

- 1 Presentation
 - 2 Installation
 - 3 Using PL7-3 software
-

A

- 4 Operation with the terminal online
 - 5 Configuration mode
-

B

- 6 Program mode : Grafcet language
-

C

- 7 Editor : Ladder language
-

D

- 8 Program mode : Literal language
-

E

- 9 Constant mode
 - 10 Data mode
 - 11 Debug mode
-

F

- 12 Transfer mode
 - 13 Document/Print mode
 - 14 Search/Replace mode
-

G

- 15 Appendix
-

H

Section	Page
1 Presentation	A1/1
1.1 General	A1/2
1.2 Configuration required for PL7-3 software	A1/6
2 Installation	A2/1
2.1 Hardware check	A2/2
2.2 Connections	A2/4
2.3 Software installation	A2/8
2.4 PL7-3 files	A2/10
3 Using PL7-3 software	A3/1
3.1 Accessing PL7-3 software	A3/2
3.2 Defining the main window	A3/4
3.3 Presentation of operating modes	A3/10
3.4 Backing up and restoring PL7-3 applications	A3/12
3.5 Using the symbol database	A3/17
3.6 Quitting the PL7-3 function	A3/19
3.7 PL7-3V6 keyboard shortcuts	A3/19
3.8 Using notepads	A3/23
3.9 Using menus - General layout	A3/23
3.10 Using "user-definable lists"	A3/24
3.11 Rule for entering data in the PL7-3 V6 entry zone	A3/25

Section	Page
4 Operation with terminal online	B4/1
4.1 Physical and logical connections	B4/2
4.2 Reservation of a connected PLC	B4/3
4.3 Principles of terminal to PLC connection	B4/5
4.4 Operation in online mode, connected locally	B4/6
4.5 Operation in online mode, connected via a network	B4/6
4.6 Repeating the logical connection request	B4/7
5 Configuration mode	B5/1
5.1 Presentation of CONFIGURATION mode	B5/2
5.2 Configuring the application	B5/4
5.3 Configuring tasks	B5/6
5.4 Configuring Grafcet	B5/7
5.5 Configuring the user memory	B5/9
5.6 Configuring bit, word and function block objects	B5/14
5.7 Configuring the optional function blocks	B5/18
5.8 Configuring the network services	B5/20
5.9 Configuring the link	B5/21
5.10 Modifying the configuration in online mode	B5/22
5.11 Reconfiguration function	B5/23

Section	Page
6 Program mode : Grafcet language	C6/1
6.1 Changes made since V52	C6/2
6.2 Selecting the task and module	C6/3
6.3 Reading the program	C6/6
6.4 Grafcet zoom function	C6/12
6.5 Creating or modifying the program	C6/15
6.6 Multiple selection	C6/27
7 Editor : Ladder language	D7/1
7.1 Changes made since V52	D7/2
7.2 Selecting the task and module	D7/3
7.3 Reading the program	D7/5
7.4 Ladder language zoom function	D7/12
7.5 Creating or modifying the program	D7/14
7.6 Multiple selection of objects	D7/27
7.7 Searching for the program addresses of a variable	D7/31
8 Program mode : Literal language	E8/1
8.1 Changes made since V52	E8/2
8.2 Selecting the task and the module	E8/3
8.3 Reading the program	E8/5
8.4 Literal language zoom function	E8/11

Section		Page
	8.5 Creating or modifying the program	E8/13
	8.6 Searching for the program addresses of a variable	E8/24
9	Constant mode	F9/1
	9.1 Presentation of CONSTANT mode	F9/2
	9.2 Constant words	F9/3
	9.3 Internal OFB constants	F9/6
	9.4 Managing Grafcet situations	F9/10
10	Data mode	F10/1
	10.1 Presentation of DATA mode	F10/2
	10.2 Description of the screen and keys	F10/6
	10.3 Reading data	F10/11
	10.4 Writing and forcing data	F10/15
11	Debug mode	F11/1
	11.1 Changes made since V52	F11/2
	11.2 Presentation of DEBUG mode	F11/3
	11.3 Predefining screens	F11/11
	11.4 Displaying predefined screens in realtime	F11/17
	11.5 Defining a breakpoint	F11/22
	11.6 Executing the program	F11/24
	11.7 Accessing other modes	F11/29

Section	Page
11.8 Searching for the program addresses of a variable	F11/31
12 Transfer mode	G12/1
12.1 Presentation of TRANSFER mode	G12/2
12.2 Processor - terminal memory transfers	12/3
12.3 Save/Restore : internal words	12/7
13 Document/Print mode	G13/1
13.1 Presentation of DOCUMENT/PRINT mode	13/2
13.2 Composition of the documentation file	13/5
13.3 Printing	13/14
13.4 Description of the listings	13/21
14 Search/Replace mode	G14/1
14.1 Presentation of SEARCH/REPLACE mode	G14/2
14.2 Search function	G14/4
14.3 Replace function	G14/6
14.4 Diagnostics function	G14/8
14.5 Function for moving "FIP" equipment	G14/10
14.6 Linking searches	G14/12
14.7 Example of a listing	G14/14

Section	Page
15 Appendix	H15/1
15.1 Memory allocation map	H15/2
15.2 Entering the application comment	H15/3
15.3 Selecting a processor	H15/4
15.4 Saving and retrieving files	H15/5
15.5 Search for the program addresses of a variable using <Alt><X>	H15/6
15.6 Possibilities open to operators depending on their rights of access	H15/8
15.7 Converting a V4 application to V5	H15/9
15.8 PL7-3 Help	H15/11
15.9 V5 Modular version	H15/13
15.10 List of symbols	H15/23

About this document

PL7-3 V6 is an upgrade of the V52 program, and is marketed as 2 separate products :

- XTEL PACK V52,
- XTEL PACK V6.

This document is common to both the above-mentioned products and incorporates PL7-3 versions 3, 4 and 5, since PL7-3 V6 can be used to create and modify applications for levels V2, V3, V4 (V41, V42 and V43) and V5 (V5. and V52). V6 applications do not exist.

- on a V5 station, the XTEL-CONF tool is used to configure rack I/O, remote I/O and tasks and to generate the application (.APP). PL7-3 is the means of configuring the other software resources (objects, OFBs, etc),
- on a V4 station, the XTEL-MEM tool is used to configure the type of processor and the memory capacity, divide the memory between the various station functions and generate the global application (.APP). PL7-3 is the means of configuring the other hardware and software resources (rack I/O, objects, OFBs, etc),
- on a V2 or V3 station, PL7-3 governs the whole application.

New features of PL7-3 V6 in relation to previous versions :

- Completely revised user interface for TSX V5 and PMX V5 XTEL type stations (OS/2 PM user interface which conforms to the accompanying standard IBM CUA 93).
- Programming of COM words for in-rack FIPWAY modules (TSX FPM 100).
- For TSXV4 / PMXV4 / TSX V3 type XTEL stations, PL7-3 V6 offers exactly the same functions and user interface as PL7-3 V52.

PROGRAM mode

This mode can be accessed by pressing **<ALT>+<F1>** from any screen. It is permanently available, even with an editor open (with certain exceptions).

Editing the program

- General rules of equivalence between PL7-3 V52 and PL7-3 V6 editor keys.

Functions	V52	V6
Editors		
Entry	(Program Address Selection)	(Program Address Selection)
Exit : CLEAR QUIT	<PG UP> <END>	<CTRL>+<PG UP> (Edit menu, quit)
Entry to modify mode	<F4>	<F4> or <CTRL>+<M>
Exit modif : Cancel OK	<PG UP> <ENTER>	<ESC> or <CTRL>+<PG UP>* <ENTER> or <CTRL>+<C>
Grafcet Zoom		
Entry	<PG DOWN>	<CTRL>+<PG DOWN> or <ENTER> or double-click
Exit	<PG UP>	<CTRL>+<PG UP> or <ESC> or <ENTER>
Zoom In/Out : Entry	<PG DOWN>	<CTRL>+<PG DOWN> or <SPACE>
Exit : CLEAR	<PG UP>	<CTRL>+<PG UP>
Ladder, Literal Zoom		
Zoom on block : Entry	<PG DOWN>	<CTRL>+<PG DOWN> or <SPACE> or double-click
Exit : CLEAR QUIT	<PG UP> <END>	Button : OK or Cancel Button : OK or Cancel
Zoom on call/jump : Entry	<PG DOWN>	<CTRL>+<PG DOWN> or double-click
Exit : CLEAR QUIT	<PG UP> <END>	<CTRL>+<PG UP> (Edit menu, quit)

- In LITERAL (STRUCTURED TEXT) or LADDER language when a statement or a rung is being modified, it is possible to access the rest of the module.
-

Literal editor

To access the Literal editor using the keyboard :

- activate PROGRAM BROWSER by pressing **<ALT>+<F1>**,
- using the **<TAB>** key, move the cursor to the Task column,
- select a task using the up/down arrow keys,
- select a module, then a label, in the same way,
- move the cursor to the Task column and confirm selection of the program address by pressing **<ENTER>**.

To access the Literal editor using the mouse :

- activate PROGRAM BROWSER by clicking on the Program icon in the toolbar in the MAIN WINDOW,
 - select the program address by clicking directly on the task, module and label required,
 - confirm the selection by double-clicking on the chosen task.
- To enter, modify or delete a label or a comment, click in the label or comment zone (or press the **<←>** key to access the label zone directly. Example : L10 : **←** SET B0), and make a direct entry and confirm by pressing **<ENTER>** or by clicking on the confirmation icon in the toolbar.
 - To enter, modify or delete a new statement, position the cursor at the beginning of the statement concerned. A square-shaped cursor shows that you are still in DISPLAY mode. To change to MODIFY mode press **<SPACE>** ; the cursor then becomes a line. To confirm the statement press **<ENTER>** : the cursor becomes square-shaped again.
 - It is possible to copy, cut and paste between different modules.
 - The mouse can be used for a multiple selection of statements : to do this click on the statements to be grouped while holding the **<CTRL>** key down.
 - A key word completion function simplifies entry of instructions : in MODIFY mode, type the beginning of the key word and then press **<SPACE>** to call the completion function.

Ladder Editor

To access the Ladder editor using the keyboard :

- activate PROGRAM BROWSER by pressing <ALT>+<F1> ,
- using the <TAB> key, move the cursor to the Task column,
- select a task using the up/down arrow keys,
- select a module, then a label, in the same way,
- move the cursor to the Task column and confirm selection of the program address by pressing <ENTER>.

To access the Ladder editor using the mouse :

- activate PROGRAM BROWSER by clicking on the Program icon in the toolbar in the MAIN WINDOW,
- select the program address by clicking directly on the task, module and label required,
- confirm the selection by double-clicking on the chosen task.
- To enter, modify or delete a label or a comment, double-click in the label or comment zone, make the entry and confirm by pressing <ENTER> or by clicking on the confirmation icon of the toolbar.
- It is possible to cut, copy, paste and drag (drag and drop) all or part of a rung.
- A paste operation can be performed within the same rung or in another rung of another module.
- It is possible to cancel or redo edit operations (Undo, Redo).
- It is possible to symbolize objects without quitting the editor.

Grafcet Editor

To access the Grafcet editor using the keyboard :

- activate PROGRAM BROWSER by pressing <ALT>+<F1> ,
- using the <TAB> key, move the cursor to the Task column,
- select a task with the up/down arrow keys,
- select a module, then a page, in the same way,
- move the cursor to the Task column and confirm selection of the program address by pressing <ENTER>.

To access the Grafcet editor using the mouse :

- activate PROGRAM BROWSER by clicking on the Program icon in the toolbar in the MAIN WINDOW,
- select the program address by clicking directly on the task, module and page required,
- confirm the selection by double-clicking on the chosen task.
- To modify a page, press <F4> or use the **Modify Page** function in the **Edit** menu, or click on the modification icon.

- To enter ZOOM mode : press **<SPACE>**, or **<ENTER>**, or **<CTRL>+<PAGE DOWN>** or click on the zoom icon, or use the **Grafcet Zoom** function from the **Edit** menu, or double-click in the editor.
- To zoom in on an action/transition condition, or a step :
 - enter ZOOM mode (see above),
 - position the cursor on the object concerned, by clicking once on it or using the arrow keys,
 - zoom in on this object by pressing **<SPACE>**.

Use these new functions wherever possible : they will save you time. Use the **Undo** function in each editor as necessary, to cancel your modifications.

DEBUG mode

- This mode can now be accessed directly from the language editors.
- Predefined screens are defined at PROGRAM BROWSER level, which is accessed by pressing **<ALT>+<F1>** :
 - **Add** button from the PROGRAM ADDRESS SELECTION page for program screens,
 - **Add** button from the MODULE SELECTION page for the vignettes,
 - **Data** button from the DEBUG page for data screens.
- Six screens are accessed from the **Predefined Screens** tab of the PROGRAM BROWSER, by selecting screens from the list using the arrow keys or the mouse.
- Predefined screens can be accessed quickly using the **<ALT>+<1>** to **<ALT>+<6>** keys. No screen is predefined by default.
- To debug a Literal program :
 - an inserted breakpoint is displayed in a green zone in the editor,
 - in step by step operation the next statement to be executed is displayed in a yellow zone,
 - the animation of a Literal statement is now easier to read : half a screen is now reserved for displaying the original code of the statement.

DATA mode

- This mode is accessed directly from the MAIN WINDOW, the editors, or the predefined screens.

CONTROL, ADJUST and V52 TERMINAL modes

The functions of these modes are integrated in other modes

- The CONTROL mode functions are integrated in the PL7-3 MAIN WINDOW menus.
- The ADJUST mode functions are incorporated in DEBUG mode.
- The TERMINAL mode is integrated in DOCUMENT/PRINT mode (Parameters tab).

Software compatibility

For a V5 application, on a V5 station, PL7-3 V6 **is compatible** with the following software :

X-TEL V5
MINI X-TEL V5
XTEL-MOD V5
XTEL PACK V52 and XTEL PACK V6

For a V2, V3 or V4 application, on a V3 or V4 station, PL7-3 V6 **is compatible** with the following software :

XTEL-MOD V41
XTEL-MOD V42

However, **it is not compatible**, because of direct access to XTEL-SDBASE, with the following software :

X-TEL V42
MINI X-TEL V43

The **coexistence** of several versions of PL7-3 on the same station is **impossible**.

If PL7-3 software is re-installed on a station, the version already installed on the station can no longer be accessed. The application files present on the station are of course saved.

Compatibility between applications and PL7-3 V6

PL7-3 V6 does not produce V6 applications. Its compatibility with applications is the same as that of PL7-3 V52.

General note for this document

- The term **program element** is used generically to refer to either a Ladder rung or a Literal statement
- PLC stands for Programmable Logic Controller (TSX or PMX).

Section	Page
1.1 General	1/2
1.1-1 Functions offered by PL7-3 software	1/3
1.2 Configuration required for PL7-3 software	1/6

1.1 General

This manual describes the installation and use of PL7-3 V6 software on PC compatible microcomputers.

This software can be used to program and operate the following PLCs :

- TSX 67-30/87-10/87-20,
- TSX 47-30/67-20/87-30,
- TSX 47-4xx/67-4xx/87-4xx/107-4xx,
- PMX 47-4xx/67-4xx/87-4xx/107-4xx.

The following version of the software is available :

- **PL7-3 programming and operation** : TXT L PL7 3 V6E

This version can be used to program, debug and operate the above-mentioned sequential PLCs, offline or online, connected locally or via a network. It offers Grafset, Ladder and Literal (Structured Text) programming languages and can accept optional function blocks (OFBs) which can be used by TSX and PMX 47-4xx/67-4xx/87-4xx/107-4xx PLCs.

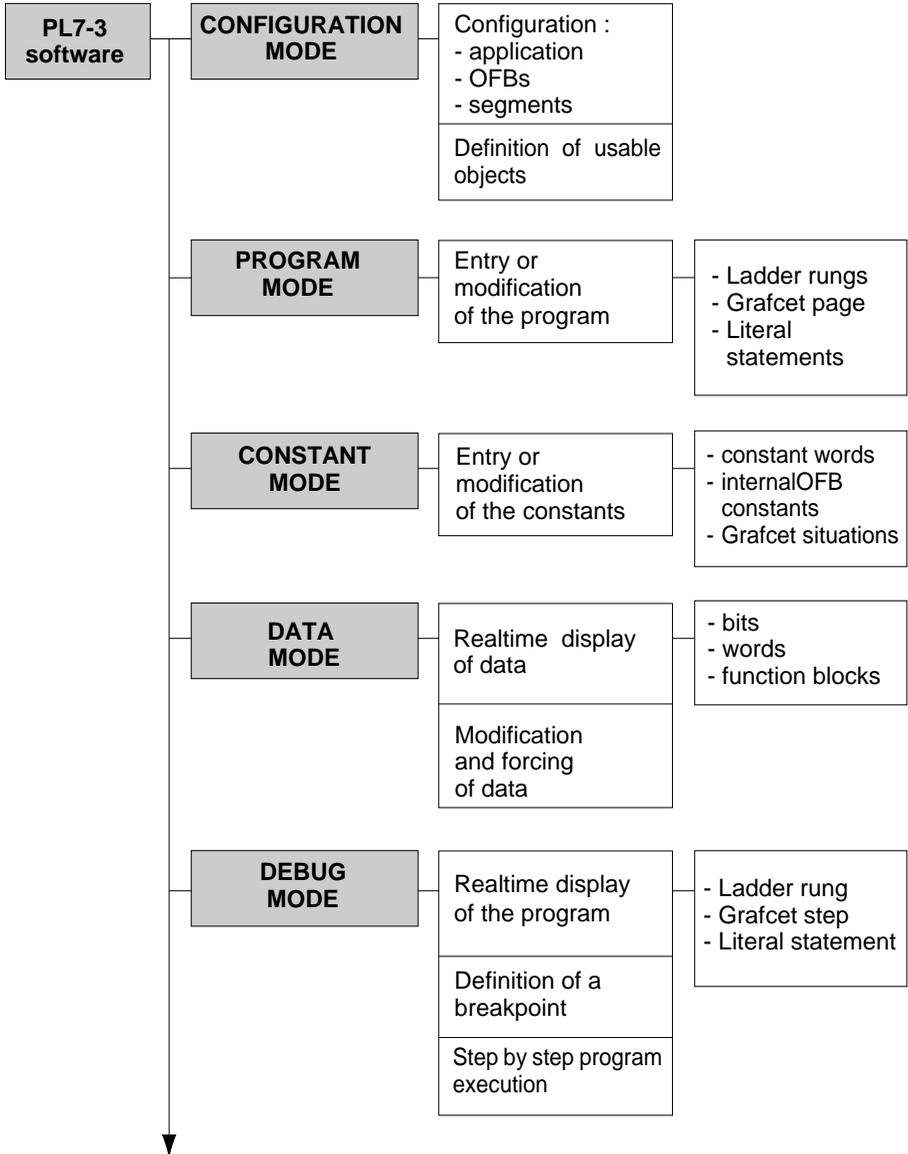
1.1-1 Functions offered by PL7-3 software

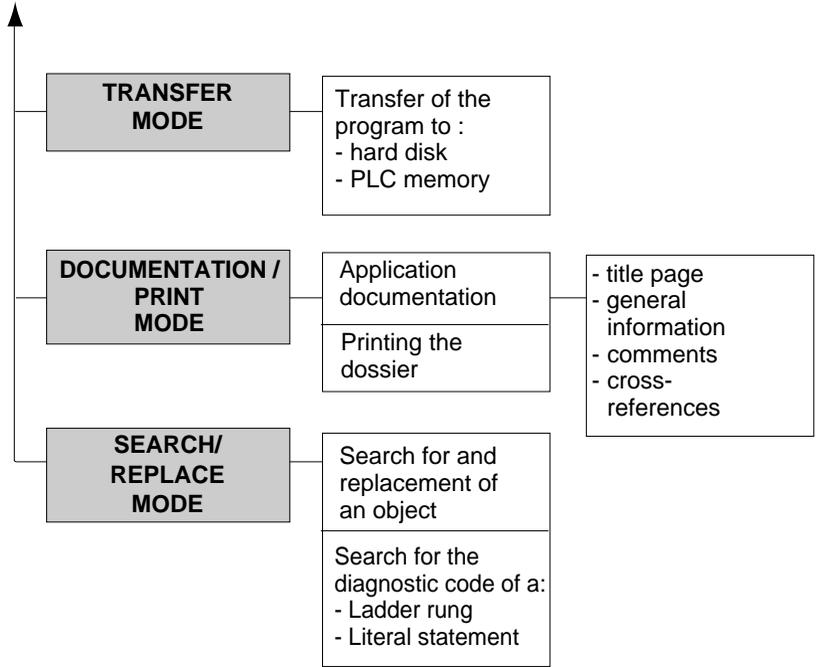
PL7-3 software offers the following services :

- entry or modification of the PL7-3 program : Grafcet, Ladder or Literal language,
- storage of the program on the hard disk,
- transfer of the program between the terminal and the PLC (via the hard disk),
- search for and replacement of an object,
- documentation of the program and printout of the documentation file, including :
 - title page,
 - general information pages,
 - program comments,
 - network information,
 - cross-references.
- storage of the documentation file on the hard disk.

In online mode, connected locally or via a network, it can also be used for :

- programming and program modification in online mode and with the PLC stopped or running,
- debugging the program,
- adjusting the application parameters.





1.2 Configuration required for PL7-3 software

To use PL7-3 software, you need a terminal equipped with :

- OS/2 V1.3, V2.1 or WARP 3.0 Red or WARP 3.0 Blue operating system,
- X-TEL software workshop.

You therefore need one of the following hardware configurations :

- Recommended configuration :

Programming terminal	RAM memory	Disk space available on hard disk	Microprocessor
T FTX 517/507 xx	12 Mb	10 Mb	80486 DX
T FTX 417 40	12 Mb	10 Mb	80486 DX
PC compatible* microcomputer	12 Mb	10 Mb	80486 DX

(*) with a high definition EGA or VGA color screen.

- Minimum configuration :

Programming terminal	RAM memory	Disk space available on hard disk	Microprocessor
T FTX xxx xx	8 Mb	8 Mb	80286
PC compatible microcomputer	8 Mb	8 Mb	80286

This minimum configuration can be used to launch only PL7-3 V6 at a given time.

Note

Do not forget to leave enough disk space for your operating system "swapper" (20 Mb recommended).

Section		Page
2.1	Hardware check	2/2
2.2	Connections	2/4
2.2-1	Installing the software key	2/4
2.2-2	Connection to the mains	2/4
2.2-3	FTX 417 → PLC connection	2/5
2.2-4	FTX 507/517 → PLC connection	2/5
2.2-5	PC compatible → PLC connection	2/6
2.3	Software installation	2/8
2.3-1	Preliminary operations	2/8
2.3-2	Installation procedure	2/8
2.4	PL7-3 files	2/10

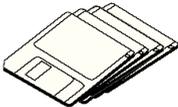
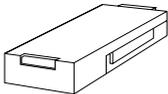
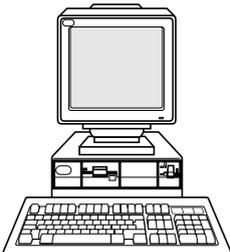
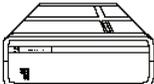
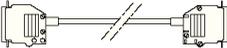
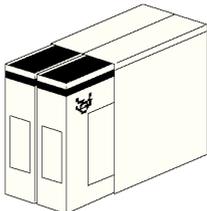
2.1 Hardware check

To use PL7-3 software, the following hardware is required :

- an FTX 417 terminal, an FTX 507/517 terminal or a PC compatible microcomputer (see required configuration, section 1.2), equipped with the OS/2 operating system (Version 1.3 and above or WARP 3.0) and the X-TEL software workshop (Version 52 and above),
- a terminal / PLC connecting cable, for an FTX 417 or FTX 507/517 terminal,
- a terminal / PLC connection kit, for a PC compatible, composed of :
 - an RS 232C / current loop converter,
 - a converter / microcomputer connecting cable with a 9-pin connector,
 - a converter / microcomputer connecting cable with a 25-pin connector,
 - a converter / PLC connecting cable,
- an XTEL PACK V52 (for OS/2 >= V1.3) or XTEL PACK V6 (for OS/2 >= V2.1 or WARP 3.0) software package.

Note

PL7-3 V6 software used with XTEL PACK V52 requires that FIX 2.1 be installed under OS/2 V2.1.

Terminal	Terminal/PLC connection kit	Software package
<p>FTX 417</p> 	<p>T FTX CB 7 4</p> 	<p>XTEL PACK diskettes</p> 
<p>FTX 507/517</p> 	<p>T FTX CB 7 5</p> 	<p>Software key</p> 
<p>ASA approved PC compatible microcomputer</p> 	<p>TSX TAC 03</p> <p>TSX CTC 00 converter/PLC cable</p>  <p>TSX TE 01 RS232/CL converter</p>  <p>TSX CTC 01 converter/microcomputer cable</p>  <p>TSX CTC 02 converter/microcomputer cable</p>  <p>TSX SCC 02 software key support</p> 	<p>Documentation TXT DR PL7 3 V5E TXT DM PL7 3 V5E TXT DM FBD V4E</p> 

2.2 Connections

The connections specific to the terminal (monitor, keyboard, mouse, printer) are described in the manufacturer's documentation. This section therefore only describes the installation of the software key and the following connections :

- mains,
- terminal to PLC.

2.2-1 Installing the software key

Position the software key in the empty slot in the key support.

This operation must be performed with the power off.

Note

This software key contains the access right necessary to use PL7-3 software.

The Key Manager tool supplied with the software workshop can be used to transfer this right to the master key. This makes the PL7-3 key unusable and thus frees the corresponding slot on the key support.

2.2-2 Connection to the mains

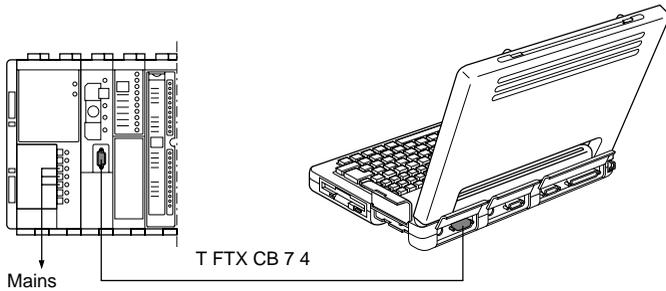
Before connecting to the mains, check that each of the devices to be connected is correctly adjusted to operate on 220V or 110V as appropriate.

Then, observing the precautions indicated in the manufacturer's documentation, connect the various devices to the mains :

- terminal,
- RS232/current loop converter,
- PLC.

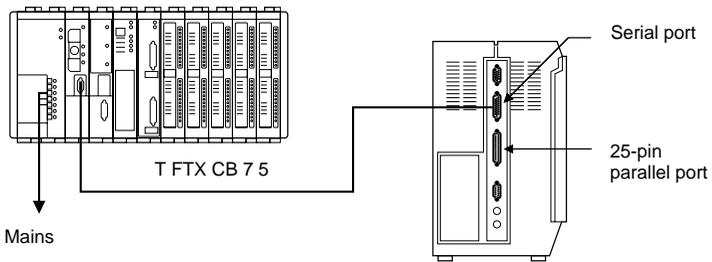
2.2-3 FTX 417 → PLC connection

This connection requires the T FTX CB 7 4 connecting cable



2.2-4 FTX 507/517 → PLC connection

This connection requires the T FTX CB 7 5 connecting cable



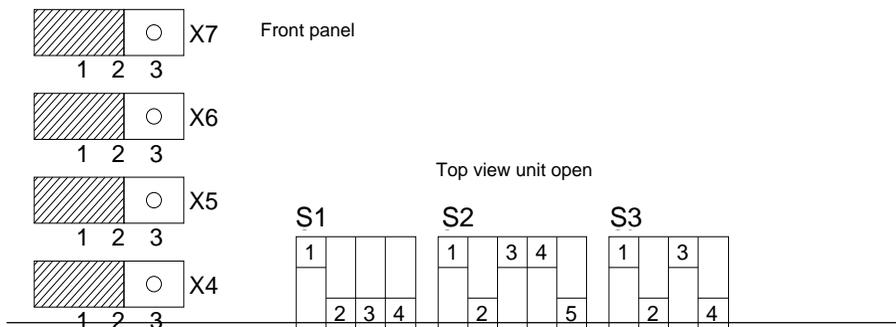
2.2-5 PC compatible → PLC connection

This connection requires the TSX TAC 03 connection kit. It is performed in three steps :

1. Checking the configuration of the TSX TE 01 converter

The RS 232C/20mA CL converter is normally set at the factory; however it is advisable to check the positions of the microswitches and jumpers :

- remove the cover from the unit by inserting a screwdriver in the slots on each side of the unit,
- check that the positions of the 13 microswitches and the 4 jumpers are correct (see diagram below),
- choose the supply voltage using switch S4 : 110V or 220V.

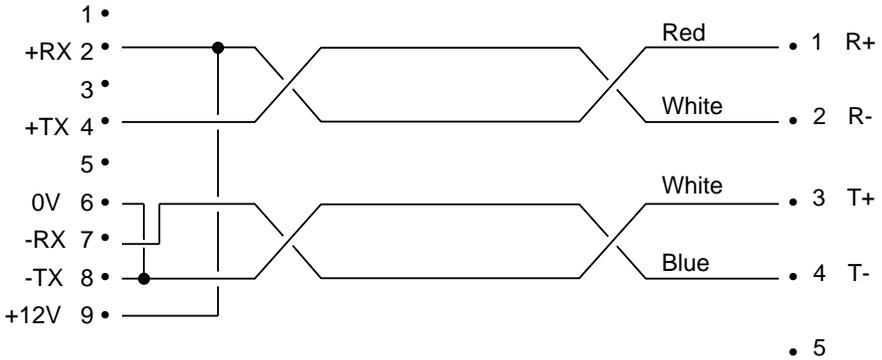


IMPORTANT

The connection may be damaged if this configuration is not used, or if any other type of converter is used.

2.2-6 PLC terminal port → converter connection

This connection uses the TSX CTC 00 cable, wired as shown below :

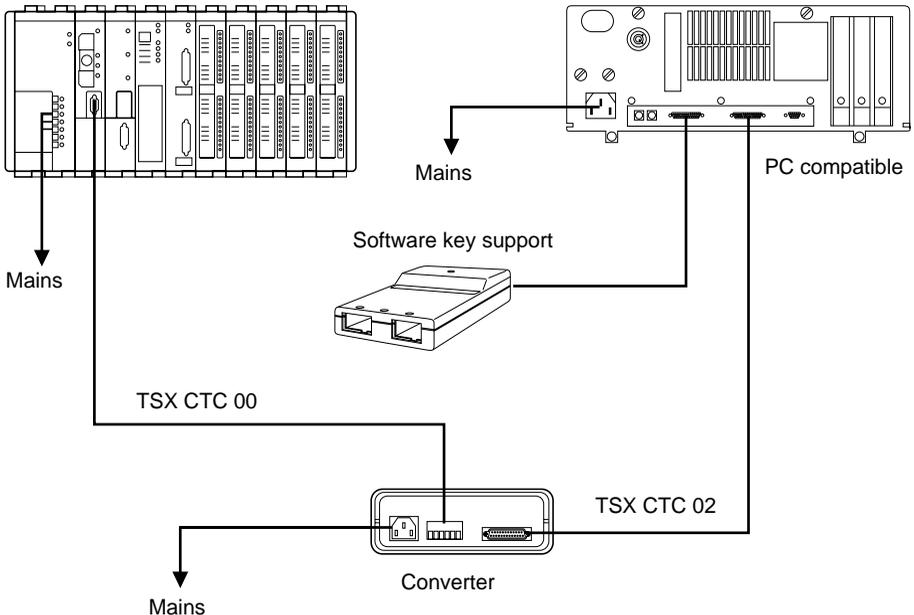


9-pin connector on PLC terminal port

"LINE CONNECTION" 5-pin screw terminal block on the converter

3. Converter → microcomputer connection

This connection uses the TSX CTC 01 or TSX CTC 02 cable, connected between the 9-pin or 25-pin "serial link" microcomputer connector and the 25-pin "V24/RS-232 C CONNECTION" converter connector.



2.3 Software installation

2.3-1 Preliminary operations

Before installing PL7-3 software on the hard disk, it is advisable to :

- read the licensing and warranty certificate concerning the restrictions which apply to copying and installing the software,
- make backup copies of the original installation diskettes and use the backup copies to work with, so as to avoid accidental damage to the originals.

IMPORTANT

The XTEL PACK Vxx diskettes are supplied set to the "write-protect" position, which should not be altered.

2.3-2 Installation procedure

The following operations must be performed before installing PL7-3 software :

1. **close all OS/2 and XTEL sessions in progress** (refer to the corresponding instruction manual).
2. **check that the X-TEL software workshop has already been installed** (refer to the corresponding instruction manual).

Then proceed with installing PL7-3. Otherwise you must install the software workshop first (refer to the corresponding instruction manual).

- **Installing PL7-3 software with XTELPACK V52**

- start a full screen OS/2 session,
- insert XTEL PACK diskette no. 1 in the disk drive,
- enter the identifier of the drive containing the diskette (a: or b:), and press **<Enter>**,
- at the new prompt (eg : [a:\] or [b:\]), enter the **install** command, and press **<Enter>**,
- follow the procedure displayed on the screen : replace diskette no. 1 with diskette no. 2 when requested to do so and then press **<Enter>**.

- **Installation with XTEL PACK V6**

Refer to the PACKS software V6 installation manual.

2.4 PL7-3 files

PL7-3 files are identified by a name of 8 characters maximum, followed by an extension of 3 characters which indicates the type of file. They are stored in the various software workshop subdirectories.

The application and modular documentation files are stored at station level under the PL7_3 subdirectory :

APP	_____	xxx.APP	<i>application description</i>
PL7_3	— APPLI	xxx.BIN	<i>PL7-3 application binary file</i>
		xxx.DAT	<i>data storage</i>
		xxx.XR1	<i>cross-references</i>
		to xxx.XR4	<i>cross-references</i>
		xxx.TIT	<i>title page</i>
		xxx.DES	<i>general information</i>
		xxx.NET	<i>network information</i>
		xxx.COM	<i>program comments</i>
		xxx.CRT	<i>documentation footer</i>
		xxx.DOC	<i>documentation printed in a file</i>
		xxx.MRK	<i>marker for cross-references</i>
	— MOD	xxx.LAD	<i>Ladder source archive</i>
		xxx.LIT	<i>Literal source archive</i>
		xxx.GR7	<i>Grafcet source archive</i>
		xxx.CST	<i>constant word source archive</i>
		xxx.OBC	<i>OFB constant source archive</i>
		xxx.OBM	<i>OFB constant source archive</i> <i>(basic module level)</i>
		xxx.CDE	<i>documentation menu</i>
		xxx.COM	<i>module comments</i>
		xxx.M5M	<i>V5 macro-modules</i>
		xxx.DDA	<i>data screen</i>
		xxx.DPR	<i>program address</i>

PL7-3 software also manages the following files :

xxx.TMP	temporary file,
xxx.BAK	backup file,
xxx.OFB	OFB file (in system space only).

Section	Page
3.1 Accessing PL7-3 software	3/2
3.1-1 Using the Browser (XTEL-BROWSER)	3/2
3.1-2 From the X-TEL software workshop	3/3
3.2 Defining the main window	3/4
3.2-1 Main window	3/4
3.2-2 Menu bar	3/5
3.2-3 Toolbar	3/8
3.2-4 Help balloons	3/8
3.2-5 Status bar	3/9
3.3 Presentation of operating modes	3/10
3.4 Backing up and restoring PL7-3 applications	3/12
3.4-1 Breakdown of a PL7-3 application	3/12
3.4-2 Backing up the application data on disk	3/13
3.4-3 Restoring application data from disk	3/14
3.5 Using the symbol database	3/17
3.5-1 Reminders	3/17
3.5-2 Initialization and updating of the symbol database	3/17
3.6 Quitting the PL7-3 function	3/19
3.7 PL7-3V6 keyboard shortcuts	3/19
3.8 Using notepads	3/23
3.9 Using menus - General layout	3/23
3.10 Using "user-definable lists"	3/24
3.11 Rule for entering data in the PL7-3 V6 entry zone	3/25

3.1 Accessing PL7-3 software

3.1-1 Using the Browser (XTEL-BROWSER)

The Browser is a tool which can be accessed from the Telemecanique Group window or from the Display menu in the X-TEL main windows. It can be used to display the hierarchy of elements in the software workshop and for quick access to the various levels : opening all the windows in the software workshop or launching a function or a tool (see the software workshop documentation : TXT DM XTEL V6 or TXT DM XTEL V52E).

To reduce the PL7-3 window to an icon or to close the session, proceed as follows :

Procedure	Mouse	Keyboard
To reduce the PL7-3 window to an icon	click on the "minimize" button	<Alt> <↓> Minimize <Enter>
To open a PL7-3 session which is in icon form	double-click on the icon	<Ctrl><Esc> <↓> or <↑> <Enter>
To close a PL7-3 session	in the File menu, select Quit	<Ctrl><Esc> <↓> or <↑> <Enter> <Alt><↓> Stop/close <Enter>

3.1-2 From the X-TEL software workshop

The access procedure is exactly the same for all PL7-X software.

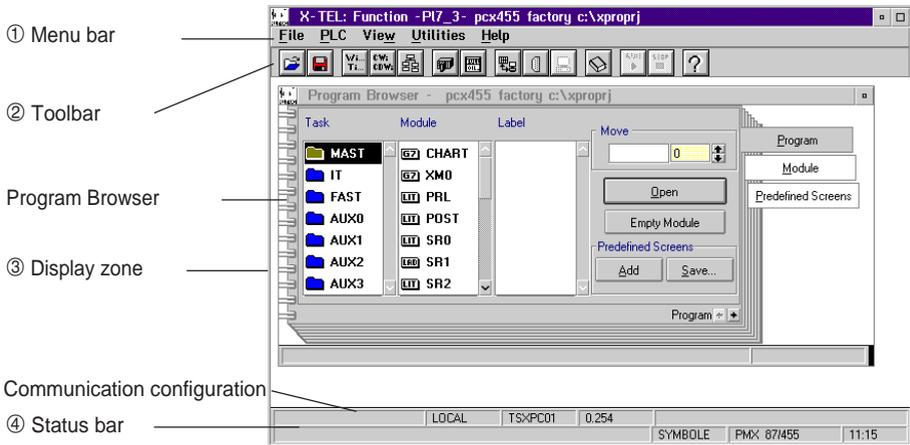
To access PL7-3, open the corresponding window. This is done as follows :

Procedure	Mouse	Keyboard
1. Select the Telemecanique icon	double-click on the icon	<↓> <↑>, <←>, <→>, <←→>, <Enter>
2. Open the User window	double-click on the XTEL icon	<←>, <→>, <Enter>
3. Enter the user parameters, which opens the Volumes window	click in the data entry zone, xxx, Confirm	<↓>, xxx, <Enter>
4. Open a volume	double-click on the Volume icon	<Alt> <←> Window <↓> <↓> Volumes <Enter> <Tab> <F2>
5. Open a project	double-click on the Project icon	as for Volumes <↓> Projects
6. Open a station	double-click on the Station icon	as for Volumes <↓> Stations
7. Open the PL7-3 window	double-click on the PL7_3 icon	as for Volumes <↓> Functions
8. For ease of use, open the PL7-3 window to full-screen	double-click in the window title bar	<Alt> <↓>Maximize <Enter>

3.2 Defining the main window

3.2-1 Main window

This first screen can be used to access all the possibilities offered by the terminal. It is composed of four parts (see below).



(PROGRAM mode window by default)

From the MAIN WINDOW the user selects the terminal operating mode :

- **offline,**
- **online.**

By default the terminal offers :

- **offline operating mode**
- **PLC processor : TSX 67-30, TSX 47-400, PMX 47-420, TSX 47-405 or PMX 47-425 depending on the type of station**
- **Program Browser mode :**
This function, which is used to select program addresses, is offered when the software is opened.

3.2-2 Menu bar

FILE MENU :

Open	in offline mode, this key is used to retrieve the application data from the disk (see section 3.4-3) : - application program (xxx.BIN file), - cross-references (xxx.XRi files), - application program source (xxx.M5M file).
Save	in offline mode, this key is used to save the application data to disk, either in the current file, or under another file name (see section 3.4-2) : - application program (xxx.BIN file), - list of cross-references (xxx.XRi files).
Quit	is used to quit the PL7-3 function (see section 3.4).

PROCESSOR MENU :

Local Memory	(offline operation) This mode, designed particularly for using a terminal in a design office, is used to configure the application, enter the program on the internal RAM memory, archive it on the hard disk, print it, document the application, save, etc. Note : the DATA and DEBUG modes cannot be accessed in offline mode.
TSX Memory	(online operation) In this operating mode, the terminal uses the PLC memory (onboard RAM and cartridge). Two types of operation are offered : connected locally the terminal is connected to a TSX 7 or PMX 7 PLC, the memory used is the user memory supported by this PLC. This mode is used, for example, for debugging, adjusting and modifying an application online, and for archiving on the hard disk. connected via a TELWAY, MAPWAY, ETHWAY or FIPWAY network the terminal is connected to the network (except TELWAY) or to a PLC which is connected to the network. This type of operation offers the same possibilities (debugging, adjustment, etc) as the locally connected type, but with a remote PLC. Access times depend on the network load. The station address is defined at Project level, under X-TEL.
Transfer, Application	is used for PLC → Terminal/Disk transfers
Transfer, Data Words	is used for Wi word transfers (PLC → Terminal/Disk).

PROCESSOR MENU (Cont.) :

Run	offered in online mode; launches program scanning if the PLC is stopped
Stop	offered in online mode; stops program execution if the PLC is running
Init	offered in online mode; initializes Grafcet and the instruction pointer if the PLC is stopped

For each operating mode, the following table shows the options available to a user with "Program Max." access rights, with the terminal offline or online :

Operating mode	Online mode	Offline mode
CONFIGURATION	Read, write	Read, write
PROGRAM	Read, write	Read, write
CONSTANT	Read, write	Read, write
DATA	Yes	No
DEBUG	Yes	No
TRANSFER	Terminal → Processor Processor → disk	Terminal → disk
DOCUMENT/PRINT	Yes	Yes
SEARCH/REPLACE	Yes	Yes

Reminder of the different user categories in X-TEL :

- Operate min : access to display tools,
- Operate Max : access to operator and process diagnostics tools,
- Adjust min : access to application adjustment tools,
- Adjust Max : access to adjustment, PLC operating mode monitoring, maintenance and diagnostics tools,
- Program min : access to all design tools,
- Program Max : access to all workshop tools in offline mode, online with the PLC stopped or online with the PLC running.

See section 15.8 for information on accessing PL7-3 operating modes.

VIEW MENU :

Variables	immediate change to display variables (addresses)
Symbols	immediate change to display symbols
OFB	accesses the description of the OFBs configured in the application (V4 or V5 application only)
Memory Usage	displays the memory usage screen (see section 15.1)
Tool box	is used to show or hide the toolbar (see section 3.2-3).

TOOLS MENU :

Configuration Program Constants Documentation Data Search/ Replace	for options relating to the operating modes see section 3.3
Linking XTEL-CONF	links PL7-3 to the configuration defined in XTEL-CONF. If no configuration has been defined, it is suggested that XTEL-CONF be launched.
Activate XTEL-SDBASE Tool	is used to open the XTEL-SDBASE application.

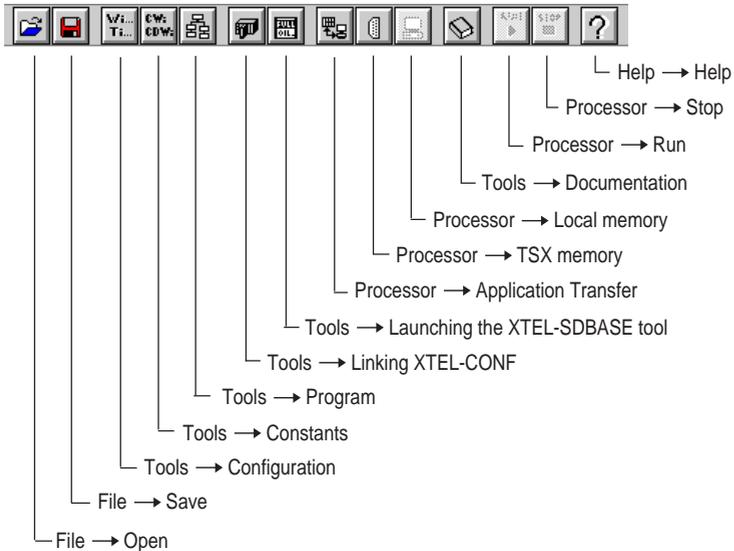
HELP MENU :

Help	calls up the PL7-3 general help function.
Product Information	product version.

3.2-3 Toolbar

This toolbar is designed to give quicker access to the main functions of the menu bar.

Definition of buttons



3.2-4 Help balloons

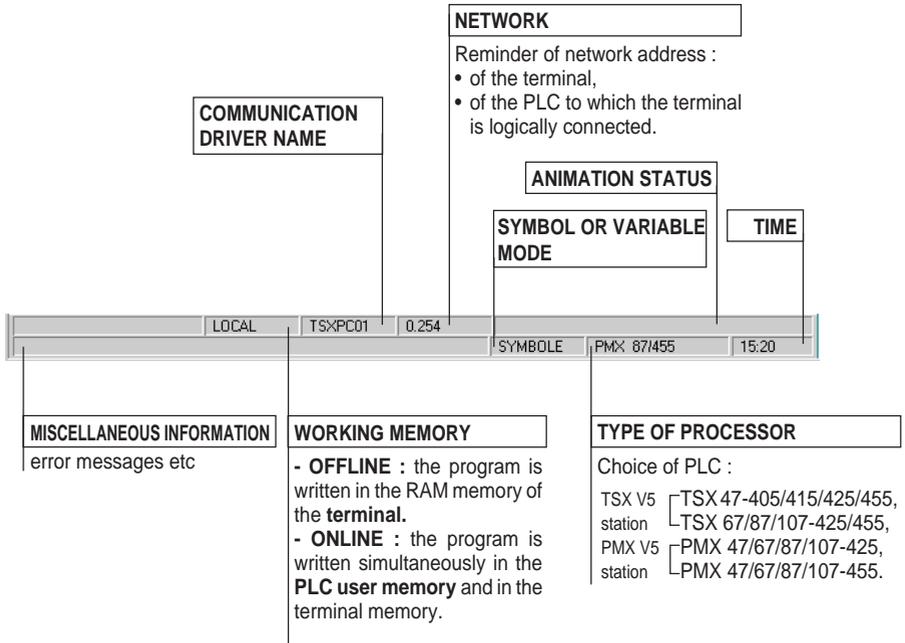


If the mouse is held still over a button on the toolbar, a message explaining the function of the button will appear.

3.2-5 Status bar

The role of this is to display information relating to the application.

Definition of the status bar :



3.3 Presentation of operating modes

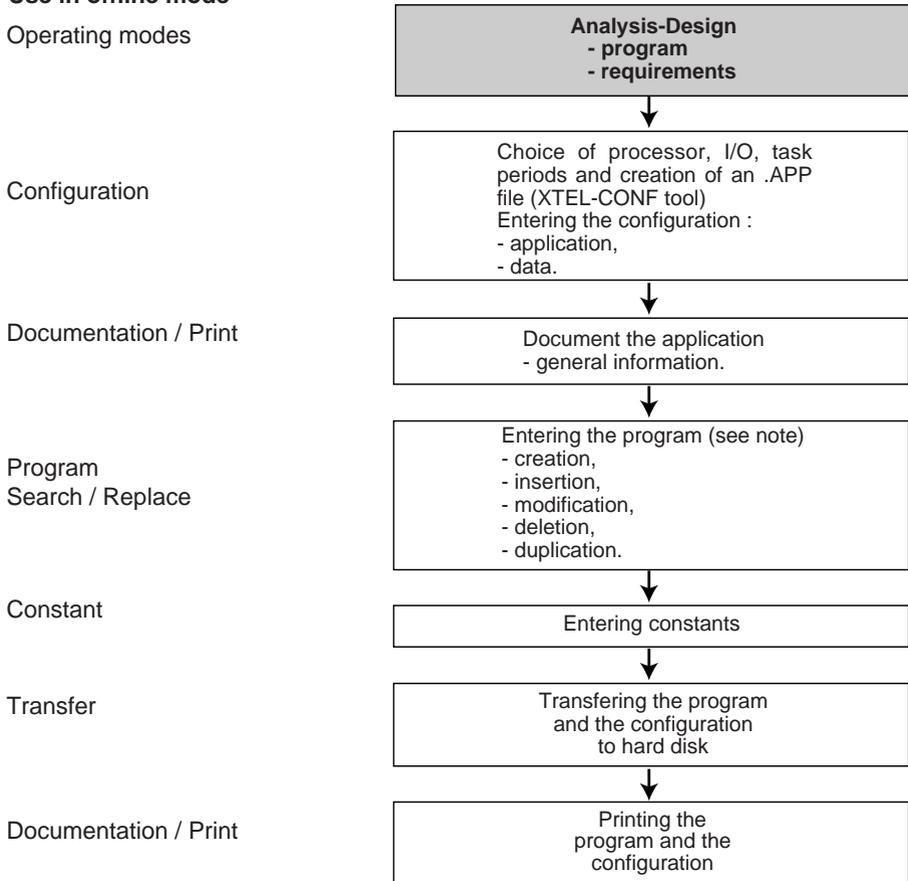
PL7-3 software offers 8 operating modes which can be used for the : design, debugging, adjustment, maintenance and documentation of a control system :

- Configuration
- Constants
- Program
- Data
- Debug
- Transfer
- Search/Replace
- Documentation/Print

Note

The Control, Adjust and Terminal modes of previous versions are no longer available as offline modes.

Use in offline mode

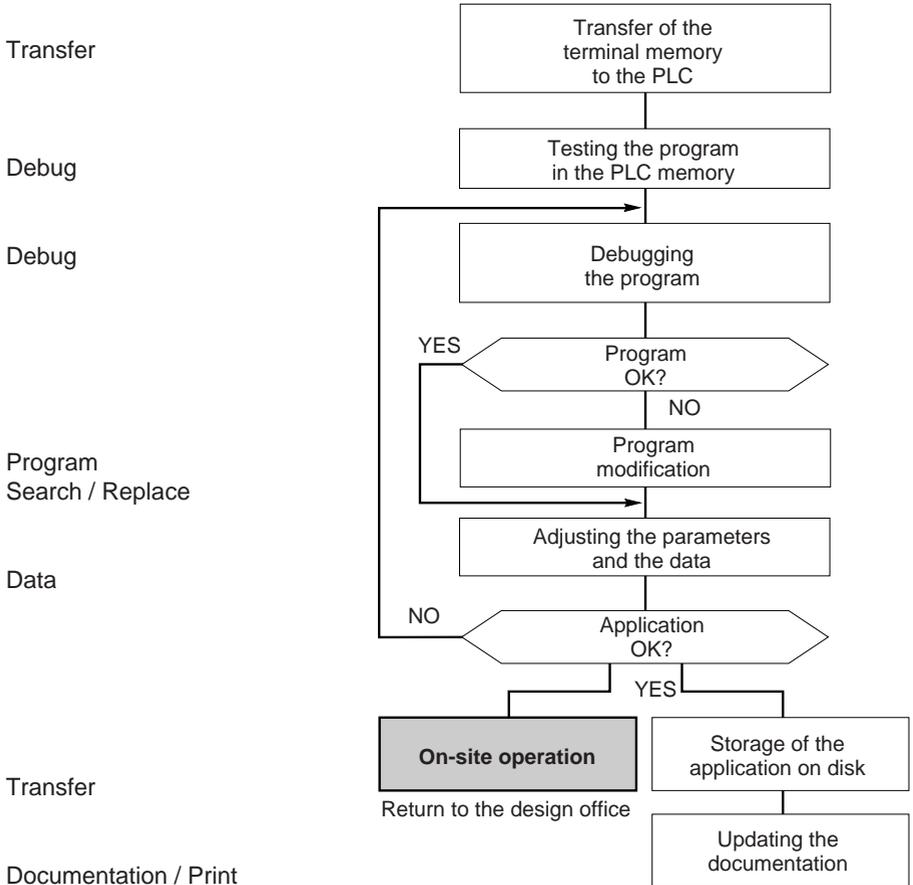


Note

Entry of the configuration, program and symbols is performed in the terminal RAM memory. It is therefore advisable to perform frequent saves.

Use in online mode

Operating modes



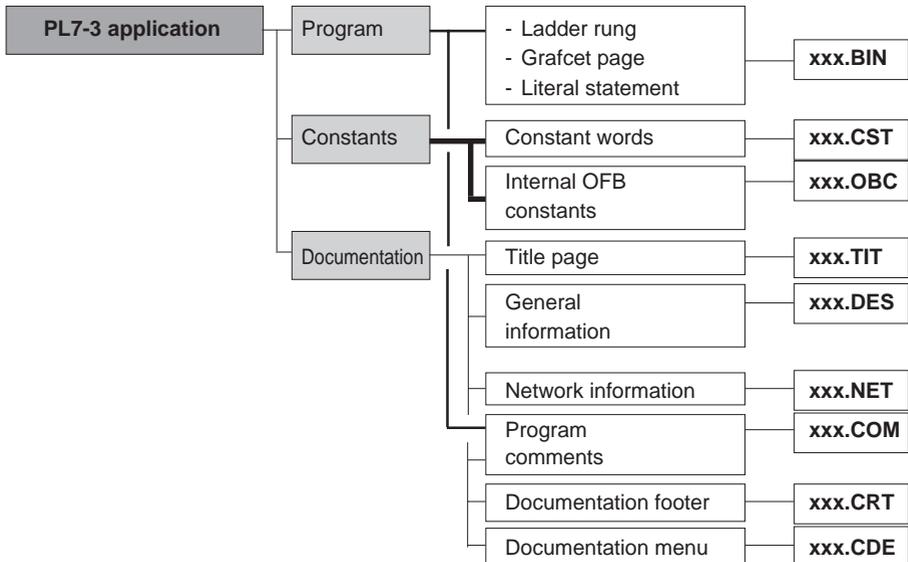
3.4 Backing up and restoring PL7-3 applications

3.4-1 Breakdown of a PL7-3 application

A PL7-3 application is composed of :

- the application program, accessible in PROGRAM mode,
- constants, accessible in CONSTANT mode,
- the application documentation file, accessible in DOCUMENT/PRINT mode.

It is broken down into files (xxx.BIN, xxx.COM, etc) which are used to back up and restore the various parts of the PL7-3 application.



Notes

- the program files (xxx.BIN) and documentation footer files (xxx.CRT) are not saved automatically. It is therefore advisable to perform frequent saves to avoid losing any information if there is a power break.
- the other files (xxx.TIT to xxx.CDE) are automatically saved when the file is closed (as soon as the operator quits the corresponding function).

3.4-2 Backing up the application data on disk

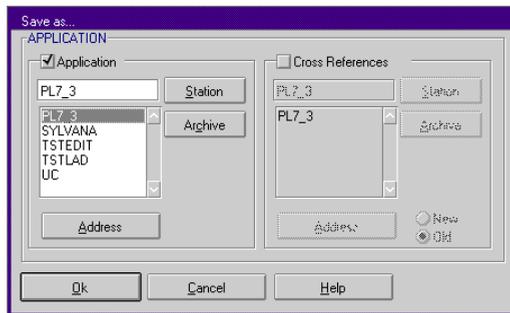
PL7-3 software offers several possibilities for backing up and restoring PL7-3 application data :

- using the **Save** option from the **File** menu bar, available on the main screen in offline mode,
- using the modes corresponding to each file handled :
 - Transfer mode for the xxx.BIN file (section 12.3).

Using the toolbar gives quicker access to the files.

Backup procedure

- File, Save** accesses the screen below which is used to back up application data if it is present in the terminal memory.
- application (xxx.BIN file) in the APPLI zone,
 - table of cross-references (xxx.XR1 and xxx.XR4 files) in the APPLI zone.



- Archive** assigns the global archive name (PL7_3) as the file name.
- Station** assigns the name of the current station as the file name.
- OK** saves the selection made. If the file already exists, the new program will overwrite the old one, after confirmation by **Yes**.
- Cancel** cancels the selections made and exits the dialog box without starting the action.
- Old** selects the existing tables of cross-references.
- New** recalculates the tables of cross-references.
- Directory** lists all of the files already stored in the PL7_3\APPLI or PL7-3\MOD sub-directory according to the selection made.
- Help** calls up the online help function.

The **Application** and **Cross-references** check boxes are used to select the files to be saved.

The **Application** and **Cross-references** entry fields are used to modify the name of the current archive file.

3.4-3 Restoring application data from disk

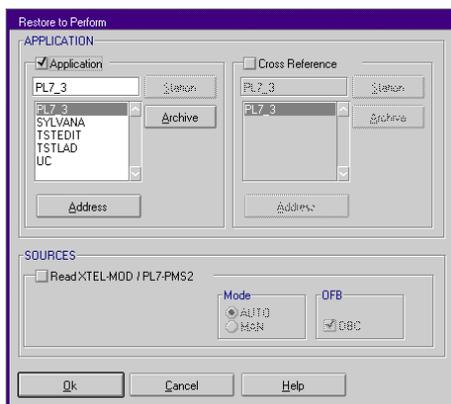
Restore procedure

File, Open via the menu bar accesses the screen below which is used to load the following to the terminal memory :

- application data saved on disk :
 - application (xxx.BIN file, APPLI zone),
- table of cross-references (XR1 to XR4 files, APPLI zone),
- application source (macro-modules to be integrated in the application from the xxx. M5M file, MOD zone),
- standard constants (xxx.CST file, MOD zone).

This screen is accessible in online mode, with the PLC running or stopped, for :

- the application source,
- the standard constants.



- Archive** assigns the global archive name (PL7_3) as the name of the file to be retrieved.
- Station** assigns the name of the current station as the name of the file to be retrieved.
- OK** saves the selection made and starts restoring the selected files.
- Cancel** cancels the selections made and exits the dialog box without starting the action.
- OBC** specific to reading activated XTEL-MOD/PMS2 is used to read XTEL-MOD or PMS2 source modules, with or without internal OFB constants.
- Directory** lists all files already stored in the PL7_3\APPLI or PL7-3\MOD subdirectory according to the selection made.
- Help** calls up the online help function.

The **Application** and **Cross-references** selection windows are used to select the files to be restored.

The **Application** and **Cross-references** check boxes confirm the type of files to be restored.

The **Mode** group of buttons selects the automatic (**AUTO**) or manual (**MANU**) mode for reading the xxx.M5M file (reading the V5 source). AUTO mode is selected by default.

Note

If the cross-references read do not correspond to the current application, a message will indicate this to the user.

AUTO and MANU modes in «read source»

«Read source» concerns the integration of macro-modules in the application.

• Reminder of macro-modules

Definition of a macro-module

A macro-module consists of several modules that are handled as a single unit by the user.

Principle of generating a macro-module

the user must perform the following operations :

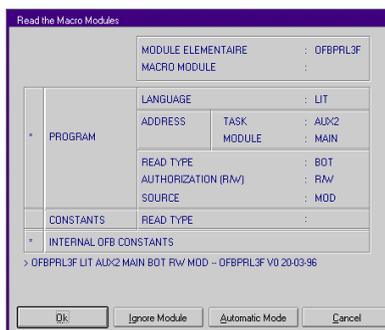
- for each module the user must generate, under PL7-3 in PROGRAM mode, the original module files and then create each module using the XTEL-MOD tool,
- using the XTEL-MOD tool, the user must then define the macro-module and the modules which it contains. The elements which make up a macro-module are program files (xxx.LIT, xxx.LAD, xxx.GR7), constants files (xxx.CST), internal OFB constants files (xxx.OBC) and program comment files (xxx.COM). The reading of a macro-module is directed by a descriptive file (xxx.M5M or xxx.MCR) which contains the list of macro-modules to be read and retains in memory the macro-modules which have already been read.

• Reading the source file in automatic mode (AUTO)

In normal operation this mode (default selection) does not require any action by the user. It stops on any errors which may arise from the descriptive file (xxx.M5M or xxx.MCR) or the program or constants files (xxx.LIT, etc, xxx.CST).

• Reading the source file in manual mode (MANU)

In this mode, the terminal stops before each module or macro-module is read and the following screen is displayed.



OK	starts reading.
Ignore Module	ignores the module or macro-module and goes to the next one.
Automatic Mode	continues reading in automatic mode, as before.
Abort	stops reading the source.

Note

The creation, use and handling of modules and macro-modules are described in the XTEL-MOD TXT DM MDD V52E manual.

3.5 Using the symbol database

3.5-1 Reminders

Principle of the XTEL-SDBASE tool

Under the software workshop, the XTEL-SDBASE station tool manages the global system database (SDB) of a station and makes a common database, containing each variable, its symbol, a comment and an extended comment, available to all programs on the station.

Since PL7_3V5, PL7-3 no longer has a local copy of the symbol database (.SCY file), but instead reads the symbols directly in XTEL-SDBASE.

The principle and use of the XTEL-SDBASE tool are described in detail in the XTEL integrated tools manual, reference TXT DM TLS V6E (or TXT DM TLS V52E).

Definition of a symbol

A symbol represents a character string conforming to a defined syntax permitting the symbolic representation of an object belonging to Series 7 languages.

A symbol comprises a maximum of 8 characters and always starts with a letter or the character # if the symbol is generated automatically by SDBASE.

The first character is a **capital letter**, the other characters can be lower case letters, figures, or the characters «_» «\$» «%» «|» «~».

The following are not admitted as symbols: PL7 objects and instructions (W0, B10, If, Then, etc), extended ASCII codes. However, a symbol can start with the name of a PL7 object or instruction (W0_mast, B10prl, etc).

3.5-2 Initialization and updating of the symbol database

Initialization

In order to use the symbol database from PL7-3, the user must perform the following actions :

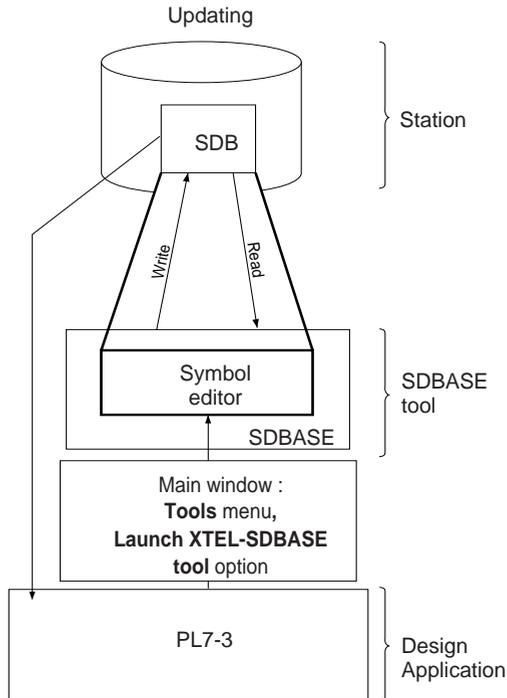
- With the SDBASE tool, initialization of the station symbol database :
 - either by manually entering the symbols and associated data
 - or by using the Merge function to integrate the symbols archived and previously entered in a xxx.SCY or xxx.SCZ file.

Updating

To modify the PL7-3 symbol database, the user can call up the XTEL-SDBASE tool symbol editor from within PL7-3 as follows :

- From the main window, the **Launching the XTEL-SDBASE tool** option in the **Tools** menu calls up the XTEL-SDBASE tool symbol editor without quitting PL7-3, in order to modify or insert symbols.
After modification or insertion of symbols, pressing the **<F3>** function key twice in succession allows the user to exit the XTEL-SDBASE editor.
- From the Literal and Ladder editors, the **Symbolize** option from the **Option** menu calls up a symbols window which can be used to modify or insert symbols.

Diagram



Important

In the main window screen, PL7-3 does not reserve the database in read/write mode. However, in all other modes it does reserve the database and prohibits any access to SDBASE concurrent with that which it is processing.

3.6 Quitting the PL7-3 function

This screen, which is accessed from the main window screen by selecting **Quit** from the **File** menu or by pressing the **<F4>** key is used to quit the PL7-3 function.

If modifications have been made beforehand, a window appears to allow these modifications to be saved if necessary.

3.7 PL7-3V6 keyboard shortcuts

Headings	Keys	Function
Window Management	<Alt>	Accesses the action zone (band) of the window
	<Down arrow>	Displays a pull-down menu
	<Alt> <Letter>	Displays the pull-down menu where <letter> is underlined in the menu name
	<Down arrow> <Up arrow>	Moves within the menu
	<Esc>	Accesses the body of the window
	<Ctrl Esc>	Accesses the list of windows (of active tasks)
	<Alt Esc>	Accesses the other windows in succession (other active tasks)
System menu	<Shift> <Esc> or <Alt> <Space>	Accesses the system menu
	<Alt> <F4>	Closes the active window
	<Alt> <F5>	Restores the active window to its original position and size
	<Alt> <F1>	Accesses the Program Browser from the main PL7-3 window
	<Alt> <F7>	Accesses the mode for moving the active window
	<Left arrow> <Right arrow>	Moves the window
	<Down arrow> <Up arrow>	
	<Enter>	Confirms the current position
	<Alt><F8>	Accesses the modification of the window size
	<Left arrow> <Right arrow>	Modifies the size of the window
	<Down arrow> <Up arrow>	
	<Enter>	Confirms the current size
	<Alt> <F9>	Minimizes the active window to icon form
<Alt> <F10>	Maximizes the active window	

Special functions	<p style="text-align: center;"><F1></p> <p style="text-align: center;"><Alt> <V></p> <p style="text-align: center;"><Alt> <S></p> <p style="text-align: center;"><Alt> <T></p> <p style="text-align: center;"><Alt> <O></p> <p style="text-align: center;"><Alt> <C></p> <p style="text-align: center;"><Alt> <K></p>	<p>Accesses help on the product and the keys</p> <p>Immediate change to display variables (addresses)</p> <p>Immediate change to display symbols</p> <p>Displays the memory allocation screen</p> <p>Accesses the description of the OFBs configured in the application</p> <p>Accesses the program comment for the label or current GRAFCET page or the comment for the selected module (from Program Browser)</p>
Function keys	<p style="text-align: center;"><Enter></p> <p style="text-align: center;"><Esc></p> <p style="text-align: center;"><Backspace></p> <p style="text-align: center;"></p> <p style="text-align: center;"><Page Up></p> <p style="text-align: center;"><Page Down></p> <p style="text-align: center;"><Insert></p> <p style="text-align: center;"><Alt> <Right arrow> or <^></p> <p style="text-align: center;"><Ctrl> <Page Up></p> <p style="text-align: center;"><Ctrl> <Page Down></p>	<p>Confirms the current operation</p> <p>Cancels the current operation</p> <p>Deletes the character to the left of the cursor</p> <p>Deletes the character highlighted by the cursor</p> <p>Accesses the previous page (editors)</p> <p>Accesses the next page (editors)</p> <p>Inserts an instruction (statement, rung) or a variable/constant</p> <p>Transfer instruction (assignment)</p> <p>Moves up a step in the screen sequence. Closes the current editor</p> <p>Accesses the object (or its parameters) highlighted by the cursor</p>
Keys common to editors	<p style="text-align: center;"><Esc></p> <p style="text-align: center;"><Shift> </p> <p style="text-align: center;"><Ctrl> <Insert></p> <p style="text-align: center;"><Shift> <Insert></p> <p style="text-align: center;"><Ctrl> <G></p> <p style="text-align: center;"><Ctrl> <S></p> <p style="text-align: center;"><Ctrl> <M></p> <p style="text-align: center;"><Ctrl> <O></p> <p style="text-align: center;"><Ctrl> <C></p> <p style="text-align: center;"><Ctrl> <Page Up></p> <p style="text-align: center;"><Ctrl> <Page Down></p> <p style="text-align: center;"><Alt> <F1></p>	<p> Cancels the current operation</p> <p>Cut</p> <p>Copy</p> <p>Paste</p> <p>Go to</p> <p>Save in a file</p> <p>Modify</p> <p>Edit an OFB</p> <p>Confirm</p> <p>Zoom</p> <p>Close</p> <p>Accesses the Browser</p>

Keys common to editors (cont.)	<p><Alt> <K> <Alt> <X></p> <p><Ctrl> <F5></p> <p><Alt> <C></p> <p><Ctrl> <F9></p> <p><F10></p>	<p>Enters the comment</p> <p>Accesses the list of variables used in a program element. Only possible to access in the editors</p> <p>Accesses SDBASE (list of symbols)</p> <p>Accesses the module comment selected (from Program Browser)</p> <p>Copies the current screen to the printer</p> <p>Accesses the menu in the same way as <Alt></p>
Program browser	<p><Alt> <F1></p> <p><Alt> <Page Down> <Alt> <Page Up></p> <p><Tab></p> <p><Enter></p> <p><Left arrow> <Right arrow> <Down arrow> <Up arrow></p>	<p>Accesses the Program Browser for each PL7-3 screen</p> <p>Accesses the Program, Module and Predefined Screens tabs</p> <p>Moves the Focus from group to group</p> <p>Acts on the selection indicated by the Focus</p> <p>Moves the Focus within a group</p>
Application/ constants configuration	<p><Alt> <I> then <C> <Tab></p> <p><Left arrow> <Right arrow> <Down arrow> <Up arrow></p> <p><Space></p> <p><Alt> <Right arrow> <Alt> <Left arrow></p> <p><Ctrl> </> <Ctrl> <\> <Enter></p> <p><Alt> <Page Down> <Alt> <Page Up></p> <p><Alt> <F4></p>	<p>Calls up the configuration screen</p> <p>Moves the Focus from group to group</p> <p>Moves the Focus within a group</p> <p>Sets or resets options</p> <p>Moves within a group which contains Srx type parameters</p> <p>Selects all the elements</p> <p>Deselects all the elements</p> <p>Confirms the selections</p> <p>Accesses the tabs</p> <p>Closes and saves the configuration</p>

Literal	Right-hand mouse button <F4> or <Ctrl> <M> <Space> <Shift> <Ctrl> <Insert> <Shift> <Insert> <Ctrl> <D>	Displays a contextual menu Modifies the line Selects the line Cut Copy Paste Inserts a diagnostic OFB
Ladder	Right-hand mouse button <Shift> <Ctrl> <Insert> <Shift> <Insert> <Space> <Shift> <F10> <Ctrl> <Page Down> <Ctrl> <Page Down>	Displays a contextual menu Cut Copy Paste When modifying the rung, with the cursor on the label, this is used to modify the label (same for a comment) Places a diagnostic OFB Warning : the cursor must be on the last line of the rung being modified Zoom OFB ; Srx ; Function block Returns Srx to mast ; function block
Data mode	<Insert> <Insert> w10=10-30 <Ctrl> <Insert> <Ctrl> <O> <Ctrl> <S> <F8> <Ctrl> 	Inserts a new value in a table Inserts W10 to W30 Next variable Retrieves a saved list Saves the list in the left-hand box Displays all objects in realtime Deletes the table
Data mode	<Insert> <Ctrl> <Insert> <Ctrl> <O> <Ctrl> <S> <F8> <Tab> or <Ctrl> <O> <Ctrl> <Page Up> <Ctrl> <Page Down>	Inserts a new value in the left-hand table Next variable Retrieves a saved list Saves a list in the left-hand box Displays all objects in realtime Changes the Focus on the table to the right or the left table Log of a variable Next variables

Data mode (cont.)	<Esc>	Stops the log ; stops the next variables
	<Ctrl> 	Deletes the right or left table
	<S>	Searches for forced objects
	<F2>	Modifies the value (set)
	<F3>	Resets
	<F4>	Forces to 1
	<F5>	Forces a bit to 0/blocks a step activity
	<F6>	Unforces a bit/unblocks a step activity (Xi)
	<Ctrl> <F8>	Replaces the selected object with the next object
	<Shift> <F8>	Replaces the selected object with the previous object

Terminology :

- **FOCUS** : a dotted rectangle moving from group to group, controlled by the <TAB> key.
The Focus activates the current selection.
- **TABS** : zones on the right or at the bottom of configuration screens.
They are used to go to a set of configuration functions.

3.8 Using notepads

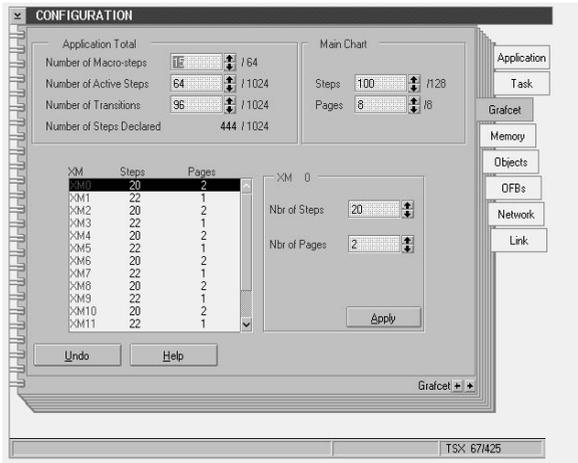
PL7-3 V6 uses the idea of notepads to present certain operating modes (configuration, constants, etc). To use the PL7-3 V6 notepads (via the keyboard and the mouse) refer to the OS/2 system documentation.

3.9 Using menus - General layout

PL7-3 V6 conforms to the general layout rules for OS/2. To use menus (go to a menu using the keyboard, etc) dialog boxes, message boxes, refer to the OS/2 system documentation.

3.10 Using "user-definable lists"

In several PL7-3 V6 options, (configuration, constants, etc) a list gives the parameters which can be modified by the user.



The right-hand group gives the parameters of the element currently selected from the list, which can be modified. A modification to this group (entered in the data entry zone then confirmed by pressing **<ENTER>** or using the arrow keys) affects all the elements selected in the left-hand list simultaneously.

The left-hand list can be a multiple selection list. In this case, the operating modes available are as follows :

<SHIFT> + Click	Move with <↓><↑> and <SHIFT> + <SPACE>	selects the zone from the anchor point to the current mouse position.
<CTRL> + Click	Move with <↓><↑> and <CTRL> + <SPACE>	multiple selection of individual elements selected with the mouse.
Drag mouse		selects all elements highlighted as the mouse moves.
Drag mouse	<CTRL> + <↓>	selects all the elements in the list.
	<SHIFT> + <↓>	deselects all the elements in the list.

"Apply"

This button has the effect of applying the parameters of the line which is in the focus (small dotted box around the element in the list) to all the other elements selected in the list.

3.11 Rule for entering data in the PL7-3 V6 entry zone

In all the PL7-3 V6 options, the software uses simple or incremental data entry zones to manage user entries.

simple entry zone

incremental entry zone

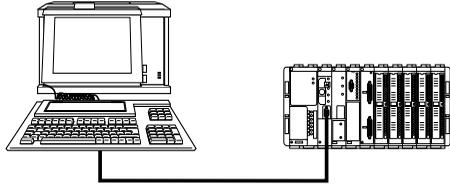
The rule for entering data in these zones is to confirm the entry by pressing **<ENTER>**. If you exit the zone without pressing **<ENTER>**, the current entry is not taken into account and the previous value is retained.

Section	Page
4.1 Physical and logical connections	4/2
4.1-1 Physical connection	4/2
4.1-2 Logical connection	4/2
4.2 Reservation of a connected PLC	4/3
4.2-1 Reservation in online mode, connected via the network	4/3
4.2-2 Dereservation	4/3
4.2-3 Online mode, connected locally	4/4
4.3 Principles of terminal to PLC connection	4/5
4.3-1 Identical PLC and terminal memories	4/5
4.3-2 Terminal to PLC connection	4/5
4.4 Operation in online mode, connected locally	4/6
4.4-1 Connection procedure	4/6
4.5 Operation in online mode, connected via a network	4/6
4.5-1 Connection procedure	4/6
4.6 Repeating the logical connection request	4/7

4.1 Physical and logical connections

4.1-1 Physical connection

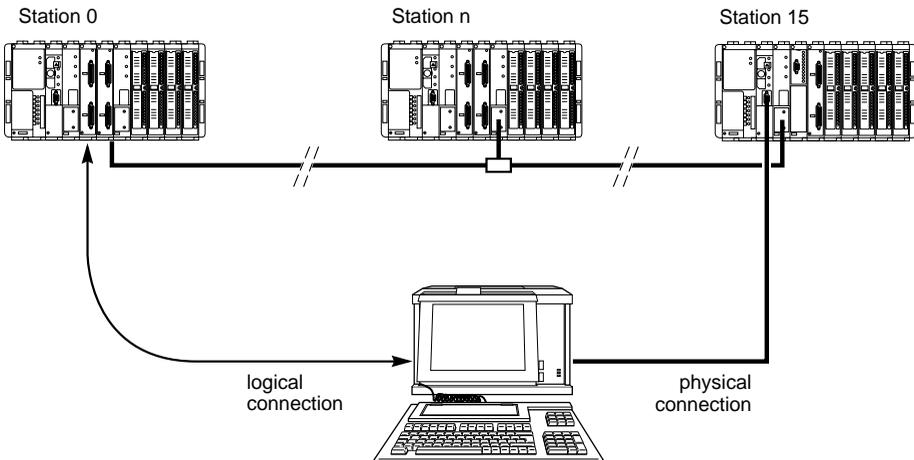
Physical connection is defined by the physical link (connecting cable) between the terminal and the PLC terminal port.



4.1-2 Logical connection

Logical connection corresponds to the communication established between the terminal and the PLC with which it is communicating.

When a PLC is connected locally to the terminal, the logical connection corresponds to the physical connection.



For example, in the TELWAY network shown above the terminal is physically connected to station 15. The logical connection can be established with any PLC on the network (here station 0).

The terminal can therefore communicate with the memory of PLC station 0.

The choice is determined by the user who :

- defines the logical connection at Project level, by selecting a station in the secondary Stations window of the X-TEL software workshop,
- selects **TSX Memory** from the **Processor** menu in the main screen,
- reads the physical (TERMINAL) and logical (TSX) addresses in the section of the status bar reserved for this purpose.

4.2 Reservation of a connected PLC

4.2-1 Reservation in online mode, connected via the network

On a TELWAY, MAPWAY, ETHWAY or FIPWAY network, any FTX 507/517 terminal or microcomputer can be physically connected to any TSX/PMX 47, 67, 87 or 107 station. Consequently, several terminals can request logical connection to the same PLC station in the authorized operating modes.

In order to avoid conflicts of access and procedure, a PLC cannot usually be logically connected to two terminals. Each terminal therefore automatically reserves the PLC to which it is logically connected (except in TRANSFER mode).

This reservation only applies to the zone reserved for PL7-3 and not to the zones reserved for other functions (communication, axis control, etc), which can always be accessed by a terminal.

As soon as one of the following operating modes is selected, the PLC is automatically reserved :

PROGRAM	CONSTANT
DATA	DEBUG
DOCUMENT/PRINT	SEARCH/REPLACE
CONFIGURATION	

A terminal cannot therefore be logically connected in these modes to a PLC that is already reserved by another terminal. If logical connection is requested, a message indicates that the PLC is already reserved.

In TRANSFER mode no reservation is made, therefore a terminal can be logically connected to a PLC which has already been reserved.

4.2-2 Dereservation

The reservation of a PLC is canceled one minute after it is logically disconnected from the terminal which made the reservation. This corresponds to the quitting of online mode by the terminal which made the reservation, or to the physical disconnection of the terminal.

Note

If the PLC program contained a breakpoint, dereservation of the PLC automatically cancels the breakpoint. However, if the task was stopped on the breakpoint at the time of dereservation, the task remains stopped. Dereservation has no effect on the state of forced bits.

4.2-3 Online mode, connected locally

In the case of a terminal connected to a PLC which is not on a network (logical connection = physical connection), the method of reservation and dereservation is exactly the same as that described on the previous page.

B

4.3 Principles of terminal to PLC connection

Access to operating modes depends on the operating mode of the terminal (offline or online). Certain operating modes are only available when the terminal is online. The rules of access are the same whether the terminal is connected locally or remotely via a TELWAY, MAPWAY, ETHWAY or FIPWAY network.

The operating modes which make a reservation require the contents of the memory in the PLC and the contents of the memory in the terminal to be identical.

4.3-1 Identical PLC and terminal memories

To ensure that these memories are identical, which is essential in online mode, the user must :

- either load the memory of the terminal from the hard disk containing exactly the same application program as that in the memory of the PLC to be connected (see section 12.3).
- or load the local memory of the terminal from the PLC memory by using the **Transfer Application** option in the **Processor** menu (see section 12.2).

4.3-2 Terminal to PLC connection

When the memories in the terminal and the PLC are identical, connection is immediate (when connected locally).

- **PROGRAM mode**
During program creation or modification, the terminal works with its internal memory and at the same time automatically updates the memory in the PLC.
- **DEBUG, DATA mode**
Only useful data is exchanged (animation, realtime values, etc). This optimizes the performance of the terminal in online mode so that it is similar to that obtained in offline mode.

It is recommended that the user should always have an up-to-date backup of the application.

An application always requires that memories are identical, and so transferring the application to the terminal will be preferably performed from the backup on the hard disk.

4.4 Operation in online mode, connected locally

4.4-1 Connection procedure

The procedure for changing the operating mode from offline → online, connected locally, is performed from the main window using the **TSX Memory** option in the **Processor** menu.

If the contents of the memory in the terminal and the memory in the PLC are not identical, a dialog window appears.

- Yes** transfers the memory of the PLC to the internal memory of the terminal by overwriting it.
When the logical connection is established (which takes several seconds), the first screen of the selected mode is displayed.
- No** cancels the connection request. The user can then select another mode.

Note

If the PLC and terminal memories are identical (configuration and program) logical connection is immediate.

4.5 Operation in online mode, connected via a network

4.5-1 Connection procedure

The procedure for changing the operating mode from offline → online, connected via the network is identical to that described above, except that the address of the station to which the terminal is to be logically connected must be defined by the user at Project level.

When the terminal is physically connected to a station on a TELWAY, MAPWAY, ETHWAY or FIPWAY network, it recognizes the number of the station and displays it on the main screen, in the section of the status bar reserved for the network address.

Having defined the station which is logically connected to the terminal, the main screen accesses the operating mode selection (**PROCESSOR MEMORY** from the **TSX Memory** option, under **Processor**).

If the contents of the memory in the terminal and the memory in the PLC are not identical, a dialog window appears.

- Yes** transfers the memory of the PLC to the internal memory of the terminal by overwriting it and displays a message. The time taken for connection may be longer than for a local connection (for example, with MAPWAY which requires certain preparation).
- No** cancels the connection request.

4.6 Repeating the logical connection request

During a logical connection request, the station which is physically connected may not be able to communicate, and this is indicated by an error message :

Connection error between terminal and PLC

- PLC is not connected or not powered up,
- PLC is faulty.

Restart repeats the connection request,

Cancel if the terminal has not detected a fault, this returns to offline mode and preserves the application.
Otherwise, the terminal is reinitialized and loaded with the default configuration and the application that was in the internal memory of the terminal is lost.

Driver initialization error

- PLC is not connected or not powered up,
- PLC is faulty,
- module is missing,
network is overloaded.

The terminal then continually repeats the connection request. If connection becomes possible, communication is established. Otherwise :

Restart repeats the request to communicate with the PLC.

Cancel if the terminal has not detected a fault, this request is canceled and the application is preserved. Otherwise, the terminal is loaded with the default configuration and the application that was in the internal memory of the terminal is lost.

Section	Page
5.1 Presentation of CONFIGURATION mode	5/2
5.2 Configuring the application	5/4
5.3 Configuring tasks	5/6
5.4 Configuring Grafcet	5/7
5.5 Configuring the user memory	5/9
5.5-1 Segments	5/10
5.6 Configuring bit, word and function block objects	5/14
5.6-1 Procedure	5/14
5.6-2 Configuring the control system function blocks	5/15
5.6-3 Configuring the text blocks	5/16
5.7 Configuring the optional function blocks	5/18
5.8 Configuring the network services	5/20
5.9 Configuring the link	5/21
5.10 Modifying the configuration in online mode	5/22
5.11 Reconfiguration function	5/23
5.11-1 Procedure	5/23
5.11-2 Reconfiguration check	5/25
5.11-3 Reconfiguration	5/26
5.11-4 Abort reconfiguration	5/26
5.11-5 Reconfiguration check printout	5/27
5.11-6 Return to data entry	5/27

5.1 Presentation of CONFIGURATION mode

CONFIGURATION mode is used to define the context of the physical and logical operation of the application. The configuration, essential for creating or modifying the application, is used to check the consistency of the objects entered (type, number, etc), in relation to those declared in this mode. Any inconsistencies detected are indicated by an error message.

This mode has 8 headings which are used to configure the :

- application,
- tasks,
- Grafcet if used,
- user memory and memory segments (code and graphics),
- bits, words and function blocks,
- optional function blocks,
- network,
- PLC terminal port.

To quit the mode, confirm any modifications made, which causes either :

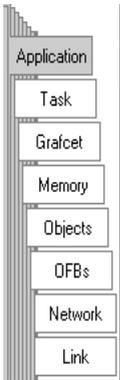
- reconfiguration of the application if the modifications are major. In this case they are only authorized in offline mode.
It is recommended that a diagnostic check of the new configuration is performed before reconfiguring the application. This check is executed by the terminal and ensures consistency between the new configuration and the existing program.
- or no reconfiguration of the application, if the modifications are minor (modifications of cold start conditions or of function block parameters). In this case, they are authorized in both offline and online modes, whether the PLC is running or stopped. However, they are not taken into account by the application program until after a cold start (SY0 = 1).

When PL7-3 is accessed, it is automatically initialized with the current station application : PL7_3.BIN. There are three possible scenarios :

1. The PL7_3.BIN file exists on the disk. In this case, the current application is read automatically,
2. The PL7_3.BIN file does not exist on the disk. The system automatically links PL7-3 to the configuration entered in the XTEL-CONF station tool,
3. The PL7_3.BIN file does not exist on the disk and no configuration has been entered in XTEL-CONF. The system prompts the user to launch the XTEL-CONF tool to enter a configuration.

Configuration mode is accessed from the **Configuration** option in the **Tools** menu.

The tabs on the right of the notepad are used to configure the :



- **application :**

- date, name and version of the application,
- number of subroutines,
- use of Grafcet,
- cold start conditions.

- **tasks :**

- number of subroutines,
- display of associated I/O modules.

- **Grafcet** if it is defined in the application :

- number of steps, macro-steps and pages.

- **memory structure of the application :**

- display of memory volumes defined and managed by XTEL-CONF,
- modification of memory segments in the event of saturation of the application graphics or code spaces.

- **bit, word and function block objects :**

- number and parameters (function block).

- **optional function blocks (OFBs) :**

- type and number.

- **network :**

- number of common words and type of activity.

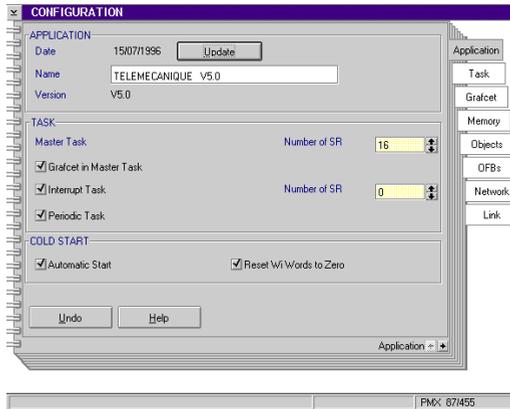
- **PLC terminal port :**

- speed, format and parity.

5.2 Configuring the application

This screen, accessed from the notepad using the **Application** tab, is used to modify the general application parameters : modification date, name and version of the application, presence of Grafcet, cold start conditions.

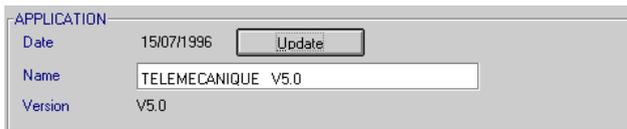
The consistency and syntax of the entries is checked by the terminal. If there is an error, a message is displayed in the status bar.



APPLICATION DATE

Used to date the creation or modification of the application. This function uses the date of the terminal.

Update assigns the current date as the application date (in offline mode only).



APPLICATION NAME

Used to define the application name which will appear on all the terminal screens and on all pages of the printed documentation file. By default this name is TELEMECANIQUE V5.0.

Modification permits entry of an application name (up to a maximum of 21 characters), in offline mode only.

VERSION

In offline mode and when creating an application, the terminal systematically offers the following version by default :

- V5.5 if the processor selected is V5 and if the application has more than 512 steps or uses the new segment configuration functions,
- V5.0 if the processor selected is V5.

In online mode (or in offline mode with an application in the terminal memory), the version displayed is that read in the application. This must be initialized beforehand by a transfer (in online mode) or a read operation (in offline mode).

MASTER TASK

Used to define, in the dialog zone, the maximum number of subroutines assigned to this task (128 maximum). By default the terminal offers 16 subroutines.

GRAF CET

Used to declare the use of Grafcet, in offline mode only. By default, Grafcet language is not used.

The screenshot shows a configuration dialog box with two main sections: 'TASK' and 'COLD START'.
 In the 'TASK' section:
 - 'Master Task' has a 'Number of SR' set to 16.
 - 'Grafcet in Master Task' is checked.
 - 'Interrupt Task' has a 'Number of SR' set to 0.
 - 'Periodic Task' is checked.
 In the 'COLD START' section:
 - 'Automatic Start' is checked.
 - 'Reset Wi Words to Zero' is checked.

INTERRUPT TASKS

Used to declare the use of the interrupt task, which has the highest priority, and to define, in the dialog zone, the number of subroutines assigned to the interrupt task (128 maximum). It is strongly recommended to keep this task as short as possible. The terminal offers 0 subroutines by default.

PERIODIC TASKS

Indicates the presence of periodic tasks (fast or auxiliary) which have been declared and their parameters set under XTEL-CONF.

COLD START

The **Automatic Start** option when activated is used to start the application program automatically on a cold restart. If this option is not activated, the application program is not executed automatically on a cold restart.

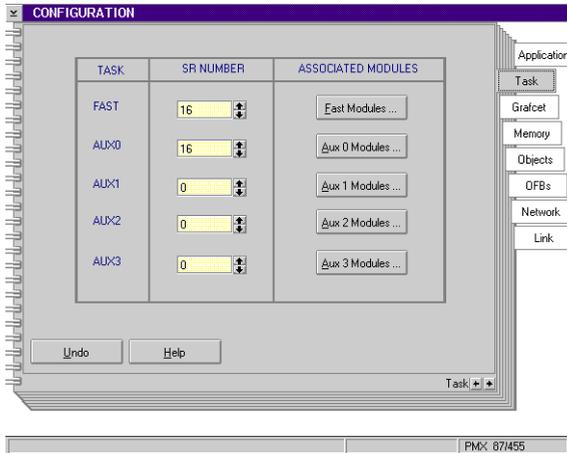
The **Reset Wi Words to Zero** option when activated is used to reset Wi and DWi words to zero on a cold restart. If this option is not activated, the Wi and DWi words remain unchanged on a cold restart.

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

5.3 Configuring tasks

This screen is accessed from the **Task** tab, and is used to define the number of subroutines for each task and to display the I/O modules associated with each task.



The **SR Number** field is used to define the number of subroutines assigned to the selected task (128 maximum per task).

Modules... accesses the screen below and is used to display the associated I/O modules.



Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

<TAB> is used to move from the "SR Number" column to the "Associated Modules" column containing the group of buttons.

<↓>, <↑> are used to move within a column or group of buttons.

<→>, <←>

<SHIFT> are used to move within the "SR Number" column.

+<↓>, <↑>

5.4 Configuring Grafcet

This screen is accessed from the **Grafcet** tab on condition that :

- the use of Grafcet is declared in the application configuration (see section 5.2),
- the memory space declared is sufficient.

Grafcet mode is used to declare :

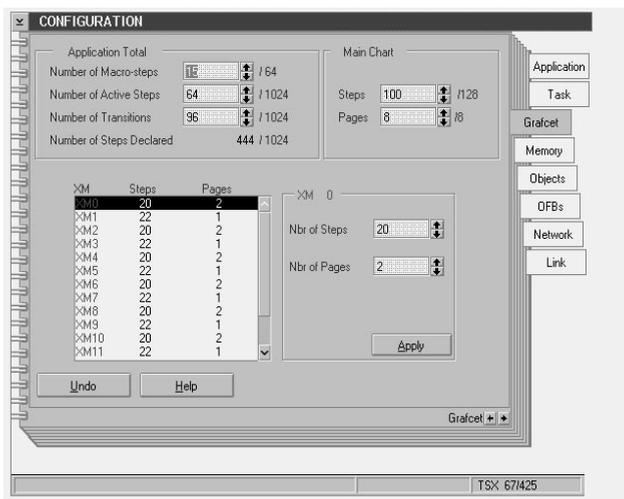
- the number of steps and pages of the main chart,
- the number of macro-steps,
- the number of steps and pages for each macro-step.

and with TSX Model 40 V5 stations :

- the number of steps which can be active at the same time,
- the number of transitions which can be enabled at the same time.

The first time that Grafcet is used in an application, the default configuration is as follows :

- CHART : 64 steps, 8 pages, 8 macro-steps,
- for each MACRO-STEP : 8 steps, 1 page.
- 64 steps active simultaneously,
- 96 transitions enabled simultaneously.



The **Number of Macro-steps** field is used to configure the number of macro-steps.

The **Number of Active Steps** field is used to define the maximum number of steps which are simultaneously active.

The **Number of Transitions** field is used to define the maximum number of transitions which are simultaneously enabled.

The **Steps (Main Chart)** field is used to configure the number of steps in the Main Chart.

The **Pages (Main Chart)** field is used to configure the number of pages in the Main Chart.

The **No. of Steps** field is used to configure the number of steps for the selected macro-step.

Reminder : the total number of configured steps must not exceed 1024 (V5.5 application) or 512 (V5.0 application maximum).

The **No. of Pages** field is used to configure the number of pages for the selected macro-step.

See section 3.13 to determine which mode should be used to modify the list parameters (Apply button)

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

<TAB> is used to move from one group to another.

<SHIFT>+<↓>, is used to move within a group.

<↑>, **<→>**, **<←>**

Notes

- a step declared in the main chart or in a macro-step occupies one bit in the bit memory (step activity bit).
- a macro-step is characterized by an input step (IN) and an output step (OUT). Irrespective of the number of steps which are assigned to it, a macro-step occupies 3 bits in the bit memory (2 bits for the IN and OUT steps and 1 macro-step activity bit).
- a word used to determine the activity time may be assigned to each step in the data memory.
- The list displayed allows multiple selection. The modifications made in the right-hand part of the dialog box therefore apply to all lines selected from the list.

5.5 Configuring the user memory

This screen is accessed from the **Memory** tab and allows the user to display the configuration of the application memory.

SPACE		ADDRESS START	VOLUMES (Words)		
All	Part		Occupied	Used	Reserved
DATA	Standard	0	12960	12953	81024
	OFB	12960	19432	19432	
PROG	Standard	81024	39504	28112	118488
	OFB	120528	75576	75572	
CNST	Standard	196104	168	161	
	OFB	196272	2960	2960	
IDP		199232	280	278	

The memory configuration menu is a simplified menu, which summarizes the memory map of the application. No data can be entered.

The memory configuration manages the following logic spaces :

- data (DATA),
- program (PROG),
- constants (CNST),
- program identifier (IDP).

Each space (data, program or constants), is divided into a space reserved for the application (STANDARD) and a space reserved for the optional function blocks (OFB).

The logic memory spaces are divided into two reserved physical spaces : a PL7-3 DATA space and a PL7-3 "proummable" space.

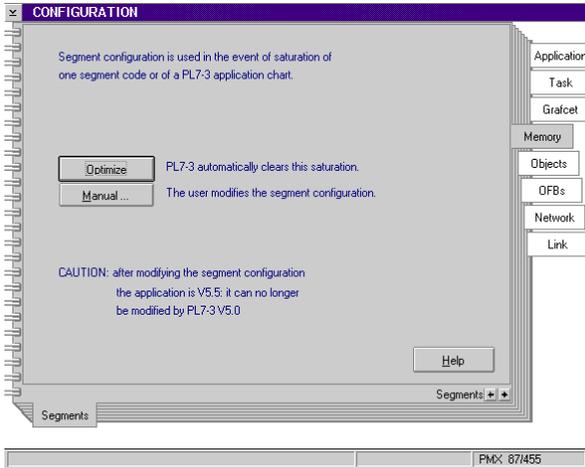
The volumes described are :

- RESERVED : attributed to PL7-3 by XTEL-CONF (or XTEL-MEM),
- USED : actually used by the logic spaces,
- OCCUPIED : USED + free space.

The memory configuration is always optimized. On each modification to the configuration other than the memory (object, application, etc), which entails a modification of the memory spaces, the system reorganizes the occupation of the memory in order to optimize it.

5.5-1 Segments

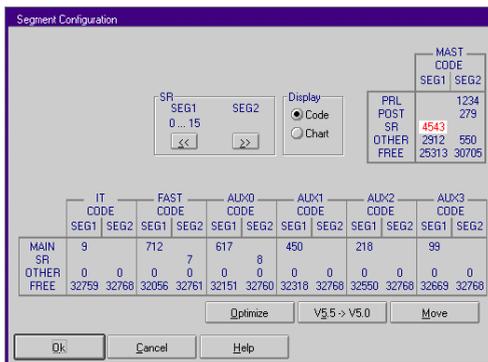
This tab, available at the bottom of the notepad window, accesses the function which is used to modify code or graphics segments in the SRI, PRL, POST or MAIN modules when saturation occurs. This is indicated to the user by the message "Memory full" (reserved for V5 applications only).



Optimize
Manual

automatically "frees memory" in the segments.

accesses the following screen for manual modification of the segment configuration.



Optimize

is used to automatically optimize the code and graphics segments.

V5.5→V5.0

configures the segments with their initial value, in order to revert to a V5.0 application.



is used to divide the SR in the current task into two parts : SR0 to SR(x-1) memorized in segment 1 and SRx to SRmax memorized in segment 2. This can only be done if SR is selected in the current task.



displays the screen with the graphics or code segments.

Move

moves the selected module from one task segment to another (code segment or graphics segment).

OK

confirms the changes made, after confirmation by **Yes** starts reconfiguration. If reconfiguration takes place after having modified the mapping of the memory segments, **this operation can take some time**. At the end of configuration, the code or graphics modules have been moved from one segment to another. A V5 application which has undergone these changes then becomes, after reconfiguration, a V5.5 application.

Cancel

quits without saving the modifications made in the dialog box.

Help

calls up the program help function.

Movement using the keyboard

<↓>, <↑>

are used to move between the different groups : MAST, IT, FAST,

<→>, <←>

AUX0, AUX1, AUX2, AUX3.

<PG UP>

is used to move from the MAST group to groups at the bottom of the

<PG DOWN>

dialog box (and vice versa).

<TAB>

is used to move between the "SR" group, the "Display" group, the ("Optimize", "V5.5 → V5.0", "Move") group of buttons, and the ("OK", "Cancel", "Help") group of buttons.

<SHIFT>+<↓>, <SHIFT>+<↑>

are used to move within the groups mentioned above.

<→>, <←>, <↔>

Note

The following screen, accessed by pressing <Alt>+<T> displays the memory occupation for each segment. This screen is accessible from all PL7-3 operating modes.

MEMORY ALLOCATION					
STANDARD PROG SPACE					
	VOLUME (WORDS)			CHART	
	RESERVE1	USED	FREE	SEG1	SEG2
CODE+GRAPH	33624	21173	12451	IT	9
SYSTEM	5880	5880	0	FAST	712
TOTAL	39504	27053	12451	MAST	7455
				AUX0	617
				AUX1	450
				AUX2	218
				AUX3	99

Note : reminder of the user memory

A PL7-3 application consists of a maximum of 7 tasks :
IT, FAST, MAST, AUX0, AUX1, AUX2 and AUX3, each split up into 2 code segments and 2 graphics segments.

For applications earlier than V5.5, the segment contents is fixed and cannot be modified :

- for a task without Grafcet
 - the first code segment contains the MAIN code objects,
 - the second code segment contains the SR code objects,
 - the first graphics segment contains graphic information concerning the task,
 - the second graphics segment contains the sources associated with the diagnostic OFBs.
- for a task with Grafcet : MAST task
 - the first code segment contains the Grafcet graphics interpreter, the code objects for the Grafcet pages, the CHART and the XM, the SR code objects, the compilation tables and the MAIN,
 - the second code segment contains the code objects for actions/transition conditions, the PRL code objects and the POST code objects,
 - the first graphics segment contains graphic information concerning the task,
 - the second graphics segment contains the sources associated with the diagnostic OFBs.

For more information, see the PL7-3 Reference Manual (TXT DR PL7-3 V52E).

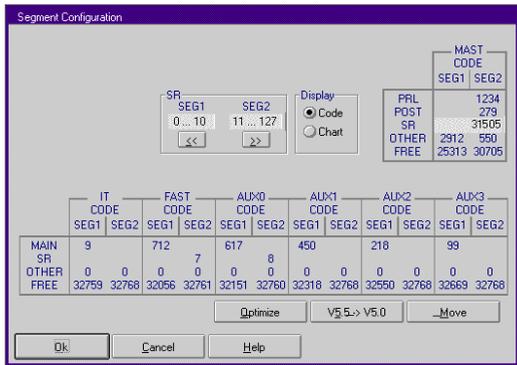
For V5.5 applications, the contents of segment 1 can be moved to segment 2 (and vice versa) **within the same task**, although the following restriction applies : only elements in bold print can be moved.

APPLICATION WITH GRAFCET GRAPHICS, MAST TASK		APPLICATION WITHOUT GRAFCET, MAST TASK	
CODE		OR	
<i>SEGMENT No1</i>	<i>SEGMENT No2</i>	ANY APPLICATION, OTHER TASKS	
GR7 INTERPRETER GR7 CODE PAGES GRAPHICS CODE XMi CODE GR7 TABLES MAIN CODE FOR SRI	ACTION CODE TRANSITION CONDITION CODE PRL CODE POST CODE	CODE	
		<i>SEGMENT No1</i>	<i>SEGMENT No2</i>
		MAIN CODE OTHER CODES	SRI CODE OTHER CODES
GRAPHICS		GRAPHICS	
<i>SEGMENT No1</i>	<i>SEGMENT No2</i>	<i>SEGMENT No1</i>	<i>SEGMENT No2</i>
OTHER GRAPHICS DATA SRI GRAPHICS DATA PRL GRAPHICS DATA POST GRAPHICS DATA	DIAG OFB SOURCE	OTHER GRAPHICS DATA MAIN GRAPHICS DATA SRI GRAPHICS DATA	DIAG OFB SOURCE

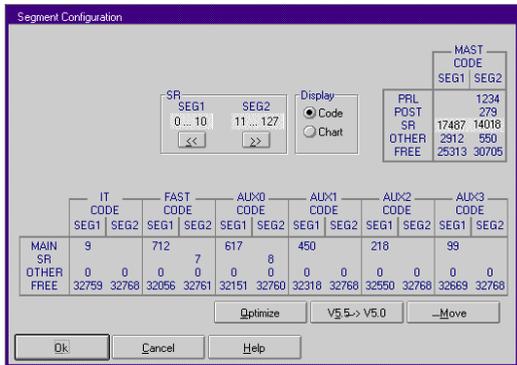
Freeing memory does not for example resolve the problem of saturation of the catalogue which occupies a segment of 32 Kwords shared with the application descriptor.

Example

New SRs cannot be entered because the segment is nearly full (SEG2 = 31505 words).



Optimize is used to split the SR code into segments 1 and 2, which provides space for the entry of new SRs.



5.6 Configuring bit, word and function block objects

5.6-1 Procedure

This screen, accessed from the **Objects** tab, is used to configure :

- the number of bits and words,
- the number of Grafcet step activity time values,
- the number of function blocks of each type.

Any modification to the number of objects necessitates a reconfiguration of the application (accessible in offline mode).

The upper part of the screen indicates the maximum number of objects and the number used :
For all applications, the number of objects used must always be less than or equal to the maximum number of objects.

The lower part of the screen is used to define the objects used. The number of these objects corresponds to the sum of the objects declared in each of the columns BITS, WORDS and FUNCTION BLOCKS.

The screenshot shows a software configuration window titled "CONFIGURATION". At the top, there are two columns: "MAX NUMBER" and "NUMBER OCCUPIED". Below these are three main sections: BITS, WORDS, and FUNCTION BLOCKS. Each section contains a list of object types with their respective counts and spinners for adjustment. The BITS section includes System (SY), Internal (B), I/O (I/O), Remote I/O, and OFB Objects. The WORDS section includes System (SW), Internal (W/DW), Const. (CW/CDW), Common A (CDM), Common B (CDMB), Common C (CDMC), Common D (CDMD), Grafcet (X), Grafcet (X,V), and Grafcet Data. The FUNCTION BLOCKS section includes Timer (T), Monostable (M), Counter (C), Register (R), and Text (TXT). On the right side, there are buttons for Application, Task, Grafcet, Memory, Objects, OFBs, Network, and Link. At the bottom, there are buttons for Undo and Help, and a status bar showing "PMX: 871455".

	MAX NUMBER	NUMBER OCCUPIED
Standard Data	38264	12953
Standard Const	11560	161
Bits	4096	3440

	BITS	WORDS	FUNCTION BLOCKS
System (SY)	128	System (SW) 128	Timer (T) 128
Internal (B)	2000	Internal (W/DW) 5000	Monostable (M) 8
I/O (I/O)	512	Const. (CW/CDW) 128	Counter (C) 8
Remote I/O	640	Common A (CDM) 0	Register (R) 0
OFB Objects	0	Common B (CDMB) 0	Text (TXT) 8
		Common C (CDMC) 0	
		Common D (CDMD) 0	
Grafcet (X)	116	Grafcet (X,V) 15	
Grafcet (X,M)	1	Grafcet Data 1050	

The only objects which can be configured by the user are :

- BITS (number) : internal bits (B);
- WORDS (number) : internal words (W,DW), constant words (CW,CDW), Grafcet step activity time values (Xi,V);
- FUNCTION BLOCKS (number) : timers, monostables, counters, registers, text blocks.

<TAB> is used to move from one column to another.

<SHIFT>+ is used to move within a column.

<↓>, <↑>

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

5.6-2 Configuring the control system function blocks

Function blocks are configured using the tabs at the bottom of the notepad. All the function blocks declared are configured with the default values, which can be read and modified in offline mode.

These values can also be modified in online mode except for :

- the time bases of timers and monostables,
- the length of registers.

Function blocks	Default parameter values
Timers	PRST = 9999, M = Y, TB = 1min
Monostables	PRST = 9999, M = Y, TB = 1min
Counters	PRST = 9999, M = Y
Registers	LENGTH = 16, TYPE = LIFO

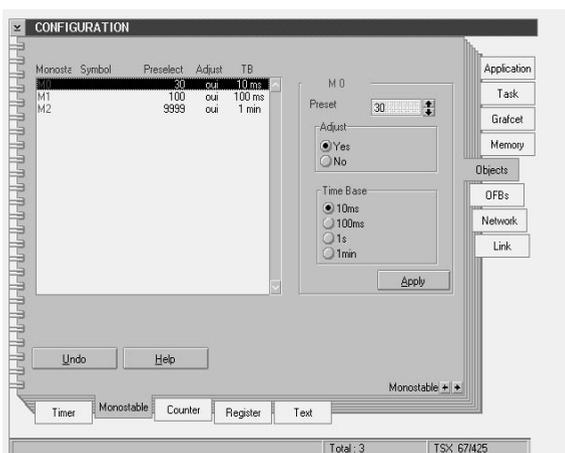
The **Preset** field is used to modify the preset value of the selected function block.

The **Yes** and **No** buttons authorize modification of the preset value in adjust mode.

The **10ms**, **100ms**, **1s** and **1min** buttons are used to select the time base. The number of timers and monostables configured with a time base of 10ms must not exceed 8.

The **Length** field (in the Register configuration) is used to define the length of the current register stack (0 to 16). Each element of the register is a word of 16 bits.

The **LIFO** or **FIFO** buttons (in the Register configuration) define the type of register selected : LIFO (Last In, First Out) or FIFO (First In, First Out).



<TAB> is used to move from one group to another.

<↓>, <↑> are used to move within a group.

See section 3.24 to determine which operating mode should be used to modify the list parameters (Apply button)

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

Note

The list displayed allows multiple selection. The modifications made in the right-hand part of the dialog box therefore apply to all the lines selected from the list.

5.6-3 Configuring the text blocks

Text blocks are configured using the tabs at the bottom of the notepad. All the text blocks declared are configured with the default values, which can be read and modified in offline and online mode.

Function blocks	Default parameter values
Text blocks	LOCAL/NETWORK = LOCAL, TYPE = CPL, ADR. MODE = DIRECT, LONG. RECEPT. = 0 M = H'00FF', L = 0, C = H'0000'

The **Loc** and **Net** buttons define the type of communication :

- Loc : exchange of data with a locally connected PLC,
- Net : exchange of data with a network connected PLC.

The **Cpl**, **Ter**, **Sys**, **Tlg**, **Txt** buttons define the type of exchange :

- Cpl : exchange of data with an intelligent module,
- Ter : exchange of data with a peripheral device, via the terminal port,
- Sys : exchange of requests with the system,
- Tlg : exchange of telegrams (priority messages) with a Tlg block in another PLC.
- Txt : exchange of data with a Txt block in another PLC.

The **Dir** and **Ind** buttons define the type of addressing :

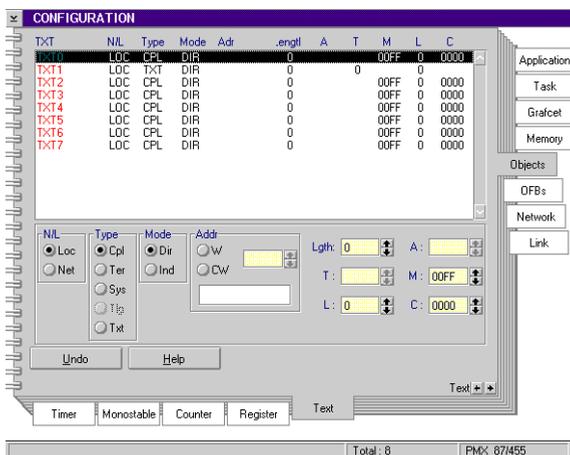
- Direct : the table to be transmitted or received is defined by its start address and its length
- Indirect : the tables to be transmitted or received are defined by an addressing table of 6 words.

The **addressing** field is used to enter in the operator input line :

- in direct addressing, the address of the first word of the table to be transmitted or received,
- in indirect addressing, the address of the first word of the addressing table.

The **Length** field is used to enter (in direct addressing only) the length (in bytes) of the reception table. The maximum length of the reception table depends on the type of exchange :

- 30 bytes for an exchange via TELWAY,
- 256 bytes for an exchange via MAPWAY,
- 256 bytes for an exchange via ETHWAY,
- 128 bytes for an exchange via FIPWAY,
- 254 bytes for a CPL type exchange,
- 126 bytes for a SYS type exchange,
- 126 bytes for a TER type exchange,
- 254 bytes for a TXT type exchange,
- 32 bytes for a TLG type exchange.



The **A** field is used to enter, for a NET type exchange, the number of the destination station. The **T** field is used to enter the number of the destination text block (exchange with a TXT block). The **M** field is used to enter the address of the destination module (exchange with a CPL block). The **L** field is used to enter, in direct addressing, the maximum length of the transmission table (in bytes). The maximum length is the same as that of the reception table. The **C** field is used to enter the code of the type of work to be done (exchange with a CPL or SYS block).

- <TAB> is used to move from one group to another.
- <↓> <↑> are used to move within the groups.
- <SHIFT>+ <↓>, <↑> are used to move from one incremental entry zone to another (the <↓>, <↑> being used in the operation of this incremental entry zone).
- Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).
- Help calls up the program help function.

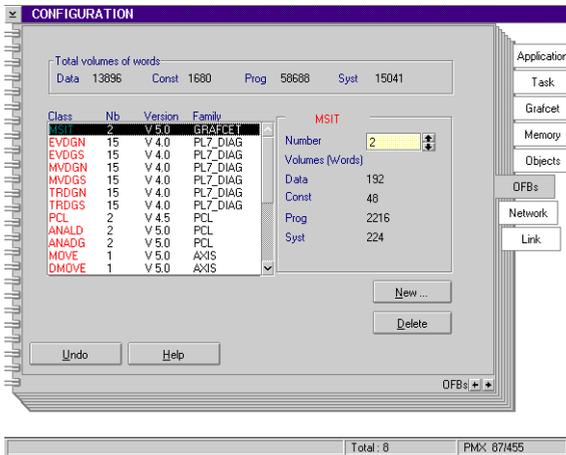
5.7 Configuring the optional function blocks

This screen is accessed from the **OFBs** tab. It is used to configure the optional function blocks (OFBs) used in the application :

- selection of the types of OFB to be used in the application, from those available in the terminal,
- definition of the number of OFBs of each type selected.

The OFBs available in the terminal are stores on the hard disk by family.

Any modification to the configuration of the OFBs necessitates a reconfiguration of the application.

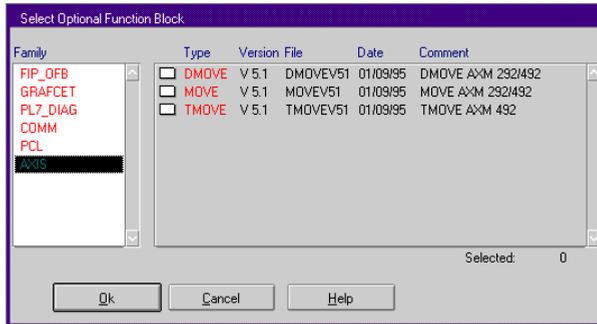


New is used to list the types of OFB available in the terminal in order to select those to be used in the application.

The types of OFB available in the terminal are displayed by family (communication, axis control, etc).

Delete deletes the type of OFB selected.

The **Number** field assigns the number of OFBs to the selected type.



Families and types are selected from the lists in the screen above. Once a family has been selected, the **OK** button confirms the selection.

<TAB> is used to move from one group to another.

<↓>, <↑> are used to move within a group.

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

5.8 Configuring the network services

This screen, accessed from the **Network** tab, is used to configure the number of common words and the type of activity.

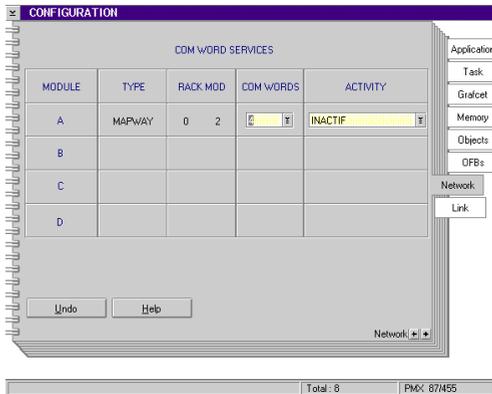
As this depends on the number, location and type of network modules configured, the network services cannot be configured until after the I/O configuration has been declared.

In a multinetwork station, the common words associated with the various network modules are designated by COM, COMB, COMC and COMD, in the geographic order of the position of the network modules (rack and slot number).

FIPWAY Modules

- In V52, a FIPWAY module is integrated into the PLC processor.
- In V6, FIPWAY rack-mounted modules can be used as well as an integrated module.

The FIPWAY common words of the FIPWAY module integrated into the PLC processor always occupy the last "geographic" position, ie after the last rack-mounted module. When a rack-mounted module is inserted or removed from the I/O configuration, the "geographic" position of the integrated FIPWAY "module" is therefore modified.



The **Com Words** field defines the number of common words associated with the selected module : 0, 4, 8, 16, 32 or 64.

The number of common words authorized depends on the type of module (TELWAY, MAPWAY, ETHWAY or FIPWAY) and the number of stations on the network.

The **Activity** field defines the type of activity of the common words :

- inactive,
- read/write,
- read only.

<TAB> is used to move from one module to the other.

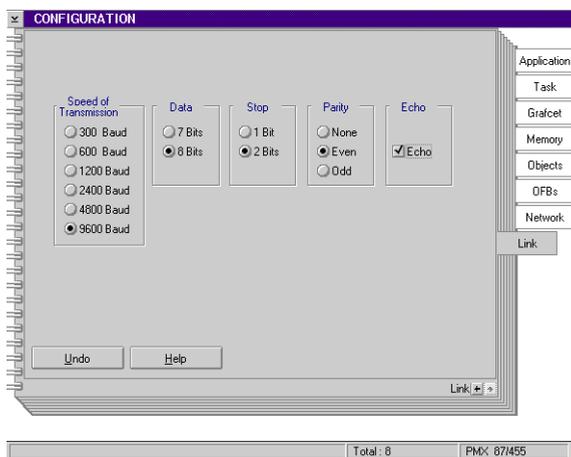
<→>, <↔> are used to from one column to the other.

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

5.9 Configuring the link

This screen, accessed from the **Link** tab, is used to configure the parameters of the PLC terminal link when it is not connected to a terminal or to TELEMECANIQUE TSX7 software.



The **300, 600, 1200, 2400, 4800** and **9600 bauds** buttons define the transmission speed.

The **7** and **8 Bits** buttons define the number of data bits.

The **1** and **2 Bits** buttons define the number of stop bits.

The **None, Even,** and **Odd** buttons define the parity.

The check box defines a transmission with or without **Echo**.

<TAB> is used to move from one group to the other.

<↓>, <↑> is used to move within a group.

Undo is used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

5.10 Modifying the configuration in online mode

The structural parameters of the application which are necessary to debug a program in the PLC memory, and which do not cause any inconsistency, can be modified in online mode.

The configuration parameters affected by this function are :

- the start-up parameters (auto start, reset Wi to zero),
- the function block parameters, except for the time bases of timers and monostables and the length of registers.

Modifying function block parameters

All the parameters of the function blocks already configured by the user, or by default, can be modified in online mode (except for the time bases of timers and monostables and the length of registers) :

- from CONFIGURATION mode, entered when the PLC is stopped,
- from DEBUG mode or PROGRAM mode.

In this case the procedure differs according to the type of language :
See section 8.

Important

- none of the modifications made to the configuration by the user in online mode are taken into account UNTIL the PLC is initialized.
- this initialization must be performed with the PLC stopped. To initialize the PLC, select the DATA or DEBUG mode and then set system bit SY0 to 1. This is equivalent to a cold restart.
- this bit is AUTOMATICALLY reset to 0 by the system.

5.11 Reconfiguration function

5.11-1 Procedure

The reconfiguration function is used, in offline mode, to validate a new application configuration. After reconfiguration, all the modifications declared are transferred to the internal memory of the terminal.

The reconfiguration procedure must be performed with care and, in the case of an extension to the configuration, only after checking that there is enough space available in the Program memory space to write the program.

<Alt>+<T> display, irrespective of the mode, the amount of memory space available for writing the program.

Notes

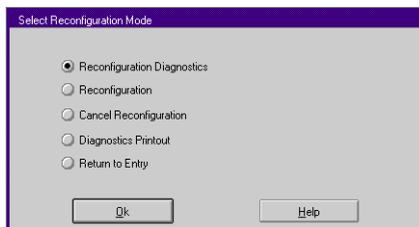
- before making any modification to the configuration, users are recommended to save the existing application.
- Before a reconfiguration operation, users are recommended to perform the reconfiguration check proposed by the terminal (RECONFIGURATION CHECK function).

The reconfiguration function is accessible once the configuration notepad window has been closed, confirmation (VALIDATION) is required to access the RECONFIGURATION MODE SELECTION window.



Yes confirms all modifications made to the configuration, which may necessitate a reconfiguration of the application. In this case, it is advisable to first perform a reconfiguration check.

No cancels the modifications made.

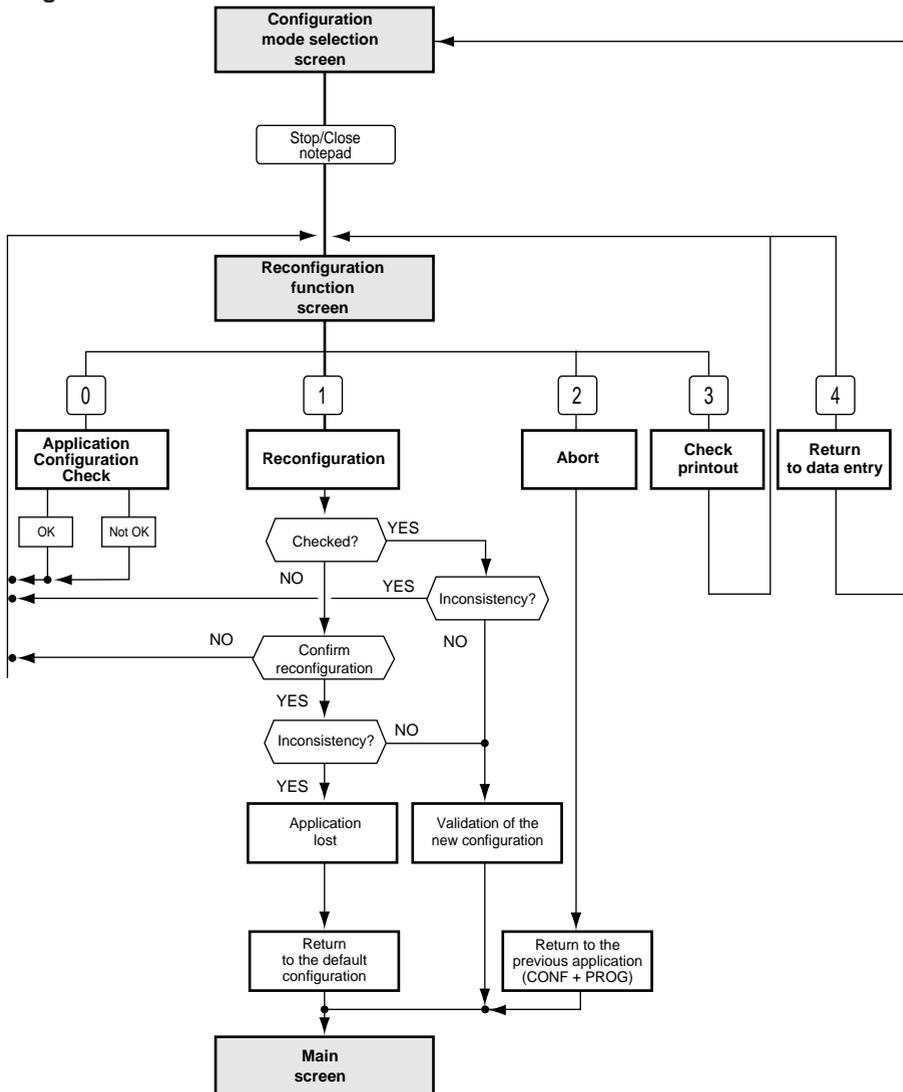


The screen offers five options :

- 0 - RECONFIGURATION CHECK (section 5.11-2),
- 1 - RECONFIGURATION (section 5.11-3),
- 2 - ABORT RECONFIGURATION (section 5.11-4),
- 3 - RECONFIGURATION CHECK PRINTOUT,
- 4 - RETURN TO DATA ENTRY.

For safety, by default the terminal offers the reconfiguration check.

Diagram



5.11-2 Reconfiguration check

This function, accessible from the RECONFIGURATION MODE SELECTION window, is used to check consistency between the user program and the new configuration.

At the end of the check, if the new configuration is correct, the terminal displays the message : 16133 CHECK OK. The user can then launch the reconfiguration.

If any inconsistencies are detected between the existing user program and the new configuration, the checking procedure is interrupted. The error is displayed on the screen with the window CONFIGURATION ERROR (for example ERROR ON CHECK).



Next restarts the check until the next error is encountered or until the check ends by returning to the RECONFIGURATION MODE SELECTION screen with the message : 16134 PROG AND CONF MISMATCH.

Abort stops the check and returns to the RECONFIGURATION MODE SELECTION screen with the message : 16134 PROG AND CONF MISMATCH.

Reconfiguration is prohibited if any inconsistencies are found between the user program and the new configuration.

The inconsistencies in the application must be corrected before a new reconfiguration can be attempted. This can be done by either :

- modifying the program to adapt it to the new configuration, or
- modifying the new configuration.

5.11-3 Reconfiguration

This function is used to transfer the new configuration into the memory reserved for the application.

Two options are available :

- check has not been made,
- check has already been made.

Check not made

If reconfiguration is successful, the terminal returns to the main window and displays the message 16136 RECONFIGURATION OK.

If there is a discrepancy between the program and the new configuration, the first discordant program element is displayed on the screen with a message (see reconfiguration check).

At the end of the operation the following message appears :

CONFIGURATION FAILED. RETURN TO MAIN MENU WITH DEFAULT APPLICATION.

The application is lost

Check made

If a discrepancy detected during the check has not been corrected, reconfiguration is impossible. Otherwise, once reconfiguration is complete, the terminal returns to the MODE SELECTION screen with the message 16136 RECONFIGURATION OK.

If reconfiguration is performed after modifying the memory segment mapping, **this operation can take some time**. On completion of configuration, the code or graphics modules will have been moved from one segment to another.

A V5.0 application with modules which have been moved then becomes a V5.5 application after reconfiguration.

5.11-4 Abort reconfiguration

This function is used to abort the reconfiguration, which cancels all the modifications and returns to the old configuration.

When this operation is completed, the terminal displays the MAIN PL7-3 screen.

5.11-5 Reconfiguration check printout

This function is used to print all the errors generated by the reconfiguration check to a file or printer.

5.11-6 Return to data entry

This function is used to return to the configuration. It is also used to correct the parameters which caused an error.

Section	Page
6.1 Changes made since V52	6/2
6.2 Selecting the task and module	6/3
6.2-1 Selecting the program address	6/3
6.2-2 MODULE SELECTION screen	6/4
6.3 Reading the program	6/6
6.3-1 Reading the pages of a Grafcet module	6/6
6.3-2 Menu bar	6/7
6.3-3 Toolbar	6/9
6.3-4 Searching for a step in a module	6/9
6.3-5 Archiving and retrieving a Grafcet module	6/10
6.4 Grafcet zoom function	6/12
6.4-1 Zooming in on a macro-step	6/13
6.4-2 Zooming in on a transition	6/13
6.4-3 Zooming in on a step	6/14
6.5 Creating or modifying the program	6/15
6.5-1 Principle of data entry	6/15
6.5-2 Entering a linear sequence	6/18
6.5-3 Entering simultaneous sequences	6/19
6.5-4 Entering a connector	6/19
6.5-5 Drawing paths	6/20
6.5-6 Entering comments	6/22
6.5-7 Confirming a page	6/23
6.5-8 Modifying the elements in a page	6/23
6.5-9 Deleting a module, a page or an element	6/25
6.6 Multiple selection	6/27
6.6-1 Presentation	6/27
6.6-2 Moving graphic elements	6/27
6.6-3 Deleting graphic elements	6/28

6.1 Changes made since V52

To access the Grafcet editor using the keyboard :

- activate PROGRAM BROWSER by pressing **<ALT>+<F1>**,
- using the **<TAB>** key, move the cursor to the Task column,
- select a task using the up and down arrow keys,
- select a module and a page in the same way,
- move the cursor to the Task column and confirm selection of the program address by pressing **<ENTER>**.

To access the Grafcet editor using the mouse :

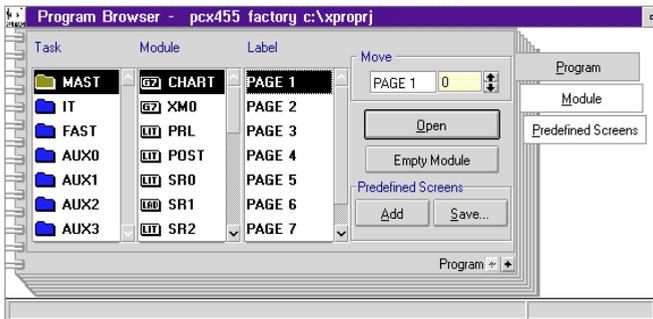
- activate PROGRAM BROWSER by clicking on the Program icon in the MAIN WINDOW toolbar,
- select the program address by clicking on the task, module and page required,
- confirm the selection by double-clicking on the task selected.
- To modify a page, press **<F6>** or use the **Modify Page** function in the **Edit** menu, or click on the modification icon.
- To change to ZOOM mode :
 - use the **<CTRL>+<PAGE DOWN>** keys,
 - click on the zoom icon,
 - use the **Grafcet Zoom** function in the **Edit** menu,
 - double-click in the editor.

6.2 Selecting the task and module

6.2-1 Selecting the program address

Before displaying an existing program or entering a new one, its address must be defined (by selecting the task, module, and page). Also entering a program in Grafcet means that this language must have been defined in CONFIGURATION mode.

The PROGRAM BROWSER screen is the initial screen in PROGRAM mode, and can be accessed using the **Program** option in the **Tool** menu, or by pressing **<Alt> + <F1>**. When the application is launched this is the default screen.



A task and its subsets are selected in the Browser using the **<↑>****<↓>** keys to select the element required, and the **<←>****<→>** keys to move between columns.

Selecting the task

Only the master task can be used for programming in Grafcet language.

Selecting the module

Permits selection of the master task program module from 5 possible options :

- CHART : main chart,
- XM... : macro-step,
- PRL : preprocessing,
- POST : postprocessing,
- SR... : subroutine.

The main **Module** tab on the right of the notepad, accesses the MODULE SELECTION screen which is the call screen for the basic modules and macro-modules of the application.

Selecting the page

The page is selected in the label column of the PROGRAM BROWSER.

Open is used to open the selected network.

6.2-2 MODULE SELECTION screen

This screen can be accessed from the main **Module** tab and contains a list of the basic modules and macro-modules in the application (for more information, see section 15.9 V5 Modular version). It is used to gain direct access to :

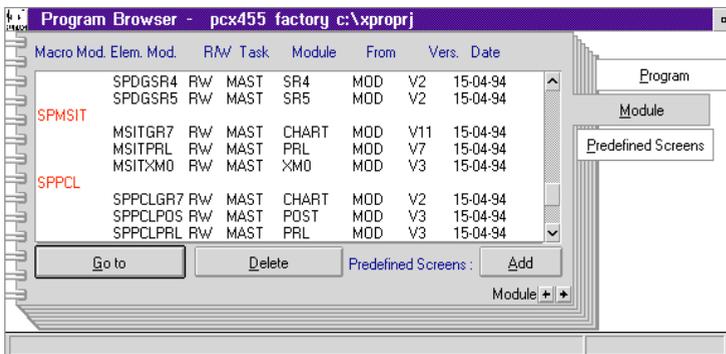
- the start of the code of this module, if a basic module is selected,
- the graphic display of this module, if a macro-module is selected.

It shows the following information for each basic module :

- name of the macro-module with which it is associated (if relevant),
- name of the basic module,
- level of protection :
 - RW : no protection (reading and writing are possible),
 - R : write-protected (only reading is possible),
- program address : task and module,
- source : XTEL-MOD or PL7-PMS2,
- version, defined in XTEL-MOD,
- date of creation or last modification, defined under XTEL-MOD.

This information is sent by the STATION.M5M file and stored in an internal PL7-3 table : application module table. Since this table is an integral part of the PL7-3 application, its size has to be taken into account in the xxx.BIN table. It is therefore stored in the PLC.

Basic modules and macro-modules should never be displayed on the same line, so that they can be selected separately. They are ranked in alphabetical order and those basic modules which are not associated with a macro-module are grouped together at the beginning of the first screen.



Go to	confirms the selection.
Delete	deletes, after confirmation, the basic module or macro-module indicated by the cursor. A write-protected basic module cannot be deleted if it is part of a macro-module. In this case, delete the macro-module. If the basic part is not part of a macro-module, it can be deleted.
Add	adds the current selection (tasks, modules, and possibly a page) to the DEBUG list.
<TAB>	used to move from one group to the other.
<↑><↓>	used to move within a group.

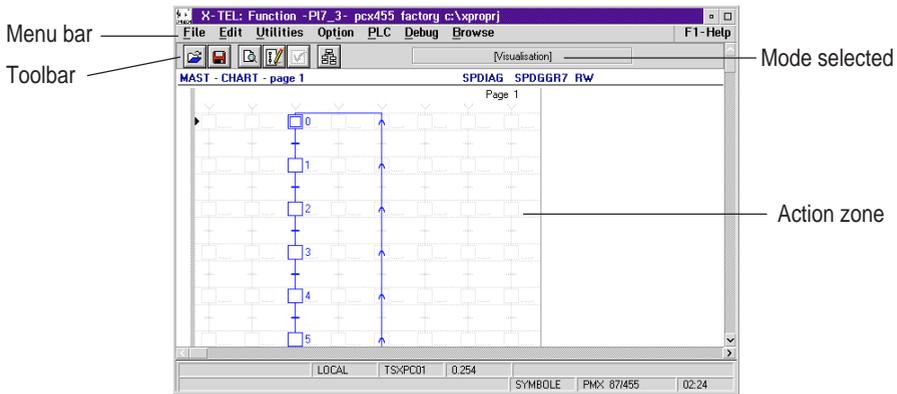
6.3 Reading the program

6.3-1 Reading the pages of a Grafcet module

Grafcet DISPLAY screen

This screen can be accessed from the PROGRAM BROWSER screen by confirming the selections made (task, module, page).

A Grafcet module is composed of several pages (a maximum of 8 can be configured in CONFIGURATION mode). Each page, numbered from 1 to 8, is shown on the screen.



Direct access to a page

	<CTRL>+<Home>	used to move to the first page of the module.
	<CTRL>+<END>	used to move to the last page of the module.
	<Page Up>, <Page Down>	in addition to the cursor, scroll bars are used to move within the program.
Edit, Go to → page		used to directly access another page of the module. After confirming with <ENTER> this page is displayed.
File, Close	<CTRL>+<F4>	used to close the editor and return to the PROGRAM BROWSER screen.

6.3-2 Menu bar

FILE MENU :

Open	opens a dialog box which can be used to select the file to be retrieved.
Save	archives the current Grafcet module in a "source" file on disk.
Close	closes the file currently on screen, and brings up a dialog box in order to save modifications after confirmation.
Exit	quits GRAFCET mode.

EDIT MENU :

Delete	deletes the element(s) selected.
Go to → Page → Step → Macro-step	goes to a page number. goes to a step. goes to a macro-step.
Select Objects	selects one or more objects.
Modify Page	activates modification mode.
Grafcet Zoom	activates Zoom mode.
Confirm	confirms modifications made.
Cancel	in MODIFY mode this cancels any modifications made and returns to the original page (prior to modification).
Actions → Activation → Continuous → Deactivation	selects the type of action to be programmed/displayed. action on activation. continuous action. action on deactivation.
Activity time	assigns an activity time to a step.
Language → Ladder Language → Literal Language	enables the action to be entered in Ladder language. enables the action to be entered in Literal language.

UTILITIES MENU :

Animate	confirms the option to refresh the screen in realtime.
Freeze	freezes animation of the screen to allow a situation to be analyzed.
Edit Program Comment	accesses the program comment on the GRAFCET page.

UTILITIES MENU (CONT'D) :

Edit Symbols	launches the XTEL-SDBASE tool.
---------------------	--------------------------------

OPTION MENU :

Toolbar	displays or hides the toolbar.
Cancel Help Balloons	activates or deactivates the toolbar help balloons.

PROCESSOR MENU :

Run	launches execution of the program.
Stop	stops execution of the program.

DEBUG MENU :

Start Task	launches or stops execution of the task.
Task Monitored	selects the task to be executed and monitored.
Breakpoints → Insert → Remove → View	accesses PROGRAM BROWSER to define the task, module and program element on which the breakpoint can be inserted. not possible in GRAFCET mode. not possible in GRAFCET mode.
Continue	not possible in GRAFCET mode.
Step by step	not possible in GRAFCET mode.
Last Stop	displays the last element executed.
Data Mode	accesses DATA mode.
Data Line mode	accesses DATA LINE mode.

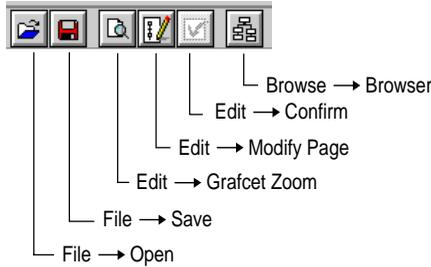
BROWSE MENU :

Data Window	accesses the data line window.
BROWSER	displays the Program Browser window.
Screen 1...6	switches from one predefined screen to another.

6.3-3 Toolbar

This toolbar provides quicker access to the main functions on the menu bar in GRAFCET mode.

Defining the buttons on the toolbar :



6.3-4 Searching for a step in a module

Searching for a step

Edit, Go to → Step		is used to search for a step in a module (CHART or XM).
	<ENTER>	confirms the entry and displays the page containing the required step.

Searching for a macro-step

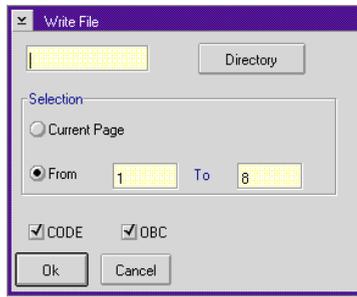
Edit, Go to →Macro-step		is used to search for a macro-step in a module (CHART or XM).
	<ENTER>	confirms the entry and displays the page containing the required macro-step.

6.3-5 Archiving and retrieving a Grafcet module

Archiving

File, Save	<CTRL>+<S>	is used to archive the current Grafcet module in a "source" file on disk. This xxx.GR7 file, saved in the PL7_3\MOD directory, allows the module to be reused in another application and integrated in a module base.
-------------------	-------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The following window is displayed :



The entry field is used to define the file name.

The **Directory** button is used to access the list of files in the save directory.

The **Current Page** button is used to store the current page.

The **FROM** button is used to select a page to save.

The **CODE** check box is used to archive the Grafcet module, with or without the code for the associated actions and transition conditions.

The **OBC** check box is used to archive the Grafcet module, with or without the internal OFB constants. If the OBC option is selected, a file <name_OFB>.OBC is created for each module OFB which contains internal constants. In addition, an xxx.OBM file (with the same name as the xxx.GR7 file) provides a list of all OFBs which have generated an .OBC file.

Help calls up the program help function.

<TAB> used to move from one group to the other.

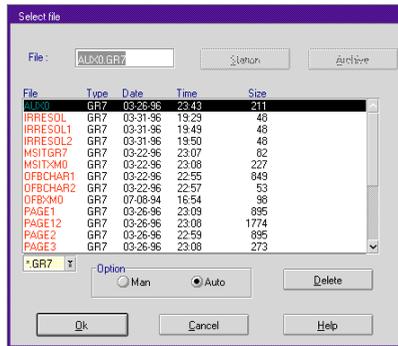
<↑><↓> used to move within a group.

<SPACE> used to activate the buttons and check boxes.

Retrieving

File, Open	used to retrieve a Grafcet "source" file to the working RAM memory of the terminal.
-------------------	-------------------------------------------------------------------------------------

The following window is displayed :



The entry field is used to define the file name.

The list of files is used to select a file to be retrieved from those in the current directory.

The **AUTO** and **MAN** buttons define the retrieval mode :

AUTO : global retrieval with stop on error only,

MAN : manual retrieval with confirmation on each element read.

Archive assigns the global archive name (PL7_3) as the file name.

Station assigns the name of the current station as the file name.

OK saves the selection made and starts retrieving the file selected.

Cancel cancels the selections made and exits the dialog box without triggering any action.

Help calls up the online help function.

<TAB> used to move from one group to the other.

<↑><↓> used to move within a group.

<→><←>

<↑><↓> used to move in the list or in the inset.

Notes

For more information on the V5 modular version, see section 15 - Appendix.

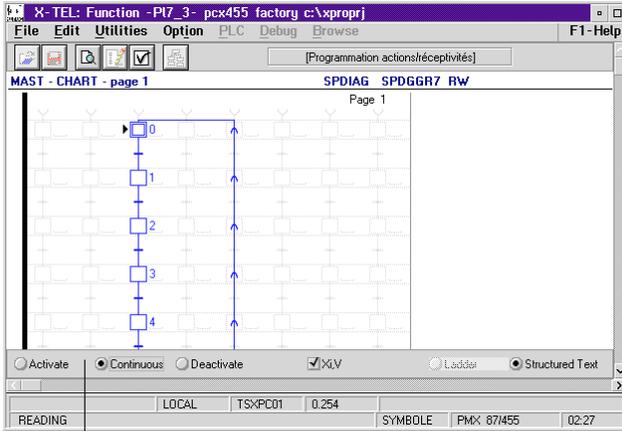
A Grafcet "source" file can be retrieved in online or offline operation with the PLC in STOP mode, but it cannot be accessed with the PLC online in RUN mode.

A CHART module "source" file cannot be retrieved to an XM module and vice versa.

6.4 Grafcet zoom function

The Grafcet zoom function is used, from the DISPLAY screen, to access the reading, writing or modification of :

- the expansion of a macro-step,
- a transition condition associated with a transition,
- an action associated with a step.



Actions and transition conditions function palette

Procedure for using the Grafcet zoom function

The procedure is the same for a macro-step a transition or a step. Once the terminal is in GRAFCET ZOOM mode, modification mode can be accessed.

<p>Edit, Grafcet Zoom</p>	<p><CTRL> + <PG DOWN></p>	<p>activates the GRAFCET ZOOM function. In this mode a transition which has not been programmed is displayed in reverse video. In order to execute the Grafcet chart, the transition is considered as always being true. A cursor appears on the screen to select the step or transition to be "zoomed" into. Using this function again switches the terminal to DISPLAY mode. The terminal then displays the contents of the element selected by the cursor (expansion of a macro-step, contents of a transition condition or action). Pressing these keys again returns the terminal to GRAFCET ZOOM mode. This operation can be repeated in order to descend through the various levels of the program.</p>
--------------------------------------	--------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.4-1 Zooming in on a macro-step

ZOOM mode is activated and the cursor is positioned on a macro-step,

	<SPACE> or <CTRL>+<PG DOWN>	displays the expansion of its macro-module. Pressing these keys again will repeat this operation down through 64 hierarchical levels (64 consecutive macro-steps).
--	----------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

This procedure is used to access an empty macro-step, so that its expansion can be programmed (see section 6.5).

6.4-2 Zooming in on a transition

GRAF CET ZOOM mode is activated and the cursor is positioned on a transition. If the transition condition has not yet been programmed, the programming language must be defined : Literal or Ladder language.

The transition condition has not been programmed

The **Ladder** button is used to program the transition condition in Ladder language.

The **Literal** button is used to program the transition condition in Literal language.

	<SPACE> or <CTRL>+<PG DOWN>	accesses the transition condition.
--	----------------------------------------------------------------	------------------------------------

The transition condition is already programmed

	<SPACE> or <CTRL>+<PG DOWN>	accesses the transition condition.
	<DElete>	deletes the transition condition.

6.4-3 Zooming in on a step

ZOOM mode is activated and the cursor is positioned on a step.

Selecting the type of action

The **Activation** button activates the selected step.

The **Continuous** button is used for a continuous action.

The **Deactivation** button deactivates the selected step.

Entering an action

The **Literal** button is used to enter the action in Literal language.

The **Ladder** button is used to enter the action in Ladder language.

	<SPACE> or <CTRL>+<PGDOWN>	accesses the action. The terminal then displays an empty module. The entry obeys the syntax rules of the language. This operation can also be performed by double-clicking on the element to be modified.
--	---------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Reading an action

Having selected the type of action (the step having already been programmed) :

	<SPACE> or <CTRL>+<PGDOWN>	accesses the action. This operation can also be performed by double-clicking on the element to be modified.
--	---------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------

Assigning an activation time to the step

Within the limits set during configuration, the terminal can be used to assign an activity time value to the step highlighted by the cursor.

XIV is only present if at least one active time was declared in the configuration. It is used to alternately assign or not assign an active time to the step (flag XIV is selected “absent” or “present” using the selection button).

If a step does not have an activity time, it can be read but its current value remains frozen at -1.

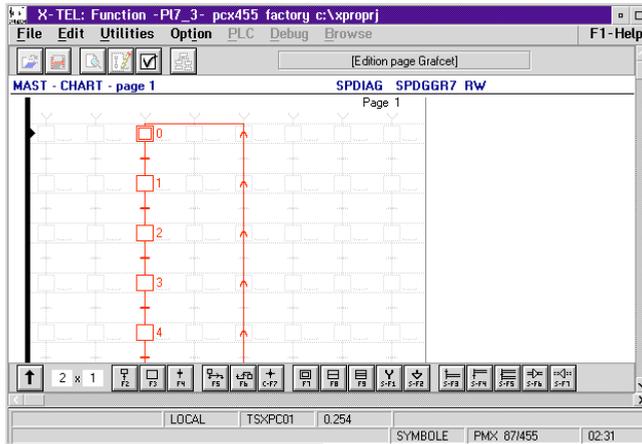
6.5 Creating or modifying the program

6.5-1 Principle of data entry

The procedure for entering a new program or for modifying an existing program is as follows :

- define the address (task, module and page) of the program to be entered or modified. This accesses the Grafcet DISPLAY screen,
- activate the MODIFY mode via the **Edit** menu, **Modify Page** option or **<CTRL>+<m>**. The page displayed can then be modified.

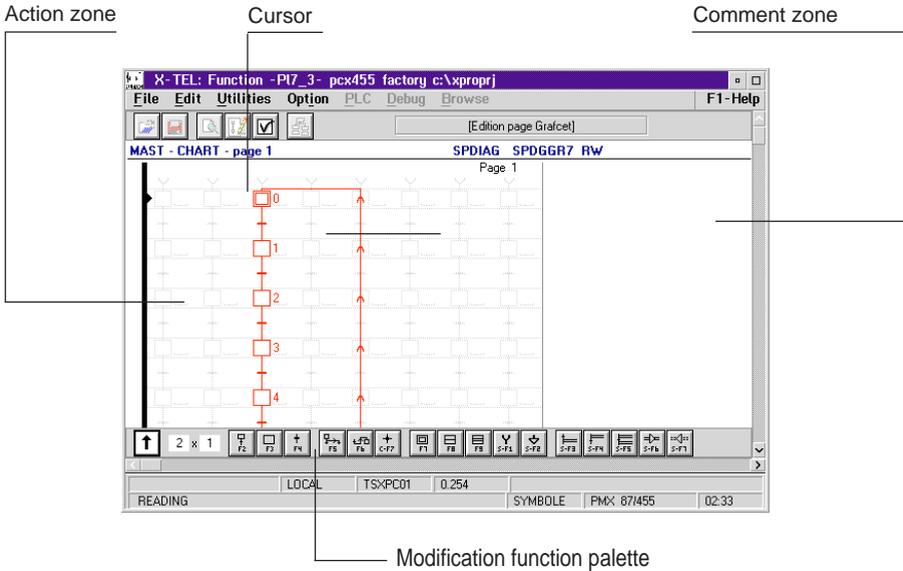
A Grafcet page can be modified in offline or online mode. However modification with the PLC running is impossible, irrespective of the processor used.



	<DELeTe>	deletes the element to the right of the cursor.
	<ENTER>	every page entered must be confirmed so as to be compiled and stored in the memory. The terminal returns to DISPLAY mode.
	<SPACE>	is used to modify the number of a step or macro-step.

Grafcet MODIFICATION screen

The following screen (blank page) appears when a new program is being entered.



The cursor can be moved anywhere on the screen Grafcet using the mouse or the $\leftarrow \uparrow$, $\leftarrow \downarrow$, $\leftarrow \rightarrow$ and $\leftarrow \leftrightarrow$ keys.

The $\leftarrow \text{PG UP}$ and $\leftarrow \text{PG DOWN}$ keys are used to move several lines at a time on the screen.

The $\leftarrow \text{HOME}$ key is used to move to the start of the current page (the page containing the cursor).

Elements are entered using the palette keys.



is used to deselect the function key used.

It is then possible to draw :

- linear sequences in Grafcet charts or subsets (see section 6.5-2),
- simultaneous sequences (see section 6.5-3),
- sequence selections which are defined by paths. Two buttons are used for this :

transition → step path,

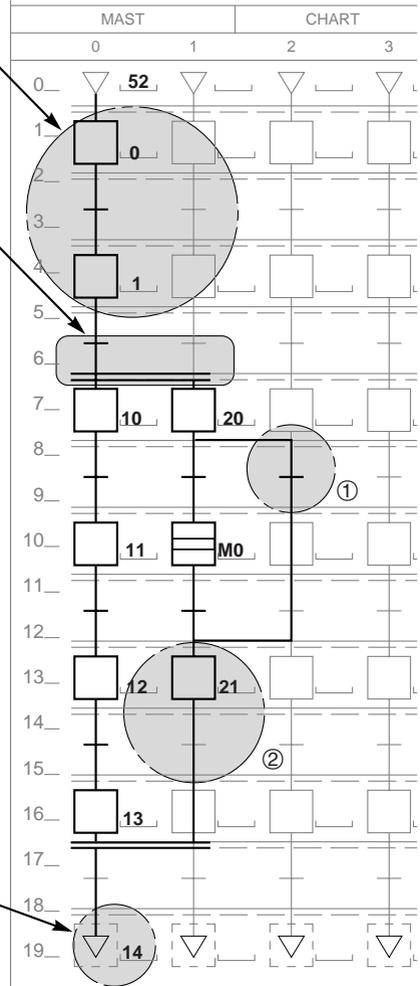


step → transition path.



The entry of paths is described in section 6.5-5.

- destination or source connectors (see section 6.5-4).



It is also possible to enter comments associated with the Grafcet chart (see section 6.5-6)

6.5-2 Entering a linear sequence

The procedure for entering a linear sequence is the same whether it is in the main chart or in a macro-step.

Entering a step

This key enters the graphic element and the step number can then be entered by :



- pressing soft key **<F3>** if the number offered is correct,
- pressing the numeric keys, if the number is not correct. This new number entered must be confirmed by **<ENTER>**.

Entering a transition



Enters the graphic element. In the case of a linear sequence, the transitions can be entered directly by pressing soft key **<F4>**.

Fast entry of a linear sequence



It is possible to program a linear sequence whose step numbers are in ascending order.

The step number is given by the last step number modified or entered plus one, if it is free, or by the last free number.

If a step number is not correct, it is still possible to enter the required number and confirm it with **<ENTER>**.

Entering a macro-step

The procedure is the same as for a step but the number of the macro-step must be entered systematically by the user and confirmed with **<ENTER>**.



Enters the graphic element for the mono-active macro-step.



Enters the graphic element for the multi-active macro-step.

6.5-3 Entering simultaneous sequences

The terminal does not impose any particular order when entering the program, except of course the alternation of steps and transitions.



Enters the transition from one step to several steps (simultaneous activation transition).



Enters the transition from several steps to one step (simultaneous deactivation transition).



Enters the transition from several steps to several steps.



This key does not appear unless the cursor is positioned at the transition for simultaneous activation or deactivation of steps. It is used to extend the two activation or deactivation lines.



Deletes an extension which is too long.

6.5-4 Entering a connector

Source connector

This can be entered on lines 1, 4, 7, 10, 13, 16 and 19 and is linked to a step.



Enters the graphic element for the source connector and enters the number of the source step or macro-step by :

- pressing the soft key **<F1>** if the number suggested is correct (number of the last step entered),
- pressing the numeric keys if the number suggested is not correct. The new number entered must be confirmed with **<ENTER>**.

M is used to enter the macro-step number. It must be confirmed with **<ENTER>**.

Destination connector



Enters the graphic element for the destination connector. The procedure for entering the number is the same as for the source connector.

6.5-5 Drawing paths

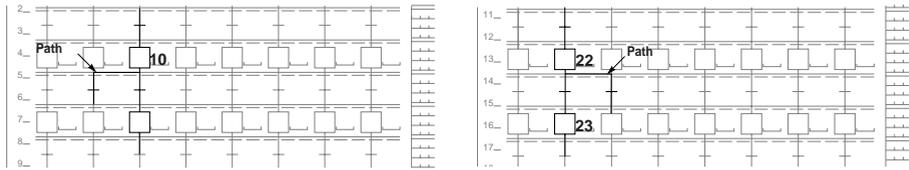
Paths are used to extend directed links connecting steps to transitions or transitions to steps. They can be extended upwards and downwards, or to the left and right.

The terminal offers two types of path :

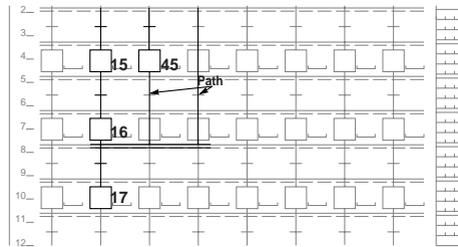
Step - transition path

This is used to :

- draw the start of a sequence selection to the right or to the left,



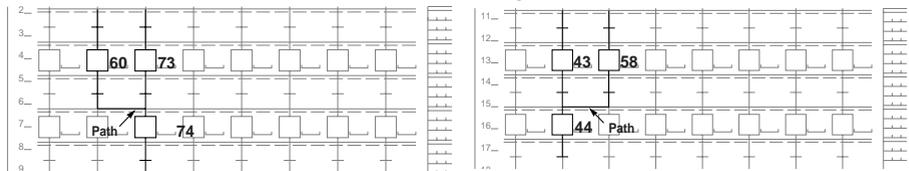
- draw a simultaneous step deactivation prior to a transition.



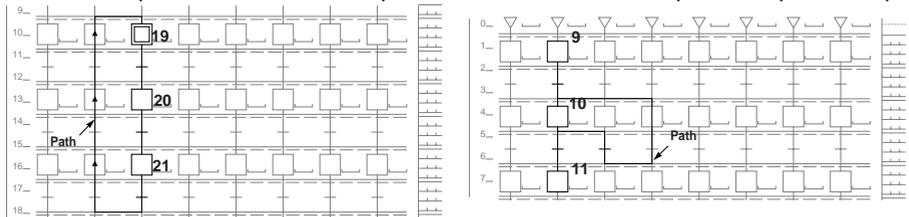
Transition - step path

This is used to :

- draw the end of a sequence selection to the right or left,



- draw a loop from the end of a sequence back to the initial step, or to repeat a step.



Keys used

The keys offered are the same for both types of path and perform the same drawing functions.



Step - transition path : this key is offered by the terminal on transition lines.



Transition - step path : this key is offered by the terminal on step lines.



Used to draw paths.



These can be drawn using the mouse. Every click with the mouse causes a path line to be drawn between the last and current cursor positions.

An arrow indicates the line which is confirmed by clicking on the right-hand mouse button.

<ESCAPE>

Exits the working mode and returns to MODIFY mode. If a line is being entered, the whole line is deleted.

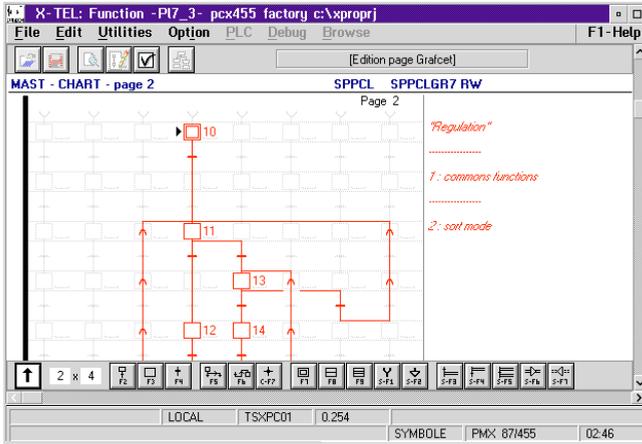
Note

When a step is deleted, the whole path downstream of that step is also deleted.



6.5-6 Entering comments

A comment of up to 60 characters can be associated with each step line and each transition line.



- <SPACE> used to enter or modify a comment line.
- <ENTER> confirms the entry. The cursor moves to the next comment line.
- <↑><↓> position the cursor on the selected comment line.
- <←> returns the cursor to the graphic zone.

6.5-7 Confirming a page

The whole Grafcet page must be confirmed before exiting MODIFY mode.

Edit, Confirm	<CTRL>+<C>	from MODIFY mode, causes the Grafcet page to be compiled.
--------------------------	-------------------------------	-----------------------------------------------------------

If the Grafcet page is not consistent, the terminal displays an error message which shows the user which modifications should be made.

6.5-8 Modifying the elements in a page

Any element can be modified in a page if the terminal is in MODIFY mode : steps, macro-steps, transitions.

The modification is only taken into account after the page has been confirmed.

Modifying a step

The modification procedure is the same as the data entry procedure :

- position the cursor on the step to be modified,
- press the soft key corresponding to the graphic symbol required. The previous graphic symbol is “overwritten”,
- enter the step number and confirm with **<ENTER>**.

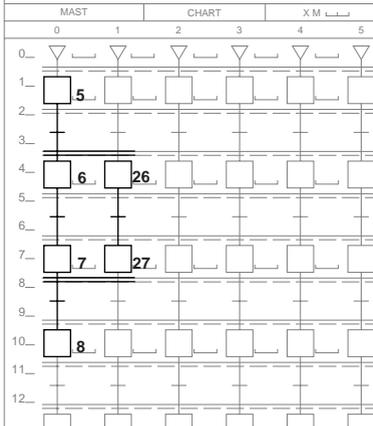
Modifying a step number

- position the cursor on the step to be modified,
- press **<SPACE>**,
- enter the step number and confirm with **<ENTER>**.

Confirming the modification of a step number (by confirming the page) deletes the following elements :

- the actions associated with the step, whether they are actions on activation, continuous actions or actions on deactivation,
- the transition condition associated with the upstream transition,
- the transition condition associated with the downstream transition, if it has been programmed.

Simultaneous activations and deactivations :



MODIFICATION OF THE STEP NUMBER	ELEMENTS DELETED			
	ACTIONS ASSOCIATED WITH THE STEP	CONDITIONS ASSOCIATED WITH THE TRANSITIONS		
		UPSTREAM	DOWNSTREAM	
STEP 6	YES	T5 > 6 YES	T6 > 7 YES	
STEP 7	YES	T6 > 7 YES	T7 > 8 YES	
STEP 26	YES	T5 > 26 NO	T26 > 27 YES	
STEP 27	YES	T26 > 27 YES	T27 > 8 NO	

Modifying a macro-step

The procedure is the same as for modifying a step :

- **changing a mono-active macro-step into a multi-active macro-step** and vice versa, without modifying the address :

The transition conditions associated with the downstream transitions and the macro-step expansion are not modified.

- **modifying the macro-step number :**

The transition conditions associated with upstream and downstream transitions are deleted. The macro-step expansion is not deleted, but remains assigned to the source macro-step.

- **replacing a macro-step by a step :**

The procedure is the same as for modifying a macro-step number. If the macro-step which has been replaced is no longer used, it is advisable to delete its associated expansion.

Modifying a transition

The procedure is as follows :

- position the cursor immediately before the transition to be modified,
- press the soft key corresponding to the graphic symbol required. The previous symbol is "overwritten".

If the steps upstream and downstream to the transition have not been modified, the transition condition associated with this transition is not deleted.

It is also possible to start or end a sequence selection without modifying the transition condition.

Modifying a connector

The number of the source or destination step, as well as the memory address of the transition condition, are modified. The deleted transition condition must be re-entered.

Returning to the original page

	<ENTER> or <CTRL>+<C>	confirms the modified page.
Edit, Cancel	<ESCAPE>	cancels the modifications and returns to the original page (prior to modification).

6.5-9 Deleting a module, a page or an element

Deleting a module

This is performed from the PROGRAM BROWSER screen which is used to select the module to be deleted. A task and its subsets are selected in the Browser using the <↑><↓> keys to select the element required, and the <←><→> keys to move between columns.

Empty Module after confirmation, this is used to delete the selected module.



This procedure deletes the expansion of the macro-step.

Warning

If a program module (task/module) contains a write-protected basic module, this program module cannot be deleted.

Deleting a page

Page deletion is performed in DISPLAY mode.

Two cases can occur :

- if the Grafcet chart(s) is(are) shown on the page to be deleted,
<DELETE> is used, after confirmation, to delete the selected page.
- if the Grafcet chart(s) is(are) spread over two or more pages in the same module, the terminal checks the consistency of the connectors. If any inconsistencies are detected, it is impossible to quit the module without performing the requested modifications. Once these have been made, the page can be deleted.

Comment

Grafcet pages are created during configuration. Deleting a page means deleting the contents of that page. The "deleted" page can then be reprogrammed.

Deleting an element

If the terminal is in MODIFY mode, it is possible to delete the following elements from the page :

- delete a step,
- delete a macro-step,
- delete a transition,
- delete a path.

After positioning the cursor to the left of the element to be deleted :

<DELETE> deletes the element.

<ESCAPE> cancels deletion of the element(s) and returns to DISPLAY mode.

The deletion of a step also deletes :

- the transition or path downstream of the step,
- the actions associated with the step.

The deletion of a macro-step deletes any downstream transitions. The deletion of a macro-step in a chart does not delete the macro-step expansion. This can be used elsewhere. The deletion of a macro-step and its expansion are described earlier in this section.

When deleting a transition, path or connector, the cursor must be positioned on the step-transition link line at the start of the path. In all three cases, the associated transition condition is deleted.

When deleting a complex path line, the terminal checks the consistency of the paths which have not been deleted which intersect with the deleted path.

6.6 Multiple selection

6.6-1 Presentation

Multiple selection is used to move, delete or copy several graphic elements on the same page simultaneously.

There are two ways of performing multiple selection (framed in green) :

KEYBOARD :

Activate the multiple selection mode via the **Edit** menu, **Select Objects** option. The **<→>**, **<←>**, **<↓>**, **<↑>** keys are used to increase the size of the window, horizontally and vertically from the position of the cursor. Once the multiple selection zone has been determined, confirm with **<ENTER>**.

The maximum size the window can be increased to is the size of the terminal screen. It does not include the comment if it exists.

MOUSE :

Holding down the right-hand mouse button, define the multiple selection zone. Releasing this button confirms the selection.

6.6-2 Moving graphic elements

It is possible to move one or more graphic elements (multiple selection) on the same page in MODIFY mode.

KEYBOARD :

The **<→><←>** keys are used to move the selection horizontally.

The **<↓><↑>** keys are used to move the selection vertically.

MOUSE :

Once the group of objects to be moved has been selected, click inside the green frame and hold the mouse button down until the destination has been reached. Releasing the mouse button causes the movement of the graphic objects to be taken into account.

Notes

Some graphic elements cannot use certain positions.

6.6-3 Deleting graphic elements

Having defined a selection containing graphic elements to be deleted :

<DELETE> deletes the graphic elements contained in the window.



Section	Page
7.1 Changes made since V52	7/2
7.2 Selecting the task and module	7/3
7.2-1 Selecting the program address	7/3
7.2-2 MODULE SELECTION screen	7/4
7.3 Reading the program	7/5
7.3-1 Reading rungs	7/5
7.3-2 Menu bar	7/6
7.3-3 Contextual menu	7/8
7.3-4 Toolbar	7/9
7.3-5 Archiving and retrieving a Ladder language module	7/10
7.4 Ladder language zoom function	7/12
7.4-1 Zooming in on a function block	7/12
7.4-2 Zooming in on an OFB	7/13
7.4-3 Zooming in on a CALL coil	7/13
7.5 Creating or modifying the program	7/14
7.5-1 Principle of data entry	7/14
7.5-2 Entering or modifying a label	7/15
7.5-3 Entering or modifying a comment	7/15
7.5-4 Entering graphic elements in the rung	7/16
7.5-5 Entering or modifying a function block	7/17
7.5-6 Entering or modifying a comparison block or an operate block	7/18
7.5-7 Entering or modifying an OFB	7/19
7.5-8 Diagnostics associated with a rung	7/21
7.5-9 Modifying an element in a rung	7/22
7.5-10 Confirming a rung	7/23
7.5-11 Deleting a module or rung	7/23
7.5-12 Duplicating rungs	7/24
7.5-13 Multiple selection of rungs	7/26
7.6 Multiple selection of objects	7/27
7.6-1 Presentation	7/27
7.6-2 Moving graphic elements	7/28
7.6-3 Duplicating graphic elements	7/29
7.6-4 Deleting graphic elements	7/30
7.6-5 Cancel function	7/30
7.7 Searching for the program addresses of a variable	7/31
	7 / 1

7.1 Changes made since V52

To access the Ladder editor using the keyboard :

- activate PROGRAM BROWSER by pressing **<ALT>+<F1>**,
- using the **<TAB>** key, move the cursor to the Task column,
- select a task using the up and down arrow keys,
- select a module and a label in the same way,
- move the cursor to the Task column and confirm the program address by pressing **<ENTER>**.

To access the Ladder editor using the mouse :

- activate PROGRAM BROWSER by clicking on the Program icon in the MAIN WINDOW toolbar,
- select the program address by clicking on the desired task, module and label,
- confirm the selection by double-clicking on the task selected.
- To enter, modify or delete a label or a comment, double-click in the label or comment zone, perform the entry and confirm with **<ENTER>**, or click on the confirmation icon on the toolbar.
- It is possible to perform copy, paste and cut operations and to drag (drag and drop) all or part of a rung.
- Items can be pasted into the same rung or into another rung on a different module.
- Contextual menu : a menu which appears by clicking on the right-hand mouse button in the editor window and is used to gain faster access to the most common actions.

Moving the cursor :

<↑><↓> used to move the cursor.

<→><←>

<CTRL>+ places the cursor in the last or first column of the rung.

<→><←>

<CTRL>+ places the cursor on the next or previous rung.

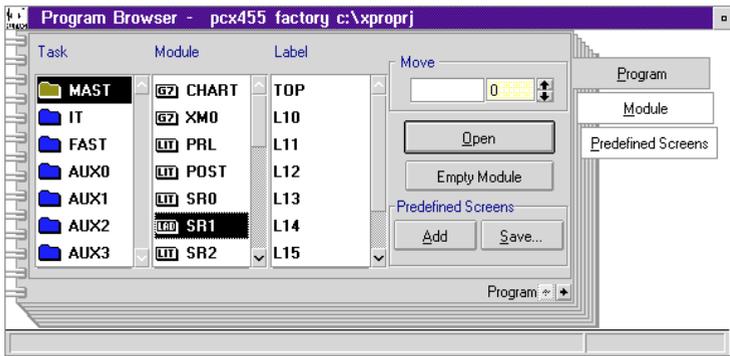
<↑><↓>

7.2 Selecting the task and module

7.2-1 Selecting the program address

Before displaying an existing program or entering a new one, its address must be defined (select the task, module, and rung).

The PROGRAM BROWSER notepad is the first screen displayed in PROGRAM mode, and can be accessed using the **Program** option in the **Tools** menu or by pressing **<ALT> + <F1>**. This screen is the default screen when the application is launched.



A task and its subsets are selected in the Browser using the **<↓>**, **<↑>** keys to select the required element, and the **<←>**, **<→>** keys to move between columns.

Selecting the task

- MAST : master task,
- IT : interrupt task if this is configured,
- FAST : fast task if this is configured,
- AUX0 to AUX3 : auxiliary tasks 0 to 3 if these have been configured and if the type of processor allows these auxiliary tasks.

Selecting the module

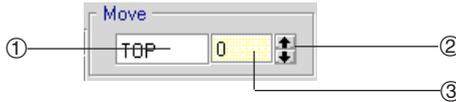
- MAIN : main module,
- SR... : subroutine.

The main **Module** tab on the right of the notepad accesses the MODULE SELECTION screen which is the call screen for the basic modules and macro-modules of the application (see section 7.2-2).

Selecting the rung

The **Move** group is used to directly access a rung by entering its label name. This procedure is also used to select an unlabeled rung, as long as a labeled rung is present in the module.

Example : if an unlabeled rung is located 7 rungs downstream of the rung labeled L38 and 2 rungs upstream of the L61 rung, it can be selected by label L38+7 or L61-2.



- ① The first field is used to enter the text of the label (eg : L38, L38+7, TOP, etc)
- ② The arrows are used to increase or decrease movement.
- ③ The second field is used to enter the movement associated with the label entered in the first field (eg :+12, -15, etc)

TOP selects the first rung of the selected module.

BOTTOM selects the last rung of the selected module.

If the selected rung does not exist in the module, the terminal proposes that the label is re-entered.

Open opens the selected rung.

There are then two possibilities :

1 - the module is empty :

A dialog box asks which language is required. Selecting LADDER causes the Ladder language editor to appear.

2 - the module is already in use :

The program can be read (see section 7.3).

7.2-2 MODULE SELECTION screen

This screen has the same functions as those described for GRAFCET language (see section 6.2-2).

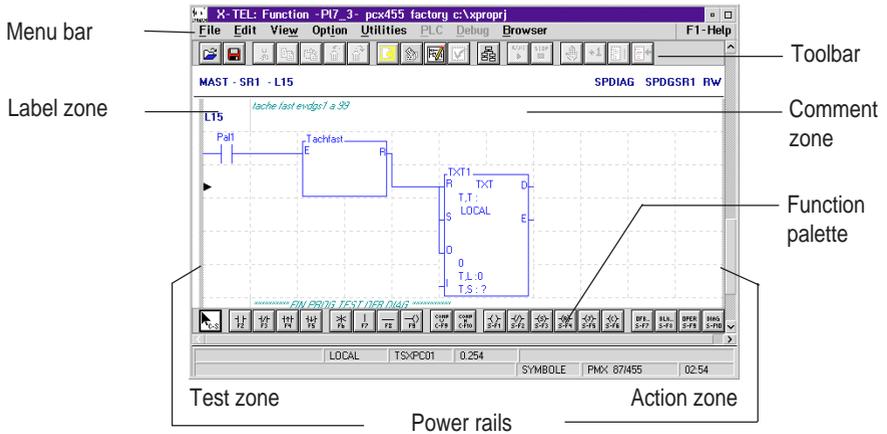
7.3 Reading the program

7.3-1 Reading rungs

Rung DISPLAY screen

This screen, which can be accessed from the PROGRAM BROWSER screen by confirming the selections made, is used to display the program.

An action programmed in Ladder language is accessed by pressing the **<SPACE>** bar in the zoom function (see section 7.4).



Direct access to a rung

	<CTRL>+<Home>	displays the first rung of the module.
	<CTRL>+<END>	displays the last rung of the module.
	<Page Up> <Page Down>	and the scroll bars are used to move within the program.
Edit, Go to	<CTRL>+<G>	used to move to : <ul style="list-style-type: none"> • a labeled rung (eg : L24), • an unlabeled rung addressed with reference to a label (eg : L12+5). <ENTER> confirms entry of the label. <ESCAPE> cancels the entry.

7.3-2 Menu bar

FILE MENU :

Open	displays a dialog box from which the file to be retrieved can be selected.
Save	used to archive the current Ladder language module in a "source" file on disk.
Close	returns to the previous Zoom level.
Exit	used to quit LADDER mode.

EDIT MENU :

Undo	used to undo the last modifications made to the rung.
Redo	reperforms the actions which have been undone.
Cut	deletes the selected rung and copies it to the buffer memory.
Copy	copies the selected rung to the buffer memory.
Paste	retrieves the copied rung from the buffer memory.
Delete	deletes the selected element(s).
Insert a Rung	inserts a rung.
Modify	switches the terminal to modify mode.
Confirm	confirms the modifications made.
Cancel Modifications	reverts to the previous version.
Symbolize a Rung	accesses symbol entry.
Select, → Rung → Objects	used to select : a rung, a graphic element.
Zoom	accesses the contents of a function block or a subroutine.
Go to	is used to position the cursor on a labeled rung.

VIEW MENU :

Variables	immediate change to display variables (addresses).
Symbols	immediate change to display symbols.

UTILITIES MENU :

Animate	launches animation of the screen displayed.
Freeze	freezes animation of the screen displayed.
Cross-references	accesses the cross-references.
OFBs	accesses a description of the OFBs configured in the application.
Edit Program Comment	accesses the LADDER page program comment.
Edit Symbols	accesses the list of symbols.

OPTION MENU :

Toolbar	displays or hides the toolbar.
Symbolize	displays or hides the variable (address), symbol and comment entry bar.
Cancel Help Balloons	activates or deactivates the toolbar help balloons.

PROCESSOR MENU :

Run	launches execution of the program.
Stop	stops execution of the program.

DEBUG MENU :

Start Task	launches or stops execution of the task.
Task Monitored	selects the task to be executed and monitored.
Breakpoints	
→ Insert	places the breakpoint on the program element displayed in the predefined screen.
→ Remove	deletes the breakpoint.
→ View	displays the place where the breakpoint was defined.
Continue	executes a task scan.
Step by step	executes the next element.
Last Stop	displays the last element executed.
Data Mode	accesses DATA mode.
Data Line mode	accesses DATA LINE mode.

BROWSE MENU :

Symbolize window	accesses the symbols line window.
Data window	accesses the data line window.
Browser	accesses the PROGRAM BROWSER notepad.
Screen 1...6	switches from one predefined screen to another.

7.3-3 Contextual menu

This menu appears by clicking on the right-hand mouse button.

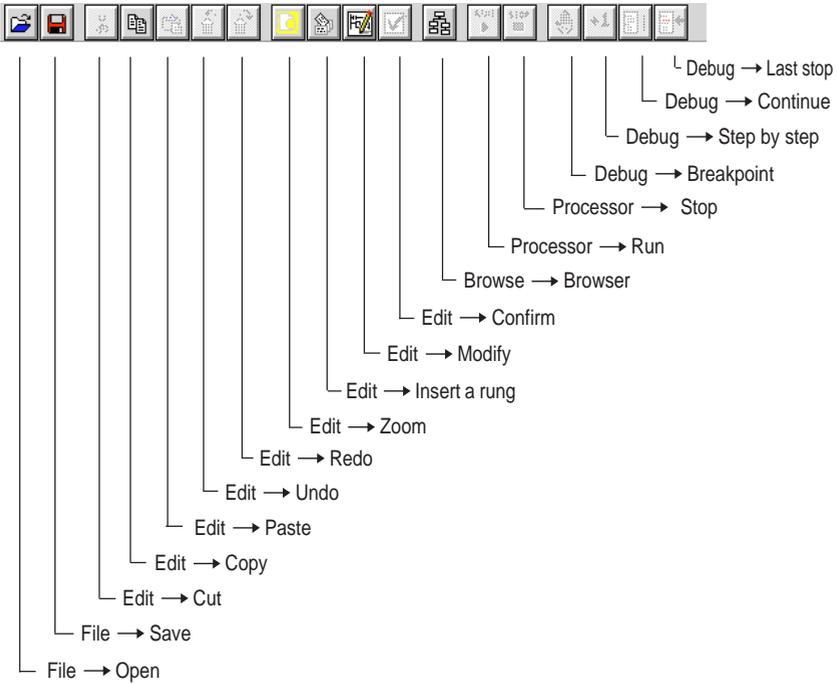
The options (see description in section 7.3-2) are as follows :

- **Close,**
- **Exit,**
- **Undo,**
- **Redo,**
- **Cut,**
- **Copy,**
- **Paste,**
- **Delete,**
- **Insert a rung,**
- **Modify,**
- **Confirm,**
- **Cancel modifications,**
- **Select a rung,**
- **Select objects,**
- **Go to.**

7.3-4 Toolbar

This toolbar provides quicker access to the main functions on the menu bar in LADDER LANGUAGE mode.

Toolbar button definitions :



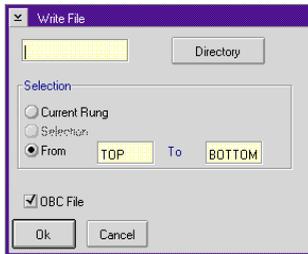
D

7.3-5 Archiving and retrieving a Ladder language module

Archiving

File, Save	<CTRL>+<S>	used to archive the current Ladder language module in a “source” file on disk. This xxx.LAD file, saved in the PL7_3\MOD directory, allows the module to be reused in another application and integrated in a module base.
-------------------	-------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The following window is displayed :



The entry field is used to define the file name.

The **Directory** button is used to access the list of files in the save directory.

The **Current Rung** button is used to archive the current rung (containing the cursor).

The **Select** button is used to archive the selected rungs only (see how to select several rungs later on this section).

The **FROM** button is used to select a page to save.

The **OBC** check box is used to archive the Ladder language module, with or without the internal OFB constants. If the OBC option is selected, a file <name_OFB>.OBC is created for each module OFB which contains internal constants. In addition, an xxx.OBM file (with the same name as the xxx.LAD file) provides a list of all OFBs which have generated an .OBC file.

<TAB> used to move from one group to the other.

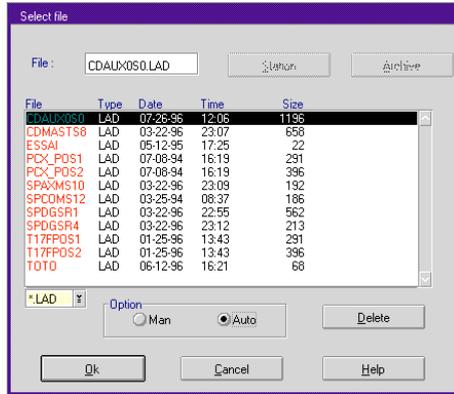
<↑><↓> used to move within a group.

<SPACE> used to activate the buttons and check boxes.

Retrieving

File, Open		used to retrieve a “source” file to the working RAM memory of the terminal.
-------------------	--	-----------------------------------------------------------------------------

The following window is displayed :



The entry field is used to define the file name.

The list of files is used to select a file to be retrieved from those in the current directory.

The **AUTO** and **MAN** buttons define the retrieval mode :

AUTO : global retrieval with stop on error only,

MAN : manual retrieval with confirmation on each element read.

Archive assigns the global archive name (PL7_3) as the file name.

Station assigns the name of the current station as the file name.

OK saves the selection made and starts retrieving the file selected.

Cancel cancels the selections made and exits the dialog box without triggering any action.

Help calls up the online help function.

<TAB> used to move from one group to the other.

<↑><↓> used to move within a group.

<←><→>

<↑><↓> used to move within the list or in the inset.

Note

For more information on the V5 modular version, see section 15 - Appendix.

7.4 Ladder language zoom function

The Ladder language zoom function accesses, from the editor in DISPLAY mode, the contents of :

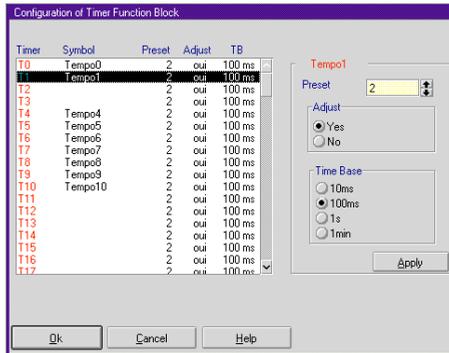
- a function block,
- an operate block containing a call to an OFB (EXEC),
- a subroutine called by a CALL coil.

Procedure for using the zoom function

Edit, Zoom	<CTRL> +<PG DOWN>	activates ZOOM mode on the element selected.
-----------------------	------------------------------------------	----------------------------------------------

7.4-1 Zooming in on a function block

To read the contents of a function block, place the cursor on the control system function block, and activate ZOOM mode (**<CTRL>+<PG DOWN>**) from the editor DISPLAY mode.



OK used to return to the calling screen and take account of any possible modifications to the block configuration parameters.

Cancel used to return to the calling screen leaving the configuration unchanged.

<TAB> used to move from one group to the other.

<↓>, <↑> used to move within a group.

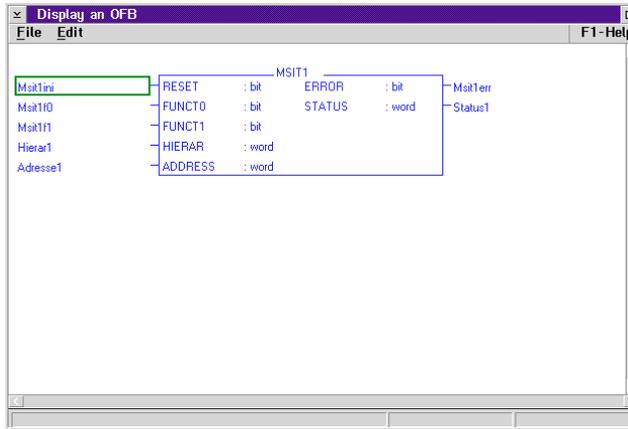
See section 3.13 to determine which operating mode should be used to modify the list parameters (**Apply** button)

Undo used to return the data on the notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

7.4-2 Zooming in on an OFB

To read the contents of an OFB, position the cursor on the operate block containing the OFB EXEC instruction, and activate ZOOM mode (<CTRL>+<PG DOWN>) from the editor DISPLAY mode.



Edit, Constants		accesses the internal OFB constants.
File, Quit	<ENTER>	returns to the calling screen

7.4-3 Zooming in on a CALL coil

To access a subroutine, position the cursor on the CALL coil concerned, and activate ZOOM mode (<CTRL>+<PG DOWN>) from the editor DISPLAY mode. Eight zoom levels can be accessed in this way.

If the subroutine has not yet been programmed, the terminal gives the option of selecting the language in which it must be entered.

File, Close	<CTRL>+<PG UP>	moves up one zoom level.
Menu, Quit		returns to the first zoom level.

7.5 Creating or modifying the program

7.5-1 Principle of data entry

To create or modify a rung, as described in this section, the terminal must be in MODIFY mode.

Modifying a program from DISPLAY mode

- If the module is to be modified :

Edit, accesses the mode for modifying the rung displayed. This operation can
Modify also be performed by double-clicking on the rung to be modified.

Authorization to access modifications changes the color of the rung to red.

- If the module is to be deleted :

File, closes the editor and returns to the PROGRAM BROWSER notepad.
Quit

Empty deletes the contents of the selected module.
Module

Creating a program from DISPLAY mode

The selected module has no programmed rung. A rung is inserted automatically.

Access to writing the program is given by :

- the **Edit** menu bar, and either the **Insert a Rung** or **Modify** options,

The terminal switches to INSERT mode, the two power rails are displayed in bold. The terminal does not impose any order on the entry of the various elements.

When inserting a rung at the end of a module, the insertion point must be placed on the end of program mark.

7.5-2 Entering or modifying a label

Entering a label is optional. The cursor must be in the label zone. Double-clicking in this zone or pressing the **<SPACE>** bar enables the entry of a new label or the modification of an existing label. **<ENTER>** confirms this entry.

Deleting a label

Position the cursor on the label :

Edit, Delete	<DELETE>	deletes the label.
-------------------------	-----------------------	--------------------

7.5-3 Entering or modifying a comment

Entering a comment assigned to a rung is optional. The cursor must be in the comment zone.

The comment, consisting of a maximum of 222 characters, appears, after confirmation, on three lines at the top of the screen.

Double-clicking in the comment zone or pressing the **<SPACE>** bar is used to enter a comment or modify an existing comment. The **<ENTER>** key confirms this entry.

	<SHIFT> + <ENTER>	changes line.
--	------------------------------------------	---------------

Comments

- if the rung occupies all 7 lines of the screen, it is not possible to enter a comment without modifying the rung (deleting a line, etc),
- if the rung occupies less than 7 lines, the entry of a comment shifts the rung down if the first line is used or, the rung is reorganized to fit on six lines if it fills the whole screen.

Deleting a comment

Position the cursor in the comment zone.

Edit, Delete	<DELETE>	deletes the comment.
-------------------------	-----------------------	----------------------

7.5-4 Entering graphic elements in a rung

General

Graphic elements can be entered (created or modified) using the keyboard or the mouse.

KEYBOARD :

- position the cursor in the zone where the object is to be created.
- select the object to be created in the palette using the soft keys offered.
- enter the bit object or associated symbol (if necessary) then confirm with **<ENTER>**.

MOUSE :

- click on the graphic object required in the palette to select it.
- click in the zone where the object is to be created.
- enter the bit object or associated symbol (if necessary) then confirm with **<ENTER>**.

Entering a contact

Definition of the soft keys associated with the graphic elements :



Entry of a normally-open contact.



Entry of a rising edge contact.



Entry of a normally-closed contact.



Entry of a falling edge contact.

Note

When the bit object or the symbol associated with the graphic element is too big for the entry zone, it is displayed in part and an inverted comma symbolizes that it has been abbreviated.

Entering a coil

This is only possible in the action zone.

Defining soft keys associated with the coils :



Entry of a direct output coil.



Entry of a jump output coil (JUMP coil).



Entry of an inverse output coil.



Entry of a subroutine call output coil (CALL coil).



Entry of a set output coil.



Entry of a hash output coil when programming the conditions associated with transitions.



Entry of a reset output coil.

In the case of a jump output coil (jump to another rung), enter the label of the rung (eg. L115).

In the case of a subroutine call output coil, enter its number (eg. SR5).

Note

When the bit object or the symbol associated with the output coil is too big for the entry zone, it is displayed in part and an inverted comma symbolizes that it has been abbreviated.

Entering a connection

Definition of the soft keys associated with connections :



Entry of a horizontal link.



Entry of a vertical link downwards from the position of the cursor.



Entry of a horizontal link from the position of the cursor to the action zone.



Deletion of a vertical link. The cursor must be positioned at the top of the vertical link.

7.5-5 Entering or modifying a function block

Control system function blocks (timer, register, monostable and counter), text blocks and control blocks cannot be entered unless they have been configured. They are entered in two stages :

- entry of the graphic element,
- entry of the block number or mnemonic, then confirmation with **<ENTER>** which displays the block parameters (default values if they have not been modified). **<ESCAPE>** cancels the selection.

Temporisateur (Timer) enters a timer.

Monostable (Monostable) enters a monostable.

Compteur (Counter) enters an up/down counter.

Registre (Register) enters a LIFO or FIFO register

Communication (Text) enters a text block.

Contrôle (Control) enters a control block.



Modifying parameters

(See section 7.4-1 - Zooming in on a function block).

Note

Modification of configuration parameters in online mode is only taken into account when bit SY0 is set to 1.

7.5-6 Entering or modifying a comparison block or an operate block

Vertical comparison block

This block is entered in the test zone test and is not numbered.



Enters the graphic element. The terminal then suggests entering the both operands one after the other. After confirming with **<ENTER>** they will be displayed in both windows shown in reverse video in the block. The first operand entered must be an internal word Wi.

Note

If no space is left between the 2 objects entered, the terminal interprets the entries as a single object and displays an error message.

Horizontal comparison block

This block uses Literal (Structured Text) language.



Enters the graphic element. The terminal then suggests entering the arithmetic expression (80 characters maximum). After confirmation with **<ENTER>**, the expression is displayed either completely or partially in the block, depending on the number of characters.

Operate block

This block is automatically positioned in the action zone. Like the horizontal comparison block, it uses Literal language.



Enters the graphic element. The terminal then suggests entry of the arithmetic expression (512 characters maximum). After confirming with **<ENTER>**, the expression is displayed either completely or partially in the block. This entry must follow the rules of Literal language.

OFB block



Enters an operate block containing an OFB EXEC. The terminal provides a list of configured OFBs. **<ENTER>** confirms the OFB, **<ESCAPE>** cancels the selection.

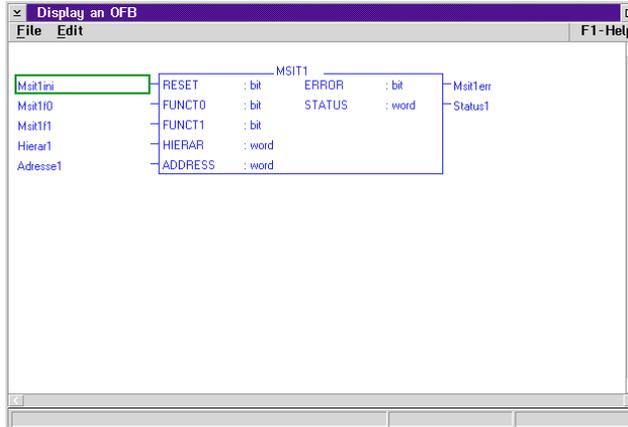
Confirming an OFB gains access to the parameter modification screen for the selection.

7.5-7 Entering or modifying an OFB

Entry or modification of an OFB follows the same rules as those used for other graphic objects (see section 7.5-4) but with the following differences :

- an OFB is only inserted into the action zone. The OFB graphic element is in fact, an OPERATE block containing the call to an OFB (EXEC),
- in the case of adding an OFB, the user is invited to select an OFB from a list of OFBs configured in the application. The OFB parameter modification screen appears once the OFB is selected. OFB instance 0 is the default OFB. See below how to modify the instance of the OFB parameters.
- in the case of a modification, the OFB parameter modification screen is called directly.

Modifying the OFB parameters or instance



	<↑><↓>	moves the selection within the same group of parameters (inputs or outputs).
	<←><→>	changes group (inputs, instance or outputs).
	<SPACE>	modifies the current parameter.
	Alphanumeric characters	direct entry of a new parameter.
File, Quit	<ENTER>	confirms the modifications made.
	<ESCAPE>	cancels the modifications made.
Edit, Constants		accesses modification of the OFB for the current instance.

Entering an OFB from an OPERATE block

It is possible to enter an OFB when entering an OPERATE block. Simply select "OFB" from the palette.

If the call to the OFB is already entered (EXEC <OFB> + instance) the OFB modification screen will be called directly and positioned on the OFB entered.

If the instance is not entered, the 0 instance will be selected by default.

If the EXEC key word is not found, the user can make a selection from the list of OFBs configured in the application.

7.5-8 Diagnostics associated with a rung

It is possible to associate a diagnostic check with a rung in order to monitor a Boolean condition or a transition. This diagnostic check is performed by calling an application diagnostic optional function block (OFB). Some OFBs can cause the source of the monitored rung to be memorized. It is not possible to delete the last diagnostic check, or add the first diagnostic check with memorization of the source when the terminal is connected to a PLC that is running.

Entry or modification of a diagnostic OFB

Entry or modification of an OFB follows the same rules as those used for other graphic objects (see section 7.5-4) but with the following differences :

- a diagnostic OFB can only be inserted on the last line of the rung,
- a new selection in the "DIAG" object palette, or pressing the **<CTRL>+<PAGE DOWN>** keys, accesses the OFB modification screen. In the case of an addition, the user can make a selection from the list of diagnostic OFBs configured in the application.

Condition associated with the execution of a diagnostic OFB

It is possible to condition execution of a diagnostic OFB by inserting the following before calling the OFB (EXEC) :

IF <Boolean expression> THEN

7.5-9 Modifying an element in a rung

Deleting an element

This is performed in MODIFY mode. The procedure is the same for contacts, coils, function blocks, comparison blocks and operate blocks.

Select the element to be deleted.

<DELETE> deletes this element. A new element can be entered.

Modifying an element

The modification procedure is the same whatever the graphic element to be replaced.

KEYBOARD :

- select the object to be modified
- enter the new element from the palette
- enter the new bit object or associated mnemonic, if necessary
- confirm the new entry with **<ENTER>**.

MOUSE :

- select the new element from the palette
- click on the object to be modified
- enter the new bit object or associated mnemonic, if necessary
- confirm the new entry with **<ENTER>**.

Modifying a bit object or mnemonic associated with an element :

KEYBOARD :

- place the cursor on the element to be modified
- **<SPACE>** accesses modification of the bit object or mnemonic associated with a contact or a coil.
- confirm the new entry with **<ENTER>**.

MOUSE :

- double-clicking on the object to be modified accesses modification of the bit object or mnemonic associated with a contact or a coil.
- confirm the new entry with **<ENTER>**.

7.5-10 Confirming a rung

In MODIFY mode, when a rung has been entered, it must be confirmed so that it can be compiled and stored in the memory.

Edit, Confirm	<CTRL>+<C>	confirms the rung entered and switches the terminal to DISPLAY mode. The memorized rung can then be read.
--------------------------	-------------------------------	-----------------------------------------------------------------------------------------------------------

7.5-11 Deleting a module or rung

Deleting a module

This is performed from the PROGRAM BROWSER notepad which is used to select the module to be deleted. A task and its subsets are selected in the Browser using the **<↓>**, **<↑>** keys to select the required element, and the **<←>**, **<→>** keys to move between columns.

Empty Module after confirmation, this deletes the content of the selected module.

Deleting a rung

Deleting a rung applies to the current rung, to the current selection (see the following section on how to select several rungs) or to a range of rungs.

In DISPLAY mode, pressing the **<DELETE>** key or using the **Delete** option in the **Edit** menu, causes the following dialog box to appear :



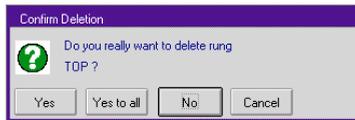
The selection group buttons are used to select to which rungs deletion will apply :

- current rung : the rung containing the cursor will be deleted,
- selection : if one or more rungs are selected, they will be deleted,
- from <label> to <label> : a range of rungs defined by the rungs in the two entry zones will be deleted.

Note :

The rungs defining the range of rungs must be marked by TOP, BOTTOM, a label (eg : L20) or a label plus a movement (eg : L30+1, L40-2).

The confirmation box indicates whether deletion of each rung must be confirmed prior to execution.



- Yes** deletes the rung and continues the operation.
- No** does not delete the rung and continues the operation.
- Yes to All** deletes the rung and continues the operation without requesting confirmation.
- Cancel** cancels the operation.

Note

An operation to delete without confirmation can be stopped by pressing the **<ESCAPE>** key.

D

7.5-12 Duplicating rungs

This procedure is used to copy one or more rungs into a buffer memory, and to duplicate them any number of times in any module programmed in Ladder language. The rungs copied are those which have been selected.

Copying the rung into the buffer memory

Edit, Copy	<CTRL>+<INS>	used to copy the selected rungs in DISPLAY mode. This rung remains in the buffer memory until the next copy action is performed.
Edit, Cut	<SHIFT>+	as well as being copied into the buffer memory, the selected rungs are deleted.

7.5-13 Multiple selection of rungs

Multiple rung selection is used to define a list of rungs (contiguous or not) which can then be :

- deleted,
- copied into a buffer memory,
- written in a source file.

There are two ways of performing multiple selection :

KEYBOARD :

- position the cursor in the rung to be selected or deselected (arrow keys).
- The **<SPACE>** bar or the **Edit** menu, **Select → Rung** option, can be used to select the current rung and to deselect any previously selected rungs.
- **<CTRL>+<SPACE>** adds the current rung to those previously selected. Deselects the rung if it has already been selected.
- **<SHIFT>+<SPACE>** adds to the current selection the rungs between the last rung selected and the current rung.

MOUSE :

- Position the mouse pointer tool to the left of the rung to be selected or deselected (before the power rail). The mouse pointer tool then changes to a horizontal arrow.
- Clicking with the **left-hand mouse button** selects the current rung and deselects the previously selected rungs.
- **<CTRL>+ click with left button** adds the current rung to those already selected. Deselects the rung if it has already been selected.
- **<SHIFT>+ click with left button** adds the rungs between the last rung selected and the current rung to the current selection.
- By clicking and holding down the left-hand mouse button and moving to the top or bottom edges of the screen, the user can scroll through the rungs. Moving the mouse pointer tool towards the top edge of the screen scrolls to the start of the program; moving the pointer to the bottom edge of the screen scrolls to the end of the program. In this way every new rung appearing on the screen will be selected, or deselected if the last operation was a deselection operation (**<CTRL>+<SPACE>** or **<CTRL>+Click** on a previously selected rung).

7.6 Multiple selection of objects

7.6-1 Presentation

Multiple selection is used to move, delete, or copy several graphic elements simultaneously in the same rung, or in another rung.

There are two ways of performing multiple selection :

Selection using a frame

Accessing the **Edit** menu, **Select → Objects** option changes the form of the mouse pointer tool to the symbol for object selection mode. A frame then can be drawn using the mouse (click on the left button to mark the top left-hand corner of the frame, and drag the mouse to increase the size of the frame).

Pressing **<ESCAPE>** when defining a selection frame cancels the mode.

<ALT> + click with left button are used to directly define the selection frame.

Once the frame has been defined, the objects inside it are selected. If **<SHIFT>** is pressed at the start of frame definition (**Edit, Select → Objects + click with left button** or **<SHIFT> + <ALT> + click with left button**) this selection is added to the selected objects.

Selecting objects one by one

A selection can be defined by adding or removing objects one by one.

KEYBOARD :

- The arrow keys **<→>**, **<←>**, **<↑>**, **<↓>** move the cursor and select the object highlighted by the cursor.
- If the **<SHIFT>** key is pressed at the same time as the arrow keys **<→>**, **<←>**, **<↑>**, **<↓>**, the object highlighted by the cursor is added, or removed if it has already been selected.

MOUSE :

- The left button is used to select the highlighted object.
- **<SHIFT> + click with left button** adds the highlighted object to the selection, or removes it if it was already selected.

7.6-2 Moving graphic elements

It is possible to move one or more graphic elements (multiple selection) in the same rung in MODIFY mode.

MOUSE :

Once the group of objects to be moved has been selected, click on this group and hold the mouse button down until the destination is reached. Releasing the mouse button takes account of the movement of the graphic objects.

If the <CTRL> key is held down at the same time as the mouse button, the objects will be duplicated.

KEYBOARD :

Copy/Paste the selection to move it.

Notes

Some graphic elements are not allowed to use certain positions. When the object is being moved, a no-entry sign will indicate any position which cannot be used.

7.6-3 Duplicating graphic elements

This procedure is used to copy one or more graphic elements into a buffer memory, and to duplicate them any number of times in any rung or module programmed in Ladder language.

Copying graphic elements into a buffer memory

Edit, Copy	<CTRL>+<INS>	used to copy the selected graphic elements. These graphic elements remain in the buffer memory until the next copy action is performed.
Edit, Cut	<SHIFT>+	as well as being copied into the buffer memory, the objects are deleted. This is only allowed in MODIFY mode.

Restoring copied graphic elements

They can only be restored in MODIFY mode in the same programming module or in another module.

Edit, Paste	<SHIFT>+<INS>	restores the memorized graphic elements.
------------------------	----------------------------------	------------------------------------------

Restoring previously copied objects displays an image of these objects in the centre of the screen.

KEYBOARD :

The image can be moved using the arrow keys. The **<ENTER>** key inserts the objects; the **<ESCAPE>** key cancels the restoration operation.

MOUSE :

Move the image and click when you reach the required position.

Notes

- When copying one or more elements on to another rung or module, the source rung must be confirmed once the elements have been copied in order to be able to access the rung in which the element(s) must be pasted.

7.6-4 Deleting graphic elements

Having defined a selection containing the graphic elements to be deleted.

Edit, Delete	<DELETE>	deletes the graphic elements contained in the selection.
-------------------------	-----------------------	----------------------------------------------------------

7.6-5 Cancel function

The cancel function (**Undo** option in the **Edit** menu), is used to cancel the most recent modifications made to the rung, as long as the rung has not been confirmed.

Edit, Redo		is used once one or more cancelations have been performed.
Edit, Undo	<ALT> + <BACKSPACE>	cancels the last insert or delete operation for any given object.

D

7.7 Searching for the program addresses of a variable

- <Alt>+<X>** When a program is displayed from a selected rung, this command is used to access the :
- display of all the variables in the rung,
 - selection of a variable and display of all the program addresses relating to this variable,
 - selection of a program address and display of the corresponding program element.

The variable is either a standard PL7-3 variable (Wi, Bi, etc) or an OFB variable. This function is explained in section 15.5 of part H.

D

Section	Page
8.1 Changes made since V52	8/2
8.2 Selecting the task and the module	8/3
8.2-1 Selecting the program address	8/3
8.2-2 MODULE SELECTION screen	8/4
8.3 Reading the program	8/5
8.3-1 Read a program module	8/5
8.3-2 Menu bar	8/6
8.3-3 Toolbar	8/8
8.3-4 Archiving and retrieving a Literal language module	8/9
8.4 Literal language zoom function	8/11
8.4-1 Zooming in on a function block	8/11
8.4-2 Zooming in on a CALL instruction	8/12
8.4-3 Zooming in on an OFB	8/12
8.5 Creating or modifying the program	8/13
8.5-1 Principle of data entry	8/13
8.5-2 Entering or modifying a label	8/15
8.5-3 Entering or modifying a comment	8/15
8.5-4 Entering or modifying a Literal expression - completion	8/16
8.5-5 Floating point type instructions	8/17
8.5-6 Diagnostics associated with a program line	8/19
8.5-7 Confirming a program line	8/19
8.5-8 Deleting a module or a program line	8/20
8.5-9 Duplicating a program line	8/21
8.5-10 Multiple selection of statements	8/23
8.6 Searching for the program addresses of a variable	8/24

8.1 Changes made since V52

To access the Literal editor using the keyboard :

- activate PROGRAM BROWSER by pressing **<ALT>+<F1>**,
- using the **<TAB>** key, move the cursor to the Task column,
- select a task using the up and down arrow keys,
- select a module and a label in the same way,
- move the cursor to the Task column and confirm the program address by pressing **<ENTER>**.

To access the Literal editor using the mouse :

- activate PROGRAM BROWSER by clicking on the Program icon in the MAIN WINDOW toolbar,
- select the program address by clicking directly on the desired task, module and label,
- confirm the selection by double-clicking on the selected task.
- To enter, modify or delete a label or a comment, click on the label or comment zone with the mouse, (or press the **<←→>** arrow to go directly to the label zone. eg : L10 : **← SET B0**), enter the label or comment directly and confirm by pressing **<ENTER>** or by clicking on the confirmation icon on the toolbar.
- To enter, modify or delete a new statement, position the cursor at the start of the statement in question. The square-shaped cursor indicates that you are still in DISPLAY mode. To switch to MODIFY mode press the **<SPACE>** bar. The cursor then becomes a line cursor. To confirm the statement press **<ENTER>** : the cursor becomes square-shaped again.
- It is possible to perform copy, paste and cut operations between different modules.
- The mouse is used for multiple selection of statements : to do this, click on the statements to be grouped, while holding down the **<CTRL>** key.
- A key word completion function simplifies entry of instructions : in MODIFY mode, type the start of a key word then press the **<SPACE>** bar to call the completion function.
- Contextual menu : a menu, which appears by clicking on the right-hand mouse button in the editor window, used to gain faster access to the most common actions.

Moving the cursor :

<↑><↓> used to move the cursor.

<←><→>

<CTRL>+<←><→> positions the cursor on the first statement.

<←><→><←><→>

<CTRL>+<←><→> positions the cursor on the next or the last statement.

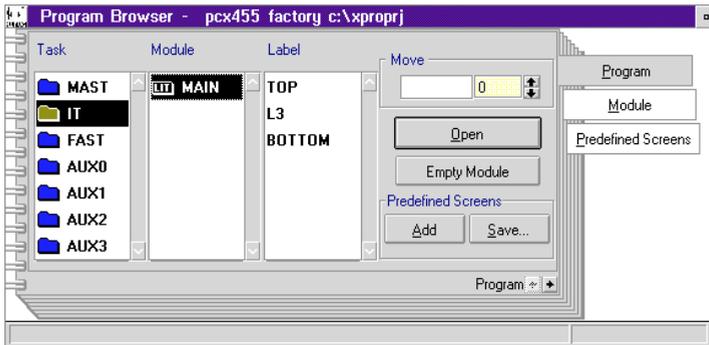
<↑><↓>

8.2 Selecting the task and the module

8.2-1 Selecting the program address

Before displaying an existing program or entering a new one, its address must be defined (by selecting the task, module, program line).

The PROGRAM BROWSER screen is the initial screen in PROGRAM mode, and can be accessed using the **Program** option in the **Tools** menu, or by pressing **<ALT>+<F1>**. When the application is launched this is the default screen.



A task and its subsets are selected in the Browser using the **<↓>**, **<↑>** keys to select the element required, and the **<←>**, **<→>** keys to move between columns.

Selecting the task

- MAST : master task,
- IT : interrupt task if this has been configured,
- FAST : fast task if this has been configured,
- AUX0 to AUX3 : auxiliary tasks 0 to 3 if they are configured and if the type of processor allows these auxiliary tasks.

Selecting the module

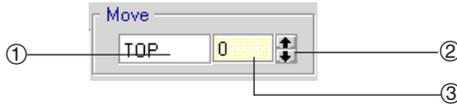
- MAIN : main module,
- SR... : subroutine.

The main **Module** tab, on the right of the notepad, accesses the MODULE SELECTION screen which is the call screen for the basic modules and macro-modules in the application (see section 6.2-2).

Selecting the program line

The **Move** group is used to directly access a statement by entering its label name. This procedure is also used to select an unlabeled statement, as long as a labeled line is in the module.

Example : if an unlabeled statement is located 7 statements downstream of the statement labeled L38 and 2 statements upstream of the L61 statement, it can be selected using label L38+7 or L61-2.



- ① The first field is used to enter the name of the label (eg : L38, L38+7, TOP, etc)
- ② The arrows are used to increase or decrease movement.
- ③ The second field is used to enter the movement associated with the label entered in the first field (eg :+12, -15, etc)

TOP selects the first statement of the selected module.

BOTTOM selects the last statement of the selected module.

If the selected statement does not exist in the module, the terminal proposes that the label is re-entered.

Open used to open the selected statement.

There are then two possibilities :

1 - the module is empty :

A dialog box asks which language is required. Selecting LITERAL causes the Literal (Structured Text) language editor to appear.

2 - the module is already in use :

The program can be read line by line (see section 8.3).

8.2-2 MODULE SELECTION screen

This screen has the same functions as those described for GRAFCET language (see section 6.2-2).

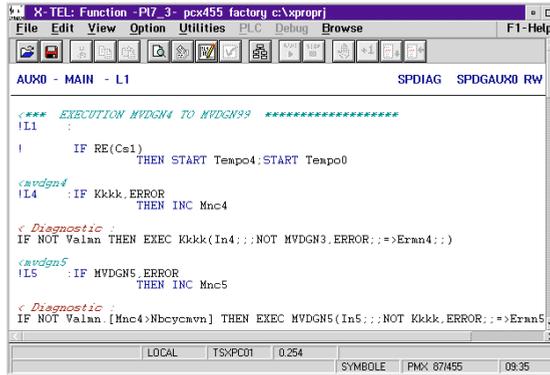
8.3 Reading the program

8.3-1 Read a program module

Literal DISPLAY screen

This screen, which is accessed from the PROGRAM BROWSER screen by confirming the selections made, is used to display the program.

An action programmed in Literal language is accessed by pressing the <SPACE> bar in the zoom function (see section 8.4).



Direct access to a program line

	<Ctrl>+<Home>	used to move to the start of the module.
	<Ctrl>+<END>	used to move to the end of the module.
	<Page Up> <Page Down>	and the scroll bars are used to move within the program.
Edit, Go to	<Ctrl>+<G>	used to move to : <ul style="list-style-type: none"> • a labeled statement (eg : L24), • an unlabeled statement addressed with reference to a label (eg : L12+5).
	<ENTER>	confirms entry of the label.
	<ESCAPE>	cancels the entry.

Depending on the editor user mode, the cursor appears as either :

- a square (■) : at the start of the line. This indicates that the current mode is DISPLAY mode. Any entry will insert a new line.
- line cursor (|) : any entry will modify the line containing the cursor.

8.3-2 Menu bar

FILE MENU :

Open	displays a dialog box from which the file to be retrieved can be selected.
Save	used to archive the current Literal language module, in a "source" file on disk.
Close	returns to the previous Zoom level.
Exit	used to quit LITERAL mode.

EDIT MENU :

Undo	used to undo the last operations performed on the statement.
Cut	deletes and copies the selected statements to the buffer memory.
Copy	copies the selected statements to the buffer memory.
Paste	retrieves the copied or cut statements from the buffer memory.
Modify Line	accesses modification of the program line indicated by the cursor.
Insert Line	accesses INSERT mode in order to add a statement before the current line.
Confirm	compiles the inserted or modified statement and stores it in the memory.
Delete	after confirmation, deletes the line indicated by the cursor.
Edit OFB	in MODIFY mode, this is used to add or modify an OFB.
Insert Diagnostic OFB	used to associate a diagnostic OFB with the statement currently being modified. If an OFB has already been associated, this menu is changed to "Delete Diagnostic OFB", making it possible to delete the OFB.
Zoom	used to access the contents of a function block or a subroutine.
Go to	used to position the cursor on a labeled statement.
New Line	used to insert a new line into a statement currently being modified.
Assignment	used to insert the assignment sign (→) into a statement currently being modified.
Selection	used to switch to statement selection mode.

VIEW MENU :

Variables	immediate change to display VARIABLES (addresses).
Symbols	immediate change to display SYMBOLS.
Display Diagnostic OFB	displays or hides the diagnostic OFBs in the editor.

OPTION MENU :

Toolbar	displays or hides the toolbar.
Symbolize	displays or hides the variable (address), symbol and comment entry bar.
Font	used to select the character font in which the program is displayed.
Cancel Help Balloons	deactivates the toolbar help balloons.
Color Syntax	switches to color syntax.
Key words Completed	activates/deactivates the key word completion function.

UTILITIES MENU :

Animate	launches animation of the screen displayed.
Freeze	freezes animation of the screen displayed.
Cross-references	accesses the cross-references.
OFBs	accesses a description of the OFBs configured in the application.
Edit Program Comment	accesses the current LITERAL statement program comment.
Edit symbols	accesses the list of symbols.

PROCESSOR MENU :

Run	starts the processor.
Stop	stops the processor.

DEBUG MENU :

Stop Task	interrupts the task displayed.
Task Monitored	selects the task to be monitored.
Breakpoints → Insert	positions the breakpoint on the program element displayed in the predefined screen.
→ Remove	deletes the breakpoint.
→ View	displays the place where the breakpoint was defined.
Continue	executes a task scan.
Step by Step	executes the next element.
Last Stop	displays the last element executed.
Data Mode	used to access DATA mode.
Data Line mode	used to access DATA LINE mode.

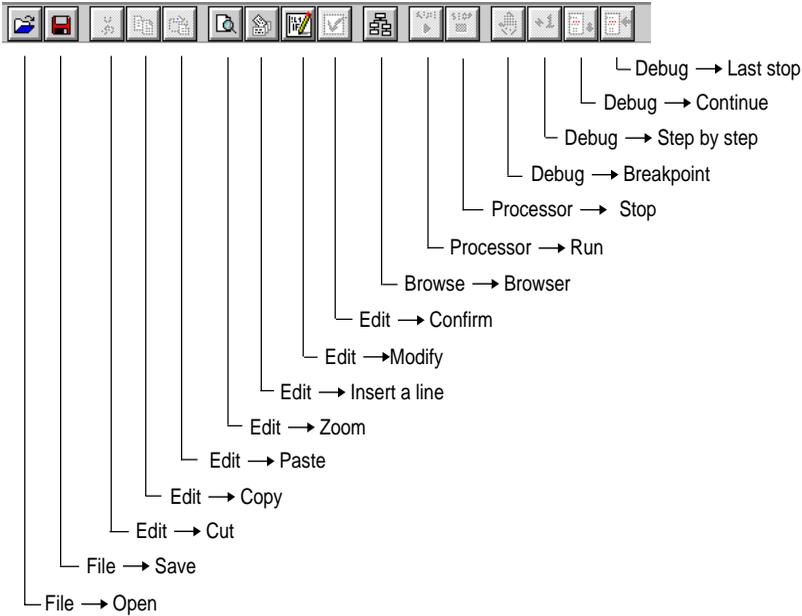
BROWSE MENU :

Symbolize window	accesses the symbols line window.
Data window	accesses the data line window.
Browser	displays the Program Browser window.
Screen 1...6	switches from one predefined screen to another.

8.3-3 Toolbar

This toolbar provides quicker access to the main functions on the menu bar in LITERAL mode.

Toolbar button definitions :

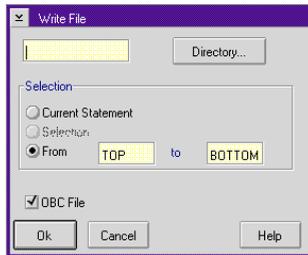


8.3-4 Archiving and retrieving a Literal language module

Archiving

File, Save	<CTRL>+<S>	used to archive the current Literal module, in a “source” file on disk. This xxx.LIT file, saved in the PL7_3\MOD directory, allows the module to be reused in another application and integrated in a module base.
-------------------	-------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The following window is displayed :



The entry field is used to define the file name.

The **Directory** button is used to access the list of Literal program files (.LIT files) in the application.

The **Current Statement** button only saves the statement highlighted by the cursor.

The **Select** button only saves the selected statements : these statements appear in reverse video in the editor.

The **FROM** button is used to select a statement to be saved.

The **OBC** check box is used to archive the Literal language module, with or without the internal OFB constants. If the OBC option is selected, a <name_OFB>.OBC file is created for each module OFB containing internal constants.

In addition, an xxx.OBM file (with the same name as the xxx.LIT file) gives a list of all the OFBs which have generated .OBC file.

Help calls up the program help function.

<TAB> used to move from one group to the other.

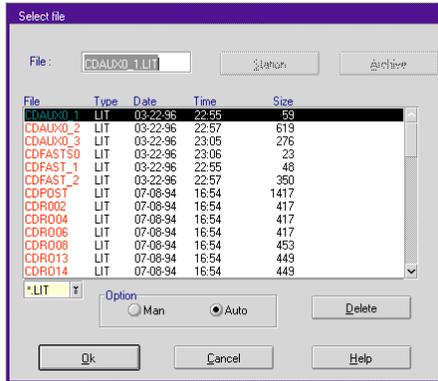
<↑><↓> used to move within a group.

<SPACE> used to activate the buttons and the check boxes.

Retrieving

File, Open	used to retrieve a “source” file to the working RAM memory of the terminal. Accesses a list of all the .LIT files in the application.
-------------------	---------------------------------------------------------------------------------------------------------------------------------------

The following window is displayed :



The entry field is used to define the file name.

The list of files is used to select a file to be retrieved from those in the current directory.

The **AUTO** and **MAN** buttons define the retrieval mode :

AUTO : global retrieval with stop on error only,

MAN : manual retrieval with confirmation on each element read.

Archive assigns the global archive name (PL7_3) as the file name.

Station assigns the name of the current station as the file name.

OK saves the selection made and starts retrieving the file selected.

Cancel cancels the selections made and exits the dialog box without triggering any action.

Help calls up the online help function.

<TAB> used to move from one group to the other.

<↑><↓> used to move within a group.

<→><←>

<↑><↓> used to move within the list or the inset.

Note

For more information on the V5 modular version, see section 15 - Appendix.

8.4 Literal language zoom function

The Literal language zoom function accesses, from the DISPLAY screen, the contents of :

- the parameters of a function,
- a subroutine called by the CALL instruction.

Procedure for using the zoom function

Edit, Zoom	<CTRL>+ <PG DOWN>	from the DISPLAY screen, this key activates ZOOM mode. The line pointer must already be on the function to be displayed.
-----------------------	------------------------------------------	--------------------------------------------------------------------------------------------------------------------------

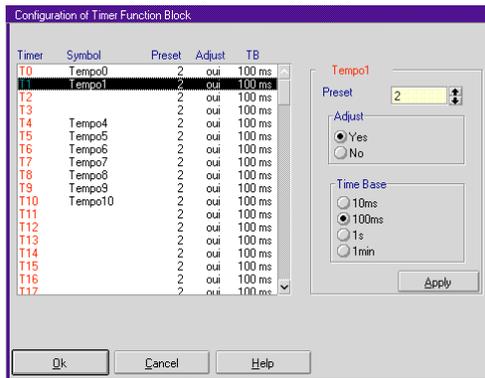
8.4-1 Zooming in on a function block

Several zoom levels are permitted on control system functions.

Example : monostable M1 and counter C0,

The terminal is in DISPLAY mode and the cursor is positioned on the function block to be “zoomed” into.

Edit, Zoom	<CTRL>+ <PG DOWN>	used to read or modify the function configuration parameters.
-----------------------	------------------------------------------	---------------------------------------------------------------



OK used to return to the calling screen and take account of any modifications to the block configuration parameters.

Cancel used to return to the calling screen leaving the configuration unchanged.

8.4-2 Zooming in on a CALL instruction

The terminal is in DISPLAY mode and the cursor is positioned on the CALL instruction, calling the subroutine to be displayed, modified or entered.

Eight zoom levels can be used in the case of nested subroutines.

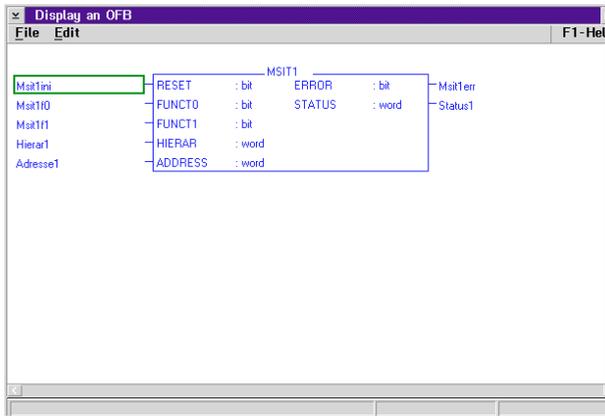
Edit, Zoom	<CTRL>+ <PG DOWN>	accesses the subroutine (descends one zoom level).
	<CTRL>+ <PG UP>	moves up one zoom level.
Menu, Quit		returns to the first zoom level.

If the subroutine has not yet been programmed, the terminal gives the option of selecting the language in which it must be entered.

8.4-3 Zooming in on an OFB

The terminal is in DISPLAY mode and the cursor is positioned on the EXEC instruction of the OFB to be displayed, modified or entered.

Edit, Zoom	<CTRL>+ <PG DOWN>	used to read or modify I/O parameters and internal OFB constants.
-----------------------	------------------------------------------	-------------------------------------------------------------------



Edit, Constants		accesses the internal OFB constants.
File, Quit		returns to the calling screen

8.5 Creating or modifying the program

8.5-1 Principle of data entry

To create or modify a statement, as described in this section, the terminal must be in INSERT or MODIFY mode.

Modifying a program from DISPLAY mode

Edit, Modify line	<CTRL>+<M> or <F4>	accesses modification of the program line indicated by the line pointer.
Edit, Insert line	<INS>	accesses the INSERT mode, so as to insert a program line before the current line.

Creating a program from DISPLAY mode

Edit, Insert Line	<SPACE>	accesses the function for writing the program line by line. The terminal switches to INSERT mode.
------------------------------	----------------------	---------------------------------------------------------------------------------------------------

It is also possible to return directly to the line to be entered. Simply ensure that the cursor is already at the required insertion point.

Statement address

Previously entered program line

```
AUX0 - MAIN - L122+2                                CDTBXTOR CDAUX0_3 RW

|      R034,0,8[8]→R012,0,8[8]
|      R012,0,8[8]→R058,0,8[8]
|
|***** END      FLASHING ON TEX DMS16xx *****
|I121  :
|***** START    FLASHING ON TEX DMS1025 *****
|I122  :
|      IF RI22,0,6 THEN Seconde→R09,0,8→R09,0,9
|              ELSE RESET R09,0,8;RESET R09,0,9
|
|      R09,0,8[2]→R010,0,8[2]
|
|***** END      FLASHING ON TEX DMS1025 *****
|I123  :
|EOP
```

Program line being entered

Rules of entry

Writing a program line includes :

- optional entry of the label,
- optional entry of the associated comment,
- entry of the Literal expression.

After entering or modifying a program line, it is compiled and stored in the memory by pressing **<ENTER>**. It then appears in the upper part of the screen and can be viewed in DISPLAY mode. If the terminal was in MODIFY mode during data entry, it will now be in DISPLAY mode. If the terminal was in INSERT mode during data entry, a new insertion is offered.

Syntax rules

At least one space must be left between the objects, separators and instructions represented by alphanumeric characters.

Example : IF B1 THEN W2 AND W3 → W6

No space must be left between the characters of a single object, separator or instruction.

A space is not compulsory between objects, separators and instructions represented by operation signs or punctuation marks.

Example : IF [W5>7] THEN (W2+W6)*3 → W8;W1 → W9

Entry errors

During entry, illegal keys are indicated by an audible warning. After confirming an insertion or a modification, if the entry is incorrect, an error message is displayed and the cursor is positioned on the error so that it can be corrected. The possible errors are :

- incorrect syntax,
- access to an object which is not configured,
- programming an illegal instruction,
- creating a label which already exists in the module.

8.5-2 Entering or modifying a label

Entering a label is optional. One label is sufficient for defining a program line. The label number must be between 1 and 999 and can only be used once in the program.

To enter a label, switch to MODIFY mode, position the cursor to the right of the mark for the start of statement "!" and type in the number of the required label.

To modify a label, make corrections using the <DELETE>, <BACKSPACE> and arrow keys or the mouse.

To delete a label, delete the "L" or delete all the characters of the label using the <DELETE> and <BACK SPACE> keys. Alternatively, a quicker way is to position the cursor after the "L" and delete it using the <BACKSPACE> key. The whole label will be deleted.

8.5-3 Entering or modifying a comment

Entry of a comment is optional.

After confirmation, the comment, composed of 77 characters maximum, appears before the program line.

To enter a comment :

- switch to MODIFY mode,
- position the cursor above the statement to receive the comment,
- enter the comment,
- press <ENTER> to confirm the entry and return to DISPLAY mode.

To modify a comment, position the cursor on the comment and switch to MODIFY mode.

To delete a comment, delete all its characters.

8.5-4 Entering or modifying a Literal expression - completion

Instructions are entered or modified using the alphanumeric keyboard (typed in full).

A completion function is used to speed up entry of key words. To use the completion function, simply enter the first character(s) which make up the key word, and press the **<SPACE>** bar. If, having entered the characters, there is no ambiguity as to what the word might be, it is automatically completed. Otherwise a list appears enabling the required word to be selected.

Examples :

- Entering "T", then pressing the **<SPACE>** bar : "THEN" is written automatically (it is the only possibility).
- Entering "IN", then pressing the **<SPACE>** bar : a selection box appears, enabling the user to select one of the following : "INC", "INFF", "INIT" and "INPUT".

• Modifying entries

- <←><→>** move the cursor horizontally within the statement.
- <↑><↓>** move the cursor vertically within the statement.
- <BACK SPACE>** deletes the character to the left of the cursor.
- <DELETE>** deletes the character selected by the cursor.

Entering an OFB

In order to simplify entry of the I/O parameters of an OFB, it is possible to obtain a graphic representation of the OFB by pressing the **<CTRL>+<O>** keys in MODIFY mode. The statement currently being edited is analyzed in order to find the EXEC key word followed by an OFB and its instance (eg : EXEC MREADO).

If the key word is not found, the user is able to select an OFB from the list of OFBs configured in the application.

Once the OFB has been selected, the OFB parameter editor is called (instance 0 is suggested by default). (For more information on this, see the section on the Ladder language editor).

8.5-5 Floating point type instructions

These instructions are only available for PMX V5 processors equipped with a math coprocessor. They are not available on TSX processors.

Floating point objects do not exist in PL7-3. These floating point type instructions manipulate internal double words DWi and constant double words CDWi or double word type OFB extract objects.

Immediate values in floating point format are only allowed for direct assignments. These values can be entered with a fixed decimal point (eg 435.26) or with a floating point (eg 4.3526e+02 or 43.52e1). In all cases, the value is stored in the following format : 1 digit before the decimal point, a maximum of 6 digits after the decimal point and 2 figures maximum for the exponent. Taking into account a possible sign for the mantissa and the exponent, the decimal point and the character E (or e) to separate the mantissa and the exponent, this allows a maximum of 13 characters.

For a fixed decimal point, it is possible to enter as many digits as allowed for floating point numbers. However the value may be altered when storing in floating point format.

For example :

- if the user entered the value 123456789.25
- the value stored will be 1.234567e8 ie. 123456700

Whatever the format of the immediate value (floating point or fixed point), the character "." **must** be entered to define the decimal point.

Examples

Value entered	Value stored
123.	1.23e2
445.e26	4.45e28
0.0	0.
1234567890123.	1.234567e12
123	this value is not accepted : integer
445e26	this value is not accepted : no "." character
0	this value is not accepted : integer

Assigning an immediate value function

Description	Example	Function
->	1.42 E+02->DW25	Assignment in floating point format of an immediate value in a double word.

Calculation functions

Description	Example	Function
ADDF	ADDF(DW10;DW20)->DW30	Addition of 2 operands in floating point format.
SUBF	SUBF(DW10;DW20)->DW30	Subtraction of 2 operands in floating point format.
MULF	MULF(DW10;DW20)->DW30	Multiplication of 2 operands in floating point format.
DIVF	DIVF(DW10;DW20)->DW30	Division of 2 operands in floating point format.
SQRTF	SQRTF(DW20)->DW30	Calculation of the square root of an operand in floating point format.

Conversion functions

Description	Example	Function
BTF	BTF(DW20)->DW30	Conversion of an integer to a floating point number.
ATF	ATF(DW20)->DW30	Conversion of an ASCII character string to a floating point number.
FTB	FTB(DW20)->DW30	Conversion of a floating point number to an integer.
FTA	FTA(DW20)->DW30	Conversion of a floating point number to an ASCII character string.
DTF	DTF(DW20)->DW30	Conversion of a BCD number to a floating point number.
FTD	FTD(DW20)->DW30	Conversion of a floating point number to a BCD number.

Comparison functions

Description	Example	Function
EQUF	EQUF(DW10;DW20)->B3	Test the equality of 2 operands in floating point format.
SUPF	SUPF(DW10;DW20)->B3	Test that the first operand is always greater than the second in floating point format.
INFF	INFF(DW10;DW20)->B3	Test that the first operand is always less than the second in floating point format.

8.5-6 Diagnostics associated with a program line

It is possible to associate a diagnostic check with a program line, in order to monitor a Boolean condition or a transition. This diagnostic check is performed by calling an application diagnostic optional function block (OFB). Some of these OFBs can cause the source of the monitored statement to be memorized. It is not possible to delete the last diagnostic check, or to add the first diagnostic check with memorization of the source, when the terminal is connected to a PLC in RUN mode.

Editing a Diagnostic OFB	<CTRL>+<D>	adds/removes the diagnostics for a program line, in INSERT or MODIFY mode.
View, Display Diagnostic OFB		displays/hides the diagnostics. If the diagnostic OFBs are hidden, the letter "H" appears to the left of the monitored lines. Otherwise, the "Diagnostics : " expression appears below the monitored lines, as well as the call to the corresponding diagnostic OFB.

Entry or modification of the characteristics of diagnostic OFBs

Position the cursor in the diagnostics zone (marked by the "<diagnostics>" label). Select the **Edit OFB** option from the **Edit** menu, then continue using the same procedure as that for any OFB.

8.5-7 Confirming a program line

In INSERT or MODIFY mode, when a program line is entered or modified, it must be confirmed so that it can be compiled and stored in the memory.

<ENTER> confirms the entry.

8.5-8 Deleting a module or a program line

Deleting a module

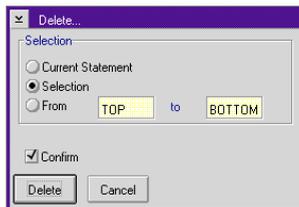
This is performed from the PROGRAM BROWSER notepad which is used to select the module to be deleted. A task and its subsets are selected in the Browser using the <↓>, <↑> keys to select the element required, and the <←>, <→> keys to move between columns.

Empty Module after confirmation, deletes the contents of the selected module.

Deleting a program line

Deletion of statements applies to current statements, to the current selection (see following section for selection of several statements) or a range of statements.

In DISPLAY mode, pressing the <DELETE> key or selecting the **Delete** option in the **Edit** menu, displays the following dialog box :



The buttons in the selection group are used to select which statements are to be deleted :

- current statement : statement containing the cursor will be deleted,
- selection : if one or more statements are selected, they will be deleted,
- from <label> to <label> : a range of statements defined by the statements in the two entry zones will be deleted.

Note :

The statements defining the range of statements must be marked by TOP, BOTTOM, a label (eg : L20) or a label plus a movement (eg : L30+1, L40-2).

The confirmation box indicates whether deletion of each statement must be confirmed prior to execution.



- Yes** deletes the statements and continues the operation.
- No** does not delete the statements and continues the operation.
- Yes to All** deletes the statements and continues the operation without requesting confirmation.
- Cancel** cancels the operation.

Note

An operation to delete without confirmation can be stopped by pressing the **<ESCAPE>** key.

8.5-9 Duplicating a program line

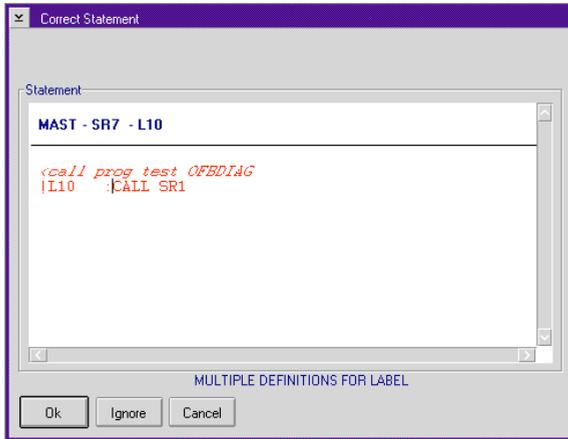
This procedure is used to copy one or more statements into a buffer memory, and to duplicate them any number of times in any module programmed in Literal language. The statements copied are those which have been selected.

Copying the statement into the buffer memory

Edit, Copy	<CTRL>+<INS>	used to copy the selected statements in DISPLAY mode. These statements remain in the buffer memory until the next copy action is performed.
Edit, Cut	<SHIFT>+	in addition to being copied into the buffer memory, the selected statements are deleted.

Restoring copied statements

They can only be restored in DISPLAY mode in the same programming module or in another module. If restoring a statement causes an error, the following dialog box appears for correcting the error.



**Edit,
Paste**

<SHIFT>+<INS>

restores the memorized statements.

OK confirms the corrections made.

Ignore ignores the statement (it will not be inserted) and continues the operation.

Cancel stops the insert operation.

8.5-10 Multiple selection of statements

Multiple selection of statements is used to define a list of statements (contiguous or not) which can then be :

- deleted,
- copied into a buffer memory,
- written in a source file.

There are two ways of performing multiple selection :

KEYBOARD :

- position the cursor in the statement to be selected or deselected (arrow keys).
- The **<SPACE>** bar or the **Select** option in the **Edit** menu are used to select the current statement and deselect previously selected statements.
- **<CTRL>+<SPACE>** adds the current statement to the statement already selected. Deselects the statements if they have already been selected.
- **<SHIFT>+<SPACE>** adds the statements between the last selected statement and the current statement to the current selection.

MOUSE :

- Position the mouse pointer tool to the left of the statement to be selected or deselected (before the power rail). The mouse pointer tool then changes to a horizontal arrow.
- Clicking with the **left-hand mouse button** selects the current statement and deselects statements which have previously been selected.
- **<CTRL>+ Click with left-hand mouse button** adds the current statement to those already selected. Deselects the statements if they have already been selected.
- **<SHIFT>+ Click with left-hand mouse button** adds the statements between the last selected statement and the current statement to the current selection.
- By clicking and holding down the left-hand mouse button and moving to the top or bottom edges of the screen, the user can scroll through the statements. Moving the mouse pointer tool towards the top edge of the screen scrolls to the start of the program; moving the pointer to the bottom edge of the screen scrolls to the end of the program. In this way every new statement appearing on the screen will be selected, or deselected if the last operation was a deselection operation (**<CTRL>+<SPACE>** or **<CTRL>+Click** on a previously selected statement).

8.6 Searching for the program addresses of a variable

- <Alt>+<X>** When a program is displayed from a selected statement, this command is used to access the :
- display of all the variables in a statement,
 - selection of a variable and display of all the program addresses relating to this variable,
 - selection of a program address and display of the corresponding program element.

The variable is either a standard PL7-3 variable (Wi, Bi etc) or an OFB variable. This function is explained in section 15.5 of part H.

Section		Page
9.1	Presentation of CONSTANT mode	9/2
9.2	Constant words	9/3
	9.2-1 Displaying constant words	9/3
	9.2-2 Modifying constant words	9/3
	9.2-3 Saving and retrieving constant words	9/4
9.3	Internal OFB constants	9/6
	9.3-1 Selecting an OFB	9/6
	9.3-2 Displaying and modifying the internal constants of an OFB	9/7
9.4	Managing Grafcet situations	9/10
	9.4-1 Definition of the descriptor of a partial Grafcet chart	9/11
	9.4-2 Definiting a forcing situation	9/12

9.1 Presentation of CONSTANT mode

CONSTANT mode is used to read and modify :

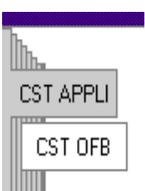
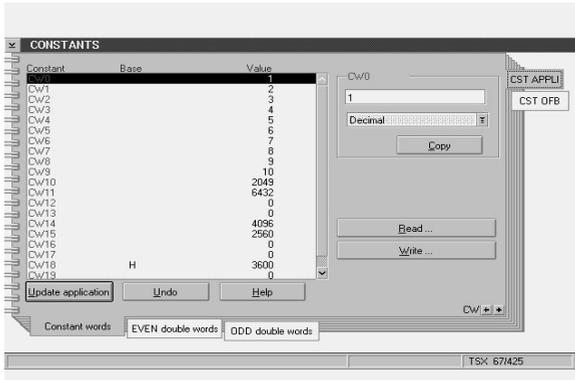
- values of the single and double length constant words declared in the configuration (128 by default).
- values of the internal OFB constants declared in the configuration.

This mode is also used to manage Grafcet situations.

It is also possible to read constant words and OFB constants when online, from DATA and DEBUG modes.

The number of constant words and the number and type of usable OFBs are declared in CONFIGURATION mode (see section 5).

Tools, Constants accesses constant mode. The following screen is displayed :



APPLI CST accesses the display and modification of constant words.

OFB CST accesses the display and modification of internal OFB constants and of the parameters required for managing Grafcet situations: Grafcet descriptors, forcing situations.

Read is used to read constant words from a file.

Write is used to save constant words on a file.

<TAB> is used to move from one group to another.

<↓>, <↑> are used to move within a group.

<←>, <→>

Undo is used to return the data on a notepad page to its initial state (ie. when the page was selected).

Help calls up the program help function.

9.2 Constant words

9.2-1 Displaying constant words

The value of constant words, which is 0 by default, can be displayed in the form of single length words or double length words.

The double length words occupy the same geographic location as single length words in the same position : CDWi occupies the same location as the words CWi and CWi+1.

Each constant word is associated with a display base (decimal by default). The possible display bases are :

- decimal,
- hexadecimal,
- DCB,
- logic,
- message,
- start of message,
- end of message,

Double length constant words can be aligned to the CWi words with even addresses (EVEN DOUBLE WORD) or with odd addresses (ODD DOUBLE WORD).

Constant Words tab : displays the single length constant words

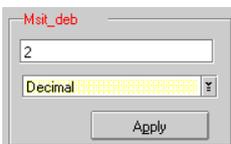
Even Double Words tab : displays the CWi words with even addresses.

Odd Double Words tab : displays the CWi words with odd addresses.

The display base is associated with the acquisition of the immediate value : "floating point" base or decimal base. It is impossible to alter the display base of double words, due to the incompatibility of integer and floating point formats. In "floating point" base, display of "?????" indicates that the value does not appear in the definitions of floating point numbers.

All CDWi declared to be floating point type in SDBASE (with the letter F in the Display field) are automatically displayed in the "floating point" base.

9.2-2 Modifying constant words



The entry field is used to modify the value of the active constant (highlighted) and all selected constants.

The pull-down list modifies the active constant base (single length word only).

The possible bases are as follows : decimal (DEC), hexadecimal (HEX), binary coded decimal (BCD), logic (BIN), message (MES), start of message (SOM) and end of message (EOM).

Copy is used to copy the value and the active constant base to all selected constants.

Update application

confirms all the modifications made to the constant words displayed on the screen.

Undo

cancels all the modifications made to the constant words since entering the notepad page.

Notes

Entries in message mode are not limited to 2 characters, but can comprise a string of 64 characters maximum. In this case, the characters are stored 2 by 2 in the words following the one selected. Care must therefore be taken to ensure that the entry of a string of characters does not modify the contents of the following words.

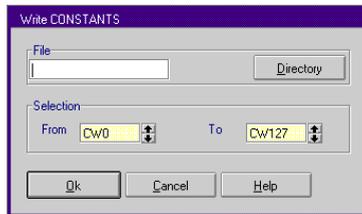
An entered value may have fewer characters than are normally required by the format of the selected code, in which case the missing characters are considered as zeros and as most significant.

Example : value entered 8C
 value memorized 008C

9.2-3 Saving and retrieving constant words

Write is used to save the constant words in a file. The following screen is displayed :

Directory lists all the files already archived in the PL7-3MOD subdirectory.



The entry field is used to specify the name of the archive file.

The **FROM** field is used to define the name or symbol of the first constant word to be archived.

The **TO** window is used to define the name or the symbol of the last constant word to be archived.

<TAB> is used to move from one group to another.

<↓>, <↑> is used to move within a group.

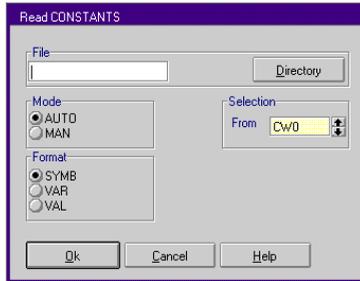
<→>, <←>

<SHIFT>+ is used to move from one incremental entry zone to another.

<↓>, <↑>

Help calls up the program help function.

Read is used to read the constant words previously saved in a xxx.CST file. The following screen is displayed :



Directory lists all the files already archived in the PL7-3MOD subdirectory.

The **Auto** and **Manual** buttons define the mode of retrieval :

- **AUTOMATIC** : global retrieval, stopping on errors only,
- **MANUAL** : manual retrieval with confirmation of each element read.

The **SYMB**, **VAR** and **VAL** buttons are used to define the read mode :

- **SYMBOL** : reading directed by the symbols,
- **VARIABLE** : reading directed by the CW,
- **VALUE** : reading directed by the values.

The **FROM** field is used to define the name or symbol of the single or double length constant word which will receive the first value read (beginning of the retrieval zone).

<TAB> is used to move from one group to another.

<↓>, <↑> is used to move within a group.

<→>, <←>

Help calls up the program help function.

Warning : if the file read contains symbols, these will define the zones to be retrieved. For example :

file read	table of symbols
MAX = 50	MAX: CW5
CW4 = 52	MIN : CW12
MIN = 20	
Retrieval from MAX to MIN :	CW5 = 50
CW6 = 52	
CW12 = 20	

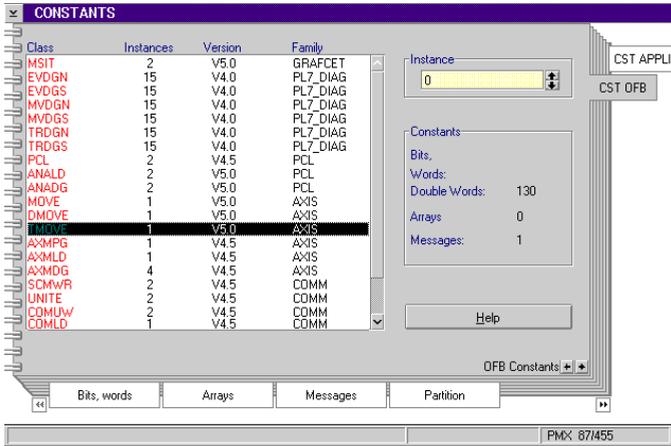
Note

Double words are saved and then retrieved in their display base : floating point or decimal (a CDW displayed in floating point format is therefore saved in floating point format).

9.3 Internal OFB constants

9.3-1 Selecting an OFB

This screen accesses the display or modification of the application internal OFB constants.



An OFB is selected by selecting the required OFB from the list, and entering the number in the **Instance** entry field.

The **Constants** zone is used to display the number of Bits, Words, Double Words, Tables, and Messages for the selected OFB.

Display of the selected internal OFB constants is accessed by selecting one of the secondary tabs which define the type of constants displayed.

To return to the OFB selection screen, select the **OFB CST** main tab.

<TAB> is used to move from one group to another.

<↓>, <↑> is used to move within a group.

<→>, <←>

Help calls up the program help function.

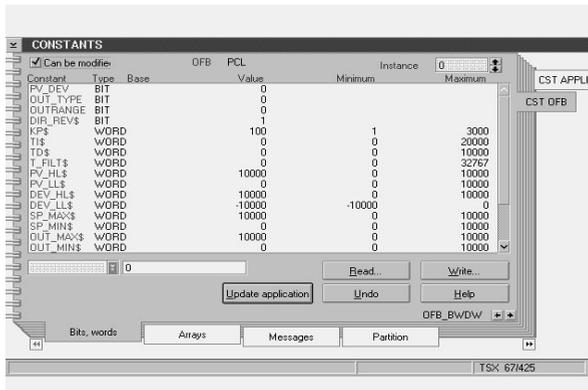
9.3-2 Displaying and modifying the internal constants of an OFB

The internal constants of an OFB are presented on 3 types of screen which display respectively :

- bit, word or double word type constants,
- word table or double word table type constants,
- message type constants.

These screens can be accessed at the bottom of the notepad using the secondary tabs. If one of these types of constant does not exist for a given OFB, the corresponding screen is not shown.

Bit, word or double word type constant :



is used to modify the base of the selected internal constant.

The authorized bases are : decimal, hexadecimal and logic.

With a PMX station, double words can also be displayed in floating point format. In this case, the MIN and MAX values are also displayed in floating point format. If a value does not appear in the definition of floating point numbers, the column shows "????".

If a double word is declared as floating point type in SDBASE (with the letter F in the Display field) or if it is a floating point type OFB element (for example an internal OFB constant from the PMS2 family), it is displayed in the "floating point" base by default.

The value entry zone is used to modify the selected constant. With a PMX station, double words can be entered as floating point values. In this case, if the immediate value entered does not appear in the definition of floating point words, a syntax error is indicated.

The **Modifiable** box, which can be ticked, allows or prohibits modification of the internal constants of the current OFB, using ADJUST software.

The **Instance** entry field is used to determine the instance of OFB to be displayed.

Update application

confirms all the modifications made to the constants of the current OFB.

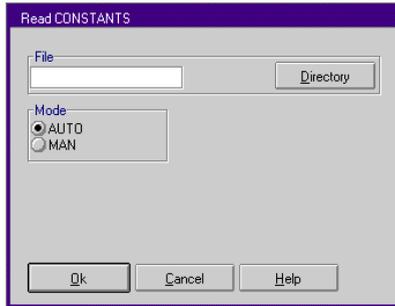
Undo

cancels all the modifications made since entering the notepad page.

Help

calls up the program help function.

Read is used to read the OFB constants previously saved in an xxx.OBC file. The following screen is displayed :



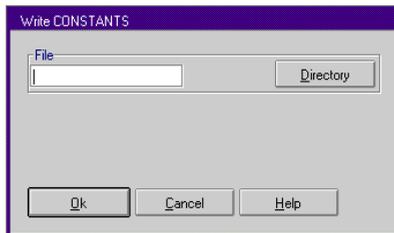
The list of existing files is used to select the name of the file to be retrieved.

Directory lists all the files already archived in the PL7-3\MOD subdirectory.

The **Auto** and **Manual** buttons define the mode of retrieval :

- **AUTOMATIC** : global retrieval, stopping on errors only,
- **MANUAL** : manual retrieval with confirmation of each element read.

Write is used to store internal OFB constants in a file. The following screen is displayed :



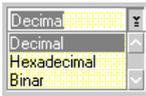
The entry field is used to specify the name of the archive file.

The list is used to select the name of an existing archive file. When the save operation is performed, the new file will overwrite the old one with the same name.

Directory lists all the files already archived in the PL7-3\MOD subdirectory.

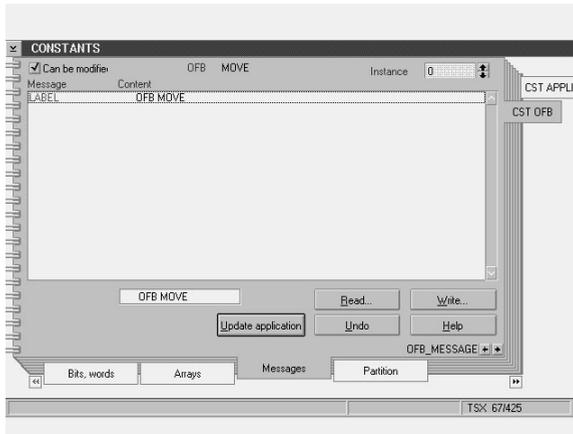
Archive assigns the global archive name (PL7_3) as the name of the file.

Station assigns the name of the XTEL station as the name of the file.

Table type constant :

is used to modify the base of the selected table. The authorized bases are : decimal, hexadecimal and logic. With a PMX station, double words can also be displayed in floating point format.

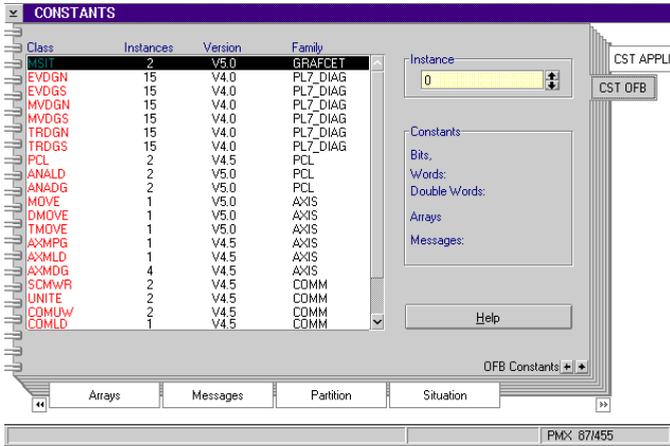
The entry zone is used to modify the value of the element of the selected table.

Message type constant :

The entry zone is used to define the message of the selected OFB.

9.4 Managing Grafcet situations

CONSTANT mode is used to define the descriptor for a partial Grafcet chart and to initialize the different forcing situations for each MSIT OFB. From the OFB selection screen, accessible by using the **CST OFB** tab in the CONSTANT mode entry screen (see section 9.3-1), select the MSIT OFB to be defined :



Partition this secondary tab is used to define the partial Grafcet chart descriptor :

- partition associated with a partial Grafcet chart; namely steps and macro-steps which it comprises,
- hierarchy of a partial Grafcet chart.

Situation this secondary tab is used to define the forcing situation of the partial Grafcet chart. This will be stored in the internal constants of the application in accordance with the following rule : the number of CWs occupied is $9 * \text{number of modules (CHART, XM) in the partition} + 1$.
The screen associated with the **Situation** tab will only be displayed if the partial Grafcet chart descriptor has been defined. Otherwise, the message "Empty Partition" is displayed.

9.4-1 Definition of the descriptor of a partial Grafcet chart

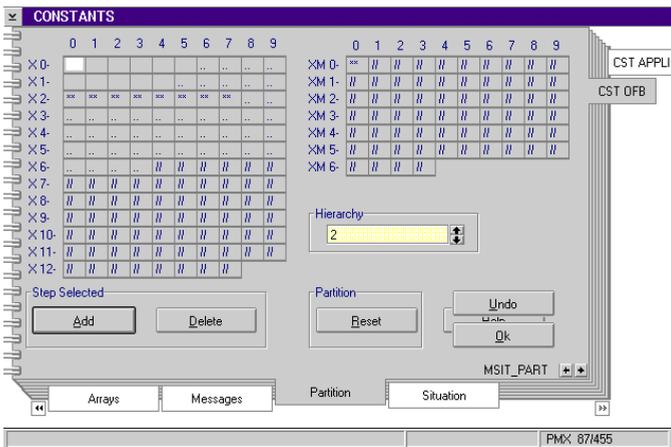
This screen can be accessed from the OPTIONAL FUNCTION BLOCK CONFIGURATION screen using the **Partition** secondary tab. It is used to define the hierarchical level and the steps and macro-steps which comprise the partial Grafcet chart, associated with the selected MSIT OFB.

The steps and macro-steps of the application are shown as a table containing :

- empty boxes : steps or macro-steps configured and called,
- "." boxes : steps or macro-steps configured but not called,
- "/" boxes : steps or macro-steps not configured.

The arrow keys or the mouse enable movement within the empty boxes (steps or macro-steps configured and called) in order to select the steps or macro-steps which make up the partial Grafcet chart.

The "*" boxes represent the steps or macro-steps selected.



- Add** selects the step or macro-step highlighted by the cursor.
- Delete** deletes the step or macro-step highlighted by the cursor.
- Reset** after confirmation, cancels all selections of steps and macro-steps.
- Hierarchy** is used to define the hierarchical level of the partial Grafcet chart : 0 to 255. 0 corresponds to the highest hierarchical level and 255 to the lowest.
- <TAB>** is used to move from one group to another.
- <↓>, <↑>** is used to move within a group.
- Help** calls up the program help function.

F



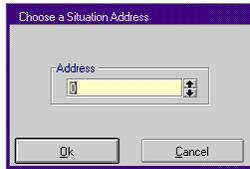
move the cursor in the accessible boxes : empty or "" boxes.



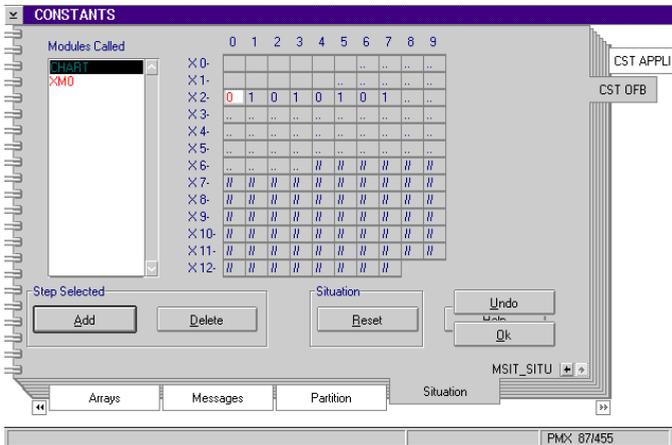
On leaving the partition definition screen, a confirmation window is used to confirm the actions made and to update the descriptor of the partial Grafcet chart, contained in the internal constants of the MSIT OFB.

9.4-2 Defining a forcing situation

Having defined the descriptor of a partial Grafcet chart, a forcing situation can be defined : list of active steps. To do this, from the OPTIONAL FUNCTION BLOCK CONFIGURATION screen, select the **Situation** secondary tab which allows the forcing situation for the partial Grafcet chart, associated with the selected OFB MSIT, to be defined. This situation will be saved in the internal constants (CWi) of the application, from the selected address :



The program then indicates the number of CWs occupied by the new situation. After confirmation by **OK**, the module selection screen is displayed :



This screen is made up of two sections :

- a list of accessible modules,
- a configuration table associated with a module selected from the previous list.

The list in the called modules zone is used to select a module : the corresponding screen is displayed in the configuration table which contains :

- empty boxes : configured steps, which are called but do not appear in the partial Grafcet chart,
- "." boxes : configured module steps, which are not called,
- "/" boxes : non configured module steps,
- "0" boxes : called, configured steps, which are part of the partial Grafcet chart, but not of the forcing situation,
- "1" boxes : called, configured steps, which are part of the partial Grafcet chart and the forcing situation.

Add selects the step highlighted by the cursor, so that it is included in the forcing situation. The box then takes the value "1".

Delete cancels the selection of the step highlighted by the cursor, so that it is no longer included in the forcing situation and the box then takes the value "0".

Reset after confirmation, cancels all selections of steps.

Help calls up the program help function.

<↑><↓> move the cursor to the accessible steps.

<←><→>

<TAB> is used to change the focus between the list of modules, the table and the buttons.

On leaving the forcing situation screen, the module for which the forcing situation is defined appears preceded by "****" in the list of called modules.

Section	Page
10.1 Presentation of DATA mode	10/2
10.1-1 Role	10/2
10.1-2 Objects accessible in DATA mode	10/3
10.2 Description of the screen and keys	10/6
10.2-1 Description of the screen	10/6
10.2-2 Menu bar	10/7
10.2-3 Toolbar	10/8
10.2-4 Role of keys	10/9
10.3 Reading data	10/11
10.3-1 Reading objects	10/11
10.3-2 Reading a list of consecutive objects (RH)	10/12
10.3-3 Object log	10/13
10.3-4 Searching for forced or blocked objects	10/13
10.3-5 Selection of the index or address	10/14
10.4 Writing and forcing data	10/15
10.4-1 Writing and forcing a bit	10/15
10.4-2 Writing a list of consecutive bits	10/16
10.4-3 Modifying and blocking a step	10/17
10.4-4 Writing words	10/18
10.4-5 Writing a list of consecutive words	10/18
10.4-6 Saving and restoring a predefined data screen	10/19

If the terminal does not already have in its internal memory the application of the PLC to which it is connected, it suggests transferring the application from the PLC to the terminal (Processor → terminal transfer; see section 12).

Yes launches the Processor → terminal transfer.

No cancels the transfer request.

10.1-2 Objects accessible in DATA mode

The tables below give the list of objects accessible in DATA mode, together with the possibilities offered by each of them.

Bit objects

Object		Reading	Writing (RE)SET	Forcing (0/1)	Index (FD)
"In rack" I/O module	I/Oxy,i	yes	yes	yes	3(x-y-i)
"In rack" I/O module	I/Oxy,s	yes	no	no	2(x-y)
Internal bit	Bi	yes	yes	yes	1(i)
System bit	SYi	yes	yes	no	1(i)
Grafcet bit	Xi	yes	yes	blocking	1(i)
Grafcet bit	XMi	yes	no	no	1(i)
Grafcet bit	Xj,i	yes	yes	blocking	2(i-j)
Remote I/O module	RI/ROx,y,i	yes	yes	yes	1(i)
Remote I/O module	RDx,y,i	yes	no	no	1(i)

Word objects

Object		Reading	Writing (RE)SET	Base	Index (FD)
Register word	I/OWxy,i	yes	yes	no	3(x-y-i)
Internal word	Wi - DWi	yes	yes	yes	1(i)
Constant word	CWi - CDWi	yes	yes	yes	1(i)
System word	SWi	yes	yes	yes	1(i)
Common word	COM(X)i,j	yes	yes	yes	2(i-j)
Grafcet step activity time value	Xi,V - Xj,i,V	yes	no	yes	1(i)
Remote I/O module	STSx,y,i	yes	no	yes	1(i)
	RiW-ROWx,y,i	yes	no	yes	1(i)

Bit objects extracted from words

Object		Reading	Writing (RE)SET	Forcing (0/1)	Index (FD)
Register word	I/Ow _{xy,i,j}	yes	no	no	4(x-y-i-j)
Internal word	W _{i,j}	yes	yes	no	2(i-j)
Internal word	DW _{i,j}	no	no	no	2(i-j)
Constant word	CW _{i,j}	yes	yes	no	2(i-j)
System word	SW _{i,j}	yes	yes	no	2(i-j)
Common word	COM(X) _{i,j,k}	yes	yes	no	3(i-j-k)
Remote I/O module	STS _{x,y,i,b}	yes	no	no	1(i)
	RIW _{x,y,i,b}	yes	no	no	1(i)
	ROW _{x,y,i,b}	yes	yes	no	1(i)

Optional function block objects

Object	Reading	Writing	Forcing (0/1)	Base	(RE) SET	Index (FD)
Timer						
T _{i,P}	yes	yes (1)	-	yes	-	1(i)
T _{i,V}	yes	no	-	yes	-	1(i)
T _{i,R} - T _{i,D}	yes	no	no	-	no	1(i)
Monostable						
M _{i,P}	yes	yes (1)	-	yes	-	1(i)
M _{i,V}	yes	no	-	yes	-	1(i)
M _{i,R}	yes	no	no	-	no	1(i)
Counter						
C _{i,P}	yes	yes (1)	-	yes	-	1(i)
C _{i,V}	yes	no	-	yes	-	1(i)
C _{i,E} - C _{i,D} - C _{i,F}	yes	no	no	-	no	1(i)
Register						
R _{i,I}	yes	yes	-	yes	-	1(i)
R _{i,O}	yes	yes	-	yes	-	1(i)
R _{i,E} - R _{i,F}	yes	no	no	-	no	1(i)
Text						
TXT _{i,L}	yes	yes	-	no	-	1(i)
TXT _{i,S/V}	yes	no	-	no	-	1(i)
TXT _{i,A/M/C/T}	yes	yes	-	no	-	1(i)
TXT _{i,D/E}	yes	no	no	-	no	1(i)
Control						
CTRL _{i,R}	yes	no	no	-	no	1(i)

(1) if the block is configured with modification.

Optional function block objects

Object	Reading	Writing	Forcing (0/1)	Base	(RE) SET	Index (FD)
OFB						
OFBi,input	yes (2)	yes (2)	no	yes	yes	1(i)
OFBi,output	yes	no	no	yes	no	1(i)
OFBi,constant	yes (3)	yes (3)	no	yes	yes	1(i)
OFBi,data	yes (4)	yes (4)	no	yes	yes	1(i)

(2) bits, words and double words only,

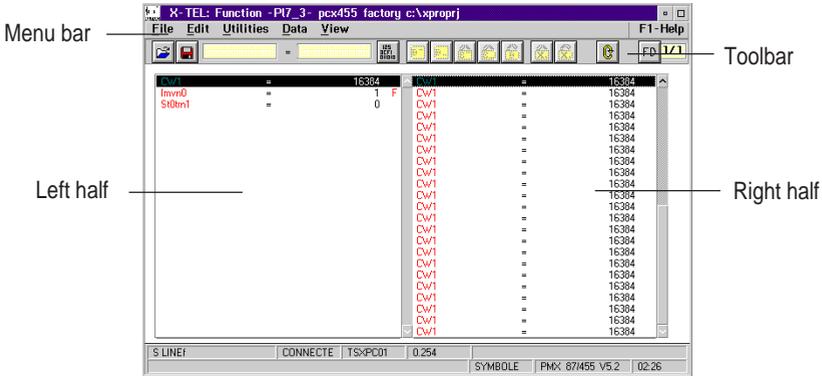
(3) bits, words, double words, word tables and double word tables only,

(4) bits, words, doubles words, bit tables, word tables and double word tables only.

10.2 Description of the screen and keys

10.2-1 Description of the screen

The screen is divided into two halves, each of which has a specific role :



Left half (LH)

This half is used to display objects or lists of objects of different natures or of the same nature but not in consecutive order.

The objects displayed in this half are saved when the mode is quit.

In "refresh" mode the list of objects in this half is displayed in realtime.

Right half (RH)

This half is used to display lists of objects of the same nature in consecutive order.

The objects displayed in this half are not saved when the mode is quit.

In "refresh" mode, only the highlighted object (in reverse video) is displayed with its previous states displayed on the preceding lines.

10.2-2 Menu bar**FILE MENU :**

Open	retrieves a data list previously saved in a file.
Save As	saves the list of objects displayed in the left half in a file.
Quit	quits DATA mode

EDIT MENU :

Refresh	displays all the objects in the left half in realtime.
Insert	is used to insert an object.
Delete	deletes the selected object.
Next	displays in the right half the objects following the object selected in the left half (<ESC> interrupts the scrolling).
Log	displays the changes in the value of the object (<ESC> interrupts the scrolling).
Search	in the right half, launches the search for forced or blocked objects of the same nature as the selected object.

DATA MENU :

Object	accesses modification of the selected object.
Value	is used to specify the value of the object.
Base	is used to specify the base in which the object is displayed.
Incremented Index	is used to select the index or address from which a list will be established or a search made in successive increments.

VIEW MENU :

Variables	immediate change to display variables (addresses).
Symbols	immediate change to display symbols.

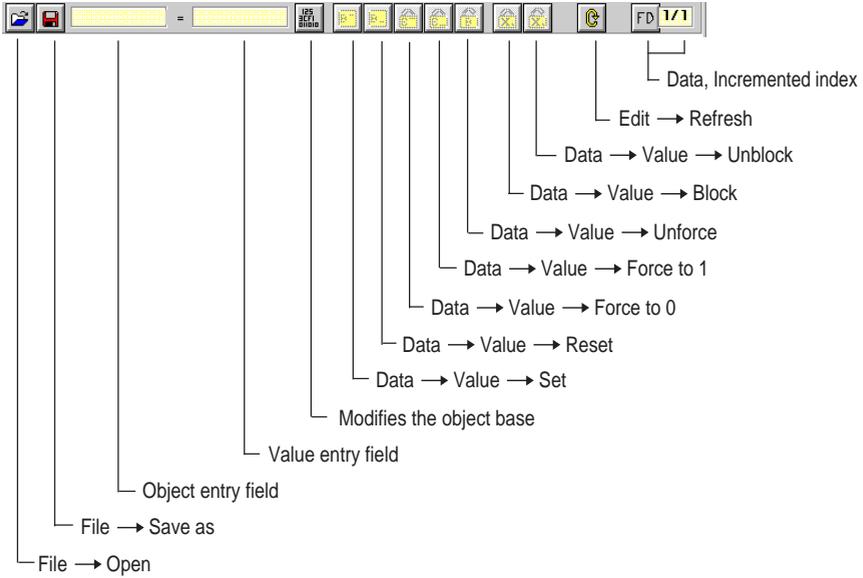
UTILITIES MENU :

Edit Symbols	launches the XTEL-SDBASE tool.
---------------------	--------------------------------

10.2-3 Toolbar

This toolbar provides quicker access to the main functions of the menu bar in DATA mode.

Definition of toolbar buttons :



F

10.2-4 Role of keys

<D>, <H>, , <L> and <M> are the keys used to select the base in which an object is displayed. The <F> key is also accessible on PMX stations, for display in "floating point" base.

Role of common keys

File, Open	<CTRL>+<O>	Is used to retrieve a list of data previously saved in a file.
File, Save As	<CTRL>+<S>	is used to save the list of objects displayed in the left half in a file.
File, Close		exits DATA mode. Depending on the context, cancels an operation in progress.

Edit, Refresh	<R>	displays in realtime all the elements of a list of objects created in the left half.
Edit, Insert	<INSERT>	accesses the entry band in order to insert an object.
Edit, Delete	<Delete>	deletes the selected object.
Edit, Next	<CTRL>+<Pg DOWN>	lists the object highlighted by the cursor together with the following objects in the right half. This list depends on the incremented index.
Edit, Log	<CTRL>+<Pg UP>	is used to display the change in the value of an object.
Edit, Search	<S>	starting from the object highlighted by the cursor (in reverse video), searches for forced or blocked objects of the same type as the selected object.

Data, Value→Set	<F2>	accesses the entry band, at the object highlighted by the cursor, (and to its value), in order to modify it.
Data, Object	<F8>	accesses the entry band, at the object highlighted by the cursor, in order to modify it.
	<CTRL> <F8>	replaces the selected object with the next object, eg : W1 by W2.
	<SHIFT> <F8>	replaces the selected object with the previous object, eg : W2 by W1.

Data, Base	<F9>	specifies the base in which the word or double word object is displayed.
Data, Incremented Index	<F7>	is used to select the index or address from which a list will be established or a search made in successive increments.
	<ESC>	is used to interrupt a search or log.
	<CTRL> + <DELETE>	deletes the contents of the selected half.

Role of keys specific to each half

Left half (LH)

<CTRL> + <←→>	accesses the RH, and copies into it the object highlighted by the cursor.
<↑>	moves the cursor up.
<↓>	moves the cursor down.
<←→>	accesses the RH.
<F8>	displays all the objects of the LH in realtime.
<CTRL> <PG UP>	displays the log of the highlighted object in the RH.
<CTRL> <PG DOWN>	displays the objects following the object selected by the cursor in the RH.
<CTRL>+ <INS>	inserts the next object after the selected object eg : B12 after B11.
<SHIFT>+ <INS>	inserts the previous object after the selected object eg : B11 after B12.

Right half (RH)

<CTRL> + <←→> or <ENTER>	accesses the LH, and copies into it the object highlighted by the cursor.
<↑>	displays the previous object (previous address).
<↓>	displays the next object (next address).
<←→>	accesses the LH

F

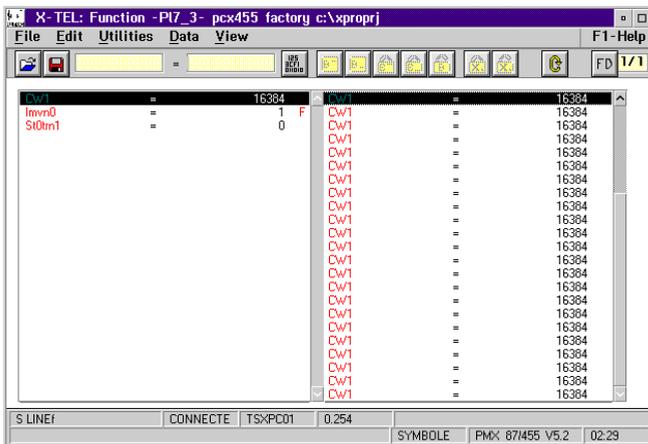
Realtime display of a list of objects

Edit, Refresh	<R>	displays in realtime all the objects in a list of objects created in the left half. No entries are possible during the execution of this function.
	<ESC>	cancels the refresh operation and memorizes all the values of the listed objects, except that of the object shown in reverse video which remains displayed in realtime.

10.3-2 Reading a list of consecutive objects (RH)

Having read the "original" object in the list, following the procedure described below :

Edit, Next	<CTRL>+<PG DOWN>	displays the object entered, together with all following objects, in the RH. Only the object shown in reverse video is displayed in realtime.
	<ESC>	stops the insertion in the RH.



It is possible to select the base in which the values of word objects are displayed either by selecting the **Base** option from the **Data** menu, or by pressing **<F9>**. With a PMX station, this key is also used to display double word objects in floating point format.

10.3-3 Data log

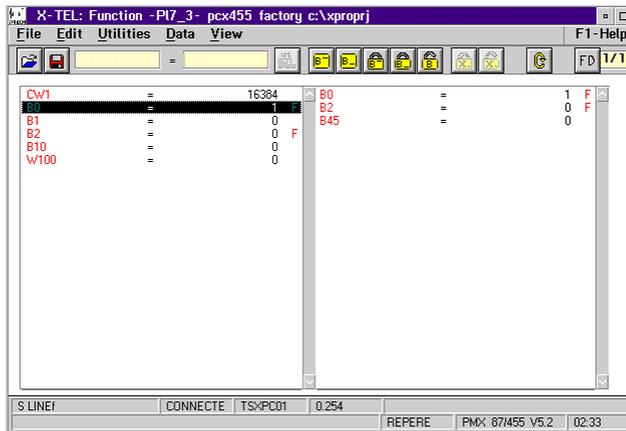
This function is used to display selected data in realtime in all the LH.

Edit, Log	<CTRL>+ <PG UP>	launches the execution of the "log" function. The value of the data is refreshed on each scan and listed on 19 lines in the right half of the screen.
	<ESC>	cancels the refresh of the object on the current line.

10.3-4 Searching for forced or blocked objects

Enter the object which is the starting point of the search, or position the cursor on it if it is already displayed.

Edit, Search	<S>	launches the search in the RH. The search stops as soon as all the forced objects (indicated by F) or blocked objects (indicated by B) have been found. All the objects found are displayed on the screen, so that a complete list of forced and blocked objects can be established.
-------------------------	------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<ESC> stops a search in progress.

For objects with multiple indexes, the index on which the search is to be performed is selected using the soft key **<F7>** (see section 10.3-5).

10.3-5 Selection of the index or address

Data, Incremented Index is used to select the index or address from which a list will be established (using the **Next** option from the **Edit** menu) or a search made in successive increments.

The index window in the top part of the screen has two numbers :

- the number on the right is fixed by the terminal and depends on the number of indexes of the highlighted object.

FD1/1 object with one index. Example : W4 or SY20.

FD1/2 object with two indexes. Example : W4,1.

FD1/3 object with three indexes. Example COM0,1,3 or iW14,1,2.

FD1/4 object with four indexes. Example ROW4,3,2,1

- the number on the left increases with successive presses of F7 and shows the incremented index (1 corresponds to the index furthest to the right and the highest number furthest to the left).

Examples

FD1/1	B1	B2	B3
FD1/2	W15,1	W15,2	W15,3
FD2/2	W15,1	W16,1	W17,1
FD2/3	COM1,0,0	COM1,1,0	COM1,2,0
FD3/3	COM1,0,0	COM2,0,0	COM3,0,0
FD3/4	ROW4,3,2,1	ROW4,4,2,1	ROW4,5,2,1
FD4/4	ROW4,3,2,1	ROW5,3,2,1	ROW6,3,2,1

Note

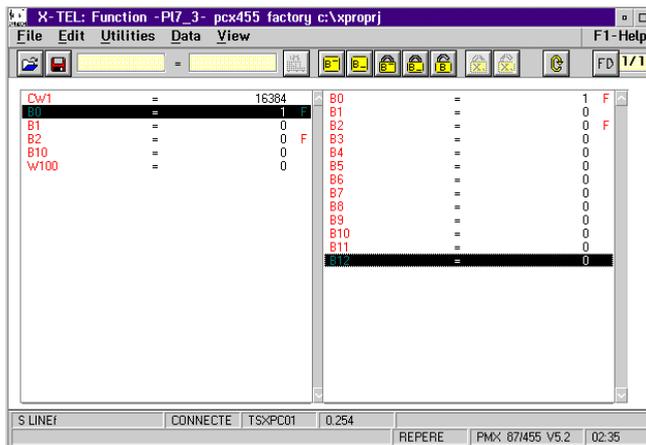
For discrete I/O bit objects, the list of output bit objects can follow the input bit objects if there is an input module in the next slot, and vice versa.

10.4 Writing and forcing data

10.4-1 Writing and forcing a bit

To modify the state of a bit object, it must first be selected or entered, then use the **Value** → option from the **Data** menu

SET	<F2>	sets the bit highlighted by the cursor to 1.
RESET	<F3>	sets the bit highlighted by the cursor to 0.
FORCE TO 1	<F4>	forces the bit highlighted by the cursor to 1.
FORCE TO 0	<F5>	forces the bit highlighted by the cursor to 0.
UNFORCE	<F6>	cancels the forcing of the bit highlighted by the cursor.



A forced bit is indicated by the letter F.

Important

The forcing of all bits is cancelled by a cold restart (SY0 = 1).

10.4-2 Writing a list of consecutive bits

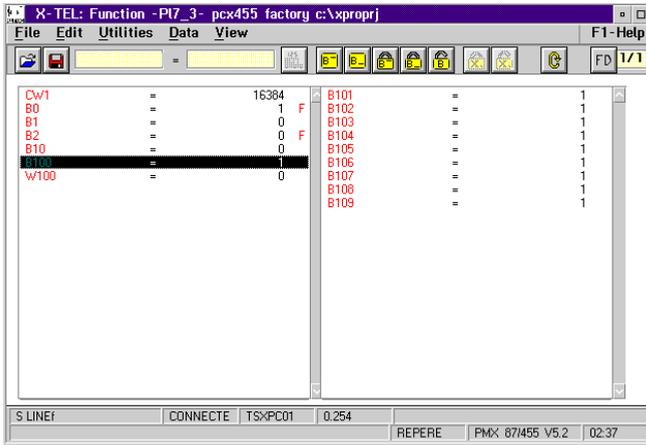
A list of consecutive bits can be set to 1 or 0. To do this the entry of the start object must be followed by the length of the list and then 1 or 0.

Example : Set bits B100 to B109 to 1

Edit, Insert or **<INSERT>** can be used to enter the start object (B100) in the object entry field : B100.

Then **<TAB>** is used to enter the value (1) and the length of the list (10) in the value entry field : 1-10.

<ENTER> confirms the entry. The list of bits is then displayed in the right half of the screen.



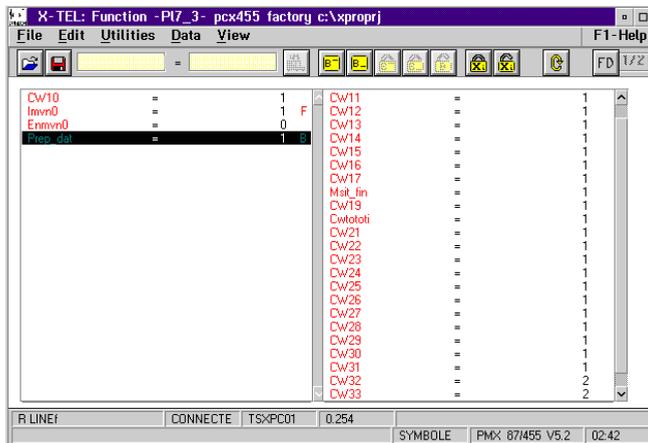
Comments on I/O bits

- the forcing of an output bit to 1 or 0 while the program is running activates or deactivates the corresponding output.
- the setting of output bits to 1 or 0 has no effect on the corresponding outputs, if the application program writes them.
- as long as the program does not write these bits, the setting of these bits to 1 or 0 by the terminal is taken into account :
 - by the user program when it reads these bits,
 - when the outputs are updated.

10.4-3 Modifying and blocking a step

When a step bit is highlighted by the cursor :

Data, Value→ Set	<F2>	activates the step. Activation of this step : <ul style="list-style-type: none"> • does not deactivate the upstream steps, • enables the transition(s) downstream of the step, • executes actions on activation and continuous actions
Data, Value→ Reset	<F3>	deactivates the step and does not execute actions on deactivation.
Data, Value→ Block	<F4>	blocks the step; ie. the transition(s) downstream of the step are disabled. The Grafcet chart evolves up to the blocked step, then stops on this step whatever the state of the transition condition. Several steps can be blocked simultaneously.
Data, Value→ Unblock	<F5>	unblocks the step; ie. the transition(s) downstream are enabled.



- the blocking of a step in a chart has no effect on the evolution of connected charts.
- a blocked step is indicated by the letter B.
- it is possible to activate a step by entering it directly as equal to 1; for example X1=1.
- it is also possible to activate or deactivate several steps simultaneously; for example X3-5=1 : all the steps from 3 to 7 are activated.

10.4-4 Writing words

The contents of a word can be modified directly by entering its value in the value field.

Data, Value	<F2>	is used to enter the value of the object.
--------------------	-------------------	-------------------------------------------

In the case of a bit, **<CTRL> + <F2>** are used to enter the value of the object.

10.4-5 Writing a list of consecutive words

The values of a list of consecutive words can be modified as follows :

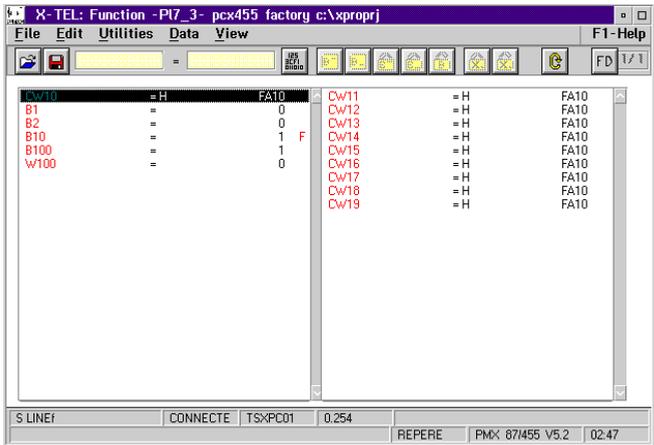
Example : writing the constant words CW10 to CW19 with the hexadecimal value H'FA10'.

Edit, Insert or **<INS>** is used to enter the start object (CW10) in the object field : CW10

Data, Base → **Hexadecimal** or **<F9>** selects the hexadecimal base and authorizes entry of the value : FA10.

Data, Value or **<F2>** are used to enter the value (FA10) and the length of the list (10) in the value entry field : FA10/10.

<ENTER> confirms the entry.



Comments on I/O register words

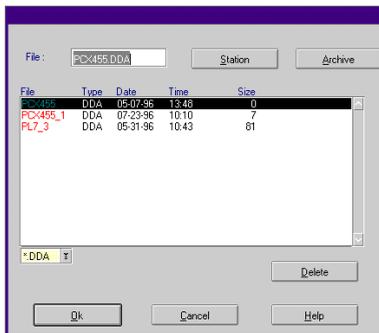
- the modification of output register words has no effect on the corresponding module if the user program also writes these words.
- as long as the program does not write these words, their modification by the terminal is taken into account :
 - by the user program when it reads these words,
 - when the register words are updated.

Note

The preset words associated with function blocks can only be written if the parameter modification option has been selected during configuration.

10.4-6 Saving and restoring a predefined data screen

File, Save As	<CTRL>+<S>	is used to save the list of objects displayed in the LH in a file.
--------------------------	-------------------------------	--------------------------------------------------------------------



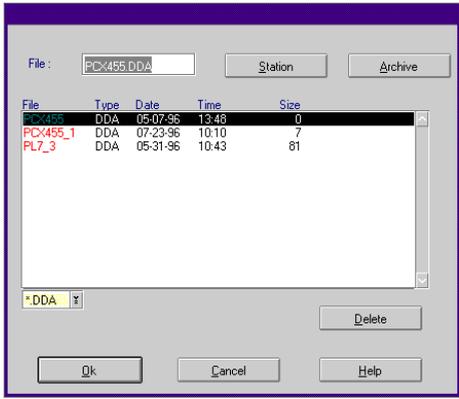
The entry field is used to define the name of the file.

The list is used to display existing files. When saving, the new file will overwrite the old file with the same name (after confirmation).

The inset at the bottom of the list is used to select the type of file to be displayed in the list.

- Station** assigns the name of the station as the file name.
- Archive** assigns the global archive name (PL7_3) as the file name.
- Delete** deletes the selected file after confirmation
- OK** confirms the selection made
- Cancel** exits without confirming the selection.
- Help** calls up the program help function.

File, Open	<CTRL>+<O>	is used to retrieve a list of data previously saved in a file.
-------------------	-------------------------------	----------------------------------------------------------------



The entry field is used to define the name of the file to be retrieved.

The list is used to display the existing files and to select the file to be retrieved.

The inset at the bottom of the list is used to select the type of file to be displayed in the list.

- Station** assigns the name of the station as the file name.
- Archive** assigns the global archive name (PL7_3) as the file name.
- Delete** deletes the selected file after confirmation
- OK** confirms the selection made
- Cancel** exits without confirming the selection.
- Help** calls up the program help function.

F

Section	Page
11.1 Changes made since V52	11/2
11.2 Presentation of DEBUG mode	11/3
11.2-1 Role	11/3
11.2-2 Diagrams	11/4
11.2-3 Presentation of the screen	11/10
11.3 Predefining screens	11/11
11.3-1 Role	11/11
11.3-2 Predefining a program screen	11/12
11.3-3 Predefining a data screen	11/13
11.3-4 Predefining a vignette	11/14
11.3-5 Saving and retrieving a predefined screen	11/15
11.4 Displaying predefined screens in realtime	11/17
11.4-1 Presentation	11/17
11.4-2 Displaying a LITERAL program screen	11/18
11.4-3 Displaying a rung	11/18
11.4-4 Displaying a Grafcet screen	11/19
11.4-5 Displaying a DATA screen	11/20
11.4-6 Displaying a vignette	11/21
11.5 Defining a breakpoint	11/22
11.5-1 Inserting and deleting a breakpoint	11/22
11.6 Executing the program	11/24
11.6-1 Execution	11/24
11.6-2 Influence of the master task on the state of outputs	11/28
11.7 Accessing other modes	11/29
11.7-1 DATA LINE mode	11/29
11.7-2 SEARCH/REPLACE mode	11/30
11.7-3 DATA mode	11/30
11.8 Searching for the program addresses of a variable	11/31

11.1 Changes made since V52

- This mode is now directly accessible from the language editors.
- Predefined screens are defined at PROGRAM BROWSER level, which can be accessed by pressing **<ALT>+<F1>** :
 - **Add** button of the Program Address Selection page for program screens,
 - **Add** button of the Module Selection page for vignettes,
 - **Data** button of the Debug page for data screens.
- The six screens are accessed via the **Predefined Screens** tab of the PROGRAM BROWSER, by selecting the screens from the list with the movement arrows or the mouse.
- The user can browse between screens using the **<ALT>+<1>** to **<ALT>+<6>** keys. No screen is predefined by default.
- To debug a Literal program :
 - an inserted breakpoint is displayed in a green zone in the editor,
 - in step by step, the next statement to be executed is displayed in a yellow zone.

11.2 Presentation of DEBUG mode

11.2-1 Role

After the program has been entered on the terminal, in program mode, the execution of the program must be monitored. The DEBUG mode offers a set of functions which enables the user to execute, display and modify the program.

This mode, accessible in online mode, is used to :

start or stop execution :

- of the program,
- of one or more tasks in the program,

display and animate in realtime :

- up to 6 predefined screens each of which may contain :
 - a rung,
 - a Literal statement,
 - a Grafcet page,
 - a list of data,
 - a vignette,
- the program elements or Grafcet pages upstream or downstream of the predefined screens,

modify (without leaving DEBUG mode) :

- a program,
- the function block parameters,

insert a breakpoint in a task and execute the task :

- scan by scan,
- rung by rung,
- statement by statement,

access the DATA and DATA LINE modes, which authorize :

- reading and writing of bit or word objects,
- forcing of bit objects,
- blocking of Grafcet advancement,
- forcing the state of I/O bits,

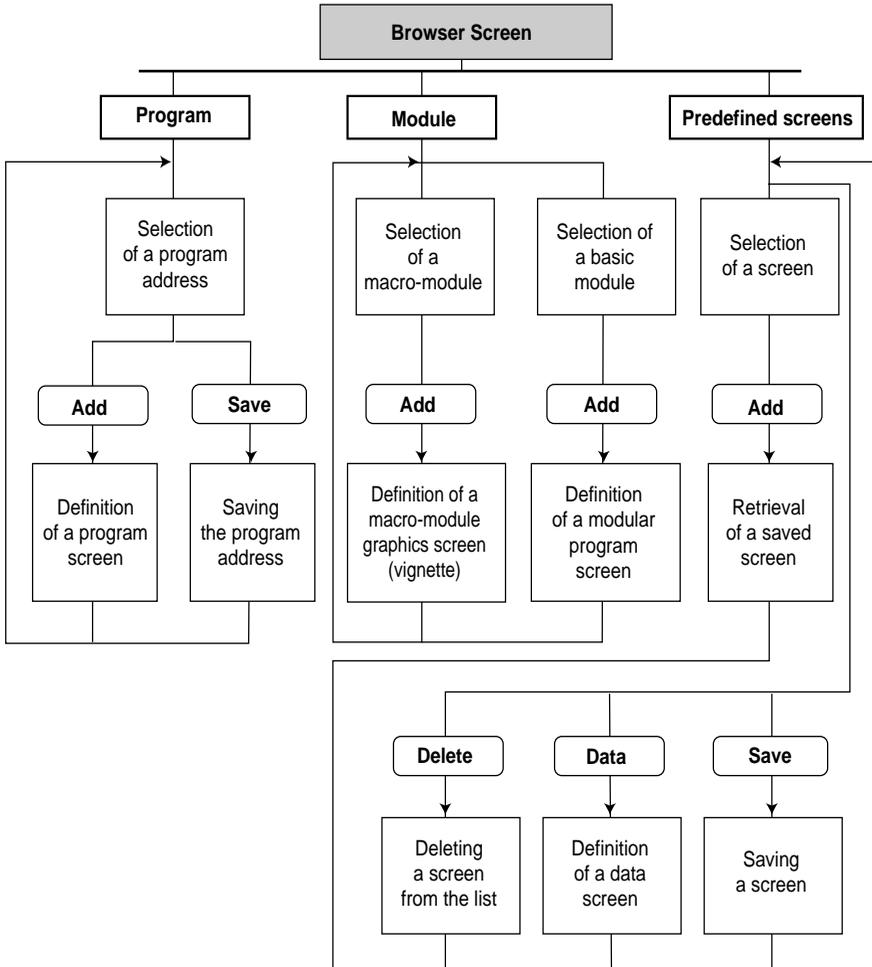
access SEARCH/REPLACE mode in order to :

- search for an object in all or part of the program,
- replace objects with compatible objects in all or part of the program,

freeze the realtime animation of a screen

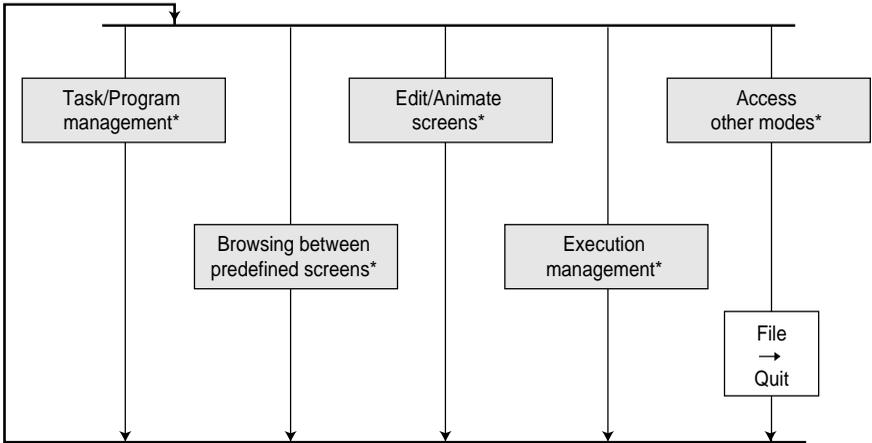
11.2-2 Diagrams

Definition of predefined screens



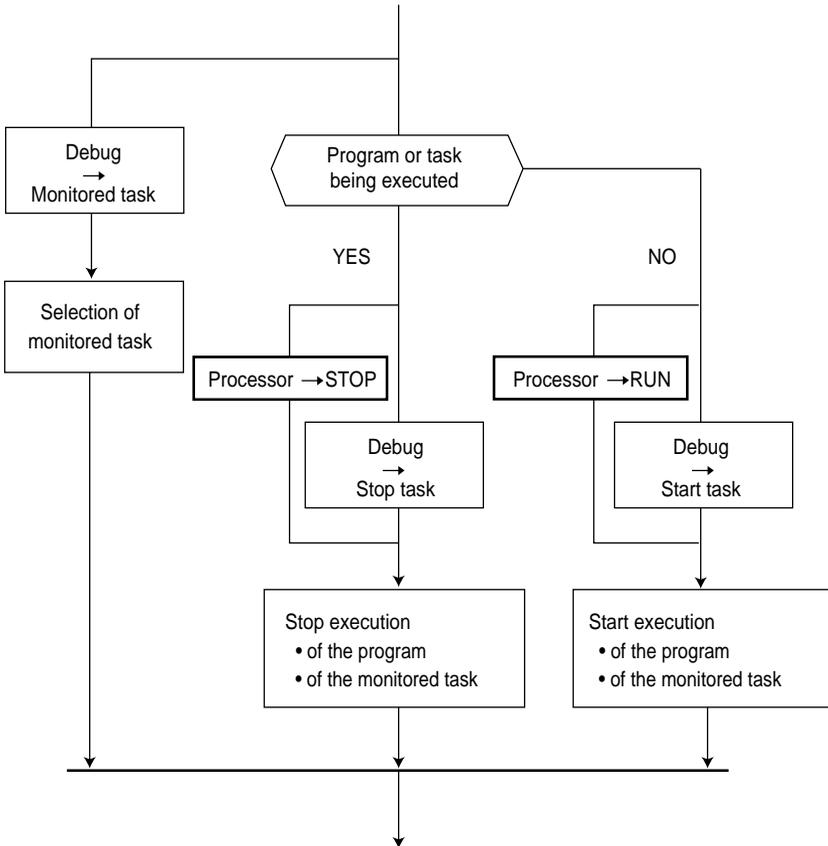
* See Edit/Animate screens diagram

Debug functions



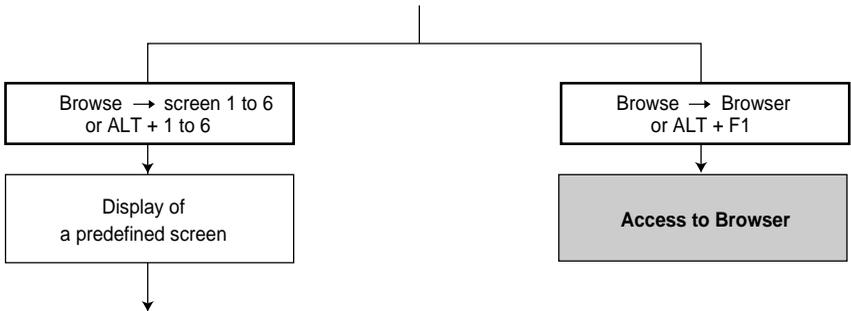
* See the next five diagrams

Task/program management

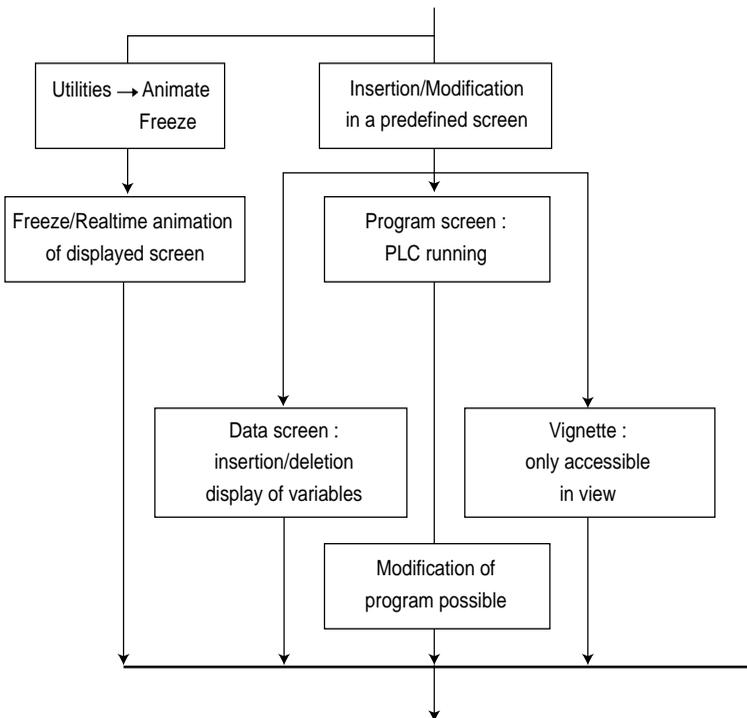


F

Browsing between screens

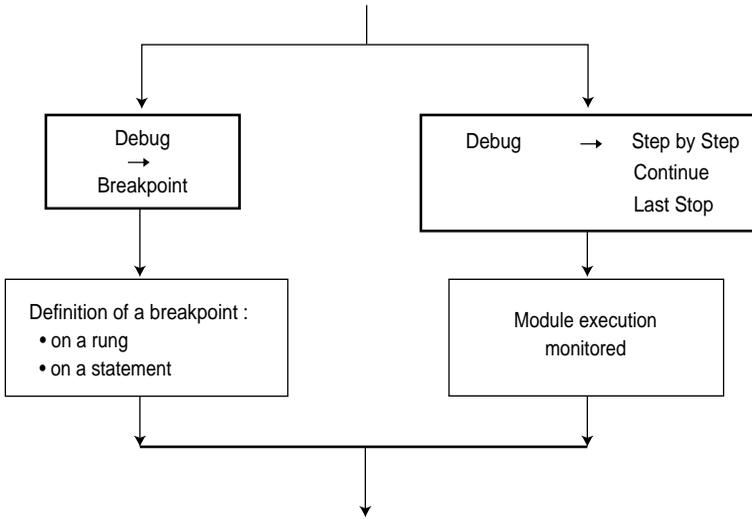


Edit/Animate screens

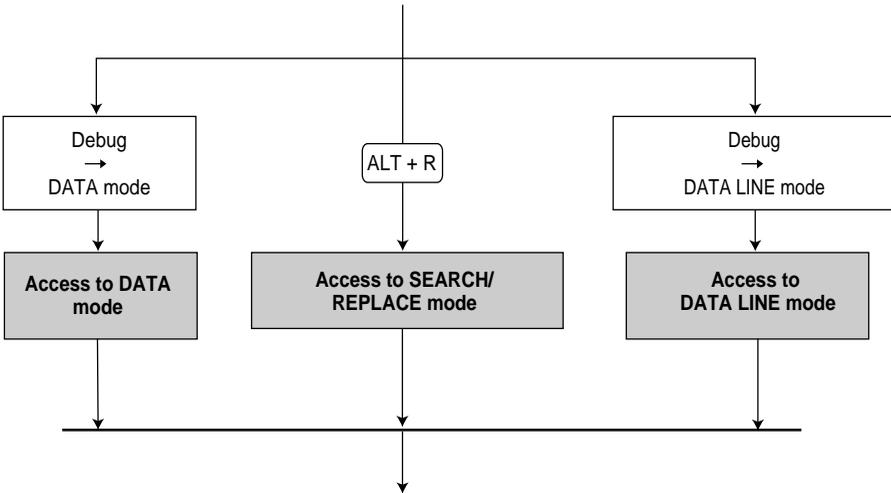


F

Managing the execution of a program

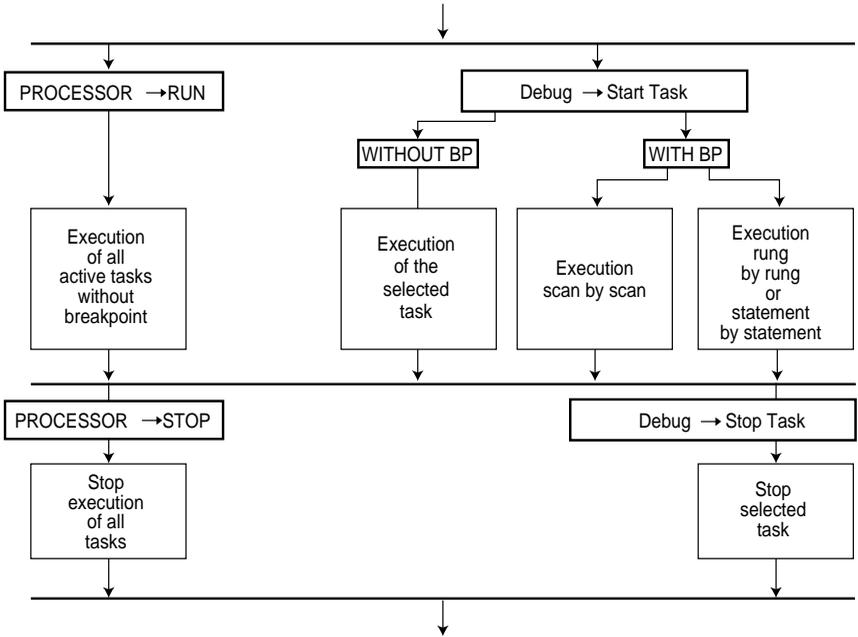


Access to other modes



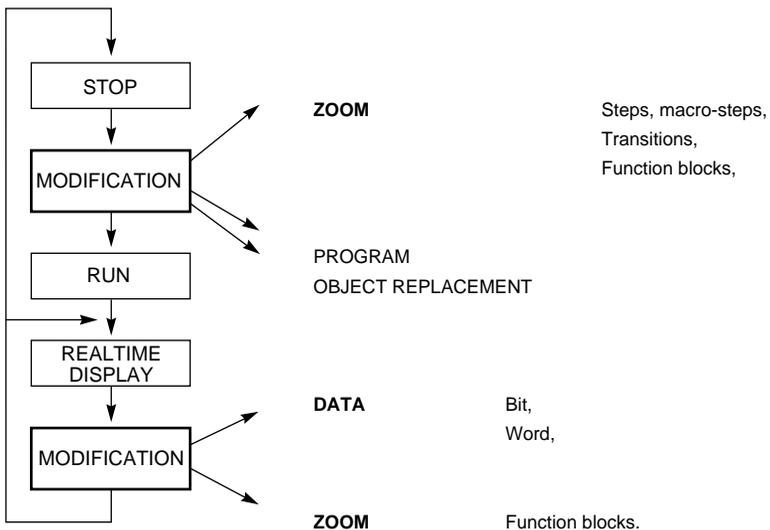
F

Principle of program execution



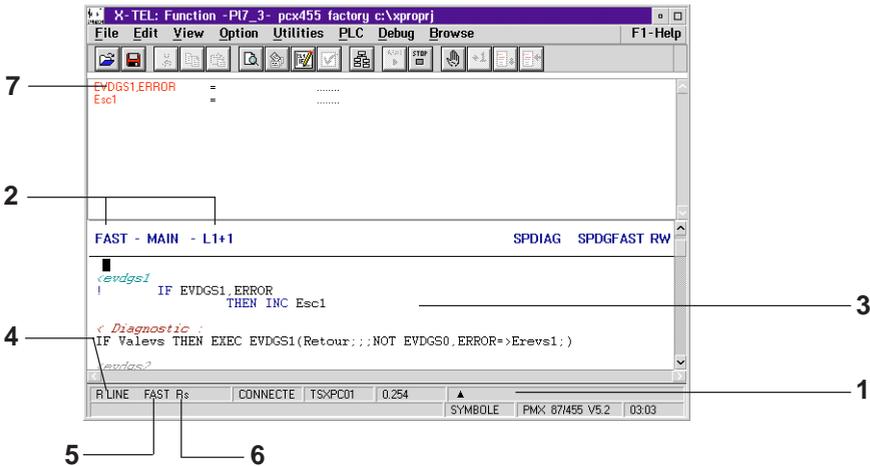
Principle of program modification

The debug scan time is very short.



11.2-3 Presentation of the screen

The information displayed in debug mode is as follows :



- 1 realtime screen update indicator.
- 2 program address of the element displayed :
 - task : MAST, FAST, AUX,
 - module : MAIN, SR, CHART, XM, PRL, POST,
 - program element :
 - Lx if the element is a rung or a statement,
 - page x if the element is a Grafcet page,
 - movement in relation to the label.
- 3 zone reserved for the element displayed in realtime.
- 4 type of logical connection with the PLC :
 - LINE : valid connection,
 - S before LINE indicates that the PLC is stopped,
 - R before LINE indicates that at least one task is being executed
 - f after LINE indicates that at least one bit is forced.
- 5 BP indicates the presence of a breakpoint in a configured task and /■ a blocked step.
- 6 status zone indicating the task selected on starting the mode, and its status :
 - R task running (RUN),
 - S task not executed (STOP),
 - B task whose execution is blocked at a breakpoint,
 - s task not active, not activated by the control block.
 - a task active, activated by the control block (CTRLi).
- 7 realtime display of objects included in the statement referenced by the line pointer (in ANIMATION mode only).

11.3 Predefining screens

11.3-1 Role

This function is used to predefine the contents of 6 screens. Each screen may contain :

- a program element or a Grafcet page for a program screen,
- a list of addressable objects of any type for a data screen,
- a vignette.

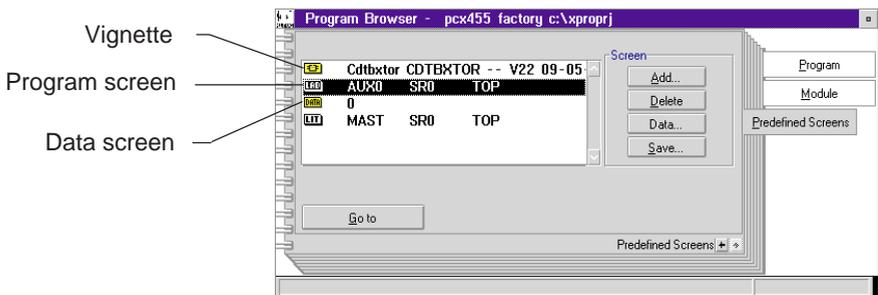
This memorization is effective as long as the terminal is operating online or during an online → offline → online sequence, if no transfer is necessary.

Screens are defined at Browser level :

- **Program** tab for program screens (Add button),
- **Module** tab for vignettes (Add button),
- **Predefined Screens** tab for data screens (Data button).

Managing screens

Unlike previous versions of PL7-3, the management of DEBUG screens is performed directly from the PROGRAM BROWSER by selecting the **Predefined Screens** tab :



The following functions can be applied to a screen selected from the list :

- Go to** displays the predefined screen (see 11.3).
- Delete** deletes the screen.
- Save** saves the screen in a file.

If the list has less than six elements :

- Add** retrieves a screen from a file and puts it in the list.

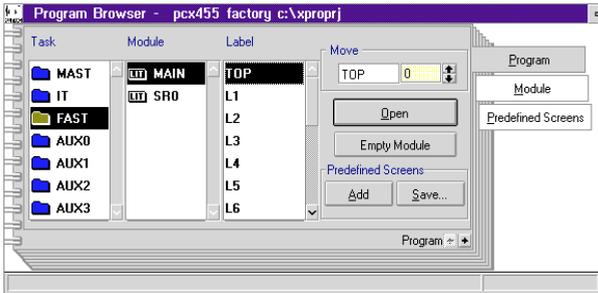
Note

Screens defined from the Program and Module tabs are automatically added to the list, as are Data screens. No screen is predefined by default.

11.3-2 Predefining a program screen

Remember that in V6, the DEBUG mode is integrated in the PROGRAM mode. Consequently, the screen predefinition function is accessible from the Browser : the **Program** tab of the Browser gives access to this function.

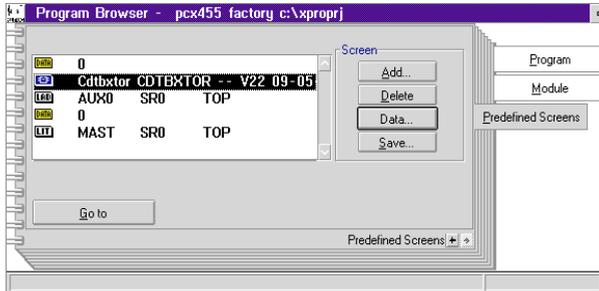
The address of the element which the user wishes to be memorized is defined by the task, the module and the label or page if the element is a Grafcet page.



- Add** inserts the new screen into the list of predefined screens if the list has less than 6 elements. If not, the action is ignored.
- Save** is used to save the program address (see section 11.3-5).

11.3-3 Predefining a data screen

This function is accessible from the **Predefined Screens** tab of the Browser and enables the user to define a list of bit or word objects, the status or contents of which will be displayed in realtime.

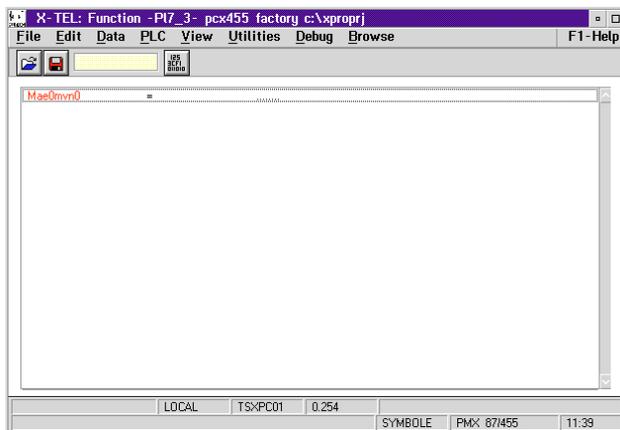


All types of objects can be mixed in the same list and the entry order is immaterial whatever the type of object. The list only concerns configured objects.

See DATA mode (section 10) for further details on the list of accessible objects, the base in which the word object values are displayed, etc.

Data inserts a data screen into the list of predefined screens. As this new entity is empty for the moment, it is called **DATA 0**.

Go to selecting a data screen (such as **DATA 0**) from the list gives access to data editing :



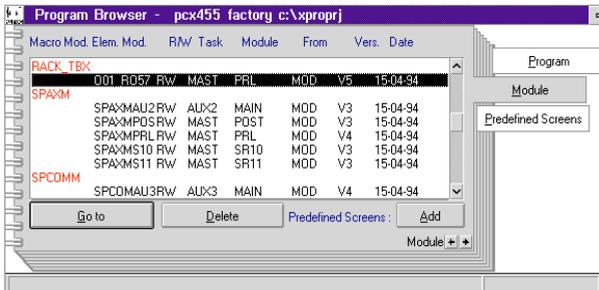
This screen is a simplified version of DATA mode. The edit functions are the same as for data mode :

When the cursor is positioned on an object,

	<↑><↓>	move the cursor within the column.
Edit, Delete	<DELETE>	deletes the object indicated by the cursor and shifts the following objects up.
Edit, Insert	<INSERT>	inserts an object at the point indicated by the cursor and shifts the following objects down.
File, Close	<CTRL> <F4>	returns to the previous screen. The data screen is now called DATA followed by the number of objects it contains, as well as the name of the first object (for example : DATA 4 W12...).

11.3-4 Predefining a vignette

This function, accessible from the **Module** tab of the Browser, enables the user to define a screen which represents a macro-module graphically.



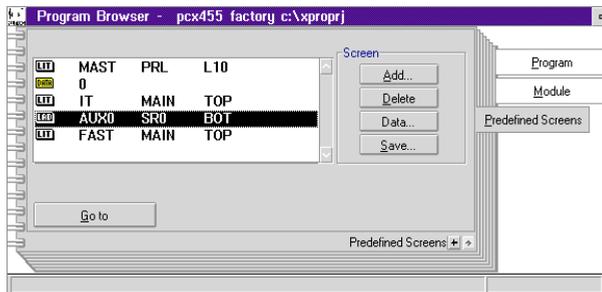
Add

inserts the new screen into the list of predefined screens if the list has less than 6 elements. If not, the action is ignored.

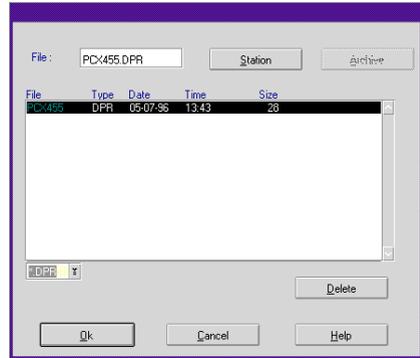
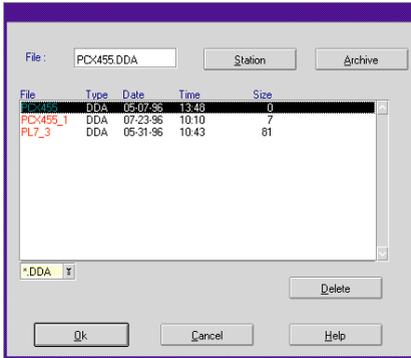
- Go to** is used to access directly the start of a basic module code or the graphic view of a macro-module.
- Delete** is used to delete the code of an unprotected basic module or a macro-module.

11.3-5 Saving and retrieving a predefined screen

This function is used to save a program address or a list of data in a file, then to retrieve it later.



- Save** is used to save the following in a file :
- for a program screen : the program address,
 - for a data screen : the list of objects displayed on the screen.
- Add** is used to retrieve the contents of a file previously saved :
- program address (xxx.DPR file),
 - list of objects to be displayed on the screen (xxx.DDA file).



The entry field is used to define the name of the file (to be saved or retrieved).

Station assigns the name of the station as the name of the file (to be saved or retrieved).

Archive assigns the global archive name (PL7_3) as the name of the file (to be saved or retrieved).

F

11.4 Displaying predefined screens in realtime

11.4-1 Presentation

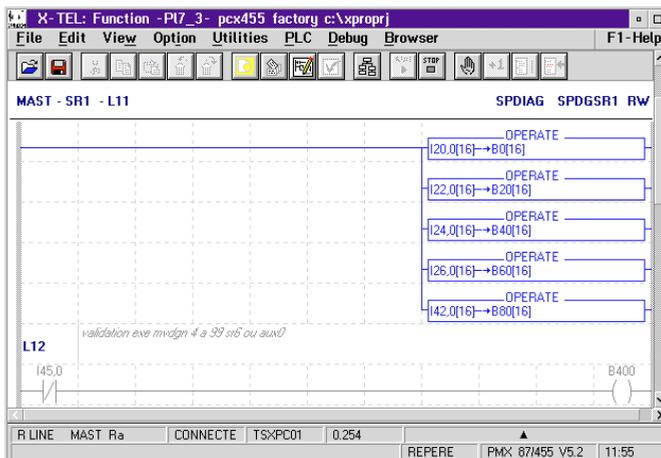
Screens can be displayed by pressing **Go to**, from the **Predefined Screens** tab of the Browser, if a screen has been selected from the list of predefined screens.

It is used to display, by realtime animation, the graphic elements of Ladder or Grafcet language. In Literal language, all the objects which make up a statement are displayed with their status and their contents.

The statuses or contents of the objects are taken into account at the start of the task, during the exchange with the terminal. The updating of the screen depends on the language used :

- for a program element, the “photo” is taken when the rung or statement has been executed,
- for a Grafcet page or a data screen, the “photo” is taken at the start of the master task scan.

The update indicator is displayed at the bottom right of the screen. The movement of the triangle corresponds to the updating of the screen if the task is executed and not frozen.



In Animation mode only, the STATEMENT/RUNG which is animated is in color (or heavily shaded for black and white screens) whereas the others are gray.

11.4-2 Displaying a LITERAL program screen

Menu bar and toolbar

Refer to the section on LITERAL language.

The status of bit objects and the contents of word objects included in the selected statement are displayed in realtime in the upper part of the screen.

11.4-3 Displaying a rung

Menu bar and toolbar

Refer to the section on LADDER.

Displaying a rung

Realtime display is achieved, according to the elements, as follows :

- **contacts**

When the contact is closed, the graphic element appears in a different color (in reverse video).

- **coils**

When the associated bit is at state 1, the graphic element appears in a different color (in reverse video).

The JUMP and CALL coils are not animated in realtime.

- **function blocks**

When the function block output is at state 1, it appears in a different color (in reverse video), and progressively changing values are displayed. The vertical and horizontal comparison block outputs do not change color.

- **horizontal comparison or operation blocks**

Realtime display of the contents of these blocks can be accessed using the ZOOM function.

11.4-4 Displaying a Grafcet screen

Menu bar and toolbar

Refer to the section on GRAFCET.

In realtime display the graphic symbols for steps are :

-  active step.
-  active initial step.
-  active macro-step. At least one step other than the OUT step is active.
-  active macro-step. Only the OUT step is active.

Note

The Grafcet pages are updated at the start of the master task, during the exchange with the terminal. The update indicator is displayed whether the task is running or stopped.

11.4-5 Displaying a DATA screen

Unlike the DATA operating mode, which has a realtime variables display column and a log column, a data screen has only a realtime display column.

Menu bar

FILE MENU :

Open	displays a dialog box for selecting the file to be retrieved.
Save	is used to archive the list of objects in a file.
Close	is used to return to PROGRAM BROWSER.

EDIT MENU :

Insert	inserts an object in the data list.
Delete	removes an object from the data list.

DATA MENU :

Object	is used to change the value of the selected object.
Base	changes the base of the selected object.

PROCESSOR MENU :

Run	starts the processor.
Stop	stops execution on the processor.

VIEW MENU :

Variables	immediate change to display VARIABLES (addresses).
Symbols	immediate change to display SYMBOLS.

UTILITIES MENU :

Animate	launches animation of the screen displayed.
Freeze	freezes animation of the screen displayed.
Edit Symbols	launches the XTEL-SDBASE tool.

DEBUG MENU :

Start Task	launches or stops execution of the task.
Task Monitored	is used to select the task to be executed and monitored.
Breakpoints → Insert → Remove → View	places the breakpoint on the program element displayed in the predefined screen. deletes the breakpoint. displays the place where the breakpoint has been defined.
Continue	executes a task scan.
Step by Step	executes the next element.
Last Stop	displays the last element executed.
Data Mode	accesses DATA mode.
Data Line Mode	accesses DATA LINE mode.

BROWSER MENU :

Data Window	accesses the DATA LINE window.
Browser	displays the Program Browser window.
Screen 1...6	switches from one predefined screen to another.

11.4-6 Displaying a vignette

See section 15-Appendix, V5 modular version.

11.5 Defining a breakpoint

11.5-1 Inserting and deleting a breakpoint

A breakpoint can be defined on a program element for all the tasks configured. This program element can be part of one of the following modules :

- the main program MAIN,
- a subroutine SR,
- pre-processing PRL or post-processing POST sections of the master task.

Defining a breakpoint

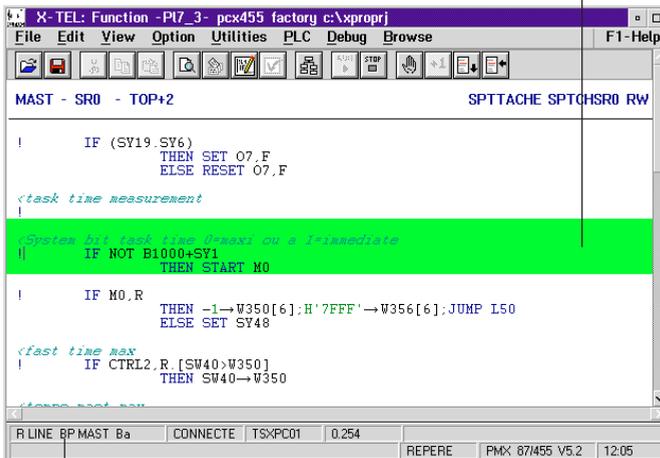
From one of the predefined screens,

Debug, Breakpoint → Insert		accesses definition of the breakpoint and places the breakpoint on the program element selected in the predefined screen.
-------------------------------------------	--	---------------------------------------------------------------------------------------------------------------------------

If a breakpoint has already been placed in the program, it will be deleted. The terminal then displays the message “PREVIOUS BP DELETED” or “BP = CURRENT SCREEN (OLD BP DELETED)”.

Redefinition of the breakpoint will only take effect after the task which contained the old breakpoint has been executed (RUN TK key).

Display of the breakpoint



Indication of the presence of a breakpoint

If a breakpoint has been defined, “BP” is displayed in the status zone :

- **in green** if the breakpoint is defined on a task that is stopped, or if the element of the breakpoint is not executed (non-called SR effective, element JUMP, etc). In this case, the breakpoint is defined but not effective.
- **in yellow** if the breakpoint is defined on an element that is executed. In this case the breakpoint is effective. Execution of the task is blocked on the element of the breakpoint (timers and monostables continue).

Blocking a task does not effect execution of the other tasks.

Displaying a breakpoint

<p>Debug, Breakpoint → View</p>		<p>displays the place where the breakpoint has been defined.</p>
---------------------------------------------------	--	------------------------------------------------------------------

Deleting a breakpoint

<p>Debug, Breakpoint → Remove</p>		<p>deletes the breakpoint. This request will only be taken into account after the task containing the breakpoint has been executed (RUN TK key). “BP” will no longer be displayed in the status zone.</p>
-----------------------------------------------------	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Note 1

Program modifications cannot be made if a breakpoint is effective (the error message MODIF PROHIBITED : EXISTING BP is displayed if a modification is requested).

Note 2

If the terminal is logically disconnected from a PLC which has a task with a breakpoint in it, when the terminal is reconnected to the PLC, the breakpoint will be deleted once the task concerned has been executed.



11.6 Executing the program

11.6-1 Execution

Periodic task statuses

The diagram below shows the different statuses that each task can take independently.

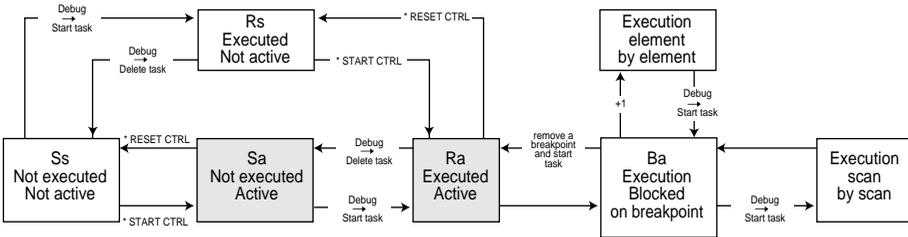
Since the master task is always active, statuses Ss and Rs only apply to fast and auxiliary tasks.

In the event of STOP or RESET CTRLi, the task stops its scan cycle (after the outputs are updated).

Only one breakpoint can be defined on all the tasks.

A task that is blocked on a breakpoint has no effect on the execution of other tasks in progress.

A breakpoint can be defined in statuses Sa and Rs, but it is only taken into account in status Ra.



* Command from the user program

Scanning a task

A task is scanned with the frequency whose period is :

- defined at configuration when the task is Rs or Ra,
- fixed at 50ms by the system for statuses Ss, Sa or Ba.

The exchange with the terminal consists of updating the information received from or sent to the terminal (RUN, STOP, Force, etc).

The updating of outputs concerns the modules declared in the task. All the modules are exchanged by default in the master task.

Common words are always exchanged in the master task.

When the task is not executed, not active or blocked on a breakpoint (statuses Ss, Sa, Rs and Ba) the I/O are not updated nor the program executed.

Task scan status : Ra	Task scan statuses : Ss, Sa, Rs and Ba
Period	Period
Exchange with the terminal	Exchange with the terminal
Updating of inputs COM-IW-I	
Execution of the user program	
Updating of outputs O-OW-COM	



Executing a program

Execution of an application program, corresponding to all the tasks configured, can be started and stopped either from the MODE SELECTION screen or directly from the DEBUG mode.

Processor, Run		starts execution of the program. The RUN indicator lamp on the PLC lights up.
Processor, Stop		stops execution of the program. The RUN indicator lamp on the PLC goes out.

The status zone indicates the status of the program.

Executing a task

When debugging an application, it is beneficial to execute each task independently.

Debug, Task Monitored is used to choose the task to be executed. A dynamic list is used to select the task.

Debug, Start Task starts execution of the selected task.

Debug, Stop Task stops execution of the selected task.

The screen's status zone shows the status of the selected task.

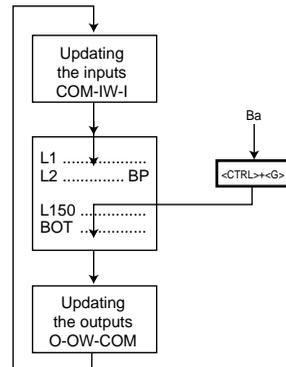
Execution scan by scan

After selecting a task, a breakpoint must be defined in order to execute it scan by scan (see section 11.4).

Debug, Continue	each activation of one of these keys executes a scan of the task. It is executed from the program element which follows the
<CTRL>+<G>	breakpoint up to and including the breakpoint program element.

Note

Exchanges with the terminal are maintained. Other tasks being executed continue their scans.



Execution rung by rung or statement by statement

This execution can be requested within a programming module MAIN, PRL, POST, SR. A breakpoint must be defined in this module and the task including this module must be selected.

Debug, Last stop		displays in realtime the last element executed in the module, which in this case is the breakpoint element.
Debug, Step by Step		each time this function is called the next element is executed, except if the breakpoint is defined on the last element of the module.

Notes

- the element which has just been executed contains a JUMP : if it is effective, the jump to the appropriate label is performed.
- the element which has just been executed contains an SR : if it is effective, the complete subroutine has just been executed,
- the element which has just been executed is the last element of a MAIN module :

Debug, Step by Step		starts the exchange of I/O associated with the task then the execution of the first element of the MAIN module.
--------------------------------	--	-----------------------------------------------------------------------------------------------------------------

- the element which has just been executed is the last element of a PRL, POST or SR module :

Debug, Step by Step		this function is no longer on offer.
Debug, Continue	<CTRL>+<G>	executes the rest of the task scan up to the breakpoint, if this is scanned.

11.6-2 Influence of the master task on the state of outputs

For safety reasons, the outputs of a PLC that is not executing its program normally are forced to 0 by default. This forcing of outputs concerns the following modules :

- discrete output,
- analog output,
- positioning,
- fast I/O coprocessor.

This forcing of the outputs to 0 can be inhibited by setting the system bit SY8 to 0 (via the program or the terminal).

In DEBUG mode, a PLC is considered to be not executing a program normally if the following conditions prevail :

- at least one task is not executed (status Ss : not executed and not active, or status Sa : not executed and active),
- a breakpoint is effective in a task (status Ba : execution blocked on a breakpoint).

Note

System bit SY8 is set to 1 by default and is used to reset all the outputs of the configuration, if the program is not executed normally. The complementary bits SY32 to SY39 (one per double rack or per pair of single racks) are used to reset outputs selectively. They are all set to 1 by default.

When SY8 = 1, only the outputs of racks whose associated system bits (SY32 to SY39) are at 1, are forced to 0 when the program is not executed normally.

When SY8 = 0, all the outputs retain their states when the program is not executed normally, irrespective of the states of SY32 to SY39.

Example : A TSX 67-40 configuration comprising one basic rack (0/1), one direct extension rack (2/3) and two remote extension racks 4 and 5.

The system bits SY8 (overall safety of outputs), SY32 (safety of outputs in rack 0/1), SY33 (safety of outputs in rack 2/3) and SY34 (safety of outputs in racks 4 and 5) are at the following states :

- SY8 = 1,
- SY32 = 1,
- SY33 = 0,
- SY34 = 0.

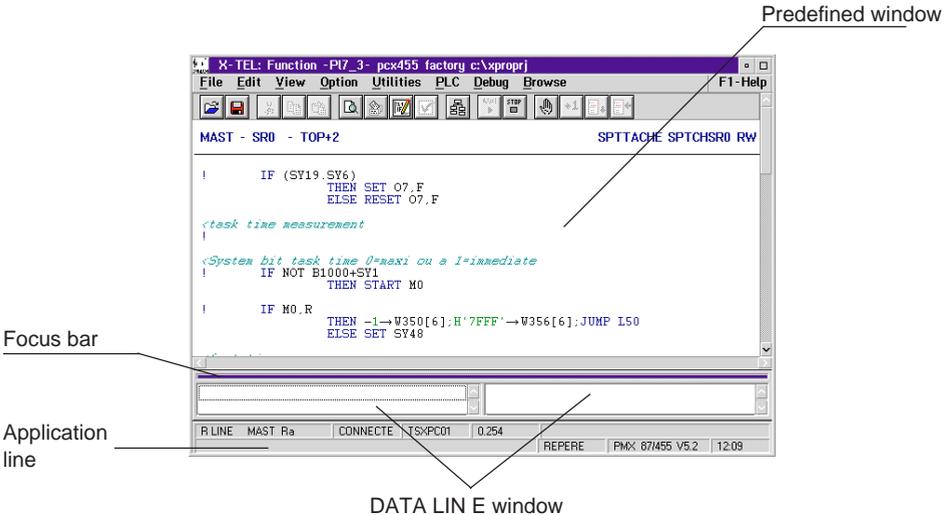
If the program is not executed, only the outputs of rack 0/1 are forced to 0, and the outputs of the other racks retain their states.

11.7 Accessing other modes

11.7-1 DATA LINE mode

This is in fact a sub-mode of DEBUG mode, accessible from the predefined screens via the function **Debug → Data Line Mode**.

Accessing this mode displays a data window between the predefined screen and the PL7-3 V6 line :



Changing focus

The focus determines whether the interactions between the user and the application affect the screen or the data line.

By default, the focus is on the predefined screen. Focus is changed in the following way : **<ALT>+<F3>** puts the focus on DATA LINE mode, **<ESC>** puts the focus back on the editor.

Using the mouse, click in the window (or screen) to be activated.

The focus bar is **blue** when the data line window is active ; it is **white** when the data line window is inactive.

Functions

The functions offered by this mode are the same as those in DATA mode (see section 10).

When the data window is active, the various menus (except View) are accessed by :

Using the keyboard, pressing **<ALT>**.

Using the mouse, clicking on the right-hand button in the window.

The **View** menu is accessed from the menu bar, as the variables display mode is common to the whole screen.

11.7-2 SEARCH/REPLACE mode

Accesses all the functions of the SEARCH/REPLACE mode (see section 14) :

- search for an object,
- replace an object by another compatible object,
- search for program elements associated with diagnostic code.

11.7-3 DATA mode

Debug, Data Mode	accesses the DATA mode (see section 10)
-----------------------------	-----------------------------------------

Close is used to return to DEBUG mode on the predefined screen which was displayed when the mode was changed.

Note

Some configuration parameters can also be modified. For further details, see section 5.10.

11.8 Searching for the program addresses of a variable

From a predefined screen, the command **<Alt>+<X>** accesses :

- the display of all the variables of the program element displayed,
- the selection of a variable and the display of all the program addresses relating to this variable,
- the selection of a program address and the display of the corresponding program element in a predefined screen.

The variable can be a standard PL7-3 variable (Wi, Bi etc) or an OFB variable.

This function is explained further in part H, section 15.5.

Section	Page
12.1 Presentation of TRANSFER mode	12/2
12.2 Processor - terminal memory transfers	12/3
12.2-1 Memory transfers from the PLC to the terminal (TSX → Terminal)	12/4
12.2-2 Memory transfers from the terminal to the PLC (Terminal → TSX)	12/5
12.2-3 Terminal - PLC memory comparison (Terminal → TSX)	12/6
12.2-4 Remote loading of a program	12/6
12.3 Save/Restore : internal words	12/7

12.1 Presentation of TRANSFER mode

This mode enables a PL7-3 application program to be transferred between :

- the internal memory of the terminal and the connected PLC,
- the hard disk of the terminal and the connected PLC,
- the hard disk and the internal memory of the terminal.

Access to the different functions

All TRANSFER mode functions can be accessed in offline mode (TERMINAL MEMORY) or in online mode (PROCESSOR MEMORY).

If the terminal is offline it changes automatically to online mode during transfer to the PLC and remains online for the period of transfer or comparison.

Transfer mode can be accessed either from the **APPLICATION Transfer** option or the **DATA WORDS Transfer** option in the **Processor** menu.

Details of the information transferred

The program comprises :

- the user program code (executable or non executable),
- the parameters and descriptors of the application configuration,
- the order code and the configuration parameters of the OFBs.

The constant words of the application (CWi and CDWi) and the internal OFB constants are always associated with the program during transfer.

Dynamic objects such as I/O, Bi, Xi, SYi bits and Wi and DWi words do not have a memory image in the terminal and are therefore not transferred during PLC-terminal exchanges.

Note

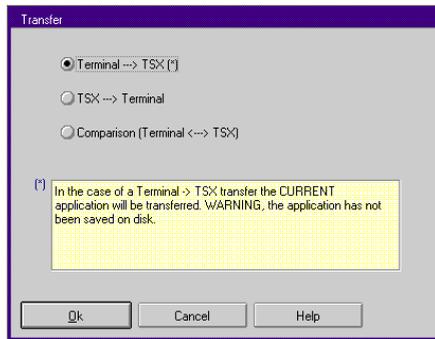
A transfer does not cause any data loss in the source memory, but completely overwrites the destination memory, replacing all previous data with the newly transferred data.

12.2 Processor - terminal memory transfers

This function, which can be accessed from the **Application Transfer** option in the **Processor** menu, is used to :

- transfer the application from the connected PLC to the internal memory of the terminal,
- transfer the application in the internal memory of the terminal to a connected PLC,
- compare the application in the internal memory of the terminal with that in the connected PLC.

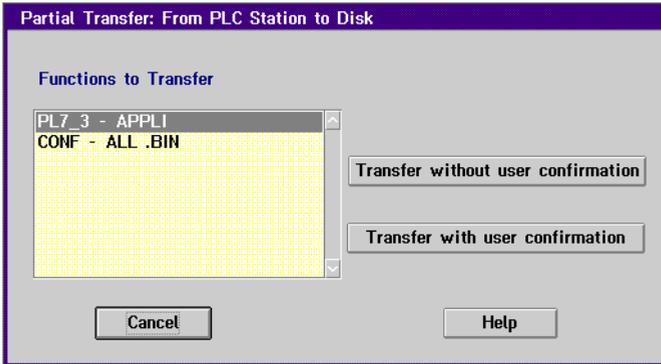
If the terminal is offline, the connection with the PLC is automatic and lasts for the duration of the transfer or comparison.



A TSX → Terminal transfer, via the PLC terminal port, can last several minutes.

12.2-1 Memory transfers from the PLC to the terminal (TSX → Terminal)

The **TSX → Terminal** button launches XTEL-TRANSFER which executes a PL7-3 transfer only, between the PLC and the hard disk (refer to the software workshop documentation). Once the transfer is complete, a positive response to the question READ PL7_3.BIN APPLICATION? enables transfer of the contents of the file to the terminal memory.



If the PLC is in RUN when the transfer is started, it will remain in RUN during and after the transfer.

Important

The PLC to Terminal PL7-3 transfer does not overwrite the PL7_3.bin file. This always retains the last saved application.

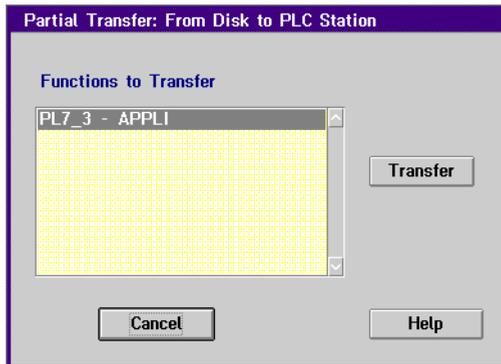
12.2-2 Memory transfers from the terminal to the PLC (Terminal → TSX)

Before a transfer, check that :

- the number and size of the RAM cartridges in the PLC are sufficient,
- the versions of the application and the PLC are compatible.

The **Terminal -> TSX** button launches the XTEL-TRANSFER tool in order to transfer the current application to the PLC.

The PL7_3.bin file, which is the image of the last saved application, is not overwritten.



If the PLC is already in RUN, it automatically switches to STOP during the transfer and remains in STOP after the transfer.

During the transfer, FAULT indicates that the application in the PLC cannot be executed for the moment. At the end of a successful transfer, the fault message disappears.

Important

- **this transfer deletes the program and the constants previously contained in the PLC memory.**
- **Only V5 applications can be transferred to a V5 processor. In addition, for this transfer to be authorized, the connected PLC must already contain a description of its dedicated memory zones. This description is entered using the XTEL-CONF and XTEL-TRANSFER tools. A PMX V5 application cannot be transferred to a TSX V5 PLC.**

12.2-3 Terminal - PLC memory comparison (Terminal → TSX)

The **Compare** button (**Terminal → TSX**) selects the comparison function, which will be executed after it has been confirmed. This function is also used to check that the contents of the internal memory and the contents of the PLC are identical.

The terminal switches to online mode automatically if it is offline. At the end of the comparison, the result is displayed (PROCESSOR MEMORY = TERMINAL MEMORY or PROCESSOR MEMORY <> TERMINAL MEMORY).

12.2-4 Remote loading of a program

This function loads a program to a PLC other than the one to which the terminal is physically connected. The PLC is then connected to the TELWAY, MAPWAY, ETHWAY or FIPWAY network.

To perform the load operation, simply specify the number of the station to which the terminal is logically connected. This address is defined under XTEL-CONF.

The transfer procedure is the same as described above, with network connection being performed automatically. Depending on the load of the network, the transfer time may be considerably extended.

12.3 Save/Restore : internal words

This function, which can be accessed from the **Transfer Data Words** option in the **Processor** menu enables :

- in offline operation :
 - transfer of data (internal words, Wi, and DWi) from the application in the internal memory of the terminal to the hard disk,
 - transfer of a file (xxx.DAT) from the hard disk to the internal memory of the terminal.
- in online operation :
 - transfer of data (internal words, Wi, and DWi) from the application in the connected PLC to the hard disk of the terminal,
 - transfer of a file (xxx.DAT) from the hard disk of the terminal to the connected PLC.

A TSX → terminal disk transfer, via the PLC terminal port, can last several minutes.



Directory lists all files already archived in the PL7_3\APPLI or PL7-3\MOD subdirectory, depending on the choice made.

OK confirms the selection made.

Cancel exits without confirming the selection.

Help calls up the program help function.

Comment

Saving or restoring an application on a disk is not performed at PL7-3 level, but at X-TEL level (see section 15.4).

Section	Page
13.1 Presentation of DOCUMENT/PRINT mode	13/2
13.1-1 General description	13/2
13.1-2 Schematic diagram	13/4
13.2 Composition of the documentation file	13/5
13.2-1 General	13/5
13.2-2 Title page	13/8
13.2-3 General information	13/9
13.2-4 Network information	13/10
13.2-5 Customized footer	13/11
13.2-6 Module information	13/12
13.3 Printing	13/14
13.3-1 Selecting the current printing parameters	13/14
13.3-2 Composition of the entire documentation file	13/15
13.3-3 Composition of the partial documentation file	13/18
13.4 Description of the listings	13/21

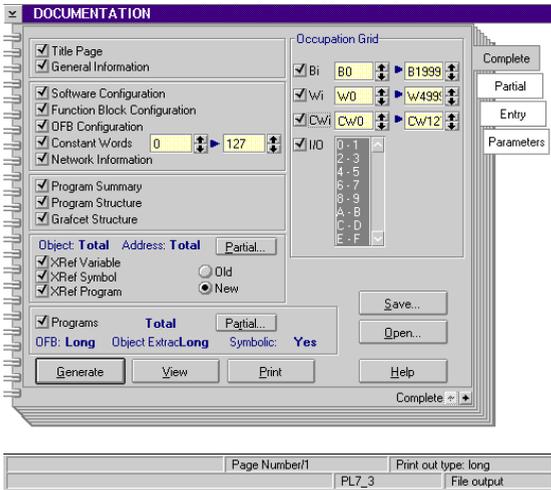
13.1 Presentation of DOCUMENT/PRINT mode

13.1-1 General description

DOCUMENT mode has two main functions :

- **documenting an application** which enables the user to enter and then archive comments on his application, in order to create a comprehensive and structured documentation file (dossier) including :
 - a title page,
 - up to 16 pages of general information,
 - program comments,
 - up to 16 pages of information concerning the networks,
 - a footer at the bottom of each page.
- **printing a documentation file** which allows the user to print a complete or partial documentation file for his application.

The DOCUMENT/PRINT mode can be accessed from the main window, either via the **Document** option in the **Tools** menu, or using the toolbar, in offline or online mode (PLC in RUN or in STOP). The following screen is displayed :



<TAB>

used to move from one group to the other.

<SPACE>

used to activate a box or a button.

<↓, <↑>

used to move within a group.

<←, >, <→>

<SHIFT>+

used to switch from one incremental entry zone to the other.

<↓, <↑>

Help

calls up the program help function.

For Long Listing or Short Listing printing (WHOLE DOCUMENTATION FILE), the following options are proposed :

- No** prints the documentation file using the parameters defined in the DOCUMENT/PRINT modes.
- Yes** creates a file which can be used by the XTEL-DOC tool, then prints this file.
The XTEL-DOC tool can only use the <Station>.DOC files stored in the Project\Station\PL7_3\APPLI directory.
If a <Station>.DOC file already exists, a message appears which allows the user to replace the old file with the new one : **Yes** or **No**, as required.

The general archive parameters are used implicitly to access the following files :

- xxx.TIT (title page),
- xxx.DES (general information),
- xxx.COM (program comments),
- xxx.NET (network information).

A fixed comment file name is defined for each task/module pair. The table, which can be accessed using the ENTRY function, lists all the task/module pairs configured and associates a predefined non-modifiable comment file name with each (see section 13.2-5).

Printing of a documentation file can be interrupted by pressing <ESCAPE>.

Note

Depending on the selection made in the **Parameters** tab (file or printer) : a file is created or the documentation file is printed.

Specific commands

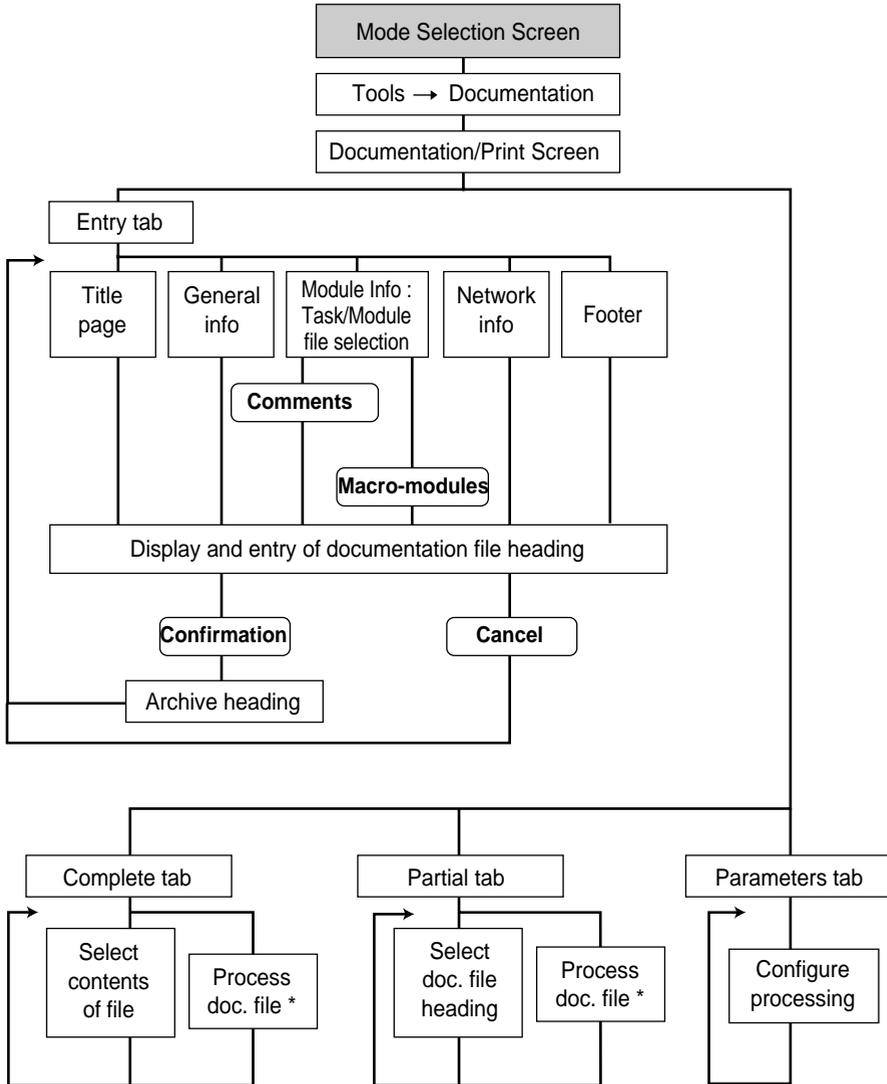
These commands, which are associated with the documentation and printing functions, can be accessed from the **Parameters** tab.

Note

When the selection of the contents of the documentation file has been defined, it is possible to store it in an xxx.CDE file using the **Save** command (button located on the main page).

The table of contents of a documentation file stored in an xxx.CDE file can be read via the **Open** command.

13.1-2 Schematic diagram



(*) Generate, view, print.

13.2 Composition of the documentation file

13.2-1 General

Entry and archiving

The information entered is stored on the hard disk. In order to do this, each entry function is associated with a file. The following table shows the correspondence between the various headings and the files :

3 -	ENTER TITLE PAGE	xxx.TIT
4 -	ENTER GENERAL INFORMATION	xxx.DES
5 -	ENTER PROGRAM COMMENTS	xxx.COM
6 -	ENTER NETWORK INFORMATION	xxx.NET
7 -	ENTER FOOTER	xxx.CRT

The configurations of the I/O, software, function blocks, OFBs, constant words and the program are stored in the xxx.BIN application file. The structure and table of contents, and the cross references are generated automatically.

In addition to these files are the files reserved for the system xxx.TMP and xxx.BAK.

Headings

The different headings and sub-headings contained in the complete printed documentation file are listed below :

Headings	Entry (operating mode)
1 Title page	DOCUMENT/PRINT
2 General information	DOCUMENT/PRINT
3 Network information	DOCUMENT/PRINT
4 Software configuration	CONFIGURATION
Function block configuration	CONFIGURATION
OFB configuration	CONFIGURATION
Constant words	CONSTANT
5 Program	PROGRAMMING
Program contents	Automatic
Program structure	Automatic
Grafcet structure	Automatic
6 Cross-references : variables	Automatic
Cross-references : symbols	Automatic
Cross-references : program	Automatic
7 Occupation grid	Automatic

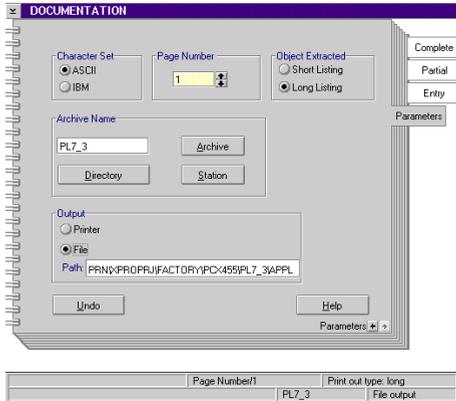
If the application is not documented, the documentation file only comprises headings 4, 5, 6 and 7.

The program file and the documentation files are independent of each other. It is therefore not necessary for the program to be stored in memory to be able to enter the information concerning the control system documentation file.

Recommendations

Before creating or modifying any file, it is recommended that :

- an archive name is defined. By default this name is the global archive name (PL7_3). This name can be changed using this screen, which can be accessed using the **Parameters** tab on the right of the notepad.



The **Path** file is used to modify the name (8 characters maximum) under which the files will be archived. It is advisable to use the same name as the program file name.

Directory lists all the file already archived in sub-directory PL7-3MOD.

Station gives the station name as the archive name.

Archive gives the global archive name (PL7_3) as the archive name.

<TAB> used to move from one group to the other.

<↓>, <↑> used to move within a group.

<→>, <←>

Undo used to return the data on the notepad in their initial state (ie. when the page was selected).

Help calls up the program help function.

- the revision index (REV) is entered on the **Entry** tab on the FOOTER heading. This index appears in the footer when the documentation file is printed.

The documentation is entered heading by heading. All the headings are automatically saved when they are confirmed.

Make frequent saves of the documentation file being created to the hard disk. When entry of the documentation file is completed, copy it immediately to diskette using the X-TEL Export function.

Anomalies, archiving principle, recovering from errors

Although most documentation headings are archived automatically, it is useful to understand the archiving principle, notably when recovering a file after a power break or an anomaly. When creating or modifying a heading, the system creates a provisional xxx.BAK file, which is later erased when the heading is confirmed.

If an anomaly occurs during entry, the current file may be erased. It is then possible to recover the information entered before the last modifications were made, by immediately copying (without opening a new heading) the xxx.BAK file, instead of the heading which was being modified when the fault occurred.

Entering documentation

This screen, which can be accessed via the main **Entry** tab of the documentation file, is used to enter the information presented in the following sections.

- <TAB> used to move from one group to the other.
- <↓>, <↑> used to move within a group.
- <→>, <←> predefined file.
- Help** calls up the program help function.

13.2-2 Title page

This page serves as a header for the documentation file. It enables entry of general information concerning the application.

Title page accesses the title page entry screen, from the **Entry** tab.

Access to the title page file (xxx.TIT) is implicit and uses the global archive parameters.

REV.	DATE	REVISION	DESIGNER	EXECUTED by

The arrow (←↑→↓) and <Tab> keys are used to position the cursor in the predefined zones.

The **Title** entry field is used to define the name of the application (64 characters).

The **Company** field defines the names of the companies (16 characters maximum) involved in designing, using and maintaining the application.

The **Department** entry field defines the names of the departments (16 characters maximum) involved in designing, using and maintaining the application.

The **Manager** entry field defines the names of the managers (16 characters maximum) involved in designing, using and maintaining the application.

The **REV.** column is used to enter the revision index of the documentation file (3 characters maximum).

The **DATE** column is used to enter the date that the documentation file was created or modified (8 characters maximum).

The **REVISION** column is used to enter a comment (32 characters maximum) to explain the revision made.

The **DESIGNER** column is used to enter the name of the designer (12 characters maximum).

The **EXECUTION** column is used to enter the name of the operator (12 characters maximum).

OK confirms entry of the title page.

Cancel cancels data entry and returns to the main window.

13.2-3 General information

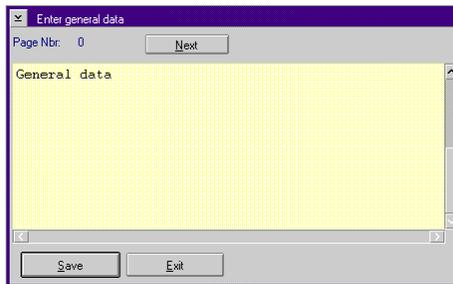
An editor enables entry of information concerning the application, including :

- a general description and global operation of the application,
- any specific procedure, usage or maintenance precaution,
- description of the program,
- a list of documents relating to the installation, etc.

General Information accesses the editor, from the **Entry** tab.

This editor is used to enter up to 16 pages of free text (pages 0 to 15). Each page comprises 56 lines of 80 columns.

Access to the general information file (xxx.DES) is implicit and uses the global archive parameters.



The parameter display line indicates the page currently being edited.

Reading the text

- Next** accesses the next page. After page 15, page 0 is displayed.
- Previous** accesses the previous page. After page 0, page 15 is displayed.
- Save** saves the information entered.
- Cancel** used to exit without saving.

Edit functions

	<CTRL>+<INS>	copies the selected text block to the buffer memory.
	<SHIFT> +<INS>	restores the text block stored in the buffer memory before the current line.
	<SHIFT> +	deletes the text block selected and stores it, in order to move or duplicate it on the current page or on another page. The selected block remains in the memory (even after it has been restored) until the next use.

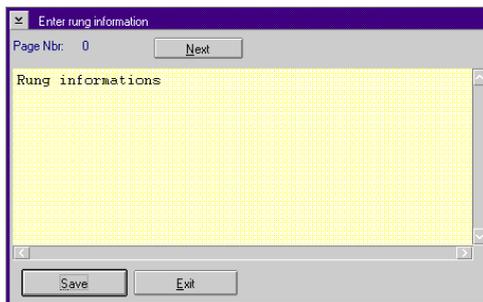
13.2-4 Network information

An editor is used to enter information concerning the application PLC network, including :

- a general description and global operation of the application,
- any specific procedure, usage or maintenance precaution,
- description and structure of the network,
- a list of documents relating to the installation etc.

Network Information accesses the editor, from the **Entry** tab.

This editor is used to enter 16 pages of free text.



Functions of the editor

The commands of this editor are the same as those for the text editor, used to enter general information (see section 13.2-3).

13.2-5 Customized footer

This function allows entry of 3 character fields in order to customize the footer which will be printed at the bottom of each page in the documentation file :

- 1 field of 25 characters maximum,
- 1 field of 45 characters maximum,
- 1 field of 3 characters maximum (rev), revision index number of the documentation file.

Footer accessed from the **Entry** tab.

The arrow keys (↑↓) select the character field to be entered or modified.

OK used to save the customized footer to an xxx.CRT file, in a PL7_3/APPLI subdirectory.

Cancel cancels data entry and returns to the previous window.

<TAB> used to move from one group to the other.

<↓>, <↑> used to move within a group.

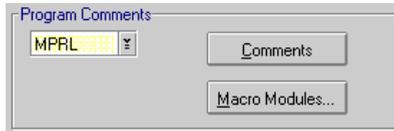
<→>, <←>

Caution

If the footer has been entered under PL7-3 V4 or PL7-3 V5, it cannot be used with PL7-3 V6. In this case, the fields in the footer must be re-entered using PL7-3 V6.

13.2-6 Module information

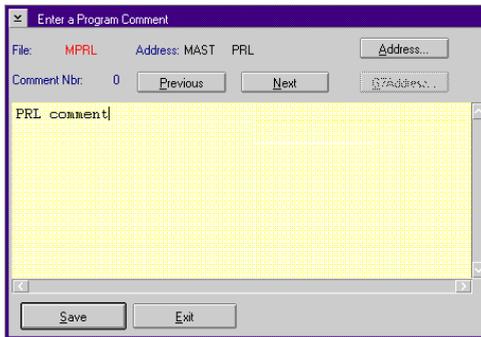
For every task/module pair, a fixed comment file name is defined. This file is stored in the MOD zone as soon as it contains at least one comment. The table created lists all the configured pairs. This table cannot be modified by the user and the task/module pairs are all associated with a predefined and non-modifiable comment file name. The list is used to select the predefined file.



Once the file has been selected it is possible to access either the program comments (with a view to entering or modifying them), by pressing the **Comments** key, or the module and macro-module information, by pressing the **Macro-modules** key.

Program comments

This function is used to assign a 10 line comment with a maximum of 80 characters to a program element. When printed this comment will appear above the associated program element (Ladder language, Literal statement, program module, etc).



Correspondence of the comment entered in DOCUMENT/PRINT mode with the program element is performed by the program address defined by the **Address** and **G7 Address** keys.

Editing comments

A dialog box is used to enter the label of the program element.

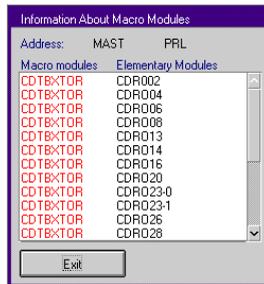
- Next** accesses the next comment, from comment 499 to 0.
- Previous** accesses the previous comment, from comment 0 to 499.
- Address** used to enter the address of the program element associated with the current comment : label.
- G7 Address** used to enter the program element address (action or transition condition), assigned to the current comment : step or transition reference.
- Save** saves the information entered.
- Cancel** used to exit without saving.

Functions of the editor

The commands of this editor are the same as those for the text editor used for entering general information (see section 13.2-3)

Information on macro-modules

This page can only be viewed. It displays the list of macro-modules and modules contained in the task/module pair selected previously.



- Cancel** returns to the previous screen.

13.3 Printing

The PRINT function is used to print :

- the complete application documentation file, with option for its contents,
- any heading of the documentation file.

Before a program can be printed, it must be loaded into the internal memory of the terminal (in offline mode) or in the user memory (in online mode).

The global archive parameters are used implicitly to access the following files :

- title page,
- general information,
- network information.

13.3-1 Selecting the current printing parameters

The parameter selection screen is accessed from the **Parameters** tab.

The **Short Listing** (Ladder language and configuration of the OFBs are more compact), and **Long Listing** buttons are used to select the type of printout.

The **Page Number** entry zone is used to enter the number of the first page of a partial documentation file.

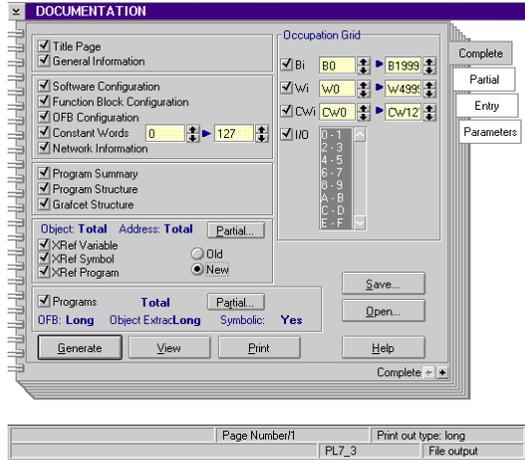
The **Archive Name** entry zones are used to modify the global archive name of the files to be printed, as well as their storage destination.

The **ASCII** and **IBM** buttons determine the type of printer : IBM semi-graphic or ASCII.

Additional function

	<ESCAPE>	cancels printing.
--	-----------------------	-------------------

13.3-2 Composition of the entire documentation file



By default, all the headings are active.

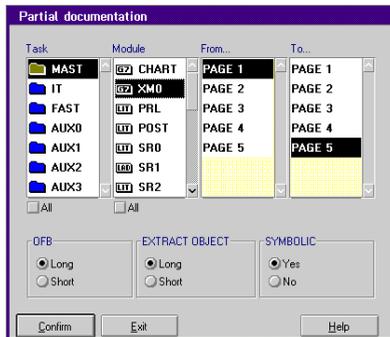
Constant words

By checking the **Constant Words** box it is possible to select all the constant words to be printed or viewed.

Program

When checking the **Program** box all headings are confirmed by default.

The **Partial** key is used to select part of the program. The following screen appears :



The **All** check box under the task column selects all the program. The **All** check box under the module column selects all the modules.

For partial selection of the program, select the task and the module if required, and the lines required.

The **OFB** button zone is used to print with (long) or without (short) the graphic form of the OFBs called.

The **Extract Object** button zone is used to print the objects and symbols with (long) or without (short) comments.

The **Symbol** button zone is used to determine whether objects are printed as symbols or variables.

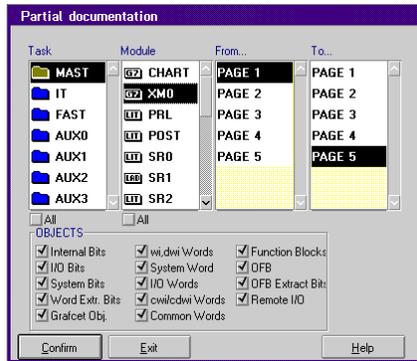
- OK** confirms the selections made.
- Cancel** used to exit without modification.
- Help** calls up the program help function.

Cross-references

The **Old** and **New** buttons are used to systematically recalculate the cross-reference tables or to print the current tables.

All the headings are confirmed by default.

The **Partial** key is used to select part of the program. The following screen appears :



The **All** check box under the task column selects the whole program. The **All** check box under the module column selects all the modules.

For partial selection of the program, select the task and the module, if required, as well as the required lines.

The **Objects** zone selects the objects to which the cross-references apply. All objects are selected by default.

- OK** confirms the selections made.
- Cancel** used to exit without modification.
- Help** calls up the program help function.

<TAB> used to move from one group to another.

<↓>, <↑> used to move within a group.

<→>, <←>

Occupation grid

By checking the **Bi / Wi / CWi** boxes it is possible to select subsets of bits, words and constant words in the application.

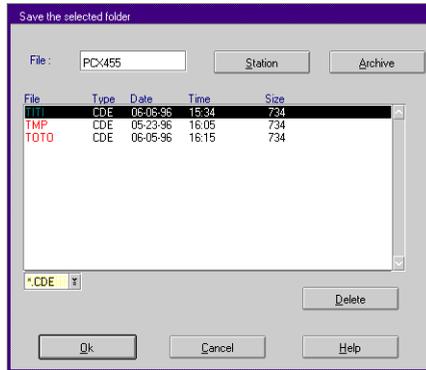
The **I/O** check box is used to select one or more double racks.

Processing the documentation file

Save used to save the menu of the customized documentation file to an xxx.CDE file.

Open used to read an xxx.CDE file which has previously been archived.

Selecting either of these two options displays the following window :



The **File** entry zone is used to enter the archive file name. When saving, the new file will overwrite the old one of the same name.

Station assigns the current station name as the file name.

Archive assigns the global archive name (PL7_3) as the file name.

Generate generates the documentation file in the xxx.doc file (PL7-3.doc by default).

View presents the documentation file on the screen as it will be printed.

Print prints the documentation file.

Help calls up the program help function.

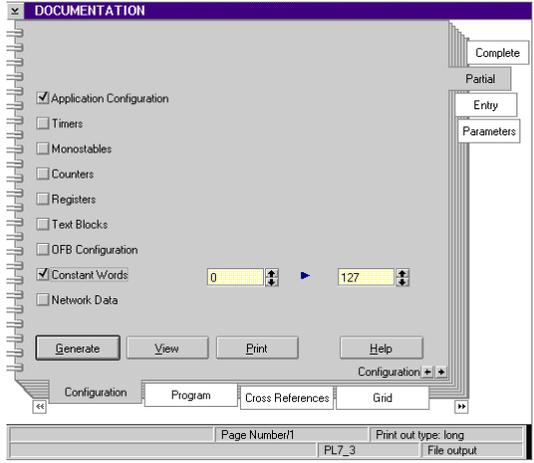
<TAB> used to move from one group to another.

<↓>, <↑> used to move within a group.

<→>, <←>

13.3-3 Composition of the partial documentation file

This function, which can be accessed from the **Partial** tab, offers the possibility of printing or viewing part of the documentation file.



The composition of the partial documentation file is subdivided into 5 screens, which can be accessed from secondary tabs on the notepad :

- configuration,
- program,
- references,
- grids,
- miscellaneous.

In each of these screens, the following functions are available :

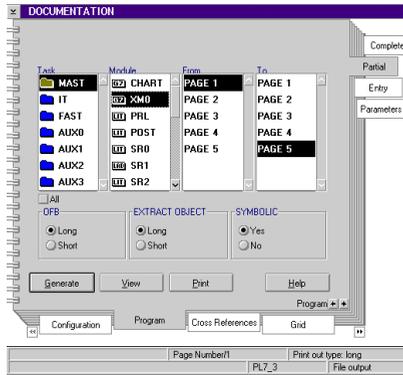
- Generate** generates the partial documentation file following the xxx.doc file.
- Consult** edits the documentation file.
- Print** prints the documentation file.
- Help** calls for software help.

- <TAB> used to move from one group to the other.
- <↓>, <↑> used to move within a group.
- <→>, <←>
- <SPACE> used to activate a check box or a button.

CONFIGURATION SECONDARY TAB

This screen, which is offered by default, is used to introduce or selectively ignore the configuration parameters (application, timers, monostables etc) when building the documentation file.

PROGRAM SECONDARY TAB :



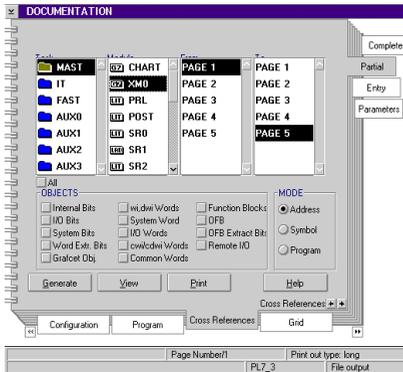
The **All** check box under the task column selects the whole program. The **All** check box under the module column selects all the modules.

For partial selection of the program, select the task and the module, if required, as well as the required lines.

The **OFB** button zone is used to print with (long) or without (short) the graphic form of the OFBs called.

The **Symbol** button zone is used to determine whether objects are printed as symbols or variables.

REFERENCES SECONDARY TAB :

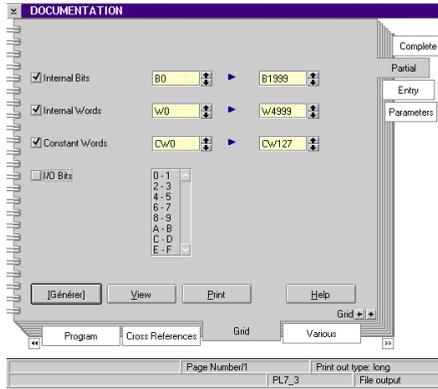


The **Objects** zone selects the objects to which the cross-references apply. All objects are selected by default.

The **All** check box under the task column is used to select the whole program. The **All** check box under the module column is used to select all the modules.

For partial selection of the program, select the task and the module, if required, as well as the required lines.

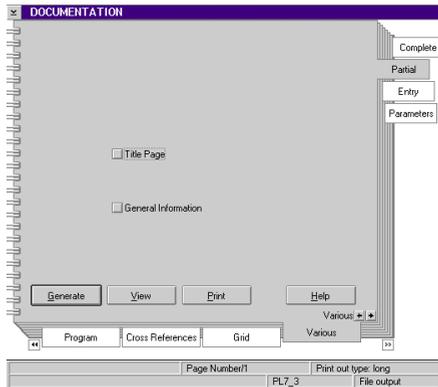
GRID SECONDARY TAB :



By checking the **Bi / Wi / CWi** boxes it is possible to select subsets of bits, words and constant words in the application.

The **I/O** check box is used to select one or more double racks.

MISCELLANEOUS SECONDARY TAB :



The buttons on this screen are used to view or print the title page and the general information.

13.4 Description of the listings

The complete listing of an application includes the following :

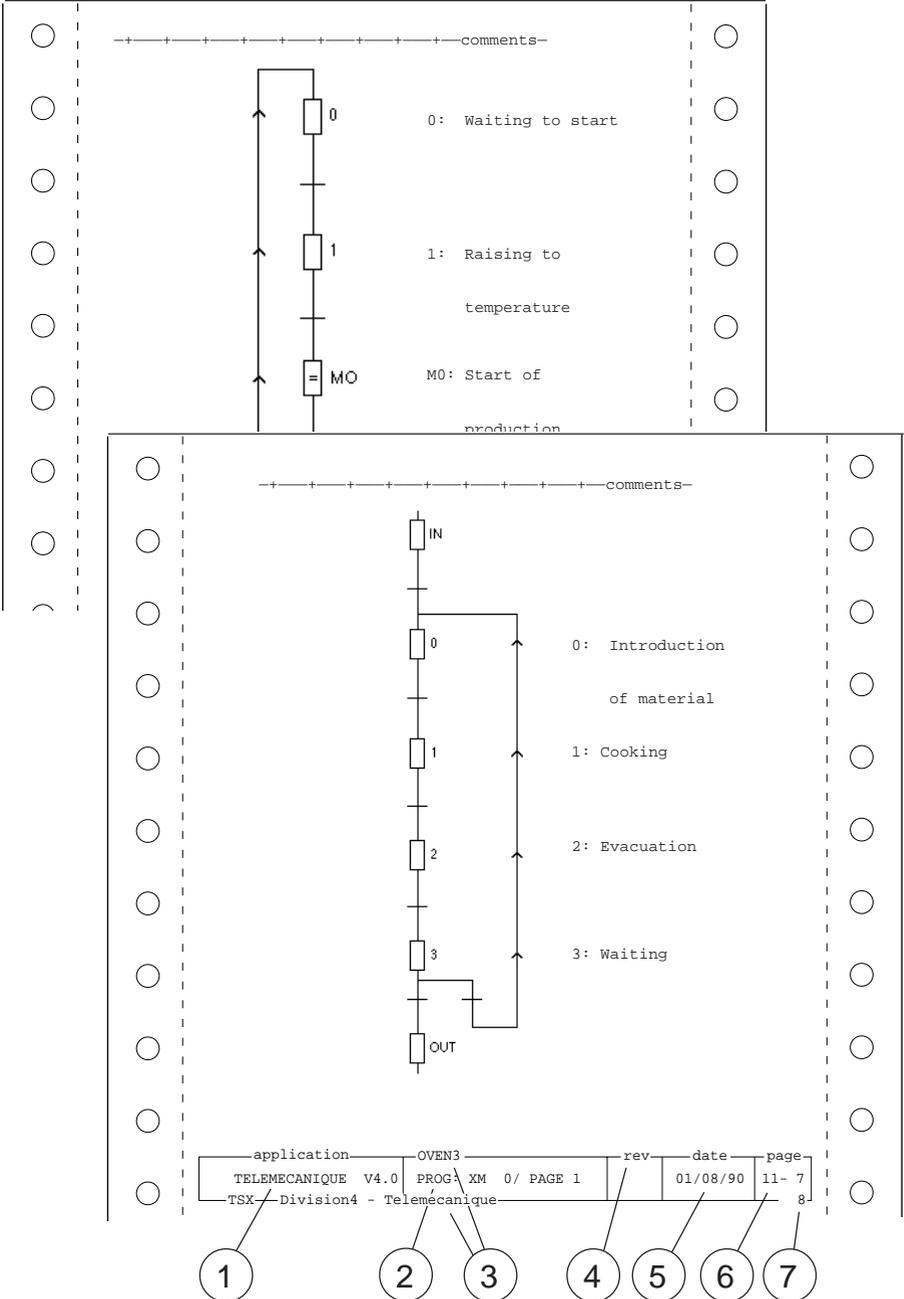
- 1 - Title page,
- 2 - General information,
- 3 - Software configuration :
 - application configuration,
 - function block configuration,
 - constant words,
- 4 - Network information,
- 5 - Program :
 - IT, fast, master and auxiliary tasks,
 - program structure, table of modules and Grafcet structure,
 - program contents,
- 6 - Cross-references :
 - by variables,
 - by symbols,
 - by subroutine calls,
- 7 - Occupation grid
 - internal bits,
 - I/O bits,
 - internal words,
 - constant words,
- 8 - Table of contents of the documentation file.

Description of a listing page

Each A4 listing page comprises :

- the information (rungs, Literal or Grafcet pages, I/O wiring, etc),
- a footer at the bottom of each page specifying :
 - 1 the name of the application entered in configuration mode,
 - 2 the description of the printed heading,
 - 3 the customization of the footer entered in the mode,
 - 4 the document version number (number entered in ENTRY mode),
 - 5 the date of printing,
 - 6 the page number for each heading,
 - 7 the absolute page number.

Example of listing pages in semi-graphic mode

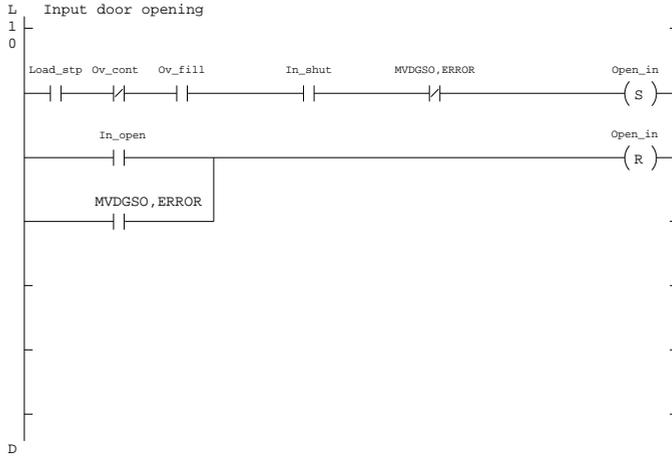


Example of program elements with comments

MONITORING OF ENTRY DOOR OPENING

Fault generated by
 -start position still present
 -arrival position not reached
 -discordancy check.

Monitoring of the fault by the movement diagnostic OFB MVDGSO.



```
EXEC MVDGSO ( ; Open_in ; In_open ; NOT ( In_open . In_shut ) ; NOT In_shut ; Nul_ofb => ; )
```

OPTIONAL FUNCTION BLOCK I/O PARAMETERS

	MVDGSO					
Open_Ed	-	INIT	bit	ERROR	bit	-
Ed_open	-	ENABLE	bit	STATUS	dwor	-
NOT (Open_ed.Ed_shut)	-	EVENT	bit	TIME	word	-
NOT Ed_shut	-	COND	bit			
Nul_ofb	-	EVENT_T0	bit			
	-	EVENT_T1	bit			

- I2,6 Open_ed Entry door open
- X0,0 Intro Introduction of material
- I2,3 Mat Material present in the oven
- I2,4 Intr_req Introduction request
- I2,5 Ed_shut Entry door shut
- O3,3 Open_ed Open entry door
- B5 Nul_ofb Always at zero

application	OVEN3	rev	date	page
TELEMECANIQUE V4.0	PROG: MAST / POST		01/08/90	11- 11
TSX—Division4 - Telemecanique				12

Example of cross-references by variables

The USE column defines the utilization of the variable in the program :

- **R** : reading of a word or a bit,
- **W** : writing of a word or a bit,
- **R-X** : reading of an indexed word,
- **W-X** : writing of an indexed word,
- **X** : index value,
- **RT** : reading a table,
- **WT** : writing a table,
- **RT-X** : reading an indexed table,
- **WT-X** : writing an indexed table,
- **RM** : reading of a complete I/O module,
- **WM** : writing of a complete I/O module,
- **INT** : interrupt management instruction.

Note

The character \$ in the USE column indicates that an object is written several times.

BITS							
VARIABLE	SYMBOL	TASK	MODULE	SUB-MODULE	LABEL	OFFSET	USE
B0	C_init	MAST	CHART	X0 ->X1	TOP		R
B1	Fault	MAST	CHART	X0 ->X1	TOP		R
B2	Temp_ok	MAST	CHART	X1 ->XM0	TOP		R
B3	Temp_low	MAST	CHART	X2 ->X0	TOP		R
B5	Nul_ofb	MAST	PRL		L15	+1	W
		MAST	POST		L10		R
		MAST	POST		L25		R
		MAST	POST		L30		R
		MAST	POST		L50		R
B50	Flt_tem1	MAST	PRL		L100		W
		MAST	PRL		L100		R
B51	Flt_tem2	MAST	PRL		L110		W
		MAST	PRL		L110		R
B52	Flt_tem3	MAST	PRL		L120		W
		MAST	PRL		L120		R
B100	Ini_aem	MAST	PRL		L15		W

application	OVEN3	rev	date	page
TELEMECANIQUE V4.0	XREF BY VARIABLES		01/08/90	14- 1
TSX - Division4 - Telemechanique				24

Example of a program structure (program tree structure)

Note

*** indicates that the corresponding program element calls one or more subroutines.

PROGRAM STRUCTURE — SUBROUTINE CALLS				
MAST _____ POST. _____ SR0				
application	OVENS	rev	date	page
TELEMECANIQUE V4.0	PROC: STRUCTURE		01/02/90	12- 1
TEI - Division 4 - Telemecanique				



Section	Page
14.1 Presentation of SEARCH/REPLACE mode	14/2
14.2 Search function	14/4
14.3 Replace function	14/6
14.4 Diagnostics function	14/8
14.5 Function for moving "FIP" equipment	14/10
14.6 Linking searches	14/12
14.7 Example of a listing	14/14

14.1 Presentation of SEARCH/REPLACE mode

SEARCH/REPLACE mode is accessible from :

- the **Search/Replace** option from the **Tools** menu, in offline or online mode, It uses either the variable or the symbol mode and has 3 functions :

Search

This function is used to search for an object in :

- all or part of an application program (PLC running or stopped),
- one or more basic modules and/or macro-modules.

Replace

This function is used to search for an object and replace it with a **compatible** object in :

- all or part of an application program (PLC running or stopped),
- one or more basic modules and/or macro-modules.

It is therefore possible, except in exceptional circumstances, to replace a bit in the bit memory (I/O bit) by a RAM bit (for example : internal bit). This applies to all types of objects : bit, word, double word, etc.

The restrictions for replacing an object by another object are as follows :

- an IOIM bit, used on an edge instruction (FE, RE) cannot be replaced by a RAM bit. RAM bits do not have an edge.
- an internal word W_i , modified during execution by a program instruction, cannot be replaced by a constant word CW_i . Constant words cannot be modified by program.
- a standard function block object cannot be replaced by a different standard function block object (for example : C_i,V cannot be replaced by M_i,V),
- replacing a step X_i by a step X_j does not involve replacement of other bit or word type objects associated with step X_i . This is also true for an XM_i macro-step.

The Replace function is also used to move "FIP" equipment in :

- all or part of an application program,
- one or more basic modules and/or macro-modules.

Diagnostics

This function is used to search for program elements with an associated diagnostic code in :

- all or part of an application program (PLC running or stopped),
- one or more basic modules and/or macro-modules.

The object searched for or replaced can be :

- a bit,
- a word or double word,
- a standard or optional function block,
- a standard or optional function block element,
- a remote I/O element (for the "FIP" equipment),
- a bit extracted from a word, a function block or remote I/O.

Searches can be performed on objects which are read, write or read/write in the program modules written in Literal or Ladder language. When replacing, the objects are automatically read/write.

The search/replace function is not authorized for bits or words indexed or included in a table.

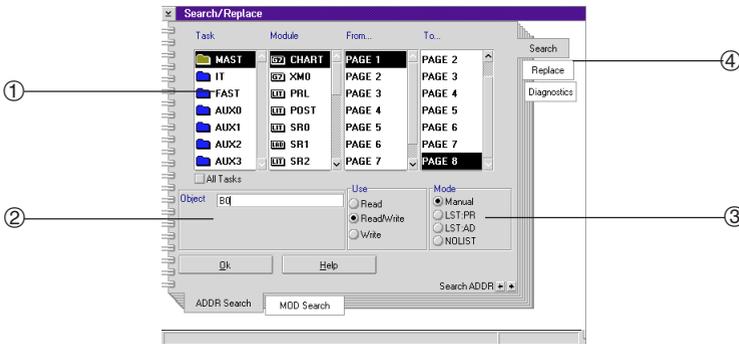
The search/replace and diagnostic functions are performed :

- either in automatic mode, with or without creation of a listing,
- or in manual mode stopping on a program element containing an object being searched/replaced. In this case the terminal waits for an operator command to continue.

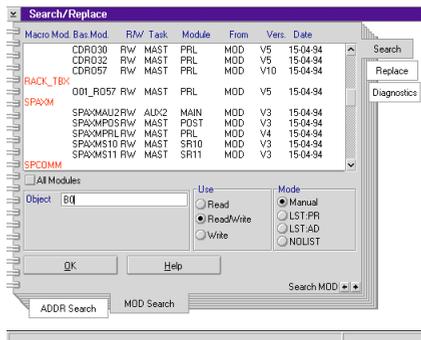
14.2 Search function

The Search function is selected from the main **Search** tab on the right of the notepad. This function is selected by default when the notepad is displayed and is used to search for an object in the application program (secondary tab Search ADR selected). The screen is used to display the options selected by default or by the user :

- ① display of the part of the application program or list of basic modules and/or macro-modules affected by the search (by default, all the program or all the basic modules and macro-modules).
The part of the program (task, module, program element) on which the search will be made is selected using the <←>, <↑> keys to select the required element, and the <↓>, <↓> keys to change the selection column, or using the mouse.
- ② indication of the object affected by the function.
- ③ display of :
 - the use criterion of the object to be found (read, write, read/write),
 - the operating mode : manual or automatic, with or without listing.
- ④ Indication of the current function.



The function to search for an object in the basic modules and/or the macro-modules can be accessed from the **Search MOD** secondary tab at the bottom of the notepad, which displays the following screen :



The **All Modules (All Tasks)** check box is used to search for an object in all of the application program or in all the basic modules and macro-modules.

The **Object** field is used to enter the object for which a search will be made. After confirming this object by pressing **<ENTER>**, the symbol associated with the object is displayed below the entry zone.

The **Use** group of buttons selects the usage criterion of the object for which a search will be made :

Read the read objects,
Read/Write all the objects,
Write the write objects.

The **Mode** group of buttons selects the operating mode for the search :

Manual : manual mode. The terminal waits for a command from the user to continue its search.

LST:PR : automatic mode with printout of a complete listing giving the labels and the program elements containing the object searched for.

LST:AD : automatic mode with printout of a partial listing giving only the labels containing the object searched for.

NOLIST : automatic mode without printout of a listing.

The **OK** button is used to launch the search operation after option selection.

Help calls up the program help function.

<TAB> is used to move from one group to another.

<↑><↓> are used to move within a group.

<→><←>

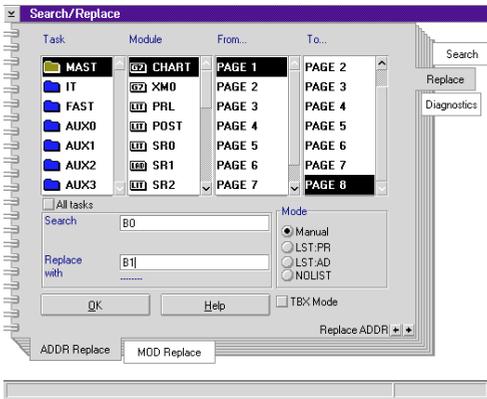
<SPACE> activates the check box.

14.3 Replace function

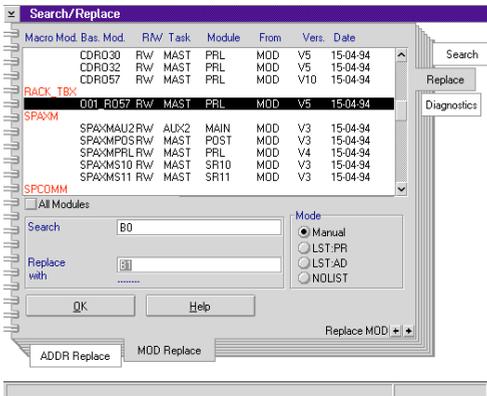
This option is selected by using the main **Replace** tab. Replacing an object by another object can be performed in either offline or online mode. The part of the program (task, module, program element) on which the replacement will be made is selected using the <↓>, <↑> keys to select the required element, and the <←>, <→> keys to change the selection column, or by using the mouse.

Replacement is possible when the PLC is running or stopped.

An object must always be replaced by a **compatible** object.
For example B1 can be replaced by B50 or by W5,3.



The function to replace an object in the basic modules and/or macro-modules can be accessed from the **Replace MOD** secondary tab at the bottom of the notepad, which displays the following screen.



The **All Modules (All Tasks)** check box is used to replace an object in all of the application program or in all the basic modules and macro-modules.

The **Search** field is used to enter the object to be replaced. After confirming this object by pressing **<ENTER>**, the symbol associated with the object is displayed below the entry zone.

The **Replace by** field is used to enter the new object.

The **Mode** group of buttons selects the operating mode for the search :

Manual : manual mode. The terminal waits for a command from the user to continue its search.

LST:PR : automatic mode with printout of a complete listing giving the labels and the program elements containing the object searched for.

LST:AD : automatic mode with printout of a partial listing giving only the labels containing the object searched for.

NOLIST : automatic mode without printout of a listing.

The **TBX Mode** check box accesses the function for moving "FIP" equipment, in the application program or in the basic modules and/or macro-modules.

The **OK** button is used to launch the search operation after option selection.

Help calls up the program help function.

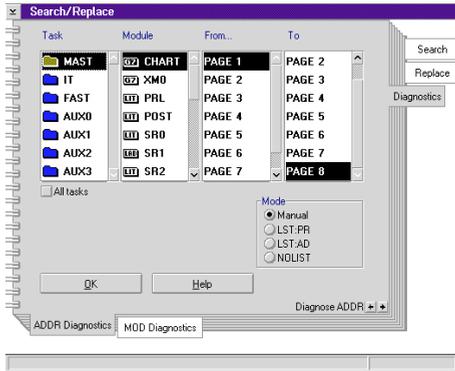
<TAB> is used to move from one group to the other.

<↑><↓> are used to move within a group.

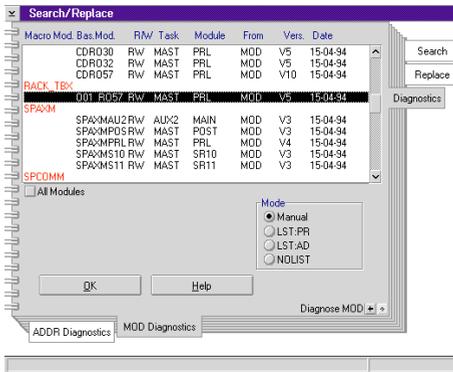
<→><←>

<SPACE> activates the check box.

14.4 Diagnostics function



The diagnostics of an object in the basic modules and/or macro-modules can be accessed from the **Diagnostics MOD** tab at the bottom of the notepad, which displays the following screen.



The **All Modules (All Tasks)** check box is used to perform a diagnostic search for an object in all of the application program or in all the basic modules and macro-modules. A diagnostic search in a macro-module consists of performing a diagnostic search in all the basic modules which make up the macro-module.

The **Mode** group of buttons selects the operating mode for the search :

Manual : manual mode. The terminal waits for a command from the user to continue its search.

LST:PR : automatic mode with printout of a complete listing giving the labels and the program elements containing the object searched for.

LST:AD : automatic mode with printout of a partial listing giving only the labels containing the object searched for.

NOLIST : automatic mode without printout of a listing.

Help	calls up the program help function.
<TAB>	is used to move from one group to the other.
<↑><↓>	are used to move within a group.
<→><←>	
<SPACE>	activates the check box.

14.5 Function for moving "FIP" equipment

This function, accessed via the **TBX Mode** command, from the main **Replace** tab and the **Replace ADR** secondary tab, is used to replace all the objects of a "FIP" device in the application program or in the basic modules and/or macro-modules; for example, after changing a TBX address on the FIPIO bus.

The source and target equipment addresses are entered in the following format :

connection n°[,module n°].

For example :

- 2,0 : connection point 2, module 0 (base),
- 2,1 : connection point 2, module 1 (extension),
- 3 : connection point 3, module 0 and 1 (base and extension).

The value entered is displayed on the screen in 3 possible formats :

- *i,0,* : all base module objects,
- *i,1,* : all extension module objects,
- *i,*,* : all base and extension module objects,

After entry, the software checks the source and target equipment compatibility : same type of equipment, same number of channels and same channel configuration (inputs or outputs). On confirmation, replacement of all the "FIP" equipment objects (RI, RO, etc) is launched automatically. When replacement is complete, the number of occurrences found and replaced is not displayed.

Role of control zones

Mode the role of this key is identical to that of the replace function, described in section 14.3. Nevertheless, in order to avoid interrupting the current replacement procedure (ABORT), which would make the application inconsistent, this key does not allow a change to **Manual** mode.

Search is used to enter the source equipment address : connection n°[,module n°].

Replace by is used to enter the target equipment address : connection n°[,module n°].

TBX Mode is used to return to the object replace function (see section 14.3).

Help calls up the program help function.

<TAB> is used to move from one group to the other.

<↑><↓> are used to move within a group.

<→><←>

<SPACE> activates the check box.

<ESC> is used to interrupt the search procedure in progress.

Note

Depending on the size of the program, moving "FIP" equipment may take some time.

14.6 Linking searches

Automatic search, replace or diagnostics

The **Mode** button group is used to select one of the automatic modes.

The **OK** button is used to launch execution of the selected function :

- automatic search, diagnostics or replace,
- with or without printout of a listing.

At the end of the search, the message OCCURRENCE(S) appears, indicating :

- the number of times the object has been found (in search mode),
- the number of times the object has been found and replaced (in replace mode),
- the number of program elements diagnosed (in diagnostics mode).

Manual search or diagnostics

The **Mode** button group is used to select manual mode.

The **OK** button is used to launch the search for the first object (in search mode) or program element diagnosed (in diagnostics mode).

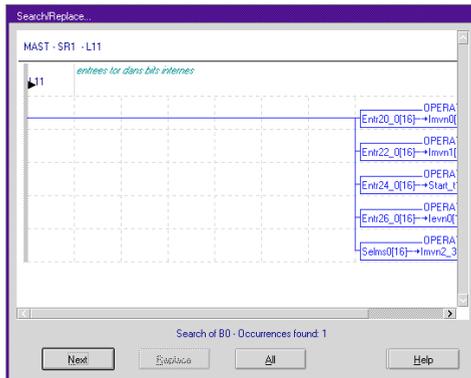
The element which contains the object is displayed on the screen.

The **Next** button is used to search for the next object or program element.

The **All** button is used to perform a search for all the occurrences of the object without displaying all the elements which contain it in succession.

The **Quit** button is used to stop the search in progress.

A message indicates the number of times the object or program element diagnosed has been found.



Help

calls up the program help function.

<TAB>

is used to move from one group to the other.

<↑><↓>

are used to move within a group.

<→><←>

Manual replacement

The **Mode** button group is used to select manual mode.

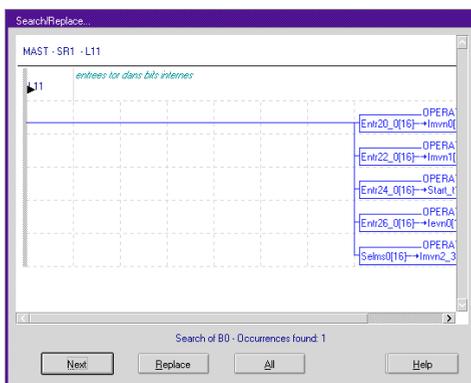
Launch the search for the first object to be replaced with the **OK** button. The element which contains it is displayed on the screen.

The **Next** button is used to search for the next object without performing a replacement.

The **Replace** button is used to confirm the replacement of the object displayed on the screen by the new object.

The **Quit** button is used to stop the replacement procedure.

A message indicates the number of times the object has been found and replaced.



Help

<TAB>

<↑><↓>

<←><→>

calls up the program help function.

is used to move from one group to the other.

are used to move within a group.

14.7 Example of a listing

LST:PR is used to printout a listing, indicating the labels and the elements containing the object before and after replacement.

<input type="radio"/>	Function	:	REPLACEMENT	<input type="radio"/>
	Object(s)	:	W100 : W113	
<input type="radio"/>	Application name	:	TELEMECANIQUE V5.0	<input type="radio"/>
	Date	:	01/08/90	
<input type="radio"/>	*****			<input type="radio"/>
	Program line	:		
<input type="radio"/>	!		IF [Temp_b1<Cons_b1]	<input type="radio"/>
			THEN SET Ma_b1	
<input type="radio"/>	New program line	:		<input type="radio"/>
	!		IF [Delta<Cons_b1]	<input type="radio"/>
<input type="radio"/>			THEN SET Ma_b1	<input type="radio"/>

LST:AD is used to print out a listing, indicating the labels and the elements containing the object replaced.

<input type="radio"/>	Function	:	REPLACEMENT	<input type="radio"/>
	Object(s)	:	W113 : W100	
<input type="radio"/>	Application name	:	TELEMECANIQUE V5.0	<input type="radio"/>
	Date	:	01/08/90	
<input type="radio"/>	Program address	:	MAST CHART S(C) 1 TOP +1	<input type="radio"/>
	Occurrence(s)	:	1 OCCURRENCE(S)	
<input type="radio"/>	Program address	:	MAST CHART S(C) 1 TOP +3	<input type="radio"/>
	Occurrence(s)	:	1 OCCURRENCE(S)	
<input type="radio"/>	Program address	:	MAST CHART S(C) 1 TOP +5	<input type="radio"/>
	Occurrence(s)	:	1 OCCURRENCE(S)	
<input type="radio"/>	Program address	:	MAST SR0 TOP +2	<input type="radio"/>
	Occurrence(s)	:	1 OCCURRENCE(S)	
<input type="radio"/>	Program address	:	MAST SR0 TOP +4	<input type="radio"/>
	Occurrence(s)	:	1 OCCURRENCE(S)	
<input type="radio"/>	Program address	:	MAST SR0 TOP +6	<input type="radio"/>
	Occurrence(s)	:	1 OCCURRENCE(S)	

Section	Page
15.1 Memory allocation map	15/2
15.2 Entering the application comment	15/3
15.3 Selecting a processor	15/4
15.4 Saving and retrieving files	15/5
15.5 Search for the program addresses of a variable using <Alt><X>	15/6
15.5-1 General	15/6
15.5-2 Example of using the <Alt>+<X> command in PROGRAM mode	15/7
15.6 Possibilities open to operators depending on their rights of access	15/8
15.7 Converting a V4 application to V5	15/9
15.8 PL7-3 Help	15/11
15.8-1 Notes on help screens	15/11
15.9 V5 Modular version	15/13
15.9-1 Principle of operation	15/13
15.9-2 Direct access to a basic module	15/13
15.9-3 Labeling an basic module	15/15
15.9-4 Protection of a basic module	15/16
15.9-5 Graphic view and animation of a macro-module	15/18
15.10 List of symbols	15/23

15.1 Memory allocation map

This screen, accessed using the <Alt>+<T> command, gives a detailed presentation of the occupation of the program memory by the standard PL7-3 program space.

The screenshot shows a window titled "MEMORY ALLOCATION" with a purple border. It contains two tables. The left-hand table, titled "STANDARD PROG SPACE", shows volume in words for reserved, used, and free space, broken down by code+graph and system. The right-hand table shows the size of tasks and their code and chart segments.

STANDARD PROG SPACE			
	VOLUME (WORDS)		
	RESERVEI	USED	FREE
CODE+GRAPH	33624	21173	12451
SYSTEM	5880	5880	0
TOTAL	39504	27053	12451

SIZE TASK	CODE		CHART	
	SEG1	SEG2	SEG1	SEG2
IT	9	0	29	0
FAST	712	7	406	862
MAST	7455	2063	6210	572
AUX0	617	8	444	0
AUX1	450	0	224	517
AUX2	218	0	170	0
AUX3	99	0	101	0

Left-hand table

This table indicates the volume reserved and the volume used in CONFIGURATION mode. It is divided into the part reserved for the system and the part used by the code and graphics for the application.

Right-hand table

This table indicates the volume occupied by the code and graphics for the application. For each task, it presents :

- 2 code segments :
 - 1st segment; code objects of the MAIN program,
 - 2nd segment; code objects of the SRs.
- 2 graphic segments :
 - 1st segment; graphic objects for the whole task,
 - 2nd segment; objects for the diagnostic source program.

Special case

When Grafacet is configured, the MAST task code segments are as follows :

- 1st segment; Grafacet interpreter, code objects relating to Grafacet, CHART, XM, SR and MAIN,
- 2nd segment; action and transition condition code objects, PRL and POST.

15.2 Entering the application comment

The application comment appears with the application name in the list of existing xxx.BIN files. For V4 and V5 applications, this information is also displayed in the File options of the TRANSFER tool.

Application comment entry is accessed using the **<Alt>+<K>** command in the CONFIGURATION, PROGRAM, CONSTANT, DATA, DEBUG and DOCUMENT/PRINT modes. Entry is performed in a user entry line and is limited to 24 characters maximum.

15.3 Selecting a processor

Selection of the processor is performed using the XTEL-CONF station tool, which defines the hardware required for the application. To do this, proceed as follows :

- in the Functions window, define the various functions (or applications) required for the application : PL7_3, comm, etc. The corresponding icons will then be displayed.
- open a session of the XTEL-CONF tool.
- pull down the Define menu and select the heading Rack I/O Config.
- double-click on rack 0/1 to open a dialog box which will enable the application to be configured : select the type of rack, processor, power supply and I/O modules.
- select the various modules installed in the rack, including the processor; confirm the configuration then exit the configuration function for in-rack modules.
- pull down the Generate menu and generate the application file.
- quit XTEL-CONF.
- when starting and initializing PL7-3 for the first time, it automatically calls up the configuration defined by the XTEL-CONF tool. The type of processor selected then appears in the status bar. During subsequent load operations, PL7-3 automatically loads the PL7_3.BIN file, which determines the type of processor.

In online mode, the connected PLC, and therefore the processor, is automatically recognized. The type and version ① then appear in the status bar.



15.4 Saving and retrieving files

For this function, use the IMPORT and EXPORT services associated with the PL7_3 function icon :

- with the mouse, click on the PL7_3 icon, which pulls down a menu,
- select Import or Export, depending on the operation to be executed.

Retrieving one or more files

This function uses the IMPORT function which permits :

- the import under PL7-3 of files generated under DOS,
- the import under PL7-3 of V5 files, stored on hard disk (in another station) or on diskette.

To do this the user must :

- progressively define the source address of the file, by ascending the source tree structure by double-clicking in the Directories zone.
- select the file(s) to be retrieved by clicking on the corresponding name(s).

As the destination address is known by the software workshop, only the storage zone APPLI or MOD needs to be selected by the user (see section 2.4).

Confirmation of the selections made copies the selected source files to the target zone.

Saving one or more files

This uses the EXPORT function which permits the transfer, to the exterior tree structure, of a copy of the files generated by the PL7-3 function.

As the source address is known by the software workshop, the user must select :

- the zone containing the files (APPLI or MOD) by double-clicking on the zone,
- the file(s) to be saved by clicking on the corresponding names.

The destination address is selected by ascending the destination tree structure by double-clicking on the various directories.

The selected files are stored by confirming the choices made.

Warning

In order to receive the files, the diskette must be formatted and not write-protected.

15.5 Search for the program addresses of a variable using <Alt><X>

15.5-1 General

<ALT>+<X> is used to access the cross-references of a variable. When a LADDER rung (1) or LITERAL statement (2) is displayed in the PROGRAM or DEBUG modes, this function permits the :

- display of all the variables of the selected program element (standard PL7-3 variables, SRI, elements extracted from OFBs),
- selection of a variable from those displayed and then display of all its program addresses (cross-references of the variable).
- selection of a program address from those displayed and then display of the corresponding program element.

15.5-2 Example of using the <Alt>+<X> command in PROGRAM mode

• Display of the program element

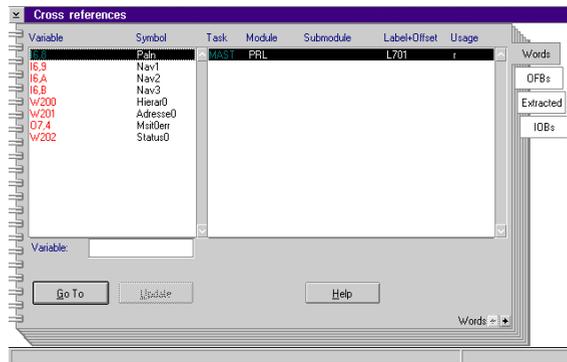
The user positions the cursor on the rung containing the variable to be analyzed. Pressing <Alt>+<X> then displays the cross-references window, and generates references.

The **Words**, **OFBs**, **Extracts** and **IOBs** tabs are used to access standard variables, OFB elements, bits extracted from OFB elements and objects associated with remote I/O respectively.

The window consists of two screens :

- the left-hand screen displays all the standard PL7-3 variables of the selected element in SYMBOL or VARIABLE mode. A dialog box is used to enter a PL7-3 object, as either a variable or a symbol, so as to display the cross-references.
- the right-hand screen gives the cross-references for these variables by supplying the program address (task, module, submodule, label and movement) of each occurrence of these references in the application.

The USE column gives the usage of the variable in the program (see section 13 - Description of listings).



Go to accesses PROGRAM mode using the program address selected in the right-hand list.

Update used to calculate new cross-references.

Caution : this function cannot be accessed when the cross-references (<Alt>+<X>) are called from a Grafcet zoom.

<HELP> calls up the program help function.

<TAB> used to move within the screen.

<→><←>

• Special case where the program address is a Grafcet address

In the case of a Grafcet address, the system does not automatically «ZOOM» in on the submodule indicated, but selects the Grafcet page concerned and places the cursor on the step indicated on the left-hand side of the submodule column.

15.6 Possibilities open to operators depending on their rights of access

Program Max.	_____				
Program min.	_____				
Adjust Max.	_____				
Adjust min.	_____				
Operate Max.	_____				
Operate min.	_____				
	↓	↓	↓	↓	↓
Monitoring the PLC (RUN/STOP/INIT)			yes	yes	yes
Access to TRANSFER mode	yes	yes	yes	yes	yes
Transfer to a connected PLC			yes	yes	yes
Access to PROGRAM mode (display)			yes	yes	yes
Modification of the program, PLC in STOP				yes	yes
Modification of the program, PLC in RUN					yes
Access to CONSTANT mode (display)	yes	yes	yes	yes	yes
Modification of the constants (CW + internal OFB constants), PLC in STOP		yes	yes	yes	yes
Modification of the constants (CW + internal OFB constants), PLC in RUN			yes	yes	yes
Access to CONFIGURATION mode (display)	yes	yes	yes	yes	yes
Modification of the configuration in offline mode				yes	yes
Modification of the configuration in online mode, PLC in STOP	yes	yes	yes	yes	yes
Modification of the configuration in online mode, PLC in RUN			yes	yes	yes
Access to DEBUG mode				yes	yes
Access to DATA mode				yes	yes
Access to DOCUMENT/PRINT mode			yes	yes	yes
Access to SEARCH/REPLACE mode			yes ⁽¹⁾	yes	yes
Replace, PLC in STOP				yes	yes
Replace, PLC in RUN					yes

⁽¹⁾ only in SEARCH mode

15.7 Converting a V4 application to V5

All V4 applications can be converted to V5 applications, provided the following operations are performed :

- 1 Retrieve the V4 application under X-TEL V5, by **Store/Retrieve** or **Copy/Paste**.
- 2 Create a V5 host station.
- 3 Start in a PL73V6V5 utility session (at the prompt), so as to return to PL7-3 V5 for V5 stations. From this point onwards, starting a PL7-3 session in a V5 station will activate PL7-3 V5. In fact converting a V4 application to V5 requires PL7-3 V5.
- 4 Start from the PL7_3 icon of the V5 host station, the **Import** function and import the following files :

StationV4\PL7_3\APPL\xxx.BIN (compulsory) : application binary,
 StationV4\PL7_3\APPL\xxx.COM (optional) : program comments,
 to the StationV5\PL7_3\APPLI directory,

then

StationV4\PL7_3\MOD\xxx.SCY (optional) : table of V4 symbols,
 StationV4\PL7_3\MOD\xxx.MOD (optional) : table of modules,
 StationV4\PL7_3\MOD\xxx.COM (optional) : module comments,
 StationV4\PL7_3\MOD\xxx.CST (optional) : constant words source,
 to the StationV5\PL7_3\MOD directory.

- 5 Start PL7-3 in the V5 station and perform the following operations :
 - reply "No" to starting XTEL-CONF,
 - access the retrieve screen by the command [RETRIEVE],
 - activate the command [STR.NAME] and enter the name of the application file to be retrieved : enter the name of the xxx.BIN file, previously imported under the V5 host station,
 - retrieve using the [RETRIEVE] command,
 - return to the operating mode selection screen (CLEAR),
 - access the store screen using the [STORE] command,
 - activate the [STR.NAME] command and enter the name of the files to be stored on disk :
 - PL7_3 for the application file (.BIN),
 - STATION for the I/O configuration file (.IOC),
 - store using the [STORE] command, which generates the following files :
 - StationV5\PL7_3\APPL\PL7_3.BIN,
 - StationV5\APP\STATION.IOC,
 - quit PL7-3.

-
- 6 Start the CONF station tool and perform the following operations :
 - from the **File** menu, activate the **CASC to CONF** item (retrieves the V4 station configuration contained in the STATION.IOC file). Reply "Yes" to the question posed,
 - in the **Define** menu, activate the **Rack I/O Config.** item and, if necessary, modify the V5 processor proposed by default. In a case where the initial V4 application comprises 24-channel and/or 32-channel modules, these modules must be redefined, with their exact reference.
 - in the **Generate** menu, activate the **And Enter Task Periods** item and configure the period of the various tasks,
 - in the **Generate** menu activate the **And Enter Application Parameters** item and if required modify the memory configuration,
 - quit the CONF tool.
 - 7 Start the SDBASE station tool and perform the following operations :
 - combine the xxx.SCY symbol file in the database. To do this, activate the **Scy ->Sdbase** item in the **Merge** menu,
 - quit the SDBASE tool,
 - 8 Start PL7-3 in the V5 station and perform the following operations :
 - access the retrieve screen by the [RETRIEVE] command,
 - activate the [STR.NAME] command and enter the name of the application file to be restored : enter the name of the xxx.BIN file, previously imported under the V5 host station,
 - retrieve using the [RETRIEVE] command. This operation enables the xxx.TIT, xxx.DES and xxx.CRT files to be automatically renamed as PL7_3.TIT, PL7_3.DES and PL7_3.CRT,
 - activate the [V5 CONF] command to associate the application program to the configuration defined under XTEL-CONF,
 - reconfigure the application : RECONFIGURATION DIAGNOSTICS function then RECONFIGURATION function.
 - 9 Store the new application on disk using the [STORE] command.
 - 10 Start in a PL73V6V5 utility session (at the prompt), so as to return to a PL7-3 V6 for V5 stations.

15.8 PL7-3 Help

PL7-3 Help uses QuickHelp, to associate a help screen with each PL7-3 language screen. For more information on QuickHelp, refer to the software workshop documentation.

When the Help function is accessed from a PL7-3 screen, the associated help screen is displayed. It is then possible to access all the help screens, using key words or the pull-down menus.

	<F1>	accesses PL7-3 Help function. The help screen associated with the current PL7-3 screen is displayed.
	<X>	used to return to the current PL7-3 screen.

15.8-1 Notes on help screens

Each help screen comprises three main parts :

- a zone used for displaying the explanatory text for the associated PL7-3 screen,
- the action bar, at the top of the screen, which is used to access the pull-down menus,
- vertical scroll bar, on the right of the screen, for scrolling through the text.

Action bar

Each of the 6 items on the action bar accesses a pull-down menu. An item can be selected using the mouse, or using the keyboard by entering the first letter (for example F for File).

File : used to print a help screen, quit PL7-3 Help, etc

View : used to search for a screen, access the previous and next screens, view the screen sequences, etc

Categories : used to access the list of screens. Selection of a screen in the list displays this screen.

References : used to give the list of key words of the help screen displayed. Selection of a key word accesses the corresponding screen.

Paste : used to store the current window or current screen in the paste file (paste.qh by default).

Options : used to modify the size of the window as well as the paste mode (information stored by the paste command is inserted or overwritten).

Note

Help on the keys can be found in the "Special keys" help item.

Text zone

A PL7-3 help screen always starts with a title which defines it (in green underlined in white on a color screen). This title is as near as possible to the name of the screen, which appears in a box in the lower part of the screen.

The rest of the text zone is divided into regions, each starting with a sub-title (in yellow on a color screen). Each region deals with a different subject : general description, possible action and maybe divided into sub-regions (shown by a yellow asterisk on a color screen).

The key words, which permit direct access to a corresponding help screen, are highlighted (in white on a color screen). The words shown in green on a color screen (apart from the titles) are the important words in the text.

Scroll bar

As its name suggests, this is used to scroll up and down the text on screen using the mouse. The arrows at the top and bottom of this bar can be used for the same purpose.

15.9 V5 Modular version

15.9-1 Principle of operation

From the sources generated under PL7-3, XTEL-MOD enables the creation of basic modules and macro-modules. Each basic module, created from a different source file, comprises at maximum a program code file in Grafcet, in Literal or in Ladder language, a file of constants, a file of symbols and a comments file. The macro-modules comprise a set of basic modules.

PL7-3 enables basic modules and macro-modules created under XTEL-MOD to be retrieved and integrated into the application.

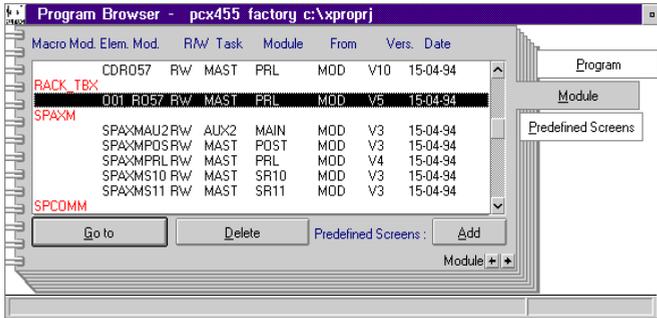
PL7-3 allows the following functions :

- direct access to the first line of code of a basic module,
- labeling of a basic module in the program and the documentation : name of the associated macro-module, name of the basic module, module protection,
- write-protection of a basic module,
- graphic view and animation of a macro-module,
- search/replace in a basic module or macro-module.

15.9-2 Direct access to a basic module

On reading basic modules using the "READ XTEL-MOD/PL7-PMS2" function in the **Open** option in the **File** screen, PL7-3 stores their installation addresses. Once this is done, the PL7-3 program can be accessed from the PROGRAM BROWSER screen and from a table containing all the macro-modules and basic modules for the application. This table can itself be accessed :

- from the PROGRAM BROWSER screen in Program mode and the main **Module** tab
- from the SEARCH/REPLACE mode screen, via the secondary **Search MOD** tab.



<↓> <↑>

used to select a basic module from the list.

Go to

accesses the first line of code in the program, of the selected basic module (the program is animated in online mode).

Delete

deletes the selected module after confirmation.

Add

used to add a fast access action to this program element in **Debug**.

<TAB>

used to move from one group to the other.

<↓> <↑>

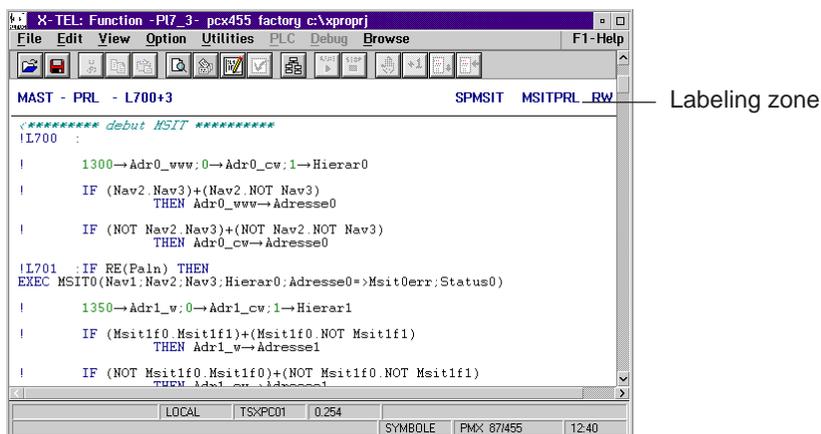
used to move within a group.

<→> <←>

15.9-3 Labeling a basic module

Labeling of basic modules only applies to program modules. Constants, symbols and program comments are not labeled. Labeling consists of displaying certain information, to show that a program line is part of a basic module. This information : name of the macro-module associated with the basic module if one exists (8 characters), name of the basic module (8 characters) and protection linked to the module (2 characters), is displayed in the "application name" field in the Program, Debug and Search/Replace screens (in manual mode only).

In Document/Print mode, each basic module is preceded by a line which indicates the start of the module and labels it. At the end of the module, a new line indicates the end and also labels the module. This information appears on screen as well as on paper when the documentation file is printed.



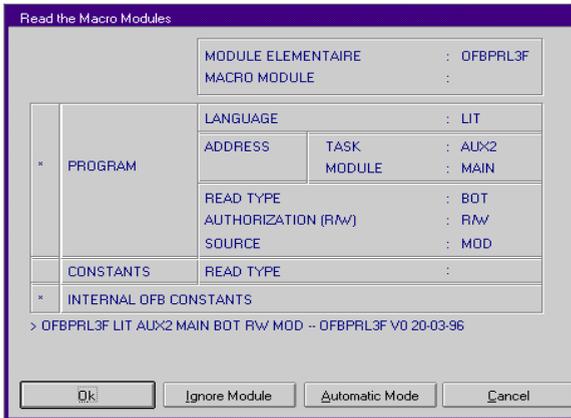
Note

If no macro-module is associated with a basic module, the name of the macro-module is replaced by the characters "." (period).

15.9-4 Protection of a basic module

A basic module can be write-protected when its use is defined under XTEL-MOD. Inserting such a module in the application program, by the PL7-3 **Open** option in the **File** menu, maintains all the functions associated with the module which appear in all the V5 modular version screens including protection.

- RW : the module is set to read and write,
- R : the module is set for read only access (write-protected).



Consequences of write-protecting a basic module, are as follows :

- it is impossible to modify or erase such a module while it is being inserted in the application (Open). If a problem occurs during this operation, it can only be abandoned : **<ESCAPE>** key.
- if a new configuration deletes a write-protected basic module, there will be a reconfiguration error.
- in PROGRAM mode, it is not possible to modify a code line of a write-protected module :
 - the name of the variables or constants cannot be modified, even within a function block (MODIFY mode),
 - a line of code cannot be inserted or deleted (**<INSERT>** and **<DELETE>**),
 - modifications cannot be made in ZOOM mode,
 - windows cannot be moved.

-
- in PROGRAM mode it is not possible to erase write-protected basic modules which are "attached" to a macro-module, nor the program modules containing a write-protected basic module. However, using the **<DELETE>** command in the PROGRAM BROWSER screen (**Module** tab), it is possible to erase :
 - the macro-modules and thus at the same time all the basic modules (write-protected or not) which are associated with them,
 - the basic modules (write-protected or not) which are not associated with a macro-module.

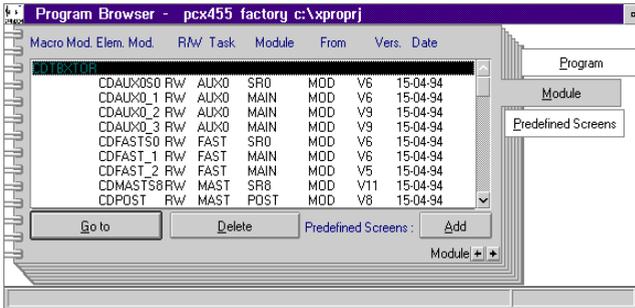
Using the **<DELETE>** command in the PROGRAM BROWSER screen, it is possible to erase :

- the program modules not containing write-protected basic modules.
- in SEARCH/REPLACE mode, replacement is refused each time it is requested on a part of the program belonging to a write-protected module.
- in DEBUG mode, the write-protected code cannot be modified. However, the values of variables can be modified.

OK	enables the module to be inserted into the program.
Ignore	does not insert the current module and moves on to the next module.
Auto Mode	used to insert all the selected modules into the program without confirmation being requested.
Abort	used to interrupt the module insertion process.

15.9-5 Graphic view and animation of a macro-module

Under XTEL-MOD, a graphic representation (vignette) can be associated with each macro-module as a block containing input variables, internal variables and output variables. These variables are freely defined by the user from the referenced variables in the basic modules which make up the macro-module and cannot be modified in PL7-3. The PROGRAM BROWSER screen, **Module** tab is used to select a macro-module and to access it using <ENTER> or **Go to**.



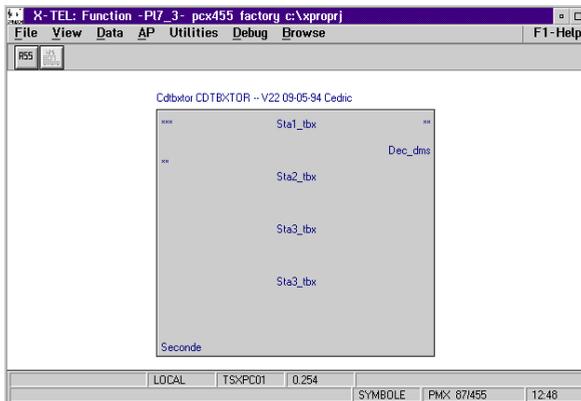
Go to used to select and access a macro-module :

- with a graphic view of the macro-module, in Program mode,
- with graphic animation of the macro-module, in online mode.

Delete deletes the macro-module or the selected module.

If the macro-module does not have an associated graphic (optional operation under XTEL-MOD), the message "18060 - NO VIGNETTE FILE" is displayed.

Graphic view of a macro-module



The first line of this screen indicates the name of the "father" module (from XTEL-MOD), the name of the macro-module and any comment.

View, Variable	<ALT>+<V>	displays the objects in the form of variables.
View, Symbol	<ALT>+<S>	displays the objects in the form of mnemonic symbols.
View, Associated Name	<ALT>+<A>	displays the objects in the form of associated names (defined under XTEL-MOD). Objects which have no associated name continue to be displayed as variables.

Note

If an object is defined in the graphic representation (vignette) associated with a macro-module, but is not configured in the application, the graphic view of the macro-module in VARIABLE mode displays the character string "****" in place of the object. However, in ASSOCIATED NAME mode, the object is displayed normally, in order to allow the user to know the name of the variable which has not been configured.

MENU BAR

FILE MENU :

Close	used to exit the vignette.
--------------	----------------------------

VIEW MENU :

Variables	displays the objects in the form of variables.
Symbols	displays the objects in the form of symbols.
Associated Name	displays the objects in the form of associated names.

TOOLBAR

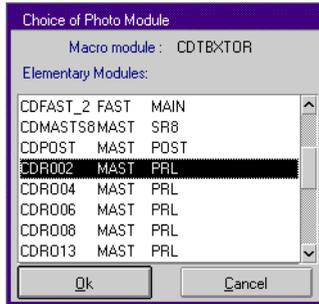
This toolbar provides quicker access to the main functions in the menu bar.



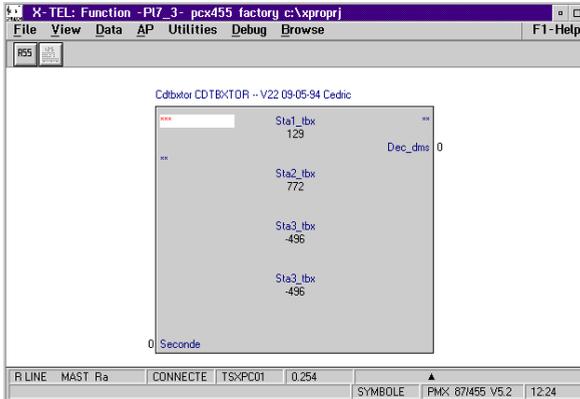
- └ Selects the display base
- └ Displays the name associated with each variable

Graphic animation of a macro-module

Animating the graphic view is like taking a "photograph" of the input, internal and output variables, at a precise moment and displaying the values. The "photograph" of the values is achieved in association with a basic module, which can be selected either under XTEL-MOD (Select Photo Module), or selected by the SELECT PHOTO MODULE window which appears when selecting the macro-module.



- if the selected basic module (for example POST) is a Literal or Ladder module, the "photo" is taken after the last program element of the module has been executed,
- if the selected module is a Grafcet module, the "photo" is taken at the start of the master task scan.



The first line on this screen indicates the name of the "father" module (from XTEL-MOD), the name of the macro-module and any comment.

View, Variable	<ALT>+<V>	displays the objects in the form of variables.
View, Symbol	<ALT>+<S>	displays the objects in the form of symbols.
View, Associated Name	<ALT>+<A>	displays the objects in the form of associated names (defined under XTEL-MOD). Objects which have no associated name continue to be displayed in the form of variables.
	<↑><↓>	position the cursor on the variables in the column. The selected variable appears in red.
	<→> <←>	used to change column.
Data, Base		used to modify the display base of the variable selected by the cursor : decimal, hexadecimal, binary coded decimal, message (ASCII) or floating point (PMX only).

Note

If an object is defined in the graphic representation associated with a macro-module, but is not configured in the application, the graphic animation of the macro-module in VARIABLE mode displays the character string "****" in place of the object. If this object is configured, but cannot be animated, it is the character string "****" which is displayed, in VARIABLE mode, in place of the object.

In all cases, in ASSOCIATED NAME mode, the object is displayed normally, in order to allow the user to know the name of the variable which has not been configured.

MENU BAR**FILE MENU :**

Close	used to exit the vignette.
--------------	----------------------------

VIEW MENU :

Variables	displays the objects in the form of variables.
Symbols	displays the objects in the form of symbols.
Associated Name	displays the objects in the form of associated names.

DATA MENU :

Base	used to specify the base in which the object is displayed.
-------------	------------------------------------------------------------

PROCESSOR MENU :

Run	starts program execution.
Stop	stops program execution.

DEBUG MENU :

Stop Task	interrupts the task displayed.
Task Monitored	selects the task to be monitored.
Breakpoints → Insert → Remove → View	not possible in VIGNETTE mode. not possible in VIGNETTE mode. displays the position where the breakpoint was defined.
Continue	not possible in VIGNETTE mode.
Step by step	not possible in VIGNETTE mode.
Last Stop	displays the last element executed.
Data Mode	accesses DATA mode.
Data Line mode	accesses DATA LINE mode.

TOOLBAR

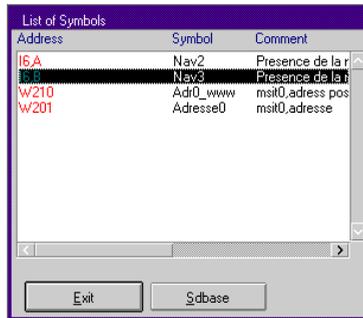
This toolbar provides quicker access to the main functions in the menu bar.



- └ Selects the display base
- └ Displays the name associated with each variable

15.10 List of symbols

This screen can be accessed using the **<Ctrl>+<F5>** command, when the program is displayed in Ladder or Literal language. It displays the list of symbols for variables contained in the selected program element (Ladder rung or Literal statement).



Sdbase
Quit

accesses the XTEL-SDBASE tool symbol editor.
used to return to the program display screen.



Avertissement

Mise à jour PACK V6

TXTLUXTLV6

A l'issue de l'installation du PACK V6 (version disquettes ou version CD ROM); veuillez procéder à l'installation du logiciel de mise à jour ci-joint.

Procédure :

- Ouvrir une fenêtre OS/2
- Insérer la disquette n°1 TXTLUXTLV6 dans le lecteur
- Saisir la commande <A:> <Entrée> puis <Setup> <Entrée>
- Suivre les instructions visualisées à l'écran, jusqu'à l'écran "Type d'installation",
=> Sélectionner "Installation Standard" si vous désirez mettre à jour automatiquement votre poste,
=> Sélectionner "Installation Personnalisée" si vous désirez sélectionner les produits et les logiciels à mettre à jour.
- Poursuivre l'installation, en suivant les instructions visualisées à l'écran. (Pour plus de détails sur cette installation, se reporter à la fiche d'accompagnement du PACK V6).

Attention :

En cas de réinstallation du PACK V6, il faut impérativement installer à nouveau ce logiciel de mise à jour TXTLUXTLV6.

Warning

PACK V6 update

TXTLUXTLV6

After having installed the PACK V6 (diskettes or CD ROM); please proceed to installation of update software attached :

Update installation :

- Open a full screen OS/2 session
- Insert the disk no.1 TXTLUXTLV6 in the drive
- Enter commmands <A:> <Enter> then <Setup> <Enter>
- Follow the instruction displayed on the screen until the "Type of Installation" screen,
=> Select "Standart Installation" to update your PC automatically,
=> Select "Customized Installation" to update the products and programs manually.
- Follow the instructions displayed on the screen. (For more details refer to the PACK V6 Information Sheet).

Remark :

In the case of reinstalling the PACK V6, it is imperative that you reinstall the update program TXTLUXTLV6.

