the same tool as the control system, thus enabling animation of the whole system via the same execution algorithm.

The main disadvantage of this method is that only the normal operation of the control system can be tested properly.

With SIMACTEL, however, the underlying principle is that of modelling the application independently of any assumption relating to the operation which will be imposed on it by the control system. In addition to testing normal operation, the operating modes can be tested under correct conditions and the behavior of the control system can be analyzed in the event of any faults occurring in the application.
This modelling method is based on the concept of the “Axis”.

B

1.2.1 Reports sent by a physical value : axis and sensors

In an automated system, the PLC sends orders in the form of outputs which change the physical values, such as the position of a cylinder.
In return the PLC receives reports from the inputs via the sensors on the machine.

In SIMACTEL, the “Axis” is used to represent the various possible values and the corresponding states of the sensors which report them.

For example, the axis describing a cylinder will have a length corresponding to the cylinder travel. Its sensors will be positioned at the end and start limits.

Types of sensor :

The sensors can be placed at any point along the axis with which they are associated.

These sensors have four types of logic behavior in SIMACTEL :

- “Positive logic” for normally open sensors. A sensor of this type is at logic level “1” when the cam which activates it is also at 1.
- “Negative logic” sensors perform the exact opposite of those described above.
- “Latch” sensors change to state “1” when the physical value which they are representing is greater than the value given by the position on the axis.
- “Release” sensors perform the exact opposite of latch sensors.

1.2.2 Change of physical values : concept of movement

The change in state of physical values in an automated system application is represented using the axes with which the sensors are associated.
The changes in these values are described in SIMACTEL by “Movements” characterized by a direction, a speed and a logic equation.

Take the example of a pneumatic cylinder controlled by a bistable directional control valve. Depending on the position of this directional control valve, which could be considered as a memory function, the cylinder extends or retracts until it reaches the furthest point of its trajectory.

Movements are made as long as there is air pressure present.
Since the directional control valve can take on two different states, it can be represented by a Boolean variable.

Suppose that the “true” state of this variable, which we will call “Dis”, corresponds to the extension of the cylinder. If “Pneu” is the variable representing the motive air pressure, the equations for the cylinder movements will be :

- for extension : Dis . Pneu
- for retraction : /Dis . Pneu.

This description of a cylinder is a very simple example, but nevertheless one which is often encountered in practice. In other cases, the movement logic equations can be quite complex. It must be ensured that they are mutually exclusive, since only one movement can be active on an axis at any one time.

B

1.2.3 Description variables and preactuators

In the cylinder movement logic equations, we have used a Boolean variable which represents the two possible positions of a directional control valve.

Generally, in actuator movement equations, variables corresponding to the state of preactuators are used : these will be directional control valves in the case of a cylinder, but they could be contactors for movements controlled by electrical actuators.

However, some modelling examples require variables, used in particular to represent the presence of parts at certain points of the machine, which can be used in movement equations.

Thus it is possible to describe these entities when modelling a machine in SIMACTEL.

In this way, a directional control valve will be represented by a bistable function, characterized by latch and release terms, these terms corresponding to solenoid valves controlling the directional control valve.

Relays and contactors as well as the description variables are described by Boolean expressions.

1.2.4 Interface with the control system : input logic equations

Communication between the control system and the automated system application is carried out via I/O. In SIMACTEL, the inputs are described by Boolean expressions, used in particular to provide the link between the sensors representing the evolution reports and the data viewed by the PLC.

Variables representing the conditions for the power supply to the PLC cards are also used in these input equations.

The sensors are not the only elements affecting the inputs of a PLC : data from the operator panels in particular is also taken into account.
This aspect of the machine configuration will be explained in more detail in the section relating to the description interface.

1.2.5 Concept of an “external variable”

The modelling of a cylinder has been used as an example to illustrate the concepts of the Axes and associated Movements.

In the equations expressing the Movements of this cylinder, we have used a variable “Dis” corresponding to the state of a bistable function characterizing the position of a directional control valve, and a term “Pneu” representing the air pressure enabling the movements of the cylinder. While the logic state of “Dis” is determined by the latch and release terms, “Pneu” represents a condition which is external to the modelling.

All variables whose state is not defined internally by the model, such as this pneumatic condition, are defined as “external variables” in SIMACTEL.

Note :

The use of the name “external variable” for variables whose state is not determined by the evolution of the model (or by communication with the control system in the case of outputs), refers to the use made of these variables in the simulation environment : their state may be modified by an external action.

In the description interface, these variables are said to be “undefined” as they appear in the description but their type, that is their origin, is not known.

1.3 Evaluating input logic equations and edges on variables

B

The description of a machine in SIMACTEL includes the representation of relays by Boolean equations.

Other variables, classed as relays in the internal operation of the simulation, can also be used for modelling purposes.

We know that the behavior of a PLC program depends on the order in which equations are written, as these are evaluated sequentially.

Thus it is important to know how the expressions representing these entities are evaluated in SIMACTEL, to be able to develop an error-free model.

SIMACTEL offers the possibility of using rising and falling edges on the description variables. Their use must therefore be specified.

1.3.1 Propagating state changes in the logic equations

In reality, changes in state which are propagated in the machine relays are made according to the relay opening and closing times.

In SIMACTEL, the switching times are considered negligible in relation to changes to the axis : the equations representing the relays are thus evaluated at a fixed simulation time, until a stable state of these elements is achieved.

Whereas in a PLC the equations are evaluated sequentially, the equations used in modelling the application are evaluated in “parallel” in SIMACTEL. This means that an evolution of a variable affects all the equations in which it occurs at the same time : the changes in state are propagated in “layers” of successive equations. This mode of evaluation is applied to description variables and relays.

For example, take the following equations :

X1 = (S1 + X1) . /R1

X2 = (R1 . X1 + X2) . /R2

Suppose that X1 = 1 and X2 = 0 initially and also that

S1 = R1 = R2 = 0

When R1 changes to 1, X2 will change to 1 and X1 will change to 0 simultaneously.

1.3.2 Edges on variables with SIMACTEL

The concept of an edge of a variable is one with which the control systems engineer is familiar. This concept exists in the standard definition of a Grafcet chart : a rising edge on an input is “true”, if the corresponding input is at logic state “1” and was at “0” during the previous execution cycle.
When modelling a machine, it is also possible to use edges on certain variables, in equations defining the description variables and relays on the one hand, and in combinations of sensors on the other hand.

• Using an edge in an equation

In the relays and variables, edges can be used on outputs, external variables (or “undefined” during the description phase), sensors and combinations of sensors and also on other relays or variables.

An edge on the variable <var> is declared by prefixing <var> with a “+” or a “-” depending on whether the rising or falling edge of <var> is used.

Note :

If the variable <var> is defined implicitly by the use of “+<var>” or “-<var>” in an equation, this variable will appear as an external variable during simulation.

• Evaluating edges

Although an edge may be seen by a PLC during a clearly-defined period corresponding to the cycle time, it is not the same in SIMACTEL (at least in terms of the edges used when modelling the application).

There is no equivalent concept to the “complete cycle” during which the variables would be updated sequentially : the equations are evaluated in successive “layers”, with equations in the same layer being calculated simultaneously.

In SIMACTEL, an edge used in the Boolean expressions in a model of the application is “true”, from the moment at which the corresponding change in state occurs, for the evaluation of the first layer of equations in which this edge is used.

Take the following equations :

b = +a

c = (b + c) . /d

where a = b = c = d = 0 initially.

If “a” changes to the true state, “b” will change to 1 for a brief moment, but will never be observed in this state. However, “c” will see this transient state and thus will be latched until “d” changes to 1.

B

• **Sequencing logic equations containing edges**

Take the following equations :

Mem-a = (+Set + Mem-a) . /Reset-a

Mem-b = (+Set . Mem-a + Mem-b) . /Reset-b

“Mem-a” is latched by the rising edge of “Set” and released by the true state of “Reset-a”.

To latch “Mem-b”, “Mem-a” must already be at 1 when “Set” changes to 1.

If “Mem-a” and “Mem-b” are at “0” initially, 2 successive changes of “Set” to “1” are required to latch “Mem-b”.

• **Application examples**

The “parallel” calculation of equations and the use of edges means that some interesting elements can be modelled, such as :

• a sequencer :

Xi = (+Vi . Xi-1 + Xi) . /Xi+1

Xi represents the logic state at step “i” of the sequencer. The rising edge of Vi causes the move to the next step “i + 1” if Xi is valid,

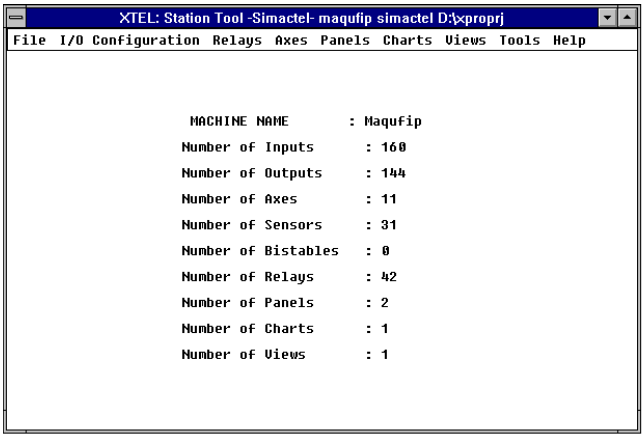
• a shift register :

Bi = +Cmd . Bi-1 + Bi . /+Cmd

Bi represents register bit “i”. The shift takes place when the variable “Cmd” changes to 1. To load the register, the above equation must be completed.

1.4 Main window of the description interface

The main window of the description interface comprises three zones. The top band accesses the description interfaces of the various parts of the installation. The central zone contains information about the machine or model being edited. The buttons for accessing the other SIMACTEL software functions (simulation, tools, etc) can be found at the bottom of the window. The role of each of these elements is explained below.



Buttons for accessing the description interfaces

Machine	Used to create, load and delete machine dossiers. When there is no machine loaded in SIMACTEL, the message "no machine selected" appears at the top left of the window.
I/O Configuration	See section 4 of this part.
Relays	See section 3 of this part.
Axes	See section 2 of this part.
Panels	See section 5 of this part.
Views	See section 6 of this part.
Charts	See section 3.1 of part C.
Models	See section 7 of this part.

B

Access to other functions	
Help	launches the online Help function on the SIMACTEL description interface
Tools	
Printing the machine	See section 2 of part D.
Saving to a floppy disk	See section 1.1 of part D.
Loading from a floppy disk	See section 1.2 of part D.
Copying a machine	See section 1.3 of part D.
Transferring a machine to a model	See section 7.1 of this part.
Tracing instantiations	See section 7.3 of this part.
Simulation	See part C.
End	Exits the SIMACTEL application.

2.1 Main axis description window

The list of axes comprising the description is given in this window. It has buttons for creating a new axis, and editing, renaming or deleting existing axes.

The time unit used for the configuration can be modified at this stage. The default unit is **seconds**, but this can be replaced by 1/100 th of a minute by clicking on the text “second” which appears in the window. This selection of the unit is important as it affects the representation of the simulation traces (see section 3, part D).

A button is used to signal any anomalies in the description. This function is described in section 2.8 of this part.

Movements are linked to a single axis and thus can only be edited via this axis. Deleting an axis deletes all its movements.

The description variables and bistables can be associated with more than one axis. Buttons are used to gain direct access to these entities.

The **Sensors** button is used to access all the sensors relating to the description. At this level, the user can determine with which axis a particular sensor is associated and open the window for editing this axis. The described elements for which this sensor is used can also be accessed. The sensor can be edited directly without having to return to the axis with which it is associated.

The **Undefined** button accesses the list of variables used at various levels in the description and which have not been defined.

These functions are explained in more detail in following sections.

2.2 Creating and editing axes

The creation of an axis starts with the entry of the mnemonic used to identify it.

An axis is named using the format ax-<no>, where <no>, by default, is the smallest integer, so that ax-<no> represents an axis which has not already been defined.

It is possible to replace <no> with another integer or with any string of up to 7 characters.

Once an axis has been created, or a request is made to edit an axis which has already been described, the axis configuration window appears.

This window is used to enter the individual axis characteristics and to access the variables associated with the axis.

2.2.1 Axis characteristics

The individual axis characteristics, entered directly in the edit window are as follows :

Type of axis

By default, the axis is linear, that is, evolutions are limited between a start and an end limit. This corresponds to the majority of actuators, such as cylinders.

An axis can be rotary, for example to describe an element in the application such as a turret tool-post with various different tools. This is the type of axis which is used when modelling the transfer of parts or which can be used to represent certain “information” axes.

Label

This is a string of up to 16 characters.

Length

This is a strictly positive number, expressed in the unit selected.

Cam width

By default, this width is zero. In practice this cam is hardly used : the activation width associated with each of the axis sensors is often preferred.

Initial position along the axis

The default position is at the start of the axis, but it can be at the end of the axis or determined as a function of one of the axis sensors. This third option can be slightly difficult and requires prior knowledge of how the sensors have been configured.

If the sensor used to initialize the axis according to the standard configuration suggested by the interface has been described, there are 3 different possibilities :

- if the sensor logic is positive or latch, the axis will be initialized after the (first) change to state 1 of the sensor,

B

- if the logic is negative, the axis will be initialized after the first time the sensor changes to 0,
- in the case of a release sensor, the axis will be initialized just before the sensor changes to 0 (or after the change to 1 when starting from the end of the axis).

However, if the sensor has been described by evolutions, initialization depends on the initial state of the sensor. If this is at 0 at the start of the axis, the axis will be initialized after the first change to 1.

If the opposite is the case, the axis will be initialized before the sensor first changes to 0.

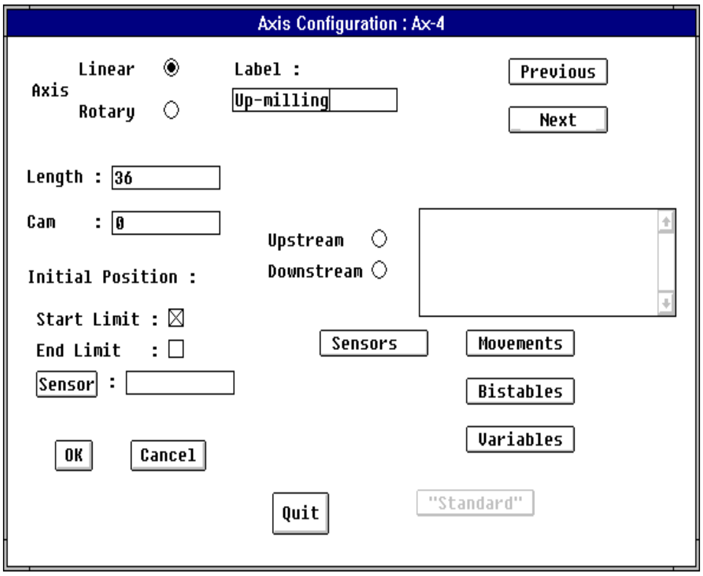
Confirming the axis characteristics

The **OK** button in the lower left-hand part of the window is used to accept the information which has just been described.

The **Cancel** button is used to return to the previous set of values which were accepted at the last confirmation.

Note

It can sometimes be useful to create a fictitious sensor in the axis, for the sole purpose of initializing it correctly.



B

2.2.2 Associations up and downstream of the axis

The sensors and movements are defined in relation to an axis : the association with this axis is thus immediate.

For other variables, the association with the axes is made according to the two following principles :

- the upstream association consists of associating with the axis the description variables used in defining the axis movements, except for the relays forming part of the general power supply (see section 3 of this part). When applied recursively, this association is used to pass on to the control system outputs,
- the downstream association consists of associating with the axis the description variables in which its sensors are used. As with the upstream association, it is applied recursively to the inputs.

The **Upstream** and **Downstream** buttons are used to display these variables in the window for editing an axis. If one of these variables is selected using the mouse, it can be edited, as long as it is one of the variables configured in the moving part.

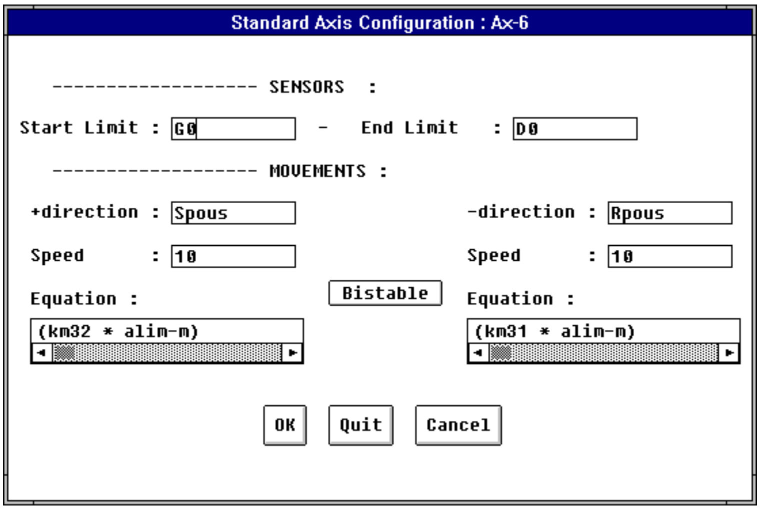
The buttons in the lower right-hand part of the screen open the windows for editing the sensors, bistables and variables linked to the axis.

2.2.3 Standard axis

The description of an axis can require the use of a large number of windows, in order to define the associated variables. This is why the concept of a “standard axis” has been introduced.

The **Standard** button in the lower left-hand part of the window is used to open a page describing a typical axis, such as a cylinder.

A standard axis includes a maximum of two sensors, which must be positioned at the start and end of the axis. A maximum of two movements (- direction and + direction) are associated with it. The **Bistable** button is used to complete this description, if required.



B

2.3 Creating and editing sensors

The Sensors button in the Axis Configuration window displays the main menu for editing the sensors associated with the axis. At this level, it is possible, as for all the variables linked to an axis, to create, edit, rename and delete sensors.

The windows for creating and editing sensors are similar. The differences are as follows :

- in the create window, the mnemonic of the created sensor must be entered. The Undefined button is used to select the name of the sensor from the list of undefined variables in the description, and then to identify one of these variables with the sensor which has been created,
- in the edit window, the mnemonic of the sensor cannot be modified : this can only be done using the Rename function. However, you can switch from one sensor to another without exiting the edit window.

  Two different ways of describing a sensor are offered in the configuration interface.

2.3.1 Standard configuration suggested by the interface

This is the default option. It consists of describing the sensor using the five following pieces of information :

The name or mnemonic

This can be taken from the mnemonics of the terminal block inputs, or from the undefined variables of the machine. The undefined variables and inputs are accessed via the corresponding button.
Clicking on one of these inputs fills in the sensor mnemonic field. The input will also be associated with the sensor, when the sensor is created, via its equation.

The position

This can be the start or the end of the axis, or a numeric value. In this latter case, the user should ensure that the start or end of the axis options are no longer selected, which can be done by confirming the value entered by pressing “Enter”.

The logic type

1. positive logic : the sensor is active (its logic state is 1) when the cam acts upon it,
2. negative logic : the state of the sensor is the opposite of the previous case,
3. latch : the sensor changes to the active state from its position and remains so until the end of the axis,
4. release : operation is the exact opposite of latch.

The activation width

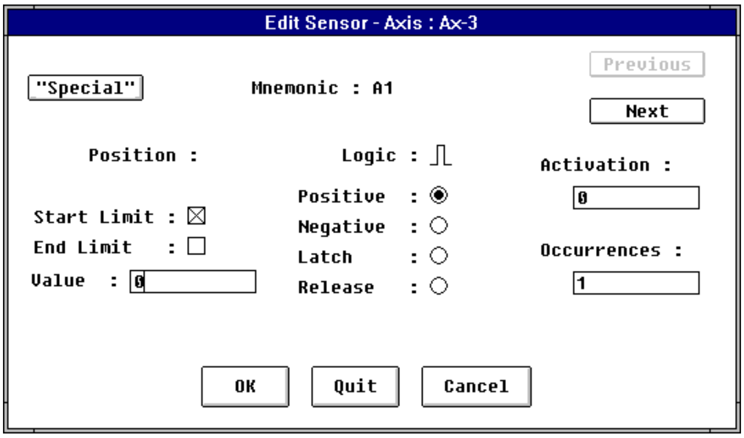
This information is the length over which the sensor is active (positive logic) or inactive (negative logic). By default, the sensor “inherits” the width of the cam associated with the axis.

The number of occurrences

This is 1 by default, and is used to repeat the sensor several times along the axis at regular intervals. This can be useful, for example in the case of a rotary axis activating a sensor via cams distributed at regular intervals. This function is only active if the type of logic is positive or negative.

Note

The activation width is different from the concept of the “cam” attached to the axis. If a linear axis has a cam which is not zero, its length is reduced accordingly and the position of each of its sensors is advanced, as it is considered that the movement of this cam is “internal” to the axis.



2.3.2 Describing a sensor by evolutions

The “standard” way of describing a sensor may be unsuitable in certain cases. This is particularly true when an axis sensor is activated by cams which are of variable widths and/or placed at irregular intervals.

This problem can be avoided by configuring a sensor as required.

By clicking on the **Special** button in the top left-hand corner of the Edit Sensor window the content of this window can be modified.

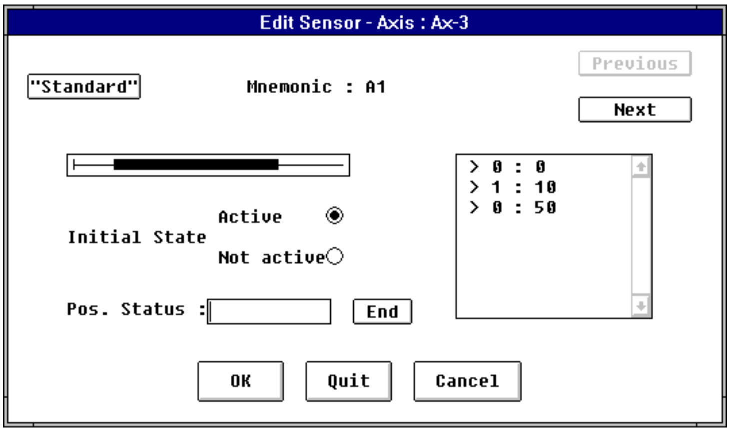
If the data has been entered in “standard” mode, it is displayed here in the form of a trend diagram on the one hand, and a series of logic evolutions of the sensor on the other hand, in a scroll-down list on the right-hand side of the window. The initial state is indicated via a 2-position selector switch.

The sensor configuration consists of giving the series of evolutions affecting the sensor starting from an initial state. These evolutions are entered via the editor at the bottom of the window.

Pressing the “Enter” key for confirmation adds the evolution position to those displayed in the scroll-down list, if the value entered is a positive number smaller than or equal to the length of the axis. The trend diagram representing the sensor evolutions is simultaneously updated.

The End button to the right of the editor is used to display the length of the axis with which the sensor is associated. This is used to set an evolution at the end of an axis after confirmation, or simply to recall this length.

An evolution is removed by clicking in the scroll-down list with the mouse. The trend diagram is updated automatically.



B

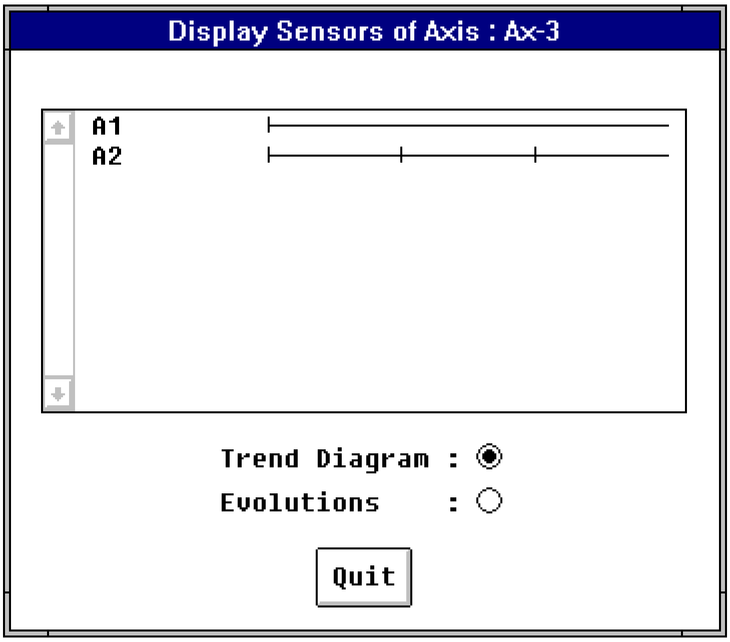
2.3.3 Checking the sensor description : display

The display button in the sensor edit menu is used to display a window showing a trend diagram of the evolutions of the axis sensors.

To check if the sensor configuration created corresponds to the desired configuration, a **Display** function is available.

To gain a more precise view of the positions at which these sensors change state and of the order in which they occur, the user can also display the succession of evolutions of these sensors all the way along the axis.

Note
At this level of representation in the form of a trend diagram, it is possible to edit the sensors by clicking on their name using the mouse. You can also correct an incorrect configuration.



B

2.4 Creating and editing movements

The movements button in the axis configuration window displays the main menu for editing movements associated with an axis. It is possible to create, edit, rename and delete movements.

Editing a movement consists of declaring :

The name or mnemonic

This can be entered in the editor or taken from the mnemonics of the terminal block outputs. The list of outputs is accessed via the “Outputs” button.
Clicking on one of these outputs creates the movement associated with the name or mnemonic and the address of the output for the equation.

The direction of the movement

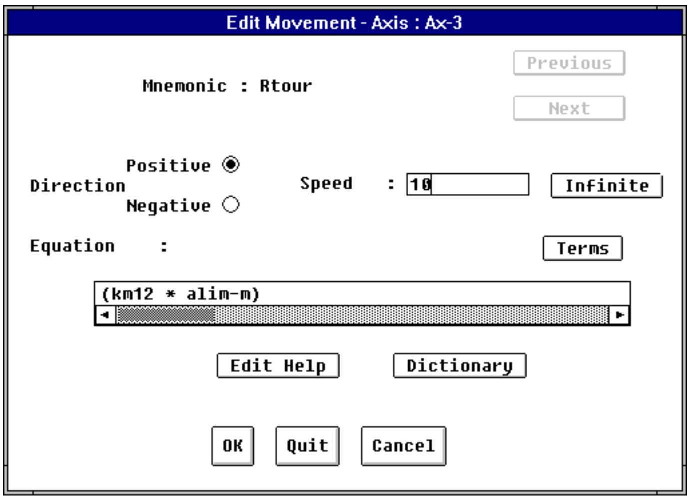
A movement with a positive direction causes movement towards a higher position on the axis (towards the end of the axis) when it is activated. A movement with a negative direction causes movement towards the start of the axis when it is activated.

The speed

This determines the duration of the movement of the axis between 2 sensors. An infinite speed leads to a zero movement duration from one sensor to another.

The logic equation

This determines the state of the movement. It is a logic function of description variables, relays, sensors, outputs, or numerical comparisons.
There are two buttons in the Edit Movement window : Previous and Next. These are used to recall in the editor dedicated to the entry of the movement equation the movement equations already defined for the axis. Since these equations generally use the same terms, this avoids retyping an equation in full : it is often simply necessary to add variables to give to the new movement equation.



2.5 Creating and editing variables

The **Variables** button in the Edit Axis window displays the main menu for creating, editing, changing the name of and deleting variables associated with the axis.

For logic variables, the following elements may be used in the equations :

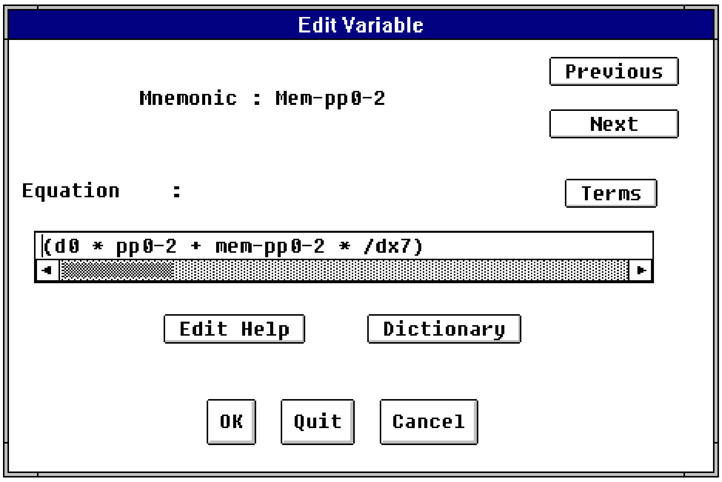
- sensors,
- operator panel variables,
- external variables,
- relays,
- outputs,
- other logic variables.

Help available when editing variables

For the list of variables created, the Undefined button is used to select a variable from those listed in the description.

For description variables and bistables, this selection is limited to the undefined variables upstream of the axis, used for example in the axis movement equation.

The Edit Help function is used to open a window from which the variables likely to be used in an equation can be selected. They can be transferred to the equation entry editors.



2.6 Creating and editing bistables

B

The SIMACTEL bistable description window contains the following elements :

The mnemonic :

This is the name of the bistable. This name is used in equations of the elements downstream of the bistable, notably movements controlled by the bistable.

The latch term :

This is the name of a description element upstream of the bistable. The bistable takes the state “1” on the rising edge of the latch term when the condition is verified. It keeps this value as long as the release has not been triggered, even if the latch term returns to state “0”.

The release term :

This is the name of a description element upstream of the bistable. The bistable takes the state “0” on the rising edge of the release term when the condition is verified. It keeps this value until a new latch has been triggered, even if the release term returns to state “ 0 ”.

The condition :

This is the name of a description element upstream of the bistable. The bistable cannot change state if the condition has not been verified. Generally, the condition is a term representing the power supply.

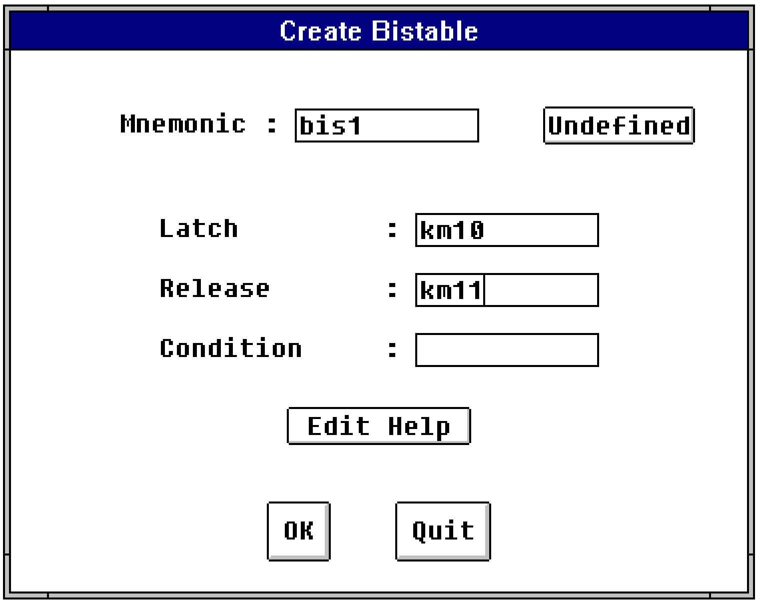
Undefined button :

For the list of variables created, the **Undefined** button is used to select a variable from those listed in the description. This selection is limited to the undefined variables upstream of the axis, used for example in the axis movement equation.

Help button :

The **Edit Help** function is used to open a window where the user can select variables which could be used in a logic equation. They can be transferred to the equation entry editors.

The **OK** and **Quit** buttons are used to exit the bistable creation and modification windows.



B

2.7 Accessing “undefined variables”

The concept of the “external variable” was introduced in the first section of this part.

In summary, if, for example, the state of the sensors depends on the position in a axis or if the state of the relays is determined by a Boolean expression, the logic value of the external variables is not defined internally by the evolution of the model.

These variables represent “modelling limits” corresponding to external conditions such as the power supply, the presence of parts entering the machine or sensors signalling the opening of a door.

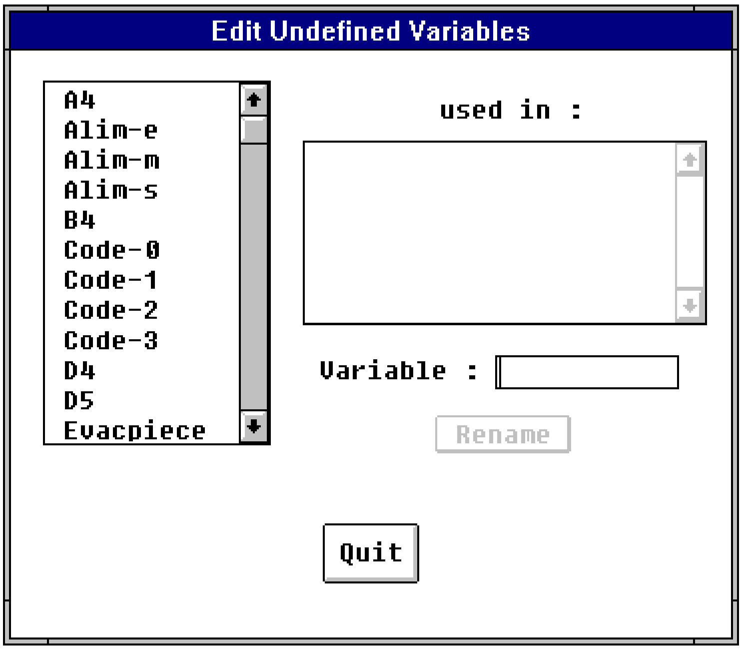
In the simulation environment, the state of these variables will be determined by an external action.

During configuration, variables can only be “temporarily undefined”. For example, if the description starts with the definition of the “I/O Configuration” (see section 4 of this part), the sensors used in the input logic equations are “undefined variables” as the axes activating these sensors have not yet been described. The insertion of one of these variables into an equation is signalled by a message.

When a modelling entity is created, the list of existing undefined variables can be accessed to name the new element.

However, the existence of “undefined variables” in the description can also result in a data entry error in the definition of an element. In order to solve this problem, the Undefined button in the first window of the axis configuration interface is used to open a window giving the list of these variables used in the description.

Selecting one of these variables displays the elements of the description in which it is used. If one of these elements is selected using the mouse, the edit window for that type of element appears, and is used to make a correction, thus removing the "undefined variable" which is not required.



B

2.8 Checking the axis description

The **Check** button in the main Edit Axis window is used to activate a consistency check of the description of the application.

The following anomalies are signaled :

- axes without sensors,
- linear axes which do not have at least one movement for each direction,
- rotary axes without a movement,
- unused variables.

B

3.1 Introduction

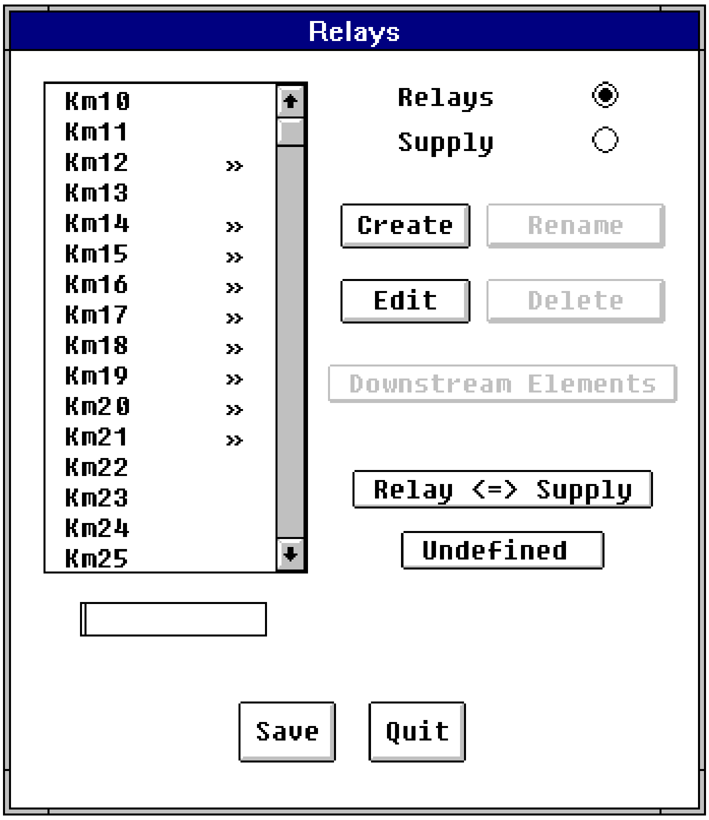
The “Relay Interface” is made up of the relays used in the machine to be simulated.

In SIMACTEL, the description of these relays consists of transcribing the electrical diagrams available in the documents in the form of Boolean expressions.

A distinction is made in the configuration interface between the “Supply” and the “Relays”. The relays used in the power supply correspond to the general machine functions such as electrical power distribution, powering up the various subsets, etc, and are not linked directly to the actuators.

The “Relays”, however, are used directly in the control of movements.

This distinction does not affect the operation of the simulator, but is important when associating the description variables with the machine axes.



3.2 Creating a relay

B

To create a relay, a name is required. This is done via an editor accepting a maximum of 10 characters.

The relay described by the user may have already been used to define an element in the model (for example another relay). In this case, an undefined variable exists with the name of the relay which is to be created : the Undefined button is used to find this name. The name of the relay which the user wishes to create can be the mnemonic of an output described in the I/O configuration. The **Mnemonics** button accesses the list of outputs with their mnemonic. The user can then select the name of the relay to be created.

It is possible to associate a time delay with a relay, by checking the **Delayed** box in the window for creating these elements.

The duration of this time delay must then be specified : its value is entered via an editor. The unit is that which was selected by the user for the machine configuration in the axis creation window.

By default, the relay is an on-delay, which means that the time delay contact appears some time after the relay logic equation has been checked.

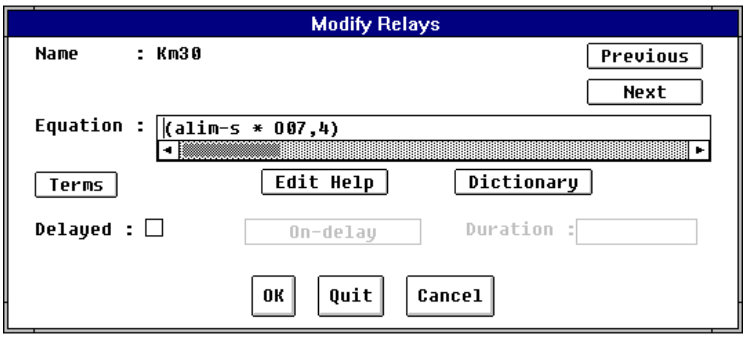
By clicking on the text **on-delay** in the creation window, the relay will be an off-delay, which means that the time delay contact will be released just after the relay logic equation becomes “false”.

The creation of a relay is accepted once the OK button has been pressed.

The creation window stays on the screen. If the name of the relay is modified, this creates a new relay.

The **Edit Help** button is used to select the variables which the user wishes to use in the relay logic equation.

The **Dictionary**, which is accessed via the button with the same name, is used to memorize words or sets of words, for reuse in other equations. This makes it easier to enter other logic equations which may contain the same parts.



3.3 Editing relays

In the edit window, it is possible to modify the characteristics of the relay, including the equation and the time delay which may be associated with it.

3.4 Taking account of time delay contacts in relay equations

We have seen that time delays can be associated with relays in the relay interface. This is only useful if it is possible to use the corresponding time delay contacts in the relay equation.

For on-delay relays, the true state of the time delay contact is fax-<relay> where “<relay>” is the name of the relay with which the time delay is associated.

The active state of the time delay contact of an off-delay relay named <relay> is /dax-<relay>.

This comes from the internal representation of time delays. They are represented by axes named ax-<relay>. fax-<relay> and dax-<relay> are variables, automatically associated with these axes, which change to the “true” state as soon as the axes arrive at the end and start limits respectively.

Example :
An on-delay relay Km1 is declared, with a duration of 10. As soon as the equation associated with this relay is checked, the simulation date will advance by 10 units so that the variable fax–km1 representing the time delay contact becomes "true".

B

4.1 Introduction

In SIMACTEL, the I/O configuration describes the interface between the modelling of the application and the control system to be validated, which in this case is a program executed in a PLC.

The description of the I/O configuration thus comprises the configuration of the in-rack and remote I/O modules. The user must also define the logic equations of the inputs activated by the simulator according to the machine model variables.

4.2 Describing the I/O cards

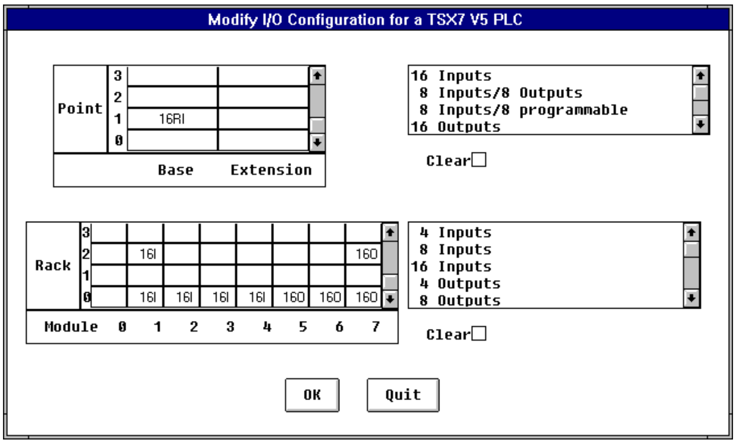
The description of an I/O configuration starts with the configuration of the I/O modules, each of which is designated by a code which indicates whether the module in question is to be simulated by SIMACTEL.

The configuration window is divided into 2 parts :

- The in-rack I/O are shown at the bottom of the screen. The grid represents the PLC modules and racks. The various permitted I/O cards are displayed on the right-hand side of the screen. The check box to the right of the text **Clear** is used, when it is selected, to remove a card from the configuration.
- The remote I/O are shown at the top of the screen. The grid represents the connection points. The various types of TBX module are displayed on the right-hand side of the screen.

Note :

- The inputs which are not simulated are not generated in **SIMACTEL**.
- In the case of a TBX module, only its type is declared (16 inputs, 16 outputs, 8 I/O, 8 inputs/8 programmable).
- For programmable TBX modules, the type of channel (I/O) must be entered for each channel (8 to 15) by selecting the corresponding button.



4.3 Editing the I/O

Configuring the I/O defines the I/O addresses. To associate a mnemonic and a comment with the I/O, it should be edited using the button provided for this purpose.

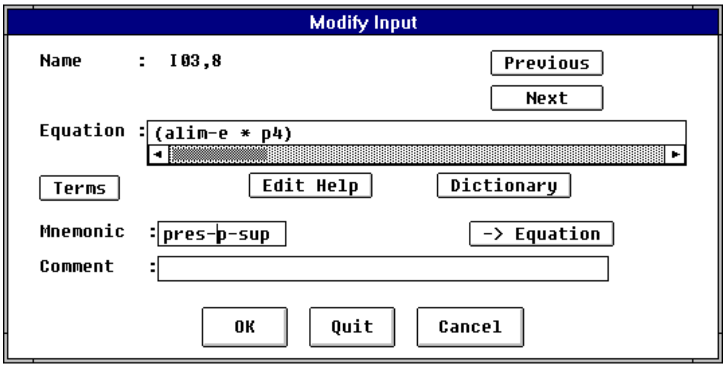
This edit function is also used to define the input equations. A Help function is provided (the **Edit Help** button) to help users when entering these equations.

This button is used to open a window from which the variables likely to be used in the input equation can be selected.

It is not possible to use an output in an input equation. Such a loop would require that an intermediate variable be defined. This can be a relay from the relay interface or a moving part description variable.

Note :

It is not possible to create or delete I/O without returning to the I/O configuration. This configuration is obtained when creating the I/O configuration or using the button **Modify I/O Config.** if you are only changing a few I/O cards.



B

4.3.1 NIOF gateway

“Neutral I/O files” have been defined to enable standard exchanges of data between the electrical wiring CAD software and the documentation software of the programmable automated systems.

The SIMACTEL configuration interface offers the possibility of reading these files, which enables a PLC I/O configuration, with the associated mnemonics and comments to be retrieved. These files can come from the XTEL-CAD module for example.

The NIOF file is generated from XTEL-CAD as follows :

- Under the XTEL-CONF tool, select "XTEL—>CAD" from the "File" menu to generate the .IOC and .IOF files.
- Under the XTEL-CAD tool, select "XTEL—>CAD" from the "File" menu to generate a file in NIOF format.

Warning : the file must be in the directory "UNITE:\XPROJPRJ\PROJECT\STATION\APP.

To read one of these files, click on the “Neutral File” button in the I/O configuration window or in the window for creating a new I/O configuration. The name of the file to be read must then be entered, and confirmation launches processing provided that this file exists.

The NIOF gateway can be called after the I/O have been configured. It is used to update each I/O mnemonic and comment. It can also be called during the I/O configuration. In this case, it also creates the automatic configuration of I/O cards.

Thus, a neutral file can be read with or without updating the I/O configuration. With the option, the I/O are configured with the types of card read in the file. Without the option, the mnemonics and comments of the existing I/O are updated.

4.3.2 SDBASE import

The **SDBASE** station tool is used to convert (export) to files containing the I/O mnemonics and comments (.SCY text files).

This file is generated from XTEL-CAD as follows :

- Under the XTEL-SDBASE tool, select "Convert\Symbol Database to Text File" from the "File" menu, then select the standard format (.SCY), and give the file name.

The **SDBASE File** button in the I/O configuration window is used to retrieve these mnemonics/comments, by giving the name of the corresponding file.

5.1 Introduction

The configuration of an operator panel consists of defining its “geometry”, or in other words, of positioning its various elements in relation to each other.

Each of the elements is described individually. The graphic representation of these elements includes “labels” which are used to identify and name the various positions of the switches.

Variables are associated with the pushbuttons and switches. These variables are activated according to the actions performed on these elements. They are used in the input logic equations or in the relays of the relay interface.

The indicator lamps and displays are controlled according to the outputs or relays and, if required, according to the moving part description variables.

5.2 Creating and editing operator panels

To create an operator panel, it must first be given a title, which may be changed later.

In the editing stage, the operator panel appears in the top left-hand corner of the screen. All the control buttons corresponding to the edit functions are on the right-hand side of the screen.

5.2.1 Creating an element

Selection of the type of element to be created opens the corresponding window (see section 5.3 of this part “defining the operator panel elements”)

Once the data entered has been confirmed, the new element is positioned in relation to an element which already exists in the operator panel selected using the mouse : this reference element then changes to reverse video.

5.2.2 Modifying an element

Once an element has been selected from the operator panel being edited, the “Edit Element” button is used to call up the window specific to this type of element.

The associated labels and the variable(s) can also be modified in this way.

B

5.2.3 Deleting and restoring an element

An operator panel element, previously selected using the mouse, can be deleted using the appropriate button. The “Restore Element” button is used to cancel the last deletion made. The restored element must be repositioned in the operator panel.

It is possible to change the position of an element in an operator panel. The operator panel is first redisplayed without the selected element which must be repositioned in relation to a reference element.

B

5.3 Defining operator panel elements

We have just seen the general principle for configuring operator panels. This section describes each of the individual elements in more detail.

5.3.1 Pushbuttons

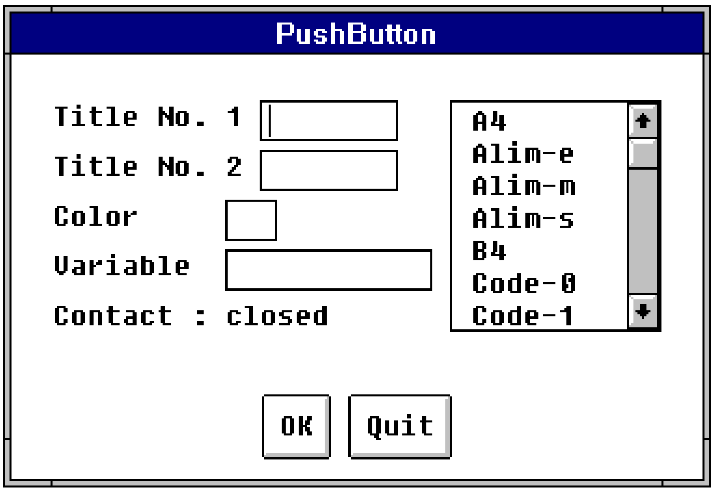
A pushbutton is identified by a title made up of 2 superposed strings, each of 6 characters.

The color field is, in fact, a letter - the initial letter of the color of the button. The pushbutton will then appear in this color.

The variable activated by the button is entered using an editor. Since this variable may already have been used in an input logic equation or in a relay from the relay interface, it is likely to be found in the list of “undefined” variables for the description : this list is therefore displayed in the window for editing the pushbuttons.

By default, the button contact is normally open : the logic state of the associated variable changes to “true” when the button is pressed. This operation is reversed by clicking on the text “normally open” : this changes to a “normally closed” contact.

The “wide” pushbutton operates in the same way as a standard pushbutton. Since it is twice the width, the title strings can be a maximum of 12 characters.

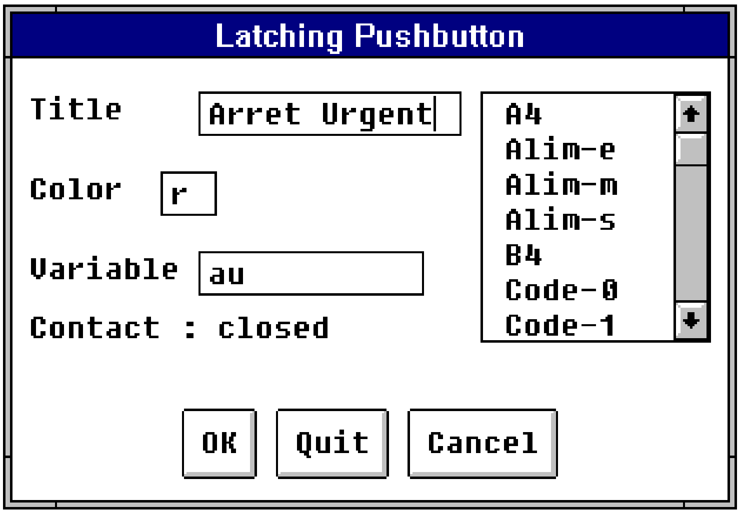


5.3.2 Latching pushbuttons

B

These buttons are used to represent elements such as “mushroom head” emergency stop buttons, which remain depressed and sometimes require a key to release them. These buttons have a title made up of a single string of up to 12 characters.

As with the simple pushbuttons, the contact can be normally open or normally closed.

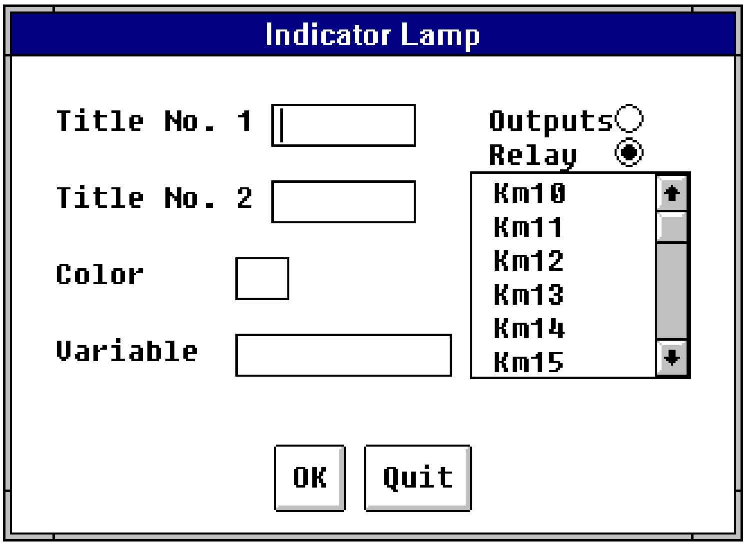


B

5.3.3 Indicator lamps

In the same way as pushbuttons, the title identifying these elements is made up of two strings of 6 characters.

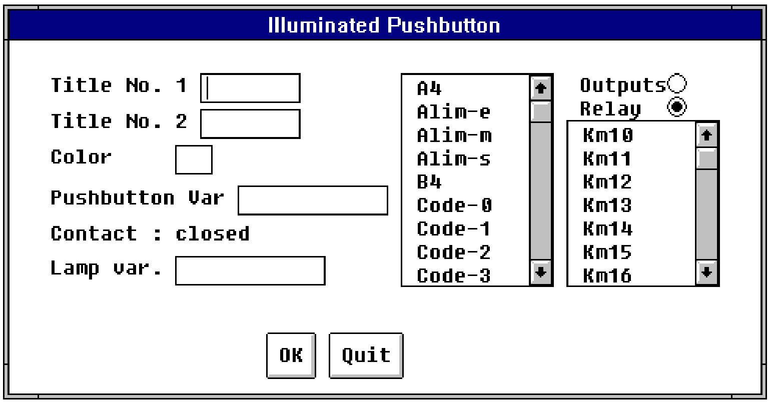
Since the variable which lights up the indicator lamp is usually an output or a relay, these entities can be accessed. In the same way as for the pushbuttons, the “wide” indicator lamps are twice as wide in the operator panels, and their title strings can be up to 12 characters.



B

5.3.4 Illuminated pushbuttons

These elements combine the “pushbutton” and “indicator lamp” functions. Thus, their configuration involves both these elements.



B

5.3.5 2-position switches

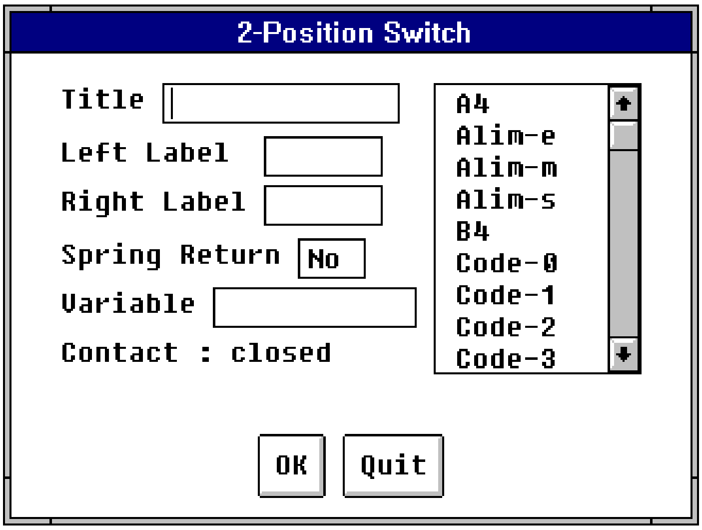
Switches occupy the space required by two “standard” pushbuttons in operator panels.

They have a 12-character title and each position is identified by a label of up to 5 characters.

If the “Spring Return” function is selected, the switch will return to the off position, to the left, as soon as it is released. This is indicated by an arrow in the graphic representation of the element.

By default, the associated variable is activated when the switch is in the right-hand position. This can be reversed by replacing “normally open contact” with “normally closed contact”, in the same way as for the pushbuttons.

The variable can be selected from the list of “undefined variables” for the description.

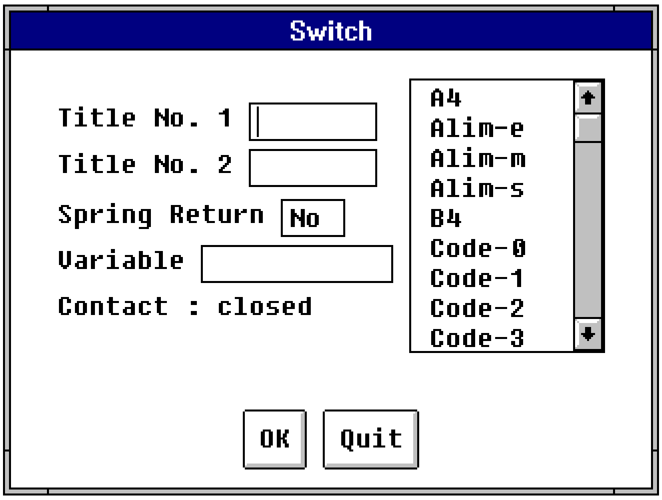


B

5.3.6 Switch

This element takes up the same space as a simple pushbutton, its title is also made up of two strings of 6 characters.

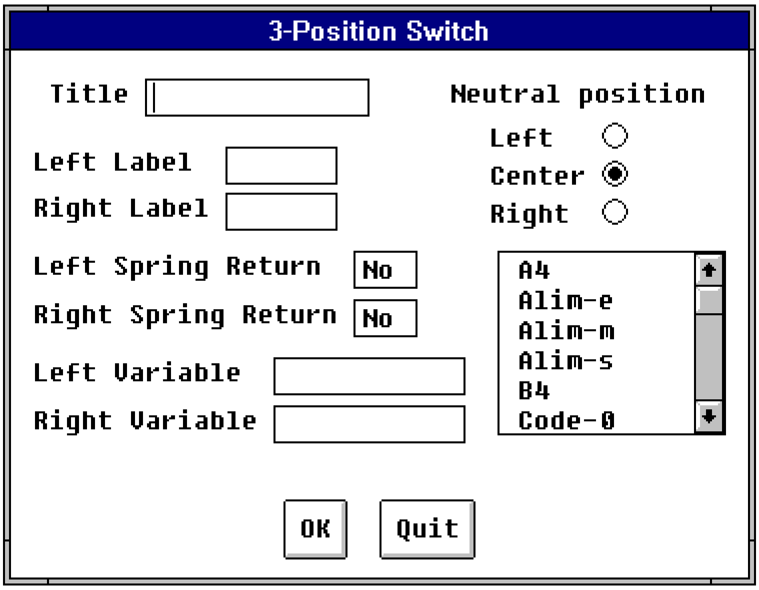
It operates in a similar way to the 2-position switch. It can also have a spring return function.



5.3.7 3-position switches

In these switches, two variables known as “right” and “left” can be activated.

By default, the neutral position, where neither of these variables is at logic state 1, is when the switch is in the central position. This position can be changed using the selector switch provided for this purpose. If the neutral position is the central position, the right and left positions can spring return to the centre.



5.3.8 N-position switches

A certain number of variables are associated with each position of this switch. These variables will be at logic state “1” if the position is selected.

By default, this switch has 2 positions. A new position can be added : this will be n+1, if n is the number of switch positions.

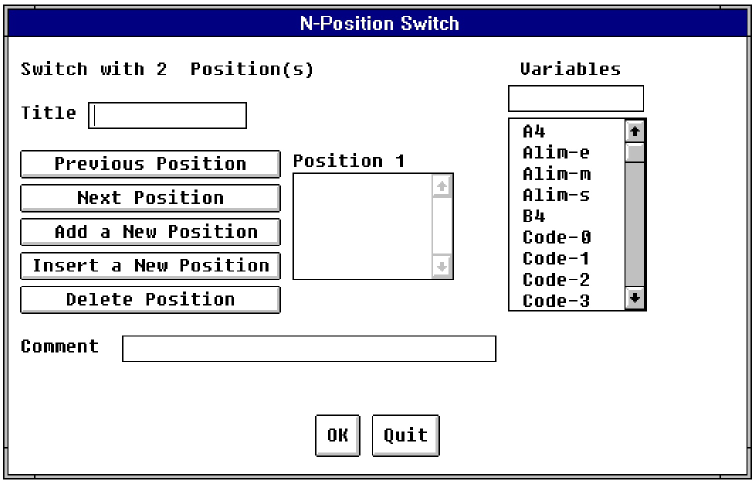
The insertion corresponds to a shift in the existing positions, from that which has been edited.

The position is changed using the “Previous Position” and “Next Position” buttons. It is possible to delete the current position.

The variables associated with the current position appear in a scroll-down list.

To add a variable to the current position, you can either enter it in the editor in the top right-hand corner of the N-position switch configuration window, or select it from the list of “undefined” variables, which is displayed below the editor.

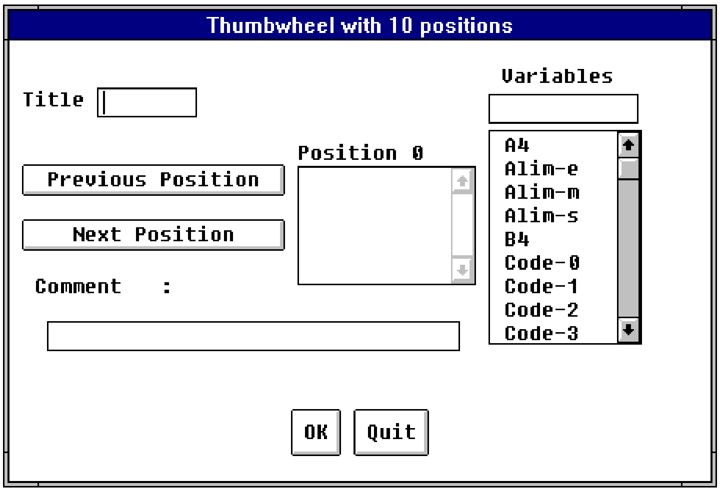
A variable is removed from the current position by clicking on it in the scroll-down list, under the text “Position <i>”. A comment can be associated with each of the switch positions. These comments will be displayed, in the operator panel, to the right of the switch.



5.3.9 Thumbwheels

These elements are configured in a similar way to the N-position switches. The number of positions is set here at 10, and these positions are numbered 0 to 9.

The thumbwheel takes up a slot half the size of the N-position switch, and its title is restricted to 6 characters.

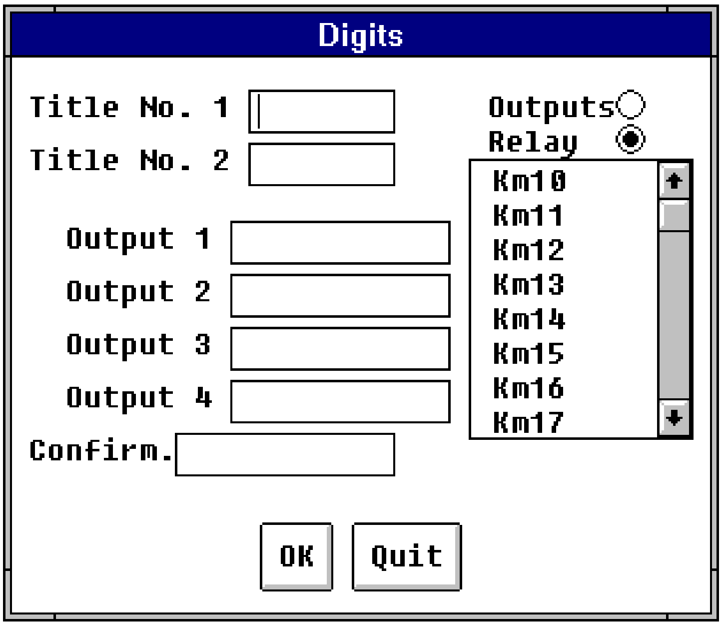


5.3.10 Hexadecimal displays (or “digits”)

These elements make it possible to display a hexadecimal value in an operator panel.

The value displayed is given by 4 bits which can be outputs or relays.

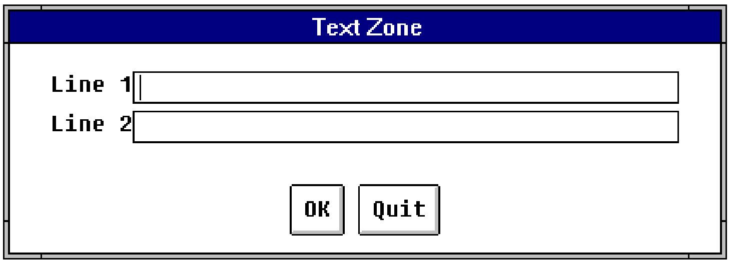
Validation is optional. If it is defined, the state of the 4 bits is only displayed if the validation variable is at “1”. The display can be validated on a rising or falling edge of a variable. In this case, it must be prefixed by “+” or “-”.



B

5.3.11 Text

This is used to insert one or two lines of comments in an operator panel.



5.3.12 Separator

The separator is used to insert "blanks" in the operator panel. These empty spaces are the same size as a "standard" pushbutton.

5.4 Duplicating operator panels

Since the description of a machine can include similar operator panels, it is possible to duplicate an operator panel which has already been described.

The variables associated with the elements are retained, and must be modified by editing the elements of the duplicate operator panel.

5.5 Cutting operator panels

This function is used to extract elements from one operator panel to create another. First select the elements to be extracted from the operator panel, then name the operator panel which will be created using these elements. The elements are positioned exactly as they were in the original operator panel.

B

5.6 Merging operator panels

This is used to add to an operator panel elements from other operator panels. Since the operator panels are made up of lines of elements, it is possible to merge them.

It is possible to add one operator panel to another in the following ways :

- above :

  The lines comprising the operator panel which has been added are placed above those of the original operator panel,

- below :

  This is the opposite of that outlined above,

- to the right above/below :

  The lines comprising the operator panel which has been added are placed at the end of those of the original operator panel, starting with the lines at the top or, alternatively, placing them in such a way that the last lines of each operator panel are next to each other,

- to the left above/below :

  The lines comprising the operator panel which has been added are placed at the beginning of those of the original operator panel.

5.7 Operator panel variables

The Variables button accesses all these variables, whose state depends on the pushbuttons and switches described in the operator panels.

Selecting one of these variables indicates to which element of which operator panel it is attached.

The Panel button is used to edit the operator panel concerned. The Element button is used to edit directly the element of the corresponding operator panel.

It is also possible to edit the entities of the description in which they figure using the Downstream Elements button.

5.8 Checking consistency

The consistency check consists of establishing that the variables associated with the operator panel elements are correct, that is that the indicator lamps and displays are controlled by outputs or relays and that the variables activated by the pushbuttons and switches are “undefined”.

If the variables are activated by different pushbuttons or switches (which is systematically the case after a panel has been duplicated), this anomaly is signaled.

5.9 Printing operator panels

This command is used to print an operator panel on the default WINDOWS printer. SIMACTEL produces a document made up of :

• A copy of the image of the operator panel selected,

• A list of the constituent elements.

Note :

Before printing an operator panel, the user should ensure that this operator panel can be displayed as a whole on the screen. If this is not the case, only the part displayed will be printed.

The SIMPRINT software, dedicated to documenting the SIMACTEL machine dossiers, can also be used to print operator panels (see section 2 of part D).

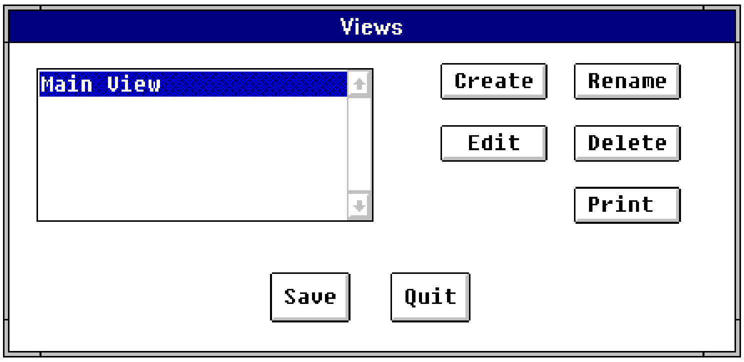
B

6.1 Introduction

The views associated with a machine are used to follow the evolution of the variables and axis movements.

A view is made up of a background on which various graphic elements relating to the machine variables can be arranged.

One of the views appears in the main window of the simulation environment. By default, it is the first of the views listed in alphabetical order. This view is characterized by having no title.



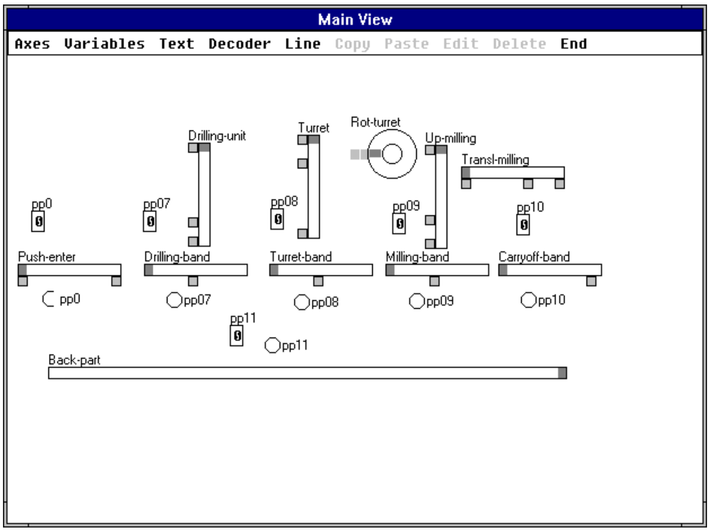
6.2 Creating and editing views

B

To create a view, it must be given a title, which may be changed later.

A special case is the blank title : the view designated in this way will appear in the simulation window. It appears at the top of the list of views described under the heading “Background”.

You then move on to the edit screen. The buttons for selecting the type of graphic element (axes, variables, texts, decoders) appear at the top of the screen, along with the buttons for editing and deleting an element from the view.



B

6.2.1 Axes

Select the axis to be represented from the machine axes. Then specify the axis characteristics (layout, reference point, size) which will be used in the view.

The layout is selected from the corresponding icons. This can be vertical, horizontal or circular. The position in the axis can be represented by a cam moving along the length of the axis.

An axis can be represented by a bargraph if the position on the graph is interpreted as being a level. In this case, the sensors are represented by arrows.

Selecting the reference point of the axis (to the left/right or above/below) determines the “0” of the coordinates. For example, a horizontal axis whose reference point is selected to the right and initial position is 0 will have its cam drawn to the right.

The size of the axis is fixed by a proportionality coefficient between 0.5 and 5. By default, it is set to 1.

When the selections relating to the axis representation have been made, it is possible to insert the axis in the view using the mouse by clicking on the location in which it is to be placed.

The last axis to be created (or drawn with the mouse) can be moved using the mouse (by pressing the “Ctrl” key on the keyboard at the same time). It can be deleted from the view by pressing the **Delete** button. The size can also be modified (“+” key to increase, “-” key to decrease).

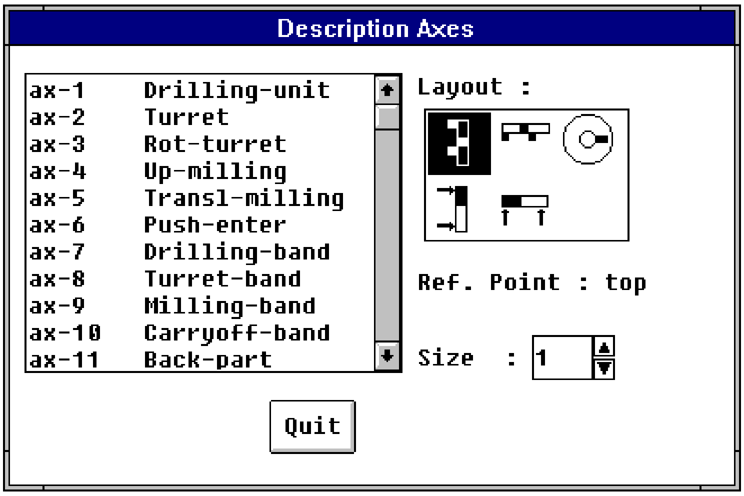
• Representing the sensors associated with the axis

To avoid overloading the representation of the axis in the view, only the sensors used in the logic equations for the inputs in the I/O configuration or the relays in the relay interface are shown.

Sensors which are only used in the description variables on the axis movements are considered to be fictitious and thus are not represented.

If the view is described before the I/O are configured, the axis sensors will not appear in the representation. This is particularly relevant if you wish to perform an internal simulation using a “mechanical Grafcet chart”.

In this case, in the simulation environment, the axis sensors used in the Grafcet chart transition conditions will appear in the view : these sensors are actually used as implicit inputs.



6.2.2 Variables

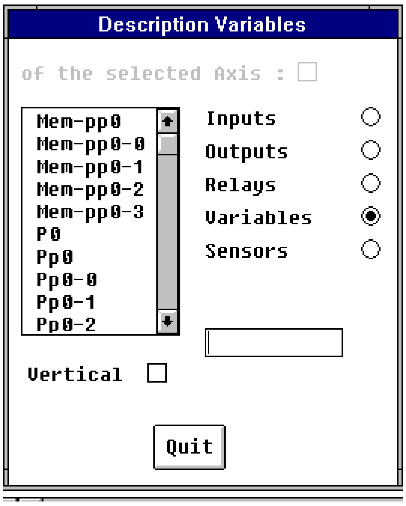
A machine variable can be represented using a “radio button” type icon, which, in simulation mode, will reflect its current state. To add a variable, first select the Variables button if the **Description Variables** window is not already visible.

The right-hand side of the window is used to select the type of variable from the inputs or outputs in the I/O configuration, the relays in the relay interface, the description variables for the moving part, and the traps. This selection is made by simply highlighting the variable in question in the list displayed using the mouse, then by selecting the point on the screen at which this variable is to appear (name + icon).

If one of axes represented in the view is selected using the mouse, these variables are restricted to those which are used in the associations upstream and downstream of the axis in question. To return to all the machine variables, either click again on the axis representation (or any element in the view) using the right-hand mouse button, or click in a zone on the view which does not include any elements.

It is possible to select a vertical representation of the variable (check box).

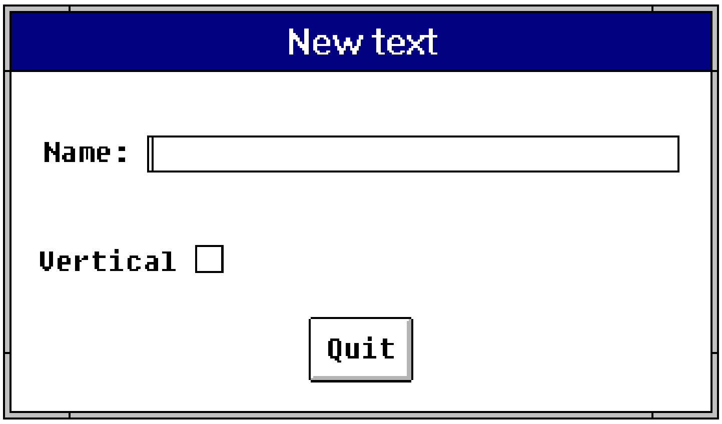
In the same way as for the axes, the variable can be moved on the screen, or deleted.



B

6.2.3 Text

Free text can be placed in the view by positioning it using the mouse. It is possible to display it vertically (check box).

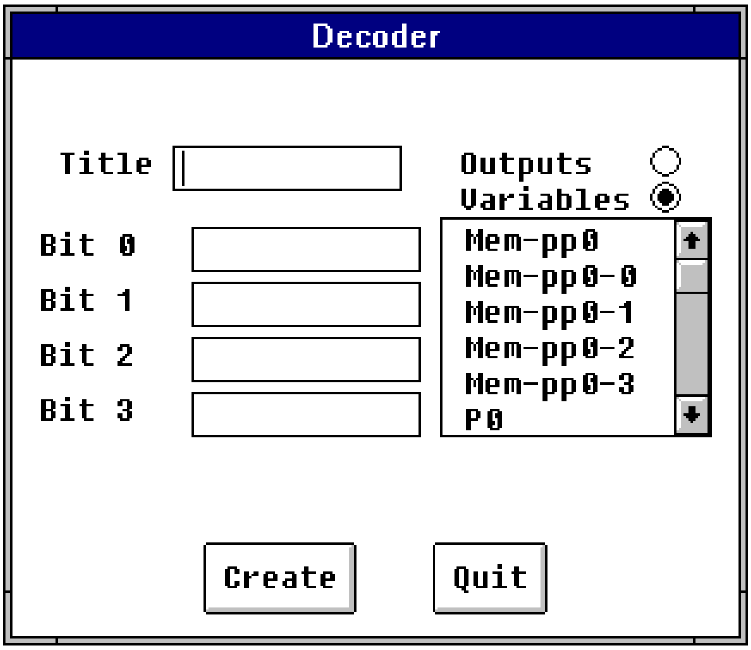


6.2.4 Decoders

In the same way as with the operator panels, hexadecimal displays can be used in the view.

The value displayed is given by 4 bits, which can be outputs or description variables.

B



B

6.2.5 Lines

Fixed lines can be added to enhance the views.

6.2.6 Copy/Paste

A copy/paste mechanism is available :

1. select the part of the view to be copied, using the left-hand mouse button by drawing a box around the zone to be copied,
2. click on the Copy button,
3. click on the Paste button then click on the required position in the view.

6.2.7 Moving elements

It is possible to move elements in the view by clicking on them and moving the mouse without releasing the button, or by using the arrow keys on the keyboard (movement pixel by pixel).

6.3 Printing views

This command is used to print a view on the WINDOWS default printer. The document created by SIMAC corresponds to a simple copy of the image of the view selected on-screen.

Note :

Before printing a view, the user must ensure that this view is displayed in full on the screen. If not, only the part displayed will be printed.

The SIMPRINT software, which generates the SIMAC machine dossier, is also used to print the views (see section 2 of part D).

7.1 Presentation

A machine can be made up of a number of similar parts, where the same subassembly may be common to several different machines. It can therefore be useful to be able to copy these “standard models” easily, thus avoiding laborious re-description and the possible introduction of errors.

Also, saving any particular information relating to the description of elements requiring a significant amount of programming can be useful.

For these reasons, the concept of a “model” has been introduced in the SIMACTEL configuration interface.

The use of this function involves two aspects : the definition of models archived in a library, and the incorporation of a model in the description of a machine.

This library is common to an XTEL project. The models defined in a station can be accessed by another station in the same project.

7.2 Defining models

A model is made up of the same components as a machine : axes, relays, operator panels, views, with the exception however of the I/O configuration. An I/O configuration requires a type of PLC to be defined, whereas a model must be general and should not depend on a particular address format.

However, a model can include the definition of the input logic equations, as the designation of these inputs is totally free. A model does not include the explicit definition of outputs, thus these variables are “undefined variables” in a model. When the model is incorporated in a machine, these variables will be identified with the outputs defined in the I/O configuration of this machine.

If the type of model requires that certain undefined variables in the model must be identified with outputs, this constraint can be integrated in the definition of the model, as we will see later.

B

7.2.1 Editing a model

A generic model is edited using the same interfaces as a machine description.

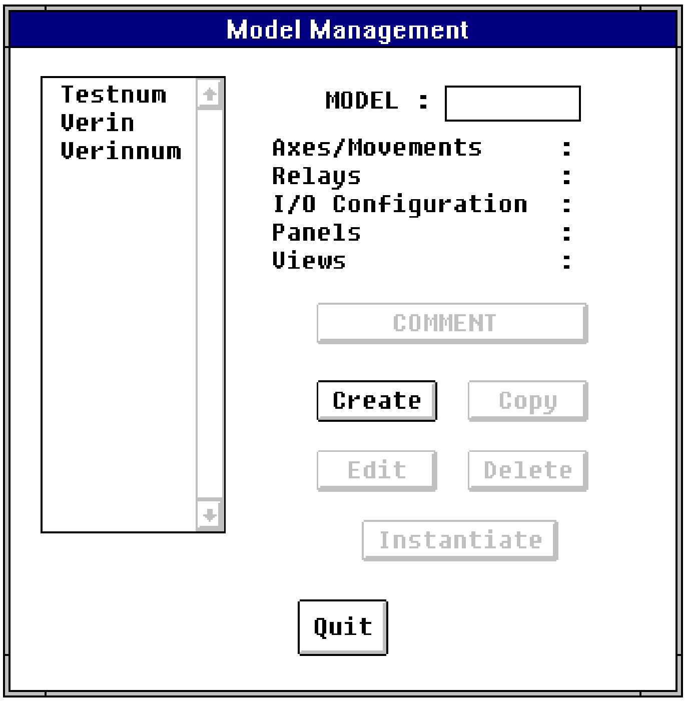
The same windows and sequence of commands are used.

The only difference is the option of adding a comment to the description variables to ensure the legibility of the model when it is used later.

In addition, a general comment can be associated with the model in the form of text which can be accessed via the **Comment** button in the model selection window and in the configuration interface background window when a model is being edited.

As with the other entities making up the description of a model, “undefined variables” can have an associated comment, added in the window providing access to these variables.

It is also in this window that the constraint to identify some of these variables with outputs can be introduced. This is done by checking the output box when an undefined variable is selected.



7.2.2 Setting parameters

The parameters of a model can be set, that is that the names of the model variables can include one (or several) parameters.

A parameter is designated by the character “$”, which may be followed by an integer between 0 and 9.

In addition to the names of the variables, parameters can be used in the labels of the axes and the description traps, in the mnemonics and comments of the inputs, in the titles and labels of operator panel elements as well as in the comments associated with the various positions of the “N-position switches” (see section 5.2 of this part).

We have seen that it can be established, in the definition of a model, that certain undefined variables are identified with outputs when the model is incorporated in a machine.

If this is the case, the name of the undefined variable in the model and its comment will become the mnemonics and comments of the output in the I/O configuration of the machine (provided that these have not already been defined, for example by reading an NIOF file).

This assignment takes place after the parameters have been replaced. For this reason, the comment associated with an undefined variable in a model can also be parameterized.

In the description interface, the parameters used in the model can be accessed via the **Parameters** button located in the first window for accessing each of the components in the description. A comment can be associated with the parameters at this level and the user can check in which description entities these parameters are used, in order to check the consistency of the parameters.

B

7.2.3 Example of a model

Using the example of the model of a cylinder controlled by a bistable directional control valve : the axis describing the travel of this cylinder is axis “Ax-1”, and its label is “Cylinder $0”.

The start and end limit sensors are Sq$00 and Sq$01 respectively. The bistable directional control valve is named D$0.

The extension and retraction movements are SV$0 and RV$0 respectively, and their equations are D$0 and S$0. The bistable latch term D$0 is YV$01, and its release term is YV$00.

The solenoid valve equations are as follows :

- YV$01 = Extension-$0 . Elec-Supp
- YV$00 = Retraction-$0 . Elec-Supp

The role of the parameter (here $0) is essentially to show in the model definition that there is a link between the names of the various variables, a link which will be propagated in the machine into which the model will be incorporated.

Thus, only the significant variables in the model are parameterized. In the example above, these are the sensors, the bistable and the solenoid valves.

The name of the axis itself is not parameterized.

The model of the cylinder contains undefined variables which are Extension-$0, Retraction-$0 and Elec-Supp.

This last variable is a condition of the electrical supply to the solenoid valves which control the directional control valve for the cylinder.

The two other variables can be identified with undefined outputs in the I/O configuration of the machine in which this model will be incorporated.

7.2.4 Definition of a model based on a machine

It is often after having described a particular machine for a simulation application that the advantages of saving part of it in the form of a generic model become apparent.

To avoid describing the same thing again, it is possible to extract part of a machine for reuse in another model. This operation involves a number of steps.

- **Transferring a machine to a model**

  This consists of copying to a model the various subassemblies which make up a machine : axes, relays, I/O configuration and operator panels. This function can be accessed from the Tools button in the configuration interface.

- **Reducing the model**

  Unless the machine has been described to validate a model, for example via local simulation, only some of the elements used in the description of this machine will be relevant to the model.

  Thus, a large part of the description will need to be deleted to extract the model which is to be retained.

  To simplify this operation, "reduction" functions are available in the model configuration interface.

- **Reducing the axes**

  The **Reduce** button in the main window for editing the axes of a model is used to select the axes which are to be retained in the model. The axes involved are selected in the left-hand side of the reduction window, and then appear in the right-hand side of this window. Reduction of the axes deletes the axes which have not been selected and those variables associated specifically with these axes.

  Thus, the variables used in the associations upstream/downstream of the selected axes are saved, as are the variables which are not associated with any axis in the description.

  By default, this reduction is also applied to the relays and inputs. This option can be deselected, in which case the reduction is only applied to the variables described in the axes/moving part.

B

• **Reducing the relays**

This reduction involves two aspects concerning the general power supply and the relays dedicated to the control of the actuators.

As regards the general power supply, reduction consists simply of deleting it totally.

For the other relays, reduction is used to retain only those relays which are linked to the axes in the model or the operator panels.

This method retains the relays used in controlling the axes or assigned by their evolutions, that is those relays used in the “upstream” and “downstream” associations. Relays in which variables activated by pushbuttons and switches are also retained, as are those involved in controlling indicator lamps and displays.

• **Reducing the inputs**

This consists of removing the inputs which are not connected to description elements. This means deleting the inputs in the logic equations in which only “general power supply” relays or “undefined” variables not used elsewhere in the description are used.

The reduction functions are used to “cut down to size” quickly a subassembly extracted from a machine.

So that this subassembly can be archived as a model which can be reused, the variables will probably have to be deleted again. Comments and, if necessary, parameters can be associated with the variables which have been retained using the **Rename** function.

7.3 Incorporating a model

A generic model can be incorporated at any time when editing a machine.

However, if the model includes the definition of inputs or if some undefined variables in the model need to be identified with outputs, the machine I/O configuration must have already been configured.

To incorporate a model in a machine description, click on the **Instantiate** button having first selected the required model in the model organization window.

Before the model can be included in the machine, a number of operations must be performed using various specific windows. These windows are activated automatically as soon as the model is loaded into the memory.

This process can be interrupted. It can be restarted using the **Adapt** button in the bottom left-hand corner of the main window for incorporating a model.

As soon as the model can be included in the machine, this button is replaced by the **Incorporate** button.

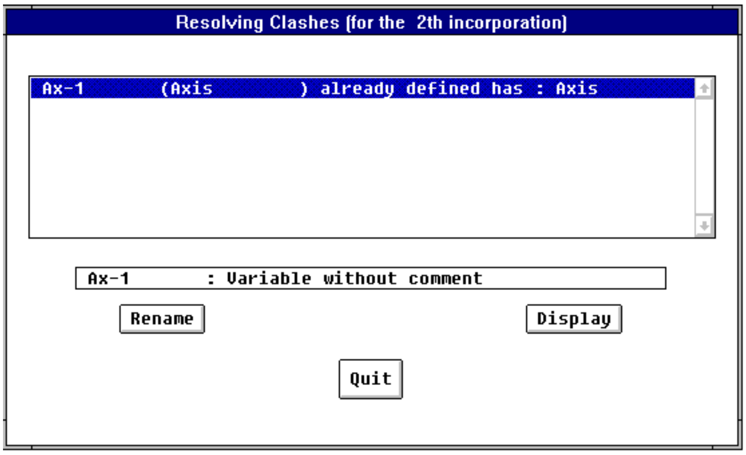
Before a model is incorporated, any clashes must be resolved, the parameter values must be set and the inputs must be identified.

7.3.1 Resolving clashes

“Clashes” correspond to incompatibilities between the model and the Machine into which it is to be incorporated, due to the existence of defined variables having the same name in both the model and the Machine.

If there are any clashes, the **Resolving Clashes** window is the first to be activated automatically during the incorporation of a model.

Take the example of the model of the cylinder used previously. The axis describing the cylinder travel is axis “Ax-1”. If an axis “Ax-1” has already been defined in the machine into which the model is to be incorporated, a clash is signaled. The clash is resolved by changing the name of the variable in the model, which, in the case of the cylinder, is the axis number.

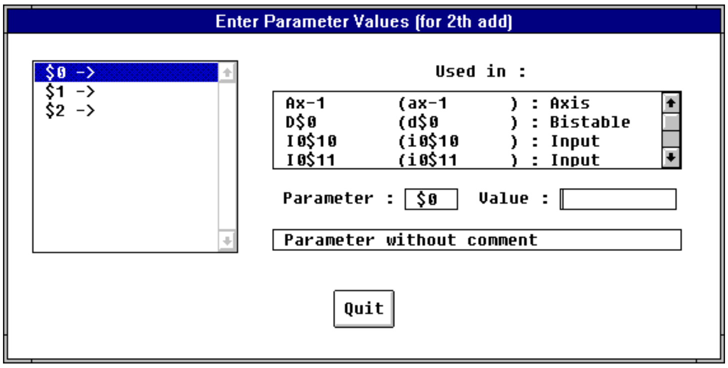


7.3.2 Setting parameter values

If parameters have been used in the definition of the model to be incorporated, they must be given a value so that the model can be "instantiated".

The parameters are displayed in the left part of the window. Selecting a parameter displays the elements of the model in which it is used.

The value to be assigned to a parameter is entered in an editor and confirmed by pressing "Enter". Only values which do not cause clashes are permitted.



B

7.3.3 Identifying the inputs

If the inputs are defined in the model and their addresses do not conform to the machine I/O configuration, or these addresses have already been assigned, the next stage necessary for adapting the model is “identifying the inputs”.

This consists of identifying the inputs in the model with unassigned addresses in the machine I/O configuration.

At this level, it is also possible to display the model inputs (the logic equations, mnemonics and comments).

7.3.4 Undefined variables

A model can include “undefined variables”, representing conditions which are external to the model.

Since a model is designed to be incorporated in a machine, these undefined variables may correspond to variables in the machine. For example, in the case of the cylinder model, the “Elec-Supp” variable used in the solenoid valve equations represents a condition of the electrical power supply. When this model is incorporated in a machine, this condition may be defined in the “Supply” part of the relay interface for this machine.

The “Identify” button in the window for accessing undefined variables in the model is used to select the variable in the machine with which one of these variables is to be identified.

The “Rename” button is used to change the name of an undefined variable in the model. If an identification exists with a machine variable, the type of the variable concerned is signaled.

Notes :

Parameterized variables, that is those with a name which includes a parameter, cannot be renamed.

In fact, the form of the name of a parameterized variable is considered to be a model constraint and thus cannot be modified.

For the same reason, these variables cannot be identified with machine variables : this identification can only be implicit, due to the evaluation of the parameters.

However, it is accepted that a parameterized undefined variable can be identified with a defined output in the machine I/O configuration. In this case, the name of the undefined variable in the model will become the mnemonic of the output and its comment will be the comment of this output.

- the undefined variables in the model which have been designated as having to be identified with outputs must then be identified with these outputs so that the model can be incorporated in the machine.
- the possibility of identifying an external variable in the model with a defined variable in the machine also applies to numeric variables.

B

7.3.5 Accessing the model description

The **Relays**, **Panels**, **Movements** and **Traps** buttons are used to access the various components of the model to be incorporated. This means that the various elements in the model can be displayed and the name of the model variables can be changed, provided that they have not been parameterized.

• Special case of operator panels

An operator panel is made up of elements acting on the state of the variables (pushbuttons and the various types of switch) and indicator lamps and displays reflecting the state of the variables.

The definition of an operator panel is said to be consistent if, in particular, the activated variables are used, for example in the relays or inputs, and if the variables controlling the indicator lamps and displays are outputs or relays.

The description of an operator panel in a model can be consistent. For this to be so, the operator panel must be accompanied by the description of a relay and/or input logic equations in which the variables in the elements of this operator panel are defined and used.

If, however, the model only comprises an operator panel, it is definitely “inconsistent”.

This consistency can be checked in the **Panels** window during the incorporation of a model using the **Variable Consistency** button. If it is not consistent, the variables concerned are shown. This concerns both variables used for the indicator lamps and displays which are neither outputs nor relays (in the model or the machine) and variables controlled by the pushbuttons and switches and which are not used. The first type of variables can be identified with relays or outputs defined in the machine. The second type can be identified with undefined variables. This explicit identification is only possible for variables which do not have parameters.

7.4 Tracing instantiations

A trace is maintained of the instantiations of models in a machine. This trace is accessed via the **Tools** button in the configuration interface.

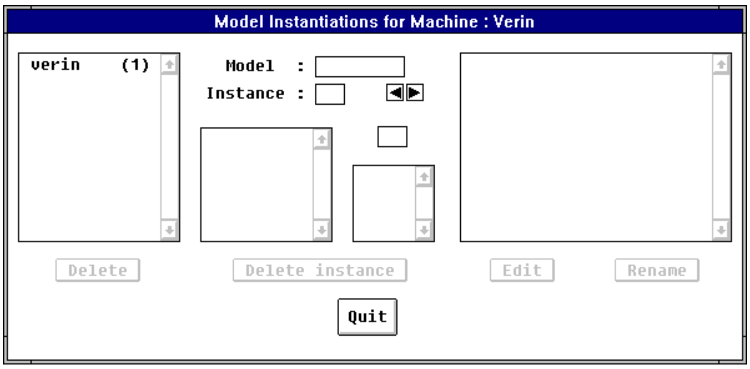
The names of the various models incorporated are listed along with the number of corresponding instances. If one of the model names is selected from the list on the left, the value given to the parameters for the first instance of this model is shown. The machine variables corresponding to this instance are displayed on the right.

If several instances exist for the selected model, they can be displayed using the arrows to the right of the number of the instance displayed in the central part of the window. The values given to the parameters and the corresponding variables are shown.

A particular instance of a model can be accessed by selecting the values which have been given to the model parameters during incorporation in the machine. To do this, click, in the central part of the window, on the parameter for which you wish to select the value. All the values taken by this parameter are then displayed, including values which have been selected for the previous parameters. If one of these values is selected, the first instance for which the corresponding parameter has this value is displayed.

When an instance is selected, the corresponding variables are displayed on the right-hand side of the window. These variables can be edited and their name can also be changed. If one of these variables is edited, the **Previous** and **Next** buttons in the corresponding edit window are used to move through all the machine variables originating from the model variable. This makes it easier to correct the instances of a model at a later date.

The **Delete** button in the list of models present in the machine is used to remove all instances of this model from the machine description. The **Delete Instance** button is used to remove the selected instance only from the machine.



B

1.1 Simulation by events

In contrast to the continuous processes of the chemical industry for example, logic and sequential aspects dominate the operation of systems controlled by PLCs.

In this respect, the evolution of a control system can be described as a series of “discrete” events.

Specifically, events are changes in the state of the I/O which ensure dialogue between the control system and the application of the automated system.

In order to take advantage of this characteristic of industrial control systems, the SIMACTEL system has been designed according to the principle of “simulation by events”. In other words it does not reproduce the evolution of the machine in real time, but reproduces time-referenced progression of successive logic situations.

According to outputs from the control system, the machine model can determine which inputs will evolve, and when.

On the one hand, these inputs can then be sent to the PLC in the same chronological order as in reality and on the other, the machine time (or “simulation time”) can be advanced.

Since the simulation evolution is independent of the realtime machine movements, simulation will, in the majority of cases, be “speeded up” in relation to reality. However, for very rapid machine evolutions, it may be slower.

1.2 Exchange of I/O between control system and application

The principle of simulation by events applied in SIMACTEL consists of predicting input change according to the output state, using a model of the application.

In order for this extrapolation to be correct, it must be done on the basis of a stable output state. Otherwise, a minimal time difference, similar to the PLC scan time, between the appearance of outputs could cause a far greater difference at the level of events generated, if these outputs were not processed simultaneously.

In the same way, in the machine model, it would not be correct to propagate the transient activation of an output during a scan time : due to the inertia of the actuators, this output would have no effect on the machine in reality.

1.2.1 Control system - application communication in internal simulation

SIMACTEL offers the possibility of checking the model of the machine to be simulated by controlling the machine with the aid of a mechanical Grafcet chart : this is “Internal Simulation”.

This simulation is performed in line with the Grafcet standard. In particular, the Grafcet chart is executed in a manner which seeks stability on the change of state of an input. Only the final, and therefore stable, state of the outputs is perceived by the model of the simulated machine.

1.2.2 I/O exchange with a PLC

Communication between the simulator, located on an FTX or PC compatible, and the PLC is performed via an OFB, which simulates the presence of I/O cards in the PLC rack.

We have seen that, in order for simulation to work correctly, the prediction of input changes must depend on a stable output state.

Communication then takes place in the following manner. Each time the simulator sends inputs to the OFB, the latter supplies them to the PLC and receives output changes, resulting from the PLC reaction, for a time period equal to several cycles.

If there are no time delays in the PLC or if their effect has been corrected according to principles which will be defined later, this time enables the PLC program to evolve and to present a stable output state. The outputs received are sent to the simulator which can then continue its processing. If the outputs continue to change, they are stored by the OFB which will then supply them to the simulator when this sends the changes in state of the next inputs.

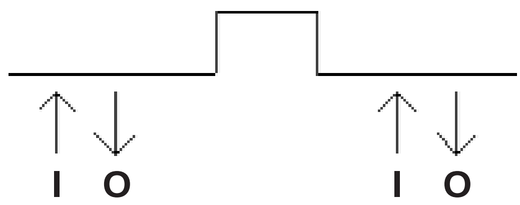
1.2.3 Time delay problems

Communication between the simulator and the PLC is based on the hypothesis that when the PLC perceives a change in the input state, the outputs that it will activate reach a stable state after several program execution scans.

However the use of time delays in the PLC program may cause this hypothesis to fail.

Time delays can be used in various ways :

- Some are used to carry out “open loop” actions, which do not generate any feedback from the application. This is the case for auxiliary functions such as the flashing of LEDs or spraying. These time delays cause no communication problems, although the period of flashing must not be too short to ensure that the active/inactive state of the output involved can be correctly perceived.
- Time delays can also be used to generate output pulses of a sufficient duration to overcome the inertia of a preactuator such as a directional control valve. The transmission of these pulsed commands can also be delayed in relation to the reception of inputs by the PLC. The problem is therefore to prevent the outputs from being filtered by the communication process between the simulator and the PLC.



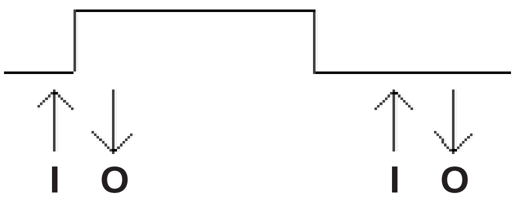
C

In effect, if the exchange of I/O takes place in a way that the rising and falling edges of a pulsed output are transmitted by the OFB to the same group of outputs, the temporary change to the active state of the output concerned will not be taken into account and will therefore not have the required effect on the simulated application.

To solve this problem, the pulse must not be produced between two successive I/O exchanges, which can sometimes happen if the change to 1 of the output is delayed in relation to the reception of inputs.

The duration of the pulse must therefore be greater than the interval of time separating two I/O exchanges.

Since this time depends on the processing carried out by the simulator and can therefore vary greatly, it is wiser to eliminate any delay between the reception of inputs and transmission of outputs by modifying the time delays in the PLC program.



- Finally, time delays can also be used to monitor application evolutions. They are used as “watchdogs” for movements or part of a cycle enabling, for example, the detection of a sensor sticking at 0 or an actuator blocking.

Although generally simulation is “speeded up” in relation to the actual machine, for certain rapid movements, the simulator can sometimes be slower than reality. It is better therefore to initially increase the value of these “watchdog” time delays. The sequential part of the animation program can thus be validated independently of the time-based aspects of the monitoring. The correct operation of monitoring time delays can then be checked by adjusting their value according to the actual response time of the simulator.

1.2.4 Checking the output changes during simulation

We have seen that the exchange of I/O between the simulator and the PLC is based on the hypothesis that when the simulator sends a change in state of the inputs to the PLC, this latter presents a stable state of its outputs at the end of several scans.

In order for simulation to take place correctly, the OFB “watchdog” must be correctly adjusted and the time delays of the PLC program must be modified as specified in the previous paragraph of this section.

After each input writing, the OFB groups the changes in state of the outputs appearing during a certain number of scans and sends them to the simulator. If the OFB watchdog is too short or if the time delays have not been appropriately modified, the outputs are susceptible to change after this time period : they are transmitted to the simulator at the next exchange of I/O.

However the simulator may have evolved the model of the machine incorrectly given the actual PLC response has not been received in time.

For example, the simulator sends an input to the PLC informing it of the arrival of an axis at an intermediate position, the PLC responds by resetting an output which should stop the movement. If the OFB “watchdog” is too short, the change to 0 of this output may not be received when the input is sent but on the next exchange : the simulator should have therefore evolved the axis outside the expected stop position, causing inconsistency in the simulation.

These outputs are identified in the server historic by the letter «a» between brackets.

Another problem which might arise during the course of simulation is the emission of pulses of too short duration by the PLC. This may arise if the time delay controlling the pulse is not of a sufficient duration in relation to the time during which the OFB groups together the output changes.

In this case, the rising and falling edges of the output are received in the same packet of outputs and the temporary change to the active state is not taken into account by the simulator.

C

2.1 Background window for the simulation environment

The main window for the simulation environment comprises three zones. The title bar contains the various commands which can be used in simulation. The zone situated below this bar is divided into three parts, dedicated to the operating modes, displaying the time and choice of period, and displaying the simulation state. The bottom of the window is reserved for displaying the background view for the simulated machine.

The simulator operating modes are Step by Step, Periodic and Continuous. They are selected by checking the corresponding boxes.

- In **Step by Step** mode, simulation is interrupted each time the output state changes.
- Periodic mode enables simulation to be stopped according to a machine time step. The value of this step is entered via a dialog box which is opened by clicking on the **Period** button. This window appears automatically if Periodic mode is selected and the Period is not defined.
- **Continuous** mode is selected when the Step by Step or Periodic modes are not selected : simulation continues without interruption as long as there are active movements.

If you are in periodic mode and you then select continuous mode, the period which was entered is lost. On the other hand, if the periodic mode is deselected, the period is retained.

The **Simulation Time** is displayed in the central part of the zone situated under the title bar. By default, this advances according to the events generated by the evolution of the application model. For relatively slow processes, this “advancement of time” can be much faster than in reality. The "real" time, according to which simulation is executed, depends on the number of calculations to be performed and not on the movement time defined during configuration of the model.

It is possible to slow simulation by defining a period with a low value using the Period button. Another way is to select **Delayed Mode** by checking on the box below the time. This enables, as far as possible, the evolution of the simulation time and the PC clock to be synchronized. In other words, the simulation time does not advance any faster than "real" time. This is only possible if the calculations to be carried out - and the exchanges with the PLC - are not too numerous. A "+" sign appears to the left of the "Delayed Mode" command if this is the case.

The right side of the window displays the simulation state. When no movement is produced, the text **Blocking** appears on a black background. During simulation, the Interrupt button enables evolution of the model, and therefore the time to be interrupted. When simulation is stopped - for example on the appearance of an output when Step by Step mode is selected - the **Restart** button enables simulation to be restarted.

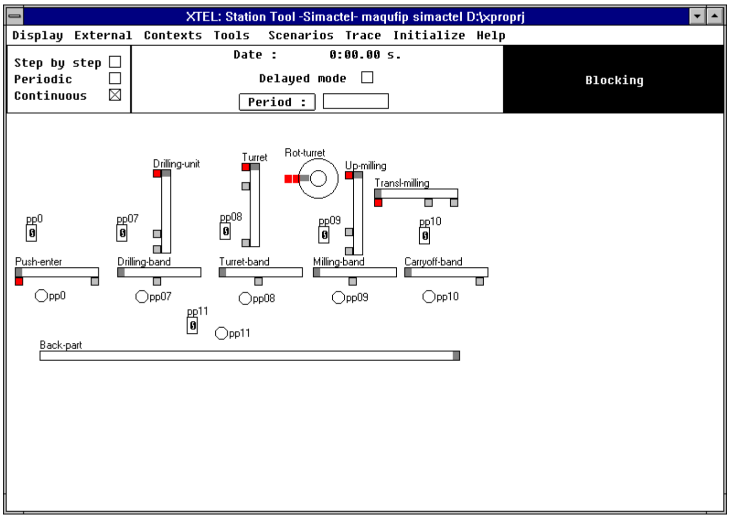
A specific state of SIMACTEL is **Interrupt on Model Problem**. This occurs when an inconsistency in the modelling process prevents simulation continuing. Possible causes are specified in the Error Messages section in the Appendix (see section 2 of part F).

The table below refers, for each of the simulation environment commands, to the relevant section of the documentation.

Simulation environment commands :

Operator panels	see section 2.2 of this part.
External variables	see section 2.2 of this part.
Contexts	see section 2.2 of this part.
Views	see section 2.3 of this part.
Tools	
Evolutions	Opens a window displaying I/O exchanges.
Event prediction	Opens a window displaying SIMACTEL event prediction.
Examination	see section 2.5 of this part.
Pages	see section 2.3 of this part.
Traps	see section 2.6 of this part.
Forcing	see section 2.4 of this part.
Description	see section 2.10 of this part.
Faults	see section 2.7 of this part.
Scenarios	see section 2.8 of this part.
Trace	see section 3 of part D.
Help	Opens the simulation online help.
Initialization	Renitializes simulation offering a context (see section 2.2 of this part).
End	Quits simulation on request if the machine state is to be saved. See section 2.10 of this part.

C



C

2.2 Actions on operator panels and external variables

2.2.1 External variables

"External variables" is the name given to terms which affect the model logic equations and whose origin is not specified (undefined variables of the description part). It generally refers to machine variables which represent energy supply conditions, sensors not activated by axes, such as safety sensors, or the presence of parts at the entry point of a machine.

By creating implicit inputs, internal Grafcet chart variables which are not managed also enter into this category : these can then be forced.

Finally we have seen that outputs which are not controlled by the Grafcet chart in internal simulation but appear in the model, change to "external variables".

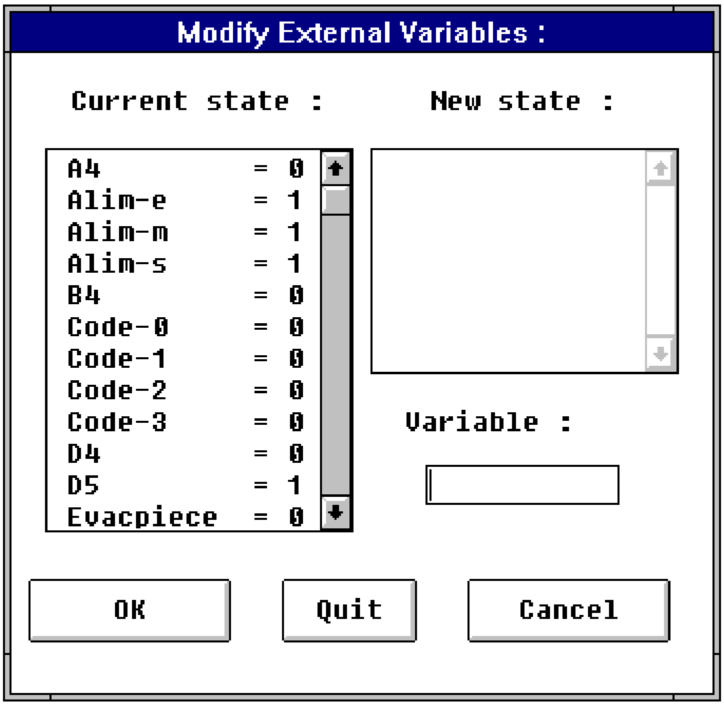
The **External** button at the top left of the simulation screen produces a window which enables the state of the variables to be modified.

The external variables are presented in the left part of this window together with their logic state.

If one of the variables is selected from the list on the left of the screen, it will appear on the right side of the screen.

The variable can be removed from the right side of the screen by clicking on it in the same way. Another way of selecting a variable in order to modify its state is to type it in the editor field in the window ; if the name entered is not that of an external variable, the first external variable with an alphabetically higher initial letter than the typed variable appears at the top of the list on the left side of the window. The OK button enables the selected changes of state to be sent to the simulator.

C



2.2.2 Operator panels

In the simulator operating environment, operator panels appear as windows which can be positioned as the user wishes. Operations on elements comprising these operator panels is carried out using a mouse according to the following procedures.

• Manipulating operator panels

If the operator panels have been described, the **Panels** button at the top left hand side of the simulation window produces a menu for choosing the operator panel to be displayed.

The **All** item activates, in a single command, all the operator panels configured. The selected operator panels appear in the position in which they were located in the last simulation. By default, they appear in the lower right hand part of the screen.

The **End** button at the top right of an operator panel removes it from the screen.

• Actions on operator panel elements

Operator panels are composed of elements such as indicator lamps and hexadecimal displays, for displaying information, and pushbuttons and switches for running the automated system. Operator action is therefore involved with the latter.

• Pushbuttons

The usual use of a pushbutton is to generate a pulse to the control system of the automated system. This can be done in SIMACTEL by clicking on the pushbutton with the mouse. The operator panel button remains pushed in as long as the mouse button is held down. The pushbutton returns to its original state when the mouse is released.

The latching pushbutton is a particular type of pushbutton which remains depressed when it is pushed in : this is the case for emergency stop buttons which sometimes require a key to release them. This type of button is found in SIMACTEL : one click of the mouse latches it, and a special command releases it.

The button can be released in two ways. The first is by clicking the **Release** command at the top of the operator panel : this changes to reverse video and by clicking the button concerned it can be released. The second is by holding down the “Ctrl” key on the keyboard and clicking the button.

It is possible to hold down a pushbutton. This can be useful for simultaneously activating several pushbuttons or for selecting a simulator function using the mouse while keeping a button depressed.

As for releasing a pushbutton, there are two ways of holding the pushbutton down :

- the first way is by clicking on the **pulse** command at the top of the operator panel : this is then replaced by the text **"hold".** If the pushbutton is clicked, it changes to the hold state which is signalled by a finger icon on the button,
- the second way is by clicking on the button while pressing the <Shift> key on the keyboard.

A pushbutton which is held down can be released, as for latching buttons, by using the operator panel **release** command or by the combined action of holding the <Ctrl> key and clicking the mouse.

C

• Switches

In SIMACTEL switches are available with 2, 3 or n positions. Switches are positioned using the left and right mouse buttons.

2 or 3 position switches may be equipped with a spring return for certain positions, for example, the left position of a 2 position switch. In this case the position is indicated by an arrow and the switch acts as a pushbutton in this position : one click of the mouse generates a pulse after which the button returns to its original position, in other words the left position for a 2 position switch or the central position for a 3 position switch.

In the same way as for pushbuttons, a switch can be held in a spring return position : the arrow indicating the spring return no longer appears.

• Thumbwheels

The selected position is incremented by clicking on the “ + ” box, and decremented by clicking on the “ - ” box.

• Analyzing the states of indicator lamps and displays

Operator panel components can be “interrogated” on their state. It is possible to find out why an indicator is on (or lit) or according to which variables the numeric value is displayed in a hexadecimal decoder. For this, the mouse cursor must be positioned on the element in question and the left and right buttons must be clicked simultaneously. For an indicator lamp, there will be an “examination” of the variable controlling it. In the case of a hexadecimal display, the state of the 4 data bits and the optional validation variable will be displayed.

Note :

To “interrogate” an indicator pushbutton, the 2 mouse buttons must be clicked and the <Ctrl> key on the keyboard must be pressed in order to avoid activation of the button function of this element.

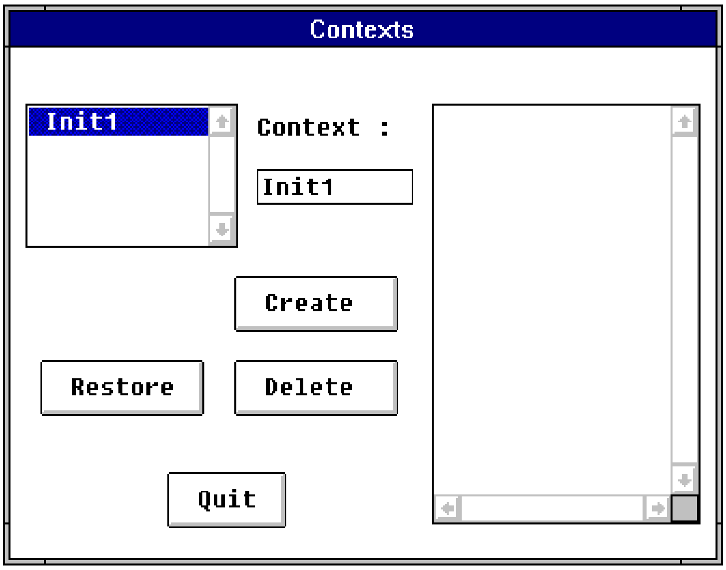
C

2.2.3 Concept of “context”

We have just seen how to manipulate operator panels and perform operations on external variables.

The **Context** function, which can be accessed via the button at the top of the simulation window, enables a current state of operator panels and external variables to be saved : the recorded context can therefore be restored each time simulation is initialized.

During simulation, this function can be used in order to check the state of the operator panel and external variables : by selecting a context from the window, enabling them to be created, the difference between the current situation and the context is shown in the right hand part of this window. Variables with a state which does not conform to the context are signalled and by clicking on one of these variables, the element, and the operator panel to which it is attached, are displayed. The **Restore** button enables the state of these variables to be made consistent with the selected context.



2.2.4 Concept of a “scenario”

The concept of context is completed by the possibility of recording and restoring a series of actions on external variables and operator panel elements in order to automate command sequences (start sequence, restart cycles, etc).

This is a specific case of the use of scenarios. The concept of a scenario is explained in section 2.8 of this part.

C

2.3 Display tools

We have just seen in the previous paragraph how the simulator can be controlled via operator panels and actions on external variables. It is also important to be able to observe evolutions of the simulated model under good conditions other than via the simple exchange of I/O or through the prediction of expected events.

Functions have been designed in order to do this : these are the “dynamic pages” and “views” functions.

The examination function also enables simulation to be observed. Due to its special nature, this function will be outlined in a separate section (see section 2.5 of this part).

2.3.1 Dynamic pages

This tool enables the state of the previously selected variables to be permanently displayed.

This function can be accessed by selecting the **Pages** item from the **Tools** menu located in the lower part of the simulation screen.

The first window is then displayed, enabling a new page to be created, and previously configured pages to be edited, displayed and deleted.

The **Create** button opens a window where the name of the page to be configured is entered. After confirmation, the page can be edited.

• Editing a page

The edit window comprises a zone on the right where the variables which constitute the edited page are displayed.

This zone is empty if a new page is to be created. The center of the window enables the selection of the type of variable to be shown in the page.

The following choices are available :

- I/O configuration,
- relays and general power supply to the relay interface,
- description variables of the “moving” part,
- operation traps.

Selecting the type of variable makes the list appear in the left part of the window.

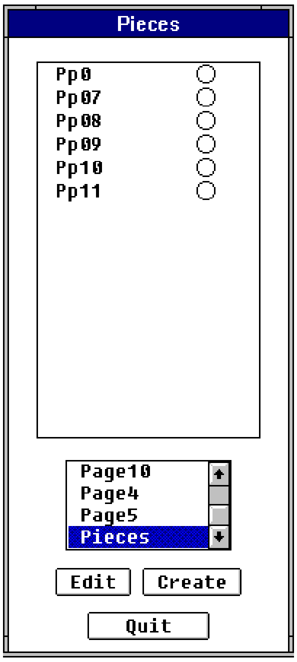
The page can be easily configured by selecting variables from this list using the mouse.

A variable can be removed from the page in the same way by clicking on it in the right part of the window. A variable can also be added to a page using the editor located at the bottom middle part of the window.

If the entered variable is correct, it is transferred to the right part of the window, after it has been confirmed by Enter.

As well as the choice of variables previously described, sensors, bistables, movements or external variables can also be incorporated in the page.

If the entered variable does not correspond to one of the variables of the type selected, the list of variables presented is scrolled alphabetically.



• Display of dynamic pages

Clicking the **Display** button in the main window of the **Pages** function displays the variables which make up the selected page in the right part of the bottom simulator window. Variables are arranged in a column and their state is indicated by a “Radio button” icon.

The display window has —> and <— buttons in order to access the following or previous page.

The **Edit** button allows the displayed page to be edited and consequently the list of variables which make up the page to be modified.

It is also possible to create a new page using the appropriate button.

Note :

The reason for the logic state of a variable displayed on the page can be analyzed using the **Examination** function (which will be covered in more detail later). This function is made available when the simulator is stopped or blocked by clicking on the variable concerned.

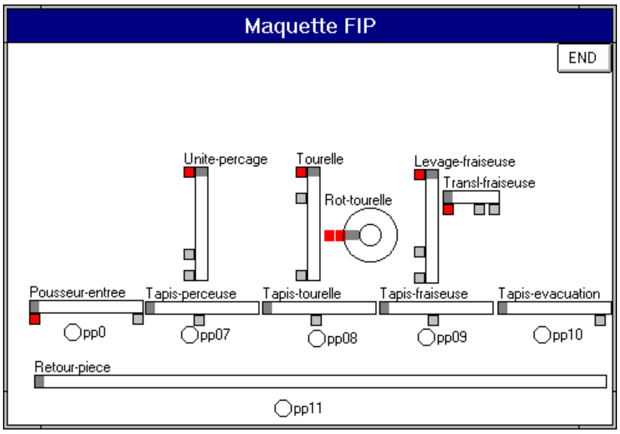
2.3.2 Views

In the simulation environment, previously described views appear in the form of windows (views), except for the view titled **Background** which appears in the main simulation window (see section 7 of part B). The Grafcet chart view windows are accessed in the same way as other views. Their use is described in section 3 of this part.

• Using views

If the views have been described, the **Views** button accesses a menu for selecting the views to be displayed.

The **All** item displays all views with a single command. For each view displayed, the **End** button enables the screen to be deleted.
When the user quits the simulation, the positions on the views on the screen are maintained.



Animating views

Animating axes
An axis is represented by a contour (vertical or horizontal rectangle or circle) in which evolutions of the current position can be seen by means of a cam or a bargraph.

The states of axis sensors can also be seen and are “illuminated” when they change to 1.

Animating variables
In the same way as for pages, the variables of a view are “illuminated” when they change to 1 and “off” when they change to 0.

Animating decoders

A calculation is made according to the value of the decoder bits and a figure indicates the current reconstructed value.

Examining view variables

When the simulator is stopped or in the event of blocking, it is possible to examine the view variables, by double-clicking on the graphic element concerned.

For example, for the axes, the standard window for examining an axis is available (see section 2.5 of this part).

2.4 Forcing description variables

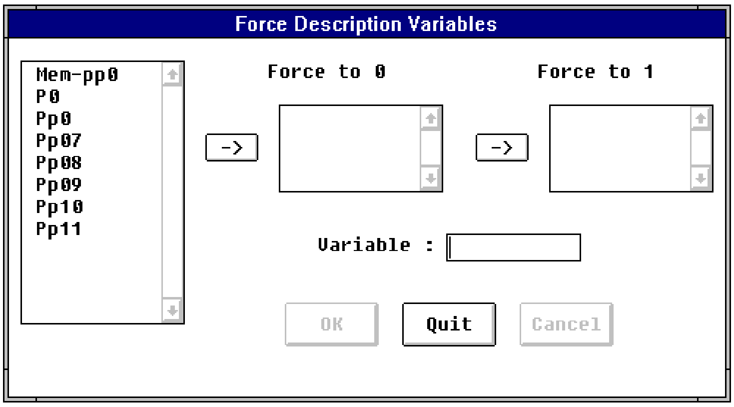
These variables are equivalent to relays in the internal operation of SIMACTEL. However, they are described in the “Axes” part of the configuration interface.

They cannot be subjected to faults in the simulation environment.

Forcing enables direct action on the logic state of these variables which is normally defined by a Boolean expression.

This enables for example, in the case where these variables represent the presence of parts, the insertion of a part into any location on the simulated machine without this having been explicitly planned in the equation for the variables concerned when modelling. Another advantage is that a modelling error (incorrect variable equation) can be temporarily corrected without having to restart the simulation after correction. Also, any “ad hoc”corrections which are not relevant can be avoided.

It must be realized however, that “forcing” may cause inconsistencies in the modelling of the application if it is not used in a controlled manner. In other words SIMACTEL must not be turned into a “control station” by ignoring the modelling effort which is this product's main feature.



C

C

2.5 Examination or analysis of situations

“Examination” is a function which can be found at various levels in SIMACTEL. It can be accessed when simulation is “stopped”, in other words on blocking, or at one of the various possible stops.

This tool has already been mentioned with regard to operator panel indicator lamps and variables displayed on the dynamic pages.

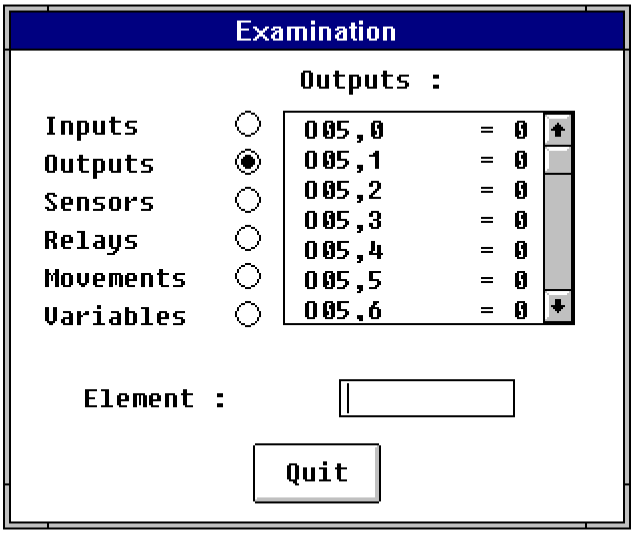
The main window of this function is activated using the Examination item of the Tools menu.

This window comprises a number of buttons and an editor. Clicking the mouse on one of these buttons displays the logic states of the type of variables selected.

An examination of one of the variables present can therefore be made by clicking on it with the mouse.

Another way of examining a variable is to enter it in the editor of the examination window.

“Enter” starts an examination of the variable if that variable is known to the simulator. If not, the list of variables present is scrolled alphabetically according to the position of the variable entered.



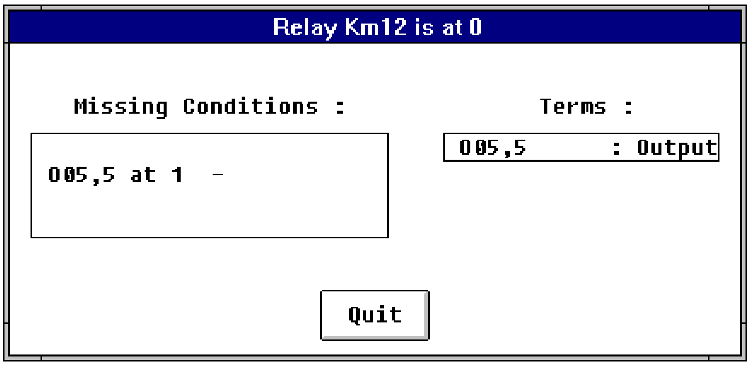
2.5.1 Variables whose state is determined by a Boolean expression

This is the case for a number of entities present in a simulation model such as the input configuration, the relays in the relay interface or the variables of the moving part.

The examination of these variables consists of presenting, if the variable is at logic state “0”, the terms which are missing for checking the variable equation.

If, on the other hand, the variable is at “1”, the explanation of this state, based on the analysis of the equation, is given. The **Set to 0** button enables conditions to be produced to reset the variable to “0”.
The examination window for a variable comprises a main zone where information is displayed.

The list of variables missing or explaining the “true” state is presented in a scroll-down list in the right part of the window. By selecting one of these variables, the examination can be carried out in more depth.



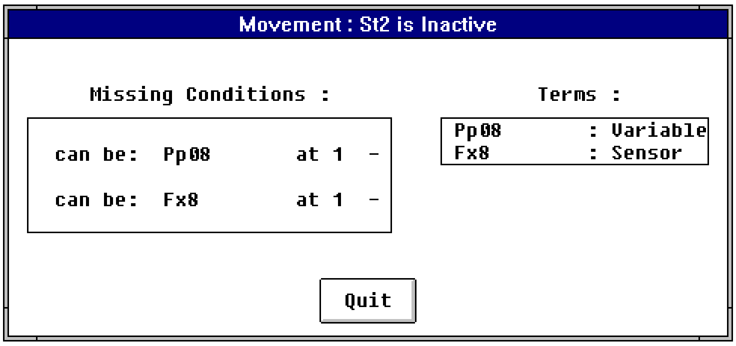
2.5.2 Movements

The examination of a movement associated with an axis is slightly different. If the state (activity) of the movement is determined by a Boolean expression, as in the case of linear axis, this movement is limited by the end or start limit of the axis according to the direction of movement.

When a movement is examined, the following is possible :

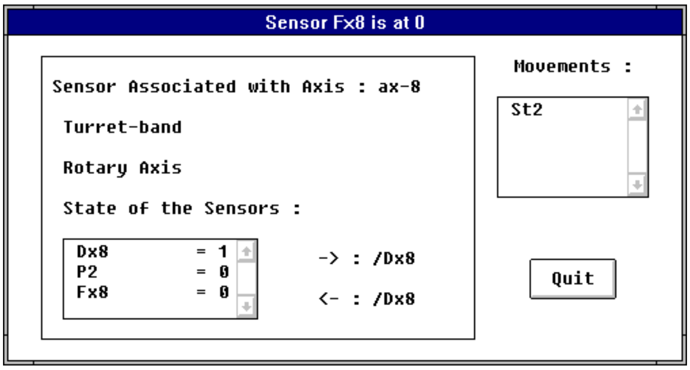
- If the axis to which the movement is associated is linear and the movement is at the end limit, this is indicated and the movement equation is not analyzed.
- If, the movement is inactive but another movement is active for the associated axis, this movement is indicated and an examination of this is started when the window is quit.
- If the movement examined is inactive without being at the end limit, the analysis of the equation provides the missing conditions.

The examination may indicate an “impossible movement” : this occurs when the sensors of the associated axis are involved in the missing conditions.



2.5.3 Sensors

The simulator can be interrogated on the state of a sensor. A specific window is allocated for this purpose. Information on the axis associated with the sensor is displayed here. In the right part of the window movement(s) are displayed which enable the sensor to be quit or attained depending on whether it is active or inactive.



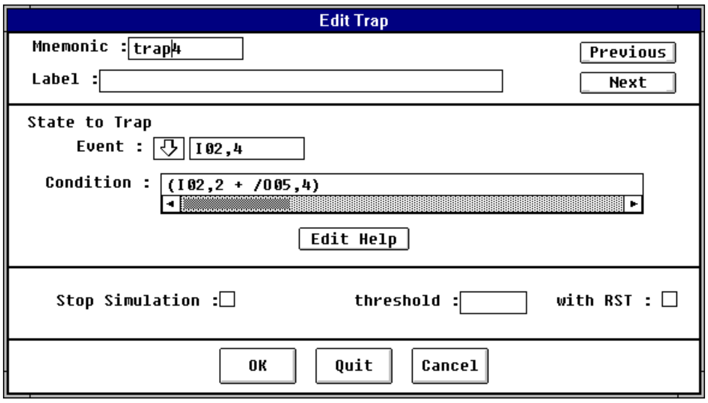
2.6 Operation traps

Using the traps, the simulator can be stopped if a particular situation should arise. Traps also enable the occurrence of events to be counted, the time lapse between two successive occurrences or the duration of a particular situation to be measured.

2.6.1 Characteristics of a trap

A trap is defined by an event and/or a condition. The variables to which these terms apply are I/O, relays and, if required, description variables. If the condition is not defined, it is considered as being always true. The trap is checked if, on the occurrence of the event, the condition is validated. If the event does not exist, the condition is systematically tested. Text of 40 characters can be added to the trap mnemonic.

C



2.6.2 Evaluating traps

Inputs, outputs and, if required, relays are involved in the definition of a trap. These variables obviously all affect each other, notably via the control system which changes outputs according to the changes in state of the inputs it perceives.

Depending on whether the test is carried out before or after the exchange between the simulator and the PLC, the same logic situations combining I/O will not observed.

The choice made in SIMACTEL is to evaluate the traps after reception of outputs from the PLC, and propagation in the relays in the relay interface.

This enables the reactions of the PLC program to be adequately tested; for example, the disappearance of an output controlling a movement when an input is activated. If we did not wait for a reaction from the PLC before testing the trap, this could be unexpectedly triggered.

2.6.3 Action of traps

A trap is activated by a message composed of a mnemonic and the trap text.

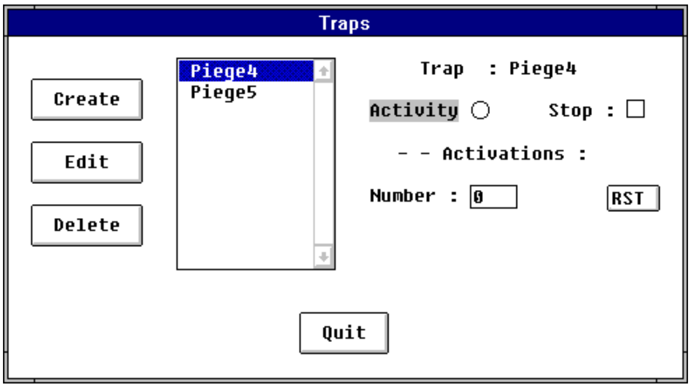
The elapsed time (cumulative total of the machine evolution times) since the last trap occurrence is indicated. If the trap does not contain a releasing element, its deactivation is signalled, as well as the duration of this activation (time during which the condition has been verified).
The threshold associated with a trap enables the start of an action to be delayed until after a certain number of occurrences.

If the **Reset** (**RST**) option has been selected, the action will start every n occurrence, n being the threshold value.
The simulation stop caused by the trap can be deactivated: its activation will be “silent” but a cumulative total of its occurrences will nevertheless be kept : the trap can therefore act as a counter.

Note :

A trap can be incorporated in a page of variables; its state is therefore displayed dynamically.

When the simulator is stopped, an “examination” can be started by clicking on the name of the trap. If the trap possesses a condition, the examination corresponds to the analysis of this condition.



2.7 Faults

Verification of normal operation and the operating modes is not generally sufficient for validating a command program. It is also useful to test the behavior of the automated system when introducing faults in the application.

The number of stops caused by faults hinders productivity. It is therefore important to check that these are detected by the control system and/or automated system monitoring and that the restart modes are correct.
In SIMACTEL, it is also possible to create faults in the model of the machine.

C

There are three types of fault :

- sticking at 0 or at 1 of the sensors, relays and bistables : when the element involved changes to the corresponding state, it remains this way until the fault is corrected (or forced to the opposite state),
- forcing involves the same elements : the change to the selected state is immediate,
- bounces on the sensors, which may occur on activation or deactivation.

These functions can be accessed by the **Faults** item in the **Tools** menu.

2.7.1 Sticking and forcing

In the left part of the window which enables this type of fault to be set, there appears a list of the entities concerned.

After selecting one of these variables, the chosen fault is set using one of the buttons.

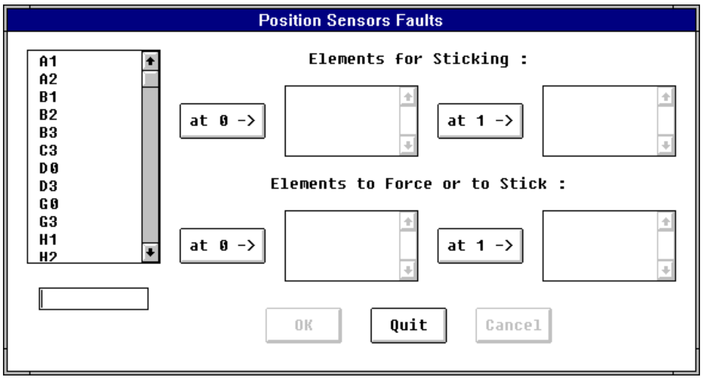
The variable on which the fault is to be placed can also be selected using the editor which is below the list in the left part of the window.

The fault is cleared from the variable by clicking on the fault zone in which it appears.

Note :

The Cancel button does not delete faults which have been set, but indicates their current state.

Sticking affecting sensors is taken into account in the input configuration equations and in the relays in the relay interface where these elements are involved. However, it is not perceived in the axis movements nor the “description variables” of the mechanical machine description. Setting a fault on a sensor therefore has the same effect as in reality, without the undesired effect of disturbing the modelling of the application.



C

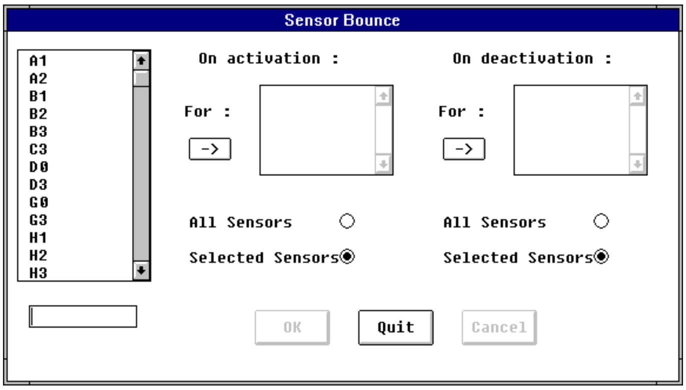
2.7.2 Sensor bounce

When a sensor affected by a “bounce” fault changes state, this fault sends to the PLC 3 successive changes of state of the inputs where the sensor occurs. The state of this sensor changes twice after changing to the correct state.

At least one PLC scan elapses before these input changes appear on the I/O bus : the PLC therefore has the time to perceive them and advance its program. The outputs are not sent to the simulator until after the three successive elements have been received.

There are two ways of setting sensor bounce.

Either the fault is set on certain sensors only, or it is set on all the sensors, except for a few specified ones. Sensors are selected in the same way as for sticking and forcing, via a list or by direct entry using an editor.



2.7.3 Notes

• It is possible to simultaneously delete all faults which have been set. To do this, use the Delete All item in the Faults sub-menu.

• Upon initialization, the simulator requests if all faults which have been set are to be deleted.

2.8 Scenarios

This function comprises several aspects.

The first is the possibility of automatically restoring sequences of actions on the operator panels and external variables. This facilitates the automated system starting procedures - power-up, start-up - or changes of operating mode during operation.

These sequences are created by saving and their operation can, if necessary, be controlled by “wait periods”.

Another use is for defining breakdown scenarios, which are useful, for example, from the point of view of operator training.

These scenarios, set up using an edit tool, enable faults to be automatically set during simulation according to certain events. To meet this requirement, it is possible to test the number of trap activations during a planned wait period : a sensor can therefore be broken after a certain number of parts have passed, for example.

Finally, the concept of “reaction” is used to check the conformity of an evolution of an automated system subassembly with a reference model.

This reference is a scenario composed of reactions. It is established by recording the changes in state of variables associated with the scenario. During operation, these changes in state are considered as reactions which must be reproduced. The scenario must therefore be executed under the same conditions : the wait periods enable it to be synchronized with the evolution of the automated system.

2.8.1 Sequences of actions on operator panels and external variables

This use is specific to scenarios and enables notably the automation of the automated system start-up procedure.

• **Record**

Firstly a scenario must be created by giving it a name and, if required, a comment.

When this has been done all that remains is to select the scenario and click on the **Record** button of the scenario management window. After confirmation, recording is activated. The **Scenario** button of the Simulation window is then grayed out and its text changes to **Record.**

C

Simultaneously a subsequent window appears. This window can be closed by clicking on **Quit**. The **Record** button in the background window enables it to be recalled. The **Save** button ends the recording and saves it to disk.

As the actions are performed on the operator panels and external variables, the changes in state for the corresponding variables are displayed in this window. The syntaxes are as follows :

- <var> <- 1
- <var> <- 0

for the change to 1 or 0 of a variable.

In the case of pushbuttons, the syntax is as follows, depending on whether the contact is normally open or normally closed :

- <var> <> 1 **or** - <var> <> 0

In the case of switches with n positions and thumbwheels, several changes of state of variables may occur simultaneously : these are “synchronous actions” whose syntax is as follows :

- <var1> <- 0

- <var2> <- 1

If actions are performed at different simulation times, an indication of the time elapsed since the previous action - or since the beginning of recording if it is the first action - is added to the recorded scenario line. For example, the syntax in the case of a change to 1 of a variable is :

- <var> <- 1 d: <duration>

In the case of synchronous actions, the indication of duration follows the first change of state displayed.

Note :

The maximum recording capacity for a scenario is 150 lines. Above this, recording is interrupted and the user is requested to save.

• **Restoring the sequence of actions**

In order to do this, click on the **Execute** button of the Scenarios management window. After confirmation, execution is launched. The **Scenarios** button is grayed out and its text changes to **Execution**.

During execution of a sequence, the operator panels are locked and modification of the state of the external variables and setting of faults is inhibited.

The recorded actions are restored, respecting possible duration differences, unless blocking occurs : in this case the action is executed without the associated wait period being taken into account. It must be remembered, however, that the time reference used is “machine time” and not “realtime”, which renders inoperative those sequences where PLC time delays are used.

If the **Execution** button is clicked a window appears in which the contents of the current scenario are displayed. The next action to be carried out is then indicated in reverse video. The **Stop** button of this window enables the execution of the scenario to be stopped before it reaches its end.

• **Other recorded actions**

Besides the change in state of the external variables and operator panels, the setting and clearing of faults on sensors, relays and bistables are also recorded, together with the forcing of Boolean modelling variables.

For example, the sticking at 1 of a sensor has the following syntax :

- <sensor> : stuck at 1

• **Completion**

The “completion” function enables actions to be added to an already defined scenario. Accessed by the **Complete** button, it consists of rerecording a scenario already executed. The text of the Scenarios button is then replaced by **"Complete"** throughout the execution phase. Once this is finished, new actions can be recorded and are added afterwards.

2.8.2 Editing a scenario

A scenario consists of a series of lines. During execution these lines are processed sequentially, as seen in the window below.

As well as actions, a scenario can include wait periods and reactions. Since the principle of editing is similar for these elements, we will only deal at present with the editing of “action” lines which form the sequences.

The edit tool edits an existing scenario - obtained by recording - and also enables a new scenario to be created.

Deleting a line removes undesirable actions which may be caused, for example, by untimely activation of a pushbutton during recording.

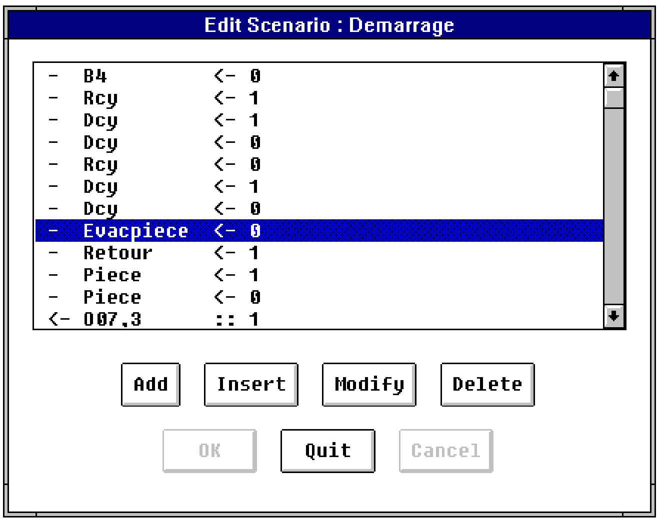
A line can be edited in order to modify or delete its associated execution wait period, especially in the case of the first line.

Finally, it is possible to add a new line at the end of the scenario or insert a line before a selected line.

The edit function can be accessed via the **Edit** button in the scenarios management window.

• Modifying a line

To do this, select the desired line and click on the **Modify** button. A window then appears for editing the scenario line. An editor located in the right part of this window contains the relevant text for the line. Modification can be performed directly via this editor : the syntax is checked by pressing “Enter”. Modification can also be performed by clicking on the **OK** button.



Another way of modifying the edited line is by using the **Revert** button in the edit window. This restores the various line elements starting at the end of the line. In the case of an action on an operator panel variable, it is possible to successively “revert” to the wait period associated with the action, to the action performed on the variable - change of state or pulse - or to the variable itself, etc.

Each time the **Revert** button is used, the substitute choices are displayed in the left part of the edit window. It is sufficient to select an element from this list in order to reconstruct a line, for which the syntax will be naturally correct.

• Creating a line

Two buttons are used for adding or inserting lines at the end of a scenario. The creation window is the same as for modification.

Initially, the editor containing the line text is empty and the selection list contains the various types of line which can be added or inserted into a scenario, including the type of line preceding and following the line to be created.

A new line can be created, in the same way as it is modified, by directly using the editor and keeping to the syntax or by successive selections from the list, the content of which is modified as the line is created.

Note :

After confirming the creation of a line, another line can be added afterwards without quitting the window.

2.8.3 Concept of wait periods

Sequences obtained by recording have the disadvantage that they do not contain synchronization conditions for controlling their execution.

However, certain procedures - notably the installation of automated systems - sometimes need to wait for a reaction from the control system (for example, a lit indicator lamp) for the rest of the sequence to be efficient.

In the context of operator training, it may be useful to execute actions, in particular setting faults, according to events generated by the evolution of the automated system.

“Wait periods” meet this requirement by postponing the execution of a scenario until an event occurs or as long as a condition has not been checked.

• Inserting wait periods by editing

One way of including a wait line in a scenario is by doing it a posteriori by editing the scenario obtained on recording.

To do this select the action - or the first of a group of synchronous actions whose execution is to be controlled by a wait period, and click on the **Insert** button. The selection list in the edit window should normally display the **Wait** item which is to be selected.

The types of variable which may occur in a wait line are then displayed.

For Boolean variables possible wait periods are, on one hand, the change to 1 or to 0 of a variable - in the case of an event - and on the other hand verification of logic state 1 or 0 of the variable - in other words a condition.

Corresponding syntaxes are as follows :

<- <var> -> 1 or <- <var> -> 0

and

<- <var> :: 1 or <- <var> :: 0

Traps :

On the one hand a distinction is made between traps which possess an event or not. In the first case the change to 1 of a trap, whose activity is transient, is expected. However, if the trap has no event, its activity - represented by its condition - is processed in the same way as other Boolean variables.

On the other hand a wait period can be formed by testing the counter associated with a trap by comparing it to a complete value. To complete this, a specific action enabling the reset to 0 of the trap counter is available : it must, if necessary, be inserted in the scenario using the edit tool.

• Recording wait periods

Another way of including a wait period in a scenario is to “prepare” the scenario before recording it.

Preparation of a scenario consists of assigning it variables - these can only be Boolean variables - before recording.

The preparation window, which can be accessed by the **Prepare** button, contains a selector switch in the top part of the window containing the **Wait** and **Reaction** items. The default setting is **Wait**, which is of most interest to us now.

The central part of the window is used to select the required type of variable : variables concerned appear on the left. If one is selected, it is placed in the list of Wait periods on the right of the window, it can be removed by clicking on it with the mouse. The required variable can be entered directly via the editor. It must then be confirmed so that the selected variables can be associated with the scenario.

During recording of the scenario, changes in state of the associated variables are automatically inserted in the actions performed in the form of wait periods.

• Executing a scenario which contains wait periods

During execution of a sequence of actions, operator panels are locked and setting of faults is inhibited.

If the scenario contains a wait line, the user has time to modify the interfaces as long as the wait period has not expired. A scenario can therefore be created which is not executed until a wait period expires, and then triggers a series of actions.

• Completing scenarios with wait periods

The principle of completion consists of rerecording a scenario which has already been executed.

If the line currently being executed is a wait instruction, it is reinserted in the scenario and control is passed over to the operator panels and to faults, but actions performed are not recorded until the wait period has expired. This enables a recording to be conditioned by including a wait period in a scenario, defined in edit.

2.8.4 Verifying reactions

The use of scenarios has, up to now, only been discussed in the form of sequences of actions whose execution can be controlled via wait periods.

This function can also be used to verify the behavior of automated system subassemblies using the “reaction” concept.

These reactions are the changes in state, obtained by recording, of variables associated with a scenario in the “preparation” phase.

In order to execute this scenario correctly later, the changes in state of variables, or “reactions”, have to be reproduced in the same way as during recording.

• Recording reactions

Firstly, we must define variables for which the changes of state are to be considered as reactions to be monitored.

This is done using the preparation window, which is displayed when recording wait periods. The principle of designating variables is similar, but this time the Reaction item in the upper part of the window must be selected.

Notes :

- A special variable entitled “Blocking” can be monitored. It is true when no movement occurs.

When recording is launched, the following window displays the scenario. This time it comprises, besides the changes in state of variables resulting from actions on the operator panels and external variables, changes in state affecting variables associated with the scenario in the form of reactions.

C

Reactions, in the same way as actions, give an indication of duration. During recording, the duration which is displayed in the window corresponds to the time elapsed since the last action or the beginning of recording. Once the scenario is saved, only the last line of a series of consecutive reactions retains this duration indication.

The syntax of a “reaction” line for the change to 1 of a variable is :

=> <var> -> 1 d:<duration>

As is the case for actions, reactions can be synchronous. This is the case if variables whose changes in state are immediate consequences of one another are associated with scenarios. On the appearance of an output, the latching of the relay and activation of the movement it controls are events which occur simultaneously with the change in state of the output. The syntax for this is :

=> <var1> -> 1 d:<duration>

<var2> -> 1

Notes :

- We have seen that a wait period enables the beginning of a recording to be conditioned on completion of a scenario. This is also true if the scenario contains reactions which are associated during preparation. Monitoring the evolution of corresponding variables is delayed until the wait period has expired.
- Wait periods can be recorded, if they are associated with a scenario via the preparation function. In this case there is exclusion between the “wait” variables and the “reaction” variables.

• Conformity of reactions

The execution of a scenario containing reactions may or may not conform to criteria that will be given based on a simple example.

Consider a scenario beginning with a wait period enabling the execution to be synchronized and containing only reactions. The wait period can be a trap characterizing the beginning of a cycle of the automated system subassembly, the reactions and movements of axes forming this module. This scenario could be constituted by including a wait line during editing.

The execution of a scenario consists, once the “start of cycle” wait period has expired, of verifying that the reactions - here the changes in state of movements of module axes -reappear in the same order and before a limit time corresponding to the duration associated with the last reaction recorded.

• Sequencing actions and reactions

If the scenario also contains actions, the principle of execution is as follows :

- actions are executed according to the wait period which may be assigned to them. While an "action" line is being processed - its execution wait period not yet elapsed - a check is made that no evolution of the variables associated with the scenario takes place.
- when a reaction is being processed, a check is made that it occurs well before the assigned wait period and that no other scenario variable evolution takes place.

• Execution report

The execution of a scenario can be displayed by clicking on the **Execution** button in the simulation window. Lines currently being processed appear in reverse video in the window.

In the case of scenarios with reactions, an execution report window is systematically displayed, it can be removed once the scenario has finished.

If the scenario execution is not correct, in other words if the reactions do not appear within the anticipated wait periods or if untimely evolutions occur, the "anomalies" are indicated in this window.

In this case simulation is interrupted and the scenario terminated. The window enables the user to see at what level in the scenario the deviation occurred.

• Completing scenarios containing reactions

In this case completion consists of rerecording the scenario - actions and evolutions of variables associated as wait periods or reactions - by firstly executing the actions of the initial scenario.

If the scenario contains wait periods, only those which are not after a reaction are retained for execution. In fact, since recording is postponed during wait processing, these reactions can no longer figure in the new scenario.

The essential use of completion in the case of reaction monitoring is to be able to correct a scenario in which errors have been made, with reference to the choice of associated variables during preparation on a first recording : it is sufficient therefore to modify these variables and to "replay" the scenario using its completion function.

2.9 Initialization

The **Initialize** command enables the machine model to be returned to how it was when the simulation was launched.

Everything takes place as if a new simulation is being started : the context selection window is displayed. If no context is selected, the operator panel variables and external variables remain in their original state.

If a machine state - or “mechanical context” - should have been saved during a previous simulation, it is possible to restore it.

In online mode, the PLC program should probably be reset in order that simulation can be correctly restarted.

2.10 Saving and restoring the machine state

When a simulation is launched, internally or with a PLC, the axes are in the initial position in which they were defined in the configuration interface and all variables whose state is described by Boolean expressions are at logic state “0”.

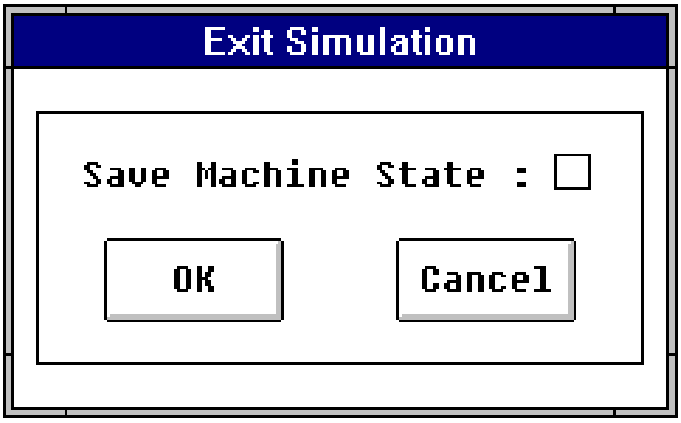
This may be disadvantageous however, when simulating a transfer system for example. In this case, it is better to be able to resume simulation if the machine is found in the state it was left for the previous simulation, with objects placed at certain points. The save machine state function offers this possibility.

The principle consists of being able to save, when quitting the simulation, the position of machine axes and the logic state of all the variables described in the “moving” part of the machine : in other words bistables and description variables. Online simulation can be performed using Grafcet charts, whose state is saved : in other words active steps, positioned internal actions and the value of counters.

The logic state of external variables or variables activated by operator panels is also retained : saving a machine state therefore includes the concept of context as defined in section 2.2 of this part.

This save may be requested when quitting simulation. To do this, check the **Save Machine State** box which appears in the confirmation window. By default, a save made previously is lost. A save can also be performed systematically if the simulation is quit in an “abnormal” way due to a bug or communication error with the PLC.

The machine state can be restored by checking the **Restore Saved State** box located in the context selection window which appears at the beginning of simulation.



2.11 Description

The “Description” function is used to display the description of the modelled machine during simulation and to make certain modifications.

This function can be accessed via the corresponding item in the **Tools** menu**.**

As regards machine modelling, it is possible to modify the logic equations for the inputs, relays and variables. It is also possible to modify the characteristics of the movements.

These modifications can all be made during internal simulation as this is a methodology phase aimed at debugging the model.

C

However, during online simulation, the modifications which can be made are limited to the “electrical” part of the model, that is, the input and relay logic equations.

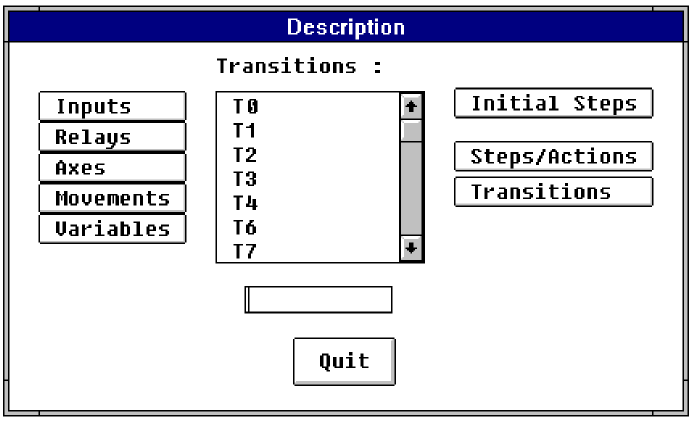
The sensors associated with the axes cannot be modified but they can be displayed in the same way as the description interface, in the form of a trend diagram or “evolutions”.

During internal simulation, it is also possible to modify the description of the control system. This means it is possible to correct the actions associated with the steps or transition conditions, for example.

The initial steps can also be modified, which makes it possible to test Grafcet charts independently.

Once simulation is complete, if modifications have been made, the simulator asks if these changes should be saved.

The old description files are then renamed with the extension .BAK.



3.1 Presentation

Internal simulation is generally executed using a Grafcet chart, which sequences the movements to reproduce the operating cycle of the simulated machine.

The “Mechanical Grafcet chart” is used to execute internal simulation without having to define an I/O configuration and a relay interface. These Grafcet charts are called “MGRAF<i>”.

I/O are implicitly created to enable communication between the Grafcet chart and the application model.

Inputs are created for any variable appearing in a transition condition and which do not correspond to an internal variable in the Grafcet chart, such as the end of a time delay or a step state.

In the same way, “fictitious” outputs are created for any step action for which no variable of the same name has been defined in the application model.

If an I/O configuration and a relay interface have been described, they are taken into account, even if this concerns simulation with a Grafcet chart. However, if the Grafcet chart actions have the same name as the relays, the relays are “skipped” during simulation.

Note :

> It is possible to execute an internal simulation without a Grafcet chart : the actuators are controlled “manually” by acting on the outputs, or directly on the relays or solenoid valves if the relay interface has not been described : these entities will appear as “external variables”.

3.2 Creating Mechanical Grafcet charts

A mechanical Grafcet chart is a Grafcet chart whose interface with the application is not defined explicitly by an I/O configuration.

Communication is performed by acting on the undefined variables of the application model via Grafcet chart actions : the actions for the steps whose name is not that of a variable defined in the model are “externalized” in the form of “implicit outputs” which may have an effect on the application.

C

This is also true for the relays in the relay interface which are not part of the power supply : these relays will not be taken into account if actions with the same name already exist in the mechanical Grafcet chart.

Conversely, the variables used in the transition conditions are directly the sensors and other variables in the application, as well as the operator panel variables. Additional external variables may be introduced if required.
Such a Grafcet chart can be created if the I/O have been configured : it is thus not allowed to use inputs in the transition conditions.

To access the interface for creating mechanical Grafcet charts, click on the CHART button in the SIMACTEL configuration interface.

3.2.1 Syntax

Grafcet charts are made up of steps and transitions linked by arcs. Actions are associated with steps and conditions are associated with transitions.

Step actions

The basic actions are actuator activations. They can also involve internal variables, or affect the time delays and counters.

Actions on Boolean variables

The actions can be set during activation of the step with which they are associated, or memorized and deactivated. In this case, the syntax is as follows :

<action> = 1 for memorization,

<action> = 0 for deactivation.

Activating time delays

An action can also be the start of a time delay. The syntax is as follows :

<Lti> if the time delay is activated with its default duration

or

<Lti> <duration>, where <duration> is a positive numeric value.

Operation of the time delay differs according to whether its activation is conditioned or not.

If there is no condition, the time delay is started when the step with which it is associated is activated and continues to run even if this step is deactivated.

However, if a condition is defined, the time delay is started when the state changes to the true state of the “logic and” formed by the activity of the step and this condition : this time delay is thus initialized on activation of the step if the condition becomes true at this moment, or when the condition is true when the step is active. If the condition is invalidated or the step deactivated, the time delay is reset to 0.

Operations on the counters

Several actions are possible on counters. They are assignment, incrementation and decrementation, and the arithmetic operations of addition, subtraction and multiplication.

The assignment syntax is as follows :

<counter> = <value>

where <value> can be a number or another counter.

These actions can be conditioned by Boolean expressions, where the syntax is the same as that for the transition conditions.

The syntax for incrementation/decrementation is as follows :

<counter> + and <counter> -

If no condition is associated with this operation, the operation is executed on activation of the step. If a condition is defined, the incrementation or decrementation is executed on the change to the true state of the “logic and” formed by the activity of the step and this condition : either on activation of the step if the condition becomes true, or each time the condition changes to “1” when the step is active.

For arithmetic operations, the syntax is :

<counter> <operator> <value>

where <operator> can be “ + ”, “ - ” or “ * ” and <value> is defined as for assignment.

These operations are performed when the step is active - and not on activation as is the case for incrementation/decrementation - and when the condition which may be associated becomes true.

If the action “cnt1 + 1” is defined in a step and the activity of this step is not “transient” in a single execution cycle, there will be an overflow on counter “cnt1”.

Transition conditions

These transition conditions are defined by “Boolean” expressions, in which predicates may be applied to the counter values.

Tests on the counter values are written as follows : <counter> <operator> <value> where :

<counter> is the counter concerned,

<operator> is a comparison operator which can be : < , >, < = ,> = , = or <>.

“Xi” terms take account of the activity of a step, in this case, step “n° i”, in a transition condition.

Edges on inputs or step activities can be tested by prefixing the corresponding variables with a “+” or “-” depending on whether it is a rising or falling edge.

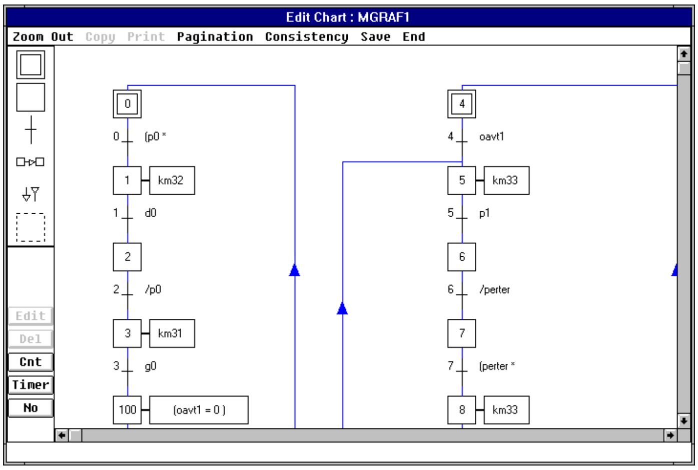
3.2.2 Interface for editing Grafcet charts

The edit window is composed of a main zone, dedicated to the drawing of the Grafcet chart.
This graphic page is actually 25 times larger than the part which is displayed on screen. It can be scrolled in order to move the zone displayed.

On the left-hand side of the edit window there are 5 icons symbolizing, from top to bottom, an initial step, a “normal” step, a transition, the link between two elements and a zone delimited by dotted lines.

There are also 5 buttons in the lower left-hand part of the window. The first 2 are used to edit and delete the components of a Grafcet chart, that is, the steps and transitions and the links between these elements. The next 2 buttons are reserved for creating counters and time delays.

C



Creating steps and transitions

To create a step or a transition, first select the corresponding icon on the left-hand side of the window using the mouse : this icon changes to reverse video when in create mode.

This mode can be exited by deselecting the icon using the right-hand mouse button. The element is created simply by clicking the left-hand mouse button in the graphic zone at the position in which you wish a step or transition to appear.
These elements are positioned at regular intervals on the graphic page based on a grid.

C

The element which has just been created is named in the left-hand side of the window, above the column of buttons.

It can be moved in the graphic page using the arrow keys on the keyboard, or directly using the mouse.

If a step or transition has been created by mistake, it can be deleted, using the Del button, after having selected the element which is not required. A confirmation is requested before the element is deleted definitively.

By default, the index of a step or transition recently created is the integer n+1, where n is the largest integer numbering an existing step or transition. The No. button is used to select the indexes from which the steps and transitions created are numbered.

Links between steps and transitions

The link between the elements can be established in two ways, depending on the graphic representation selected.

The link can be represented by an arc : in this case, the icon representing two elements linked by an arrow must be selected first in the left-hand side of the window.
The link can also be represented by a connector : this is useful when the trace of an arc in the graphic page would reduce clarity. In this case, select the icon representing two arrows, located below that selected previously.

The link is then established by first selecting the upstream element using the mouse : this is then identified by a small black square.

The downstream element is then selected. This is designated by the marker as being the upstream element for the next link. If this is not required, deselect the element using the mouse.

A link between two elements can be deleted if it has already been selected with the left-hand mouse button : the arc concerned is then designated in the left-hand side of the window and the Del button is used to delete it.

Blocks

The graphic interface offers the possibility of manipulating sets of steps and transitions.

To do this, first click on the icon representing a square in dotted lines on the left-hand side of the screen, which then changes to reverse video.

Steps and transitions are designated by framing them within a rectangle.

The top left-hand corner of this rectangle is positioned by clicking in the graphic page without releasing the mouse button.

If you drag the mouse while holding the button down, a rectangle will appear. The lower right corner of this rectangle follows the movements of the mouse. When the button is released, the rectangle is fixed on the screen : a set of steps and transitions is then designated.

The whole block can be moved at once using the mouse. It can also be deleted using the **Del** button, after confirmation.

A set of steps and transitions can also be duplicated. Having made the selection as explained earlier, click on the Copy button at the top of the window.

Clicking the mouse in the graphic zone positions the top left-hand corner of the block to be copied, which is then copied after confirmation.

Zoom out

We have seen that the Grafcet chart is described in a graphic page of which 1/25th is displayed.

To be able to have a global view of the Grafcet chart, a zoom out function is offered, which can be activated using the button at the top of the window.

In zoom out, the zone displayed in edit mode is represented by a rectangle. This rectangle can be moved using the mouse and the "Carriage return" key is used to return to edit mode.

The function for managing blocks, described earlier, can also be accessed in zoom out.

Editing steps and transitions

The graphic page is used to describe the structure of the Grafcet chart. For this to be useful, we need to define its "interpretation", that is to associate actions with the steps and conditions with the transitions.

This is achieved using the windows provided for this purpose, which are activated using the **Edit** button, having first selected a step or transition in the graphic zone. The **Previous** and **Next** buttons in these windows are used to review all the elements in the Grafcet chart belonging to the selected type.

Describing the steps

The upper part of the description window includes a selection box indicating whether it is an initial step or not : this characteristic can be modified by clicking on the box with the mouse.

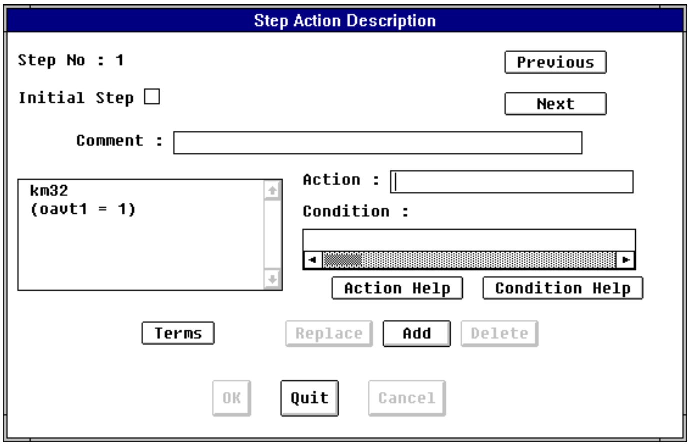
The actions associated with the step appear in a scroll-down list on the left-hand side of the window. If one of these actions is selected, it is displayed in the **Action** editor, and the condition which may be associated with it appears in the corresponding editor.

This action and its condition can then be modified, the substitution being made using the **Replace** button. The selected action can also be deleted using the corresponding button.

To create a new action, it must be defined in the **Action** and **Condition** editors and then added using the **Add** button.

There are Help functions available for defining actions. The Action Help button is used to select an internal action, a time delay or a counter.

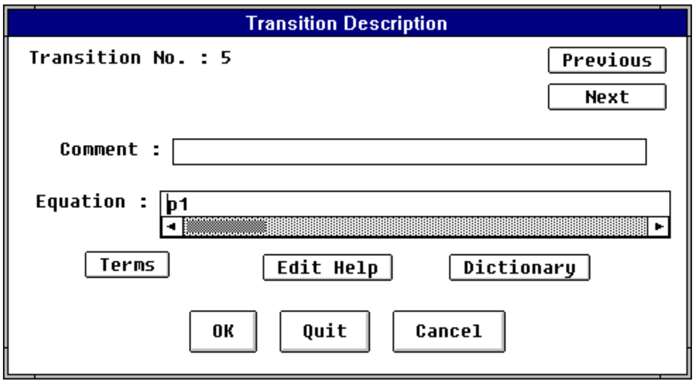
The Condition Help button is used to select terms which may be used when expressing the condition of an action, ie. internal actions, counters and end of time delay variables.



C

Describing the transitions

This description consists of defining the condition associated with the transition. The syntax is the same as for that of the step action conditions, and a similar Help function is offered.



Creating the counters and time delays

In the Grafcet chart steps, time delays can be activated or operations can be performed on the counters.

In the transitions, end of time delay variables can be taken into account and counter values can be tested.

These entities must be previously defined to be available for use in the actions and conditions : this is done using the **Cnt** and **Timer** buttons.

Counters

The **Cnt** button is used to open a window where the counters defined for the Grafcet chart being edited are displayed.

To create a new counter, enter an integer index in the editor in the right-hand side of the window and press the **Add** button.

A counter name is always in the format “Cnt<i>” where <i> is an integer. If integer <i> is such that counter “Cnt<i>” already exists, a message signals the clash and the first unassigned integer is offered in the editor.

Time delays

The window is used to create new time delays and to modify the duration of existing time delays.

To modify the duration of a time delay, select the term “Lt<i>” corresponding to the activation of this time delay. The duration of the delay is then displayed in an editor where it can be modified : this is taken into account using the **Modify** button.

To create a time delay, use the **New Time Delay** button : the smallest integer not already corresponding to an existing time delay then appears in the **Time Delay** editor. The default duration is 1, although it can, of course, be modified.

The time delay to be created can be entered directly in the **Time Delay** editor.

The **Add** button is used to take account of the new time delay, provided that the same time delay does not already exist.

A time delay can be added to the actions of a step by the step editor interface. Simply enter its name, that is “lt” followed by its number, or select the list of time delays using the **Action Help** button.

The end of the time delay in a transition equation is defined by the variable “ft” followed by the number of the time delay.

Note :

It is not possible to delete counters and time delays from the interface. Deletion is actually implicit, only the time delays and counters used in the Grafcet chart are saved when the user exits the edit screen.

3.2.3 Printing Grafcet charts

SIMACTEL can be used to print any selection block created using the mouse. The **Print** button in the top part of the window triggers, after confirmation, printing of the contents of this block. However, if the size of the rectangle representing the block is not compatible with the printer, an error message signals that this command has been aborted.

To simplify the printing of Grafcet charts, SIMACTEL provides the user with an automatic page setup function. This is activated using the **Pagination** button at the top of the window. The Grafcet chart is then displayed in zoom out by showing each page created by a rectangle with a “shadow”. The size and locations of these rectangles are calculated according to :

- the location and proximity of “sub-charts”,
- the parameters defined for the default printer : size and source of the paper.

Printing, triggered using the **Print** button, involves several options :

- Print all pages : all pages created are printed.
- Print selected pages : a page is selected by clicking in its rectangle; the page is then grayed out.
- Print a selection zone : the user may require this function when a chart is too long or too wide to be printed on a single sheet of paper. In this case, the chart is printed manually in successive selections of zones to be printed.

The document printed by SIMACTEL is made up of :

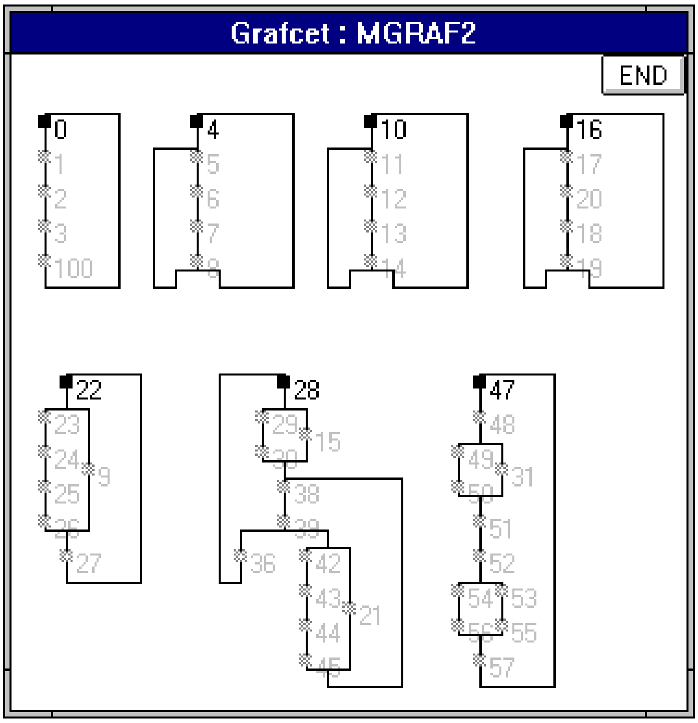
- One page containing the graphic representation of the GRAFCET CHART in which the step and transition references are found,
- One or more pages detailing each step and each transition for the selected part of the GRAFCET CHART.

3.3 Animation in offline mode

If a Grafcet chart is being executed, a view of the same name can be displayed. It contains the Grafcet chart as it has been described, in **Zoom Out** mode (reduction to 1/5th) with the steps, transitions and links. The active steps are “lit up” as the Grafcet chart evolves.

Once simulation has stopped, it is possible to open the examination window for a step or transition directly by clicking on this step or transition in the window.

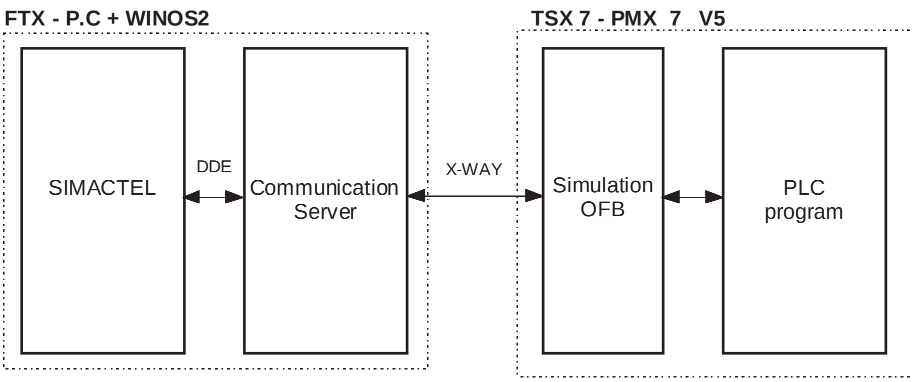
C



4.1 Principle of PLC / SIMACTEL exchanges

4.1.1 Introduction

In online mode, SIMACTEL exchanges the I/O with the PLC to simulate the behavior of the application. These exchanges are performed using the two software modules : the communication server and the simulation OFB. SIMACTEL communicates with the communication server which exchanges the I/O with the simulation OFB. The simulation OFB is responsible for providing the PLC with the I/O changes. The diagram below illustrates the I/O exchanges between SIMACTEL and the PLC.



4.1.2 Basic principle

SIMACTEL and the PLC communicate via the communication server which is responsible for exchanging the I/O with the simulation OFB. Thus, when SIMACTEL advances the application model, the changes to the inputs are communicated to the server. The server then communicates this to the OFB which then copies them to the PLC. The changes to the outputs resulting from the reaction of the PLC program are then returned to SIMACTEL by the OFB and the server.

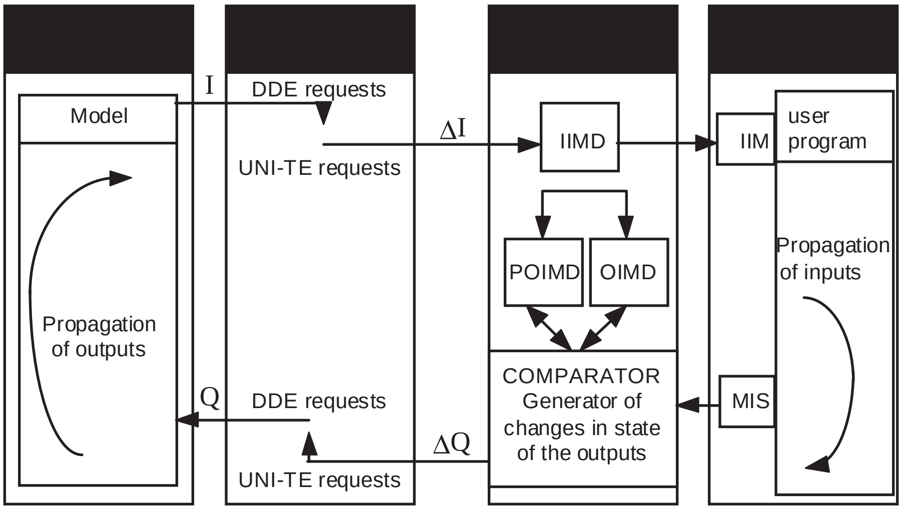
C

To provide the PLC with the changes in state of the inputs, the OFB has commands for writing directly to the PLC memory. The changes to the outputs are obtained via a direct read command from the PLC output image memory. By comparing this to the former contents memorized previously, the OFB deduces the outputs for which the state has been modified. In fact, the OFB keeps its own image memory of the I/O. A wait time used to stabilize the outputs of the PLC program is respected between sending the input changes via SIMACTEL and returning the output changes. This time, known as the “watchdog”, is generally set at 3 times the scan time of the MAST task. However, it can be modified : see section 4.3.2.

4.1.3 Mechanism for the exchanges

The OFB has an Input Image Memory (IIMD) which is updated from the changes to the application model (machine), controlled by SIMACTEL and provided by the server; this IIMD is then transferred to the PLC IIM.

The Output Image Memory of the OFB (OIMD) is updated on each PLC scan and compared to the contents of a previous Output Image Memory (POIMD), used to calculate the changes in state of the outputs which will be propagated in the machine model.



Note :

The fact that the OFB operates at the start and end of the PLC scan, and that it is substituted in the exchanges means that the explicit read and write operations placed in the user program (READBIT and WRITEBIT instructions) are not taken into account.

4.2 Setup

4.2.1 Hardware setup

Setting up a simulation platform requires that PLC-SIMACTEL communication be set up as a current loop, RS485 or RS232. This communication is based on the services offered by X-WAY. In the case of single station SIMACTEL, the hardware setup described below concerns the terminal port and UNI-TELWAY links; all other X-WAY services are compatible (FIPWAY, MAPWAY, ETHWAY) as long as the corresponding physical supports are used.

C

The PLC/SIMACTEL connection is via the RS485 or RS232 CL port on the FTX, or via the RS232 port in the case of a PC.

For type xx.455 CPUs, the use of an SCM 21 communication card will be required for any RS485 simulation (UNI-TELWAY).
As an example, the connections required when using an FTX terminal are listed below.
FTX - PLC connection

The possible connections are as follows :

• FTX CL output	>>	Terminal port
• FTX RS 485 output	>>	Integral UNI-TELWAY port (CPU xx.425)
• FTX RS 485 output	>>	SCM 21 port
• FTX RS 232 output	>>	Terminal port (TSX TE 01 converter)
• FTX RS 232 output	>>	Integral UNI-TELWAY port (Cpu xx.425) (TSX SCA 72 converter)
• FTX RS 232 output	>>	SCM 21 port (TSX SCA 72 converter)

4.2.2 Simulating I/O modules

During SIMACTEL simulation, the user can select various in-rack and remote I/O modules. He can then define the modules he wishes to be simulated by SIMACTEL and those which do not need to be. The various configuration possibilities and their checks are detailed below.

- I/O modules simulated by SIMACTEL : this type of configuration is the most commonly used and enables SIMACTEL to simulate the I/O of these modules. The OFB checks the presence of these modules in the PLC configuration and blocks the start-up of the simulation if the modules do not exist or if they differ from those declared in SIMACTEL. The modules may or may not be physically present.

- I/O modules not simulated by SIMACTEL and declared as such in SIMACTEL : this is especially used with output modules as it may be necessary, during installation on site, to wire a certain module. This is used to retrieve in SIMACTEL the state of their outputs to change the model of the simulated application. The OFB checks the presence of these modules in the PLC configuration and blocks the start-up of the simulation if the modules do not exist. It displays a non-blocking message for the simulation if the unsimulated modules are present in the PLC configuration but are not physically present.

- I/O modules not declared in SIMACTEL but present in the PLC configuration : this is used to leave out any physically wired I/O modules which are not used in the evolution of model of the simulated application. The OFB displays a non-blocking message for the simulation if the modules are not physically present.

4.3 Configuring the simulation OFB

4.3.1 Principle of loading the OFB

Before starting a simulation, the SIMACTEL OFB must be loaded into the PLC application using the following operations (refer to the PL7-3 V5 Operating Modes manual, sections 6.1, 6.7 and 6.11 of part B, section 10.3 of part F and section 14.2 of part G) :

C

- Access the SELECT CONFIGURATION MODE screen via the CONFIGURE command in the MODE SELECTION screen. This mode includes 7 headings, one of which is used to configure the Optional Function Blocks.
- Access the CONFIGURE OPTIONAL FUNCTION BLOCKS screen via the OPTIONAL FUNCTION BLOCKS command. This screen is used to load the OFB into the PLC application. The OFBs available are divided into families, thus an instance of the SIMACTEL OFB from the SIMACTEL family should be loaded.
- Use the command [NEW OFB] to list the OFBs available on the disk.
- Find the SIMACTEL family using the commands [PREV FAM] and [NEXT FAM]. The SIMACTEL OFB appears in the left-hand list.
- Use the command [INS] to select the OFB and confirm with ENTER.
- Select 1 instance of the OFB using the command [MODIFY], then the number 1 then confirm with Enter. The application must now be reconfigured for the OFB to be taken into account.
- From the SELECT RECONFIGURATION MODE screen, choose the RECONFIGURATION DIAGNOSTICS command which is used to check the consistency between the user program and the new configuration. At the end of the check, if everything is correct, the message CHECK CORRECT is displayed. Note that it is possible to abort the modifications using the [ABORT] command before reconfiguring the application.
- Then use the RECONFIGURE command to transfer the new configuration to the memory reserved for the application. Confirm the reconfiguration with the command [YES]. If the configuration is correct, the user returns to the MODE SELECTION screen and the message RECONFIGURATION CORRECT is displayed.
- Finally reload the application into the PLC and the PLC program is ready to be simulated.

4.3.2 Setting the OFB

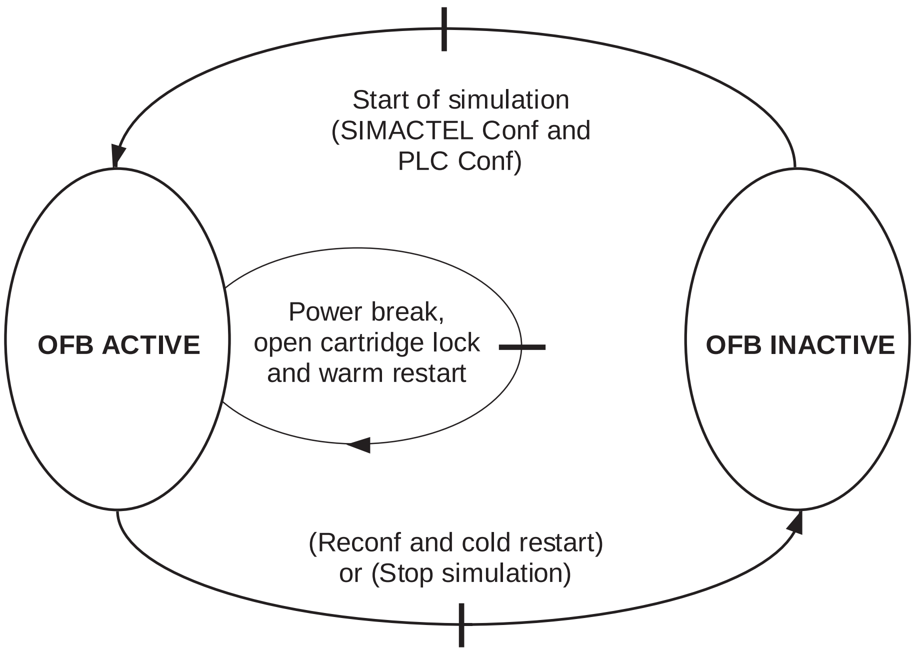
We have seen in section 4.1.2 that it was possible to wait for a certain number of PLC scans after having sent the inputs to return the outputs to SIMACTEL. As a general rule, 3 PLC scans are enough for the evolution of the user program to be considered stable. However, it is possible to modify this “Watchdog” via an OFB constant called “WATCHDOG”. To do this, proceed as follows :

- Access the CONSTANT MODE screen using the CONSTANT command in the MODE SELECTION screen.
- Use the [OFB] command to display and modify the OFB internal constants.
- Select the SIMACTEL OFB using the [SELECT] command and its number, then confirm this, which displays the internal constants of the OFB.
- Modify the value of the “WATCHDOG” constant using the [MODIFY] command then confirm. If you are in online mode, the modification is taken into account immediately, otherwise it will be valid the next time the application is loaded into the PLC.

C

4.3.3 Operating the simulation OFB

When the simulation OFB is loaded in the PLC application, it activates when the simulation has started. It is thus possible to leave it in the PLC program if you wish to execute operations other than simulation. When simulation starts, the OFB becomes active in order to manage the I/O exchanges between the PLC and SIMACTEL. At the end of simulation and until the next simulation, it remains inactive. However, in certain cases, the simulation OFB can remain active (in the event of a power break during simulation, for example). The various operating modes of the simulation OFB are explained later.



4.4 Communication server

4.4.1 Function

The communication server transmits information between SIMACTEL and the simulation OFB. When SIMACTEL advances the application model, it communicates input changes to the server. After writing these inputs in the PLC via the OFB, the server, on request from SIMACTEL, returns the output changes and transmits them to SIMACTEL. The server then has all the information on the I/O exchanged, which is why the server interface offers several functions which are explained below.

C

4.4.2 Communication parameters

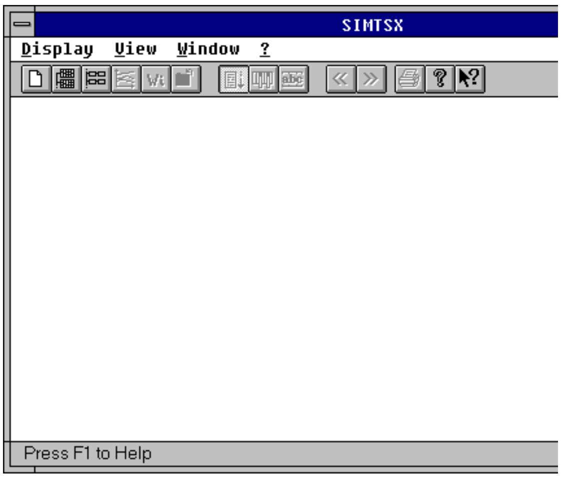
Two parameters are required to establish communication between SIMACTEL and the PLC :

- the address of the PLC connected to the simulator
- the type of driver used for communication

These two parameters are transmitted automatically by XTEL at the start of simulation.

4.4.3 The interface commands

The server interface is composed of pull-down menus and a toolbar used to access the commands. This part details for each menu the role of each of the interface commands.



4.5 The Display menu

The Display menu contains the following commands :

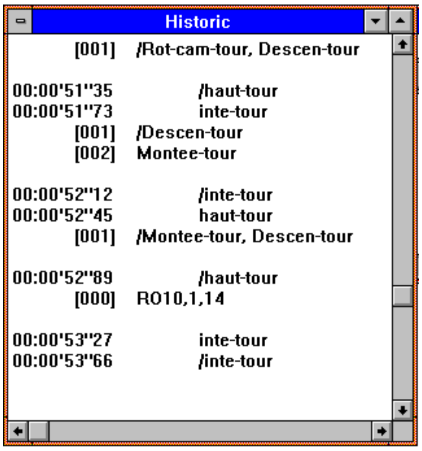
Historic	Displays the historic of the I/O.
In-Rack I/O	Displays a view of the In-Rack I/O.
Remote I/O	Displays a view of the Remote I/O.
Close	Closes the active document.
Print...	Prints the historic of the exchanges.
Print Setup...	Changes the print and printer parameters.

4.5.1 Historic command

The Historic command is used to display a window which scrolls through the successive changes to the I/O during simulation. These changes are timed in two different ways : for the inputs sent by SIMACTEL, the time from the start of simulation is displayed before each series. For the reception of output changes, the number of the PLC scan required to make these outputs appear from the last input transmission is displayed. The I/O are displayed via the Edit command; they belong to the modules configured using the Configure command. They are shown by their address or by their mnemonic (using the Mnemonics command).

• Historic window

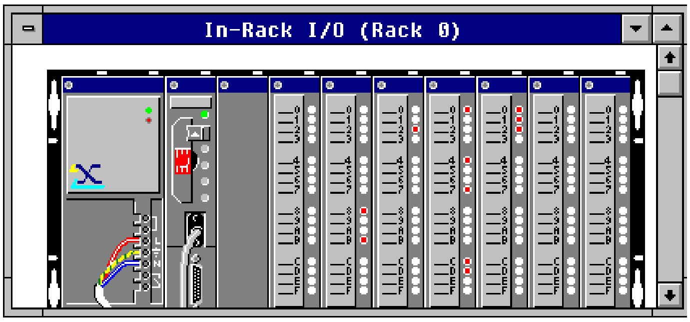
In this window, the exchanges between SIMACTEL and the PLC are displayed. The display is effective as soon as the Edit command is used. Only the I/O configured using the Configure command are displayed. If the Mnemonics command is used, the mnemonics are displayed. The sign (a) before certain outputs mean that these outputs appear in the PLC more than n cycle(s) after the changes to the inputs are sent by the simulator. N corresponds to “WATCHDOG”, the OFB wait variable and can be adjusted from 0 to 10.



4.5.2 In-Rack I/O command

Use this command to display the view of the in-rack I/O. Only the racks configured in SIMACTEL can be displayed. All the I/O modules (4,8,16 and 32) are represented by modules with 16 I/O for reasons of legibility. If 32-input or 32-output cards are used, the first 16 values are displayed on rack i and the next 16 on rack i+1, in the same slot. The change from one rack to another is made using the Previous and Next commands. This makes it possible to open several views of in-rack I/O if several racks need to be displayed at once. A double-click on one of the modules configured opens a dialog box containing the corresponding I/O with their values and associated mnemonics. The I/O modules which are not simulated, declared in the SIMACTEL configuration, are represented by cross-hatching on the view.

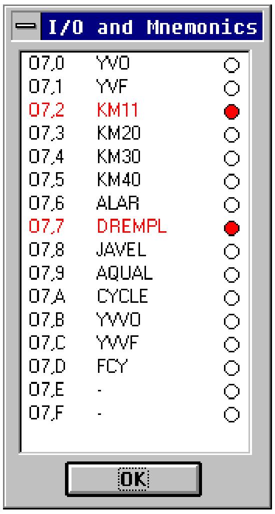
C

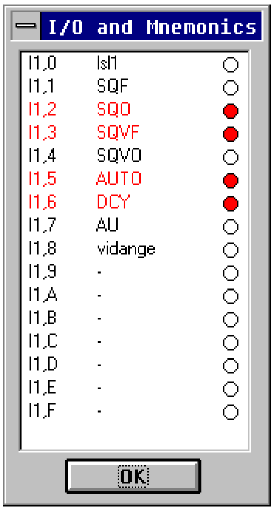


• State of the I/O and Mnemonics dialog box

The list represents the correspondence between the I/O and their mnemonics. It also indicates their state by the color of the text and a radio button displayed next to them.

- red : state 1
- black : state 0

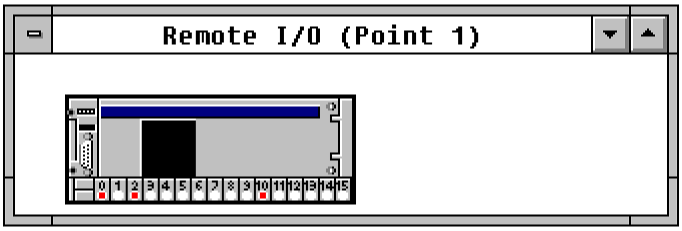




4.5.3 Remote I/O command

Use this command to display the view of the Remote I/O. Only the connection points configured in SIMACTEL can be displayed. For a connection point, the extension is only displayed if it exists. The change from one point to another is made using the Previous and Next commands. This makes it possible to open several views of remote I/O if several connection points need to be displayed at once. A double-click on one of the modules configured (base or extension) opens a dialog box containing the corresponding I/O with their values and associated mnemonics. The I/O modules which are not simulated, declared in the SIMACTEL configuration, are represented by cross-hatching on the view. If the base and extension are not simulated and only the extension is declared as such in SIMACTEL, the base will still be represented on the view but grayed out.

C



4.5.4 Close command

Use this command to close the active window. This command cannot be accessed if all the windows are closed.

4.5.5 Print... command

Use this command to print an historic. This command opens a Print dialog box in which you can select the pages, number of copies and the printer and access all the print options.

Shortcuts :

Toolbar :
Keys : CTRL+P

C

4.5.6 Print Setup command

Use this command to change the printer and print parameters. This command opens a Print Setup dialog box in which you can select the printer and access all the print options.

4.6 The Historic menu

The Historic menu is displayed when the Historic window is active.

The Historic menu contains the following commands :

Edit	Displays the selected I/O exchanges between SIMACTEL and the PLC.
Configure...	Used to select the I/O to be displayed.
Mnemonics	Used to select the mode for displaying the I/O.

C

4.6.1 Edit command

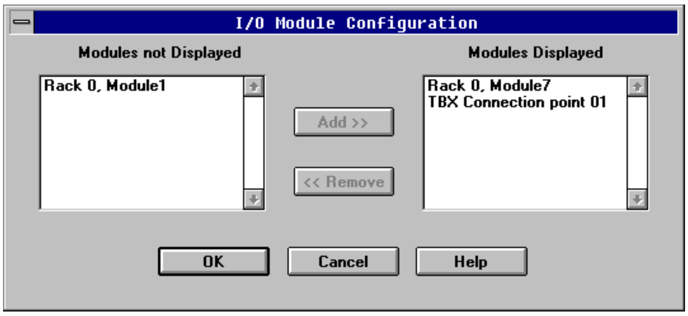
Use this command to display in the Historic window of the server the exchanges between SIMACTEL and the PLC. The I/O displayed belong to the modules configured using the **Configure...** command.

4.6.2 Configure command

Use this command to select the I/O modules which you wish to display when the **Edit** command is active. This command is then used to filter the I/O to be displayed.

• Module Configuration dialog box

The list on the right-hand side represents the I/O modules to be displayed in the Historic window of the server when the **Edit** command is active. By default, all the I/O are ready to be displayed.



To remove or add modules, the various options are as follows :

- Left mouse button to select an element then button

or

- Double-click on an element to be added or removed.
- SHIFT + left mouse button to select several adjacent modules then button

or

- CTRL + left mouse button to select several nonadjacent modules then button

or

C

4.6.3 Mnemonics command

Use this command to display in the Historic window of the server the exchanges between SIMACTEL and the PLC in the form of mnemonics. If the mnemonic does not exist, the address of the corresponding input or output will be displayed.

4.7 The View menu

The View menu is displayed when a view window is active.

The View menu contains the following commands :

Previous	Displays the view of the previous rack or connection point.
Next	Displays the view of the next rack or connection point.

4.7.1 Previous command

Use this command to return to the previous view. This command is common to in-rack I/O and remote I/O. It is effective for the active view window.

4.7.2 Next command

Use this command to move to the next view. This command is common to in-rack I/O and remote I/O. It is effective for the active view window.

4.8 The Reveal menu

The Reveal menu contains the following commands :

Toolbar	Shows or hides the toolbar.
Information bar	Shows or hides the information bar.

C

4.8.1 Toolbar command

Use this command to show or hide the toolbar.

4.8.2 Information bar command

Use this command to show or hide the information bar.

4.9 The Window menu

The Window menu contains the following commands :

New Window	Duplicates the active window.
Cascade	Arranges the windows in cascade.
Tile	Arranges the windows in tile format.
Arrange Icons	Aligns the icons of the minimized windows.
1 2...	Activates the window specified.

C

4.9.1 New Window command

Use this command to duplicate the active window. In the case of the historic, this is used to compare two different historic sequences.

4.9.2 Cascade command

Use this command to display the open windows in a cascade.

4.9.3 Vertical Tile command

Use this command to display the open windows in a vertical tile format.

4.9.4 Horizontal Tile command

Use this command to display the open windows in a horizontal tile format.

4.9.5 Arrange Icons command

Use this command to arrange the icons of the minimized windows.

4.9.5 1 2 ... command

Use this command to display the list of open documents and to activate the window required. The active window is that which is preceded by a marker in this menu.

C

4.10 The Help menu

The Help menu contains the commands for accessing the online Help function for the application :

Index	Displays the list of subjects on which you can obtain Help.
Using Help	Gives information on how to use the Help function.
About ...	Displays the application version number.

4.10.1 Index command

Use this command to access the first Help screen in the communication server. All functions of the online Help can be accessed from this screen.

Wherever you are in the online Help, you can click on the **Index** button to return to this initial screen.

4.10.2 Using Help command

Use this command to find out how to use the online Help.

4.10.3 About... command

Use this command to obtain the version number.

4.11 Miscellaneous

4.11.1 The Toolbar

The toolbar is displayed across the whole width of the application, just below the menu bar. The toolbar provides quick access to the main server commands.

Click on

Opens the historic.

Opens a view of the in-rack I/O.

Opens a view of the remote I/O.

Closes the active view.

Displays the configured I/O.

Used to configure the I/O.

Displays the I/O in the form of mnemonics.

Displays the previous rack or connection point for the active window.

Displays the next rack or connection point for the active window.

Displays information about the product.

Displays the contextual Help cursor.

4.11.2 The Information bar

Press F1 to obtain Help.

The information bar is displayed at the bottom of the main server window.

The left-hand side of the information bar explains the actions associated with the menu items when you move around the menus using the arrow keys. Similarly, this zone displays messages relating to the buttons on the toolbar when you press these buttons without releasing them. If, after having read the message associated with a toolbar button, you do not wish to execute the command, release the mouse button outside the button zone.

4.11.3 Contextual Help command

Use the contextual Help command to obtain information on part of the application. When you select the contextual Help tool from the toolbar, the mouse pointer is replaced by an arrow and a question mark. Then click on any point in your application, another button on the toolbar for example. The online Help corresponding to this item is displayed.

Shortcuts

Keys : SHIFT+F1

4.11.4 The Title bar

The title bar is displayed at the top of each window. It contains, for the main window, the name of the server followed by the name of the active window, and, for secondary windows, only the name of the window. For the view windows, the number of the rack (or connection point) displayed appears in the title bar.

To move the window, hold the mouse button down on the title bar and move the mouse. Note : You can also move the dialog boxes in the same way.

The title bar can contain the following items :

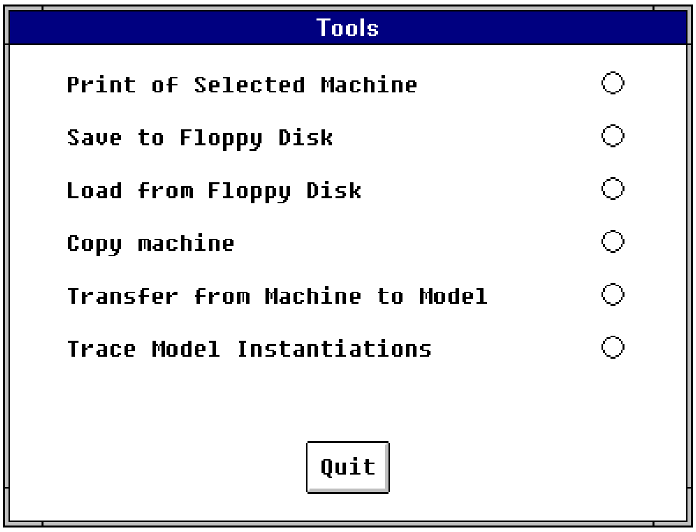
- Button activating the application Control menu
- Button to maximize the application
- Button to minimize the application
- Application name
- Restore button

C

1.1 Presentation

The functions described in this section are used to transfer machine files. To access them, select the **Tools** button in the main **configuration interface** window**.**

The Tools button is also used to access other SIMACTEL software functions explained in other sections of this manual (see section 1.4 of part B).

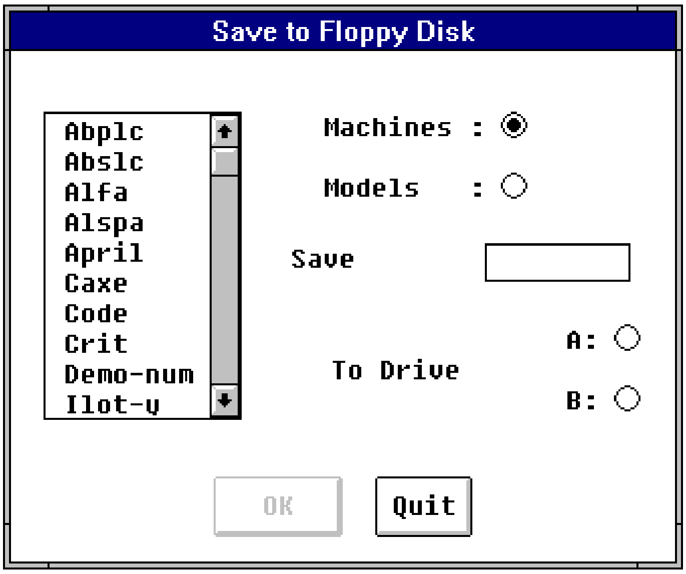


D

1.2 Saving to floppy disk

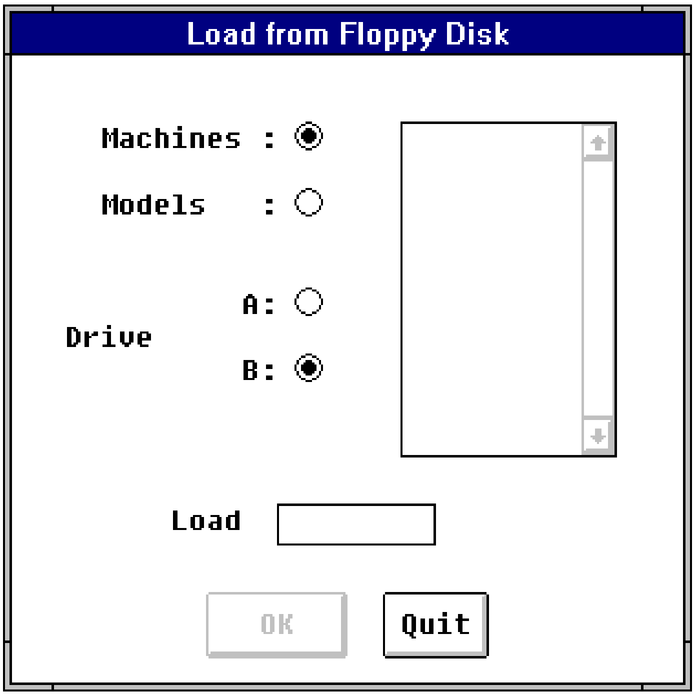
Select the machine to be saved from the scroll-down list on the upper left-hand side of the window. Its name is displayed in the editor to the right. Select the disk drive to which you wish to copy the machine by clicking one of the radio buttons (A: or B:). The OK button triggers the command.

A Windows error message appears if there is not enough space on the disk or there is no disk in the selected drive. In this case, solve the problem and retry.



1.3 Loading from a floppy disk

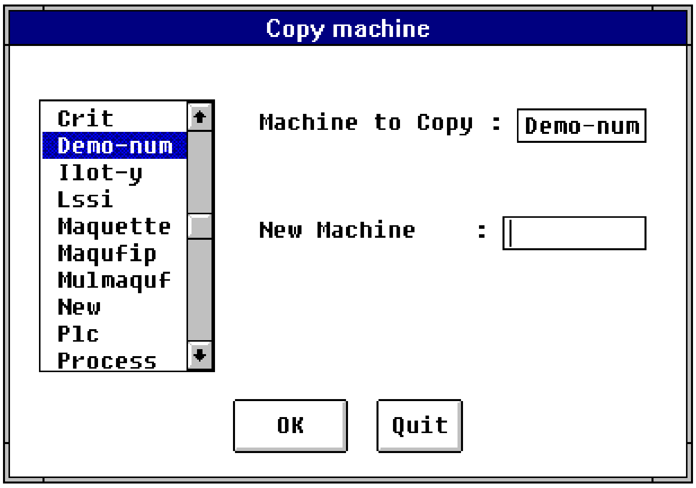
Select the disk drive from which you wish to copy the machine by clicking on one of the radio buttons (A: or B:). Select the machine to be loaded from the scroll-down list on the right-hand side of the window. This scroll-down list contains all the machines stored on the floppy disk. The OK button triggers the command.



D

1.4 Copying a machine

Select the machine to be copied from those displayed in the scroll-down list on the left-hand side of the window. The name of the selected machine is displayed in the editor to the right. Type the name of a new machine in the editor immediately below. A machine name can be a maximum of eight characters. Click on the OK button to trigger the command. The Quit button exits the window.



D

2.1 Presentation

The SIMPRINT application is a Windows application dedicated to printing the SIMACTEL machine dossiers and models. It can be launched from the Tools function in the description interface.

This section covers this application. It is divided into two parts. The first, the SIMPRINT application, introduces the application operating principles. The second, controlling the interface, is the reference manual.

2.2 The SIMPRINT application

The SIMPRINT application is used to print SIMACTEL machine dossiers and models according to a large number of parameters. Some have an effect on the format of the printed dossier, for example the choice of character font, while others affect its contents. The SIMPRINT print formats are used to archive these parameters and reuse them in other printed dossiers.

2.2.1 Print formats

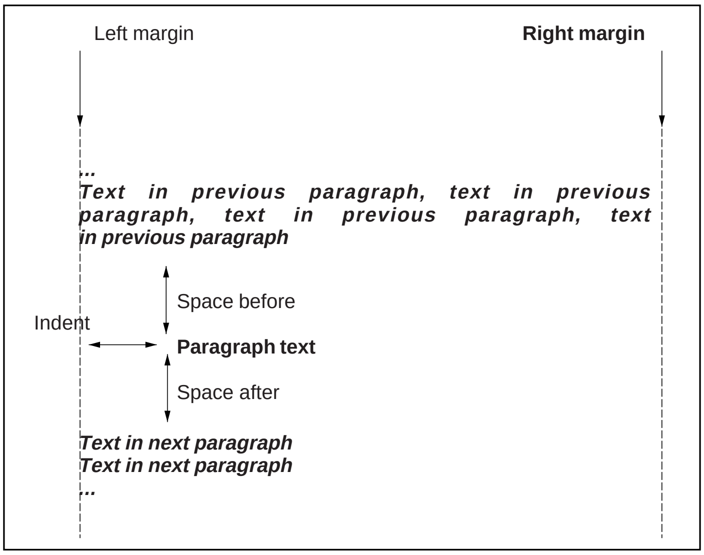
A print format is a file used by SIMPRINT to print machine dossiers and models. The information contained in this file determines which description elements are to be printed, which character fonts are to be used, or even how much space should be left between the edges of the paper and the text. Generally, SIMPRINT print format files are in “*.FMT” format. The SIMPRINT interface is used to load, modify and save these files.

The active print format is that which is loaded into the SIMPRINT application. SIMPRINT organizes the printing of data in line with this print format. Its name is given in the title bar of application1. When a machine or SIMAC model is loaded, SIMPRINT simultaneously activates the print format “DEFAUT.FMT” which is in the directory in which the SIMAC software is installed. In this way, any dossier is ready for printing as soon as it is opened. This format can be modified just like all the others.

The print format management commands are divided into two sets. The first is found under the File menu and contains all the print format file management commands. These are used to load and save the print formats (see section 2.3 of this part). The second set is found in the Format menu. The parameters of the active print format can be configured from the application interface. The General, I/O configuration, Relays, Axis and Panel commands are applied to the content of the printed dossier. They are used to select the description elements to be printed (see section 2.5 of this part).

SIMPRINT structures the printed dossier into paragraphs to which a style is associated. For each style, the following characteristics can be set :

- page break : inserts a page break before the paragraph,
- indent : configures the free space to the left of the paragraph,
- before : configures the free space above the paragraph,
- after : configures the free space below the paragraph,
- font : selects the character font to be used for the paragraph.



The styles used by SIMPRINT are as follows :

- Main title
- Identification
- Heading 1
- Heading 2
- Heading 3
- Element
- Trend diagram
- Normal text
- Header
- Footer
- Panel & View
- TOC 1
- TOC 2
- TOC 3

Main title

This style defines how to print the title of the print dossier (the first line of the dossier).

Identification

This style defines how to print the text in the fields on the first page. The fields printed are those which have been filled in in the main application window. Empty fields are ignored.

Heading 1, Heading 2, Heading 3

These styles define how to print the various titles contained in the dossier. Heading 1 relates to the various sections of the print dossier (I/O configurations, power supplies, relays, axes, variables, bistables, operator panel variables, operator panels and views). Headings 2 and 3 correspond to the subsections within these sections. The correspondence between text and styles depends on the print options selected by the user.

Element

Use this style to define how to print the names of the description elements (variable, relay, etc).

Trend diagram

This style defines how to print the trend diagrams for the sensors on the axes.

Normal text

This style defines how to print normal text in the print dossier.

Header

This style defines how to print the text in the header on each page of the dossier. The only useful parameter in this style is the character font.

Footer

This style defines how to print the text in the footer on each page of the dossier. The only useful parameter in this style is the character font.

TOC 1

This style defines how to print level 1 headings in the table of contents. These correspond to the Heading 1 style in the print dossier.

TOC 2

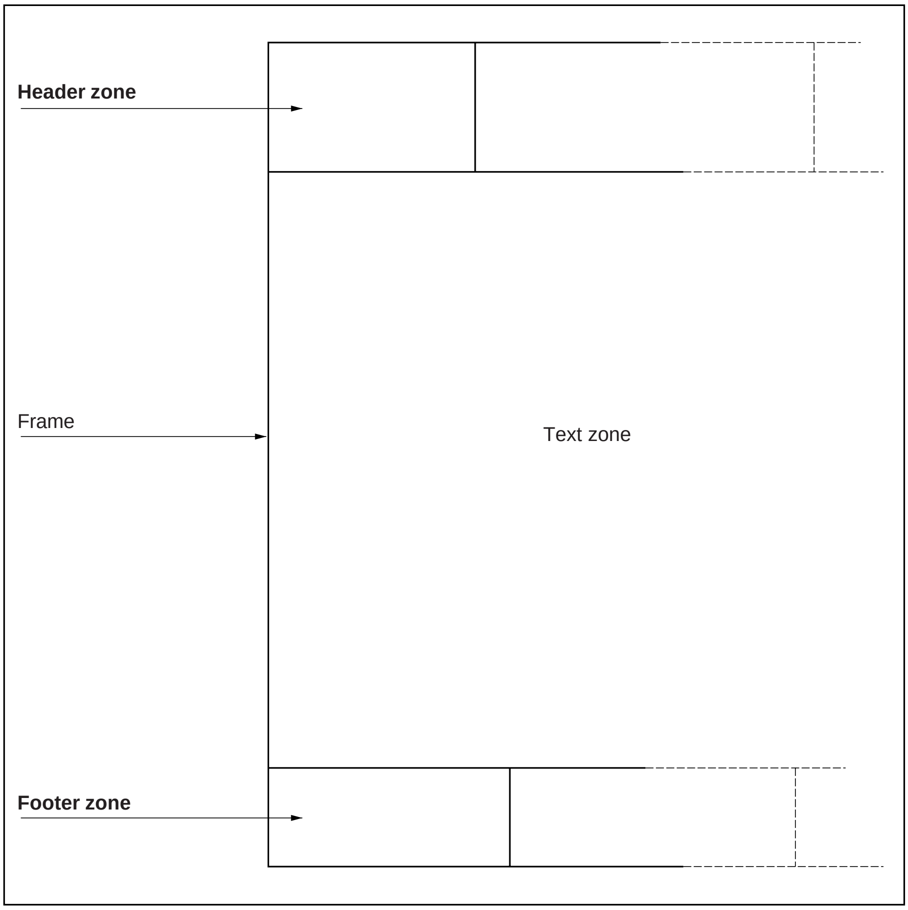
This style defines how to print level 2 headings in the table of contents. These correspond to the Heading 2 style in the print dossier.

TOC 3

This style defines how to print level 3 headings in the table of contents. These correspond to the Heading 3 style in the print dossier.

Panel & View

This style defines the space required around the drawings of the operator panels and views. Selection of the character font has no effect on this style.



D

2.2.2 The link with SIMACTEL

Use the **Tools** button in the SIMACTEL configuration interface to start printing a dossier. In the Utilities window, select the item for printing the selected machine. If the Print to “default printer” window does not appear immediately, click on the OK button in this window.

The Print to “default printer” window is used to open the following :

I/O Conf. Prints the I/O configurations.

Relays Prints the relays and power supplies.

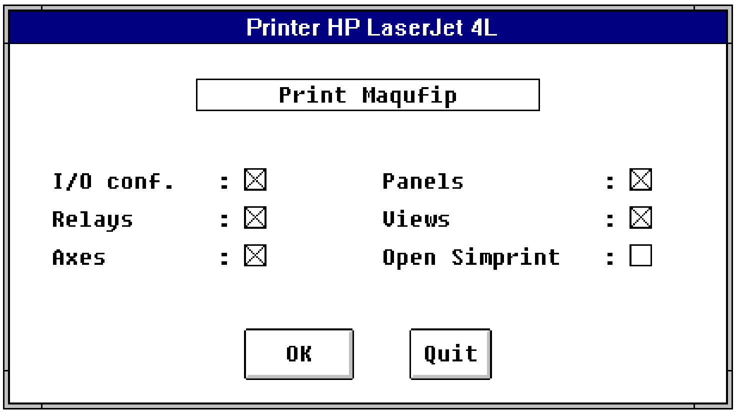
Axes Prints the axes, sensors, movements and all other description elements associated with the axes.

Panels Prints the operator panels.

View Prints the views.

Open SIMPRINT Opens the SIMPRINT software. If this box is not checked, the dossier will be printed using the default print format on the default printer without opening the SIMPRINT software.

The SIMPRINT application uses the “hard copy” of the screen to insert the drawings of the views and operator panels in the printed dossier. This operation can only be executed from SIMAC. When the command to print the selected machine has been activated, SIMACTEL starts by opening all the operator panel and view windows in sequence to store their contents in “*.DIB” files (Device Independent Bitmap). These files are used when printing via the SIMPRINT software.

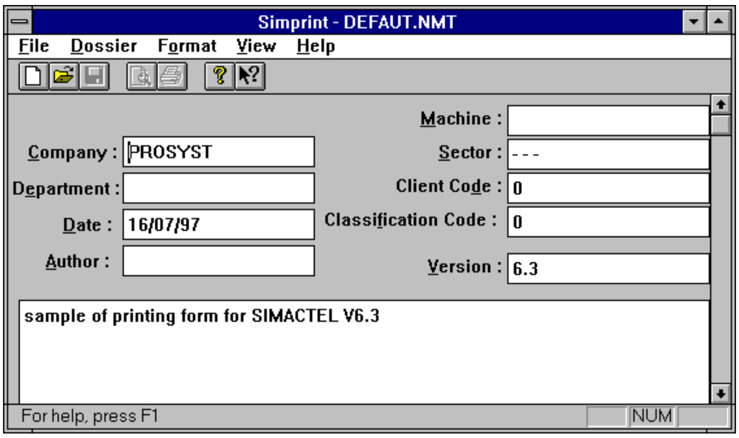


2.2.3 The fields in the main application window

The contents of the fields in the main application window are used to specify the type of dossier. They are printed on the first page, if they have been filled in, and saved in the print formats. Only the Machine field is filled in automatically by the SIMPRINT application when a SIMACTEL machine or model is opened. The fields used are as follows :

- **Company :** Name of the company which produced the original dossier.
- **Department :** Department within the company responsible for the dossier.
- **Date :** Machine dossier creation date.
- **Author :** Name of the author of the description.
- **Machine :** Read-only field. The name of the machine appears automatically.
- **Sector :** Name of the sector of the machine or industrial process.
- **Client code :** Client code number.
- **Classification code :** Archive number, information storage medium.
- **Version :** Number of the current version of the dossier.

At the bottom of the main application window is a comment zone in which the user can add additional information relating to the machine description. This zone is printed after the fields.



2.3 The File menu

The file menu contains all the commands relating to the management of the SIMPRINT print formats. It includes the following commands :

New	Creates a new print format.
Open...	Opens an existing print format file.
Save	Saves any modifications made to the active print format.
Save As...	Saves the active print format under a new name.
Print...	Prints the print dossier.
Print Preview	Displays the dossier on-screen as it will be printed.
Page Setup...	Selects a printer and its connection mode.
Last file opened	Opens one of the four most recently used print formats.
Exit	Quits the SIMPRINT application.

2.3.1 New command

This command creates a new SIMPRINT print format using the default options in the SIMPRINT application. The default character font is “Courier New”. If this font is not installed, SIMPRINT uses the system font on your computer.

The Open command is used to open an existing print format (see section 2.3.2 of this part).

Shortcuts : Toolbar :
Keys : CTRL+N

2.3.2 Open command

This command opens an existing print format from the Open dialog box. The name of the selected print format appears in the title bar of the application.

You can also create a new SIMPRINT print format using the **New** command.

Shortcuts : Toolbar :
Keys : CTRL+O

• Open dialog box

In the dialog box, the following options are used to select the print format to be opened :

File Name

Contains the name of the file to be opened. This list only displays the files with an extension which matches the selection (see below).

List Files of Type

Selects the type of file you wish to open. The SIMPRINT print formats generally have the extension "*.fmt".

D

Directories

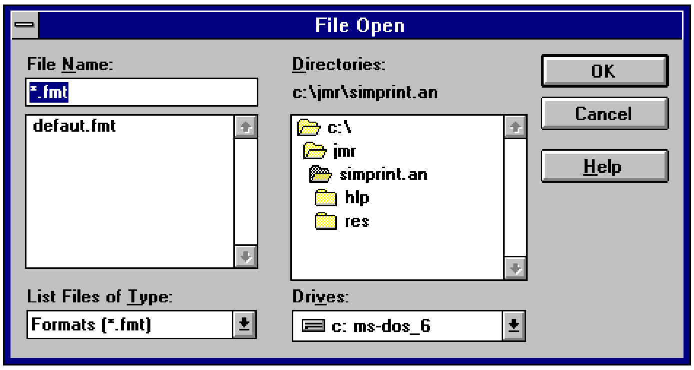
Selects the directory in which SIMPRINT has archived the print format to be opened.

Drives

Selects the drive in which SIMPRINT has archived the print format to be opened.

Network[2]

Connects SIMPRINT to another network node.



[2] This option only appears if the computer is connected to a local area network.

2.3.3 Save command

This command saves the active print format with its name and current location. When a print format is saved for the first time, SIMPRINT opens the Save As dialog box (see section 2.3.4 of this part) which is used to name the print format. This command is grayed out if no modification has been made to the active print format.

The Save As command changes the name and the location of a print format.

Shortcuts : Toolbar :

Keys : CTRL+S

2.3.4 Save As command

D

This command saves the active print format under a new name and/or to another location. It opens the **Save As** dialog box.

The Save command is used to save any modifications made to the active print format (see section 2.3.3 of this part).

• Save As dialog box

The Save As dialog box contains the following options :

File Name

Contains the new name of the archive file for the active print format. A file name includes a maximum of eight characters for the name and three characters for the extension. SIMPRINT automatically adds the extension specified in the list of formats to the file name.

Drives

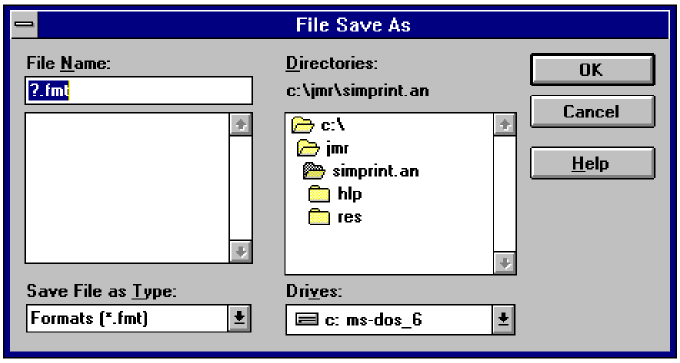
Selects the drive on which the print format is to be archived.

Directories

Selects the directory in which the print format is to be archived.

Network[3]

Connects SIMPRINT to another network node.



D

[3] This option only appears if the computer is connected to a local area network.

2.3.5 Print command

This command prints a SIMACTEL dossier from the Print dialog box. This dialog box is used to select the pages to be printed, the number of copies and the printer, and to access the other print options.

This command is grayed out when there is no SIMACTEL machine dossier or model available.

Shortcuts : Toolbar :
Keys : CTRL+P

• Print dialog box

The dialog box contains the following options :

Printer

Shows which printer is selected and the connection mode used. The Setup option modifies these parameters.

Setup

Opens the Print Setup dialog box which can be accessed directly using the Page Setup command (see section 2.3.7 of this part). This option is used to select a printer and its connection mode.

Print Range

Determines the pages to be printed :

All Prints the whole dossier.
Selection Option not compatible with the SIMPRINT application.
Pages Prints the pages between the number entered in the From field and that entered in the To field.

Copies

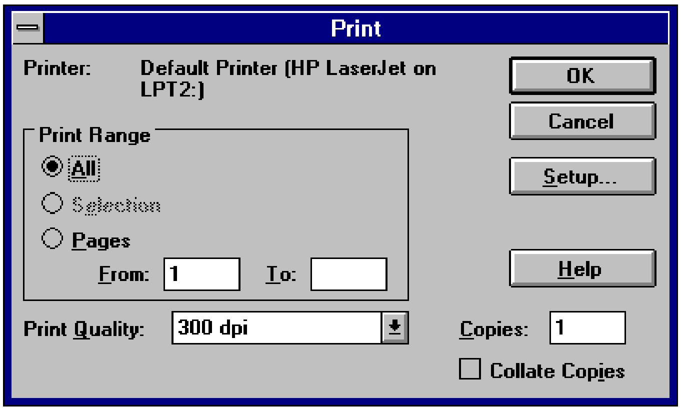
Shows the number of copies to be printed.

Collate Copies

Prints the copies in order instead of printing several copies of the same page at once.

Print Quality

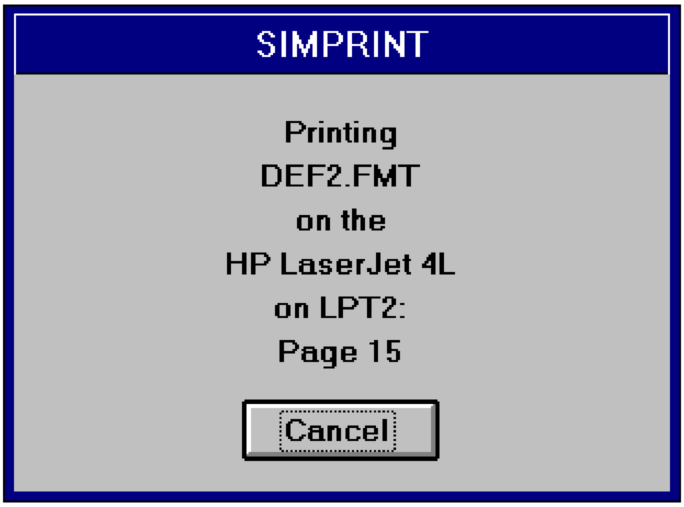
Selects the print quality. Generally, a poorer print quality requires less time.



• **Printing dialog box**

This dialog box is displayed when SIMPRINT sends the dossier to the printer. The page number indicates how printing is progressing.

The Cancel button is used to abort the print function before it finishes.



D

2.3.6 Print Preview command

This command displays the dossier on screen as it will be printed. The main SIMPRINT application window is replaced by a window in which one or two pages are displayed in the selected print format. The print preview toolbar is used to display one or two pages at once, to move up or down within the dossier displayed, to zoom in on a page and to start printing.

This command is grayed out when there is no SIMACTEL machine dossier or model loaded into SIMPRINT.

Shortcuts : Toolbar :

Keys : ALT+V

• Print preview toolbar

The print preview toolbar offers the following options :

Print

Opens the Print dialog box to start printing the dossier (see section 2.3.5 of this part).

Next Page

Displays the next page in the dossier.

Previous Page

Displays the previous page in the dossier.

One Page/Two Pages

Displays one or two pages at once.

Zoom +

Changes the display to give a more detailed view of the page displayed.

Zoom -

Changes the display to give a less detailed view of the page displayed.

Close

Closes the print preview and returns to the main SIMPRINT window.

2.3.7 Page Setup command

This command selects the printer and its connection mode. It opens a Print Setup dialog box.

• Print Setup dialog box

The following options are used to select a printer and a connection mode.

Printer

Selects the printer to be used. This is either the default printer, or another printer in the list. The Windows Control Panel is used to install new printers.

Orientation

Selects portrait or landscape.

Size

Selects the size of the paper on which the dossier will be printed.

Source

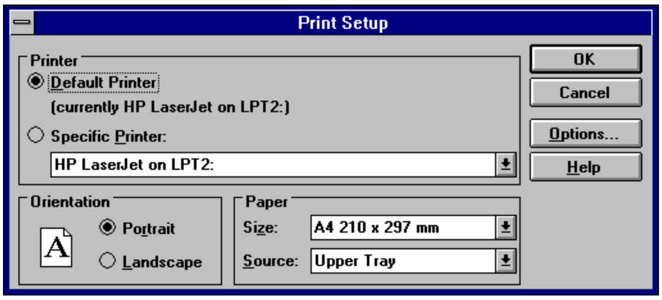
Some printers have several paper supply sources. This option selects the required source.

Options

Displays a dialog box used to modify other options. This dialog box is specific to the printer used.

Network[4]

Use this option to connect to another network node.



D

[4] This option only appears if the computer is connected to a local area network.

2.3.8 Last file opened

The numbers and names of the files listed at the bottom of the file menu are used to open any of the last four print formats most recently used.

2.3.9 Exit

This command quits SIMPRINT. It is equivalent to the Close command in the Application Control menu. SIMPRINT asks at this moment to save the modifications made to the active print format.

Shortcuts :	Mouse :	Double-click on the Application Control button
	Keys :	ALT+F4

2.4 The Dossier menu

The Dossier menu contains the commands for opening the SIMACTEL machine dossiers and models. These commands are :

• Open Machine	Opens a SIMACTEL machine dossier.
• Open Model	Opens a SIMACTEL model.

2.4.1 Open Machine command

This command opens a machine dossier from those listed in the Open Machine Dossier dialog box.

If a machine dossier is already open, a marker is positioned before the Open Machine command in the menu.

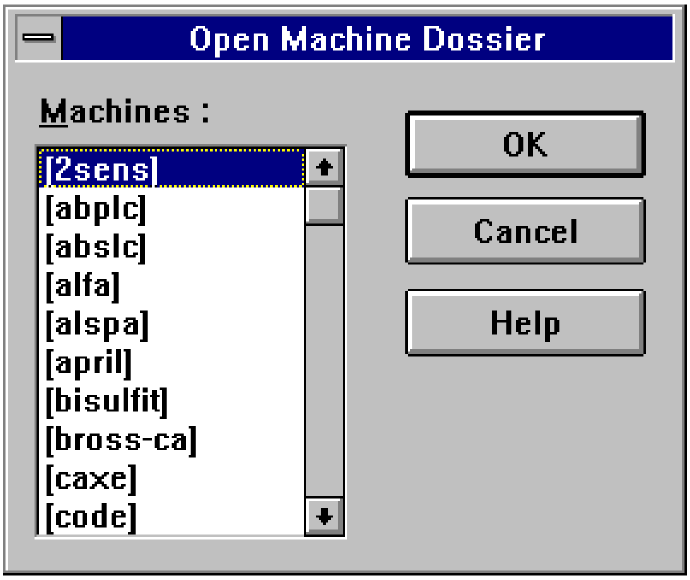
Note :

The name of the SIMACTEL machine loaded using this command is displayed in the machine field in the main application window. If the machine has not already been opened from SIMACTEL, the images representing the operator panels and views of the machine may not be up-to-date or may be missing.

This command is not available if a model is currently being edited.

• **Open Machine Dossier dialog box**

The list in the dialog box contains all the dossiers available on your computer. To open a dossier, select it from the list and click on the OK button or double-click directly on the element. The Cancel button is used to abort this command.



D

2.4.2 Open Model command

This command opens a SIMACTEL model from those listed in the Open Model dialog box.

If a SIMACTEL model is already open, a marker is positioned before the Open Model command in the menu.

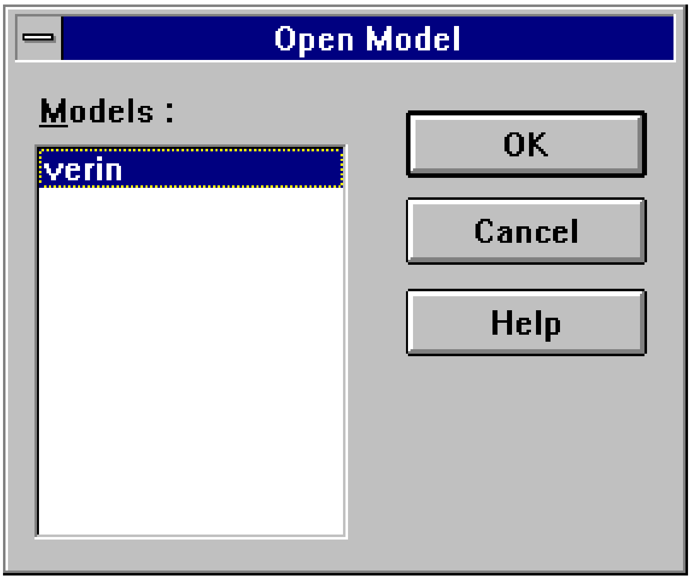
Note :

The name of the model loaded using this command is displayed in the machine field in the main application window. If the model has not already been opened from SIMACTEL, the images representing the operator panels and views of the model may not be up-to-date or may be missing.

This command is not available if a machine is currently being edited.

- Open Model dialog box

The list in the dialog box contains all the SIMACTEL models located in the same directory as the one which has been loaded. To open a model, select it from the list and click on the OK button, or double-click directly on the element. The Cancel button is used to abort this command.



2.5 The Format menu

The Format menu contains all the options for printing the print dossiers in the SIMPRINT application.

• General	Sets the margins, the header and the footer in the print format.
• Table of Contents	Configures the printing of the table of contents.
• I/O configuration	Configures the printing of the inputs and outputs.
• Relays	Configures the printing of the relays and power supplies.
• Axis	Configures the printing of the axes.
• Panel	Configures the printing of the elements in the operator panel.
• Paragraphs	Selects the style for each type of paragraph.

D

2.5.1 General command

This command accesses all the configuration parameters. It opens a General Configuration dialog box which is used to set the various options offered by the SIMPRINT application.

- **General Configuration dialog box**

The options in the dialog box are used to configure the print option available in SIMPRINT. You can display immediately the result obtained using the Print Preview command in the File menu (see section 2.3.6 of this part).

Margins

Each check box in this group contains the value of one of the margins in centimeters. The header will be printed above the top margin. The footer will be printed below the bottom margin. The left and right margins correspond to the free space required at each side of the pages in the dossier.

Elements

Each check box in this group corresponds to a part in the print dossier. If the box is checked, the corresponding part in the print dossier is printed.

To change the state of a box, simply click on it with the mouse. When you click on some of the boxes, SIMPRINT opens another dialog box which enables you to select more precise print options.

The options which can be accessed are as follows :

Instances	Prints the instances of the model.
I/O Config.	Prints the I/O configurations.
Relays/Supplies	Prints the relays and power supplies.
Axes	Prints the axes, sensors, movements and all the other description elements associated with the axes.

Op. Panels	Prints the operator panels.
Table of Contents	Prints the table of contents.
Border	Prints a border around each page in the dossier. The border is split into three zones : the header, a central zone where the dossier is printed, and the footer.
Header	Prints the header at the top of each page.
Footer	Prints the footer at the bottom of each page.
Views	Prints the views.

Header and Footer

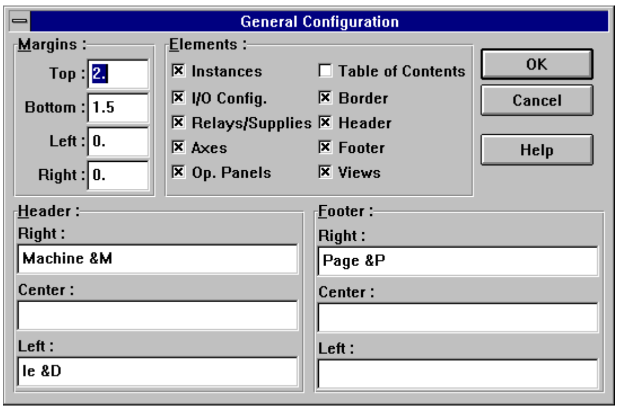
The header and footer of the SIMPRINT print dossiers are split into three zones (right, center and left). For each of these zones, you can type some text which will be printed with the dossier.

These text zones can also contain the following commands :

- **&P** This sequence of characters is replaced during printing by the page number.
- **&M** This sequence of characters is replaced during printing by the name of the machine or model.
- **&D** This sequence of characters is replaced by the current date.
- **&&** This sequence of characters is replaced by the character &.

Note :

The Table of Contents, I/O Configuration, Relays, Axis and Panel commands have an effect on the check boxes in the General Configuration screen. For example, if you select a panel configuration where no element in the operator panel should be printed, SIMPRINT will automatically deactivate the Op. Panel check box in the General Configuration dialog box. Conversely, if you deactivate the check box in the General Configuration dialog box, the Panel item in the menu will no longer be accessible (grayed out).



2.5.2 Table of Contents command

This command directly accesses the parameters relating to printing the table of contents. It opens the Table of Contents Configuration dialog box.

- **Table of Contents Configuration dialog box**

This dialog box contains the following options :

Level of Detail

This pull-down list is used to select the level of detail required for the table of contents.

Level 1 : Table of contents containing the list of Heading 1 styles.

Level 2 : Table of contents containing the list of Heading 1 and Heading 2 styles.

Level 3 : Table of contents containing the list of Heading 1, Heading 2 and Heading 3 styles.

D

Position

Selects the position of the table of contents in relation to the rest of the dossier.

Start of dossier Positions the table of contents at the start of the dossier.

End of dossier Positions the table of contents at the end of the dossier.

Right-Aligned Page Numbers

If this box is checked, the page numbers will be aligned to the right in the table of contents.

Leaders

Select the type of leader character which will be used between the headings and page numbers. This list is only active if the Right-Aligned Page Numbers box is checked.

None No leader characters are used in the table of contents.

Underscore Draws an unbroken line between the heading and page number.

Dotted Line Draws a dotted line between the heading and page number.

2.5.3 I/O Configuration command

This command directly accesses the parameters relating to printing the I/O configurations. It opens the I/O Configuration dialog box.

- **I/O Configuration dialog box**

This dialog box contains the following options :

I/O Cards

When this box is checked, SIMPRINT adds the list of I/O cards in the PLC I/O configurations to the print dossier.

Inputs

Select one of the three radio buttons in the group to define the required level of printing for the inputs.

No	No inputs are printed.
List	Only the names of the inputs along with their mnemonic and comment are printed.
Complete	Includes the input logic equation in the print dossier.

D

Outputs

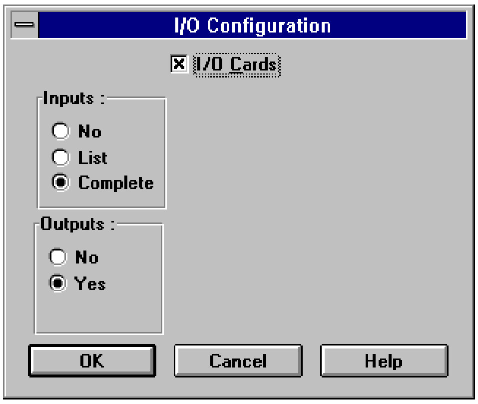
Select one of the two radio buttons in the group to define the required level of printing for the outputs.

No	No outputs in the print dossier
Yes	Adds the name of each output accompanied by its mnemonic and comment to the print dossier.

Note :

If all these options are deactivated (no printing of I/O cards, inputs or outputs), SIMPRINT automatically deactivates the **I/O Config.** check box in the **General Configuration** dialog box.

This function can only be accessed if the I/O Config. box is checked in the **General Configuration** dialog box (see section 2.5.1 of this part).



D

2.5.4 Relays command

This command provides direct access to the parameters relating to printing the relays. It opens the Relay Configuration dialog box.

- **Relay Configuration dialog box**

This dialog box configures the printing of the relays in SIMPRINT.

Supplies
Select one of the three radio buttons in the group to define the required level of printing for the power supplies.

Relays

Select one of the three radio buttons in the group to define the required level of printing for the relays.

Radio button options

No	No printing for this type of element.
List	Only prints the list of names of elements of this type.
Complete	Prints all the elements in the print dossier.

Note :

The model comments are printed next to the name of the elements (list mode and complete mode).
If all the options in the dialog box are deactivated (no printing of relays or power supplies), SIMPRINT automatically deactivates the Relays check box in the General Configuration dialog box.
This function can only be accessed if the Relays/Supp. box is checked in the General Configuration dialog box (see section 2.5.1 of this part).

D

2.5.5 Axis command

This command provides direct access to the parameters relating to printing the axes. It opens the Axis Configuration dialog box.

• **Axis Configuration dialog box**

This dialog box configures the printing for the axes and the other mechanical description elements in SIMACTEL. SIMPRINT offers several levels of printing for each of the following types of element :

Sensor
Selects the level of printing for the sensors. The sensors are printed after the axis to which they are attached.

Movement
Selects the level of printing for the Movements. The movements are printed after the axis to which they are attached.

Variable
Selects the level of printing for the variables.

Bistable
Selects the level of printing for the bistables.

Axis
Selects the level of printing for the axes.

Num. Var.

Selects the level of printing for the numeric variables.

Radio button options

No	No elements of this type are printed.
List	Add the list of names of elements of this type to the printed dossier.
Complete	Adds the full description of elements of this type to the printed dossier.

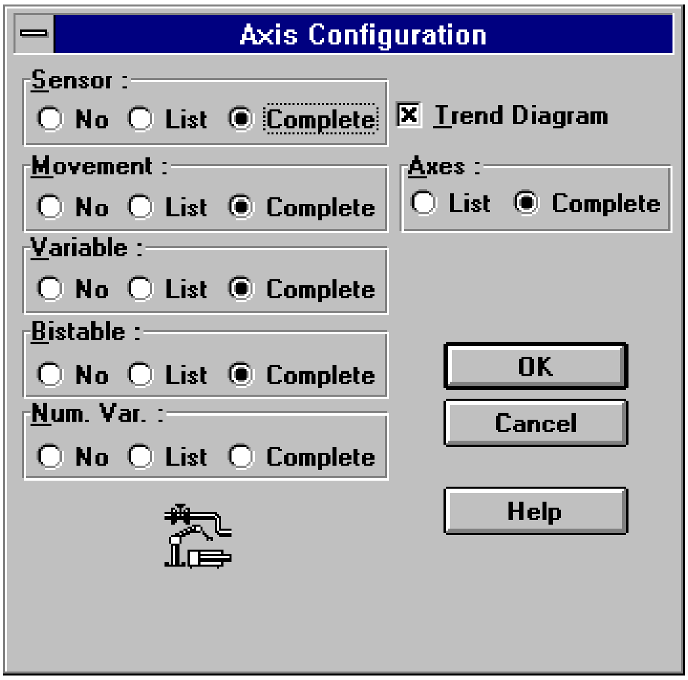
Trend Diagram

When this box is checked, SIMPRINT adds the trend diagram for the sensors of each axis to the print dossier.

Note

The model comments are printed next to the names of the elements.
This function can only be accessed if the Axes box is checked in the General Configuration dialog box (see section 2.5.1 of this part).

D



2.5.6 Panel command

This command provides direct access to the parameters relating to printing the operator panels. It opens the Panel Configuration dialog box.

• Panel Configuration dialog box

This dialog box is used to configure the printing of the operator panels in SIMAC.

Display of Panels
Selection of this check box adds the drawings of the operator panels to the print dossier.

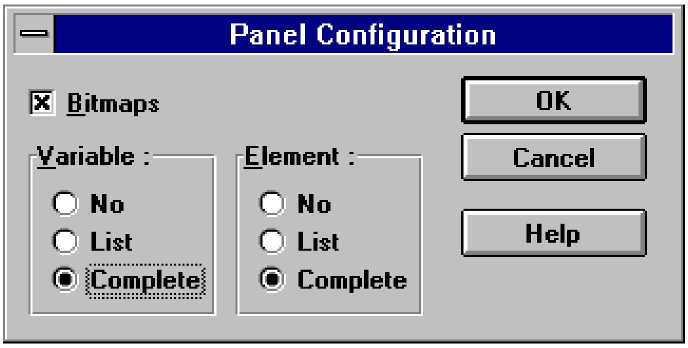
Variable
This group of radio buttons is used to select the level of printing for the operator panel variables.

Element
This group of radio buttons is used to select the level of printing for the operator panel elements.

Radio button options

No	The mechanical description elements are not printed.
List	Add the list of names of elements of this type to the printed dossier.
Complete	Add the full description of elements of this type to the printed dossier.

D



2.5.7 Paragraphs command

This command provides direct access to the parameters relating to printing the paragraphs. It opens the Paragraph Configuration dialog box.

• Paragraph Configuration dialog box

This dialog box changes the character font and the other options for each type of paragraph.

Pull-down list (Title, etc)

The elements in the pull-down list are the various paragraph styles used in SIMPRINT. Each style has corresponding lines of text in the print dossier. The styles defined in SIMPRINT are as follows :

Main Title	This style defines how to print the title of the print dossier (the first line of the dossier).
Identification	This style defines how to print the text in the fields on the first page. The fields printed are those which have been filled in in the main application window. Empty fields are ignored.
Heading 1 Heading 2 Heading 3	These styles define how to print the various titles contained in the dossier. Heading 1 relates to the various sections of the print dossier (I/O configurations, power supplies, relays, axes, variables, bistables, operator panel variables, operator panels and views). Headings 2 and 3 correspond to the subsections within these sections. The correspondence between text and styles depends on the print options selected by the user.
Element	Use this style to define how to print the names of the description elements (variable, relay, etc).
Trend Diagram	This style defines how to print the trend diagrams for the sensors and axes.
Normal Text	This style defines how to print normal text in the print dossier.

D

Header	This style defines how to print the text in the header on each page of the dossier. The only useful parameter in this style is the selected character font.
Footer	This style defines how to print the text in the footer on each page of the dossier. The only useful parameter in this style is the selected character font.
Panel & View	This style defines the space required around the drawings of the operator panels and views. Selection of the character font has no effect on this style.

Page Break

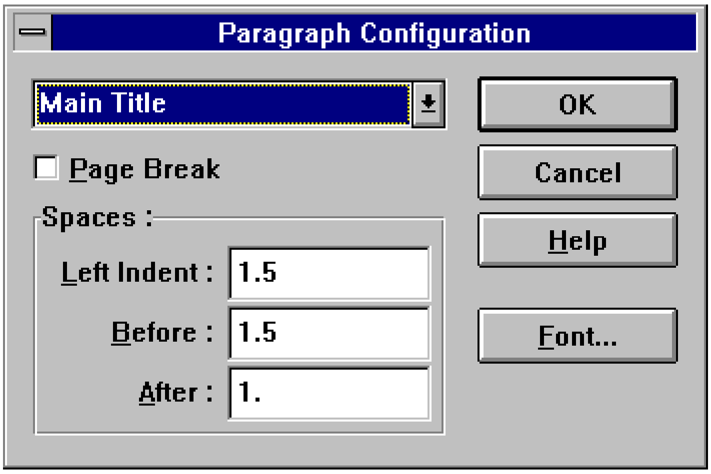
Inserts a page break before each paragraph using this style if this box is checked. This option has no effect on the Header and Footer styles.

Spacing

Left Indent	Type the size of the indent required in relation to the left margin for all paragraphs using this style. The size must be given in centimeters.
Before	Type the size of the space required above the first line of text in each paragraph using this style. The size must be given in centimeters.
After	Type the size of the space required below the last line of text in each paragraph using this style. The size must be given in centimeters.

Font

Opens the Windows dialog box for selecting a character font.



2.6 The Reveal menu

The Reveal menu contains the following commands :

- Toolbar — Shows or hides the toolbar
- Information bar — Shows or hides the information bar.

2.6.1 Toolbar command

This command shows or hides the toolbar which contains all the buttons for accessing the most commonly used SIMPRINT commands. A marker appears next to the element in the menu when the toolbar is displayed.

The toolbar is displayed across the whole width of the application, just below the menu bar, and provides quick access to the main SIMPRINT application commands.

Opens a new print format (see section 2.3.1 of this part).

Opens an existing print format. SIMPRINT opens the Open dialog box from which you can locate and open a print format (see section 2.3.2 of this part).

Saves the print format under its current name. If the print format is not already named, SIMPRINT opens the Save As dialog box (see section 2.3.3 of this part).

Displays a print preview of the dossier using the active print format (see section 2.3.6 of this part).

Prints the dossier using the active print format (see section 2.3.5 of this part).

Opens the About SIMPRINT dialog box (see section 2.6.3 of this part).

Changes the mouse pointer tool to the contextual Help pointer tool.

2.6.2 Information bar command

This command shows and hides the information bar which informs the user on the actions made on the interface SIMPRINT. A marker appears next to the command in the menu if the information bar is displayed.

The left part of the information bar explains, when you move through the menus using the arrow keys, the actions associated with the menu elements. In the same way, this zone displays messages relating to the toolbar buttons when you press these buttons without releasing them. If, after having read the message associated with a toolbar button, you do not wish to execute the command, release the mouse button outside of the button zone.

The right part informs the user of the status of the following keys :

CAP Caps Lock is active.

NUM Num Lock is active.

SCRL Scroll Lock is active.

2.7 The Help menu

The Help menu contains the commands for accessing the online Help function :

- Contents — Opens the SIMPRINT online Help at the page containing the index.
- Search for Help on... — Opens the general Windows Help on using the Help function.
- About SIMPRINT — Opens the About SIMPRINT dialog box.

2.7.1 Contents command

This command opens the initial SIMPRINT application Help screen. From this screen, it is possible to access all the online Help for the application.

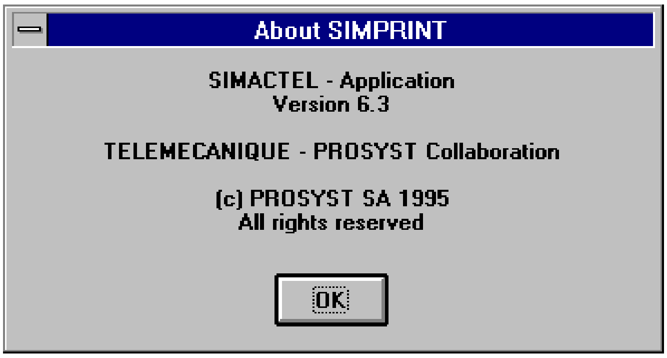
Wherever you are in the online Help, you can click on the Index button to return to this initial screen.

2.7.2 Search for Help on... command

This command opens the general Windows online Help function.

2.7.3 About SIMPRINT command

This command opens the About SIMPRINT dialog box containing the version number as well as the copyright ® of the SIMPRINT application.



D

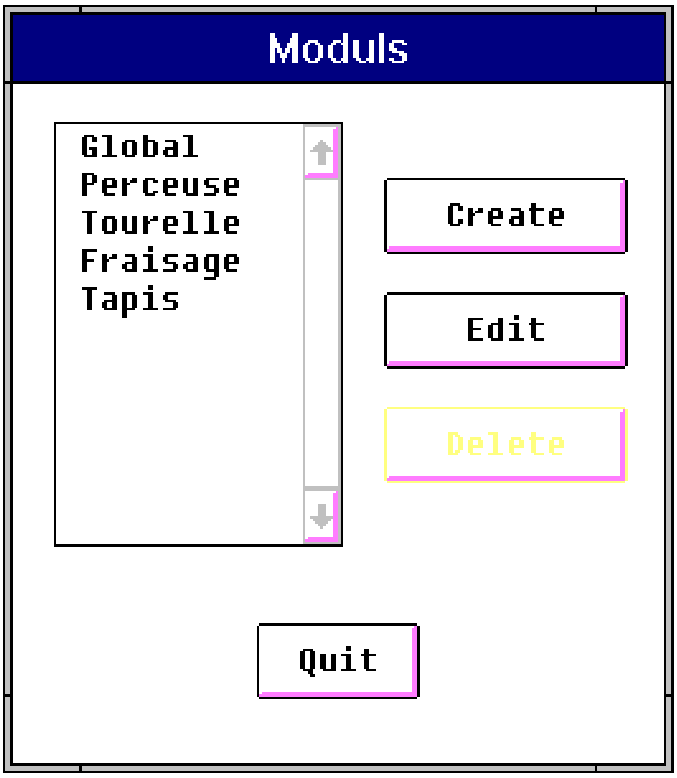
2.8 Printing problems and advice

1. If printing is slow : The print quality depends on several parameters. Depending on the character fonts used and the type of printer, the time required can vary within a range of one to ten. This is also true for the size of the file sent to the Print Manager. For example, PostScript printers send basic graphics to draw a line, while other printers need to send bitmaps. It is not possible to define simple rules. The user must configure the SIMPRINT print formats according to his hardware and requirements.

2. If the last character (the furthest character to the right) in the boxes on the right-hand side of the header and/or footer is not printed : This is due to the problem in Windows of calculating the size of a character string according to the font used. To resolve this problem, simply add a space after the last character in the string in question.

3.1 Configuration

Before executing a trace on an installation, the first operation is to configure the modules in order to structure the representation of the machine. To do this, select the Configure item from the Trace menu in the main window in the SIMACTEL simulation environment.
An automated machine is often made up of mechanical subassemblies which may be relatively independent of the animation program. The “modules” function is used to give the simulator this type of structure. A module is a set of axes. It is said to be active as soon as a movement is made on one of the axes which make up the module. It is said to be inactive if there is no movement.
The module configuration has some similarities with that of the pages of variables (see section 2.3, part B). The Create button is used to construct a module; after having entered the name of the module, the user moves to the edit screen.



3.1.1 Editing a module

The window for editing modules comprises, on the left-hand side, a scroll-down list in which all the axes of the simulated machine are displayed. On the right-hand side of the screen are the axes which comprise the module being edited.

If an axis is selected from one of these two lists, its label appears in the top center of the window and the buttons become active.

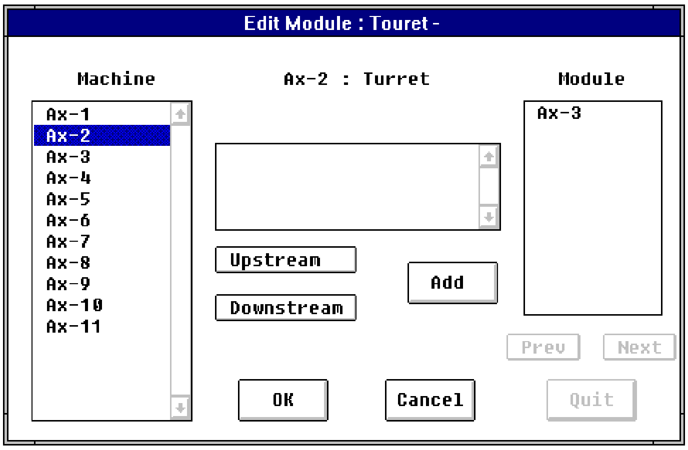
The Downstream button is used to display in the center of the window all the variables which are located “downstream” of the axis, that is those on which any evolutions to the axis are likely to have an effect.

This involves the axis sensors, the combinations in which these sensors are used, and any description variables or relays and inputs concerned with the axis.

Similarly, the Upstream button is used to display the variables located “upstream” of the axis, that is those used in the evolutions to the axis. This involves movements, relays, bistables and outputs controlling the axis.

These variables are obtained from the existing chart of the description variables, excluding the relays which are part of the general power supply.

The Add button is also active if an axis is selected from the list on the left-hand side (all the machine axes) : this button is used to add the axis to the module. If a module axis is selected (right-hand side of the window), the machine axis will be inserted before this axis in the module, otherwise it is simply added at the end of the module. However, if it is an axis belonging to the module which is selected, the button is used to delete it from the module.



D

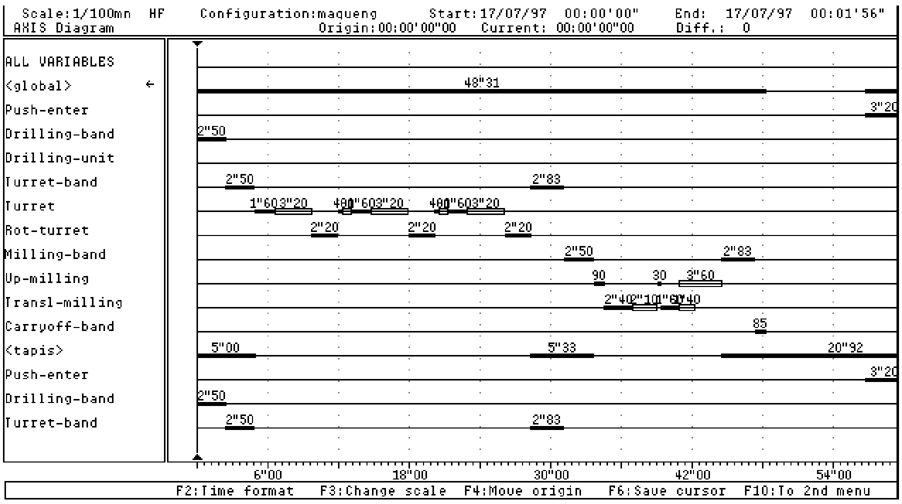
3.2 Presentation elements - General functions

3.2.1 General presentation

This section explains the basic principles required for using the trend diagram representation software. It describes the following :

- the elements in the representation window,
- how to order the representation.
- how to print all or part of a trend diagram,
- how to create and use the reference cycles.

The first diagram presented to the user when he views a record, is called the Axis Diagram.



This diagram offers an overall view of the operation giving the changes in state of the modules (active, inactive), and, for each of them, the activity of the associated axes.

The first line, All Variables, is used to access a diagram of the evolutions of all the I/O and machine movements.

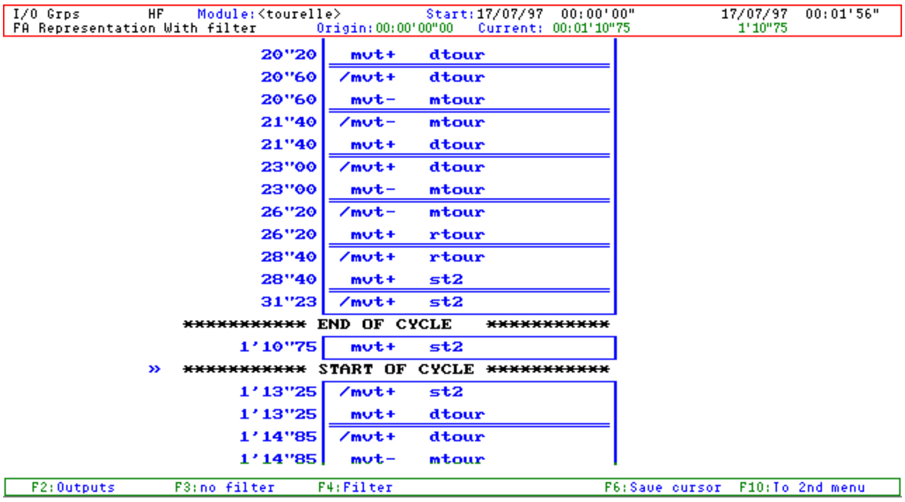
If no module has been created before the record, the diagram simply shows the activity of the axes.

Moving within this trend diagram the user can select a specific instant in the record (horizontal cursor keys) and a module or axis (vertical cursor keys).

If a module is selected, pressing the “Enter” key displays the diagram of the module movements.

If an axis is selected, pressing the “Enter” key displays the diagram of the axis I/O and a summary in the first line of the activity of the axis itself.

Function keys can also be used to switch to a functional representation, known as an FA representation, from the movement diagrams and the I/O diagrams (see below).



D

In addition to changing the time base, various functions are used to :

- calculate the time difference between two events,
- organize the sensors/actuators used according to numerical, chronological or structural criteria,
- modify the reference point of the diagrams shown,
- print diagrams.

3.2.2 General principle

The general principle of the man-machine interface function and the functions which can be accessed at each level are summarized in the table below :

Type of diagram	Fx function keys	Shift + Fx function keys	Alt + Fx function keys
Axis diagram	F1: Information F2: Time/Mechanical Format F3: Zoom F4: Change ref. point F6: Memorize cursor F8: Print screen F9: Printout Tab: Menu window	None	None
Movement diagram I/O diagram	F2: I/O or Outputs F3: Change scale F4: Num/Chronological order F6: Memorize cursor F7: I/O label F8: Print screen F9: Printout Tab: Menu window	Shift+F1: FA Representation Shift+F2: Change time format Shift+F3: Input color Shift+F4: Change ref. point Shift+F5: Zoom 1 min Shift+F8: Cont/evolution mvmt	Alt+F1: Save reference Alt+F2: Comparison active Alt+F3: Ref. cycle duration Alt+F4: Activate ref. movement Alt+F5: Activate resynchro Alt+F6: Find deviation Alt+F7: Change reference ref. point Alt+F8: Reference cycle info
FA Representation	F2: Sensor/Act. or Actuator F3: With or without filtering F4: Filter/add selected I/O F6: Memorize cursor F8: Print screen F9: Printout Tab: Menu window	Shift+F1: Representation Shift+F2: Change time format Shift+F3: Input color Shift+F4: Change ref. point Shift+F5: Zoom 1 min Shift+F8: Cont/evolution mvmt	Alt+F1: Save reference Alt+F2: Comparison active Alt+F3: Ref. cycle duration Alt+F4: Activate ref. movement Alt+F5: Activate resynchro Alt+F6: Find deviation Alt+F7: Change reference ref. point Alt+F8: Reference cycle info

Pressing the “Tab” key displays a menu which contains all the functions available for the current representation.

Each function can thus be accessed either via a function key (“F2” for example), a combination of keystrokes (“Alt+F2” for example), or the menu described below :
This menu is made up of 2 parts :
• the first part contains the title of the function keys,
• the second contains information relating to the pages in the menu. In our example :
• x indicates that pressing the LEFT arrow key will display the previous page in the menu,
• v indicates that pressing the RIGHT arrow key will display the next page in the menu,

D

- The menu windows are defined in a pull-down list as shown in the illustration below (once the last window has been displayed, the user is returned to the first). The number of the window in a menu differs depending on the type of representation.

Once this menu is displayed on screen, it is possible to :

Select a function :

by selecting a function key (eg : by pressing “F3” to change the scale),

by placing the cursor (represented by an arrow to the right of the menu label) before the chosen function using the up or down arrow keys (! à) then confirming the operation with “Enter”.(ø)

The menu will close and the selected action will be executed. The next time the “Tab” key is pressed, the menu will reappear, with the cursor next to the last function selected.

View another page in the menu by :

pressing the left and right arrow keys (x v).

View a particular page in the menu :

“Shift” : to display the “Shift+Fx” type functions,

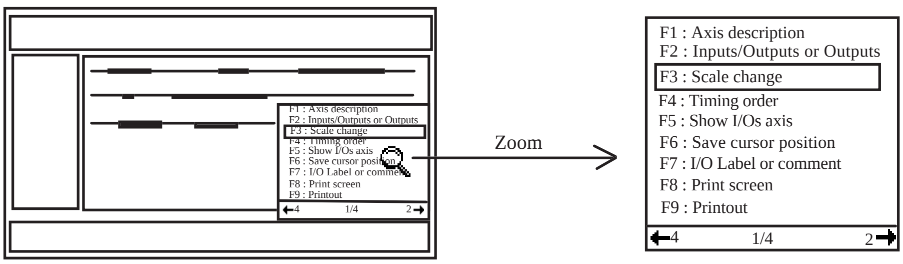
“Alt” : to display the “Alt+Fx” type functions,

“END” : to display the arrow, “Ctrl”, “Esc”, etc, keys.

Clear the menu window by :

pressing “Esc” or “Tab” (no action is performed),

selecting a function as explained above.



3.2.3 Page header

The top of the screen page contains the following elements :

- indication of the current time scale,
- the name of the machine in the axis diagram, the label of the module in the module movement diagram, the name of the axis in the axis I/O diagram,
- the record start and end dates and times,
- the type of diagram shown,
- indication of the current duration format,
- TF : Time Format (hh : mn’ ss» xx),
- MF : Mechanical Format (hh: mm’ yyy),
- indications of the reference point and the current position of the cursor.

By default, the reference point corresponds to the record start time. The deviation is the difference between the current cursor position and a previously memorized position (by default, this position corresponds to the reference point).

D

3.2.4 Moving around in the trend diagram

• The cursor

The cursor is represented by a vertical bar which can be moved around on the page using the left and right arrow keys in fifths of a time unit in the axis diagram. In the movement and I/O diagrams, pressing a left or right arrow key moves the cursor between evolutions (this includes all the evolutions to all the I/O and movements).

With each movement, information relating to the deviation and the current position in the page header are updated.

Keys :

“Shift+F8” : used to switch from a continuous mode (in fifths of the time unit) to an evolution mode (move from evolution to evolution).
“Ctrl+Á” or “Ctrl+Ë” : quicker movement (10 units).
“Ctrl+Home” or “Ctrl+End” : moves the cursor to the start or end of the trend diagram.

• Page break

A page represents 100 time units in a record (6 s, 1 min, 10 min, 100 min depending on the scale). When the cursor is moved to 5 time units from the end of a page, a page break is generated which shifts the window shown on screen by 50 time units.

3.2.5 Precision

The precision of the diagram is directly linked to the modelling of the application. The times displayed in the trend diagrams and the FA representations are calculated based on the distances and speeds declared in SIMACTEL.
3 time scales are available :

• in the axis diagram :

1 min or 60 s,
1/10 min or 6 s,
1/100 min or 600 ms.

• in the movement diagram and the I/O diagram :
1 min or 60 s,
1/10 min or 6 s,
1/100 min or 600 ms,
1/1000 min or 60 ms.

The graph is positioned on screen with a precision of 1/5 of a time unit or 12 ms maximum.

D

Keys :
“F3” changes the time scale.
“Shift+F5” changes to the 1 min time scale in the movement and I/O diagrams.

3.2.6 Indications of time

• The duration

The principle for constructing a diagram consists of tracing the horizontal segments representing the activation periods for each input, output or movement, or periods during which an active or inactive state of a module is maintained.

For each segment, the duration is indicated in the current format. By default, the time format is used and only the significant figures are shown. Two types of format are possible :
- Time Format (TF) hh : mn’ ss» xx, (example: 2’7"38),
- Mechanical Format (MF) hh: mm’ yyy, (example: 2’138).

The indication > followed by a duration above a segment (for example : > 2 ”38) signifies that the corresponding element was already in the state indicated before the reference point or stays in this state after the end of the record.

The indication X above a segment (example : X) indicates that the activation is less than or equal to :

- 10 ms in time format,
- 1/100 th of a minute in mechanical format

The precision of the duration is always the same regardless of which time scale is used.

Keys :
“F2” : changes the date format in the modules diagram,
“Shift+F2” : changes the date format in the movement and I/O diagrams.

• Start and end of activation

When the cursor is placed at the start or end of the segment, the indication of the difference in the header corresponds to the difference between the instant when the event concerned occurred and the reference point.

The position indication corresponds to exact time at which the event occurred.

The reference point in the axis diagram represents the record start time by default. It can be modified via the “change reference point” function :

Keys :
“F4” changes the reference point of a record in the axis diagram.
“Shift+F4” changes the reference point of a record in the movement and I/O diagrams.

• Time difference between two events

The “F6” key is used to memorize the current cursor position. Pressing this key places a triangular marker on the time scale at the current cursor position. If the cursor is moved, this automatically calculates the difference between the memorized position and the current position.

For example :

Ref. point : 15:37’53"32 Current : 15:47’26"11 Difference : 3"10/15:47’23"1

Key :
“F6” memorizes the position of the cursor.

• Vertical movement

The window shown on screen can be moved vertically by one line using the up and down arrow keys.

Keys :
“PgDn” and “PgUp” are used to move the window from module to module in the axis diagram or by 10 lines in the other diagrams.
“Home” and “End” are used to position the cursor at the first or last line.

D

3.3 Axis diagram

3.3.1 Types of element shown

A screen page contains 16 lines representing the activity of the modules and axes over one time period. The activity of an axis is represented by a thick line (forwards) or a triple line (backwards) with a time representing the period above it.

• The “all variables” module

The first module named “all variables” is fictitious. It contains all the I/O declared in SIMACTEL. It is used to display all the I/O in the same diagram.

D

3.3.2 Changing the reference point

If a record contains more than 3000 changes in state of a module, the diagram represents the first 3000 from the reference point. This constitutes a sector.

Changing the reference point consists of moving the reference point of the diagram to the current cursor position, which moves the sector displayed in the overall record.

Note

> Changing the reference point when the cursor is placed on the reference point repositions the reference point at the start of the record. The I/O diagram is limited to the first 3000 changes in state of the module I/O selected from the instant corresponding to the current cursor position. To display the remainder of the record, simply move to the end of the sector, then change the reference point.

Keys :

“F4” changes the reference point in the axis diagram.

3.3.3 Selecting a module

Selecting a module displays the movements corresponding to the selected module, shown axis by axis.

The position of the cursor in the axis diagram will set the cursor reference point in the module movement diagram.

Keys :
“!”, “à” moves up or down,
“Home” and “End” moves to the first or last module,
“Enter” selects the module.

3.3.4 Selecting an axis

Selecting an axis displays the I/O and the movements of the axis. The axis movements are placed at the end of the list. The first line summarizes the activity of the axis movements.

The position of the cursor in the axis diagram will set the cursor reference point in the axis I/O diagram.

Keys :
“!”, “à” moves up or down,
“Home” and “End” moves to the first or last module,
“Enter” selects the axis.

3.3.5 Exiting the axis diagram

Pressing the “Esc” key exits the axis diagram. This action can also be used to exit the representation software.

3.4 Module movement diagram

Selecting a module and an instant in a record in the axis diagram displays the module movement diagram.

The reference point of this diagram corresponds to the current cursor position in the axis diagram at the moment of confirmation. Any cursor movement will be reflected in the axis diagram.

3.4.1 Types of element shown

• Axis status line

The top of the trend diagram shows the module status line as it appears in the axis diagram. The user can then make a direct correspondence between the activity of the module and the activity of the movements.

D

3.4.2 Selecting the elements shown

The order in which the elements are shown on a trend diagram can be selected by the user. The types of element shown are always given in the page header.

The sort in chronological order can be completed by comparing the time at which the first edges of each element appear starting from the memorized position ("F6" key). Elements which have never been active or which are already active at the corresponding instant are shown at the end of the list.

Key :
"F4" is used to switch from numerical order to chronological order and vice versa.

3.4.3 Exiting the movement diagram

Pressing the "Esc" key exits the movement diagram. This action can also be used to return to the axis diagram.

3.5 Axis I/O diagram

Selecting an axis and an instant in a record in the axis diagram displays the I/O diagram for the installation.

The reference point of this diagram corresponds to the current cursor position in the axis diagram at the moment of confirmation. Any cursor movement will be reflected in the axis diagram.

3.5.1 Types of element shown

• Selecting Inputs or I/O

By default, the elements shown on screen are the I/O then the movements and the axes to be monitored for the module. The I/O are sorted in numerical order of their PLC denomination.

Key :

“F2” is used to switch between the diagram of the outputs and that of the I/O and vice versa.

• Differentiating the colors of the input and output evolutions

By default, the I/O evolutions are shown in yellow. However, to simplify reading of the trend diagram, it is possible to evolution the colors of the input evolutions. Three colors are available : yellow, blue, white.

Key :

“Shift+F3” is used to change the colors of the input evolutions by switching from yellow to blue, blue to white then white to yellow respectively.

• Axis status line

The top of the trend diagram shows the axis status line as it appears in the axis diagram. The user can then make a direct correspondence between the activity of the axis movements and the evolutions of the I/O.

3.5.2 Selecting the elements shown

The order in which the elements are shown on a trend diagram can be selected by the user. The types of element shown are always given in the page header.

The sort in chronological order can be completed by comparing the times at which the first edges of each element appear starting from the memorized position (“F6” key). Elements which have never been active or which are already active at the corresponding instant are shown at the end of the list.

Key :

“F4” is used to switch from numerical order to chronological order and vice versa.

3.5.3 Exiting the I/O diagram

D

Pressing the “Esc” key exits the I/O diagram. This action can also be used to return to the axis diagram.

3.6 Representation in a functional analysis table

3.6.1 Access

The structured table type representation is accessed by pressing “Shift+F1” in the I/O diagram or the movement diagram.

3.6.2 Presentation

This representation displays the same information as the trend diagram in the form of structured tables. It is used to follow the evolutions of the I/O chronologically and to label synchronous evolutions instantaneously (see next illustration).

3.6.3 Functions

The current cursor position corresponds to the time at which the first evolution is displayed on screen (marked by an arrow) and can be memorized by pressing “F6”. All the changes made to the FA representation remain valid in the trend diagram.

Note that most of the keys which can be used to represent the movements or I/O can also be used in the FA representation.

Keys :
“x ” or “ v” moves the cursor evolution by evolution,
“PgUp”, “PgDn” moves through the list of evolutions page by page.

3.6.4 Filtering

If the evolutions of one or more I/O “disrupt” the analysis or are not helpful in understanding the sequence of evolutions, the user can delete then temporarily from the representation. This operation is known as filtering.

To execute a filtering operation, simply switch to “FA representation with filtering” mode (indicated in the header) by pressing “F3”.

In “FA Representation” mode, if filtering is not active, any I/O which have already been filtered are marked with an *. In this case, pressing the “F4” key redisplays the evolutions of the selected I/O in the representation with filtering.

Filtering only operates with the FA representation. However, it is possible to use this filtering function in the I/O diagram.

Keys :
“F3” : Activates or deactivates the representation with filtering,
“F4” : Used to filter from the representation the I/O selected by the arrow, or to restore I/O which have already been filtered (marked by *).

3.6.5 Exiting the FA representation

Pressing the “Esc” key exits the FA representation. This action can also be used to return to “standard” mode in the movement or I/O diagrams.

3.7 Comparing cycles

3.7.1 Principle

This function is used to show two trend diagrams on screen saved at different times. The simultaneous representation of these two records is used to display the deviations in the operation of an installation at a given moment, in relation to the reference operation.

A reference trend diagram is memorized in the trend diagram by selecting all or part of this trend diagram and saving it.

The comparison can then be activated at any time. The reference trend diagram is placed next to the trend diagram already on screen. It is possible to move one trend diagram in relation to another to match up the changes in state of the I/O.

During the comparison of the trend diagrams, all functions are still available : changing the time base, chronological sort, selecting Inputs or I/O, etc.

This comparison of cycles can be used in “I/O diagram”, “movement diagram” or “FA representation” mode.

D

3.7.2 Creating a reference cycle

The reference cycle is defined by the memorized position (accessible via “F6”), and by the current cursor position.

If the two are mixed together, the whole record is saved (1500 evolutions maximum). A reference cycle can either be created in the I/O trend diagram, or in the structured table representation.

The following table summarizes the steps for creating a reference cycle :

Action	Keys
Place the cursor in the position at which you wish to start the reference cycle. For example, at the “START CYCLE” instant	“x”,“v”,“Ctrl+x” “Ctrl+v” in I/O representation mode, “!”, “à”, “PgUp” or “PgDn” in FA mode,
Memorize this position	“F6” (memo cursor)
Place the cursor in the position at which you wish to end the reference record. For example, at the “END CYCLE” instant	“x”,“v”,“Ctrl+x” “Ctrl+v” in I/O representation mode, “!”, “à”, “PgUp” or “PgDn” in FA mode,
Save the reference record for the 2 positions indicated	“Alt+F1”
Confirm the reference cycle record	“Y” to confirm “N” to cancel creation.
The reference cycle will then be resynchronized automatically to the same start cycle conditions in the record.	Automatic

D

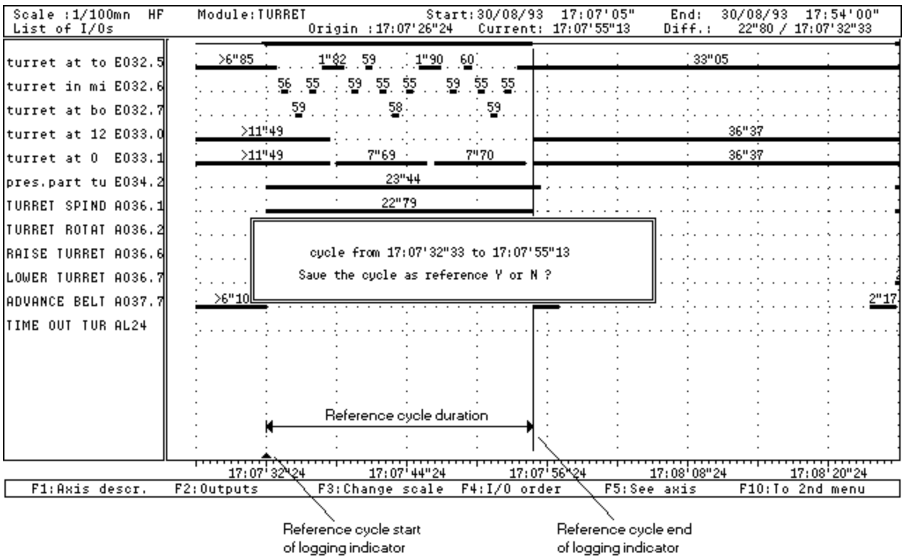
Once the operation has been confirmed, the reference cycle is automatically saved. It can be recreated at any time.

A reference cycle record includes :
- the evolutions of all the I/O (within the limit of 1500 evolutions),
- data relating to the instant at which the record starts,
- the first change in state of the I/O (from all the axis I/O),
- the logic state of the I/O shown on screen when this changes state.

This information can be displayed by pressing “Alt+F7”.
It is possible to create as many reference cycles as there are elements in the axis diagram. A single reference cycle can be created for each element.

Keys affecting the reference cycles :

- “Alt+F2” deactivates or activates the reference cycle display (if it exists)
- “Alt+F3” displays the durations for the reference cycle,
- “Alt+F4” moves the reference cycle over time,
- “Alt+F5” performs an automatic search for deviations,
- “Alt+F6” changes the reference point of the reference cycle over time,
- “Alt+F7” displays the synchronization information for the reference cycle,
- “+” (plus) resynchronizes the reference cycle to the start of the record,
- “-” (minus) resynchronizes the reference cycle to the end of the record.



3.7.3 Activating the comparison

The reference trend diagram is displayed in blue with a different marker to distinguish it in trend diagram mode, or on the right-hand side in structured table mode.

In comparison mode, it is always possible to modify the time base (key “F3”), display the outputs only or the I/O (key “F2”), etc.

Key :

“Alt+F2” superposes the reference and current cycles.

3.7.4 Displaying the durations for the evolutions of the reference cycle

By default, the durations for the evolutions displayed are those of the current cycle. However, it is possible to display the durations for the evolutions of the reference cycle in trend diagram mode.

Keys :

“Shift+F2” displays the durations for the evolutions of the reference cycle in time or mechanical format,

“Alt+F3” alternates between indications of the duration for the current record with those of the reference record.

3.7.5 Displaying time deviations in structured table mode

The time indicated for each evolution of the current record (that on the left-hand side) corresponds to the deviation calculated between the current time and the reference time. Consider the following example :

- input I02,5 : active after 4’10’’45 on the current record (previous illustration) and after 4’10’’35 on the reference record (next and previous illustration) will display a positive time deviation of 10 hundredths of a second (meaning that the record at the time of this evolution, has a delay of 10 hundredths of a second on the reference cycle).
- input I02,3 : active after 4’13’’05 on the current record (previous illustration) and after 4’13’’10 on the reference record (next and previous illustration) will display a negative time deviation of (-5) hundredths of a second (meaning that the record at the time of this evolution, is 5 hundredths of a second ahead of the reference cycle).

Key :

“Alt+F3” displays the time deviations.

3.7.6 Moving the reference record

To simplify comparison, it is possible to move the reference record in the diagram. This can be done in three different ways :

- each time the cursor is moved, simply press “Alt+F4” and use the “ x”, “v” or “Ë” keys in trend diagram mode, “!” or “à” in structured table mode. The reference record then moves in the diagram with the cursor.
- between cycles, press the “+” (plus) or “-” (minus) keys to find the next or previous cycle from the cursor position. If the start of the reference record corresponds to the start of a cycle, the reference record is then moved from cycle to cycle in the trend diagram.
- by shifting the reference record to the current cursor position by pressing “Alt+F6”.

Moving the reference cycle from cycle to cycle using the “+” or “-” keys consists of finding in the current record the same conditions as the instant of the start of the reference record. This is the change in state corresponding to the first evolution with the same I/O logic states displayed during the save.

Any evolution of the current record which does not correspond to an evolution of the reference cycle is considered a deviation.

The functional deviations are displayed in red and marked with the symbol “#”.

3.7.7 Displaying reference cycle synchronization information

The following information is displayed :

- the date and time of the record,
- data relating to the instant of the start of the record,
- the first change in state of the I/O (from all the module I/O),
- the logic state of the I/O shown on screen at the instant of this change in state.

D

Key :
“Shift+F7” displays the reference cycle information.

3.7.8 Finding a deviation

A search for a deviation can be executed automatically in order to compare the full record with the reference cycle previously created. The search starts from the current cursor position and finishes at the end of the record. Each time a deviation is found, the search stops and displays the type of deviation found.

Keys :
“Shift+F5” launches the automatic search for deviations,
“Esc” ends the current automatic search for deviations.

- Principle of finding a deviation

Each time a cycle is resynchronized, the list of evolutions of the current cycle is compared to the list of evolutions of the reference cycle.

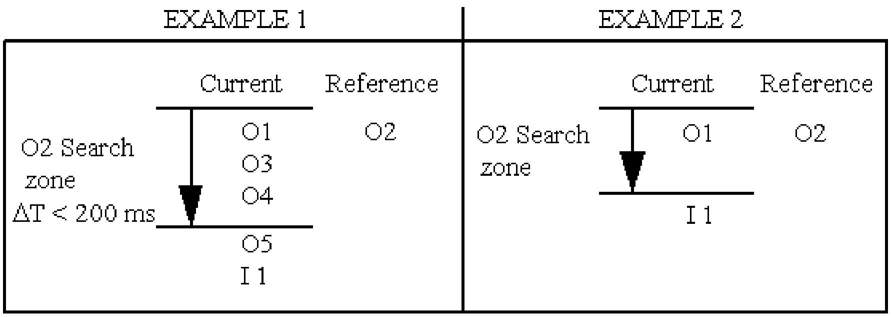
In FA representation, where filtering is also taken into account, the deviation displayed may differ from that in the trend diagram.

• 1st step

Starting from the first change in state of the cycle, the current cycle and the reference cycle are compared evolution by evolution. The first different change in state in the reference cycle is searched for in the current cycle among the following evolutions.

On a change in state of an output

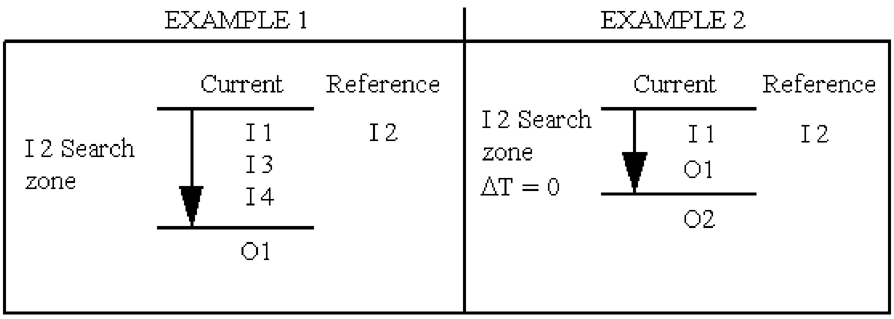
The search stops at the first change in state of an input OR at the first evolution of an output separated by more than 200 ms.



Comparison of cycle evolution by evolution for an output.

On an input

The search stops at the first change in state of an input or at the first different time.



Comparison of cycle evolution by evolution for an input.

• 2nd step

If the search for an evolution finishes, it is considered that there is no deviation. The user is then interested in the corresponding change in state in the current cycle which can be searched for in the same way in the reference cycle. Similarly, if an evolution is found, there is no deviation and it goes on to the next different evolution.
For a group of deviations, the find zone starts at the first different evolution.

• Displaying deviations

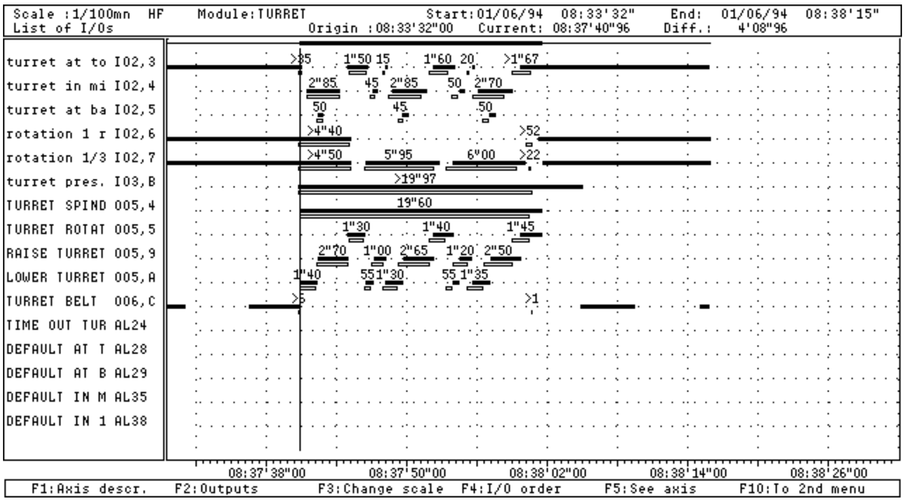
A deviation found in this way is signaled by a message at the bottom of the screen.

“ Deviation : I02,4 - CENTER TOOL POST ==> edge · not expected ”

or

“ Deviation : I02,4 - CENTER TOOL POST ==> edge Í missing ”

This concerns either an unexpected edge in the current cycle, or an edge in the reference cycle which is missing in the current cycle. The deviation is marked in the trend diagram by “DEV” in red or “dev” in blue. In FA representation mode, the line corresponding to the deviation is shown in purple.



3.8 Printing diagrams

A trend diagram can be printed at any time, whether it is the axis diagram, the movement diagram, the I/O diagram or the FA representation.

2 types of printer can be used : EPSON or compatible printers, and HP LaserJet printers.

The start of the print zone corresponds to the reference point of the record, and the end of the print zone corresponds to the current cursor position.

Printing is from the first element displayed in the list. 32 elements can be printed on an A4 sheet.

To print part or all of a trend diagram :

Action	Keys
Place the cursor in the position at which printing should start	"x","v","Ctrl+x", "Ctrl+v" in axis diagram, movement diagram or or I/O diagram mode. "!", "à", "PgUp" or "PgDn" in FA mode,
Change the reference point to mark the start of printing	"F4" in the axis diagram, "Shift+F4" in movement diagram, I/O diagram or in FA representation,
To print part of a diagram : Place the cursor in the position at which printing should finish To print all of a diagram : do not move the cursor (leave it at the reference point of the record)	"x","v","Ctrl+x", "Ctrl+v" in movement, I/O, or axis diagram mode. "!", "à", "PgUp" or "PgDn" in FA mode,
Start printing	"F9"
Select the type of printer	"1": EPSON printer "2" : HP LaserJet printer "Esc": cancels printing

Printing can be stopped at any point by pressing "Esc".
The choice of printer is only required for the first print.

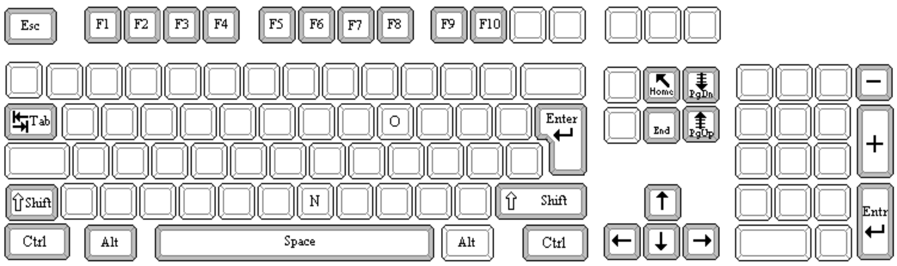
Keys :
"F8" prints the screen,
"F9" prints all or part of the trend diagram.

D

3.9 Summary of the keys used

3.9.1 Description of the keyboard

The illustration below shows the various keys used when representing trend diagrams. These keys are outlined in gray and are labeled (the layout and labels of the keys may differ depending on the type of keyboard used).



D

3.9.2 Axis diagram

• Fx type function keys

Functions	Explanation
F1 : Software information	Gives the software version
F2 : Change HMS or HM format	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
F3 : Change scale	Scale : 1 min (60 s), 1/10 min (6s), 1/100 min (600 ms)
F4 : Change ref. point	Changes the reference point of the diagram
F6 : Memorize cursor	Memorizes the cursor position
F8 : Print screen	Prints the part displayed on screen
F9 : Printout	Prints the diagram (between the memorized and current cursor positions)

• Keystrokes

Keystrokes	Explanation
Home, EnD : First/last module	Accesses the first or last module of the diagram
Ctrl+Home : Start of record	Accesses the first module of the diagram
Ctrl+End : End of record	Accesses the last module of the diagram
PgUp,PgDn : Scroll up/down	Moves the window 10 lines up or down
x v : Prev./next instant	Moves 1/5 of a time unit to the previous or next instant
Ctrl Á Ë : Rapid scroll	Moves quickly (10 time units) to the previous or next instant
! à : Prev./next module	Previous or next module
Return : Select module or axis	Displays the corresponding I/O diagram for the selected module
Tab : Display menu window	Displays the menus in a window
Esc : Exit	Exits the module diagram

D

3.9.3 Movement diagram

• Fx type function keys

Functions	Explanation
F3 : Change scale	Scale : 1 min, 1/10 min (6s), 1/100 min (600 ms), 1/1000 min (60 ms)
F4 : Chronolog./numerical order	Displays the movements in numerical or chronological order
F6 : Memo. cursor position	Memorizes the current cursor position
F7 : I/O label	All or part of label
F8 : Print screen	Prints the part displayed on screen
F9 : Printout	Prints the diagram (between the memorized and current cursor positions)

• Shift+Fx type function keys

Functions	Explanation
SF1 : FA representation	Accesses FUNCTIONAL ANALYSIS mode
SF2 : Change HMS or HM format	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
SF3 : Change Input color	Changes the color of the inputs (yellow, white or blue)
SF4 : Change ref. point	Moves the ref. point of the diagram to the current position
SF5 : ZOOM 1 min	Zoom by 1 minute
SF8 : Cont./evolution mvmt	Mode for moving the cursor vertically continuously or by evolution

• Alt+Fx type function keys

Functions	Explanation
AF1 : Save reference	Accesses FUNCTIONAL ANALYSIS mode
AF2 : Comparison active	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
AF3 : Reference cycle duration	Changes the color of the movements (yellow, white or blue)
AF4 : Change ref. point	Moves the ref. point of the diagram to the current position
AF5 : Activate resynchro.	Zoom by 1 minute
AF6 : Find deviation	Mode for moving the cursor vertically continuously or by evolution
AF7 : Change reference ref. point	Modifies the reference point of the reference cycle
AF8 : Info reference cycle	Displays information on the reference cycle

• Keystrokes

Keystrokes	Explanation
Home, End : First/last I/O.	Accesses the first or last movement of the trend diagram
Ctrl+Home : Start of record	Accesses the first movement of the trend diagram
Ctrl+End : End of record	Accesses the last movement of the trend diagram
PgUp,PgDn : Scroll up/down	Moves the window 10 lines up or down
-, + : Prev./next cycle	Moves to the previous or next cycle (in comparison only)
x v : Prev./next instant	Moves 1/5 of a time unit to the previous or next instant
Ctrl x v : Rapid scroll	Moves quickly (10 time units) to the prev. or next instant
! à : Prev./next I/O	Previous or next movement
Shift, Alt : Other menus	Displays the menus for the Shift+Fx or Alt+Fx functions
Tab : Display menu window	Displays the menus in a window
Esc : Exit	Exits the movement diagram

D

3.9.4 I/O diagram

• Fx type function keys

Functions	Explanation
F2 : I/O or Outputs	Displays the I/O or only the outputs
F3 : Change scale	Scale : 1 min, 1/10 min (6s), 1/100 min (600 ms), 1/1000 min (60 ms)
F4 : Chronolog./numerical order	Displays the I/O in numerical or chronological order
F6 : Memo. cursor position	Memorizes the current cursor position
F7 : I/O label	All or part of the label + I/O address
F8 : Print screen	Prints the part displayed on screen
F9 : Printout	Prints the diagram (between the memorized and the current cursor positions)

• Shift+Fx type function keys

Functions	Explanation
SF1 : FA representation	Accesses FUNCTIONAL ANALYSIS mode
SF2 : Change HMS or HM format	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
SF3 : Change Input color	Changes the color of the Inputs (yellow, white or blue)
SF4 : Change ref. point	Moves the ref. point of the diagram to the current position
SF5 : ZOOM 1 min	Zoom by 1 minute
SF8 : Cont./evolution mvmt	Mode for moving the cursor vertically continuously or by evolution

• Alt+Fx type function keys

Functions	Explanation
AF1 : Save reference	Accesses FUNCTIONAL ANALYSIS mode
AF2 : Comparison active	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
AF3 : Reference cycle duration	Changes the color of the inputs (yellow, white or blue)
AF4 : Change ref. point	Moves the ref. point of the diagram to the current position
AF5 : Activate resynchro.	Zoom by 1 minute
AF6 : Find deviation	Mode for moving the cursor vertically continuously or by evolution
AF7 : Change reference ref. point	Modifies the reference point of the reference cycle
AF8 : Reference cycle info	Displays information on the reference cycle

• Keystrokes

Keystrokes		Explanation
Home, End	: First/last I/O.	Accesses the first or last I/O of the trend diagram
Ctrl+Home	: Start of record	Accesses the first I/O of the trend diagram
Ctrl+End	: End of record	Accesses the last I/O of the trend diagram
PgUp,PgDn	: Scroll up/down	Moves the window 10 lines up or down
-, +	: Prev./next cycle	Moves to the previous or next cycle (in comparison only)
x v	: Prev./next instant	Moves 1/5 of a time unit to the previous or next instant
Ctrl x v	: Rapid scroll	Moves quickly (10 time units) to the prev. or next instant
! à	: Prev./next I/O	Previous or next I/O
Shift, Alt	: Other menus	Displays the menus for the Shift+Fx or Alt+Fx functions
Tab	: Display menu window	Displays the menus in a window
Esc	: Exit	Exits the I/O diagram

D

3.9.5 Functional Analysis diagram

• Fx type function keys

Functions	Explanation
F2 : I/O or Outputs	Displays the I/O or only the outputs
F3 : with or without filtering	Activates or deactivates the filter
F4 : Filter/add selected I/O	Filter or add the I/O highlighted by the cursor
F6 : Memo. cursor position	Memorizes the current cursor position
F7 : I/O label or comment	All or part of the label + denomination of I/O
F8 : Print screen	Prints the part displayed on screen
F9 : Printout	Prints the diagram (between the memorized and current cursor positions)

• **Shift+Fx type function keys**

Functions	Explanation
SF2 : Change HMS or HM format	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
SF4 : Change ref. point	Moves the ref. point of the diagram to the current position

• **Alt+Fx type function keys**

Functions	Explanation
AF1 : Save reference	Accesses FUNCTIONAL ANALYSIS mode
AF2 : Comparison active	Time (HH:MM'SS»XX) or mechanical (HH:MM'XXX) format
AF3 : Reference cycle duration	Changes the color of the Inputs (yellow, white or blue)
AF4 : Change ref. point	Moves the ref. point of the diagram to the current position
AF5 : Activate resynchro.	Zoom by 1 minute
AF6 : Find deviation	Mode for moving the cursor vertically continuously or by evolution
AF7 : Change reference ref. point	Modifies the reference point of the reference cycle
AF8 : Reference cycle info	Displays information on the reference cycle

D

• **Keystrokes**

Keystrokes		Explanation
Home,End	: First/last I/O.	Accesses the first or last evolution
PgUp,PgDn	: Scroll up/down	Moves the window 10 lines up or down
-, +	: Prev./next cycle	Moves to the previous or next cycle (in comparison only)
! à	: Prev./next evol.	Previous or next evolution
Shift, Alt	: Other menus	Displays the menus for the Shift+Fx or Alt+Fx functions
Tab	: Display menu window	Displays the menus in a window
Esc	: Exit	Exits FA representation

4.1 General information

Dynamic data exchange (D.D.E.) is an inter-process communication mode which is used to exchange data between applications. This involves transferring information, or even, informing an application of the data changes in another application.

SIMACTEL is a “D.D.E. server”. This means that it is possible for any “D.D.E. client” application to know the state of the SIMACTEL variables, to follow their evolutions, and to change their values. SIMACTEL is only a server during simulation. The SIMACTEL D.D.E. server is available for client WINDOWS applications only when the main simulation window is launched (internal simulation or simulation with PLC).

The name of the D.D.E. server for the SIMACTEL application is : SIMACTEL.

The name of the subject for which a link can be opened is the name of the machine on which the simulation is performed (eg : Model). The syntax of the subject name is the same as for the SIMACTEL machine names; the first letter in the machine name must be in upper case with the rest in lower case.

There are three types of D.D.E. request on this data which are understood by the SIMACTEL server :

- read the value of the data item (READ),
- receive information on each modification made to the data item (INFORMATION),
- change the value of the data item (WRITE).

Depending on the data, only one or some of these requests are available. For each data item, the possible requests are detailed in the remainder of this section.

Note

> To respect the D.D.E. standard as closely as possible, SIMACTEL differentiates between upper and lower case characters. Similarly, accented characters are distinguished from non accented characters.

Only the CF_TEXT format is currently recognized by the SIMACTEL server.

4.2 Information on simulation

• status

Requests : READ, INFORMATION, WRITE.

Operating status of the SIMACTEL server. The value returned belongs to all the following character strings :
• “current” if simulation is in progress,
• “interrupted” if simulation is interrupted,
• “blocked” if simulation is blocked.

To modify the simulation state, the value to be sent to the SIMACTEL server is one of the following character strings :
• “interrupt” : interrupts the simulation,
• “relaunch” : relaunches the simulation.

Note

It is not possible to modify the simulation state when the current state is “blocked”.

D

• mode

Requests : READ, WRITE.

Simulation operating mode. The value returned by the SIMACTEL server is one of the following character strings :
• “step by step” if step by step simulation mode is the current simulation mode,
• “periodic” if the current simulation mode is periodic mode,
• “continuous” if the current simulation mode is continuous mode.

To modify the data item mode, the value to be sent to the SIMACTEL server is one of the three character strings listed above.

Note

To select periodic simulation mode, a period must already have been defined.

• period

Requests : READ, WRITE.

Period associated with periodic operating mode. The value returned by the SIMACTEL server is a character string containing the value of the period (eg : “0.001”). Similarly, to change the value of the simulation period, the client application sends a character string containing a number to the SIMACTEL server (the number formats which can be understood by SIMACTEL are those accepted by the simulation interface).

The value returned is the character string “UNKNOWN” if no period is set in the SIMACTEL server.

• **realtime**

Requests : READ, WRITE.

Indicates whether the simulation is in realtime mode. The value returned (to be sent to the SIMACTEL server) is one of the following character strings :
• "1" if the realtime option is active,
• "0" if the realtime option is inactive.

• **pulse**

Requests : READ, WRITE.

Returns the state of the "Stop if Output Pulse" check box. The value returned (to be sent to the SIMACTEL server) is one of the following character strings :
• "1" if the option is active,
• "0" if the option is inactive.

• **time**

Requests : READ, INFORMATION.

Returns the current time value in the form of a character string. Depending on the type of representation selected, the value returned is one of the following :
• "HH:MM:SS.XX" (HH : hours, MM : minutes, SS : seconds, XX : remainder),
• "MM:SS.XX" (MM : minutes, SS : seconds, XX : remainder) if the time sent is less than the one hour.

Note

If the SIMACTEL model is calculated in hundredths of a minute, the time sent by the D.D.E. link is automatically converted to the hours, minutes, seconds format.

• **time:unit**

Request : READ.

Returns the unit used to calculate the time in the form of a character string. The value returned belongs to the following character strings :
• "1/100 min.",
• "sec.".

D

• type

Request : READ.

Returns the list of machine type element names in the form of a character string. Each name is delimited by the separators “CR” and “LR” (\r\n). Type is a type of SIMACTEL element. Its value is to be used in all of the following :

- “external”,
- “inputs”,
- “outputs”,
- “supplies”,
- “relays”,
- “axes”,
- “sensors”,
- “movements”,
- “bistables”,
- “variables”,
- “traps”.

An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• type:nb

Request : READ.

Returns the number of description elements for a particular type. Type is a type of SIMACTEL element. The value of the data item is to be used in all of the following :

- “external:nb”,
- “inputs:nb”,
- “outputs:nb”,
- “supplies:nb ”,
- “relays:nb”,
- “axes:nb”,
- “sensors:nb”,
- “movements:nb”,
- “bistables:nb”,
- “variables:nb”,
- “traps:nb”.

D

4.3 Managing description elements

4.3.1 Power supply

• supplies

Request : READ.

Returns the list of the machine power supplies in the form of a character string. Each supply name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• supplies:nb

Request : READ.

Returns the number of the machine power supply in the form of a character string containing a number.

• **supply:elt:state**
Requests : READ, INFORMATION.
State of the supply elt, where elt is the name of a SIMACTEL power supply written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• “0” if the supply is inactive,
• “1” if the supply is active.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL power supply.

• **supply:elt:fault**
Requests : READ, INFORMATION, WRITE.
Faults set on the supply elt, where elt is the name of a SIMACTEL power supply written in lower case. The values obtained from the SIMACTEL server belong to all the following character strings:
• “stick at 0”,
• “stick at 1”,
• “force to 0”,
• “force to 1”,
• “” (empty character string) if no fault is set on the supply elt.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL power supply.

To set a fault, the client application sends one of the character strings listed above. To remove all the faults set on the supply elt, the client application sends the character string “none” to the SIMACTEL server.

Note

> The write request has no effect if the SIMACTEL interface window for setting faults on supplies is open.

4.3.2 Axis

• **axes**
Request : READ.
Returns the list of machine axes in the form of a character string. Each axis name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• **axes:nb**
Request : READ.
Returns the number of machine axes in the form of a character string containing a number.

D

• **axis:elt:state**
Requests : READ, INFORMATION.
State of the axis elt, where elt is the name of a SIMACTEL axis written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• “0” if the axis is inactive (no current axis movement),
•“1” if the axis is active (a movement is being executed).

The character string “UNKNOWN” is returned if elt is not a SIMACTEL axis.

• **axis:elt:position**
Requests : READ, INFORMATION.
Current position of the axis elt, where elt is the name of a SIMACTEL axis written in lower case. The value returned is in the form of a character string containing a number between 0 and 1. The value returned is the percentage of outputs for the axis elt.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL axis.

• **axis:elt:length**
Request : READ.
Length of the axis elt, where elt is the name of a SIMACTEL axis written in lower case. The value is returned in the form of a character string containing a number.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL axis.

D

4.3.3 Bistable

• **bistables**
Request : READ.
Returns the list of machine bistables in the form of a character string. Each bistable name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• **bistables:nb**
Request : READ.
Returns the number of machine bistables in the form of a character string containing a number.

• **bistable:elt:state**
Requests : READ, INFORMATION.
State of the bistable elt, where elt is the name of a SIMACTEL bistable written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• “0” if the bistable is inactive,
• “1” if the bistable is active.
The character string “UNKNOWN” is returned if elt is not a SIMACTEL bistable.

• **bistable:elt:fault**
Requests : READ, INFORMATION, WRITE.
Faults set on the bistable elt, where elt is the name of a SIMACTEL bistable written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• "stick at 0",
• "stick at 1",
• "force to 0",
• "force to 1",
• "" (empty character string) if no fault has been set on the bistable elt.

The character string "UNKNOWN" is returned if elt is not a SIMACTEL bistable.
To set a fault, the client application sends one of the character strings listed above. To remove all the faults set on the bistable elt, the client application sends the character string "none" to the SIMACTEL server.

Note

The write request has no effect if the SIMACTEL interface window for setting faults on bistables is open.

4.3.4 Sensor

• **sensors**
Request : READ.
Returns the list of machine sensors in the form of a character string. Each sensor name is delimited by the separators "CR" and "LF" (\r\n).

An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• **sensors:nb**
Request : READ.
Returns the number of machine sensors in the form of a character string containing a number.

• **sensor:elt:state**
Requests : READ, INFORMATION.
State of the sensor elt, where elt is the name of a SIMACTEL sensor written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• "0" if the sensor is inactive,
• "1" if the sensor is active.

The character string "UNKNOWN" is returned if elt is not a SIMACTEL sensor.

- **sensor:elt:fault**

Requests : READ, INFORMATION, WRITE.

Faults set on the sensor elt, where elt is the name of a SIMACTEL sensor written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :

• "stick at 0",
• "stick at 1",
• "force to 0",
• "force to 1",
• "bounce on activation",
• "bounce on deactivation",
• "" (empty character string) if no fault has been set on the sensor elt.

The character string "UNKNOWN" is returned if elt is not a SIMACTEL sensor.
The result of the request can be several concatenated character strings if, for example, forcing and contact bounce are set at the same time.

To modify the faults set on the sensor elt, the client application sends one of the following character strings :

• "stick at 0",
• "stick at 1",
• "force to 0",
• "force to 1",
• "none".

D

Notes

The write request has no effect if the SIMACTEL interface window for setting faults on sensors is open. To set contact bounce from the client application, use the "bounce" request.

- **sensor:elt:bounce**

Request : WRITE.

Sets the bounce on elt, where elt is the name of a SIMACTEL sensor written in lower case. The value sent to the SIMACTEL server belong to the following character strings :

• "bounce on activation",
• "bounce on deactivation",
• "none" to remove all contact bounces set on the sensor elt.

Note

The write request has no effect if the SIMACTEL interface window for setting bounce on sensors is open.

4.3.5 Input

• inputs
Request : READ.
Returns the list of machine inputs in the form of a character string. Each input name is delimited by the separators “CR” and “LF” (\r\n).

An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• inputs:nb
Request : READ.
Returns the number of machine inputs in the form of a character string containing a number.

• input:elt:state
Request : READ, INFORMATION.
State of the input elt, where elt is the name of a SIMACTEL input written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• “0” if the input is inactive,
• “1” if the input is active.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL input.

4.3.6 External

• external
Request : READ.
Returns the list of external machine variables in the form of a character string. Each external variable name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• external:nb
Request : READ.
Returns the number of external variables in the form of a character string containing a number.

• external:elt:state
Requests : READ, INFORMATION, WRITE.
State of the external variable elt, where elt is the name of a SIMACTEL external variable written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :

• “ 0 ” if the external variable is inactive,
• “ 1 ” if the external variable is active.

The character string “UNKNOWN” is returned if elt is not an external SIMACTEL variable.

To set the value of an external variable, the client application sends one of the two character strings defined above.

Note

The write request has no effect if the SIMACTEL interface window for modifying external variables is open.

D

4.3.7 Movement

• movements

Request : READ.

Returns the list of machine movements in the form of a character string. Each movement name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• movements:nb

Request : READ.

Returns the number of machine movements in the form of a character string containing a number.

• movement:elt:state

Requests : READ, INFORMATION.

State of the movement elt, where elt is the name of a SIMACTEL movement written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :

• “0” if the movement is inactive,
• “1” if the movement is active.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL movement.

4.3.8 Trap

The traps listed and used by D.D.E. requests are only those traps defined during simulation.

• traps
Request : READ.
Returns the list of machine traps in the form of a character string. Each trap name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• traps:nb
Request : READ.
Returns the number of machine simulation traps in the form of a character string containing a number.

• trap:elt:state
Requests : READ, INFORMATION.
State of elt, where elt is the name of a SIMACTEL simulation trap written in lower case.

The values obtained from the SIMACTEL server belong to the following character strings :
• “0” if the trap is inactive,
• “1” if the trap is active.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL simulation trap.

• trap:elt:number
Request : READ.
Number of recorded activations of elt, where elt is the name of a SIMACTEL simulation trap written in lower case. The value obtained is a character string containing a number.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL simulation trap.

• trap:elt:average
Request : READ.
Average time difference recorded between two activations of elt, where elt is the name of a SIMACTEL simulation trap written in lower case. The value obtained is a character string containing a number.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL simulation trap.

• trap:elt:min
Request : READ.
Minimum time difference recorded between two activations of elt, where elt is the name of a SIMACTEL simulation trap written in lower case. The value obtained is a character string containing a number.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL simulation trap.

• **trap:elt:max**
Request : READ.
Maximum time difference recorded between two activations of elt, where elt is the name of a SIMACTEL simulation trap written in lower case. The value obtained is a character string containing a number.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL simulation trap.

4.3.9 Relays

• **relays**
Request : READ.
Returns the list of machine relays in the form of a character string. Each relay name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

D

• **relays:nb**
Request : READ.
Returns the number of machine relays in the form of a character string containing a number.

• **relay:elt:state**
Requests : READ, INFORMATION.
State of the relay elt, where elt is the name of a SIMACTEL relay written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• “0” if the relay is inactive,
• “1” if the relay is active.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL relay.

• **relay:elt:fault**
Requests : READ, INFORMATION, WRITE.
Faults set on the relay elt, where elt is the name of a SIMACTEL relay written in lower case. The values obtained from the SIMACTEL server belong to the following character strings:
• “stick at 0”,
• “stick at 1”,
• “force to 0”,
• “force to 1”,
• “” (empty character string) if no fault has been set on the relay elt.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL relay.
To set a fault, the client application sends one of the character strings listed on the previous page. To remove all the faults set on the relay elt, the client application sends the character string “none” to the SIMACTEL server.

Note

> The write request has no effect if the SIMACTEL interface window for setting faults on relays is open.

4.3.10 Output

• outputs
Request : READ.
Returns the list of machine outputs in the form of a character string. Each output name is delimited by the separators “CR” and “LF” (\r\n).

An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• outputs:nb
Request : READ.
Returns the number of machine outputs in the form of a character string containing a number.

• output:elt:state
Requests : READ, INFORMATION.
Returns the value of the output elt, where elt is the name of a SIMACTEL output written in lower case. The value belong to the following character strings :
• “1” if the output is at the high state,
• “0” otherwise.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL output.

D

4.3.11 Variable

• variables
Request : READ.
Returns the list of machine variables in the form of a character string. Each variable name is delimited by the separators “CR” and “LF” (\r\n). An empty character string is returned if the SIMACTEL server does not contain any elements of this type.

• variables:nb
Request : READ.
Returns the number of machine variables in the form of a character string containing a number.

• variable:elt:state
Requests : READ, INFORMATION.
Returns the value of the variable elt, where elt is the name of a SIMACTEL output written in lower case. The value belongs to the following elements :
• “1” if the variable is at the upper state,
• “ 0 ” otherwise.

D

The character string “UNKNOWN” is returned if elt is not a SIMACTEL variable.

• variable:elt:fault
Requests : READ, INFORMATION, WRITE.
Faults set on the variable elt, where elt is the name of a SIMACTEL variable written in lower case. The values obtained from the SIMACTEL server belong to the following character strings :
• “force to 0”,
• “force to 1”,
• “” (empty character string) if no fault has been set on the variable elt.

The character string “UNKNOWN” is returned if elt is not a SIMACTEL variable.

To set a fault, the client application sends one of the character strings listed above. To remove all the faults set on the variable elt, the client application sends the character string “none” to the SIMACTEL server.

Note

The write request has no effect if the SIMACTEL interface window for setting faults on variables is open.

4.4 Summary table

Request	Read	Write	Info.	Comment
axes	yes			list of axes, page 4-5
axes:nb	yes			number of axes, page 4-5
axis:elt:length	yes			length of an axis, page 4-6
axis:elt:position	yes		yes	position of an axis, page 4-6
axis:elt:state	yes		yes	state of an axis, page 4-6
bistable:elt:fault	yes	yes	yes	faults set on a bistable, page 4-7
bistable:elt:state	yes		yes	state of a bistable, page 4-6
bistables	yes			list of bistables, page 4-6
bistables:nb	yes			number of bistables, page 4-6
external	yes			list of external variables, page 4-9
external:elt:state	yes	yes	yes	state of an external variable, page 4-9
external:nb	yes			number of external variables, page 4-9
input:elt:state	yes	yes	yes	state of an input, page 4-9
inputs	yes			list of inputs, page 4-9
inputs:nb	yes			number of inputs, page 4-9
mode	yes	yes		current simulation mode, page 4-3
mode:period	yes	yes		period, page 4-2
movement:elt:state	yes		yes	state of a movement, page 4-10
movements	yes			list of movements, page 4-10
movements:nb	yes			number of movements, page 4-10
output:elt:state	yes		yes	state of an output, page 4-13
outputs	yes			list of outputs, page 4-13
outputs:nb	yes			number of outputs, page 4-13
pulse	yes	yes		option to stop on an output pulse, page 4-3
realtime	yes	yes		realtime option, page 4-3
relay:elt:fault	yes	yes	yes	faults set on a relay, page 4-12
relay:elt:state	yes		yes	state of a relay, page 4-12
relays	yes			list of relays, page 4-12
relays:nb	yes			number of relays, page 4-12
sensor:elt:bounce		yes		bounce set on a sensor, page 4-8
sensor:elt:fault	yes	yes	yes	fault set on a sensor, page 4-8
sensor:elt:state	yes		yes	state of a sensor, page 4-7
sensors	yes			list of sensors, page 4-7
sensors:nb	yes			number of sensors, page 4-7
status	yes	yes	yes	state of the simulation, page 4-2
supply:elt:fault	yes	yes	yes	faults set on a supply, page 4-5
supply:elt:state	yes		yes	state of the supply elt, page 4-5
supplies	yes			list of supplies, page 4-4
supplies:nb	yes			number of supplies, page 4-4
time	yes			time, page 4-3
time:unit	yes			time unit, page 4-3

D

trap:elt:average	yes			average time difference between two activations, page 4-11
trap:elt:max	yes			max. time difference between two activations, page 4-11
trap:elt:min	yes			min. time difference between two activations, page 4-11
trap:elt:number	yes			number of activations of a trap, page 4-11
trap:elt:state	yes		yes	state of a simulation trap, page 4-11
traps	yes			list of simulation traps, page 4-11
traps:nb	yes			number of simulation traps, page 4-11
variable:elt:fault	yes	yes	yes	faults set on a variable, page 4-14
variable:elt:state	yes		yes	state of a variable, page 4-14
variables	yes			list of variables, page 4-14
variables:nb	yes			number of variables, page 4-14

D

4.5 Advice and problems relating to D.D.E.

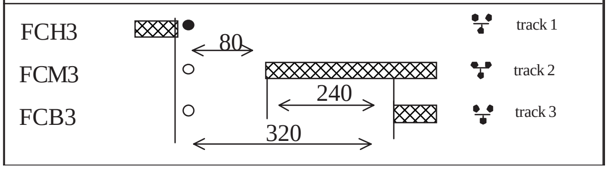
- Request mode does not operate when continuous and periodic modes are selected at the same time. Only one of the current modes is returned by the D.D.E. link.
- There are two D.D.E. link modes, synchronous and asynchronous links. With a synchronous link, the client and server processes are executed at the same rate, one waiting for the other. All evolutions in the server appear in the client. As a result, the SIMACTEL server execution is slowed down by the connection. This mode is that used by the Microsoft EXCEL D.D.E. client, for example. However, with an asynchronous link, client and server change at their own rates. In this configuration, it is possible for the client to “miss” some changes to the server which are too quick. This connection mode is that of NetDDE for Windows for WorkGroups.

1.1 Single axis

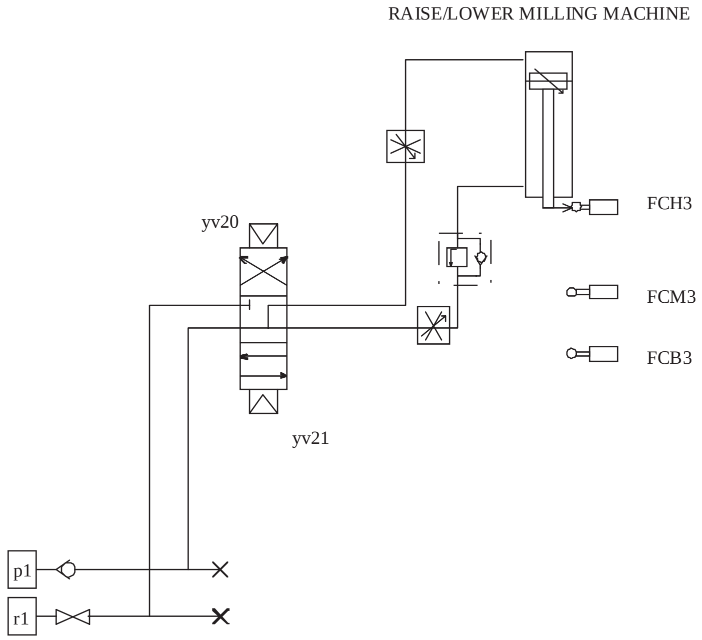
The diagram below illustrates the mechanical and pneumatic descriptions of a single linear axis.

• Mechanical and pneumatic description

cylinder : lower milling machine						
Øpiston : 100	Ørod : 56	total travel : 350mm				
unit cycle :		initial position	1 st lowering of head	2nd lowering of head	raise head	
control of coils for each movement	YV20		■	■		
	YV21				■	
travel in mm			80	240	320	
speed in m/min			80	80	100	
position of sensors and description of movements						



E



E

• **SIMACTEL description**

Axis n ° 3
• Label : lower milling machine • Type : linear • Initial position : start limit • Length : 350

Movement mvt3+
• Direction : positive • Speed : 80 • Equation : (yv20 * pneu)

Movement mvt3-
• Direction : negative • Speed : 100 • Equation : (yv21 * pneu)

Sensor FCH3	**Sensor FCM3**	**Sensor FCB3**
• Logic : positive • Position : start limit	• Logic : latch • Position : 80	• Logic : latch • Position : 240

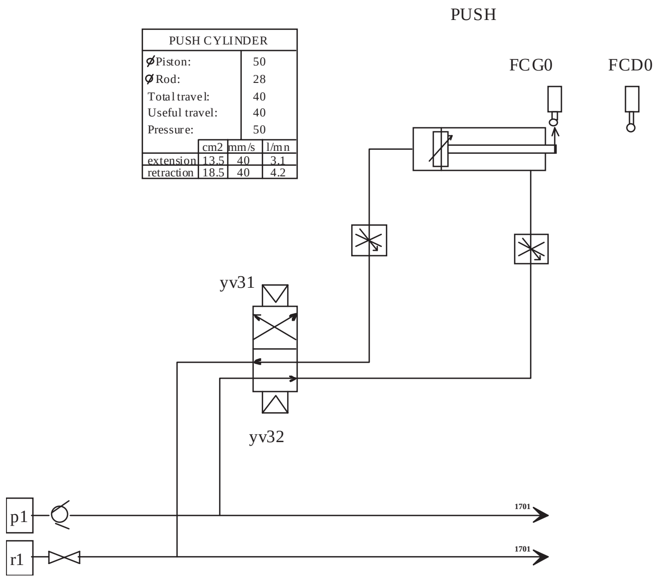
Note

The commands yv20 and yv21 activating the movements mvt3+ and mvt3- are SIMACTEL relays. pneu is a SIMACTEL general power supply term.

1.2 Push cylinder

The axis illustrated below differs from the previous example in the type of **command used to activate** the movements on the axis. The movement command is given by a bistable instead of a relay.

- **Mechanical and pneumatic description**



- **SIMACTEL description**

Axis n ° 1
- Label : push
- Type : linear
- Initial position : start limit
- Length : 40

Movement mvt1+
- Direction : positive
- Speed : 40
- Equation : (bis1 * pneu)

Movement mvt1-
- Direction : negative
- Speed : 40
- Equation : (/bis1 * pneu)

Sensor FCG0
• Logic : positive • Position : start limit

Sensor FCD0
• Logic : positive • Position : end limit

Bistable bis1
• Latch : yv31 • Release : yv32 • Condition :

Note

The commands yv31 and yv32 activating the movements mvt1+ and mvt1- are SIMACTEL relays. pneu is a SIMACTEL general power supply term.

1.3 Mechanical interlocking

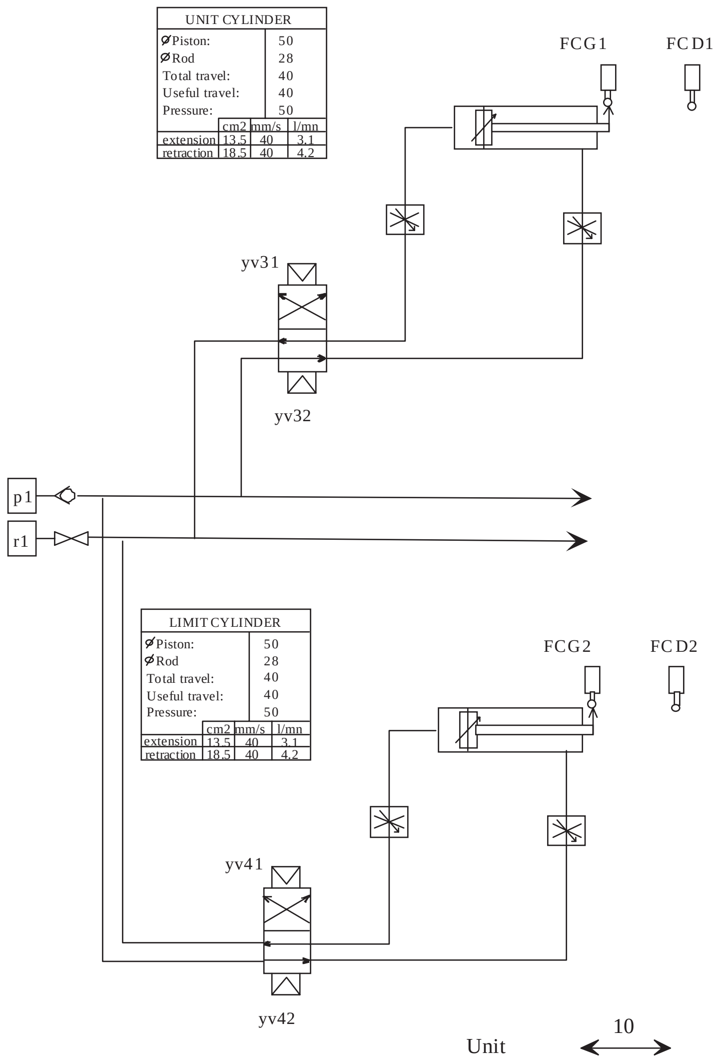
The following example concerns two single axes where one is the mechanical lock for the other. This interaction is taken into account by adding a fictitious sensor representing the free travel of axis 1 and integrating the position of axis 2 in the equation for movement mvt1+.

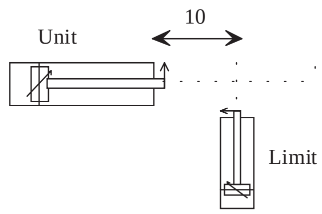
E

• Mechanical and pneumatic description

UNIT CYLINDER			
ØPiston:	50		
ØRod	28		
Total travel:	40		
Useful travel:	40		
Pressure:	50		
	cm2	mm/s	l/mn
extension	13.5	40	3.1
retraction	18.5	40	4.2

LIMIT CYLINDER			
ØPiston:	50		
ØRod	28		
Total travel:	40		
Useful travel:	40		
Pressure:	50		
	cm2	mm/s	l/mn
extension	13.5	40	3.1
retraction	18.5	40	4.2





E

• **SIMACTEL description of axis n° 1**

Axis n ° 1
- Label : unit
- Type : linear
- Initial position : start limit
- Length : 40

Movement mvt1+
- Direction : positive
- Speed : 40
- Equation :
(bis1 * pneu * (/cf + (cf * fcd2)))

Movement mvt1-
- Direction : negative
- Speed : 40
- Equation :
(/bis1 * pneu)

Sensor FCG1
- Logic : positive
- Position : start limit

Sensor FCD1
- Logic : positive
- Position : end limit

Sensor CF
- Logic : latch
- Position : 10

Bistable bis1
- Latch : yv31
- Release : yv32
- SIMACTEL description of axis n° 2

Axis n ° 2
- Label : lock
- Type : linear
- Initial position : start limit
- Length : 40

Movement mvt2+
- Direction : positive
- Speed : 40
- Equation : (bis2 * pneu)

Movement mvt2-
- Direction : negative
- Speed : 40
- Equation : (/bis2 * pneu)

Sensor FCG2
- Logic : positive
- Position : start limit

Sensor FCD2
- Logic : positive
- Position : end limit

Bistable bis2
- Latch : yv41
- Release : yv42

E

Note

The commands yv31, yv32, yv41 and yv42 activating the movements mvt1+, mvt1, mvt2+ and mvt2- are SIMACTEL relays. pneu is a general SIMACTEL power supply term.

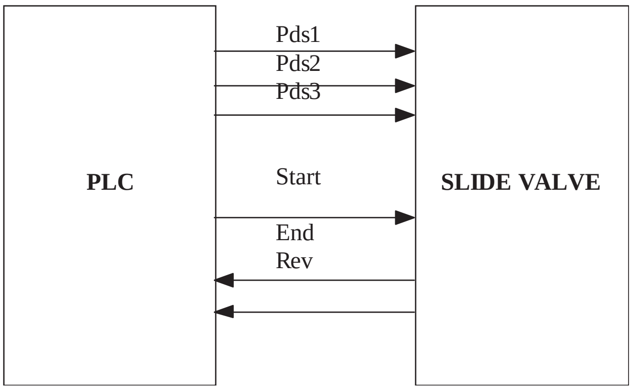
E

E

2.1 Solder slide valve

The application in SIMACTEL comprises the modelling of exchanges between the PLC and intelligent devices, such as robots, weighing equipment, or, as in the example below, a solder slide valve.

The principle is to represent each cycle of exchanges between the PLC and the device by a SIMACTEL rotary axis, where the PLC command words activate the axis movements and the axis sensors represent the equipment responses.



Pds1, **Pds2** and **Pds3** : solder codes sent by the PLC to the solder slide valve to define the time and intensity of soldering.
Start : the PLC gives the start of the solder cycle.
End : end of cycle sent by the slide valve.
Rev : relay managed by the solder slide valve used to close the selected grip.

E

• **SIMACTEL description**

The length of the axis corresponds to the solder cycle time and the movement is made at a speed of 1..

The description of the dialog is contained in the logic equation of movement mvt1. This equation includes three terms :

- The first " dx * start * cod1 " corresponds to the start of the movement on the axis at the initial position (dx = 1). The movement is active when the PLC sends the start command and the correct solder code identified by cod1. Cod1 is a SIMACTEL variable which depends on the solder code sent by the PLC. Cod1 is at state 1 if the code sent corresponds to the solder cycle simulated by the SIMACTEL axis.
- " /dx * /fx * cod1 * start " corresponds to maintaining the movement for the entire solder time. At state 1 of this term, the cam sweeps the axis and SIMACTEL sends information on the progress of the soldering to the PLC.
- " fx * /start " is the term which enables the axis to return to its initial position to be ready for the next solder cycle. This term is at state 1 when the axis is at the end limit and the start command is issued.

The fictitious sensor Rev simulates the relay Rev of the solder slide valve. The sensor fx simulates the End of cycle signal.

Axis n ° 1
• Label : lower milling machine • Type : Rotary • Initial position : start limit • Length : 100

Movement mvt1
• Direction : positive • Speed : 1 • Equation : (dx * start * cod1) + (/dx * /fx * cod1 * start) + (fx * /start)

Sensor Rev
• Logic : positive • Position : 10 • Activation width : 80

Sensor dx	**Sensor fx**
• Logic : positive • Position : start limit	• Logic : positive • Position : end limit

E

3.1 Presentation

In materials handling type processes, the PLC receives information from the machine on the presence and movement of parts. We have seen in the examples in previous sections that a machine application is represented, in SIMACTEL, by axes and Boolean variables. The parts are not represented explicitly. The SIMACTEL application description simulates the presence of parts by calculating their actions on the PLC inputs according to the PLC outputs and the time elapsed.

The following section presents the principle of modelling a conveyor system. Examples of transferring in one, then in two operating directions are given. The detection of parts, taking account of their length, is also explained.

3.2 Principle of modelling

The modelling of a conveyor system consists of dividing the system into zones where only one part may be found : this is the concept of “section”.

The presence of a part in a zone is represented by a Boolean variable “PPi”. The movement takes a certain amount of time, and an axis associated with each “section” is used to represent this time : the position on the axis is the “front edge” of the moving part. The movement of this axis represents the movement of the part in the zone described by the axis. The equation for this movement uses the variable “PPi” and a term - for example “KMVTi” representing the rotation of the motor driving the conveyor.

The movement of the part in the zone described by the axis is considered to be a process made up of three states. The initial state corresponds to the availability of the conveyor for a part arriving upstream. The second state is the movement - and also stopping at a given position - of the part on the section. The final state is the end of the transfer in this zone, where the part is made available to the next section. This process is cyclical, the axis is thus rotary. A fictitious sensor “DXi” of width zero, placed at the start of the axis, represents the availability of the section. Similarly, a sensor “FXi”, also with a zero activation width, is placed at the end of the rotary axis to represent making the part available for the rest of the transfer.

Note

An analogy may be drawn between the “true” state of the sensor “DXi” and the absence of a part on the section. This analogy is used for modelling the transfers on the machine with three machining stations used in the next section. However, we are not referring to it in this section as it causes a problem in the case of transfers with movements in two directions.

The evolution of the transfer process is described via a movement equation. The part is moved when the motor is running and :

- a part is present and the end of the section has not been reached.
- the end of the section has been reached and the part is taken into the next section or removed.

The following equation is obtained for the movement :

MVTi = KMVTi * (PPi * /FXi + FXi * /PPi)

Once it reaches “FXi” and the conveyed part - “PPi” is reset to 0 - the rotary axis returns to the initial position - on “DXi” - and is then available for a new part.

A part enters the section if the section is at its initial state and the previous section makes a part available. Taking account of the representation of states of the section via the sensors “DXi” and “FXi”, the latch term of the part present variable “PPi” is :

DXi * PPi-1 * FXi-1 * KMVTi-1

The presence of “KMVTi-1” keeps the part in the previous section if the motor for that section is not running - this should definitely not happen.

Once the transfer is finished on a section - the corresponding axis is then at “FXi” - the part can move on to the next section if this is available - the axis describing it is then “DXi+1”. However, it should be checked that this section does not already contain a part. In fact, if the motor of the next section is not controlled, the axis will have been able to stay at “DXi+1” even though a part is present on this section. In these conditions, the release term of variable “PPi” has the following expression :

RPPi = FXi * DXi+1 * /PPi+1 * KMVTi

The equation of the parts present variable “PPi” has the following final expression :

PPi = (DXi * PPi-1 * FXi-1 * KMVTi-1 + PPi) * /RPPi

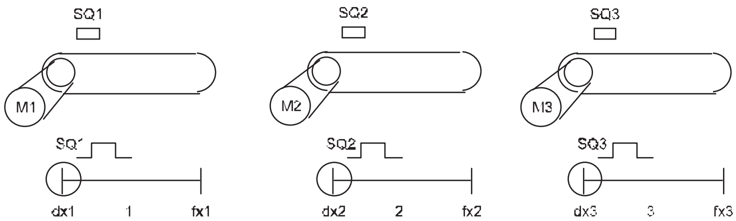
3.3 Simple forward operation

The following example shows the movement of parts on three conveyor belts in forward operation. The size of the parts is considered negligible, there is only one part per belt and the motors for the belts are in continuous operation. The belts are represented by SIMACTEL rotary axes. The presence of a part is represented by a Boolean variable.

The movements take place when the following two conditions are true :
- there is a part on the axis,
- the conveyor belt motor is in forward operation.

For a part to move from one conveyor belt to another, the installation must also meet the following two criteria :

- the downstream axis must be at the reference point (waiting for parts),
- there is no part on the downstream axis (avoid clashes).



Description of the first conveyor belt

The first conveyor belt is at the beginning of the conveyor system, a variable “BPP1” is used to introduce a part in the system.

The part appears on axis 1 when the pushbutton is pressed and the axis is at the start limit. This state is maintained until RPP1 (Boolean variable for resetting the part present) is true. RPP1 is true when axis 1 is at the end limit, axis 2 at the start limit, there is no part on the next conveyor belt and the motor of conveyor belt 1 is active.

Movement MVT1 is active as long as the motor is active, PP1 is true or to move automatically from the end limit position to the start limit position.

Equations of the first conveyor belt

- PP1 = (+BPP1 * DX1 + PP1) * /RPP1
- RPP1 = FX1 * DX2 * /PP2 * KMVT1
- MVT1 = (PP1 * /FX1 + FX1 * /PP1) * KMVT1

where :

- PP1 : part present on conveyor belt 1,
- PP2 : part present on conveyor belt 2,
- DX1 : start limit sensor of axis 1,

- DX2 : start limit sensor of axis 2,
- FX1 : end limit sensor of axis 1,
- +BPP1 : rising edge of fictitious pushbutton used to place parts on the first conveyor belt,
- RPP1 : resets part present.
- KMVT1 : controls the motor on conveyor belt 1.

Description of the second conveyor belt

The part is considered to be present on axis 2 when axis 2 is at the start limit, axis 1 is at the end limit, there is a part on axis 1 and the motor of axis 1 is active. The part present on axis 2 is held in the same way as the part present for axis 1.

The other equations are similar to those of axis 1.

Equations of the second conveyor belt

- PP2 = (DX2 * FX1 * PP1 * KMVT1 + PP2) * /RPP2
- RPP2 = FX2 * DX3 * /PP3 * KMVT2
- MVT2 = (PP2 * /FX2 + FX2 * /PP2) * KMVT2

where :

E

- PP1 : part present on conveyor belt 1,
- PP2 : part present on conveyor belt 2,
- DX2 : start limit sensor of axis 2,
- DX3 : start limit sensor of axis 3,
- FX1 : end limit sensor of axis 1,
- FX2 : end limit sensor of axis 2,
- RPP2 : resets part present,
- KMVT1 : controls the motor on conveyor belt 1.,
- KMVT2 : controls the motor on conveyor belt 2.

Description of the third conveyor belt

We assume that the parts disappear automatically at the end of conveyor belt 3 (removed to part of the installation which has not been simulated, for example). For the rest, the equations are similar to those described previously.

Equations of the third conveyor belt

- PP3 = (DX3 * FX2 * PP2 * KMVT2 + PP3) * /RPP3
- RPP2 = FX3 * KMVT3
- MVT3 = (PP3 * /FX3 + FX3 * /PP3) * KMVT3

where :

- PP2 : part present on conveyor belt 2,
- PP3 : part present on conveyor belt 3,
- DX3 : start limit sensor of axis 3,
- FX2 : end limit sensor of axis 2,
- FX3 : end limit sensor of axis 3,
- RPP3 : resets part present,
- KMVT2 : controls the motor on conveyor belt 2,
- KMVT3 : controls the motor on conveyor belt 3.

3.4 Forward and reverse operation

This example uses the description in section 3.3 but includes the reverse operation of the conveyor belt. The equations for forward movements remain the same.

SIMACTEL description

In the part present equations, in comparison with the previous example (see section 3.3 of this part), an additional term is added. When a part is moving from one conveyor belt to another, it should be checked that the conveyor belt which is to receive the part is turning in the same direction as the conveyor belt which is bringing the part to it. If not, the part must not change section.

The SIMACTEL description elements are as follows :

- PP1 : Boolean variable for part present on conveyor belt 1,
- PP2 : Boolean variable for part present on conveyor belt 2,
- PP3 : Boolean variable for part present on conveyor belt 3,
- DX1 : start limit sensor of axis 1,
- DX2 : start limit sensor of axis 2,
- DX3 : start limit sensor of axis 3,
- FX1 : end limit sensor of axis 1,
- FX2 : end limit sensor of axis 2,
- FX3 : end limit sensor of axis 3,
- +BPP1 : rising edge of fictitious pushbutton used to place parts on the first conveyor belt,
- RPP1 : Boolean variable to reset part present on conveyor belt 1,
- RPP2 : Boolean variable to reset part present on conveyor belt 2,
- RPP3 : Boolean variable to reset part present on conveyor belt 3,

- MTAV1 : controls motor for moving conveyor belt 1 forward,
- MTAV2 : controls motor for moving conveyor belt 2 forward,
- MTAV3 : controls motor for moving conveyor belt 3 forward,
- MTAR1 : controls motor for moving conveyor belt 1 backward,
- MTAR2 : controls motor for moving conveyor belt 2 backward,
- MTAR3 : controls motor for moving conveyor belt 3 backward,
- MVAV1 : forward movement of axis 1,
- MVAV2 : forward movement of axis 2,
- MVAV3 : forward movement of axis 3,
- MVAR1 : backward movement of axis 1,
- MVAR2 : backward movement of axis 2,
- MVAR3 : backward movement of axis 3.

Equations of the first conveyor belt

- PP1 = (+BPP1 * DX1 + FX1 * DX2 * PP2 * MTAR2 * /MTAV1 + PP1) * /RPP1
- RPP1 = FX1 * DX2 * /PP2 * MTAV1 * /MTAR2 + DX1 * MTAR1
- MVAV1 = (PP1 * /FX1 + FX1 * /PP1) * MTAV1
- MVAR1 = (PP1 * /DX1 + DX1 * /PP1) * MTAR1

E

Equations of the second conveyor belt

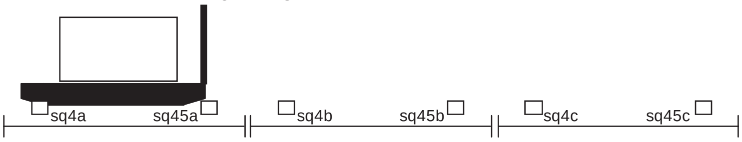
- PP2 = (DX2 * FX1 * PP1 * MTAV1 * /MTAR2 + FX2 * DX3 * PP3 * MTAR3 * /MTAV2 + PP2) * /RPP2
- RPP2 = FX2 * DX3 * /PP3 * MTAV2 * /MTAR3 + DX2 * FX1 * /PP1 * /MTAV1 * MTAR2
- MVAV2 = (PP2 * /FX2 + FX2 * /PP2) * MTAV2
- MVAR2 = (PP2 * /DX2 + DX2 * /PP2) * MTAR2

Equations of the third conveyor belt

- PP2 = (DX3 * FX2 * PP2 * MTAV2 * /MTAR3 + PP3) * /RPP3
- RPP3 = FX3 * MTAV3 + DX3 * FX2 * /PP2 * /MTAV2 * MTAR3
- MVAV3 = (PP3 * /FX3 + FX3 * /PP3) * MTAV3
- MVAR3 = (PP3 * /DX3 + DX3 * /PP3) * MTAR3

3.5 Taking account of the length of the part

The example described here has three elements. It concerns roller tables 6000 mm in length on which sleds of length 5000 mm move.

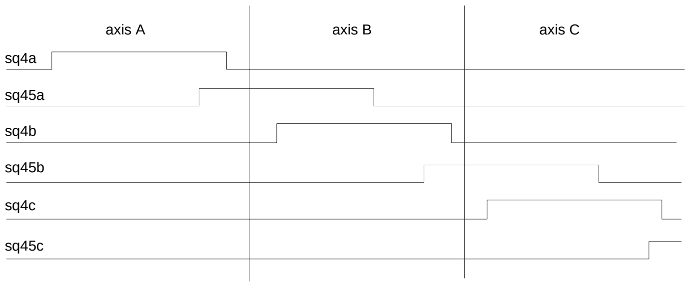


The presence of parts is represented by Boolean variables, the position on an axis describing a section corresponding to the front edge of the part. In this modelling operation, there is no mention of the length of the part, the only constraint imposed is that there should only be one part per section.

However, these parts signal their movement to the PLC via detectors. At this level at least, the length of these parts should be taken into account : in fact, the detectors are active the whole time the parts are moving.

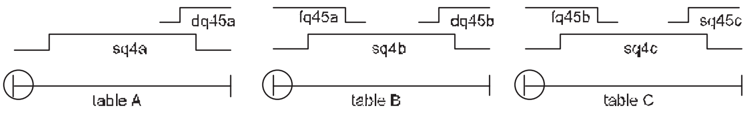
This cannot be done by simply placing real sensors on the axes. In particular, when one part starts to enter the next section, it continues to be detected in the section it is leaving, because of its length.

To do this, the detectors must be modelled by “relay” logic equations used in fictitious sensors placed on the axes, conditioned by part present variables to reproduce, as a function of the movement of the parts, the trend diagrams for activating the detectors shown below.



SIMACTEL description

The installation described using SIMACTEL is similar to that previously explained in this section. Because of the size of the parts, the real sensors SQ45A and SQ45B are activated on one axis after the other. These sensors are represented by relays activated by the part present variables and fictitious sensors on the axes.



Equations of the relays representing the sensors

• SQ45A = DQ45A * PPA + FQ45A * PPB
• SQ45B = DQ45B * PPB + FQ45C * PPC

E

4.1 SIMACTEL description

4.1.1 Presentation

The SIMACTEL software environment is supplied with an application example illustrating the various concepts outlined in this manual. This example consists of a description of the “PROSYST model”, a small machining machine usually used as a training support for software based on the AIDIAG card.

This model is composed of three machining stations in a line. Before each station is a conveyor belt on which the part to be machined is placed. These conveyor belts are used to transfer parts from one station to the next, from entry at the beginning of the line to removal at the end.

The parts placed at the entry point are fed onto the first conveyor belt, which in this case is for the drilling machine.

4.1.2 Modelling

Having presented the model, this section explains the modelling process performed using SIMACTEL.

The I/O configuration

Inputs I02,0 to I02,D correspond to the sensors activated by the actuators on the 3 model stations, ie. the drilling machine, the 3-tool turret tool post and the milling station.

Inputs I03,0 to I03,6 take account of information from the operator panels.

Inputs I03,8 to IO3,C inform the PLC of the presence of parts at the various stations and up and downstream of the transfer (entry and exit points).

I04,0 and I04,1 correspond to the “released” and “engaged” positions respectively, of a mechanical device used to lock the turret tool post during operation.

I04,D corresponds to the “left” position, that is drawn-in position of the feeder, and I04,E to the “right” position.

In the input logic equations, terms are used which represent the electrical power supply conditions for the cards.

In the same way, the outputs, distributed over three cards, are used to illuminate the indicator lamps and control the actuators and conveyor belts in the model.

The relay interface

The “supply” comprises, in particular, the relays for switching on the voltage and switching on the power, KM1 and KMPU.

The relays which do not make up part of this “supply” serve as power intermediaries between the outputs and the indicator lamps and the actuator movements.

Axis description

The description of the model includes 11 axes. Axes 1 to 5 correspond to the actuators for the 3 stations. Thus, axis 1 represents the rise/fall of the drilling machine.
This axis has 3 sensors : FCH1 corresponds to the top position of the drilling machine, FCM1 to the middle position and FCB1 to the bottom position. “D1” is the downward movement of the drilling machine, controlled by output O05,8 via Km15. O05,7 controls the upward movement “M1” via Km 14.
Similarly, axis 2 represents the rise/fall of the 3-tool turret tool post.
The rotation of the turret tool post is described by axis 3. This rotary axis activates 2 sensors via different cams.
The sensor “FCA1” is active once per turn, while “FCA2” is active 3 times during one full rotation of the turret tool post.
The description of the milling machining station is made up of 2 axes, axis 4 corresponding to the rise/fall, axis 5 to the translation.
Axis 6 represents the feeder used to supply the parts. As there are only 2 sensors placed at the start and end, this is a “standard” axis.
The next axes, 7 to 10, correspond to the modelling of the conveyor belts which move the parts between stations.
Axis 12 is, like axis 6, a “standard” axis. It models a mechanical lock used to block the turret tool post during machining. This mechanical locking is achieved by prohibiting the “RTRN” rotation movement of the turret tool post if sensor “FCVE” is active.

E

Modelling the conveyor belts

This description is slightly complicated and requires that the manner in which the transfer of parts is modelled using SIMAC representation tools be explained first.

To be modelled, a transfer must be divided into zones which can each only contain one part : the concept of the “section”.
A “section” can be in one of three distinct states. It is “free” if it contains no part, or “full” in the opposite case. The third state corresponds to the availability of the part for the next section.

In SIMAC, a “section” is described by a rotary axis. The free state is represented by a sensor “dxi” at the start of the axis, where “i” is the section index. A sensor “fxi” at the end of the axis corresponds to the availability of the part for the next section.

On the real machine, the sensors are activated by the movement of the parts. In SIMAC, the real sensors are represented by sensors along the axis used to simulate operation.

The presence of a part in section “i” is indicated by a Boolean variable “pi”.

This variable is latched if a part is available in the previous section and if section “i” is free. It is released when the part arrives at the end of section “i”, if the next section is available. The equation for this variable pi is :

pi = [fxi-1 . dxi + pi] . /(fxi . dxi+1) (1)

The movement associated with the axis representing the section describes the evolution of the part in the section. An evolution occurs, if a part is present, and the motor “Mi” causing the part to move (the motor driving the conveyor belt for example) is operating. This movement takes place until the end of the axis is reached. The equation of the movement includes a term :

pi . Mi . /fxi

When fxi is reached and the part moves to the next section, the axis representing section “i” must be set to “dxi”. As this axis is rotary, this is obtained by completing the equation of the movement by the term :

/pi . fxi

The equation of the movement defined in this way is not strictly correct.

Suppose that one part is in section “i” and that motor Mi is not operating : the corresponding axis will remain at sensor “dxi”. If now another part is brought by the previous section, when the axis describing this section reaches “fxi-1”, the part present “pi-1” of this section will be released : the part will then be confused with that already present on section “i”.

To avoid this, the term “pi . dxi” is added in the equation of a movement : as soon as a part enters a section, this term quits the sensor dxi, thus avoiding the loss of a new part which could arrive.

The equation of the movement associated with the axis representing section “i” is thus finally :

If = pi . (dxi + Mi) . /fxi + /pi . fxi (2)

This equation could be refined : for example, you may wish to take account of the fact that, up to a certain position in the section, the part still needs to be driven by the motor of the previous section. In this case, a fictitious sensor would be placed on the axis, defining a zone in which motor Mi-1 is used to move section “i”.

The equations (1) and (2) are applied in the description of the model to describe the conveyor belt for the turret tool post, which is located between the conveyor belt for the drilling machine and that of the milling machine. The conveyor belt for the milling machine is similar. The corresponding axes are axes 8 and 9, and the associated part present variables are pp8 and pp9.

E

The modelling of the conveyor belt for the drilling machine is slightly different : the feeder is upstream of this, and not another conveyor belt.

If equation (2) is still valid to describe the movement of the part on the conveyor belt for the drilling machine, the equation describing the presence of the part must be adapted.

A part is available for the drilling machine conveyor belt when, having been placed at the entrance (the feeder must be in the left position), the feeder has moved it to the right. The feeder may then return to the left position, while the conveyor belt for the drilling machine is not strictly free and has therefore not "taken over responsibility" for the part.

Thus, it can be seen that the availability of a part for the drilling machine conveyor belt is a piece of information which must be memorized : this role is played by the variable "Mem-pp0".

In these conditions, the equation of the presence of a part on the drilling machine conveyor belt is :

pp7 = (dx7 . Mem-pp0 + pp7) . /(fx7 . dx8)

We have seen that the feeder is represented by a "standard" axis. The sensors at either end are "Fcg0" and "Fcd0", corresponding respectively to the left and right positions of the feeder.

The variable corresponding to the presence of a part on the feeder is "pp0". A part may be placed at the entrance if the feeder is in the left position.

E

The latch term of pp0 will be : fcg0 . part, where "part" is an external variable representing the presence of a part at the entry point.

The part will be available for the drilling machine conveyor belt if the feeder has moved it to the right position. This part will then be taken over by the conveyor belt when the axis representing that belt is at "dx7", the equation of the variable Mem-pp0 is thus :

Mem-pp0 = fcd0 . pp0 + Mem-pp0 . /dx7

The variable pp0, representing the presence of a part at the entry point, will also be released, provided that the availability for the drilling machine conveyor belt has been memorized, or that the drilling machine conveyor belt is available when the feeder arrives at the right position.

The equation of this variable is thus :

pp0 = (fcg0 . part + pp0) . /[(fcd0 + Mem-p0) . dx7]

We will now see the removal of the parts. When they arrive at the end of the milling machine conveyor belt, which is the last station on the line, they enter an exit zone. The variable corresponding to the presence of a part at the exit point is PP10, for which the equation is :

pp10 = (dx10 . fx9 + pp10) . /(fx10 . exitpart)

where "exitpart" is an external variable used to remove the parts present at the exit point.

To inform the control system of the presence of a part on one of the conveyor belts, sensors are placed in front of each of the stations.

Activation of these sensors then appears in axes 7, 8 and 9 which represent the evolution of a part on these conveyor belts. The exit zone sensor is placed at the end of axis 10. The sensor corresponding to the presence of a part at the entry point, Fcp0, is described by an equation :

Fcp0 = pp0 + b7 . pp7

pp0 is the variable representing the presence of a part at the entry point. It is then used logically in the expression of fcp0.

b7 is a sensor associated with axis 7, describing the drilling machine conveyor belt. This sensor is active over a certain length at the start of axis 7.

This is due to the operation of the sensor indicating the presence of a part at the entry point : this sensor remains active while the part is already on the drilling machine conveyor belt, from where the role of the term b7 . pp7 in the expression of fcp0 is obtained.

Note

> The description is completed by axis 11 used to loop the parts moving from the exit to the entry point.
> The loop is made when the external variable “Return” is at logic state “1”.

The operator panel

The operator panel is used to control this model when it is controlled by a PLC : it comprises the buttons for switching on the voltage, switching on the power, and starting the cycle.

There is also a fictitious pushbutton used to introduce parts at the machine entry point. The “Return part” switch, when it is in the closed position, is used to loop parts via axis 11.

The Grafcet charts

The file MGRAF1 contains Grafcet charts used to animate the model in local mode.
The first 3 Grafcet charts each control one of the stations.
The 4 other Grafcet charts manage supply of parts and the 3 conveyor belts.

Note

> These Grafcet charts known as “Mechanical” charts do not manage the operating modes.
> During internal simulation, simply introduce a part at the entry point via the “part” pushbutton on the operator panel to activate the model.

4.2 Electrical and mechanical dossier

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

4.3 Print dossier

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

E

1.1 Presentation

During simulation in online mode, we have seen that it is necessary to use a simulation OFB for exchanges between SIMACTEL and the PLC. This OFB is then added, under PL7-3, to the PLC application before transfer to the PLC. However, there is no need to modify the program itself to be able to use the simulation OFB. Thus, the simulation can reproduce the real operating conditions of this program.

It should be noted, however, that the simulation OFB has a considerable impact on the whole PLC application. In fact, it represents a memory size of about 13 Kbytes which is added to the total application size.

It also affects the PLC scan time. Here, the impact of the simulation OFB is in fact limited, particularly for in-rack I/O configurations taking account of the underlying principle : the OFB inhibits exchanges on the backplane bus so that there is no update of the I/O image memory in the PLC. In fact, it is the OFB which copies the detected changes in the I/O to this memory. Consequently, all the processor time used for I/O exchanges on the bus is recovered, which makes up for the simulation OFB calculation time.

However, the impact of the OFB is slightly greater for remote I/O configurations since, in this case, there is no inhibition of exchanges on the TBX. It is therefore necessary, on each scan, to copy the OFB I/O image memory to the PLC image memory. The greater the number of remote I/O, the more time consuming this operation becomes.

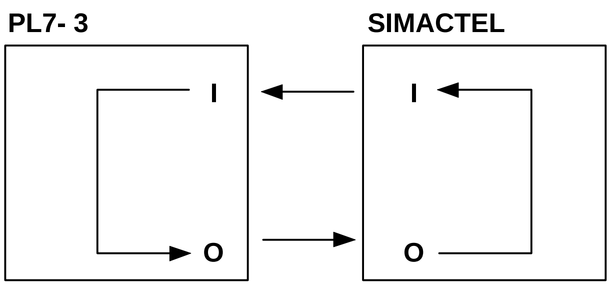
The graphs on the following pages give the figures relating to the impact of the simulation OFB on the PLC scan time.

1.2 Impact of the simulation OFB on the PLC scan time

1.2.1 In-rack I/O

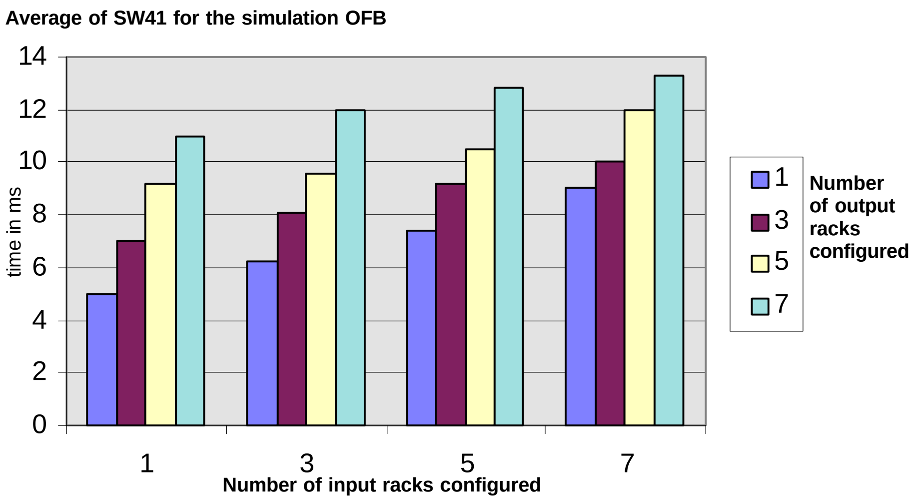
There are three different types of measurement :

• Systematic copy of the I/O in ascending order for all the I/O configured



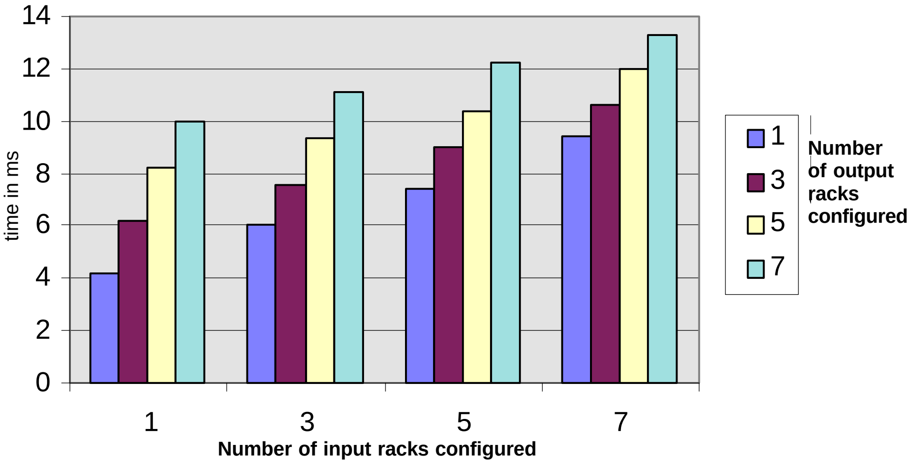
All the I/O are configured in the MAST task.

Measurement principle : comparison of scan time SW41 with and without the simulation OFB. SY48 is set to 1 to enable the update of SW41. Note that the scan time measured by SW41 represents the real execution time of the PLC scan and not the time configured during creation of the application.
The results obtained are as follows :

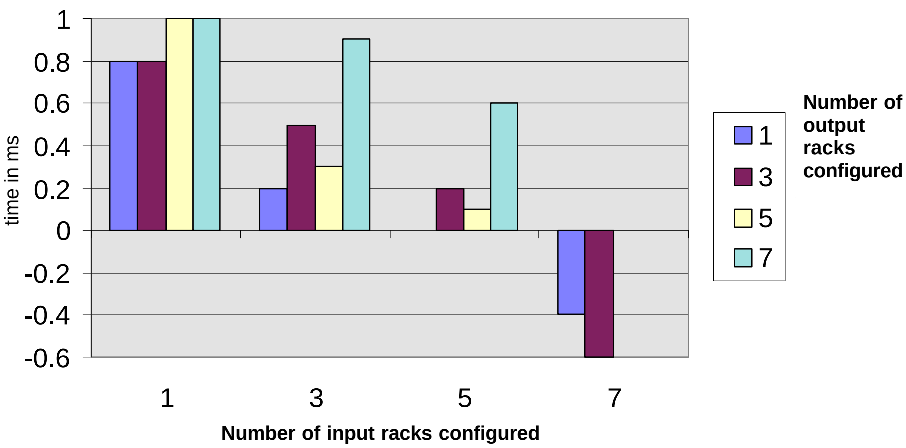


F

Actual average of SW41



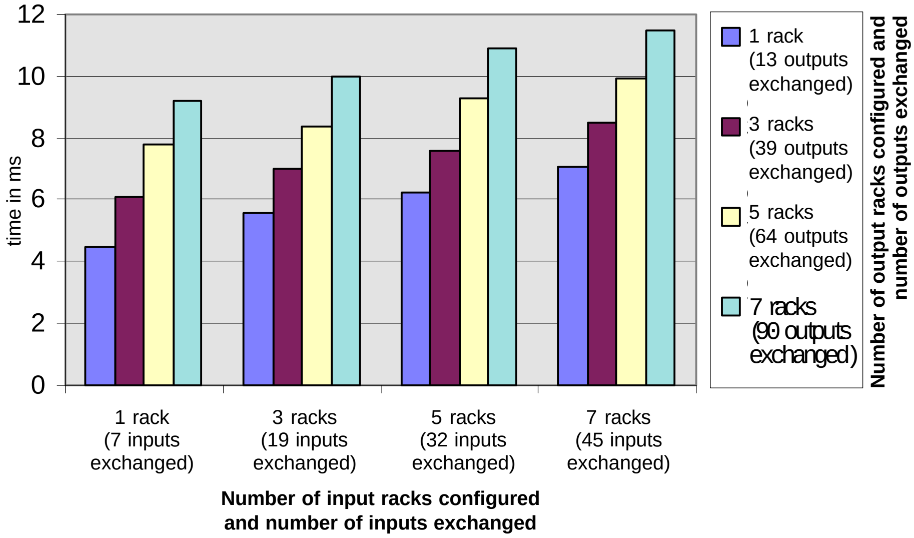
Difference between SW41 with the OFB and the actual SW41



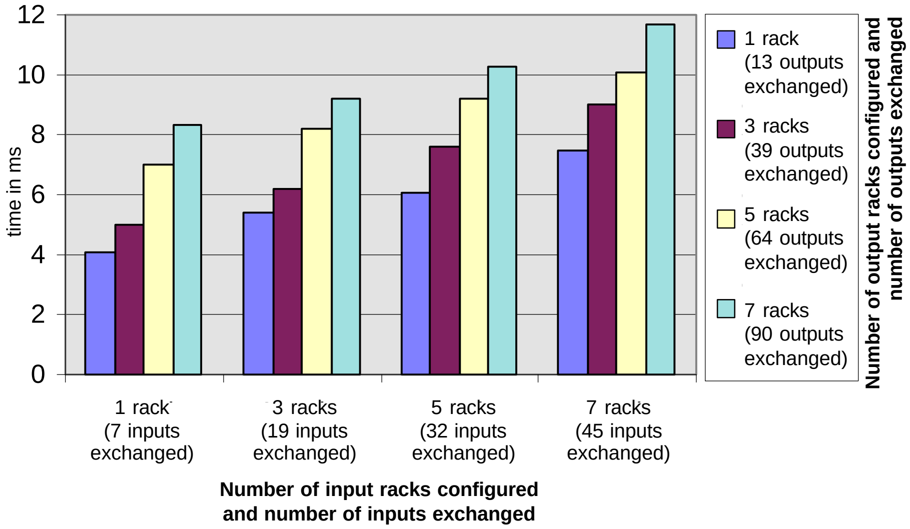
F

• To represent the real context of simulation by events more closely, 5% of the configured inputs and 10% of the configured outputs are copied systematically. The measurement principle remains the same and the results are as follows :

Average SW41 with the simulation OFB

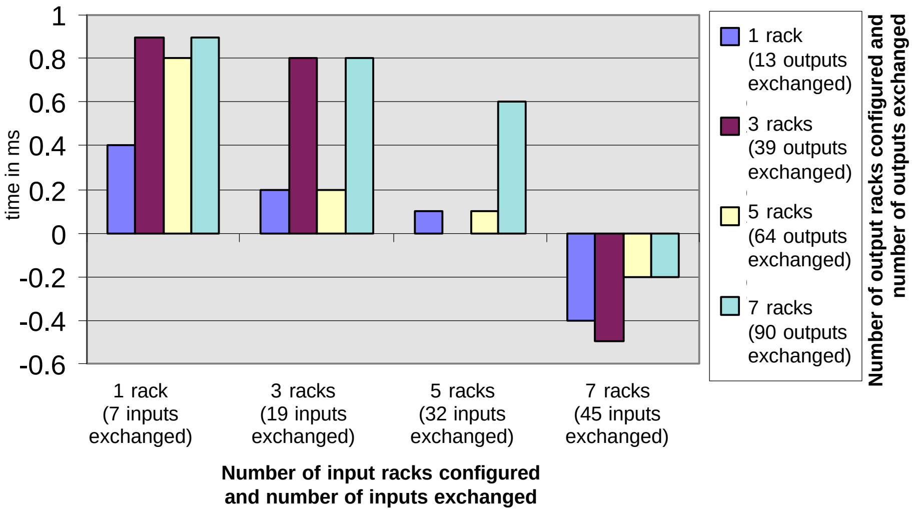


Actual average SW41



F

Difference between SW41 with the OFB and actual SW41



- Based on the Model application (see section 4 in part E).
  Measurement principle : measurement of the time for copying an input to an output via the PLC with and without the simulation OFB. This measurement has been made via a terminal used to monitor the I/O exchanges between SIMACTEL and the PLC.

The results obtained are as follows :

- processing time without the OFB : 9;9;11;12;9;8;10;9;9;8;6;8;6;8;12;11;6;6;8;10;6; for an average time of 8.6ms

- processing time with the OFB : 8;7;7;6;10;7;6;8;9;6;7;6;6;10;5;6;7;6;9;6; for an average time of 7.1 ms

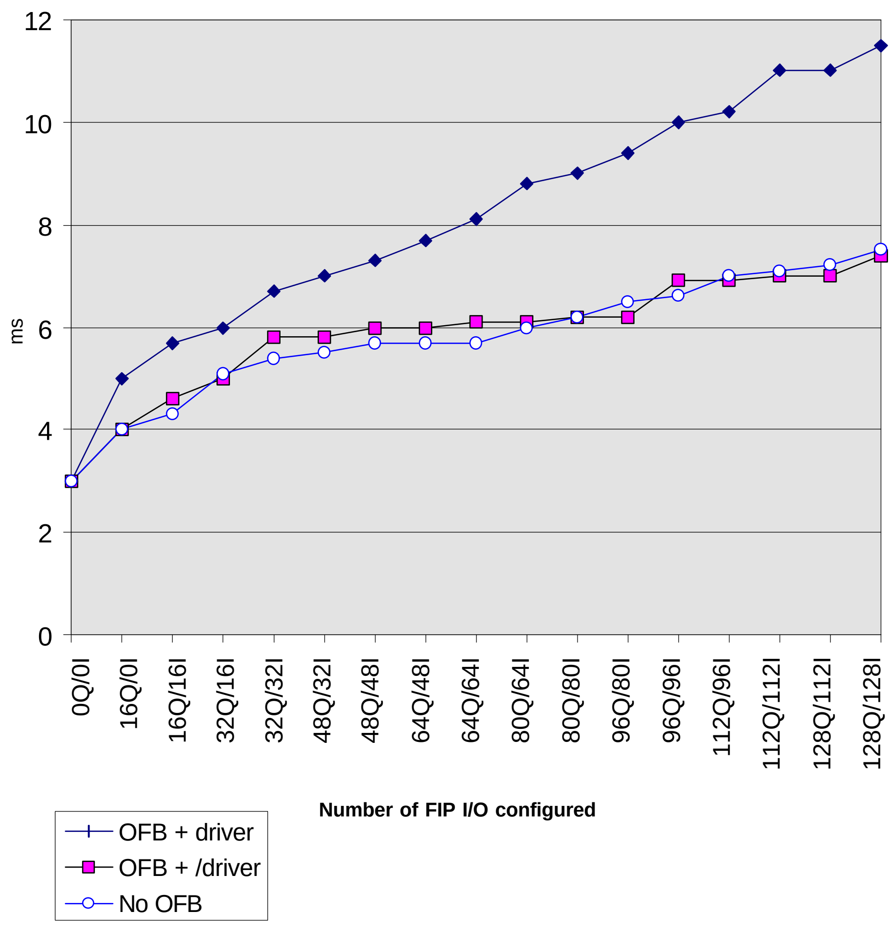
F

1.2.2 Remote I/O

The measurement made consists of gradually increasing the number of remote I/O configured : 16 channels are added at one time alternating between outputs and inputs. The first 16 channels are outputs. All the I/O are configured in the MAST task. Without the simulation OFB, only the PLC updates the remote I/O in its I/O image memory. However, when the OFB is active, it copies, on each scan, its own I/O image memory into that of the PLC just after this has been updated by the PLC system. By measuring the scan time with and without the OFB, the impact of this on the execution time can be deduced.

Measurement principle : read the value of SW41 with and without the simulation OFB. The results are as follows :

Average SW41 with and without the simulation OFB



F

1.3 Impact of the medium and tools used

The time required for a full simulation cycle includes the OFB execution time, the time taken to calculate the event prediction by SIMACTEL, and the UNI-TE exchanges between the communication server and the simulation OFB and also the SIMACTEL-Server exchanges. Therefore, we have tested simulation on the Model used in part E (Examples) using various media : Terminal port and UNI-TELWAY. To do this, the average time it takes for a part to travel in the model application has been measured, with or without PL7-3 debug.

The results are as follows :

- Without PL7-3 debug mode

  Terminal port : average travel time for one part is 42.2 seconds

  UNI-TELWAY : average travel time for one part is 40.5 seconds

- With PL7-3 debug mode

  Terminal port : average travel time for one part is 45 seconds

  UNI-TELWAY : average travel time for one part is 41.5 seconds

F

F

2.1 Messages linked with the environment

These messages concern the availability of resources required for the correct operation of SIMACTEL.

“WINDOWS resource problem - close windows”

WINDOWS windows are required to display the operator panels and views, particularly because of the number of elements represented in the views. When a request is made to display operator panels or views, a test of the available resources is made. If resources are low - this can be checked in the “About” window in Program Manager - this message is displayed. You should then close one or more of the views displayed on screen.

“Copy (or Save or Load) failed”

These messages may appear when using tools for copying a machine or model, transferring a machine to a model and saving or loading a machine or model to - or from - a floppy disk. The most likely cause of this is a lack of space available on the hard disk or the floppy disk during a save operation. The disk format may also be incorrect.

2.2 Messages linked with modelling

These messages may appear during online or offline simulation.

Their origin stems from an anomaly in the modelling process, preventing the continuation of simulation, which is then interrupted.

“Unknown state of variables”

This message appears when SIMACTEL is not in a position to determine the state of the description variables or relays. A simple example is the following equation : a = /a

This phenomenon appears, for example, if an R/S switch is described in SIMACTEL using two logic equations and they activate the Set and Reset inputs at the same time!

This anomaly can be resolved by correcting the expressions of the variables in question or by reestablishing a stable state using one of the terms in the equations.

“Simultaneous activities for axis : ..”

This message signals an anomaly due to the non exclusive nature of movement equations associated with a single axis. In this case, you need to modify the movement equations concerned so that they differ.

“Infinite evolution of : ”

This message appears when a loop occurs in a rotary axis changing at an infinite speed. The problem is that the equation is never invalidated during the movement. This equation needs to be corrected so that the movement to which an infinite speed has been given can be deactivated by the change in state of one of the sensors in the axis.

“Chart instability”

This message signals that a “stationary situation” appears in a Grafcet chart. This means that a closed cycle of transitions for which the conditions are simultaneously true has occurred. Some transitions in the loop therefore need to be locked.

Another possibility is that there is a counter overflow : this concerns an anomaly in an arithmetic operation - which should be replaced by an incrementation - made in a step.

2.3 Messages linked with PLC exchanges

The various error messages linked with exchanges between SIMACTEL and the PLC and their causes are explained below.

“Cannot establish link”

SIMACTEL has not succeeded in launching the communication server.

“CONNECTION FAILED”

If this message is not accompanied by other messages, this means that :
- the physical connection has not been established correctly
- the simulation OFB is missing in the PLC application
- the PLC is switched off

If this message is accompanied by the message “CPU NOT COMPATIBLE”, this means that :

- the version of the PLC CPU is not V5

If this message is accompanied by the message “SIMACTEL CONFIGURATION<>PLC CONFIGURATION”, this means that :
- the SIMACTEL configuration does not correspond to the configuration described in the PLC program.

F

2.4 Messages linked with printing

The various error messages linked with printing and their causes are explained below.

"Cannot open the machine dossier"

SIMACTEL has not succeeded in opening the machine dossier requested. This message appears when the software cannot read the information contained in the machine description files. If this message appears, contact your technical support center.

"Cannot open the model"

SIMACTEL has not succeeded in opening the model requested. This message appears when the software cannot read the information contained in the model description files. If this message appears, contact your technical support center.

"Incompatibility between the print format, the machine dossier and the printer characteristics. SIMPRINT cannot print your machine dossier ! MODIFY YOUR PRINT FORMAT..."

This message appears when SIMACTEL cannot print the document with the print format requested. This happens when one of the sections in the document is too large to be printed on a single page. To correct this problem, select a smaller character font so that all sections can fit on one page.

"Insufficient memory! Close applications to free some memory"

When too many WINDOWS applications are open at the same time, the computer does not have enough free memory to execute SIMACTEL correctly. Close as many applications as possible to free some memory when this message appears.

"Another instance of SIMPRINT is already open! Close this other instance first before using the print tool"

This message appears when the print tool is opened when the SIMPRINT application is already active. To correct this problem, close the active SIMPRINT application, then use the new print tool.

"Cannot print"

This message appears when a WINDOWS print error occurs. To correct this problem, refer to the WINDOWS user manual.

"File *.dib is too large to be opened by SIMPRINT !" and "Cannot open file *.dib!"

These messages appear when SIMACTEL encounters problems printing operator panels and views. Contact your technical support center to correct this problem.

F