



- 1 Introduction **A**
  - 2 Installation
  - 3 Using PL7-3 Software
- 

- 4 Operation with Terminal Connected
  - 5 Control Mode **B**
  - 6 Configuration Mode
- 

- 7 Program Mode : Grafcet Language **C**
- 

- 8 Program Mode : Ladder Language **D**
- 

- 9 Program Mode : Literal Language **E**
- 

- 10 Constant Mode
  - 11 Adjust Mode **F**
  - 12 Data Mode
  - 13 Debug Mode
- 

- 14 Transfer Mode
  - 15 Document / Print Mode
  - 16 Search / Replace Mode **G**
  - 17 Terminal Mode
- 

- 18 Appendices **H**

---

## About this document

---

PL7-3 V52 is an upgrade of the V5 program, and like V5 is marketed as three separate products :

- PL7-3 which offers Grafcet, Ladder and Literal language. The reference for this product is TXT L PL7 3 V5E
- PL7-3 GLD which offers Grafcet and Ladder language. The reference for this product is TXT L PL7 3 D V5E
- PL7-3 GLT which offers Grafcet and Literal language. The reference for this product is TXT L PL7 3 T V5E.

This document is common to all three of the above-mentioned products and incorporates PL7-3 versions 3, 4 and 5, since PL7-3 V5 can be used to create and modify applications for levels V2, V3, V4 (V41, V42 and V43) and V5 (V5 and V52).

- On a V2 or V3 station, PL7-3 governs the whole application
- 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-mounted I/O, objects, OFBs, etc.)
- On a V5 station, the XTEL-CONF tool is used to configure rack-mounted I/O, distributed I/O and tasks, and generates the application (.APP). PL7-3 is the means of configuring the other software resources (objects, OFBs, etc.).

Where a section does not apply to all three of the above-mentioned products, the section begins with an explanation of which products are concerned : PL7-3 for PL7-3 V52, PL7-3 GLD and/or PL7-3 GLT.

In addition to the restrictions defined at the beginning of the section, reference notes at the bottom of the page are used to identify paragraphs or descriptions which only apply to one product or which exclude one. For example (1) *except PL7-3* for a description which excludes PL7-3 V52 or (2) *PL7-3 GLT only*.

Information which only applies to the use of PL7-3 V52 with a station of level V4 or earlier is shown in grey.

---

## **New features of PL7-3 V52 in relation to previous versions**

---

The main enhancements to the TXT L PL7 3 V52E software in relation to TXT L PL7 3 V5E are as follows :

### **Increase in number of Grafcet steps:**

The number of Grafcet steps which it is possible to configure has increased from 512 to **1024** steps.

- **Comment** : if for an application, the number of steps declared or the number of steps which are active is simultaneously greater than 512, then following reconfiguration, it becomes a level V5.5 application and can no longer be accessed by PL7-3 V5 software.

To retain the application at level V5.0, the user must configure the number of steps declared or active simultaneously to a value which is less than or equal to 512.

### **Development of cross reference services :**

Regardless of the application level (V2, V3, V4 or V5), the RETRIEVE function of the PL7-3 V52 software, suggests reading the cross references (XRi files). The system ensures that the XRi files read are compatible with the current application.

Once the table of cross references has been created (by calculation or by reading the XRi files) it is no longer suggested that this be recalculated as long as the program has not been modified in any of the screens concerned (Alt-X, Documentation, STORE, etc.).

### **Display of values in floating point (CDW, DW, DW element of OFB) :**

All variables declared as floating type in sdbase (with the letter F in the field 'Visualize') are displayed in floating point format in PL7-3 screens.

The floating type OFB elements are automatically displayed in floating point in PL7-3, even if they have no mnemonic symbol in sdbase or if they are declared as non floating (without the letter F).

### **Development of copy/paste on the Alt-C screen :**

The commands PUT COM and GET COM in the ENTER PROGRAM COMMENT screen enable comments to be stored/retrieved from one screen to another.

### **Abolition of the size limit of code or graph segments :**

This new function is useful in the event of overflow of a code or graph segment in the PL7-3 application. Restrictions do exist however in certain memory overflow situations (refer to section 6.3-3).

To eliminate the application overflow, the user can use an automatic function offered in the segment configuration screen. It is also possible to modify this configuration manually.

**To remove any segment restriction :** for example, overflow of graphic information segments or overflow of the first or second code segment with Grafcet charts, PL7-3 V52 offers the possibility of automatically or manually transferring the code or graph of the saturated module, from one segment to another.

A level V5.0 application, containing transferred modules, becomes a level V5.5 application after reconfiguration and can no longer be accessed by PL7-3 V5 software. This operation is not irreversible. In fact, if the user configures the segments with their initial value, and then reconfigures the application, it returns to being a V5.0 application.

**Reading internal OFB constants :**

This function is optional ([OBC] key). It is used to take account of the OFB constants as part of an elementary module (XTEL-MOD).

If the OBC option is selected, by creating (WRITE) in PL7-3 a program source file (.LIT, .LAD or .GR7) an .OBM file containing the names of the internal OFB constants in the program module is automatically created.

When XTEL-MOD automatic reading occurs (RETRIEVE screen), the OFB constants, if any, are taken into account.

**Direct access to SRis or XMis, from the program address selection screen :**

The program address selection screen offers an additional column from which a SRi or XMi module may be directly accessed (without entering the number).

---

## Differences between PL7-3 GLD or PL7-3 GLT and PL7-3 V52

---

Enhancements to the TXT L PL7 3 D V52E and TXT L PL7 3 T V52E programs in relation to TXT L PL7 3 V52E refer to operational limitations designed to improve ease of use.

### Limitation on the number of languages :

- TXT L PL7 3 D V52E only offers 2 languages : Grafcet and Ladder, for programming the various modules, actions and transition conditions
- TXT L PL7 3 T V52E only offers 2 languages : Grafcet and Literal, for programming the various modules, actions and transition conditions.

### No remote programming :

The only PLC addresses which are accepted are the local station or its UNI-TELWAY extension. In other words, the only possible means of access are local mode (with the terminal connected to the PLC programming port) or via the UNI-TELWAY bus (with the terminal and the PLC connected on the same UNI-TELWAY bus).

---

## Software compatibility

---

For a level V5 application, on a level V5 station, PL7-3 V52 **is compatible** with the software :

X-TEL V5 and X-TEL V52

MINI X-TEL V5 and MINI X-TEL V52

XTEL-MOD V5 and XTEL-MOD V52

For a level V2, V3 or V4 application, on a level V3 or V4 station, PL7-3 V52 **is compatible** with the software :

XTEL-MOD V41

XTEL-MOD V42

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

X-TEL V42

MINI X-TEL V43

It is **not possible** for more than one version of PL7-3 to **co-exist** on the same workstation.

Once a new version of PL7-3 is installed, the version which was already installed on the station is no longer accessible. The application files which already exist on the station are of course retained.

---

## Compatibility between PL7-3 V5.5 and PL7-3 GLD or PL7-3 GLT applications

---

Applications which have been developed using PL7-3 V52, PL7-3 GLD or PL7-3 GLT are compatible, bearing in mind the restrictions imposed by PL7-3 GLD and PL7-3 GLT :

- An application with a xxx.BIN file or source files, developed using PL7-3 GLD or PL7-3 GLT, can be read, modified and documented in PL7-3 V52 without any limitations
- If a binary application file (xxx.BIN) generated in PL7-3 V5, then retrieved in PL7-3 GLD or PL7-3 GLT, includes modules programmed in a language which is not accessible, only operating modes Control, Transfer and Adjust can be used
- Source files (xxx.GR7, xxx.LIT, etc.) generated in PL7-3 V52 can be read by PL7-3 GLD or PL7-3 GLT, bearing in mind certain language restrictions :
  - xxx.LIT files can only be read by PL7-3 GLT
  - xxx.LAD files can only be read by PL7-3 GLD
  - Grafcet source files (xxx.GR7) can be read by any version of PL7-3. Actions and transition conditions programmed in a language which is not accessible are ignored and signalled by an error message.

It is therefore possible to start to develop an application with PL7-3 GLD or PL7-3 GLT and to follow its development with PL7-3 V52, to benefit from the additional services. The reverse is also true, as it is possible to start developing an application with PL7-3 V52, and continue with PL7-3 GLD or PL7-3 GLT, taking the available functions into account.

---

## General note for this document

---

The term **program element** is used generically to refer to either a Ladder network or a Literal statement.

---



<b>Section</b>	<b>Page</b>
<b>1 Introduction</b>	
Contents	1/1
1.1 General	
1.2 Configuration Required for PL7-3 Software	
<b>2 Installation</b>	
Contents	2/1
2.1 Hardware Check	
2.2 Connections	
2.3 Software Installation	
2.4 PL7-3 Files	
<b>3 Using PL7-3 Software</b>	
Contents	3/1
3.1 Using the Keyboard	
3.2 Using the Mouse	
3.3 Using the Display Screen	
3.4 Accessing PL7-3 Software	
3.5 Operating Mode Selection	
3.6 Presentation of the Operating Modes	
3.7 Storage and Retrieval of PL7-3 applications	
3.8 Calling the XTEL-MEM Tool from PL7-3	
3.9 Using the Symbol Database	
3.10 Quitting the PL7-3 Function	







<b>Sub-section</b>	<b>Page</b>
<b>1.1 General</b>	1/2
1.1-1 Functions offered by PL7-3 software	1/3
<b>1.2 Configuration required for PL7-3 software</b>	1/6
This section ends on page	1/6

---

## 1.1 General

---

This manual describes the installation and use of PL7-3 software on terminals equipped with the OS/2 operating system and the X-TEL or MINI-XTEL software workshop :

- Telemecanique's FTX 417-20 terminal
- Telemecanique's FTX 507 terminal
- An IBM PS/2 or compatible computer equipped with a 80286, 80386 or 80486 processor.

On these terminals, PL7-3 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 versions of the software are available :

- **PL7-3 programming and operation : TXT L PL7 3 V5E**

This version can be used to program, debug and operate the above-mentioned sequential PLCs, whether in local mode, in locally connected mode or connected to a network. It offers Grafcet, Ladder and Literal programming languages and can receive the optional function blocks (OFBs) which can be used by TSX and PMX 47-4xx/67-4xx/87-4xx/107-4xx PLCs.

- **PL7-3 GLD programming and operation : TXT L PL7 3 D V5E**

This version can be used to program, debug and operate the above-mentioned sequential PLCs, whether in local mode or in locally connected mode. It offers Grafcet and Ladder programming languages and can receive the optional function blocks which can be used by TSX and PMX PLCs.

- **PL7-3 GLT programming and operation : TXT L PL7 3 T V5E**

This version can be used to program, debug and operate the above-mentioned sequential PLCs, whether in local mode or in locally connected mode. It offers Grafcet and Literal programming languages and can receive the optional function blocks which can be used by TSX and PMX 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 (in Grafcet, Ladder (1) or Literal (2) 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 print-out of the dossier, including :
  - The title page
  - The general information pages
  - The I/O wiring
  - The program comments
  - The network information
  - The cross references
- Storage of the dossier on the hard disk.

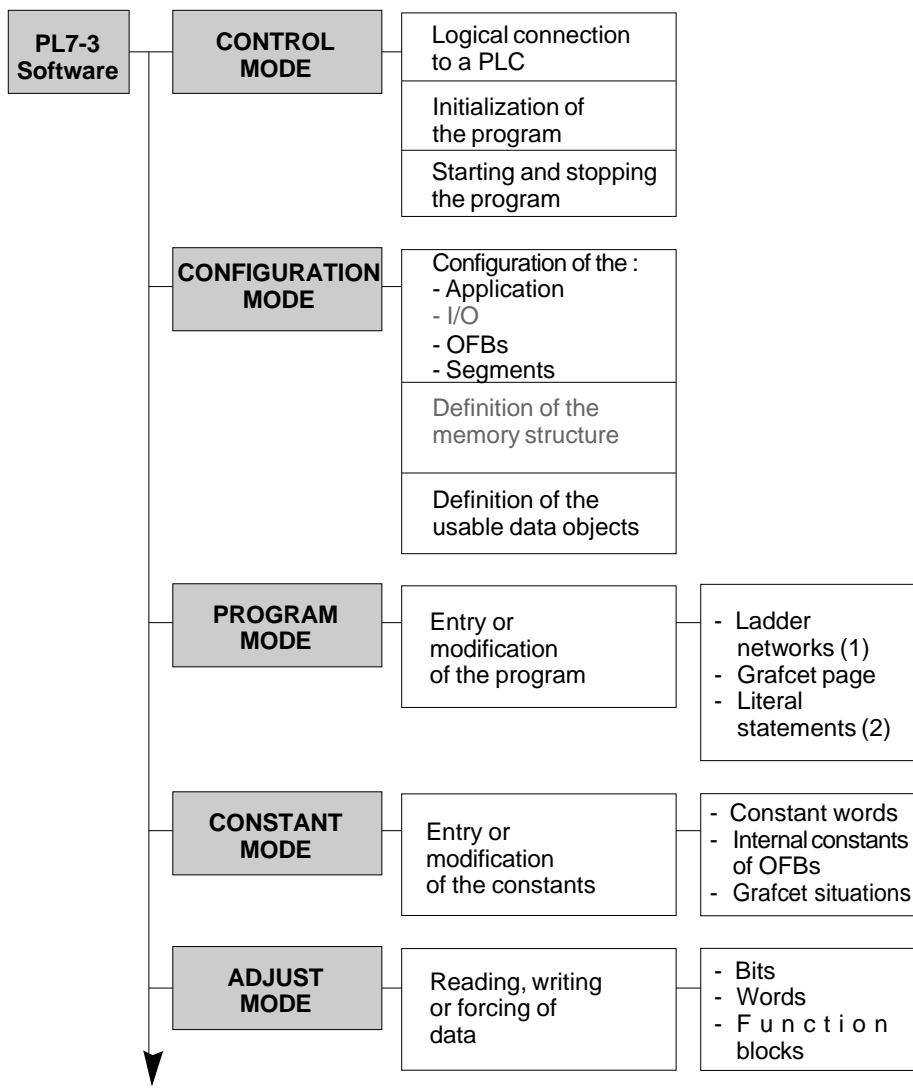
In locally connected mode or when connected via the network (3), it can also be used for :

- On-line programming and program modification in connected mode with the PLC stopped (V2, V3, V4 and V5 processors) or running (V4 and V5 processors)
- Debugging the program
- Adjusting the application parameters.

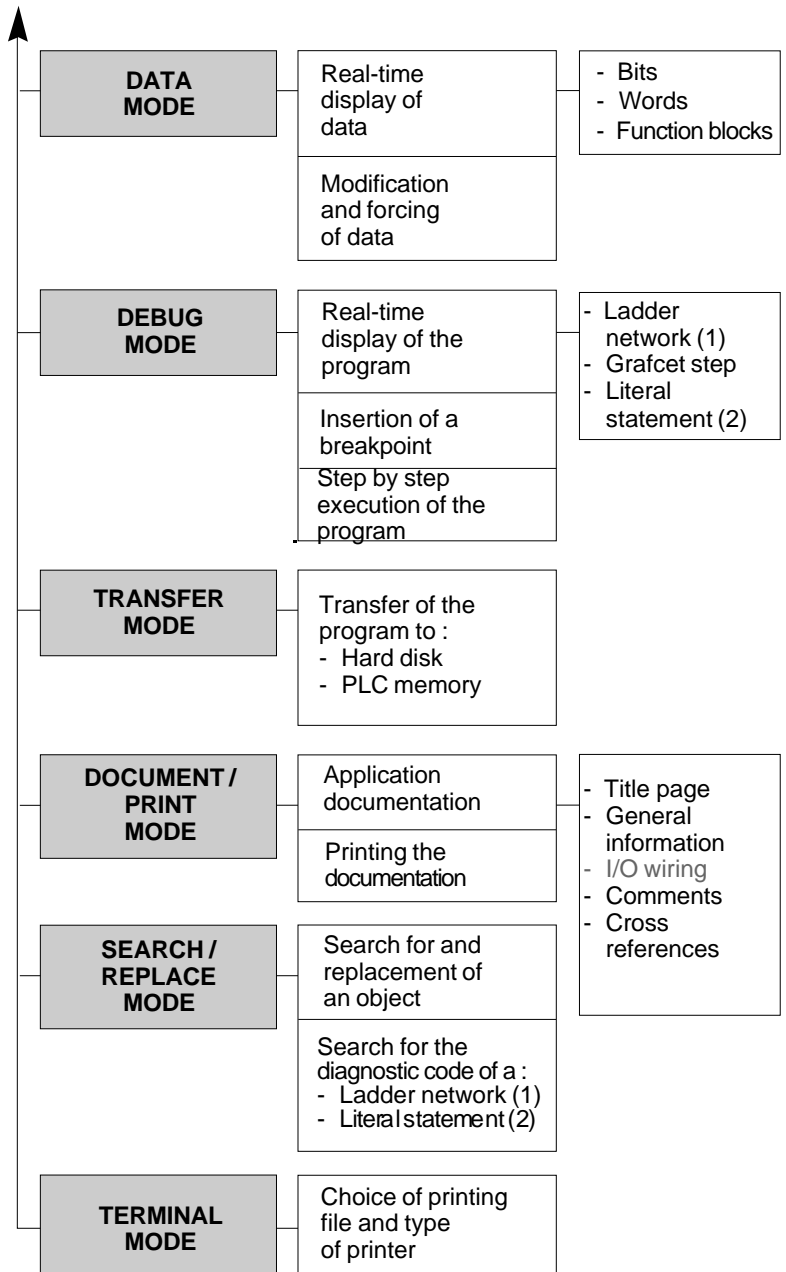
(1) Except PL7-3 GLT

(2) Except PL7-3 GLD

(3) PL7-3 only



(1) Except PL7-3 GLT  
 (2) Except PL7-3 GLD



(1) Except PL7-3 GLT (2) Except PL7-3 GLD (3) PL7-3 only

---

## 1.2 Configuration Required for PL7-3 Software

---

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

- The OS/2 operating system, version 1.2 minimum
- The X-TEL or MINI X-TEL software workshop.

You therefore need one of the following hardware configurations :

- A Telemecanique FTX 417-20 terminal :
  - T FTX 417 20 4 Mb RAM, 40 Mb hard disk
- A Telemecanique FTX 507 terminal :
  - T FTX 507 30 4Mb RAM, 40 Mb hard disk
  - T FTX 507 50 4Mb RAM, 120 Mb hard disk
- An IBM PS/2 or compatible PC with :
  - 80286 microprocessor minimum (80386 recommended)
  - 4Mb RAM minimum (6 Mb recommended)
  - 40Mb hard disk
  - An EGA or VGA high definition colour monitor
  - A national or international keyboard
  - A mouse (recommended)



<b>Sub-section</b>	<b>Page</b>
<b>2.1 Hardware check</b>	<b>2/2</b>
<b>2.2 Connections</b>	<b>2/4</b>
2.2-1 Installing the software key	2/4
2.2-2 Connection to main power source	2/4
2.2-3 Connecting the FTX 417-20 terminal to the PLC	2/5
2.2-4 Connecting the FTX 507 terminal to the PLC	2/5
2.2-5 Connecting the IBM PS/2 or compatible PC to the PLC	2/6
<b>2.3 Software Installation</b>	<b>2/8</b>
2.3-1 Preliminary steps	2/8
2.3-2 Installation procedure	2/8
<b>2.4 PL7-3 Files</b>	<b>2/10</b>
This section ends on page	2/10



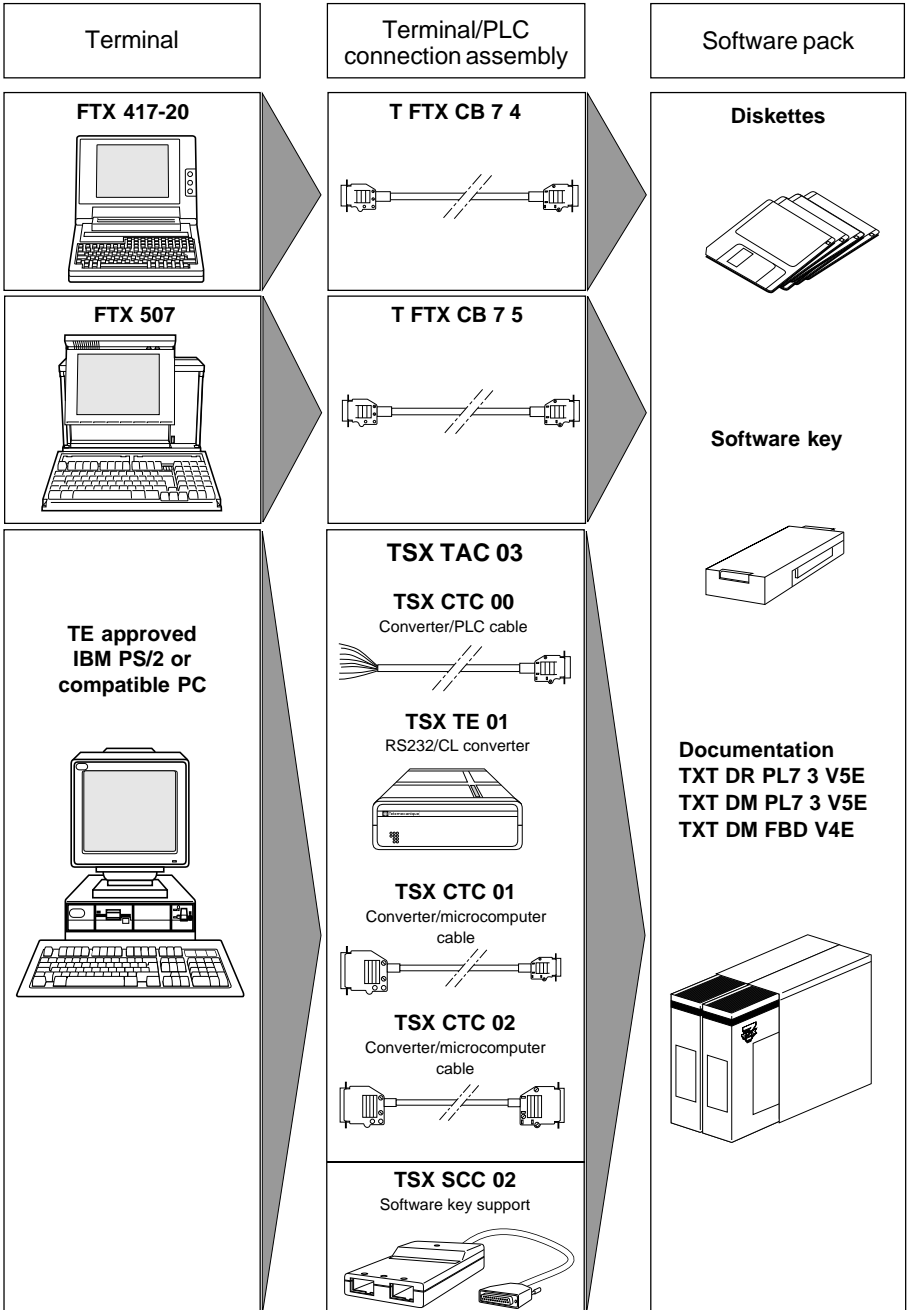
---

## 2.1 Hardware Check

---

To use PL7-3 software, you need the following hardware :

- An FTX 417-20, an FTX 507 terminal or an IBM PS/2 or compatible PC (see Section 1.2 for required configuration), equipped with the OS/2 operating system and the X-TEL or MINI X-TEL software workshop.
- A terminal / PLC connecting cable, in the case of an FTX 417-20 or FTX 507 terminal.
- A terminal / PLC connection assembly in the case of an IBM PS/2 or compatible PC, composed of :
  - An RS 232C / current loop converter
  - A converter / micro-computer connecting cable with a 9-pin connector
  - A converter / micro-computer connecting cable with a 25-pin connector
  - A converter / PLC connecting cable.
- A PL7-3 software pack comprising :
  - Four 3 1/2" diskettes
  - A software key
  - The PL7-3 V5 reference manual TXT DR PL7 3 V5E
  - The OFB diagnostic manual TXT DM FBD V4E
  - This manual TXT DM PL7 3 V5E.



---

## 2.2 Connections

---

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

- The connection of the terminal to the main power source.
- The connection of the terminal to the PLC.

---

### 2.2-1 Installing the software key

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

**This operation must be done with the terminal powered down.**

#### Note

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

The Key Manager delivered with the software workshop allows this right of use to be transferred to the master key. This makes the PL7-3 software key unusable and it can therefore be removed, thus freeing the corresponding slot on the software key support.

---

### 2.2-2 Connection to the main power source

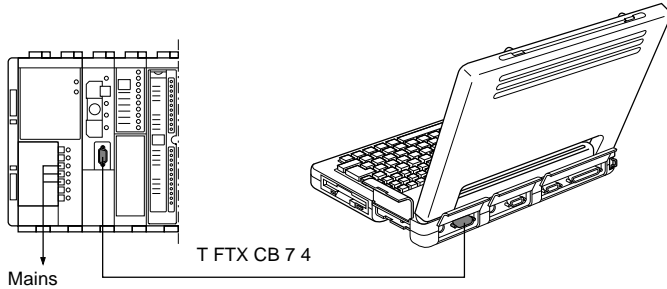
Before making any connections to the main power source, check that each of the elements of the system is correctly adjusted to operate on your mains power voltage (220V or 110V).

Then, observing the precautions indicated in the manufacturers' documentation, connect the various elements to the main power source :

- Terminal
- RS232/current loop converter
- PLC.

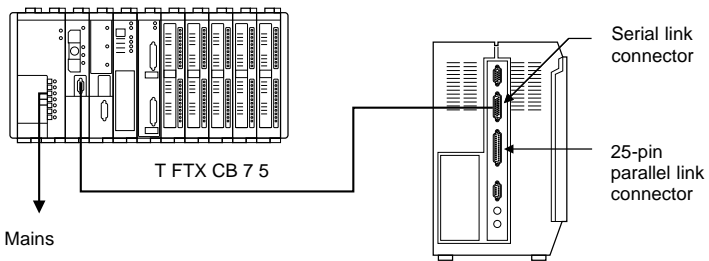
### 2.2-3 Connecting the FTX 417-20 terminal to the PLC

This connection requires the T FTX CB 7 4 connecting cable



### 2.2-4 Connecting the FTX 507 terminal to the PLC

This connection requires the T FTX CB 7 5 connecting cable



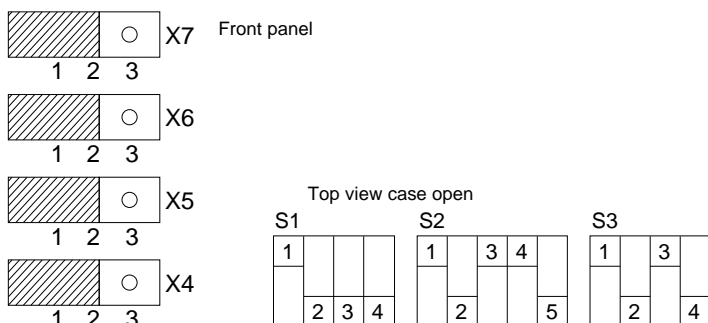
## 2.2-5 Connecting the IBM PS/2 or compatible PC to the PLC

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

### ① Checking the configuration of the TSX TE 01 converter

The converter is normally adjusted at the factory, however it is advisable to check the positions of the switches and jumpers :

- Remove the cover from the case by inserting a screwdriver in one of the slots on the side of the case and turning it until the cover can be lifted off.
- Check that the positions of the 13 switches and the 4 jumpers are correct (see the diagram below).
- Set switch S4 to the required voltage : 110V or 220V.

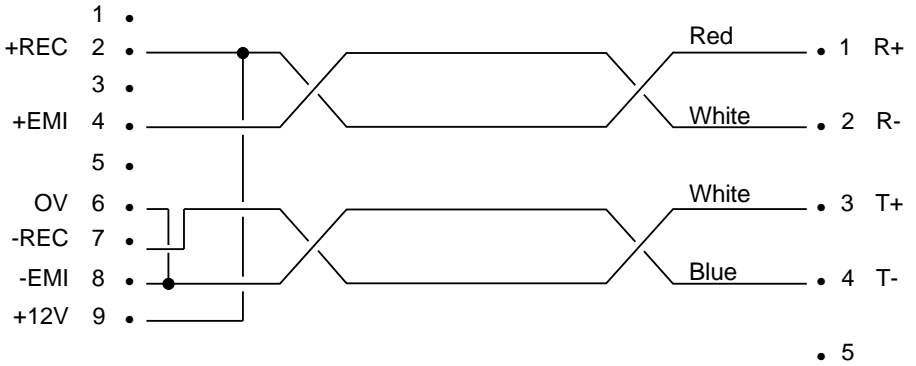


### IMPORTANT

The connection may be damaged if these settings are not correct, or if any other type of converter is used.

**② Connecting the PLC programming port to the converter**

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

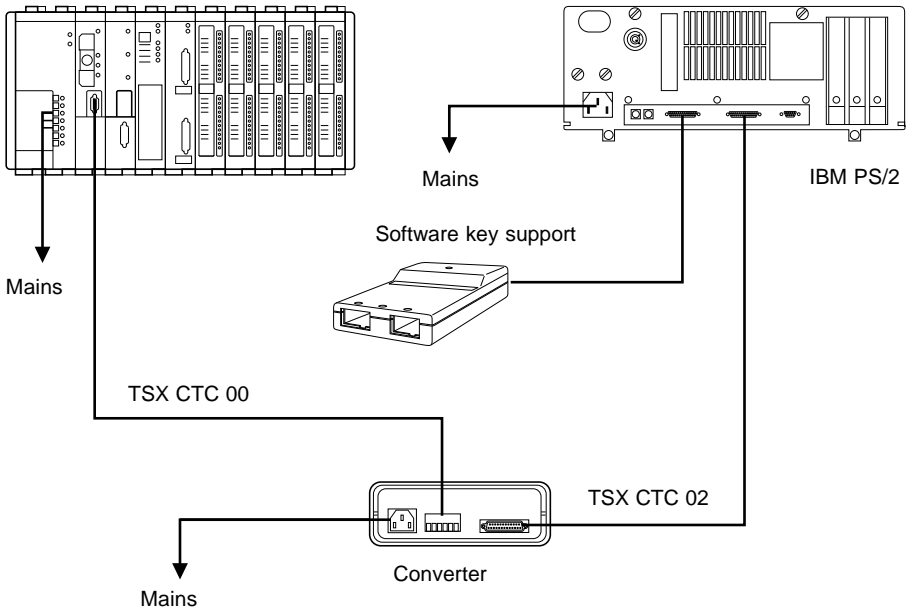


9-pin connector on  
 PLC programming port

“LINE CONNECTION”  
 5-pin screw terminal  
 on the converter

**③ Connecting the converter to the computer**

This connection uses the TSX CTC 01 or TSX CTC 02 cable connected between the 9-pin or 25-pin "serial link" connector on the computer and the 25-pin connector (“V24/RS-232 C CONNECTION”) on the converter.



---

## 2.3 Software Installation

---

### 2.3-1 Preliminary steps

Before installing the PL7-3 software on the hard disk, it is recommended that you :

- Read the license and guarantee certificates concerning the restrictions that apply to copying and installing the software.
- Make back-up copies of the original installation diskettes and use the back-up copies to work with, so as to avoid accidental damage to the originals.

### IMPORTANT

The PL7-3 diskettes are delivered set to the "write protect" position, which should not be altered.

---

### 2.3-2 Installation procedure

The following operations must be carried out before installing the PL7-3 software :

**1 • Check that the X-TEL or MINI X-TEL software workshop is already installed** (refer to the corresponding instruction manual).

Should this be the case, continue installing PL7-3. Otherwise you must install the software workshop first (refer to the corresponding instruction manual).

**2 • Close all ongoing sessions** (refer to the corresponding instruction manual).

---

### Installing PL7-3 software

- Start a full-screen OS/2 session,
- Insert PL7-3 diskette No. 1 in the disk drive,
- Identify the drive containing the diskette (a: or b:) and press <Enter>,
- After the new prompt (for example : [a:\] or [b:\]), enter the command **install** and press <Enter>,
- Follow the procedure displayed on the screen : replace diskette No.1 by diskette No.2 when requested to do so and then press <Enter>,
- When the installation is complete, insert the Grafcet OFB disk in the disk drive, then enter the command **install** and press <Enter>,
- When the installation is complete, install the OFBs in the PL7-DIAG family in the same way (refer to the corresponding instruction manual).

#### Note

If PL7-3 already exists on the station where PL7-3 GLD or PL7-3 GLT is being installed, a warning message is displayed. Installation can then be interrupted by pressing <Q> on the keyboard. The same is also true when installing PL7-3, if PL7-3 GLD or PL7-3 GLT already exist on the station.

#### Important

Under OS/2 2.1, if the screen display under PL7-3 has been altered, reinstall the X-TEL or MINI X-TEL software workshop using the command :

**install FIX21**

At the end of installation, reinitialise the work station using the command :

<Ctrl><Alt><Del>.



## 2.4 PL7-3 Files

PL7-3 files are identified by a name of 8 characters maximum, followed by a suffix of 3 characters which indicates the type of file.

They are stored in the various sub-directories of the software workshop.

The application and modular documentation files are stored by the station under the PL7\_3 sub-directory :

APP	_____	xxx.APP	<i>application description</i>
		xxx.IOC	<i>I/O configuration</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.CA1	<i>I/O wiring (racks 0/1)</i>
		to xxx.CA8	<i>I/O wiring (racks E/F)</i>
		xxx.NET	<i>network information</i>
		xxx.COM	<i>program comments</i>
		xxx.CRT	<i>documentation cartridge</i>
		xxx.DOC	<i>documentation printed in a file</i>
		xxx.SYM	<i>table of mnemonics V3 (binary)</i>
		xxx.MRK	<i>marker for cross-references</i>
	_____	MOD	_____
		xxx.MOD	<i>table of modules</i>
		xxx.LAD	<i>Ladder source archive (1)</i>
		xxx.LIT	<i>Literal source archive (2)</i>
		xxx.GR7	<i>Grafcet source archive</i>
		xxx.CST	<i>constant word source archive</i>
		xxx.OBC	<i>OFB constant word source archive</i>
		xxx.OBM	<i>OFB constant word source archive (elementary module level)</i>
		xxx.CDE	<i>documentation menu</i>
		xxx.SCY	<i>table of V4 mnemonic symbols (ASCII)</i>
		xxx.COM	<i>module comments</i>
		xxx.MCR	<i>V4 macro-modules</i>
		xxx.M5M	<i>V5 macro-modules</i>
		xxx.DDA	<i>data screen</i>
		xxx.DPR	<i>program address</i>

PL7-3 also manages the following files :

- xxx.TMP temporary file
- xxx.BAK back-up file
- xxx.OFB OFB file (in system space only).

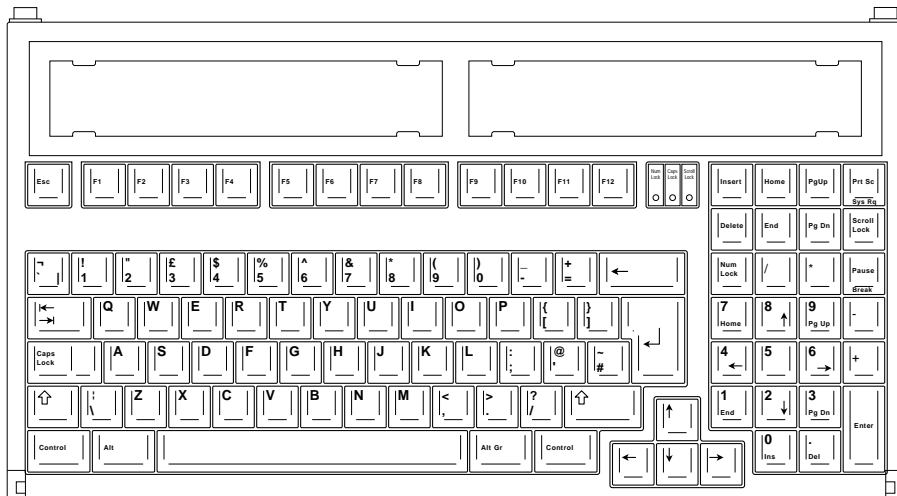
(1) Except PL7-3 GLT (2) Except PL7-3 GLD



<b>Sub-section</b>	<b>Page</b>
<b>3.1 Using the keyboard</b>	3/2
<b>3.2 Using the mouse</b>	3/5
<b>3.3 Using the display screen</b>	3/7
<b>3.4 Accessing PL7-3 Software</b>	3/9
3.4-1 From the X-TEL software workshop	3/9
3.4-2 From the MINI X-TEL software workshop	3/11
<b>3.5 Operating Mode Selection</b>	3/12
3.5-1 Mode Selection Display	3/12
3.5-2 Operating Modes	3/14
<b>3.6 Presentation of the operating modes</b>	3/16
<b>3.7 Storage and Retrieval of PL7-3 Applications</b>	3/18
3.7-1 Breakdown of a PL7-3 application	3/18
3.7-2 Storing the Application Program to Disk	3/19
3.7-3 Retrieving the Application Program from disk	3/21
<b>3.8 Calling the XTEL-MEM Tool from PL7-3</b>	3/25
<b>3.9 Using the Symbol Database</b>	3/27
3.9-1 General	3/27
3.9-2 Initialization and Updating of the Symbol Database	3/27
<b>3.10 Quitting the PL7-3 Function</b>	3/29
This section ends on page	3/30

### 3.1 Using the Keyboard

To use PL7-3 software, Telemecanique recommends an AZERTY or QWERTY keyboard with 102 keys.



#### Managing windows

The window management keys are not specific to PL7-3, but are defined by the OS/2 presentation manager. This table is therefore simply a reminder (for further information refer to the OS/2 and software workshop documentation).

Keyboard keys	Function
<Alt> or <F10>	Access to the action zone of the window.
<←><→>	Movement within the action zone of the window.
<↓>	Display a pulldown menu.
<↓><↑>	Movement within the menu.
<Esc>	Access to the body of the window.
<Tab> or <Shift><Tab>	Access to the icons of a secondary window.
<Ctrl><Esc>	Access to the Task Manager window.
<Alt><Tab>	Access to other windows (other sessions).
<Alt><Esc>	Successively activates/deactivates the windows (sessions).
<Alt><F4> to <Alt><F10>	Manages the X-TEL window (see system menu).

---

### Special PL7-3 functions

Certain special PL7-3 functions are accessible from the keyboard. The current task is interrupted and returned to afterwards. The table below lists these functions :

Keyboard keys	Function
<Ctrl><F9>	Copies the current screen to a printer.
<Alt><V>	Changes to the "display variables" mode.
<Alt><S>	Changes to the "display symbols" mode.
<Ctrl><F5>	Access to the list of symbols (see Section 18.12).
<Alt><T>	Displays the memory allocation table (see Section 18.1).
<Alt><O>	Access to the description of the OFBs configured in the application (V4 or V5 applications only).
<Alt><C>	Access to the program comments of the label or Grafcet page displayed (in Program mode only) or to the comment of the module (from the Task and Module selection screen) (see Section 15.2-6).
<Alt><K>	Entry of the application comments (see Section 18.2).
<Alt><X>	Access to the list of variables (standard PL7-3 variables, SRi, elements extracted from OFBs) used in a program element. Access is possible in the Program and Debug modes (see Section 18.5).

---

### Soft keys





















The soft keys, displayed at the bottom of the PL7-3 screens, are accessible on the keyboard through keys F1 to F9.

**In the PL7-3 software, in the help screens and in the rest of this manual, these keys are always designated by their screen designation.**

## Function keys

The PL7-3 function keys CLEAR, ZOOM, QUIT, etc are not always marked on the keyboard, but can be accessed by another key or combination of keys (as shown in the table below).

**In the PL7-3 software, in the help screens and in the rest of this manual, these keys are always designated by their PL7-3 function (CLEAR, ZOOM, QUIT, etc).**

PL7-3 Function	Keyboard keys		Description
	AZERTY	QWERTY	
ENTER			Validates the current operation.
QUIT			Cancels the current operation. Returns to the Mode Selection Display.
HELP			Displays the help screen.
RUB			Deletes the character to the left of the cursor.
DEL			Deletes the character or line in reverse video.
CLEAR			Cancels the current entry. Returns to the previous screen.
ZOOM			Accesses the parameters of the object indicated by the cursor.
INS			Inserts a space when entering characters. Permits insertion of an instruction.
→	 	 	Transfer instruction.

---

## 3.2 Using the Mouse

---

The mouse is represented on the screen by an arrow.

It can be used to select a function in a menu, to move the cursor, to validate or abandon a function, or to activate a soft key or a function key.

The mouse has a left button and a right button. Each action on one of these buttons is equivalent to pressing one or more keys on the keyboard.

For each button, three types of actions are possible :

- The “**click**”, which consists of pressing and releasing the button.
- The “**double click**”, which consists of pressing the button twice and releasing it.
- “**Drag**”, which consists of pressing a button and moving the mouse while holding the button down.

### Window management (reminder)

Button	Action	Functions obtained
Left	Double-click on an icon	Opens the corresponding window (creation of a session).
Left	Double-click on an item in a window	Activates the item.
Left	Click on a window	Activates the window.
Left	Dragging down a menu	Activates the selected item in the menu when the button is released.

---

**Use in PL7-3 screens**

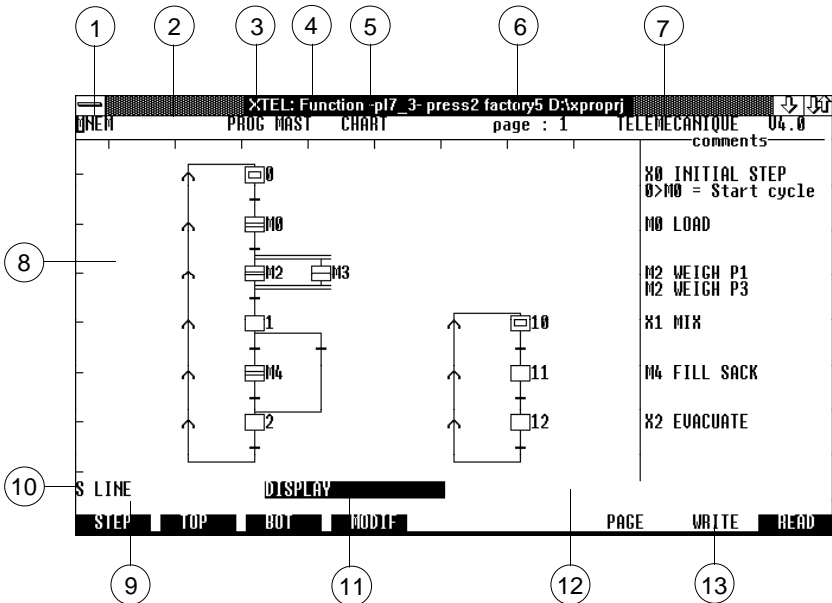
<b>Button</b>	<b>Action</b>	<b>Functions obtained</b>
Left	Click on a soft key	Activates the soft key.
Left	Click on a number of the active menu	Selects the corresponding function.
Left	Double-click	ENTER function : validates the current operation.
Left	Double-click on the PL-7 window input zone	Same as function <Alt><F10> Maximize. This function makes the PL-7 window full-screen size.
Left	Dragging	Moves the cursor. Dragging down an item in the action bar pulls down the menu of the item concerned.
Right	Click	CLEAR function : <ul style="list-style-type: none"> <li>• Cancels the current entry.</li> <li>• Returns to the previous screen.</li> </ul>
Right	Double-click	QUIT function : <ul style="list-style-type: none"> <li>• Cancels the current operation.</li> <li>• Returns to the Mode Selection Display.</li> </ul>
Right	Dragging	Displays a pulldown menu which permits the selection of a function key. The selection is effective as soon as the button is released. To leave the menu without activating a function key, release the button outside the menu. The keys displayed by the menu are as follows (see Section 3.1) : <HELP>, <RUB>, <DEL>, <CLEAR>, <ZOOM>, <INS>, <↖>, <→>.

---

### 3.3 Using the Display Screen

The window that displays the PL7-3 screens is called the display screen. All the elements specific to the software workshop (icons, title of the window, window commands, etc) are described in the corresponding documentation.

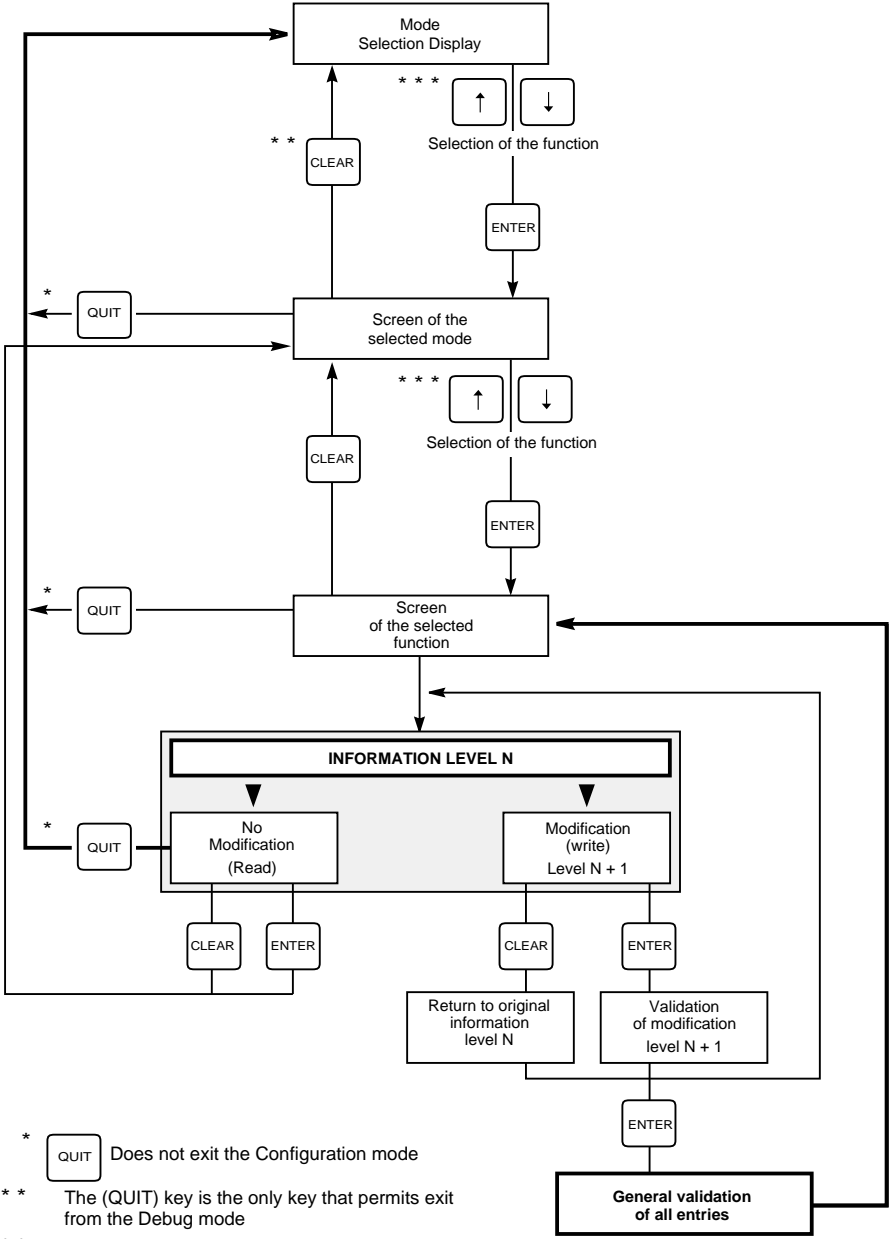
#### Information displayed



- ① Current display mode : variable or mnemonic
- ② Mnemonic name of the module
- ③ Current operating mode : Configuration, Program, Debug, Adjust, Transfer, etc
- ④ Current task : Master task, Fast task, etc
- ⑤ Current program module : CHART, Pre-processing, etc
- ⑥ Current program element : label, page, etc
- ⑦ Name of the application or information on the current elementary module
- ⑧ Field available for the application
- ⑨ Operator input or parameter display line
- ⑩ Status zone : RUN, STOP, etc
- ⑪ Work mode : Display, Configuration, Insert, etc
- ⑫ Message zone
- ⑬ Soft keys



**Sequence of screens and keyboard entries**



- \* **QUIT** Does not exit the Configuration mode
- \*\* The (QUIT) key is the only key that permits exit from the Debug mode
- \*\*\* **↑** **↓** The cursor control keys or the numeric keys can be used to select the function

### 3.4 Accessing PL7-3 software

#### 3.4-1 From the X-TEL software workshop

The procedure for accessing PL7-3 software is exactly the same for all versions. It consists of opening the corresponding window, which can be done in the following ways.

Procedure	Mouse	Keyboard
1. Select Telemecanique icon,	Double click on the icon	<↓> <↑>, <→>, <←> <Enter>
2. Open the User window,	Double click on XTEL icon	<←>, <→> <Enter>
3. Enter the user parameters, which opens the Volumes window,	Click in the input zone, xxx, Validation	<↓>, xxx, <Enter>
4. Open a volume,	Double click on the icon of the volume to be opened	<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 function,	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 operator input zone	<Alt> <↓> Maximize <Enter>

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

Procedure	Mouse	Keyboard
To turn the PL7-3 window into an icon.	Click on Minimize	<Alt> <↓> Minimize <Enter>
To open a PL7-3 session which has already been turned into an icon	Double click on the icon	<Ctrl><Esc> <↓> or <↑> <Enter>
To close a PL7-3 session	Click on minimize windows icon using right hand button, click on PL7-3 icon Click on Close	<Ctrl><Esc> <↓> or <↑> <Enter> <Alt> <↓> Close <Enter>

### Using the browser (XTEL-BROWSER)

The browser is a V5 tool which can be accessed from the Telemecanique 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 V5E).

### 3.4-2 From the MINI X-TEL software workshop

Accessing PL7-3 software is a question of opening the corresponding window. To do this :

Procedure	Mouse	Keyboard
1. Select Telemecanique icon,	Double click on the icon	<↓> <↑>, <←>, <→> <Enter>
2. Open the software workshop window, or start XTEL-Browser	Double click on corresponding icon	<←>, <→> <Enter>
3. Open the PL7-3 window,	Double click on the PL7_3 icon	<↓> <↑>, <←>, <→> <Enter>
4. For ease of use, open the PL7-3 window full-screen.	Double click in the window input zone	<Alt> <↓> Maximize <Enter>

#### Notes

If the automatic "reboot" option is selected when the MINI X-TEL software workshop is installed (for OS/2 versions 1.3 or later), the Telemecanique window is brought up automatically when the terminal is powered up.

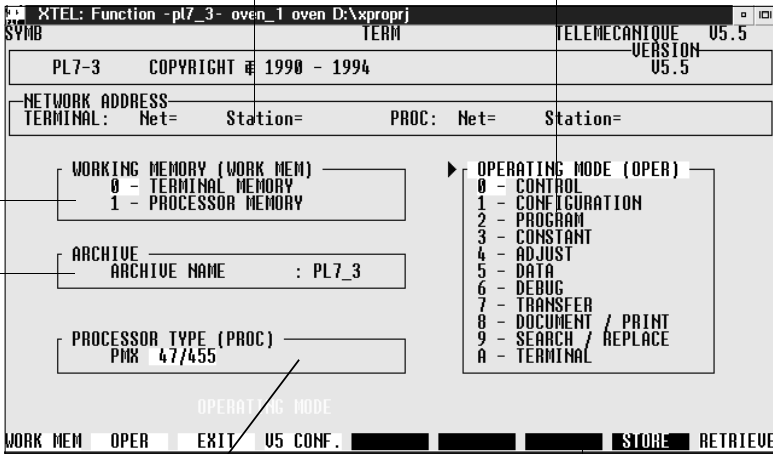
The same procedure for turning the PL7-3 window into an icon or closing the session is used for either software workshop : X-TEL or MINI X-TEL (see Section 3.4-1).

### 3.5 Operating Mode Selection

#### 3.5-1 Mode Selection Display

This first display gives access to all the possibilities offer by the terminal.

WORKING MEMORY	NETWORK ADDRESS	OPERATING MODES
<p><b>0-Local</b>:The program is written in the RAM memory of the terminal</p> <p><b>1-Connected</b>:The program is written simultaneously in the user memory of the PLC memory of the terminal</p>	<p>Reminder of the network address :</p> <ul style="list-style-type: none"> <li>• of the terminal</li> <li>• of the PLC to which the terminal is logically connected</li> </ul>	<p>11 modes are available which permit :</p> <ul style="list-style-type: none"> <li>• Creation</li> <li>• Documentation</li> <li>• Debugging, adjustment and maintenance</li> <li>• Program transfer and storage.</li> </ul>



ARCHIVE NAME	TYPE OF PROCESSOR	
Storage name of the application : PL7_3 in V5)	<p>Choice of PLC : (*)</p> <p>TSX V3 station [ TSX 67-30, TSX 87-10/20                   [ TSX 47-30, TSX 67-20                   [ TSX 87-30</p> <p>TSX V4 station [ TSX 47-400/410/411/420                   [ TSX 67/87/107-410/420</p> <p>PMX V4 station [ PMX 67/87/107-410/420                   [ PMX 107-430</p> <p>TSX V5 station [ TSX 47-405/415/425/455                   [ TSX 67/87/107-425/455</p> <p>PMX V5 station [ PMX 47/67/87/107-425                   [ PMX 47/67/87/107-455</p> <p>By default in Local mode :</p> <ul style="list-style-type: none"> <li>• V3 station : TSX 67-30</li> <li>• V4 station : TSX 47-400/PMX 47-420</li> <li>• V5 station : PMX 47-420/PMX 47-425</li> </ul>	<p>These keys give access to the various selections available :</p> <ul style="list-style-type: none"> <li>• WORKING MEMORY</li> <li>• OPERATING MODE</li> <li>• PROCESSOR TYPE</li> <li>• ARCHIVE NAME</li> <li>• STORE/RETRIEVE (in Local mode)</li> <li>• INIT-RUN-STOP (in Connected mode)</li> </ul>

(\*) The selection of a V4 processor must be done by the XTEL-MEM station tool, which enables creation of an xxx.APP application file. XTEL-CONF is used to select a V5 processor.

The Mode Selection display is used to select the working memory (Terminal or TSX memory, ie. Local or Connected mode), the operating mode and, in Local mode only, the type of processor.

By default, the Mode Selection display offers the following :

- **Local mode**  
WORKING MEMORY (WORK MEM)  
0 - TERMINAL MEMORY
- **The PLC processor (TSX 67-30, TSX 47-400, PMX 47-420, TSX 47-405 or PMX 47-425 depending on the type of station)**  
PROCESSOR TYPE (PROC)  
TSX 47-405, for a TSX V5 station
- **Control mode**  
OPERATING MODE (OPER)  
0 - CONTROL

### Soft key functions

**[WORK MEM]** Selects the working memory, which determines whether Local or Connected mode is selected.

**[OPER]** Selects the operating mode (Configuration, Program, Debug, etc)

**[EXIT]** Displays the Exit screen, which allows the operator to save files that are not automatically saved and to quit the PL7-3 function (see Section 3.8).

**[PROC]** Permits selection of the type of processor (in Local mode only) by proposing the list of possible types (for V2/V3 processors) or by permitting the selection of an .APP file (V4 processors) (see Section 18.3).

**[V5 CONF.]** Links PL7-3 with the configuration defined with XTEL-CONF. If no configuration has yet been defined, the program suggests that XTEL-CONF should be run.

**[ARCHIV]** Permits entry of the name under which the application is stored.

**[INIT]** Is proposed only in Connected mode and only if the program is not already running. It initializes the Grafset chart and the instruction pointer. After the initialization, the message INIT OK is displayed.

**[RUN PG]** Is proposed only in Connected mode. It starts the execution of the program and the letter "R" appears in the status zone. If other tasks have been configured (Fast task, auxiliary tasks), they also start running as soon as the conditions necessary for their activation are true.

**[STOP PG]** Is proposed only in Connected mode. It stops the execution of the program and the letter "S" replaces the letter "R" in the status zone.

---

**[STORE]** In Local mode, this key is used to store the following application data to the hard disk :

- The application program (xxx.BIN file)
- The I/O configuration (xxx.IOC file)
- The list of modules (xxx.MOD file)
- The list of cross references (xxx.XRi files)
- The symbol table conversion (SYM.→ .SCY ).

**[RETRIEVE]** In Local mode, this key is used to retrieve the following application data from the hard disk :

- The application program (xxx.BIN file)
  - Cross-references (xxx.XRi file),
  - The list of modules (xxx.MOD file)
  - The constant words (xxx.CST file)
  - The source file of the application program (xxx.M5M or xxx.MCR file).
- 

### 3.5-2 Operating Modes

#### In Local mode (0 - TERMINAL MEMORY)

The user must select the type of processor and the operating mode and then validate the selections made. The Local mode is especially intended for using the terminal in the design office for configuring the application, entering the program in the internal RAM memory, storing it on diskette or hard disk, and documenting and printing the application.

#### Note

The Control, Adjust, Data and Debug modes cannot be accessed when the terminal is in Local Mode.

#### In Connected mode (1 - PROCESSOR MEMORY)

In this case, the terminal uses the memory of the PLC to which it is connected (onboard RAM and memory cartridge). Two types of operation are possible :

##### • Locally connected

The terminal is directly connected to a TSX 7 or PMX 7 PLC, and the memory which is used is the User memory supported by this PLC. This mode can be used, for example, for the on-line debugging, adjustment and modification of an application, and for storing it to hard disk.

##### • Network connected TELWAY, MAPWAY, ETHWAY or FIPWAY (1)

The terminal is directly connected to a network (except TELWAY), or to a PLC which is itself connected to a network. This type of operation offers the same possibilities (debugging, adjustment, etc) as in the locally connected mode, but with a remote PLC.

The access times depend on the network load.

The address of the station is defined at Project level under X-TEL.

(1) PL7-3 only

---

The following table gives the possibilities of use by a maximum programmer for each operating mode, with the terminal in Local or Connected mode.

Operating mode	Connected mode	Local mode
0 - CONTROL	Yes	No
1 - CONFIGURATION	Read, write	Read, write
2 - PROGRAM	Read, write	Read, write
3 - CONSTANT	Read, write	Read, write
4 - ADJUST	Yes	No
5 - DATA	Yes	No
6 - DEBUG	Yes	No
7 - TRANSFER	Terminal ↔ Processor Processor ↔ disk	Terminal ↔ disk
8 - DOCUMENT/PRINT	Yes	Yes
9 - SEARCH/REPLACE	Yes	Yes
A - TERMINAL	Yes	Yes

#### Reminder of the different levels of access rights of X-TEL users :

- Operate min. : access to the display tools
- Operate max. : access to the operating and process diagnostic tools
- Adjust min. : access to the application adjustment tools
- Adjust max. : access to the adjustment, PLC operation (Run/Stop) maintenance and diagnostic tools
- Program min. : access to all the design tools
- Program max. : access to all the workshop tools in Local mode, and in Connected mode with the PLC running or stopped.

For the access rights to PL7-3 operating modes by the different categories of users, refer to Section 18.6.



## 3.6 Presentation of the Operating Modes

The PL7-3 software proposes 11 operating modes which permit the :

- Design
- Debugging
- Adjustment
- Maintenance
- and Documentation of an automated system.

### In Local mode

Operating modes

Configuration

Document/Print

Program  
Search/Replace

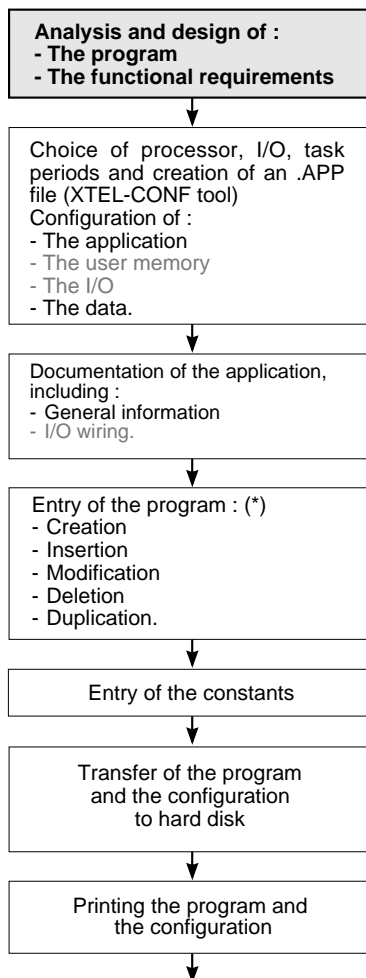
Constant

Transfer

Document/Print

### Note

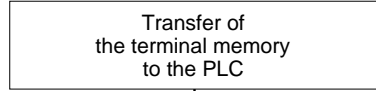
As the configuration, the program and the mnemonics are entered in the RAM memory of the terminal, it is recommended to make frequent saves to the hard disk.



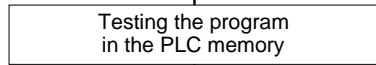
**In Connected mode**

**Operating modes**

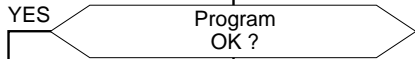
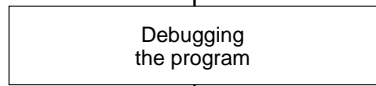
Transfer



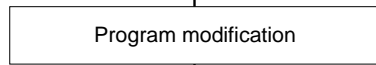
Debug



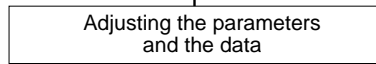
Debug Data



Program Search/Replace



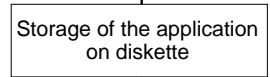
Adjust Data



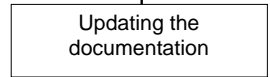
Transfer



Return to the design office



Document/Print



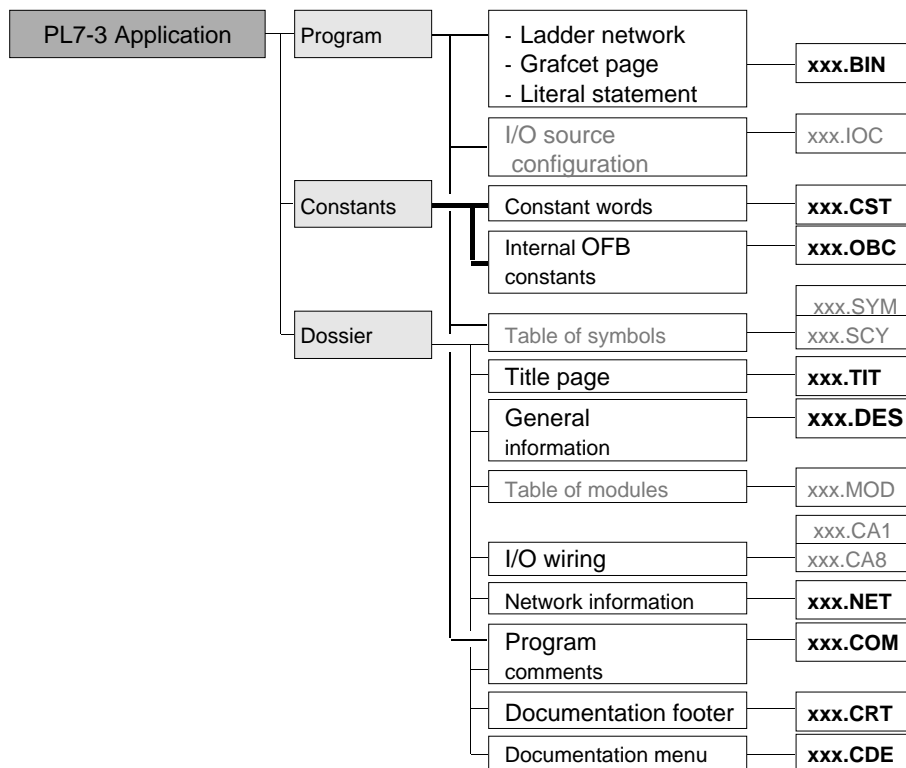
## 3.7 Storage and Retrieval of PL7-3 Applications

### 3.7-1 Breakdown of a PL7-3 application

A PL7-3 application is composed of :

- The application program, accessible in the Program mode
- The application dossier, accessible in the Document/Print mode.

It is broken down into files (xxx.BIN, xxx.COM, etc.) which permit the storage and retrieval of the various parts of the PL7-3 application.



#### Notes

- The program files (xxx.BIN), mnemonic symbols (xxx.SCY or xxx.SYM) and Documentation footer file (xxx.CRT) are not saved automatically. It is therefore recommended to make frequent saves to avoid any loss of 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).
- The xxx.IOC file is accessible only in Configuration mode.

### 3.7-2 Storing the Application Program to Disk

PL7-3 software offers several possibilities for storing and retrieving PL7-3 application data :

- By using the [STORE] and [RETRIEVE] soft keys on the Mode Selection display in the local mode
- By using the modes corresponding to each file handled :
  - Transfer mode for the xxx.BIN file (see Section 14.3)
  - Configuration mode for the xxx.IOC file (see section 6.4)
  - Program mode for the xxx.MOD file (see section 7.1-3).

Using the soft keys gives faster access to the files.

#### Storage procedure

**[STORE]** Permits storage of the following application data if it is present in the terminal memory :

- Application (xxx.BIN file) in APPLI zone,
- I/O configuration (xxx.IOC file) in APP zone under STATION,
- List of modules (.MOD file) in MOD zone,
- Table of cross references (xxx.XR1 and xxx.XR4 files) in APPLI zone.

XTEL: Function -pl7 3- trolley1 factory1 E:\xproj

SYMB STORE TELEMECANIQUE U5.0

Selection	Store	Files	Format	OLD-NEW
<input type="checkbox"/>	APPLICATION (.BIN)	PL7 3		

CLEAR or QUIT: return to MODE CHOICE

STR.NAME UNSELECT SEL.CUR. STORE

---

**[STR.NAME]** Enables modification of the current file archive name (in reverse video in the table) and provides access to the following soft keys :

**[DIR]** lists all the files already stored under the corresponding PL7\_3\APPLI or PL7\_3\MOD sub-directory.

<↓><↑> moves the cursor onto the required file.

**[SEARCH]** permits entry of the name (8 characters maximum) of the file (without suffix) to be searched for. After validation, only the files having this name are displayed.

**[DELETE]** deletes the selected file (after confirmation).

**[STATION]** assigns the name of the current station as the name of the file.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.

<ENTER> validates the selection. If the file already exists, the new program will overwrite the old one, after confirmation by [YES].

**[UNSELECT]** Unselects the file indicated by the cursor from the selection to be saved. The asterisk in the selection column is deleted.

**[SELECT]** Selects the file indicated by the cursor for storage. An asterisk appears in the selection column. This soft key appears alternately with the [UNSELECT] soft key.

**[SEL.CUR.]** Selects only the file indicated by the cursor for storage. The other files are not saved.

**[STORE]** Saves all the selected files.

If the file already exists, the new data will overwrite the old file, after confirmation by [YES]. If it is a level 4 station and the xxx.BIN application file is selected, PL7-3 calls up the XTEL-MEM tool to update the xxx.APP file.

The same function can be obtained by pressing <ENTER>.

The following soft keys appear only when the cursor is positioned on the line of the file concerned :

**[OLD]** Selects the existing tables of cross references.

**[NEW]** Recomputes the tables of cross references.

### 3.7-3 Retrieving the Application Program from disk

#### Retrieval procedure

**[RETRIEVE]** Permits loading to the terminal memory of :

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

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

- The application source,
- The standard constants .

#### Note

If the application includes modules programmed in a language which is not supported, the PL7-3 GLD and PL7-3 GLT programs only authorize the Control, Adjust and Transfer modes.

SYMB XTEL: Function -pl7\_3- trolley1 factory1 E:\xproprj RETRIEVE TELEMECANIQUE 05.0

Selection	Recovering	Files	Mode
□	APPLICATION (.BIN)	PL7_3	

	READ CONSTANTS (.CST)	DIR	SEARCH
	READ XTEL-MOD/PL7-PMS2		DELETE

CLEAR or QUIT: return to MODE CHOICE

STR.NAME UNSELECT SEL.CUR RETRIEVE SOURCE

**[STR.NAME]** Permits modification of the name of the current file and gives access to the following soft keys :

**[DIR]** lists all the files in the PL7\_3 or MOD sub-directory (depending on the selection made).

<↓><↑> moves the cursor onto the required file.

**[SEARCH]** permits entry of the name of the file to be searched for.

**[DELETE]** deletes the selected file (after confirmation).

**[STATION]** assigns the name of the current station as the name of the file to be retrieved.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file to be retrieved.

<ENTER> validates the selection and starts retrieval of the file.

- 
- [SOURCE]** Provides access to the "Read source" field.
- [APPLI]** Provides access to the "Application" field.  
This soft key appears alternately with the [SOURCE] soft key.
- [UNSELECT]** Unselects the file indicated by the cursor from the selection to be retrieved. The asterisk in the selection column is deleted.
- [SELECT]** Selects the file, indicated by the cursor, to be retrieved. An asterisk appears in the selection column. This soft key appears alternately with the [UNSELECT] soft key.
- [SEL.CUR.]** Selects only the file indicated by the cursor. The other files are not retrieved.
- [RETRIEVE]** Validates the selection and starts retrieval of the selected files.  
The same function can be obtained by pressing <ENTER>.
- [MODE]** Selects the automatic (AUTO) or manual (MANU) mode for reading the xxx.MOD files (list of modules), xxx.MCR files (V4 read source), xxx.M5M (V5 Read source) and xxx.CST (standard constants read source).  
The AUTO mode is selected by default.
- [FORMAT]** Specific to the reading of an xxx.CST file. Defines the retrieval format : retrieval directed by symbols (SYMB), addresses (VAR) or by values (VAL).
- [FROM]** Specific to the reading of an xxx.CST file. Used to define the CW or CDW address or symbol (in SYMB or VAL format) which will receive the first value to be read (start of the retrieval zone).
- [TYPE]** Specific to the reading of an xxx.CST file. Used to define the type of constant (in VAR format) which will receive the first value to be read : CW, CDW\_ODD (even) or CDW\_EVEN (uneven).
- [OBC]** Specific to the reading of XTEL-MOD/PMS2. Used to read the XTEL-MOD or PMS2 source modules, with or without the internal OFB constants.

**Note**

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

## "Read source" in AUTO and MANU modes

"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 the Program mode, the original files of the module (WRITE soft key) and then create each module using the XTEL-MOD tool.
- Using the same tool, the user must then define the macro-module and the modules that it contains. The elements composing a macro-module are program files (xxx.LIT, xxx.LAD, xxx.GR7), constant files (xxx.CST), internal constant files from OFB (xxx.OBC) and comments 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 that have already been read.

### • Reading the source file in the automatic mode (AUTO)

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

### • Reading the source file in the manual mode (MAN)

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

```

XTEL: Function -pl7_3- oven_1 oven D:\xproj
  READ XTEL-MOD          BASIC MODULE   : OUI_MOD3
  PL7-PHS2              MACRO MODULE   :
  * PROGRAM              LANGUAGE      : LIT
                        ADDRESS        : MAST
                        TASK MODULE    : PRL
                        TYPE OF READING : NEW
                        AUTHORIZATION (R/W) : R/W
                        FROM           : MOD
  * INTERNAL CONSTANTS OF OFB          TYPE OF READING :
  ENTER: confirm CLEAR/OUT: ignore the module
  > OUI_MOD3 LIT MAST PRL NEW RW MOD -- OUI_MOD3 U0 10-17-94
  ABORT
  
```



- 
- <ENTER>** Starts reading.
- <CLEAR>** Ignores the current module or macro-module and goes to the next one.
- [ABORT]** Stops reading.
- [AUTO]** Continues reading in the automatic mode, as before.

**Note**

The creation, use and handling of the modules and macro-modules is described in XTEL-MOD manual TXT DM MDD V5E.

### 3.8 Calling the XTEL-MEM Tool From PL7-3

Each time the xxx.BIN files are saved, the updating of the corresponding xxx.APP file is proposed by calling the XTEL-MEM tool from PL7-3.

After saving the xxx.BIN file to disk (by using the [STORE] or [EXIT] soft keys, or by making a Terminal to Disk transfer in the Transfer mode), the following dialog screen is displayed :

XTEL: Function -pl7_3_oven factory D:\xproj				
MNEM	STORE	TELEMECANIQUE	04.5	
Selection	Store	Files	Format	OLD-NEW
*	APPLICATION (.BIN and .APP)	OVEN		
MEM LAUNCH TO UPDATE .APP FILE ?				
NO				YES

**[NO]** The xxx.APP file is not updated.

**[YES]** Displays the following dialog screen which permits entry of the name of the xxx.APP file to be updated.

XTEL: Function -pl7_3_oven factory D:\xproj				
MNEM	STORE	TELEMECANIQUE	04.5	
Selection	Store	Files	Format	OLD-NEW
*	APPLICATION (.BIN and .APP)	OVEN		
STORAGE NAME : OVEN				
DIR		STATION		ARCHIV

---

**[DIR]** lists all the files already stored under the PL7\_3\APPLI and APP sub-directory.

<↓><↑> moves the cursor onto the name of the selected file.

**[SEARCH]** permits the entry (8 characters maximum) of the name (without its suffix) of the file to be searched for. After validation, only the files having the name that has been entered are displayed.

**[DELETE]** deletes the selected file (after confirmation).

**[STATION]** assigns the name of the current station as the name of the file.

**[ARCHIV]** assigns the global archive name as the name of the file.

If the user knows the name of the xxx.APP file in which his xxx.BIN file is to be integrated, <ENTER> starts the XTEL-MEM tool and the automatic updating of the xxx.APP file.

If the user does not know the name of the xxx.APP file in which his xxx.BIN file is to be integrated, <CLEAR> or <QUIT> starts the XTEL-MEM tool and displays the main window of this tool which permits selection of the name of the xxx.APP file and allows modifications to be made, as in a normal MEM session.

If the XTEL-MEM tool is already running, an error message indicates that automatic updating of the xxx.APP file is not possible. In this case, the user must :

- Either close the current MEM session and start the automatic updating procedure of the xxx.APP file with PL7-3
- Or update the xxx.APP file directly with the XTEL-MEM tool.

---

## 3.9 Using the Symbol Database

---

### 3.9-1 General

#### Principle of the XTEL-SDBASE tool

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

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 the use of the XTEL-SDBASE tool are described in detail in the manuals TXT DM XTEL V5E and TXT DM BJR V5E.

#### Definition of a symbol

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

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), the extended ASCII codes. On the other hand, a symbol can start with the name of a PL7 object or instruction (W0\_mast, B10prl, etc).

---

### 3.9-2 Initialization and Updating of the Symbol Database

#### Initialization

In order to use the symbol database with 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 stored and previously entered in a xxx.SCY file.

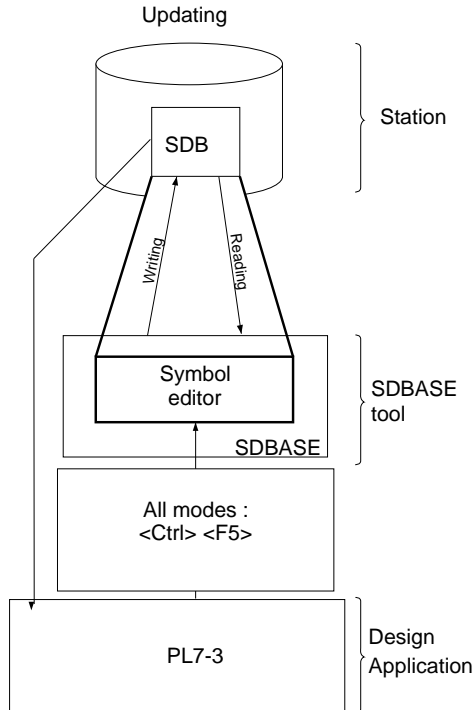
## Updating

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

- In any operating mode :  
**<Ctrl><F5>** Quits PL7-3 and calls up the symbol editor of the XTEL-SDBASE tool for modification or insertion of the symbols.

After modification or insertion of the symbols, pressing the function key **<F3>** twice in succession allows the user to exit from the XTEL-SDBASE editor.

## Block diagram



## Important

In the MODE SELECTION display, PL7-3 does not make any reservations in the database in read/write mode. However, in all other modes it does reserve the database and prohibits any access to SDBASE.

### 3.10 Quitting the PL7-3 Function

This screen can be accessed from the Mode Selection display by pressing the EXIT key. It permits :

- Saving of files that are not saved automatically. The list of items to be saved and the names of the corresponding files are displayed in a table :
  - Application (xxx.BIN) and I/O configuration (xxx. IOC)
  - List of modules (xxx.MOD)
  - Table of cross references (xxx.XR1 to xxx.XR2)
- Automatic updating of the xxx.APP file after modification of a xxx.BIN file
- Quitting the PL7-3 function.

Selection	Store	Files	Format	OLD-NEW
<input checked="" type="checkbox"/>	APPLICATION (.BIN)	PL7_3		

CLEAR or QUIT: return to MODE CHOICE

STR.NAME UNSELECT STO&EXIT      SEL.CUR      STORE      EXIT

**[STR.NAME]** Permits entry of the storage name of the file (in reverse video in the table).

**[DIR]** lists the existing files. Saving a new file will overwrite an old one of the same name.

**<↓><↑>** moves the cursor onto the name of the selected file.

**[SEARCH]** permits entry of the name (without its suffix) of the file to be searched for.

**[DELETE]** deletes the selected file (after confirmation).

**[STATION]** assigns the name of the current station as the name of the file.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.

---

**[UNSELECT]** Cancels selection of the file to be saved (indicated by the cursor). The asterisk in the selection column is deleted.

**[SELECT]** Selects the file to be saved (indicated by the cursor). An asterisk appears in the selection column.  
This soft key appears alternately with the [UNSELECT] soft key.

**[SEL.CUR.]** Selects only the file indicated by the cursor. The other files are not saved.

**[STO&EXIT]** Saves all the files selected, calls the XTEL-MEM tool for updating of the xxx.APP file and quits the PL7-3 function.  
If the file already exists, the new data will overwrite the old file, after confirmation by [YES].

**[STORE]** Saves all the files selected and calls the XTEL-MEM tool for updating of the xxx.APP file, but without quitting the PL7-3 function.  
If the file already exists, the new data will overwrite the old file, after confirmation by [YES].

**[EXIT]** Quits the PL7-3 function without saving.

The following soft keys appear only when the cursor is positioned on the line of the file concerned :

**[OLD]** Selects the existing tables of cross references.

**[NEW]** Recomputes the tables of cross references.



<b>Section</b>	<b>Page</b>
<b>4 Operation with the terminal connected to a PLC</b>	
Contents	4/1
4.1 Physical and Logical Connection	
4.2 Reservation of a Connected PLC	
4.3 Principles of Terminal to PLC Connection	
4.4 Operation in Locally Connected Mode	
4.5 Operation in Network Connected Mode	
4.6 Repeating the Connection Request	
<b>5 Control Mode</b>	
Contents	5/1
5.1 Description of the Mode	
<b>6 Configuration Mode</b>	
Contents	6/1
6.1 Presentation of the Mode	
6.2 Configuring the Application	
6.3 Configuration of the User Memory	
6.4 Configuring the I/O Modules	
6.5 Configuring the Bits, Words and Function Blocks	
6.6 Configuring the Grafcet Chart	
6.7 Configuring the Optional Function Blocks	
6.8 Configuring the Network Services	
6.9 Miscellaneous Configuration	
6.10 Modifying the Configuration in Connected Mode	
6.11 Reconfiguration Procedure	



---

**B**



<b>Sub-section</b>	<b>Page</b>
<b>4.1 Physical and Logical Connection</b>	4/2
4.1-1 Physical Connection	4/2
4.1-2 Logical Connection	4/2
<b>4.2 Reservation of a Connected PLC</b>	4/3
4.2-1 Reservation in Network Connected Mode (1)	4/3
4.2-2 Dereservation	4/3
4.2-3 Operation in Locally Connected Mode	4/4
<b>4.3 Principles of Terminal to PLC Connection</b>	4/4
4.3-1 Identical PLC and Terminal Memories	4/4
4.3-2 Terminal to PLC Connection	4/5
<b>4.4 Operation in Locally Connected Mode</b>	4/6
4.4-1 Connection Procedure	4/6
<b>4.5 Operation in Network Connected Mode (1)</b>	4/8
4.5-1 Connection Procedure	4/8
<b>4.6 Repeating the Connection Request</b>	4/10

This section ends on page 4/10

(1) PL7-3 only

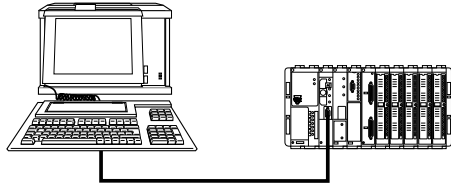
---

## 4.1 Physical and Logical Connection

---

### 4.1-1 Physical Connection

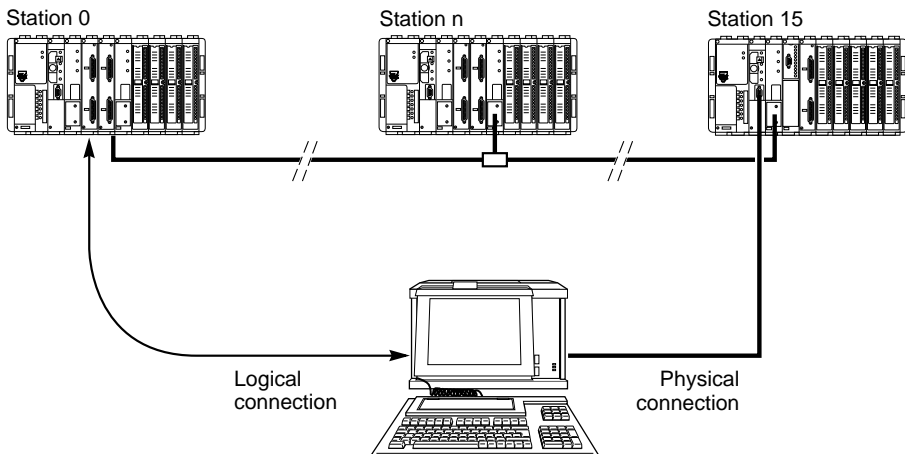
Physical connection corresponds to the local connection of a terminal to a PLC, the physical connection being established by the connecting cable.



---

### 4.1-2 Logical Connection

Logical connection corresponds to the local or remote connection between the terminal and the PLC with which it is communicating, either locally or through a network. When a terminal is locally connected to a PLC, the logical connection corresponds to the physical connection.



For example, in the TELWAY network above, the terminal is physically connected to station 15 (1). The logical connection can be established with any station on the network (here station 0).

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

The choice is determined by the user by :

- The definition of the logical connection at Project level, by the selection of a station in the secondary Stations window,
- The selection of 1 - TSX MEMORY TSX on the Mode Selection display,
- The reading of the physical (TERMINAL) and logical (TSX) addresses in the NETWORK ADDRESS box of the Mode Selection display.

(1) PL7-3 only

---

## 4.2 Reservation of a Connected PLC

---

### 4.2-1 Reservation in Network Connected Mode (1)

On a TELWAY, MAPWAY, ETHWAY or FIPWAY network, any FTX507 or microcomputer terminal can be physically connected to any station (TSX/PMX 47, 67, 87 or 107). Consequently, several terminals can request logical connection to the same station in the authorized operating modes.

In order to avoid conflicts of access and procedure, each terminal therefore automatically reserves the PLC to which it is logically connected (except in the ADJUST, TRANSFER and TERMINAL modes).

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

As soon as one of the following operating modes is selected, the terminal automatically reserves the PLC to which it is logically connected :

0 - CONTROL	1 - CONFIGURATION
2 - PROGRAM	3 - CONSTANT
5 - DATA	6 - DEBUG
8 - DOCUMENT/PRINT	9 - SEARCH/REPLACE

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

In the ADJUST, TRANSFER and TERMINAL modes, no reservation can be made. Therefore in these modes, any terminal can be logically connected to any PLC on the network.

---

### 4.2-2 Dereservation

The reservation of a PLC is cancelled one minute after it is logically disconnected from the terminal that made the reservation. This corresponds to the quitting of the connected mode by the terminal that 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. On the other hand, dereservation has no effect on the state of forced bits.

---

(1) PL7-3 only

---

### 4.2-3 Operation in Locally Connected Mode

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

---

## 4.3 Principles of Terminal to PLC Connection

---

Access to operating modes depends on the operating mode of the terminal (local or connected). Certain operating modes are only available when the terminal is connected to a PLC.

The rules of access are the same whether the terminal is connected locally or remotely via a TELWAY, MAPWAY, ETHWAY or FIPWAY (1) network.

Of the 8 operating modes that effect a reservation, 7 of them require the contents of the memory in the PLC and the contents of the memory in the terminal to be identical. These modes are :

- |                    |                    |
|--------------------|--------------------|
| 1 - CONFIGURATION  | 2 - PROGRAM        |
| 3 - CONSTANT       | 5 - DATA           |
| 6 - DEBUG          | 8 - DOCUMENT/PRINT |
| 9 - SEARCH/REPLACE |                    |

---

### 4.3-1 Identical PLC and Terminal Memories

To ensure that these memories are identical, which is essential in Connected 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 14.3).
- Load the local memory of the PLC into the memory of the terminal by using the function TRANSFER PROCESSOR MEMORY→TERMINAL (see Section 14.2).

---

(1) PL7-3 only

#### 4.3-2 Terminal to PLC Connection

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

- **PROGRAM mode**

During program creation and modification, the terminal works with its internal memory and at the same time automatically updates the memory in the PLC.

- **DEBUG, DATA and ADJUST modes**

Only the data objects that are used are exchanged (animation, real-time values, etc). This optimizes the performance of the terminal in Connected mode so that the response times are similar to those obtained in Local mode.

At least one up-to-date copy of the application program of each PLC should always be stored on disk. This not only safeguards the application, but also allows it to be transferred to the terminal from the back-up disk with a shorter transfer time than if the transfer was made direct from the PLC.

## 4.4 Operation in Locally Connected Mode

### 4.4-1 Connection Procedure

On the Mode Selection display, the procedure for changing from Local mode to Locally connected mode is as follows :

The screenshot shows a terminal window with the following content:

```
TEL: Function -pl7_3- oven_1 oven D:\xproj
SYMB          TERM          TELEMECANIQUE U5.5
PL7-3  COPYRIGHT © 1990 - 1994          VERSION
                                           U5.5
NETWORK ADDRESS
TERMINAL: Net=      Station=      PROC: Net=      Station=
WORKING MEMORY (WORK MEM)
0 - TERMINAL MEMORY
1 - PROCESSOR MEMORY
ARCHIVE
ARCHIVE NAME      : PL7_3
PROCESSOR TYPE (PROC)
PWR 47/455 U5.0
OPERATING MODE (OPER)
0 - CONTROL
1 - CONFIGURATION
2 - PROGRAM
3 - CONSTANT
4 - ADJUST
5 - DATA
6 - DEBUG
7 - TRANSFER
8 - DOCUMENT / PRINT
9 - SEARCH / REPLACE
A - TERMINAL
R LINE          OPERATING MODE
WORK MEM  OPER  EXIT  U5 CONF.  STOP PC  STORE  RETRIEVE
```

**[WORK MEM]** Allows the operating mode to be changed,

<1> positions the cursor on 1 - PROCESSOR MEMORY (Connected mode)

**[OPER]** Permits selection of the required operating mode

<ENTER> Validates the selections according to the operating mode selected :

If either 0 - CONTROL or 4 - ADJUST are selected the logical connection will be established with the PLC to which the terminal is physically connected.

If any of the following are selected,

- |                    |                    |
|--------------------|--------------------|
| 1 - CONFIGURATION  | 2 - PROGRAM        |
| 3 - CONSTANT       | 5 - DATA           |
| 6 - DEBUG          | 7 - TRANSFER       |
| 8 - DOCUMENT/PRINT | 9 - SEARCH/REPLACE |

the internal memory of the terminal must be identical with the memory in the PLC, otherwise an error message is displayed.

**[YES]** Transfers the PLC's memory 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), the logical connection is immediate.



## 4.5 Operation in Network Connected Mode (1)

### 4.5-1 Connection Procedure

The procedure for changing from Local mode to Network Connected mode is the same as that described above, except that the number of the station to which the terminal is to be logically connected must be defined (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 repeats it on the Mode Selection display in the NETWORK ADDRESS box.

```
XTEL: Function - pl7_3- oven_1 oven D:\xproj
SYMB TERM TELEMECANIQUE U5.5
PL7-3 COPYRIGHT © 1990 - 1994
NETWORK ADDRESS
TERMINAL: Net= Station= PROC: Net=1 Station=6
WORKING MEMORY (WORK MEM)
0 - TERMINAL MEMORY
1 - PROCESSOR MEMORY
ARCHIVE
ARCHIVE NAME : PL7_3
PROCESSOR TYPE (PROC)
PRK 47/455
OPERATING MODE (OPER)
0 - CONTROL
1 - CONFIGURATION
2 - PROGRAM
3 - CONSTANT
4 - ADJUST
5 - DATA
6 - DEBUG
7 - TRANSFER
8 - DOCUMENT / PRINT
9 - SEARCH / REPLACE
A - TERMINAL
WORK MEM OPER EXIT U5 CONF. STORE RETRIEVE
```

After the station number has been defined, the Mode Selection display permits selection of the operating mode (1 - PROCESSOR MEMORY) and provides access to the operating mode itself.

**<ENTER>** Validates the selections according to the mode selected.

If either 0 - CONTROL or 4 - ADJUST are selected, the logical connection will be immediately established with the selected station. As there is no updating, the contents of the internal memory of the terminal are saved.

But if any of the following are selected,

- |                    |                    |
|--------------------|--------------------|
| 1 - CONFIGURATION  | 2 - PROGRAM        |
| 3 - CONSTANT       | 5 - DATA           |
| 6 - DEBUG          | 7 - TRANSFER       |
| 8 - DOCUMENT/PRINT | 9 - SEARCH/REPLACE |

the internal memory of the terminal must be identical with the memory in the PLC, otherwise an error message is displayed.

(1) PL7-3 only

**[YES]** Transfers the PLC memory to the internal memory of the terminal by overwriting it. The time taken for connection may be longer than when using Local mode (for example when using MAPWAY which requires certain preparation).

**[NO]** Cancels the transfer request.

---

## 4.6 Repeating the Connection Request

---

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

### Connection error between console and PLC

- PLC not connected or not powered up
- PLC defective.

**<ENTER>** Repeats the connection request

**<QUIT>** If the terminal has not detected a fault, this permits a return to Local mode with the application program preserved intact. 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 not connected or not powered up
- PLC defective
- Network coupler module absent
- Network overloaded.

The terminal then continually repeats the connection request. If connection becomes possible, the communication is established, if not :

**<ENTER>** Repeats the request for communication with the PLC.

**<QUIT>** If the terminal has not detected a fault, this request is cancelled and the application program is preserved intact. Otherwise, the terminal is loaded with the default configuration and the application that was in the internal memory of the terminal is lost.



<b>Sub-section</b>	<b>Page</b>
<b>5.1 Description of the Mode</b>	<b>5/2</b>
5.1-1 Accessing the Control Mode	5/2
5.1-2 Use in the Network Connected Mode	5/2

This section ends on page 5/2



---

## 5.1 Description of the Mode

---

The CONTROL mode enables the terminal to be logically connected to a PLC for the purpose of displaying its status and controlling the execution of the program.

The connection is made without transferring the program :

- The internal memory of the terminal is not erased
- Logical connection is immediate.

---

### 5.1-1 Accessing the Control Mode

The Control Mode is accessed by selecting 0 - CONTROL on the Mode Selection display, on condition that the terminal is in Connected mode (ie working memory = TSX memory).

#### Display of the status of the connected PLC

The following data are displayed in the status zone of the screen :

- S** PLC Stopped.
- R** PLC Running with at least one task being executed (or stopped on a breakpoint).
- LINE** **Steady state (blue)** : Locally connected.  
**Blinking (yellow)** Network connected.
- f** At least one bit in the configuration has been forced to 0 or 1.

#### Control of a connected PLC

Control is effected by using the following soft keys :

- [INIT]** Initializes the PLC (after confirmation by [YES]).
- [RUN PG]** Runs the program if it was stopped (after confirmation by [YES]).
- [STOP PG]** Stops the program if it was running (after confirmation by [YES]).

---

### 5.1-2 Use in the Network Connected Mode (1)

When the terminal is connected to a station (eg station x), the internal memory of the terminal contains the application program of this station. By using CONTROL mode, the terminal can be connected to any other station on the network without losing the application program of station x. When the terminal re-establishes logical connection with station x, the connection is immediate, since acquisition of the program of station x is not necessary.

(1) PL7-3 only



<b>Sub-section</b>	<b>Page</b>
<b>6.1 Presentation of the Mode</b>	6/2
<b>6.2 Configuring the Application</b>	6/4
<b>6.3 Configuration of the User Memory</b>	6/8
6.3-1 V2 or V3 Applications	6/8
6.3-2 V4 and V5 Applications	6/10
6.3-3 V5 Application only	6/12
<b>6.4 Configuring the I/O Modules</b>	6/16
<b>6.5 Configuring the Bits, Words and Function Blocks</b>	6/22
6.5-1 Procedure	6/22
6.5-2 Configuring the Function Blocks	6/24
6.5-3 Configuring the Text Blocks	6/25
<b>6.6 Configuring the Grafcet Chart</b>	6/27
<b>6.7 Configuring the Optional Function Blocks</b>	6/29
<b>6.8 Configuring the Network Service</b>	6/31
<b>6.9 Miscellaneous Configurations</b>	6/32
<b>6.10 Modifying the Configuration in Connected Mode</b>	6/33
<b>6.11 Reconfiguration Procedure</b>	6/35
6.11-1 General	6/35
6.11-2 Reconfiguration Check	6/37
6.11-3 Reconfiguration	6/37
6.11-4 Abort Reconfiguration	6/38
6.11-5 Reconfiguration Check Printout	6/38

This section ends on page 6/38

---

## 6.1 Presentation of the Mode

---

The Configuration mode enables the user to define the context of the physical and logical operation of the application. Defining the configuration is essential for the creation or modification of the application, in order to check the coherence of the objects entered (type, number, etc) in relation to those declared. Any inconsistencies that are detected are indicated by error messages.

The Configuration mode has 7 headings that permit the configuration of :

- The application
- The user memory and the memory segments (code and graph)
- The I/O
- The bits, words and function blocks
- The Grafcet chart (if used)
- The optional function blocks (levels V4 and 5 only)
- The network
- The data link (programming port) of the PLC

To quit the mode, validate the modifications that have been made, which causes either :

- Reconfiguration of the application if the modifications are important, in which case they are authorized only in the Local mode.  
It is recommended that a diagnostic check of the new configuration is carried out before requesting reconfiguration of the application. This check is executed by the terminal and ensures coherence between the new configuration and the existing program.
- No reconfiguration of the application if the modifications are minor (modification of task times or of function block parameters). In this case, they are authorized in both Local and Connected modes whether the program is running or stopped. However, they are not taken into account by the application program until after a cold restart (SY0 = 1).

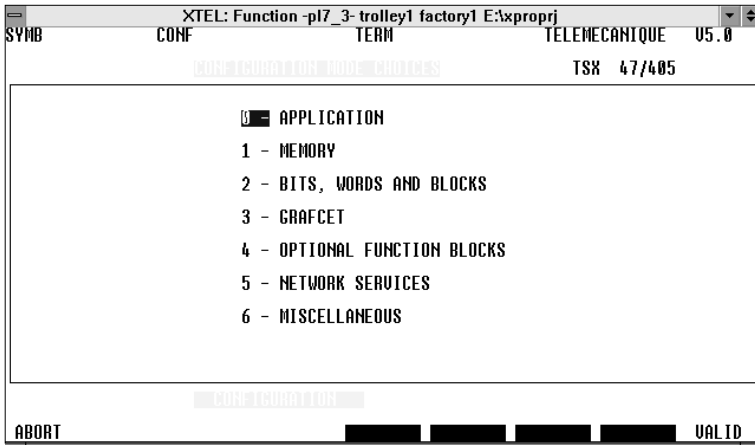
When PL7-3 is accessed, it is automatically initialized with the current station application : PL7\_3.BIN. There are three possible scenarios :

1. The PL7\_3.BIN file exists on the disk. In this event, 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 with 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 XTEL-CONF so that he can enter a configuration.

**[VALID]** Validates the modifications that have been to the configuration, which may necessitate a reconfiguration of the application. In this case, it is recommended to make a diagnostic check of the reconfiguration.

**[ABORT]** Cancels the modifications that have been made.

The Configuration mode is accessed on the Mode Selection display.



The following keys permit the configuration of :

- <0>** The application :
  - Date, name and version of the application
  - Task periods and numbers of subroutines
  - Use or not of Grafcet language
  - Cold restart conditions
- <1>** The memory structure of the application :
  - Location, capacity, type and contents of memory cartridges (V2 or V3 applications)
  - Reading of an xxx.APP file previously created with XTEL-MEM
  - Display of memory volumes defined and handled by XTEL-CONF,
  - Modifying the memory segments in the event of overflow of application code or graph spaces.
- <2>** The I/O
- <2> <3>** The bits, words and function blocks :
  - Number and parameters (function block)
- <3> <4>** The Grafcet chart (if used) :
  - Number of steps, macro-steps and pages
- <4> <5>** The optional function blocks (OFBs) :
  - Type and number
- <5> <6>** The network :
  - Number of common words and type of activity
- <6> <7>** The data link (programming port) of the PLC :
  - Speed, format and parity.



## 6.2 Configuring the Application

This screen is accessible from the previous one by selecting 0 - APPLICATION. It permits the modification of the general parameters of the application : date of modification, name and version of the application, presence and parameters of tasks, presence of Grafcet, cold restart conditions.

The coherence and syntax of the entries is checked by the terminal. If there is an error, a message is displayed.

	SYMB	CONF	TERM	TELEMECANIQUE	U5.0
				TSK	47/405
0	APPLICATION DATE	:	0 / 0 / 0		
1	APPLICATION NAME	:	TELEMECANIQUE U5.0		
2	APPLICATION VERSION	:	SYST U5.0 LIT U5.0 LAD U5.0	GRAF CET	
3	MASTER TASK				SR NUMBER : 16
4	GRAF CET		( Y / N ) : N		
5	SYNCHRONOUS TASKS		( Y / N ) : N		
6	INTERRUPT TASK (CTRL1)		( Y / N ) : N		SR NUMBER : 0
7	COLD START	:	RUN AUTO ( Y / N ) : Y	RESET Wi ( Y / N ) : N	

MODIFY

<0>

### APPLICATION DATE

Permits the dating of the creation or modification of the application. This function uses the date of the terminal.

**[MODIFY]** assigns the current date as the application date (in Local mode only).

<1>

### APPLICATION NAME

Permits the entry of the application name, which will appear on all the screens of the terminal and all the pages of the application documentation. By default, this name is TELEMECANIQUE V5.0.

**[MODIFY]** permits entry of the application name (21 characters maximum), in Local mode only.

&lt;2&gt;

**APPLICATION VERSION**

In the Local mode and for program creation, the terminal systematically proposes the following version by default :

- V5.5 if the processor selected is level V5 and if the application has more than 512 steps or uses the new segment configuration services,
- V4.0 if the processor selected is level V4
- V2.0 if the processor selected is level V2 or V3.

In Connected mode (or in Local mode with an application in the memory of the terminal), the version displayed is that read in the application. This must be initialized beforehand by transfer (in Connected mode) or by reading (in Local mode).

**[MODIFY]** permits the selection of version V2.0 or V3.0.

A V2.0 application can be transformed to V3.0, but the reverse is not possible.

&lt;3&gt;

**MASTER TASK**

Permits the definition of the period of the Master task and the maximum number of subroutines assigned to it.

**[PERIOD]** permits the entry of a new scan period (255 ms maximum). By default, the terminal proposes 50 ms.

**[MODIFY]** permits the entry of the number of subroutines assigned to the task (128 maximum). By default, the terminal proposes 16 subroutines.

&lt;4&gt;

**GRAF CET**

Permits the declaration of the use of Grafcet, in Local mode only.

**[NO]** Grafcet language not used (selected by default).

**[YES]** validates the use of Grafcet (in the Sequential section of the Master task only).

**[GRAF CET]** gives access to the Grafcet configuration screen (see Section 6.6).

&lt;5&gt;

**PERIODIC TASKS**

Indicates that the periodic tasks (Fast or Auxiliary) have been declared and their parameters set under XTEL-CONF. N indicates that these tasks are not used.

Permits the declaration of periodic tasks (Fast or Auxiliary) other than the Master task.

**[NO]** no periodic tasks required (selected by default).

**[YES]** validates the use of periodic tasks.

The use of a periodic task necessitates the declaration of :

- Its scan period
- The number of its subroutines
- The addresses of the I/O modules associated with the task

The I/O modules declared in one task cannot be declared in another.

The I/O modules that are configured but not declared in a periodic task are implicitly exchanged in the Master task.

**[CONF SYN]** displays the following screen which enables :

- configuration of the periodic tasks
- definition of the number of subroutines.

NAME	CONFIG.	SR NB	I/O MODULES ASSIGNED
0 FAST (CTRL2)	Y	0	
1 AUX0 (CTRL4)	Y	0	
2 AUX1 (CTRL5)	N	0	

**[PERIOD]** Permits definition of the period of the selected task :

- Fast task, between 5 and 254 ms
- Master task, between 5 and 255 ms
- Auxiliary tasks, between 10 and 2550 ms.

The periods of the tasks must increase according to the order of priority of the tasks : FAST < MAST < AUX.

The time bases are 1 ms for the Fast task and 10 ms for the Auxiliary tasks. The period of the task is equal to the value entered multiplied by the time base.

**[SR NB]** Permits definition of the number of subroutines assigned to the selected task (128 maximum per task).

**Note**

The code numbers of the I/O modules associated with the periodic tasks are displayed on the right of the screen.

&lt;6&gt;

**INTERRUPT TASK (CTRL1)**

Permits the declaration of the Interrupt task, which has the highest priority.

**[NO]** no Interrupt task required (selected by default).

**[YES]** validates the use of the Interrupt task.

**[MODIFY]** permits entry of 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 therefore proposes 0 subroutines by default.

&lt;7&gt;

**COLD START : RUN AUTO**

Permits definition of the cold restart conditions. This function is available for V3 to V5 applications. However, it cannot be displayed by PL7-3 V3, but only by PL7-3 V4 and V5.

**RUN AUTO**

**[YES]** selects that the application program will start automatically on a cold restart.

**[NO]** selects that the application program will not start automatically on a cold restart.

**RESET Wi**

**[YES]** selects that Wi and DWi words are reset to zero on a cold restart.

**[NO]** selects that Wi and DWi words are not reset to zero on a cold restart.

---

## 6.3 Configuration of the User Memory

---

This screen, accessible from the Configuration Mode Choices screen by selecting 1 - MEMORY, permits configuration of the application memory.

### 6.3-1 V2 or V3 Applications

Configuring the memory consists of declaring, within the limits of the memory fields of the selected processor :

- The number and location of the memory cartridges
- The capacity of the cartridges : 8, 16 or 32 Kwords
- The type of cartridges used : protected RAM or EPROM
- The contents of the cartridges, ie the length in words of the various zones :
  - Data D
  - Program P
  - Constants C.

The amount of memory space available depends on the type of PLC, since it is composed of the onboard RAM plus the user cartridges.

The total memory field must be continuous, therefore the memory cartridges must be installed without any empty slots between them, from slot 0 onwards.

For V2 or V3 processors, the order of the memory zones can be :

- Data - Program - Constants, or
- Program - Data - Constants.

However, for TSX 87-10 and TSX 87-20 PLCs, it is recommended that the memory zones are configured in the order : Program - Data - Constants.

Each memory zone is unique and continuous and can be stored in one or more cartridges. Configuration of the Program and Data zones is obligatory.

The total memory size declared must not exceed the total capacity of the cartridges, minus the program identifier (IDP), which is as follows :

- Master task : IDP = 112 words
- Fast task : IDP = 16 words
- Auxiliary task : IDP = 20 words

**Modification of the memory configuration is only possible in Local mode.**

XTEL: Function -pl7_3- stav3 project D:\xproj						
MEM. BLOCK (8 Kwords)	CART	SIZE (Kwords)	TYPE	DATA	PROG	CNST
0	ON BOARD	8	RAM	DATA: 0	PROG: 0	CNST: 0
1	0	32	EPROM	DATA: 4096	PROG: 16672	CNST: 12000
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

MEMORY

SIZE    TYPE    PROG    CNST    OBJECTS

**[CART]**

Permits the declaration of a new memory cartridge. The slot number of the cartridge is automatically defined. This key becomes effective when the reverse video strip is positioned after the last cartridge declared.

**[SIZE]** declares the size of the cartridges (8, 16 or 32 Kwords) by successively pressing the key.

This key becomes effective when the reverse video strip is positioned after the last cartridge declared.

**[TYPE]** defines the type of cartridge (RAM or EPROM) by successively pressing the key.

**[DATA] [PROG] [CNST]**

Permits the definition of the Data, Program and Constants zones respectively in the cartridge indicated by the cursor.

For information, the maximum size of the zone requested is shown in the operator input line.

Each of these keys is effective only if the declaration of the zone is compatible with the memory structure (ie with the continuity and order of the memory zones).

**[OBJECTS]** gives access to the configuration of the bits, words and function blocks (see Section 6.5).

**<DEL>** permits the deletion of the last cartridge declared, when the reverse video strip is positioned on it.

**<ENTER>** validates the screen.

---

### 6.3-2 V4 and V5 Applications

The memory configuration menu is a simplified menu which summarizes the memory map of the application. No entries are possible.

The memory configuration defines the following zones :

- Data (DATA)
- Program (PROG)
- Constants (CNST)
- Program identifier (IDP)

Each zone is divided into a space reserved for the application (STANDARD) and a space reserved for the optional function blocks (OFB).

The programmable logic memory spaces are divided into two volumes : a PL7-3 DATA volume and a PL7-3 PROG volume.

The volumes described are :

- RESERVED : attributed to PL7-3 by XTEL-MEM or XTEL-CONF
- IN USE : actually used by the logic spaces
- OCCUPIED : IN USE + free space.

By default, the memory configuration of a V4 type application occupies a volume fixed at 21 Kwords, composed of 30% for DATA and 70% for PROG. If these two volumes need to be modified, this requires :

- Use of the XTEL-MEM station memory configuration tool to define the new memory volume of the PL7-3 function and create an xxx.APP file.
- Reading of the xxx.APP file in the V4 memory configuration mode. The two memory volumes are then defined by the description given in the application file (xxx.APP).

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

XTEL: Function -pl7 3- oven3 oven3 D:\xproprj

CONF TERM TELEMECANIQUE 04.8  
47/420

WHOLE	SPACE		START ADDR.	OCCUPIED	VOLUME (WORDS)	
	PART				USED	RESERVED
DATA	STANDARD		0	3664	3659	13384
	OFB		3664	1664	1664	
PROC	STANDARD		13384	25504	7904	31224
	OFB		38888	4696	4692	
CNST	STANDARD		43584	168	161	31224
	OFB		43752	576	576	
IDP			44328	280	273	

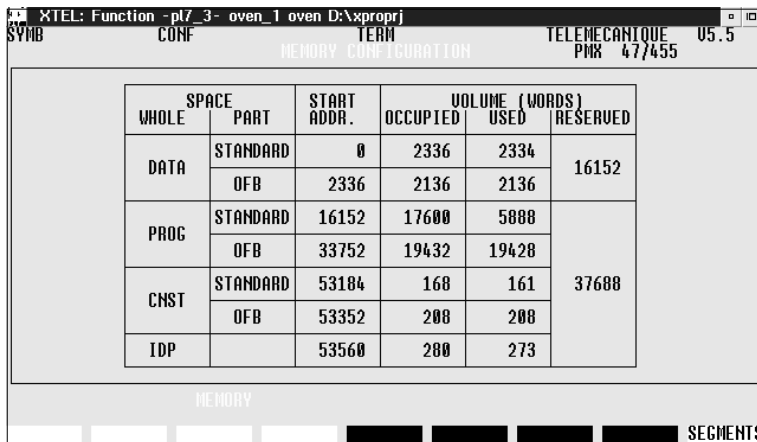
MEMORY

- [MODIFY]** Permits the entry of the xxx.APP application file. The physical volumes are modified according to the description given in this file.
- [DIR]** gives the list of existing xxx.APP files
- [STATION]** assigns the name of the current station as the file name
- [ARCHIV]** assigns the global archive name as the file name
- <ENTER>** Validates the screen.



### 6.3-3 V5 Application only

In PL7-3 V52 the CONFIGURATION mode offers a soft key for access to any code or graph segment modification, should a segment overflow occur. This is brought to the user's attention by the message "Memory Overflow".



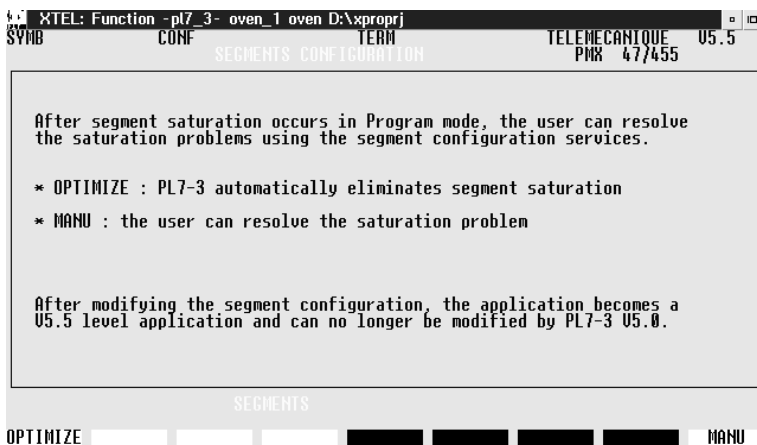
MEMORY CONFIGURATION

WHOLE	SPACE		START ADDR.	VOLUME (WORDS)		
	PART			OCCUPIED	USED	RESERVED
DATA	STANDARD	0	2336	2334	16152	
	OFB	2336	2136	2136		
PROC	STANDARD	16152	17600	5888	37688	
	OFB	33752	19432	19428		
CNST	STANDARD	53184	168	161		
	OFB	53352	208	208		
IDP		53560	280	273		

MEMORY

SEGMENTS

**[SEGMENTS]** gives access to any code or graph segment modification in the SRI, PRL, POST or MAIN modules. After pressing this key, the following screen appears :



SEGMENTS CONFIGURATION

After segment saturation occurs in Program mode, the user can resolve the saturation problems using the segment configuration services.

- \* OPTIMIZE : PL7-3 automatically eliminates segment saturation
- \* MANU : the user can resolve the saturation problem

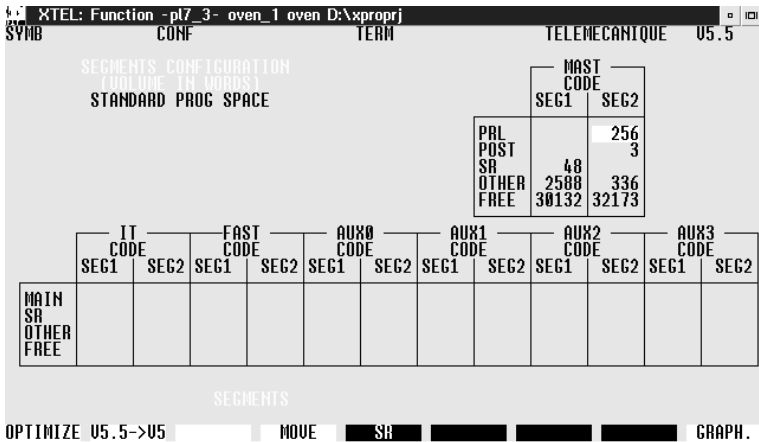
After modifying the segment configuration, the application becomes a U5.5 level application and can no longer be modified by PL7-3 U5.0.

SEGMENTS

OPTIMIZE MANU

**[OPTIMIZE]** automatically reduces overflow in the segments.

**[MANU]** gives access to the following screen for manual modification of the segment configuration.



- [OPTIMIZE]** is used to optimize displayed segments automatically (code or Graph).
- [V5.5 → V5]** configures the segments with their initial value, in order to revert to a V5.0 application.
- [MOVE]** moves the selected module from one task segment to another (code segment or graph segment ).
- [SR]** 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. Pressing [SR] displays how the SR are divided in segments 1 and 2, as well as displaying a line which allows the user to enter the new value x.
- [GRAPH.]** displays the screen with the graph segments.
- [CODE]** displays the screen with the code segments.
- <ENTER>** validates the changes made.
  - <VALID><YES>** starts a reconfiguration. If reconfiguration occurs after having modified the mapping of the memory segments, **this task can take a certain length of time**. At the end of the configuration, the code or graph modules have been moved from one segment to another. A V5.0 application which has undergone these changes then becomes, after reconfiguration, a V5.5 application.

## Note

The following screen, which is accessed by the command <Alt><T> synthetically displays the memory functions of the segments. This screen is accessible from all PL7-3 operating modes.

CODE+GRAPH	VOLUME (WORDS)		
	RESERVED	USED	FREE
SYSTEM	15800	4077	11731
TOTAL	17600	5869	11731

TASK SIZE	CODE		GRAPH	
	SEG1	SEG2	SEG1	SEG2
IT	0	0	0	0
FAST	0	0	0	0
MAST	2636	595	700	146
AUX0	0	0	0	0
AUX1	0	0	0	0
AUX2	0	0	0	0
AUX3	0	0	0	0

## Note : description reminder for 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 graph segments.

For applications lower than V5.5, the segment content is fixed and cannot be changed :

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

For more details, refer to Reference Manual PL7-3 (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**, with the following restrictions, however : only what is in bold print can be moved.

APPLICATION WITH GRAFCET, MAST TASK		APPLICATION WITHOUT GRAFCET, MAST TASK	
OR			
ANY APPLICATION, OTHER TASKS			
<b>CODE</b>		<b>CODE</b>	
<b>SEGMENT No1</b>	<b>SEGMENT No2</b>	<b>SEGMENT No1</b>	<b>SEGMENT No2</b>
GR7 INTERPRETER GR7 CODE PAGES CHART CODE XMi CODE GR7 TABLES MAIN <b>CODE FOR SRI</b>	CODE FOR ACTIONS CODE FOR TRANSITION CONDITIONS <b>PRE CODE</b> <b>POST CODE</b>	<b>MAIN CODE</b> OTHER CODES	<b>SRI CODE</b> OTHER CODES
<b>GRAPH</b>		<b>GRAPH</b>	
<b>SEGMENT No1</b>	<b>SEGMENT No2</b>	<b>SEGMENT No1</b>	<b>SEGMENT No2</b>
OTHER GRAPH. DATA <b>SRI GRAPH. DATA</b> <b>PRE GRAPH. DATA</b> <b>POST GRAPH. DATA</b>	OFB DIAG SOURCE	OTHER GRAPH. DATA <b>MAIN GRAPH. DATA</b> <b>SRI GRAPH. DATA</b>	OFB DIAG SOURCE

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

### Example

OPTIMIZE U5.5->U5      MOVE **SR**      GRAPH.

Entering new SR is not possible because the segment is nearly saturated (SEG2 = 31505 words).

OPTIMIZE U5.5->U5      MOVE **SR**      GRAPH.

The command [OPTIMIZE] is used to split the code for SR into segments 1 and 2, which provides space for the entry of new SR.

---

## 6.4 Configuring the I/O Modules

---

This screen, which is accessible from the Configuration Mode Choices screen by selecting 2 - INPUTS/OUTPUTS, allows the user to :

- Declare the code numbers of the various I/O modules used and their positions in the racks of the PLC.
- Generate an ASCII format source file (xxx.IOC) describing all the modules configured on the I/O configuration screen. This file can be accessed by the XTEL-CAD tool and can be used for the computer aided design (CAD) of electrical diagrams.

### Restrictions

- The I/O modules can only be configured in Local mode (and requires the application to be reconfigured).
- The intelligent modules can be installed and configured only in racks that are equipped with the complete bus.
- The 24 and 32-point I/O modules are seen by the processor as two 16-point modules, one located in slot x of an even numbered rack and the other in slot x of the associated odd numbered rack. It is therefore necessary to declare these modules in both of the racks.

Example : A 32-point input module located in slot 2 of rack 0 must declared in slot 2 of racks 0 and 1. The associated inputs are respectively I02,0 to I02,F and I12,0 to I12,F.

For the TSX 47-30 and TSX 67-20 PLCs and for all V4 PLCs, the basic racks are single racks with double addressing, which can receive the 24 and 32-point modules. The racks 1 and 3 are therefore virtual racks.

During the declaration of the I/O modules, the following totals appear at the top of the screen :

- The number of bits associated with the discrete and intelligent I/O modules
- The number of words associated with the intelligent modules.

XTEL: Function -pl7\_3- oven2 factory3 D:\xproprj

MEM CONF TERM Odiag Opc102 Onum

BITS USED : 400 WORDS USED : 80 TSX 67/410

rack7								
rack6	56	53	649	665	56	53		732
rack5								
rack4		729	56	53	697	56	53	53
rack3								
rack2	56	53	56	53	56	53	56	53
rack1								
rack0			13				56	53
MODULE	0	1	2	3	4	5	6	7

INPUTS/OUTPUTS

CO/TY/TA CODE TASK NEXT MOD TO LAST ALL RACK REST.COD NEXTPAGE ...

INPUTS/OUTPUTS

HELP DEL ALL WRITE READ ...

- [CO/TY/TA]** Modifies the display so as to show the **code** of the modules (eg 733, 32, 24, etc.), the **type** of the modules (eg CPL, 8IN, 8OUT, etc) or the **task** in which they are declared (eg MAST, AUX0, etc).
- [CODE]** Permits entry of the module code number in the operator input line.
- [TASK]** Permits modification of the name of the task associated with the current module (Master task by default). The appropriate soft key appears for each task configured :  
**[MAST] [AUX0]**, etc.  
 These keys permit the definition of the task in which the current module (in reverse video) is declared (Master task by default).
- [NEXT MOD]** If the display shows the code or type of the module, this key copies the code of the current module (in reverse video) onto the next one. If the display shows the task, this key assigns the task of the current module to the next one.
- [TO LAST]** If the display shows the code or type of the module, this key copies the code of the current module onto all the subsequent modules in the rack. If the display shows the task, this key assigns the task of the current module to all the subsequent modules in the rack.

---

**[ALL RACK]** If the display shows the code or type of the module, this key copies the code of the module onto all the modules in the rack. If the display shows the task, this key assigns the task of the current module to all the modules in the rack.

**[REST COD]** Assigns the code number that was memorized in a help menu by using the <ENTER> key, to the selected location.

**[NEXTPAGE]** Alternately displays the configuration of racks 0 to 7 or of racks 8 to F.

<DEL> Deletes the code number of the selected module (in reverse video).

<←><→> Permits the selection of a slot in a rack.

<↑><↓> Moves the cursor from one rack to another.

<ENTER> Validates the screen.

[./..] Displays the rest of the soft keys relating to the basic screen.

**[HELP]** Accesses the Help and I/O selection menus.

**[INTRO]** accesses the introductory screen relating to the help screens.

**[FAM LIST]** displays the list of families of modules.

<↑><↓> moves the cursor onto the families listed in the Help menu.

<ENTER> displays the list of all the modules related to the selected family.

**[I/O LIST]** displays the list of modules (Discrete I/O and modules).

<↑><↓> move the cursor onto the modules listed in the Help menu.

<ENTER> memorizes the code of the module indicated by the cursor so that it can be copied, by using the **[REST COD]** soft key, on the I/O configuration screen.

**[RACK]** displays in the form of a table, the contents of a rack : slot number, reference number, designation and code number of the modules.

**[PREVIOUS]** displays the previous screen for : family of modules, list of modules and rack.

**[NEXT]** displays the next screen for : family of modules, list of modules and rack.

<CLEAR> or <QUIT> quits the I/O Help function.

**[DEL ALL]** Deletes the codes for the current I/O configuration of the module.

- 
- [WRITE]** Permits the writing of the entire I/O configuration in a xxx.IOC file.
- [STR.NAME]** permits entry of the name of the xxx.IOC file to be written.
  - [DIR]** permits access to the xxx.APP directory of the current station where the xxx.IOC files are stored.
  - <↓><↑>** move the cursor onto the name of the selected file.
  - [SEARCH]** permits the entry (8 characters maximum) of the name (without its suffix) of the file to be searched for. After validation only the files with the name entered are displayed.
  - [DELETE]** deletes the selected file (after confirmation).
  - [STATION]** assigns the name of the current station as the name of the file.
  - [ARCHIV]** assigns the global archive name as the name of the file.
  - <ENTER>** validates the selection. If the file already exists, the new data will overwrite the existing contents of the file, after confirmation by **[YES]**.
- [READ]** Permits the reading of the I/O configuration from a xxx.IOC file in 3 different modes : AUTO/MANU/ADD.
- [STR.NAME]** permits entry of the name of the xxx.IOC file to be read.
  - [DIR]** permits access to the xxx.APP directory of the current station where the xxx.IOC files are stored.
  - <↓><↑>** move the cursor onto the name of the file to be read.
  - [SEARCH]** permits entry of the name of the file to be searched for.
  - [DELETE]** deletes the selected file (after confirmation).
  - [STATION]** assigns the name of the current station as the name of the file.
  - [ARCHIV]** assigns the global archive name as the name of the file.
  - <ENTER>** validates the selection and starts the reading of the selected file.
  - [MODE]** permits selection of the mode of reading the selected file : AUTO/MANU/ADD (see next page).

**Note**

The xxx.IOC file is proposed in the screens STORE and EXIT so that the user can save this file at any time, and in particular when quitting the PL7-3 function.



---

## Modes of reading the xxx.IOC file

### AUTO mode

This mode permits the replacement of all the modules of the I/O configuration by those read in the xxx.IOC file. The user can stop the reading at any time by pressing the [ABORT] key.

If an error is detected by the configuration monitoring system, the mode of reading changes from automatic to manual and :

- The code of the module read is displayed in the operator input line
- The code of the module configured is displayed on the screen in reverse video.

After the error is corrected, reading continues in AUTO mode.

### MANU mode

This mode allows the user to modify the existing configuration step by step according to the modules read in the xxx.IOC file and to the existing modules in the configuration. Each time that the system detects a difference between the code of the module read and the code of the module configured, the latter is displayed on the screen in reverse video, while the former is displayed in the operator input line. The user can select any of the following options:

**<ENTER>** Validates the reading. In this case, the module read in the xxx.IOC file becomes the current module, on condition that it is accepted by the configuration monitoring system.

**<CLEAR>** Deletes the code of the module read from the operator input line. The code of any other module can then be entered, on condition that it is accepted by the configuration monitoring system.

**<QUIT>** Leaves the module configured unchanged.

**[ABORT]** Stops the reading of the xxx.IOC file.

**[AUTO]** Switches to the automatic reading mode.

### ADD mode

This mode allows the user to add modules to the slots in the configuration that are not yet occupied. In the case where the module read in the xxx.IOC file finds the slot already occupied by an existing (configured) module, it is the existing (configured) module that prevails.

**Example**

Code of the module configured	Code of the module read in xxx.IOC	Mode	New module code
43	0 (*)	AUTO	0 (*)
		ADD	43
		MANU	by selection
43	35	AUTO	35
		ADD	43
		MANU	by selection
0 (*)	35	AUTO	35
		ADD	35
		MANU	by selection

(\*) No module configured in this slot.



## 6.5 Configuring the Bits, Words and Function Blocks

### 6.5-1 Procedure

This screen, accessible from the Configuration Mode Choices screen by selecting 3 - BITS, WORDS AND BLOCKS, permits the configuration of :

- The number of bits and words
- The number of Grafcet step active time values
- The number of function blocks of each type

Any modification to the numbers of these data objects necessitates a reconfiguration of the application (accessible in Local mode).

The upper part of the screen indicates the maximum number of objects and the number used.

For a V2 or V3 application, the maximum number of objects corresponds to the size of the memory spaces configured on the memory configuration screen (see Section 6.3). 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 permits the definition of the objects used. The number of these objects corresponds to the sum of the objects declared in each of the columns BITS, WORDS, FUNCTION BLOCKS.

XTEL: Function -pl7_3- trolley1 factory1 E:\xproj									
SYMB		CONF		TERM		TELEMECANIQUE		U5.0	
		DATA STANDARD		NUMBER MAX		NUMBER OCCUPIED			
		CST STANDARD		6448		771			
		BITS		13624		161			
				4096		416			
BITS			WORDS			FUNCTION BLOCKS			
0	System (SY):	128	System (SW):	128	Timer (T):	0			
1	Internal (I):	256	Internal (W,DU):	128	Monostable (M):	0			
2	In/Out (I/O):	0	Constant (CW,CDW):	128	Counter (C):	0			
3	Remote In/Out :	0	Common A (COM):	0	Register (R):	0			
4	OFB Objects :	0			Text (TXT):	0			
5									
6									
7									
8	Grafcet (X):	0	Grafcet (X,U):	0					
9	(XM):	0	Grafcet Data :	0					
BITS WORDS BLOCKS									
MODIFY		CONTENT							

---

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

- BITS (number)
  - Internal bits (B)
- WORDS (number)
  - Internal words (W,DW)
  - Constant words (CW,CDW)
  - Grafcet step active time values (Xi,V)
- FUNCTION BLOCKS (number and parameters)
  - Timers
  - Monostables
  - Counters
  - Registers
  - Text blocks

**[MODIFY]** Permits modification of the number of objects indicated by the cursor.

**[CONTENT]** Accesses the parameters of the function block indicated by the cursor.

<↑><↓> Move the cursor up or down the column. Only the configurable objects are accessible.

<←><→> Move the cursor from one column to another. Only the configurable objects are accessible.

<↘> Moves the cursor onto all the configurable objects in the column one by one, and then onto those in the next column.

<ENTER> Validates the screen.

---

## 6.5-2 Configuring the Function Blocks

All the function blocks declared are configured by default with the parameters shown below, which can be read and modified in Local mode.

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

- The time bases of timers and monostables
- The length of registers.

Function blocks	Default parameters
Timers	PRST = 9999, M = Y, TB = 1mn
Monostables	PRST = 9999, M = Y, TB = 1mn
Counters	PRST = 9999, M = Y
Registers	LENGTH = 16, TYPE = LIFO

**[CONTENT]** Displays the parameters of the selected function block.

**[PRST]** permits modification of the preset value.

**[M]** permits (in succession) the authorization (Y), or not (N), of the modification of the preset value in the Adjust mode.

**[TB]** permits (in succession) the selection of the following time bases: 10ms, 100ms, 1s and 1mn. The number of timers and the number of monostables configured with a time base of 10ms must not exceed 8.

**[NEXT]** copies the parameters of the current function block (in reverse video) into the next one.

**[TO LAST]** copies the parameters of the current function block into all the subsequent ones.

**[LENGTH]** permits the declaration of the length of the current register (0 to 16). Each element of the register is a word of 16 bits.

**[TYPE]** permits (in succession) the declaration of the type of register : LIFO (Last In, First Out) or FIFO (First In, First Out).

**<ENTER>** Validates the screen.

### 6.5-3 Configuring the Text Blocks

All the Text blocks declared are configured by default with the following parameters, which can be read and modified in Local and Connected modes.

Function blocks	Default parameters
Text blocks	NET/LOCAL = LOCAL, TYPE = CPL, ADR. MODE = DIRECT, RECEPT. LENGTH = 0 M = H'00FF', L = 0, C = H'0000'

**[CONTENT]** Displays the parameters of the selected Text block :

**[NET/LOC]** selects (in succession) the type of communication :

- LOC : exchange of data with a locally connected PLC
- NET : exchange of data with a network connected PLC

**[TYPE]** selects (in succession) the type of exchange :

- CPL : exchange of data with an intelligent module
- TER : exchange of data with a peripheral through the programming port of the PLC
- SYS : exchange of requests with the system
- TXT : exchange of data with a TXT block in another PLC
- TLG : exchange of telegrams (priority messages) with a TLG block in another PLC

**[MODE]** selects (in succession) the type of addressing :

- DIRECT : the table to be transmitted or received is defined by its start address and its length
- INDIRECT : the table to be transmitted and received are defined by an addressing table of 6 words

**[BUF ADDR]** permits entry in the operator input line of :

- 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.

---

**[LENGTH]** permits entry (in direct addressing only) of 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

**[SYMBOL]** displays either the mnemonic symbols or the address variables of the Text block.

**[TXT,A]** permits entry, for a NET type exchange, of the number of the destination station.

**[TXT,T]** permits entry, for a TXT type exchange, of the number of the destination Text block.

**[TXT,M]** permits entry, for a CPL type exchange, of the address of the destination module.

**[TXT,L]** permits entry, in direct addressing, of the maximum length of the transmission table (in bytes). The maximum length is the same as that of the reception table.

**[TXT,C]** permits entry of the code of the type of work to be effected (in an exchange with a CPL or SYS type block).

**<ENTER>** Validates the modifications.





---

<b>[MACRO]</b>	Permits declaration of the number of macro-steps.
<b>[STEP]</b>	Permits declaration of the number of steps in the main chart or selected macro-step. <b>Reminder</b> : The total number of steps declared must not exceed 1024 (V5.5 application) or 512 (V5.0 application maximum).
<b>[PAGE]</b>	Permits declaration of the number of pages in the main chart or selected macro-step.
<b>[NEXT]</b>	Copies the configuration of the current macro-step into the next one.
<b>[TO LAST]</b>	Copies the configuration of the current macro-step into all the subsequent ones on the screen.
<b>[ACT STEP]</b>	Permits declaration of the number of simultaneously active steps. This number appears in the Active Step Table at the top of the screen.
<b>[TRANSIT]</b>	Permits declaration of the number of simultaneously enabled transitions. This number appears in the Associated Transition Table at the top of the screen.
<b>&lt;ENTER&gt;</b>	Validates the modifications.

### Remarks

- 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 that is assigned to it, a macro-step occupies 3 bits in the bit memory (2 bits for the IN and OUT steps and 1 bit for the activity of the macro-step).
- A word containing the active time value may be assigned to each step in the data memory.

## 6.7 Configuring the Optional Function Blocks

This screen is accessible (for a V4 or V5 application only) by selecting OPTIONAL FUNCTION BLOCKS on the Configuration Mode Choices screen. It permits the configuration of the optional function blocks (OFBs) to be used in the application as follows :

- Selection of the types of OFBs to be used in the application, from the OFBs available in the terminal.
- Declaration of the number of OFBs of each type selected.

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

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

XTEL: Function -pl7_3- trolley1 factory1 E:\xproj							
SYMB		CONF		TERM		TELEMECANIQUE U5.0	
TOTAL VOLUME (W)		DATA: 1520		CNST: 256		PROG: 7248 SYST: 1277	
TYPE	NUMBER	VERSION	FAMILY	TYPE	NUMBER	VERSION	FAMILY
MSIT	8	U 5.0	GRAFCET				
PCL	2	U 4.5	PCL				
ANALD	2	U 5.0	PCL				
VOLUMES (WRDS): MSIT		DATA: 768		CNST: 192		PROG: 2216 SYST: 224	
MODIFY NEW OFB SEARCH DEL <b>NEXT</b> TO LAST							

**[MODIFY]** Permits modification of the number of OFBs of the type indicated by the cursor.

- 
- [NEW OFB]** Displays the types of OFBs available in the terminal, so that those to be used in the application can be selected.  
The types of OFBs available in the terminal are displayed by family (communication, axis control, etc).
- [PREV FAM]** displays the previous family.
- [NEXT FAM]** displays the next family.
- [INS]<INS>** selects the type of OFB indicated by the cursor, so that it can be used in the application. It then appears in reverse video in the column BLOCK VERSION FAMILY.
- [DEL]<DEL>** deletes the type of OFB indicated by the cursor. It then no longer appears in reverse video in the column BLOCK VERSION FAMILY.
- [NEXT]** copies the command applied to the current type of OFB (selection or deselection) onto the next type of OFB.
- [TO LAST]** copies the command applied to the current type of OFB ([INS] or [DEL]) onto all the subsequent types of OFBs.
- [FAMILY]** permits the selection of another family to be displayed.
- [PREV SCR]** displays the previous screen of the same family.
- [NEXT SCR]** displays the next screen of the same family.
- <ENTER>** validates the selections.
- [SEARCH]** Permits a search for a type of OFB by entering its name.
- [DEL]** Deletes the type of OFB indicated by the cursor.
- [NEXT]** Assigns the number of OFBs of the current type to the next type.
- [TO LAST]** Assigns the number of OFBs of the current type to all the subsequent types displayed on the screen.
- <ENTER>** Validates the screen.

## 6.8 Configuring the Network Services

This screen is accessible by selecting NETWORK SERVICES on the Configuration Mode Choices screen. It permits the configuration of the number of common words and their 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 multi-network 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 common words 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 FIPWAY module is therefore modified.

COUPLER	NET TYPE	RACK MOD	COM WORDS	ACTIVITY
A	MAPWAY	0 7	8	READ ONLY
B	FIPWAY		4	READ AND WRITE
C				
D				

**[MODIFY]** Permits modification of the current configuration.

**[WORDS]** selects (in succession) the number of common words associated with the selected network module : 0, 4, 8, 16, 32 or 64.

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

**[ACTIVITY]** selects (in succession) the type of activity of the common words :

- Inactive
- Read/write
- Read only (for V4 or V5 applications only)

**<ENTER>** validates the modifications.

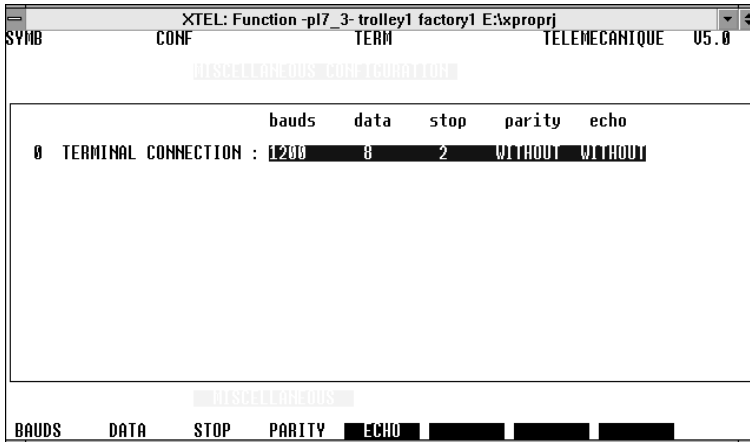
**<ENTER>** Validates the screen.

---

## 6.9 Miscellaneous Configurations

---

This screen, accessible from the Configuration Mode Choices screen by selecting MISCELLANEOUS, permits configuration of the parameters of the programming port of the PLC when the latter is not connected to a terminal or to Telemecanique TSX7 software.



**[BAUDS]**     Selects (in succession) the transmission speeds : 300, 600, 1200, 2400, 4800 or 9600 bauds.

**[DATA]**     Selects (in succession) the number of data bits : 7 or 8.

**[STOP]**     Selects (in succession) the number of stop bits : 1 or 2.

**[PARITY]**   Selects (in succession) the parity :

- EVEN
- ODD
- WITHOUT.

**[ECHO]**     Selects (in succession) transmission WITH or WITHOUT echo.

**<ENTER>**   Validates the screen.

## 6.10 Modifying the Configuration in Connected Mode

The structural parameters of the application that are necessary to debug a program in the PLC's memory, and that do not cause any incoherence, can be modified in the Connected mode.

The configuration parameters affected by this function are :

- The task periods. For a V5 application, these parameters are not accessible in PL7-3, but in XTEL-CONF.
- The function block parameters, except for the time bases of timers and monostables and the lengths of registers.

### Modifying the task times

First, the PLC must be stopped, then :

- Select the Configuration mode
- Modify the period of the selected task(s) (see Section 6.2)
- Validate the new configuration (see Section 6.11)
- Transmit the new configuration to the PLC by setting system bit SY0 to 1
- Set the PLC to RUN.

### Modifying function block parameters

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

- In the Configuration mode, with the PLC stopped
- In the Debug and Program modes, with the PLC running or stopped.

The procedure differs according to the type of language :

#### In Literal (with the PLC stopped) (1),

After positioning the line pointer on the Literal statement containing the function block(s) to be modified :

- <ZOOM>** Selects the "ZOOM 1" mode and the first zoomable function block is highlighted.
- <ZOOM>** A second action on this key calls up the configuration screen of the selected function block. The parameters of the selected function block can then be modified.
- <ENTER>** Validates the modification.
- <CLEAR>** Returns the terminal to the previous operating mode (Debug or Program).

(1) Except PL7-3 GLD

---

### In Ladder language (with the PLC stopped) (1)

Select the Debug mode and position the pointer on the ladder network containing the function block(s) to be modified :

- [PROG]**      Accesses the program  
**[MODIF]**      Permits modification.

Then position the pointer on the function block to be modified :

- <ZOOM>**      Accesses the configuration parameters of the selected function block (in succession). The parameters can then be modified  
**<ENTER>**      Validates the modifications.  
**<CLEAR>**      Successively pressing this key returns the terminal to the Debug mode.

#### Important :

- **The modifications that are made to the configuration in Connected mode do not become effective until the PLC is initialized.**
- **The initialization must be carried out with the PLC stopped. To initialize the PLC, select the Data or Debug modes and then set system bit SY0 to 1. This is equivalent to a cold restart.**
- **System bit SY0 is AUTOMATICALLY reset to 0 by the system.**

---

(1) Except PL7-3 GLT

## 6.11 Reconfiguration Procedure

### 6.11-1 General

The reconfiguration procedure permits, in Local mode, the validation of a new application configuration. After reconfiguration, all the modifications declared are transferred to the internal memory of the terminal.

The reconfiguration procedure must be carried out 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 zone to write the program.

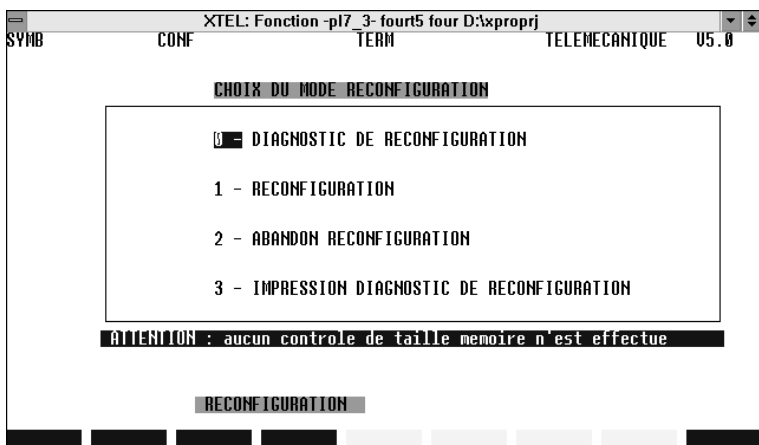
**<Alt><T>** Displays (irrespective of the mode) the amount of memory space available for writing the user program

### Remarks

- Before making any modification to the configuration, users are recommended to save the existing application by transferring it to a diskette.
- Before validating a reconfiguration, users are recommended to carry out the reconfiguration check proposed by the terminal.

The reconfiguration function is accessible from the Configuration Mode Choices screen by pressing the soft keys : [VALID] [YES].

The [ABORT] key cancels the modifications.



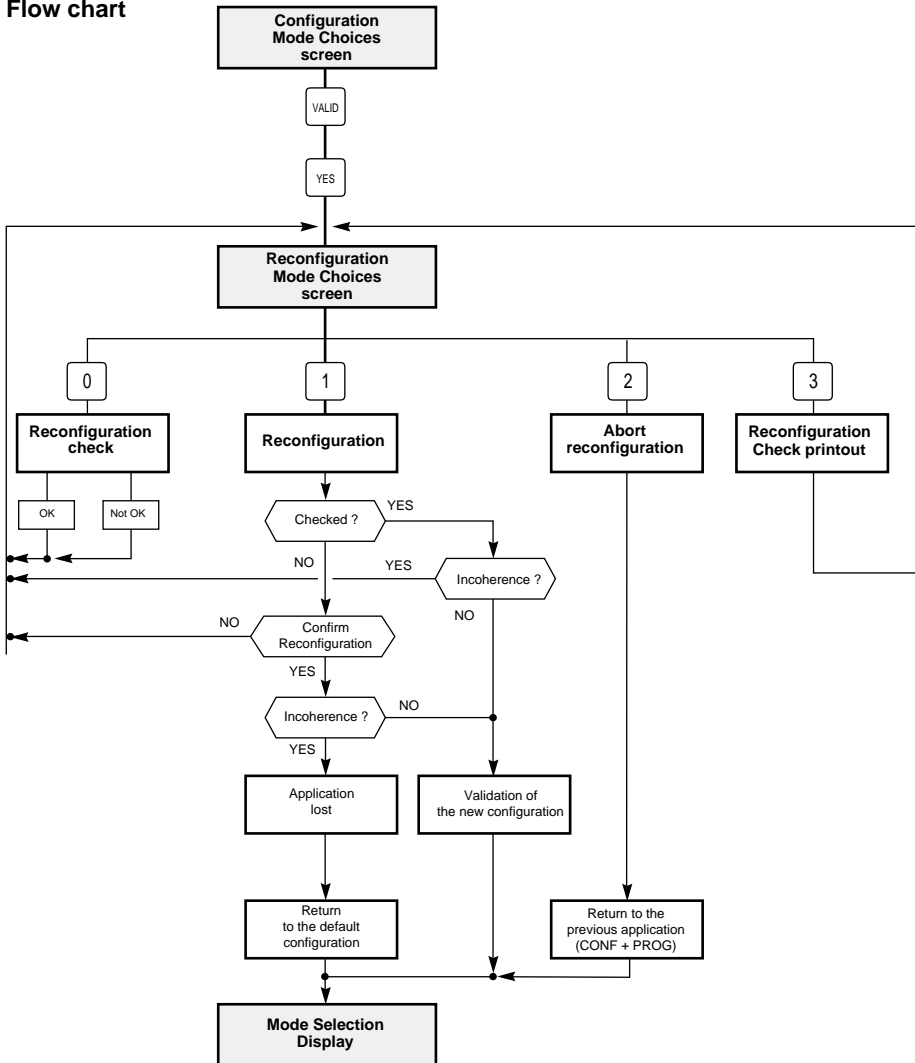


The Reconfiguration screen proposes four options :

- 0 - RECONFIGURATION CHECK (Section 6.11-2)
- 1 - RECONFIGURATION (Section 6.11-3)
- 2 - ABORT RECONFIGURATION (Section 6.11-4)
- 3 - RECONFIGURATION CHECK PRINTOUT (Section 6.11-5)

For safety, the terminal automatically proposes the Reconfiguration Check.

**Flow chart**



### 6.11-2 Reconfiguration Check

This function checks the coherence between the user program and the new configuration. It is accessible by the keys <0><ENTER>.

At the end of the check, if the new configuration is correct, the terminal displays the message : 16133 CHECK OK. The user can then select 1 - RECONFIGURATION and effectively store the new configuration in the internal memory of the terminal.

If the check is not OK, the checking procedure is interrupted and the error is displayed on the screen with the appropriate error message (for example : 2121 FUNCTION BLOCK NOT CONFIGURED).

**[NEXT]** Restarts the check until the next error is encountered, or until the check ends by returning to the Reconfiguration mode menu with the message : 16134 PROG. AND CONF. MISMATCH.

**[ABORT]** Stops the check and returns to the Reconfiguration mode menu with the message : 16134 PROG. AND CONF. MISMATCH.

If an error is detected, or if the check is aborted, reconfiguration is prohibited. The message 16135 MISMATCH DETECTED is displayed if the user tries to do a reconfiguration.

The inconsistencies in the application must then 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.

---

### 6.11-3 Reconfiguration

This function permits the transfer of the new configuration into the memory reserved for the application. It is accessed by using the keys <1><ENTER>.

Two options are available :

- Reconfiguration without the check
- Reconfiguration after the check.

#### Reconfiguration without the check

The terminal requests confirmation :

**[NO]** Returns to the Reconfiguration mode menu so that one of the other options can be selected.

**[YES]** Starts the reconfiguration procedure.

---

If the reconfiguration is effected correctly, the terminal returns to the Mode Selection display with the message 16136 RECONF OK.

If there is a discrepancy between the program and the new configuration, it is displayed on the screen so that it can be noted.

**[NEXT]** Continues the procedure until the next discrepancy is found, or until the following final message is displayed : RECONFIGURATION ABORTED.  
QUIT : RETURN TO MAIN MENU WITH TERMINAL INITIALIZE.

**The application is lost**

### **Reconfiguration after the check**

**<ENTER>** Starts the reconfiguration procedure.

If the discrepancies detected during the check have not been corrected, reconfiguration is impossible. Otherwise, the reconfiguration is effected and the terminal returns to the Mode Selection display with the message 16136 RECONF OK.

If reconfiguration is performed after mapping modification of the memory segment, **this operation can take some time**. On completion of the configuration, the code or graph 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.

---

### **6.11-4 Abort Reconfiguration**

This function permits cancellation of the reconfiguration procedure, which cancels all the modifications and returns to the old configuration. It is accessed by selecting <2><ENTER>.

**[YES]** Confirms the cancellation of the reconfiguration.

When this operation is completed, the terminal returns to the Mode Selection display.

---

### **6.11-5 Reconfiguration Check Printout**

This function permits printing to a file or to a printer (depending on the choice made in the Terminal mode) of all the errors generated by the reconfiguration check. It is accessed by selecting <3> <ENTER>.



<b>Section</b>	<b>Page</b>
<b>7 Program Mode : Grafcet Language</b>	
Contents	7/1
7.1 Selecting the Task and Module	
7.2 Reading the Program	
7.3 Zoom Function	
7.4 Entering or Modifying a Program	
7.5 Grafcet Window Mode	



---

C



Sub-section	Page
<b>7.1 Selecting the Task and Module</b>	7/2
7.1-1 Selecting the Program Address	7/2
7.1-2 MODULE SELECTION screen	7/4
7.1-3 LIST OF MODULES screen	7/5
7.1-4 Storage and Retrieval of the Table of Modules	7/6
<b>7.2 Reading the Program</b>	7/7
7.2-1 Reading the Pages of a Grafcet Module	7/7
7.2-2 Searching for a Step in a Module	7/8
7.2-3 Storage and Retrieval of a Grafcet Module	7/9
<b>7.3 Zoom Function</b>	7/10
7.3-1 Zooming on a Macro-step	7/11
7.3-2 Zooming on a Transition	7/11
7.3-3 Zooming on a Step	7/12
<b>7.4 Entering or Modifying a Program</b>	7/13
7.4-1 Procedure	7/13
7.4-2 Entering a Linear Sequence	7/16
7.4-3 Entering Simultaneous Sequences	7/17
7.4-4 Entering Connectors	7/17
7.4-5 Drawing Pathways	7/18
7.4-6 Entering Comments	7/20
7.4-7 Validating a Page	7/21
7.4-8 Modifying the Elements in a Page	7/21
7.4-9 Deleting a Module, a Page or an Element	7/23
<b>7.5 Grafcet Window Mode</b>	7/25
7.5-1 Presentation	7/25
7.5-2 Defining and Positioning the Window	7/25
7.5-3 Moving Graphic Elements	7/26
7.5-4 Deleting Graphic Elements	7/27

This section ends on page 7/28

---

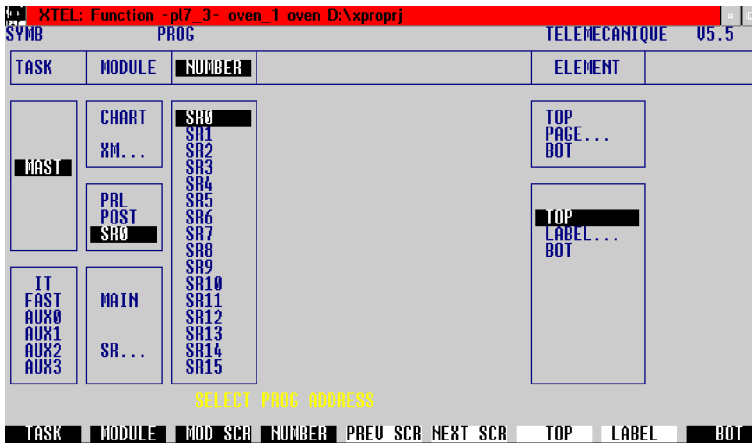
## 7.1 Selecting the Task and Module

---

### 7.1-1 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 element required). Remember that to enter a Grafcet program, the Grafcet language must have been declared in the Configuration mode.

When the Program mode is selected on the Mode Selection display, the Program Address Selection screen is displayed :



#### Selection of the task

**[TASK]** Permits selection of the task.  
Only the Master task can be used for programming in Grafcet.

### Selecting the module

**[MODULE]** Permits selection of the required program module of the Master task. The modules available are :

- CHART : Main chart
- XM... : Macro-step
- PRL : Pre-processing
- POST : Post-processing
- SR... : Subroutine

**[NUMBER]** this key appears only if an XM or SR module has been selected; it permits entry of the number of the macro-step or subroutine. <→> in the MODULE column, if the cursor is positioned on an SR... or XM...line, then this key displays the list of macro-steps or subroutines configured in the application, in the NUMBER column.

**[MOD SCR]** Accesses the MODULE SELECTION screen which constitutes the call screen for the elementary modules and macro-modules of the application.

**[NAME]** Accesses the LIST OF MODULES, so that the required module (task + module) can be entered directly by its name.

**<Alt><C>** Permits entry of the comment associated with the selected module.

**<↑><↓>** Moves the cursor up and down in the selected column.

### Selecting the element

**[TOP]** Selects (depending on the type of module) :

- The first program element in the modules PRL, POST or SR
- The first Grafcet page in the modules CHART or XM.

**[BOT]** Selects (depending on the type of module) :

- The last program element in the modules PRL, POST or SR
- The last Grafcet page in the modules CHART or XM.

**[PAGE]** If the modules CHART or XM are selected, this key permits direct access to the required Grafcet page by entering its number.

**[LABEL]** If the modules PRL, POST or SR are selected, this key permits direct access to the required program element by entering its label number.

**<ENTER>** Validates the selections.



---

## 7.1-2 MODULE SELECTION screen

This screen contains the list of elementary modules and macro-modules contained in the application (for more detailed information, see Section 18.11 V5 Module). It is used for direct access :

- if an elementary module is selected, to the start of the code for this module
- if a macro-module is selected, to the graphic view of this macro-module.

It shows the following information for each elementary module :

- Name of the macro-module with which it may or may not be associated
- Name of the elementary module
- Protection level :
  - 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 transmitted 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 file table. For this reason it is stored in the PLC.

The elementary 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 elementary modules which are not associated to a macro-module are grouped together at the beginning of the first screen.

MACRO MOD.	BASIC MOD.	R/RW	TASK	MODULE	FROM	VERS.	DATE
	MOTOR	RW	MAST	PRL	MOD	U0	11-03-93

MODUL SELECTION

PREU SCR NEXT SCR ADA SCR

- [PREV]** Displays the previous screen of the table.
- [NEXT]** Displays the next screen of the table.
- [ADR SCR]** Returns to the selection screen for the program address.
- <↑>< >** Move the cursor, for selection of an elementary module or macro-module.
- <ENTER>** Confirms the selection.
- <DEL>** Deletes, after confirmation, the elementary module or the macro-module indicated by the cursor. A write-protected elementary module cannot be deleted if it is part of a macro-module. In this case, delete the macro-module. If the elementary module is not part of a macro-module, it can be deleted.

### 7.1-3 LIST OF MODULES screen

This screen is accessed by pressing the [NAME] key on the PROGRAM ADDRESS SELECTION screen. It permits the selection of a module (task + module), the entry of a module name, and the entry of the associated comments.

TASK	MODULE	NAME	COMMENT
MAST	CHART	OVENS	U2 01-21-91 MARTIN
MAST	RNO	STRT_CVC	Start Cycle oven3
MAST	MAIN	OVENS	U0 01-21-91
MAST	SBO	INST_BUR	Instructions burners
AUX0	MAIN	TEMP_CHK	Temperature checking burners

**LIST OF MODULES**

NEXT SCR | PREV SCR | MODIFY | DUPL | CLEAN | WRITE | READ

---

<b>[NEXT]</b>	Displays the next screen of the table.
<b>[PREV]</b>	Displays the previous screen of the table.
<b>[MODIFY]</b>	In the line indicated by the cursor, successive pressures on this key permit entry or modification of the task, module, name and comment. As each column is entered or modified, it must be validated by <ENTER>.
<b>[DUPL]</b>	Copies the line indicated by the cursor into the operator input line so that it can be modified and thus create a new line in the table.
<b>[CLEAN]</b>	Deletes the whole of the table (after confirmation).

---

#### 7.1-4 Storage and Retrieval of the Table of Modules

<b>[WRITE]</b>	Enables the table of modules to be stored in a "source" file on disk. This xxx.MOD file, which is stored in the PL7_3\MOD directory, permits the table to be reused in another application and its integration in a module base.
<b>[READ]</b>	Permits the retrieval of the "source" file containing the table of modules.

For both storage and retrieval, the following keys are displayed :

<b>[STR NAME]</b>	Permits entry of the name of the file and display of the following keys :
<b>[DIR]</b>	lists all the files stored in the PL7_3\MOD directory.
<b>&lt;↑&gt;&lt; &gt;</b>	move the cursor onto the name of the desired file.
<b>[SEARCH]</b>	permits entry (8 characters max.) of the name of the file (without the suffix) to be searched for. After validation, only those files having the name that was entered are displayed.
<b>[DELETE]</b>	deletes the selected file (after confirmation).
<b>[STATION]</b>	assigns the name of the current station as the name of the file.
<b>[MODULE]</b>	assigns the name of the current module as the name of the file.
<b>[MODE]</b>	Defines the mode of retrieval (in read only) :
	AUTO : Total retrieval, stopping only on an error,
	MANU : Manual retrieval with confirmation of each element read.

## 7.2 Reading the Program

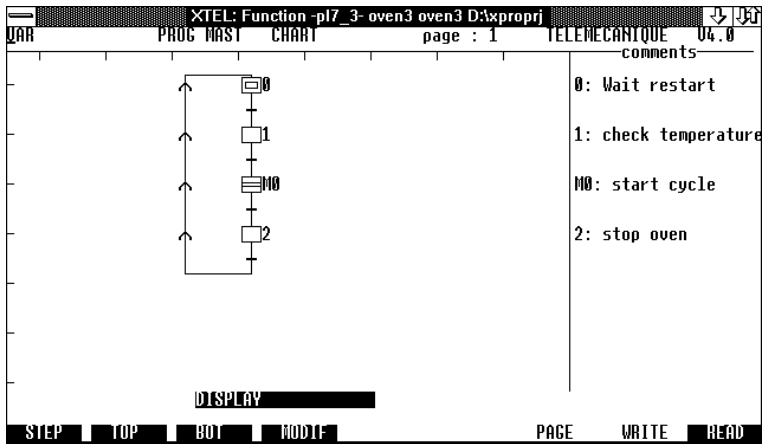
### 7.2-1 Reading the Pages of a Grafcet Module

#### Grafcet DISPLAY screen

This screen appears when the selection of the Program Address (task, module, page) is validated.

A Grafcet module is composed of several pages (8 maximum, the number required being declared in the Configuration mode). Each page, numbered from 1 to 8, is shown as one screen.

- <↑><> Permit the reading of the module page by page, by displaying the previous page or the next page.
- <CLEAR> Returns to the Program Address selection screen.

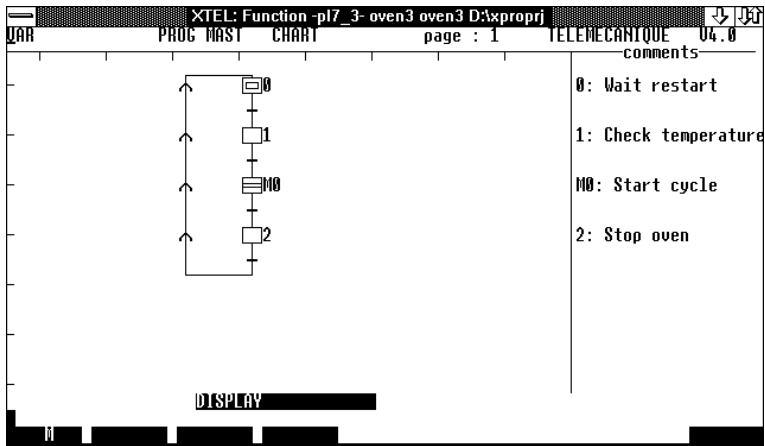


#### Direct access to a page

Irrespective of the page displayed, direct access to any other page of the module can be obtained as follows :

- [TOP]** Displays the first page of the module (page 1)
- [BOT]** Displays the last page of the module
- [PAGE]** Permits entry of the required page number (1 to 8). After validation by <ENTER> the page is displayed.  
If the page number requested is greater than the number of pages configured, an audio warning sounds and a new number can then be entered.

## 7.2-2 Searching for a Step in a Module



**[STEP]** Permits a search for a step or a macro-step in a module (CHART or XM) :

### Searching for a step

Enter the number of the step

**<ENTER>** Validates the entry and displays the page containing the step. If the step searched for does not exist in the module, a message indicates that the step has not been programmed.

### Searching for a macro-step

**[M]** Permits entry of the number of the macro-step.

**<ENTER>** Validates the entry and displays the page containing the macro-step. If the macro-step searched for does not exist in the module, a message indicates that the macro-step has not been programmed.

### 7.2-3 Storage and Retrieval of a Grafcet Module

#### [WRITE]

Permits the storage of the current Grafcet module in a "source" file on disk. This xxx.GR7 file, which is stored in the PL7\_3\MOD directory, allows the module to be reused in another application and its integration in a module base. The following soft keys are displayed :

**[STR.NAME]** permits entry of the name of the file.

**[CODE]** permits the storage of the Grafcet module, with or without the code of the associated actions and transitions.

**[OBC]** permits the storage of 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, a file xxx.OBM (with the same name as the file xxx.GR7) provides a list of all OFBs which have generated an .OBC file.

**[FROM]** permits entry of the number of the first page to be stored.

**[TO]** permits entry of the number of the last page to be stored.

**[TOP BOT]** permits storage of the complete module.

#### [READ]

Permits the retrieval of a Grafcet "source" file into the RAM memory of the terminal. Retrieval is effected by the substitution of pages in the current module. The following soft keys are displayed :

**[STR.NAME]** permits entry of the name of the file.

**[MODE]** defines the mode of retrieval :

AUTO : total retrieval, with stopping only on an error,

MANU : manual retrieval with confirmation of each element read.

#### Notes

For more detailed information on the modular V5, see Section 18 - Appendices.

A Grafcet "source" file can be retrieved when the terminal is in Local or Connected mode with the PLC stopped, but not in Connected mode with the PLC running.

The "source" file of a CHART module cannot be retrieved in an XM module and vice versa.

---

## 7.3 Zoom Function

---

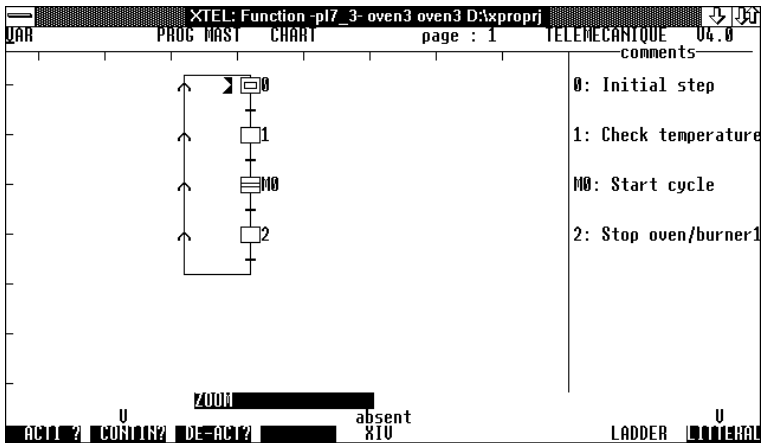
The ZOOM function permits the reading, writing or modification of :

- The expansion of a macro-step
- The condition(s) associated with a transition
- The action(s) associated with a step.

### Using the Grafcet Zoom function

The procedure is the same for a macro-step, a transition, or a step. The terminal is in the DISPLAY mode and a page of program is displayed :

**<ZOOM>** Switches the terminal to the Grafcet ZOOM mode. In this mode, a transition that is not programmed is shown in reverse video. For the execution of the program, it is considered as being always true. A cursor appears on the screen to select the step or transition to be "zoomed".



**< >** Use these keys to position the cursor on the required graphic element :  
**<↑>< >** step, macro-step or transition.

**<ZOOM>** Switches the terminal back to the DISPLAY mode and displays the contents of the selected element, which can now be modified.

**<ZOOM>** Switches the terminal back to the ZOOM mode, so that any other element on the screen can be zoomed. This operation can be repeated as often as required, so as to descend to the lowest level of analysis.

**<CLEAR>** Returns to the previous level of analysis.

### 7.3-1 Zooming on a Macro-step

The terminal is in the ZOOM mode and the cursor is positioned on a macro-step :

**<ZOOM>** Displays the expansion of the macro-step.

**<ZOOM>** This operation can be repeated as often as required down through 64 hierarchical levels (64 consecutive macro-steps)

This procedure also permits access to an empty macro-step, so that its expansion can be programmed (see Section 7.4).

### 7.3-2 Zooming on a Transition

The terminal is in the ZOOM mode and the cursor is positioned on a transition. If the transition condition has not yet been programmed, the programming language must be defined (1) : Ladder or Literal.

#### The transition condition is already programmed

**<ZOOM>** Accesses the transition condition.  
In Literal language (2), the corresponding statement is displayed in the operator input line at the bottom of the screen.  
In Ladder language (3), the corresponding ladder network is displayed.

**[MODIF]** Permits modification of the condition.

**<ENTER>** Validates the modification.

**<DEL>** Deletes the condition.

#### The transition condition is not already programmed

In this case, the transition concerned is displayed in reverse video.

**[LADDER]** Selects Ladder language for programming the transition condition. (1)

**[LITERAL]** Selects Literal language for programming the transition condition. (1)

**<ZOOM>** Accesses the transition condition.

**<INS>** permits entry of the condition in Ladder language. (3)

**[MODIF]** permits entry of the condition in Literal language. (2)

**<ENTER>** validates the entry.

(1) PL7-3 only

(2) Except PL7-3 GLD

(3) Except PL7-3 GLT



---

### 7.3-3 Zooming on a Step

The terminal is in ZOOM mode with the cursor positioned on a step.

#### Selection of the type of action

**[ACTI]** Action on activation

**[CONTIN]** Continuous action

**[DE-ACT]** Action on deactivation.

These keys are shown with a question mark if the corresponding action has not been programmed. The type of action selected is indicated on the screen by a pointer over the corresponding soft key.

#### Entering an action

**[LITERAL]** Selects Literal language. (1)

**[LADDER]** Selects Ladder language. (1)

**<ZOOM>** Accesses the action. The terminal then displays an empty screen to allow entry of the action in the selected language.

#### Reading an action

After selecting the type of action (the step being already programmed) :

**<ZOOM>** Accesses the action.

**[MODIF]** permits modification of the action.

#### Assignment of the active time value of a step

If sufficient XiV words were declared during the configuration, the key below appears when the cursor is positioned on a step.

**[XIV]** This key is a toggle. When the word "absent" is displayed over the key, it permits the assignment of an active time value to the step. When the word "present" is displayed over the key, it cancels the assignment of an active time value to the step. The key is not displayed unless at least one XiV word was declared in the configuration.

If a step has not been assigned a specific active time value, the latter can be read but its current value remains frozen at -1.

---

(1) PL7-3 only

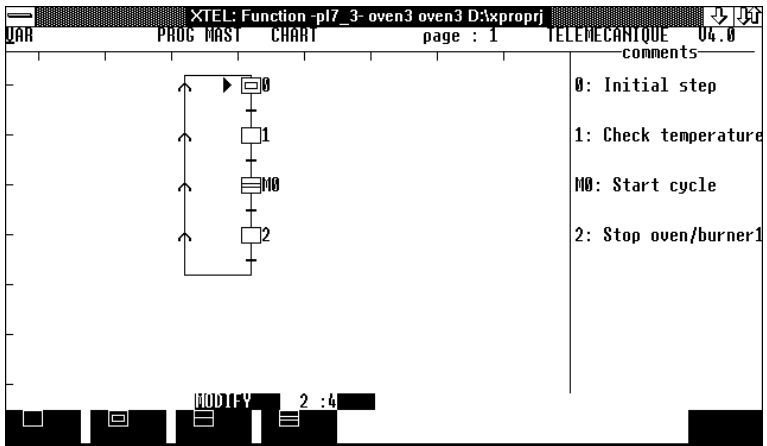
## 7.4 Entering or Modifying a Program

### 7.4-1 Procedure

The procedure for entering a new program or for modifying an existing program is the same :

- Define the address (task, module and page) of the program, the page to be written or modified appears on the screen in the DISPLAY mode.
- Select the MODIFICATION mode by pressing [MODIF]. The page displayed can then be written if it is blank, or modified if it has already been programmed.

The Grafcet can be modified with the terminal in Local or Connected mode. But modification with the PLC running is impossible, irrespective of the processor used.



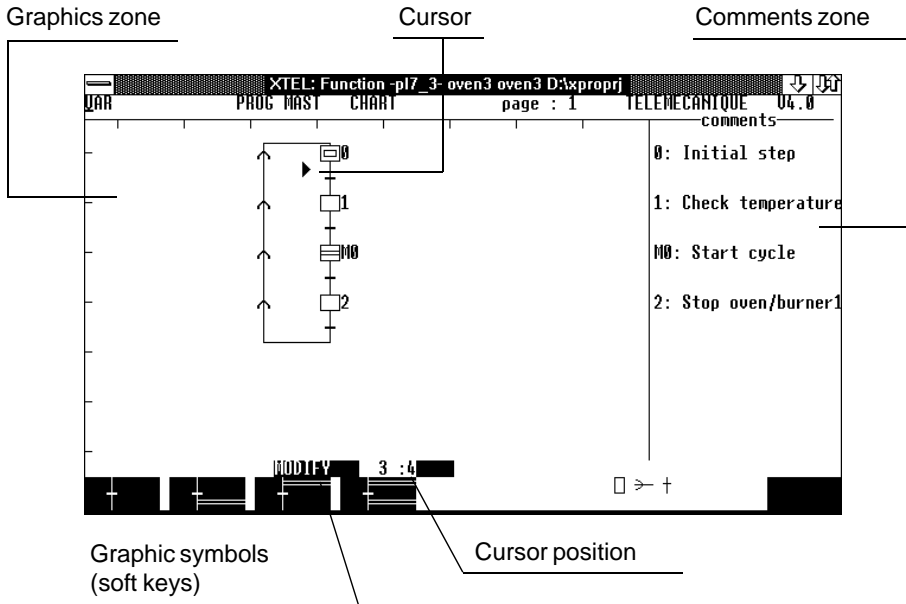
The user is continuously guided by the soft keys which ensure the correct alternation of steps and transitions, depending on the position of the cursor (step - transition - step). The alphanumeric keys are used only to enter step numbers, connector numbers or comments.

- <DEL>** Deletes the graphic symbol immediately to the right of the cursor.
- <ENTER>** Each page written or modified must be validated by pressing this key, so that it can be compiled and entered in memory. The terminal then automatically returns to DISPLAY MODE.

---

## Grafcet modification screen

When a new program is being entered, the screen below appears (empty at first) in MODIFY mode.



The cursor can be moved anywhere on the screen by using the cursor control keys or the mouse. Its position (line and column) is repeated at the bottom of the screen. For example 2:4 indicates that the cursor is on the second line of the fourth column in the graphics zone.

The Grafcet charts are entered by using the soft keys, which evolve according to the position of the cursor so as to ensure the alternation of steps and transitions.

After entering a step, the terminal automatically proposes the symbols necessary to enter a transition, and vice versa.

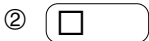
It is therefore possible to create Grafcet charts with :

- Linear sequences (see Section 7.4-2)
- Simultaneous sequences (see Section 7.4-3)
- Sequence selections, which can be prolonged as desired by using the two "pathway" keys :

Pathway from transition to step :

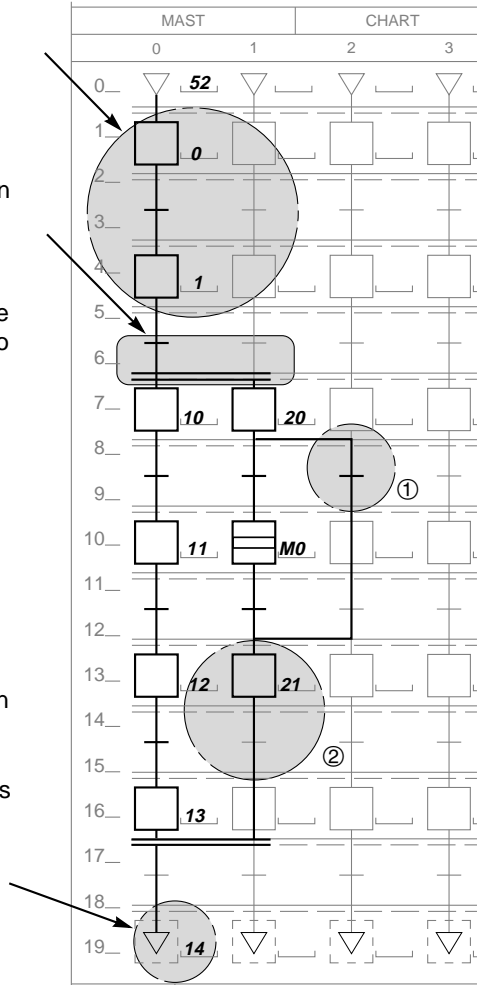


Pathway from step to transition :



The entry of pathways is described in Section 7.4-5.

- Connectors for destination or origin steps (see Section 7.4-4).



It is also possible to enter optional comments associated with the Grafcet chart (see Section 7.4-6).

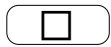
---

## 7.4-2 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 step symbol, the step number can then be entered by :



- pressing soft key F1 if the number proposed is correct
- using the numeric keys if the number proposed on F1 is not correct. The new number must then be validated by <ENTER>.

### Entering a transition



Enter the required transition symbol. In the case of a linear sequence, the transitions can be entered directly by pressing soft key F1.

### Rapid entry of a linear sequence

Rapid entry of a linear sequence can be obtained by holding down soft key F1. The steps are numbered automatically in ascending order. If the step number proposed at the start of the sequence is not correct, enter the desired number and press <ENTER>.

### Entering a macro-step

The procedure is the same as for entering a step, but no number is proposed. The number of the macro-step must always be entered by the user and validated by <ENTER>.



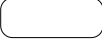

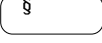
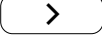
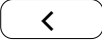
Enters the symbol for a mono-active macro-step.



Enters the symbol for a multi-active macro-step.

### 7.4-3 Entering Simultaneous Sequences


The terminal does not impose any particular order for entering simultaneous sequences, except of course the alternation of steps and transitions.

-  Enters the transition from a step to several steps (transition to simultaneous activation of steps).
-  Enters the transition from several steps to one step (transition to simultaneous deactivation of steps).
-  Enters the transition from several steps to several steps.
-  This key does not appear unless the cursor is at the level of a transition to simultaneous activation or deactivation. It permits the prolongation of the double line indicating simultaneous activation or deactivation.
-  Deletes a prolongation that is too long.

### 7.4-4 Entering Connectors

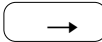
#### Source connector

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

-  Enters the source connector. The number of the step or macro-step can then be entered by :
  - Pressing soft key F1 if the number proposed is correct (number of the last step entered)
  - Using the numeric keys if the number proposed is not correct. The new number must be validated by <ENTER>

**[M]** permits entry of the number of a macro-step, which must be validated by <ENTER>.

#### Destination connector

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

## 7.4-5 Drawing Pathways

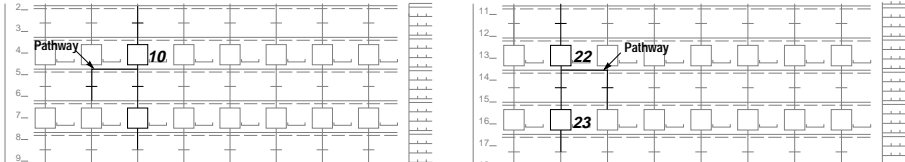
The use of pathways enables the directed links connecting steps to transition, or transitions to steps, to be prolonged to the left or right, or upwards and downwards.

The terminal therefore proposes two types of pathways :

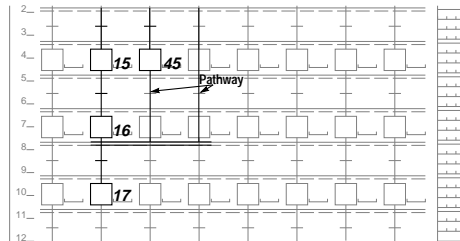
### Step - transition pathway

This type of pathway permits :

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



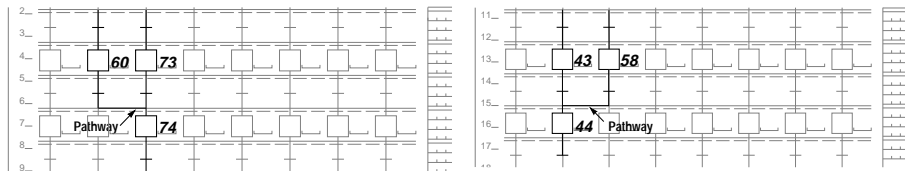
- The prolongation of a link from a step to the transition that deactivates simultaneous sequences.



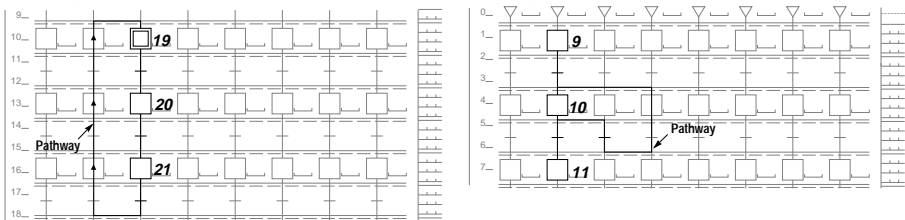
### Transition - step pathway

This pathway permits :

- The drawing of the end of a sequence selection coming from the left or right,



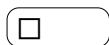
- The drawing of a loop from the end of a sequence back to the initial step, or of a loop that permits the reactivation of one or more steps in a sequence.



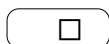
---

**Keys used**

The following keys permit the creation of pathways from a step to a transition or from a transition to a step.



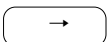
Step - transition pathway : this key is proposed by the terminal on the transition lines.



Transition - step pathway : this key is proposed by the terminal on the step lines.



Directed link upwards.



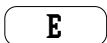
Directed link downwards.



Directed link to the right.



Directed link to the left.



Deletes one by one the elements of a pathway being entered, and permits exit from the pathway mode.



Permits the entry of a junction between two pathways of the same type (step transition or transition step).

**<CLEAR>**

Quits the pathway mode and returns to the MODIFY mode. If a pathway is being entered, the whole of the pathway is deleted.

**Note**

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

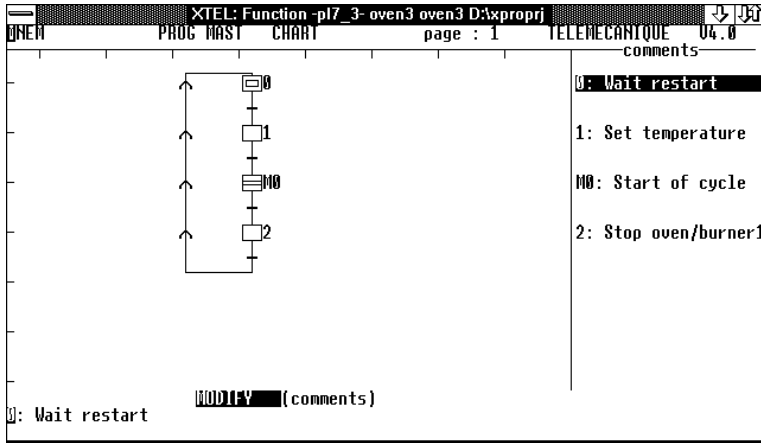


---

## 7.4-6 Entering Comments

Comments of 60 characters maximum can be associated with each step and each transition. Only the first 20 characters of the comment are displayed in the Comments zone, the remainder can be read in the input line at the bottom of the screen.

When the cursor is moved into the Comments zone, the line selected is displayed in reverse video :



**[MODIF]** Permits entry of the comment.

**<ENTER>** Validates the entry. The first 20 characters are displayed in the Comments zone and the cursor moves to the next line.

**<↑>< >** Positions the cursor on the desired comment.

**< >** Repositions the cursor in the graphic zone.

---

### 7.4-7 Validating a Page

A Grafcet page that has just been written or modified must be validated before entering a new page or quitting the MODIFY mode.

The cursor **must** be in the graphics zone at the moment of validation.

**<ENTER>** Must be pressed (in the MODIFY mode) to validate and compile a complete page of Grafcet.

If the Grafcet page is not coherent, the terminal displays an error message which indicates the corrections that must be made to the Grafcet.

---

### 7.4-8 Modifying the Elements in a Page

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

The modification or deletion is effected element by element. The modifications become effective after the validation of the page.

#### Modifying a step

The procedure for modifying a step is the same as for entering it :

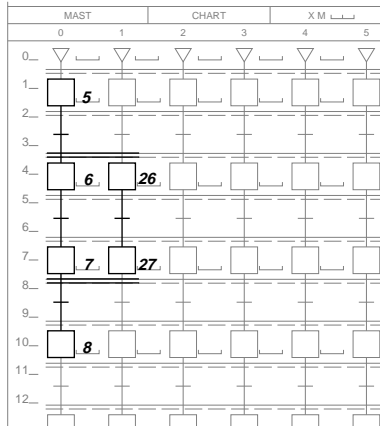
- Position the cursor on the step to be modified.
- Press the soft key corresponding to the desired graphic symbol. The existing graphic symbol is deleted.
- Enter the number of the step and validate it by <ENTER>.

This procedure permits the modification of either the graphic symbol, or the step number, or both. The modification becomes effective when the complete page is validated by pressing <ENTER>.

#### The validation of the modification of a step number (by validation of the page), causes the deletion of :

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

## Case of simultaneous activations and deactivations :



MODIFICATION OF THE NUMBER OF THE STEP	ELEMENTS DELETED				
	ACTIONS ASSOCIATED WITH THE STEP	CONDITIONS ASSOCIATED WITH THE TRANSITIONS			
		UPSTREAM	DOWNSTREAM	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

### Modification of 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 changing the macro-step number :

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

- **Modification of the number of a macro-step :**

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

- **Replacement of a macro-step by a step :**

The procedure is the same as for the modification of a macro-step number. If the macro-step that has been replaced by a step is not going to be used elsewhere, its expansion should be deleted.

### Modification of a transition

The procedure is as follows :

- Position the cursor on the transition to be modified.
- Press the soft key corresponding to the desired symbol. The existing symbol is deleted.

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

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

### Modifying a connector

If the number of a source or destination connector is modified, this means that the corresponding step has been modified, and also the memory address of the transition condition. The transition condition has therefore been deleted and must be re-entered.

### Returning to the original page

After a page has been modified, it must be validated by pressing <ENTER>. The new page is then entered in memory.

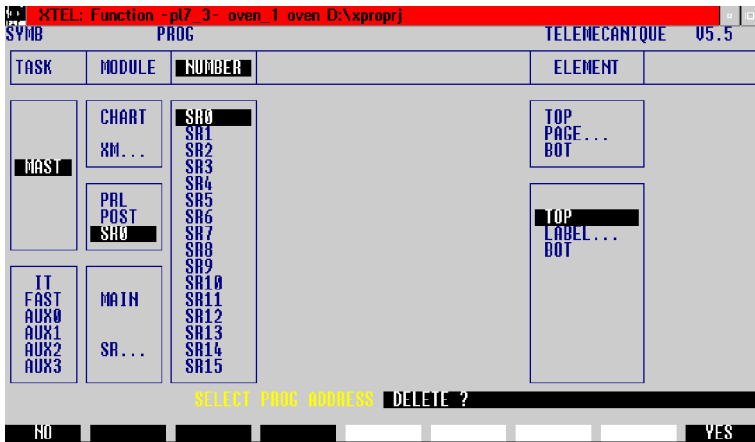
Alternatively, if <CLEAR> is pressed, the original page is displayed and the modifications are cancelled.

## 7.4-9 Deleting a Module, a Page, or an Element

### Deleting a module

The module to be deleted must be selected on the Program Address display :

<DEL> Permits deletion of the selected module (after confirmation).



The message "DELETING" blinks during the deletion of the module, and then disappears as soon as the module is deleted.

If the module selected was a macro-step, this procedure deletes the expansion of the macro-step.

### Attention

A program module (task/module) containing a write-protected elementary module cannot be deleted.

---

## Deleting a page

The contents of a Grafcet page can be deleted either in the DISPLAY mode, or while the page is being written or modified.

Two cases can occur :

- If the Grafcet chart(s) are programmed on one page only  
**<DEL>** Deletes the selected page (after confirmation).
- If the Grafcet chart(s) are programmed on two or more pages, the terminal checks the coherence of the source and destination connectors before deleting the page. If an incoherence is detected, it is impossible to quit the module without making the necessary corrections. Once the corrections are made, the page can be deleted.

### Remark

Deleting the contents of a Grafcet page does not delete the page number from the configuration. The deleted page can therefore be reprogrammed after it has been deleted.

## Deleting an element

When the terminal is in the MODIFY mode, any of the following elements can be deleted :

- A step
- A macro-step
- A transition
- A pathway.

After positioning the cursor on the element to be deleted

**<DEL>** Deletes the element.

The deletion of a step causes the deletion of :

- The transition or pathway downstream of the step
- The actions associated with the step

The deletion of a macro-step causes the deletion of the downstream transitions, but the deletion of a macro-step symbol does not delete its expansion, which can be used elsewhere. The deletion of a macro-step and its expansion are described above (see Deletion of a module).

During the deletion of a transition, a pathway or a connector, the cursor must be positioned on a transition line. In all three cases, the associated condition is deleted.

In the case of a complex pathway, the time required for the terminal to delete the pathway may be relatively long, since it has to check the coherence of the pathways which are not deleted and which intersect with the deleted pathway.

---

## 7.5 Grafcet Window Mode

---

### 7.5-1 Presentation

The Grafcet Window mode is accessible from the MODIFY mode. It permits :

- The definition of a graphic window whose size and position on the screen are determined by the submodes DEFINE WINDOW and POSITION WINDOW.
- The repositioning of the graphic elements in the window by the submodule MOVE WINDOW.

In the DEFINE WINDOW and POSITION WINDOW modes, it is also possible to delete the graphic elements contained in the window.

---

### 7.5-2 Defining and Positioning the Window

#### Defining the window

This submodule is accessible from the MODIFY mode.

- <INS>** Switches the terminal to the DEFINE WINDOW mode. A basic window shown in reverse video appears to the right of the cursor.
- < >< >** These keys allow the window to be enlarged horizontally and vertically from the position of the cursor.
- <↑>< >** The window can be enlarged to cover the whole of the graphics zone (excluding the Comments zone).

#### Positioning the window

After defining the size of the window in the DEFINE WINDOW mode :

- [INS]** Switches the terminal to the POSITION WINDOW mode. The window can then be moved anywhere in the graphics zone. Pressing this key a second time switches the terminal from the POSITION WINDOW mode to the DEFINE WINDOW mode, and vice versa.
- < >< >** Move the window horizontally.
- <↑>< >** Move the window vertically.
- <CLEAR>** Switches the terminal to the MODIFY mode and validates the operations performed in the Grafcet Window mode.

### 7.5-3 Moving Graphic Elements

The moving of graphic elements is accessible in the DEFINE WINDOW and POSITION WINDOW modes.

**[MOVE]** Switches the terminal to the MOVE WINDOW mode. The graphic elements covered by the window can then be moved.

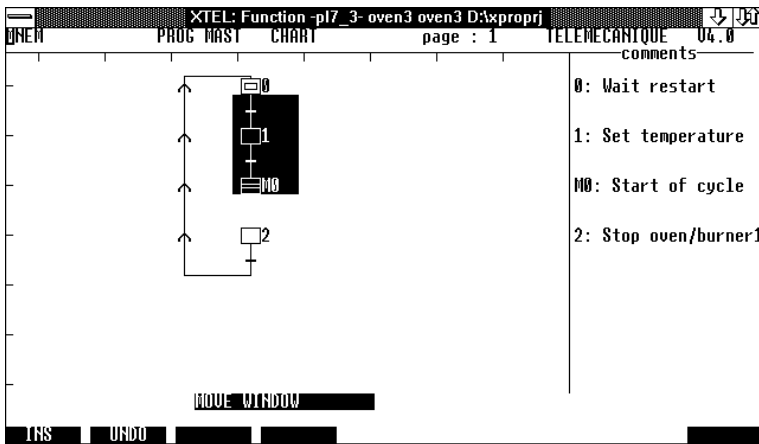
**< >>** Moves the graphic elements in the window horizontally.

**<↑>< >** Moves the graphic elements in the window vertically.

**[INS]** validates the movement of the graphic elements and switches the terminal back to its previous mode (DEFINE WINDOW or POSITION WINDOW).

**[UNDO]** cancels the movement of the graphic elements in the window.

**<CLEAR>** Switches the terminal back to the MODIFY mode and validates all the operations performed in the Grafcet Window mode.



#### Notes

- The graphic elements cannot be moved so that they overlay other graphic elements.
- The graphic elements are moved with their associated actions and transition conditions.

---

#### 7.5-4 Deleting Graphic Elements

After defining a window containing the graphic elements to be deleted :

**[DEL]** Deletes (after confirmation) the graphic elements contained in the window. This key is accessible in the DEFINE WINDOW or POSITION WINDOW modes.



---

C



Sub-section	Page
<b>8.1 Selecting the Task and Module</b>	8/2
8.1-1 Selecting the Program Address	8/2
<b>8.2 Reading the Program</b>	8/4
8.2-1 Reading the Ladder Networks	8/4
8.2-2 Storage and Retrieval of a Ladder Language Module	8/5
<b>8.3 Using the Zoom Function in Ladder Language</b>	8/6
8.3-1 Zooming on a Horizontal Comparison Block	8/6
8.3-2 Zooming on a Operation Block	8/7
8.3-3 Zooming on a SR CALL Output Coil	8/8
<b>8.4 Creating or Modifying the Program</b>	8/9
8.4-1 Using the Insert and Modify modes	8/9
8.4-2 Entering or Modifying a Label	8/10
8.4-3 Entering or Modifying a Comment	8/11
8.4-4 Creating a Ladder Network	8/11
8.4-5 Entering or Modifying a Function Block	8/13
8.4-6 Entering or Modifying a Comparison Block or Operation Block	8/13
8.4-7 Diagnostic Associated with a Ladder Network	8/15
8.4-8 Modifying an Element in a Ladder Network	8/16
8.4-9 Validating a Ladder Network	8/17
8.4-10 Deleting a Module or a Ladder Network	8/17
8.4-11 Duplicating a Ladder Network	8/17
<b>8.5 Using the Window Function</b>	8/18
8.5-1 Presentation	8/18
8.5-2 Defining and Positioning the Window	8/18
8.5-3 Moving Graphic Elements	8/19
8.5-4 Duplicating Graphic Elements	8/20
8.5-5 Deleting Graphic Elements	8/20
<b>8.6 Searching for the Program Addresses of a Variable</b>	8/20

This section ends on page 8/20

D



### Selecting the module

- [MODULE]** Permits selection of the module :
- MAIN : selects the Main program
  - SR... : selects a subroutine
- [NUMBER]** this key appears only if a subroutine module has been selected. It permits entry of the subroutine number.
- < → > in the MODULE column, if the cursor is positioned on an SR...or XM...line, then this key displays the list of subroutines configured in the application, in the NUMBER column.
- [MOD SCR]** Accesses the Module Selection screen comprising the list of elementary modules and macro-modules for the application (see Section 7.1-2).
- [NAME]** Accesses the LIST OF MODULES so that a module (task and module) can be selected by its name (see Section 7.1-3).

### Selecting the Ladder network

- [TOP]** Selects the first Ladder network of the selected module.
- [BOT]** Selects the last Ladder network of the selected module.
- [LABEL]** Permits selection of a Ladder network identified by its label. An unlabelled Ladder network can also be displayed (providing it exists) by entering the number of the nearest label, plus or minus a value.
- Example :** An unlabelled Ladder network located 7 networks downstream of the network labelled L38, and 2 networks upstream of the network labelled L61, can be selected by entering L38+7 or L61-2.

If the Ladder network concerned does not exist in the selected module, the terminal proposes the entry of a new label number.

**<ENTER>** Validates the selections.

Two cases can then occur :

#### 1 - The module is empty :

- [LADDER]** Selects the Ladder language. (1)
- <INS>** switches the terminal to INSERT mode, which permits entry of the Ladder program (see Section 8.4).

#### 2 - The module has already been programmed :

The program can then be read Ladder network by Ladder network (see Section 8.2).

(1) Except PL7-3 GLD

---

## 8.2 Reading the Program

---

### 8.2-1 Reading the Ladder Networks

#### Ladder network display screen

After the Program Address has been selected (task and module), the terminal is in the DISPLAY (Read) mode and the program can be read Ladder network by Ladder network. A Grafset step action or transition condition that has been programmed in Ladder language can be accessed by using the <ZOOM> key (see Section 7.3).

If a diagnostic OFB is associated with the Ladder network displayed, the letter D appears in the bottom left of the Ladder network. During operation, the Ladder network diagnostic is executed after the Ladder network with which it is associated.

#### Direct access to a Ladder network

**[TOP]** Displays the first Ladder network of the module. If this Ladder network is already displayed, the message TOP MODULE appears.

**[BOT]** Displays the last Ladder network of the module. If this Ladder network is already displayed, the message BOT MODULE appears.

**[LABEL]** Permits the display of :

- A Ladder network identified by a label (example : L24)
- An unlabelled Ladder network identified in relation to a label (example : L12+5)

<ENTER> validates the entry of the label.

<CLEAR> cancels the entry.

#### Reading a program Ladder network by Ladder network

<↑><↓> Permits the display of a module Ladder network by Ladder network. The message TOP MODULE or BOT MODULE appears when there are no further Ladder networks upstream or downstream of the Ladder network displayed.

---

### 8.2-2 Storage and Retrieval of a Ladder Language Module

**[WRITE]** Enables the storage of the current Ladder module in a "source" file on disk. This xxx.LAD file, which is stored in the PL7\_3\MOD directory, allows the module to be reused in another application and its integration in a module base. The following soft keys are displayed :

**[STR.NAME]** permits the name of the file to be defined.

**[OBC]** permits the storage of the Ladder 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, a file xxx.OBM (with the same name as the file xxx.LAD) provides a list of all OFBs which have generated a .OBC file .

**[FROM]** permits entry of the label of the first Ladder network to be stored.

**[TO]** permits entry of the label of the last Ladder network to be stored.

**[TOP → BOT]** permits storage of the entire module.

**[READ]** Enables retrieval of a "source" file in the RAM memory of the terminal. The following soft keys are displayed :

**[STR.NAME]** permits the name of the file to be defined.

**[MODE]** defines the mode of retrieval :

AUTO : global retrieval, with stopping on errors only.

MANU : manual retrieval with confirmation after each element read.

#### Note

For further information on the modular V5, see Section 18 - Appendices.

---

## 8.3 Using the Zoom Function in Ladder language

---

The Zoom function is a submode of the DISPLAY mode and permits access to the contents of :

- A horizontal comparison block
- An operation block
- A subroutine called by an SR CALL output coil.

### Procedure for using the zoom

**<ZOOM>** Switches the terminal from the DISPLAY screen, activates ZOOM mode; the cursor then appears on the left power rail.

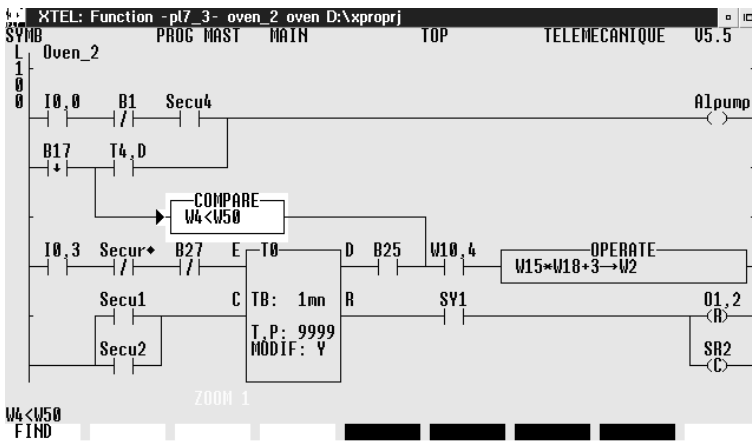
**[FIND]** Positions the cursor in succession on all the zoomable elements. In the case of an operation or comparison block, the element indicated appears in reverse video.

**<CLEAR>** Switches the terminal from ZOOM mode to DISPLAY mode.

---

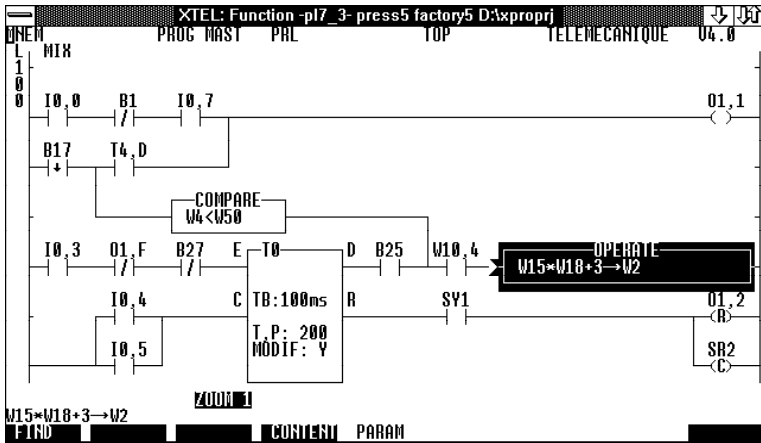
### 8.3-1 Zooming on a Horizontal Comparison Block

**<ZOOM>** Displays the contents of the comparison block.



### 8.3-2 Zooming on an Operation Block

<ZOOM> Displays the contents of the operation block.



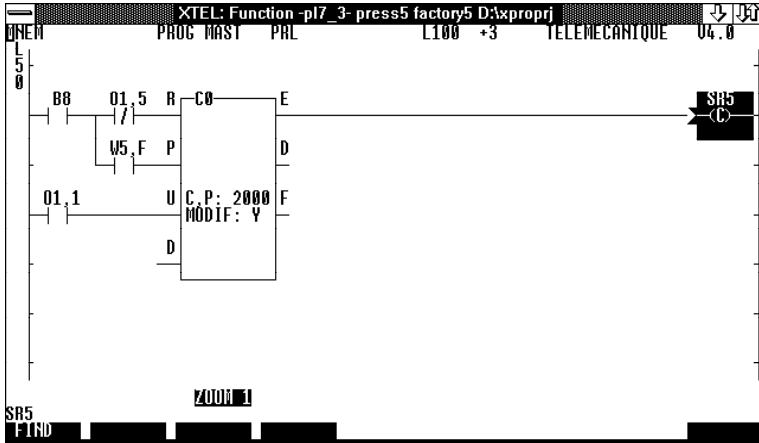
**[CONTENT]** In the case of an EXEC operation block, this key displays the internal constants of the OFB.

**[PARAM]** In the case of an EXEC operation block, this key displays the I/O parameters of the OFB.



### 8.3-3 Zooming on a SR CALL Output Coil

<ZOOM> 1st level : display of the output coil SR5 in reverse video.



<ZOOM> 2nd level : access to the subroutine.

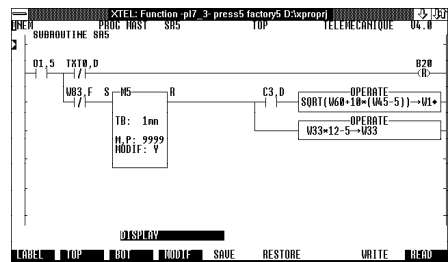
SR5 is not programmed :



The terminal proposes selection of the language (1). Subroutine SR5 can then be programmed.

In Ladder language the message MODULE EMPTY : INS TO START appears on the screen.

SR5 is programmed :



The terminal is in the DISPLAY mode and displays the first Ladder network of subroutine SR5.

(1) Except PL7-3 GLD

---

## 8.4 Creating or Modifying the Program

---

### 8.4-1 Using the Insert and Modify modes

To create or modify a Ladder network, as described in this section, the terminal must be switched from the DISPLAY mode to the INSERT or MODIFY modes.

#### Modifying a program from the DISPLAY mode

- If the module is to be modified :

**[MODIF]** Gives access to the MODIFY mode so that the Ladder network displayed can be modified.

**<INS>** Gives access to the INSERT mode so that a new Ladder network can be inserted upstream of the one displayed.

**<DEL>** Deletes the Ladder network displayed.

- If the module is to be deleted :

**<CLEAR>** Returns to the Program Address selection screen.

**<DEL>** Deletes the selected module (in reverse video), after confirmation.

#### Creating a program

The selected module is empty. The message MODULE EMPTY : INS TO START appears on the screen.

**<INS>** Switches the terminal to the INSERT mode so that the program can be written. The two power rails are then displayed.

The program is entered Ladder network by Ladder network. The operator is continuously guided by the soft key menus which vary according to the position of the cursor and propose only the symbols that can be legally entered in the position concerned, thus preventing syntax errors.

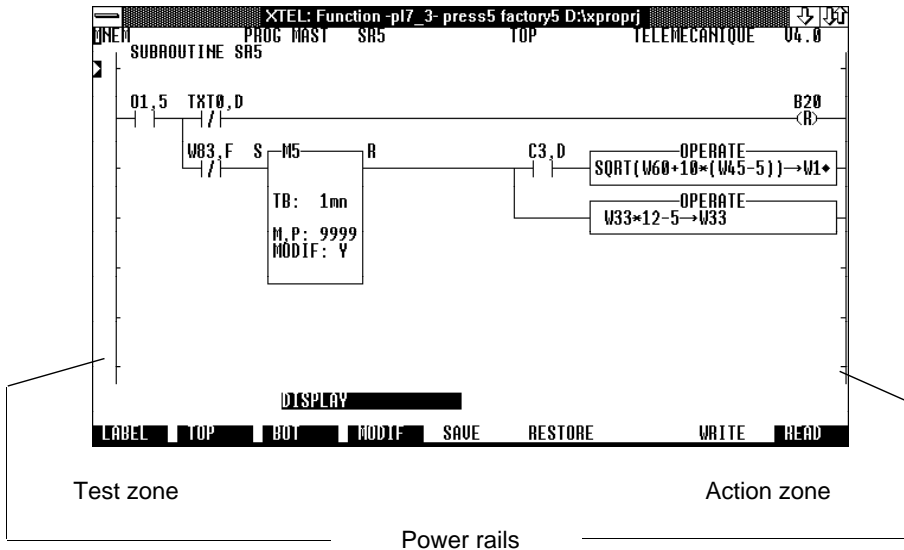
---

## Using the soft keys

For creating and modifying programs in the INSERT or MODIFY modes, five soft key menus permit the entry of all the graphic symbols required.

These soft key menus depend on the position of the cursor :

- On the left power rail
- In the Test zone
- In the Action zone



---

### 8.4-2 Entering or Modifying a Label

Entering a label is optional. The cursor must be positioned on the left power rail for the required soft key to be displayed.

**[LABEL]** Permits the entry or modification of a label, which must then be validated by <ENTER>.

#### Deleting a label

Position the cursor on the left power rail level with the first line of contacts :

**<RUB>** Deletes the label (after confirmation).

### 8.4-3 Entering or Modifying a Comment

The entry of a comment on a Ladder network is optional. The cursor must be positioned on the left power rail for the required soft key to be displayed.

The comment is limited to a maximum of 222 characters and can be written on a maximum of three lines at the top of the screen.

**[COMMENT]** Permits the entry or modification of a comment, which must then be validated by <ENTER>.

<↵> Moves the cursor to the beginning of the next line (carriage return).

#### Remarks

- If the Ladder network occupies all 7 lines of the screen, it is impossible to enter a comment without modifying the Ladder network (by reducing it).
- If the Ladder network occupies less than 7 lines, the entry of a comment shifts the Ladder network downwards as necessary, to make room for the comment.

#### Deleting a comment

Position the cursor on the left power rail at the level of the first line of contacts :

<DEL> Deletes the comment.

### 8.4-4 Creating a Ladder network

#### Entering a contact

The entry of a contact comprises two steps :

- Entry of the graphic symbol by pressing the corresponding soft key
- Entry of the associated bit or mnemonic, and validation by <ENTER>.

[H ] Entry of a normally-open (NO) contact.

[H / ] Entry of a normally-closed (NC) contact.

[H ↑ ] Entry of a rising edge (RE) contact.

[H ↓ ] Entry of a falling edge (FE) contact.

#### Note

If the associated bit or mnemonic has 6 characters or less, it is displayed in full above the graphic symbol. However, if it has more than 6 characters, it may be displayed in full or only partially, depending on the amount of space available to its right.

---

## Entering coils

Coils can be entered only in the action zone on the right of the screen.

[←] Positions the cursor in the action zone.

Like the contacts, they are entered in two steps :

- Entry of the graphic symbol by pressing the corresponding soft key
- Entry of the associated bit or mnemonic and then validation by <ENTER>. In the case of a jump output coil, enter the label of the destination Ladder network (eg L115). In the case of a subroutine call output coil, enter the number of the subroutine (eg SR5).

[{ } ] Entry of a Direct output coil.

[{ / } ] Entry of an Inverse output coil.

[{ S } ] Entry of a Set output coil.

[{ R } ] Entry of a Reset output coil.

[{ J } ] Entry of a Jump output coil.

[{ C } ] Entry of a Subroutine Call output coil.

[{ # } ] Entry of a Transition Condition output coil.

### Note

When the bit or mnemonic associated with an output coil has more than 6 characters, it is only partially displayed.

## Entering connecting links

[—] Entry of a horizontal link.

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

[↓] Entry of a vertical link downwards from the position of the cursor.

[/↑] Deletion of a vertical link. The cursor must be positioned at the top of the vertical link.

### 8.4-5 Entering or Modifying a Function Block

The standard function blocks cannot be entered unless they have been configured. They are entered in two steps :

- Entry of the graphic symbol
- Entry of the number or mnemonic of the block and validation by <ENTER>. The parameters of the block are then displayed (ie the default parameters if they have not been modified).

**[TIMER]** Enters a timer.

**[REGISTR]** Enters a LIFO or FIFO register.

**[MONOST]** Enters a monostable.

**[COUNTER]** Enters an up/down counter.

**[TXT]** Enters a text block.

**[CTRL]** Enters a control block.

#### Modifying a function block

**<ZOOM>** When the cursor is positioned in front of a function block, this key permits modification of the block number.

**<ZOOM>** A second action on this key gives access to the Configuration mode so that the parameters of the block can be modified.

**<ENTER>** Validates the modification.

#### Note

In Connected mode, the modification of the parameters of the block does not become effective until system bit SY0 is set to 1.

### 8.4-6 Entering or Modifying a Comparison Block or an Operation Block

#### Vertical comparison block

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

**[CMP-VER]** Enters the graphic element. The terminal then permits entry of the two operands separated by a space. After validation by <ENTER>, they are displayed in the two 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 two operands, the terminal interprets the entry as a single object and displays an error message.

---

### Horizontal comparison block

This block is entered in the test zone and uses the Literal language.

**[CMP-HOR]** Enters the graphic element. The terminal then permits entry of the arithmetic expression (80 characters maximum). After validation by <ENTER>, the expression is displayed in the block either completely or partially, depending on whether its number of characters is greater than 11.

### Operation block

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

**[OPERATE]** Enters the graphic element. The terminal then permits entry of the arithmetic expression (512 characters maximum). After validation by <ENTER>, the expression is displayed in the block either completely or partially, depending on whether its number of characters is greater than 23. The entry must be made according to the rules of Literal language.

### Modifying an existing block

**<ZOOM>** When the cursor is positioned in front of the block, this key permits modification of the operands (vertical comparison block) or the arithmetic expression (horizontal comparison block or operation block).

**<ENTER>** Validates the modification.

---

### 8.4-7 Diagnostic Associated with a Ladder Network

For V4 and V5 applications only, it is possible to associate a diagnostic with a Ladder network, in order to monitor a Boolean condition or a transition. This is achieved by calling a diagnostic optional function block (OFB). Certain of these OFBs memorize the Ladder network that is monitored. When the terminal is connected to a PLC that is running, it is not possible to delete the last diagnostic, or add the first diagnostic, of a Ladder network that is memorized.

When a diagnostic is associated with a Ladder network, the letter D appears in the bottom left of the Ladder network.

The code of the diagnostic is executed after the Ladder network with which it is associated.

**[DIAG] <d><D>** Permits entry or modification of the code of the diagnostic associated with a Ladder network. The following soft keys are displayed :

**[DEL DIAG]** deletes the diagnostic associated with a Ladder network.

**[CONTENT]** permits entry or modification of the internal constants of the diagnostic OFB.

**[PARAM]** permits entry or modification of the I/O parameters of the diagnostic OFB.

**[IF] [THEN]** enters the keywords IF and THEN which condition the execution of the diagnostic OFB associated with a Ladder network.

**[EXEC]** enters the keyword EXEC which commands the execution of the diagnostic OFB.

**<ENTER>** Validates the entries or modifications.



---

## 8.4-8 Modifying an Element in a Ladder Network

### Deleting an element

The terminal must be in the INSERT or MODIFY modes. The procedure is the same for contacts, coils, function blocks, comparison blocks and operation blocks.


Position the cursor in front of the element to be deleted.

**<DEL>** Deletes the graphic element. A new element can then be entered.

### Modifying an element

Replacing a contact by a horizontal link :

Position the cursor to the left of the contact to be replaced.

[>] Replaces the contact shown in reverse video by a horizontal link (after confirmation).

Replacing a contact by another :

- Position the cursor to the left of the contact to be replaced
- Enter the new contact, which is then displayed on the screen
- If necessary, enter the associated bit or mnemonic
- Validate the new entry by <ENTER>.

Replacing an output coil by another :

The procedure is the same as for a contact.

Replacing a horizontal link by a contact :

- Position the cursor in front of the link to be replaced
- Enter the new contact, which is then displayed on the screen
- Enter the associated bit or mnemonic
- Validate the entry by <ENTER>.

Modifying a bit or mnemonic associated with an element :

Position the cursor to the left of the element to be modified.

**<ZOOM>** Permits modification of the bit or mnemonic associated with a contact or coil. Validate the entry by <ENTER>.

---

### 8.4-9 Validating a Ladder Network

In the INSERT and MODIFY modes, after a Ladder network is entered it must be validated so that it can be compiled and entered in memory.

**<ENTER>** Validates the Ladder network that has just been entered and switches the terminal to the DISPLAY mode. The Ladder network can then be read or modified.

---

### 8.4-10 Deleting a Module or a Ladder Network

#### Deleting a module

After selecting the module to be deleted on the Program Address screen :

**<DEL>** Deletes the selected module (after confirmation).

#### Deleting a Ladder network

The terminal must be in the DISPLAY mode.

**<DEL>** Deletes the Ladder network displayed on the screen (after confirmation).

---

### 8.4-11 Duplicating a Ladder Network

In the DISPLAY mode, this procedure enables a Ladder network to be saved in a buffer memory so that it can be duplicated as often as required in any module programmed in Ladder language.

#### Saving a Ladder network

**[SAVE]** Saves the Ladder network displayed on the screen (after confirmation). This Ladder network remains stored in the buffer memory until another SAVE request is received.

#### Restoring a Ladder network

The Ladder network concerned can be restored in the same program module or in any other Ladder program module.

**[RESTORE]** Inserts the saved Ladder network upstream of the one displayed on the screen. The terminal is then in the INSERT mode so that :

- The Ladder network can be modified (if necessary)
- And then validated by <ENTER> (whether modified or not).

---

## 8.5 Using the Window Function

---

### 8.5-1 Presentation

The window function can be accessed from either the INSERT or MODIFY modes. It allows the operator to :

- Define a graphic window whose size and position on the screen are determined in the DEFINE WINDOW and POSITION WINDOW modes.
- Move the graphic elements contained in the window in the MOVE WINDOW mode.

In the DEFINE WINDOW and POSITION WINDOW modes, it is also possible to :

- Save the graphic elements contained in the window so that they can be duplicated in the same Ladder network or another Ladder network.
- Delete the graphic elements contained in the window.

---

### 8.5-2 Defining and Positioning the Window

#### Defining the window

The terminal must be in the INSERT or MODIFY modes.

**<INS>** Switches the terminal to the DEFINE WINDOW mode. A basic window, shown in reverse video, appears to the right of the cursor.

**<←><→>** Permit the enlargement of the window, horizontally or vertically.

**<↑><↓>** The window can be enlarged to cover the whole of the screen, not including the label and comments (if any).

#### Positioning the window

After defining the size of the window in the DEFINE WINDOW mode :

**[INS]** Switches the terminal to the POSITION WINDOW mode, so that the window can be moved in order to cover the required graphic elements. A second action on this key switches the terminal from the POSITION WINDOW mode to the DEFINE WINDOW mode, and vice versa.

**<←><→>** Move the window horizontally.

**<↑><↓>** Move the window vertically.

**<CLEAR>** Switches the terminal back to the INSERT or MODIFY modes and validates the operations performed in the window mode.

### 8.5-3 Moving the Graphic Elements

Graphic elements can be moved in the DEFINE WINDOW and POSITION WINDOW modes.

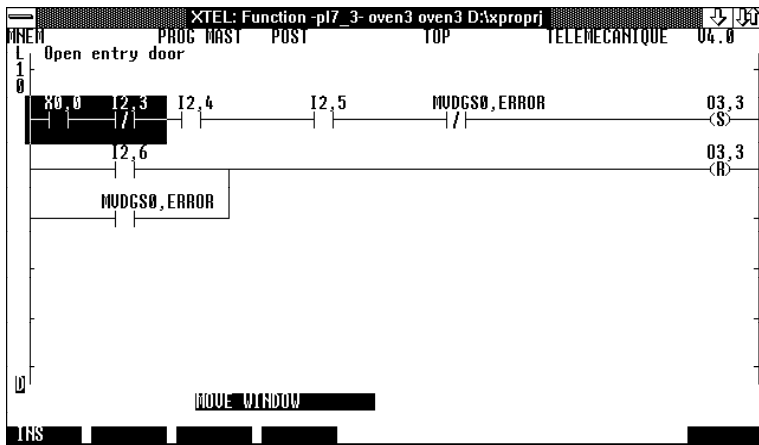
**[MOVE]** Switches the terminal to the MOVE WINDOW mode. The graphic elements contained in the window can then be moved.

**<←><→>** Move the graphic elements contained in the window left or right.

**<↑><↓>** Move the graphic elements contained in the window up or down.

**[INS]** Switches the terminal back to the previous mode (DEFINE WINDOW or POSITION WINDOW).

**<CLEAR>** Switches the terminal back to the MODIFY mode and validates all the operations performed in the window mode.



#### Notes

- The graphic elements cannot be moved to overlay other graphic elements (except for horizontal links).
- Only complete graphic elements can be moved. If the window covers only a part of a graphic element, the window cannot be moved and an error message appears.
- A window containing graphic action elements can only be moved vertically.
- A window containing only graphic test elements cannot be moved to the action zone.

---

## 8.5-4 Duplicating Graphic Elements

### Saving graphic elements

After defining a window containing the graphic elements to be duplicated :

**[SAVE]** This key is accessible in the DEFINE WINDOW or POSITION WINDOW modes. It saves the graphic elements contained in the window (after confirmation).

### Restoring graphic elements

Graphic elements that have been saved can be restored in the same program module or another program module.

The terminal must be switched to the DISPLAY mode by pressing <CLEAR>.

To restore the graphic elements in another program module, the address of the module must be selected on the Program Address screen.

**[RESTORE]** Restores the graphic elements in an "empty" Ladder network inserted upstream of the one that was displayed on the screen.

#### Notes

- A window that has been saved can be restored several times.
- A window that has been saved remains stored in the buffer memory until a new window is saved.
- The terminal always restores the window in an "empty" Ladder network.
- A window that contains Ladder language graphics cannot be restored in a Literal program module.

---

## 8.5-5 Deleting Graphic Elements

After defining a window containing the graphic elements to be deleted :

**[DEL]** This key, which is accessible in the DEFINE WINDOW or POSITION WINDOW modes, deletes the graphic elements contained in the window (after confirmation).

---

## 8.6 Searching for the Program Addresses of a Variable

When a Ladder network is shown on the screen in the Display mode, pressing <Alt><X> gives access to :

- The display of all the variables of the Ladder network
- The selection of a variable and the display of all the program addresses relating to the selected variable
- The selection of a program address and the display of the corresponding program element.

The variable concerned can be a standard PL7-3 variable (Wi, Bi ...) or an OFB variable. For further information, refer to Divider H, Section 18.5.



<b>Section</b>	<b>Page</b>
<b>9 Program Mode : Literal Language</b>	
Contents	9/1
9.1 Selecting the Task and Module	
9.2 Reading the Program	
9.3 Using the Zoom Function	
9.4 Creating or Modifying the Program	
9.5 Searching for the Program Addresses of a Variable	

Section 9 which describes programming in Literal, only applies to versions PL7-3 and PL7-3 GLT.

---

E



<b>Sub-section</b>	<b>Page</b>
<b>9.1 Selecting the Task and Module</b>	9/2
9.1-1 Selecting the Program Address	9/2
<b>9.2 Reading the Program</b>	9/4
9.2-1 Reading the Statements	9/4
9.2-2 Storage and Retrieval of a Literal Language Module	9/5
<b>9.3 Using the Zoom Function</b>	9/6
9.3-1 Zooming on Function Blocks	9/6
9.3-2 Zooming on a SR CALL Instruction	9/7
9.3-3 Zooming on an EXEC Instruction	9/8
<b>9.4 Creating or Modifying the Program</b>	9/9
9.4-1 Using the Insert and Modify modes	9/9
9.4-2 Entering or Modifying a Label	9/11
9.4-3 Entering or Modifying a Comment	9/11
9.4-4 Entering or Modifying a Literal phrase	9/12
9.4-5 Floating Point Type Instructions	9/15
9.4-6 Diagnostic Associated with a Statement	9/17
9.4-7 Validating a Statement	9/18
9.4-8 Deleting a Module or a Statement	9/18
9.4-9 Duplicating a Statement	9/19
<b>9.5 Searching for the Program Addresses of a Variable</b>	9/19

This section ends on page 9/20



---

## 9.1 Selecting the Task and Module

---

### 9.1-1 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 element (label) required.

The Program Address selection screen is the first screen displayed after selecting the Program mode (2 - PROGRAM) on the Mode Selection Display :

XTEL: Function - pl7\_3 - oven\_2 oven D:\xproprj

Symb PROG TELEMECANIQUE U5.5

TASK	MODULE	NUMBER	ELEMENT
MAST	MAIN	SR0	TOP
		SR1	LABEL...
		SR2	BOT
		SR3	
		SR4	
		SR5	
		SR6	
		SR7	
		SR8	
		SR9	
		SR10	
		SR11	
		SR12	
		SR13	
		SR14	
		SR15	

SELECT PROG ADDRESS

TASK MODULE MOD SCR NUMBER PREV SCR NEXT SCR TOP LABEL BOT

### Selecting the task

#### [TASK]

Permits selection of the task :

- MAST : Master task
- IT : Interrupt task (if it is configured)
- FAST : Fast task (if it is configured)
- AUX0 to AUX3 : Auxiliary tasks 0 to 3, if they are configured and if the type of processor selected permits the use of auxiliary tasks.

---

### Selecting the module

**[MODULE]** Permits selection of the module :

- MAIN : selects the Main program
- SR... : selects a subroutine

**[NUMBER]** this key appears only if a subroutine module has been selected. It permits entry of the subroutine number.

<→> in the MODULE column, if the cursor is positioned on an SR... or XM...line, then this key displays the list of subroutines configured in the application, in the NUMBER column.

**[MOD SCR]** Gives access to the Module Selection screen comprising the list of module elements and macro-modules for the application (see Section 7.1-2).

**[NAME]** Gives access to the LIST OF MODULES so that a module (task and module) can be selected by its name (see Section 7.1-3).

### Selecting the statement

**[TOP]** Selects the first statement of the selected module.

**[BOT]** Selects the last statement of the selected module.

**[LABEL]** Permits selection of a statement identified by its label. An unlabelled statement can also be displayed (providing a labelled statement exists) by entering the number of the nearest label, plus or minus a value.

**Example :** An unlabelled statement located 7 statements downstream of the statement labelled L38, and 2 statements upstream of the statement labelled L61, can be selected by entering L38+7 or L61-2.

If the statement concerned does not exist in the selected module, the terminal proposes the entry of a new label number.

**<ENTER>** Validates the selections.

Two cases can then occur :

#### 1 - The module is empty :

**[LITERAL]** Selects the Literal language (1)

**<INS>** switches the terminal to INSERT mode, which permits entry of the Literal program (see Section 9.4).

#### 2 - The module has already been programmed :

The program can then be read statement by statement (see Section 9.2).

---

(1) Except PL7-3 GLT

---

## 9.2 Reading the Program

---

### 9.2-1 Reading the Statements

After the Program Address has been selected (task and module), the terminal is in the DISPLAY (Read) mode and the program can be read statement by statement. A Grafset step action or transition condition that has been programmed in Literal language can be accessed by using the <ZOOM> key (see Section 7.3).

If a diagnostic OFB is associated with the statement displayed, the letter D appears in the bottom left of the statement. During operation, the diagnostic of the statement is executed after the statement with which it is associated.

#### Direct access to a statement

**[TOP]** Displays the first statement of the module. If this statement is already displayed, the message TOP MODULE appears.

**[BOT]** Displays the last statement of the module. If this statement is already displayed, the message BOT MODULE appears.

**[LABEL]** Permits the display of :

- A statement identified by a label (example : L24)
- An unlabelled statement identified in relation to a label (example : L12+5)

<ENTER> validates the entry of the label.

<CLEAR> cancels the entry.

#### Reading a program statement by statement

<↑><↓> Permits the display of a module statement by statement. The message TOP MODULE or BOT MODULE appears when there are no further statements upstream or downstream of the statement displayed.

---

## 9.2-2 Storage and Retrieval of a Literal Language Module

**[WRITE]** Permits the storage of the current Literal module in a "source" file on disk. This xxx.LIT file, which is stored in the PL7\_3\MOD directory, allows the module to be reused in another application and its integration in a module base. The following soft keys are displayed :

**[STR.NAME]** permits the name of the file to be defined.

**[OBC]** permits storage of the module in Literal language, 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 has internal constants. In addition, a file xxx.OBM (with the same name as the file xxx.LIT) provides a list of all OFBs which have generated an .OBC file.

**[FROM]** permits entry of the label of the first statement to be stored.

**[TO]** permits entry of the label of the last statement to be stored.

**[TOP → BOT]** permits storage of the complete module.

**[READ]** Permits retrieval of a "source" file in the RAM memory of the terminal. The following soft keys are displayed :

**[STR.NAME]** permits the name of the file to be defined.

**[MODE]** defines the mode of retrieval :

AUTO : global retrieval, with stopping on errors only

MANU : manual retrieval with confirmation of each element read.

### Note

For further information on the modular V5, see Section 18 - Appendices.

---

## 9.3 Using the Zoom Function

---

The Zoom function is a submode of the DISPLAY mode and permits access to :

- The parameters of a function block
- A subroutine called by an SR CALL output coil

Eight levels of zoom (from ZOOM 1 to ZOOM 8) are possible in the case of nested subroutines.

### Procedure

- <ZOOM>** Switches the terminal from the DISPLAY mode to the ZOOM mode. The pointer must be positioned beforehand on the statement containing the function block or subroutine to be accessed.
- <CLEAR>** Switches the terminal from the ZOOM mode to the DISPLAY mode.

---

### 9.3-1 Zooming on Function Blocks

Several levels of zoom can be used on function blocks.

**Example :** Monostable M1 and Counter C0.

With the terminal in the DISPLAY mode and the pointer positioned on the statement to be zoomed :

- <ZOOM>** Switches the terminal to the ZOOM 1 mode. The first zoomable function block in the statement is indicated by the cursor.
- <←→>** Can be used to position the cursor on the required function block (for example C0).

```
XTEL: Function - pl7_3 - oven_2 oven D:\xproj
SYMB  PROG MAST  SR0  TOP  +1  TELEMECANIQUE  05.5

<Fault detection
!      B10->01,C

<Power restart processing
!      IF B11+NOT B100
          THEN 0->B100[6];0->W100[10];CW10->M1,P;UP C0

ZOOM 1
```

**<ZOOM>** A second action on this key allows the operator to read or modify the configuration parameters of the function block.

NB	PRST	M	MNEMONIC
1	9999	Y	Counter
2	9999	Y	.....
3	9999	Y	.....
4	9999	Y	.....
5	9999	Y	.....
6	9999	Y	.....
7	9999	Y	.....

**COUNTERS**

NEXT TO LAST

**<CLEAR>** Returns to the previous screens.

### 9.3-2 Zooming on a SR CALL Instruction

The terminal must be in the DISPLAY mode with the pointer on the line containing the subroutine to be displayed, modified or entered.

**<ZOOM>** Switches the terminal to the ZOOM 1 mode. The first zoomable function block in the line is indicated by the cursor.

**<←→>** Positions the cursor on the CALL instruction (for example CALL SR2).

**<ZOOM>** Gives access to subroutine SR2. The terminal is in the DISPLAY mode. Two cases can occur :

- **SR2 is not programmed :**

**[LITERAL]** selects the Literal language.

(1) The subroutine SR2 can then be entered in the INSERT mode, according to the rules defined in Section 9.4.

- **SR2 is already programmed :**

The subroutine SR2 can be modified in the INSERT or MODIFY modes, according to the rules defined in Section 9.4.

(1) Except PL7-3 GLT

---

### 9.3-3 Zooming on an EXEC Instruction

With the terminal in the DISPLAY mode and the line pointer positioned on the statement to be zoomed :

**<ZOOM>** Switches the terminal to the ZOOM 1 mode and the first zoomable function block is indicated by the cursor.

**<←→>** Move the cursor onto the EXEC instruction to be zoomed.

**<ZOOM>** A new action on this key permits reading or modification of the I/O parameters and the internal constants of the OFB.

**[PARAM]** accesses the OFB I/O parameters.

**[CONTENT]** accesses the OFB internal constants.

## 9.4 Creating or Modifying the Program

### 9.4-1 Using the Insert and Modify modes

To create or modify a program, the terminal must be switched from the DISPLAY mode to the INSERT or MODIFY modes, as described below.

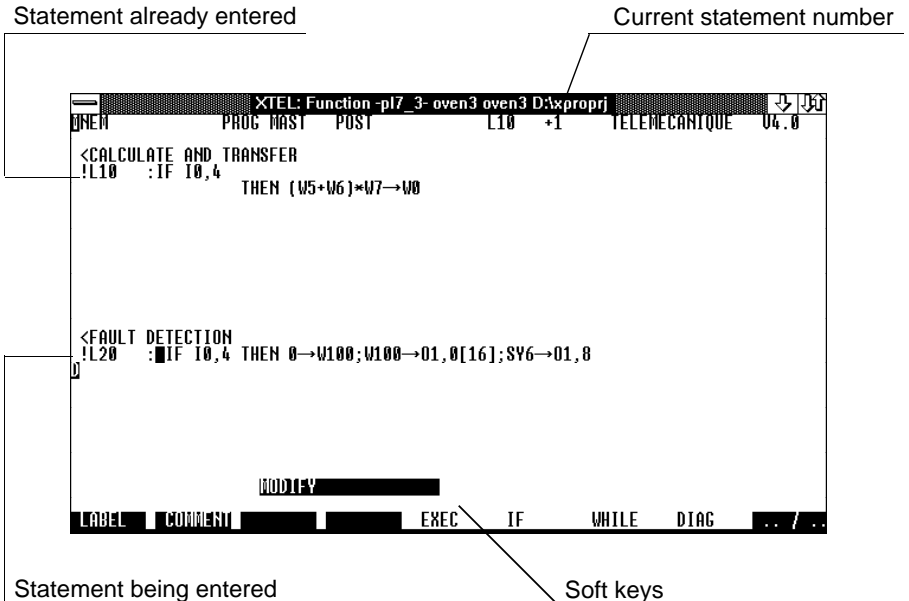
#### Modifying a program

- [MODIF]** Gives access to the MODIFY mode so that the statement indicated by the pointer can be modified.
- <INS>** Gives access to the INSERT mode so that a new statement can be inserted upstream of the one displayed.
- <DEL>** Deletes the statement indicated by the pointer (after confirmation).

#### Creating a program

The selected module is empty. The message MODULE EMPTY : INS TO START appears on the screen.

- <INS>** Switches the terminal to the INSERT mode so that the program can be entered statement by statement.





---

## Rules of entry

Writing a Literal statement comprises :

- Entry of the label number (optional)
- Entry of the comment (optional)
- Entry of the Literal phrase.

After entering or modifying a statement, it must be compiled and entered in memory by pressing <ENTER>. It then appears in the upper part of the screen. If the entry was made in the MODIFY mode, the terminal returns to the DISPLAY mode. If the entry was made in the INSERT mode, the terminal remains in this mode and proposes a new insertion.

## 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 an object, a separator or an instruction.

A space is not obligatory between the signs for an operation or between punctuation marks.

**Example :** IF (W5>7) THEN (W2+W6)\*3 → W8;W1 → W9

## Note

If a new character is entered over an old one, the old character is replaced by the new one and the cursor moves to the right.

## Entry errors

During entry, illegal keys are indicated by an audio warning. After validation of an insertion or 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 that is not configured
- Entry of an illegal instruction
- Entry of a label that already exists in the module.

## Means of entry

To create or modify a program, the following keys are available :

- Soft keys for entering the instructions
- Alphanumeric keys for entering objects and instructions
- Function keys for modifying and correcting statements.

---

### Using the soft keys

In the INSERT and MODIFY modes, the terminal proposes three basic soft key menus. These menus permit the entry of :

- An optional label or comment associated with the statement
- The conditional and iterative instructions (IF, THEN, ELSE, WHILE, DO)
- The diagnostic associated with the statement.

[./..] Switches the terminal from one soft key menu to the next one.

[M SPACES] Shifts the cursor 8 spaces to the right, so as to facilitate the layout of the statement on the screen.

[↑] On secondary soft key menus, this key gives access to the previous menu.

---

#### 9.4-2 Entering or Modifying a Label

The assignment of a numbered label to a statement is optional. However it is useful to label certain statements (eg every tenth one) so that they can be used as reference points. Each label can be numbered from 0 to 999 and each label number can be used only once in the same module.

[LABEL] Permits the entry or modification of a label, which must be validated by <ENTER>. The terminal stays in the INSERT or MODIFY modes so that the statement associated with the label can be written or modified. Entry errors are indicated by an error message.

---

#### 9.4-3 Entering or Modifying a Comment

The entry of a comment associated with a statement is optional.

After validation, the comment (composed of 78 characters maximum) appears above the Literal statement.

[COMMENT] Permits the entry of a comment or the modification of an existing comment, which must be validated by <ENTER>. The terminal stays in the INSERT or MODIFY modes so that the Literal phrase associated with the comment can be written or modified.

---

#### 9.4-4 Entering or Modifying a Literal phrase

The instructions of a Literal phrase can be entered by using the soft keys (which is the easiest way) or by typing the instructions in full on the alphanumeric keys.

The bits and words of the phrase must be entered by using the alphanumeric keys.

#### Entering a conditional or iterative phrase

**[IF][THEN][ELSE]** Permit the entry of a conditional phrase.

**[WHILE][DO]** Permit the entry of an iterative phrase.

#### Entering bit instructions

**[BIT]** Gives access to the following bit instructions :

<b>[NOT]</b>	Logical inversion.
<b>[RE]</b>	Detection of a rising edge.
<b>[FE]</b>	Detection of a falling edge.
<b>[SET]</b>	Sets the bit to 1.
<b>[RESET]</b>	Resets the bit to 0.

• **Example :** Entry of the instruction RESET B2 :

**[BIT]** Enters bit instruction.

**[RESET]** Enters the instruction RESET.

**<B><2>** Enters the bit address B2.

#### • Modifying entries

**<←><→>** Moves the cursor horizontally inside the phrase (but does not give access to the label).

**<↑><↓>** Moves the cursor vertically inside the phrase (but does not give access to the comment).

**<RUB>** Deletes the character to the left of the cursor.

**<DEL>** Deletes the character indicated by the cursor.

**<INS>** Inserts a space.

**<CLEAR>** Deletes the phrase (after confirmation).

---

**Entering a word instruction**

<b>[WORD]</b>	Gives access to the word instructions.
<b>[AND]</b>	Logical AND.
<b>[OR]</b>	Logical OR.
<b>[XOR]</b>	Exclusive OR.
<b>[INC]</b>	Incrementation.
<b>[DEC]</b>	Decrementation.
<b>[REM]</b>	Remainder of a division.
<b>[SHL]</b>	Logical shift to the left.
<b>[SHR]</b>	Logical shift to the right.
<b>[SLC]</b>	Circular shift to the left.
<b>[SRC]</b>	Circular shift to the right.
<b>[SQRT]</b>	Extraction of the square root.
<b>[CPL]</b>	Logical complement.

**Entering function block instructions**

<b>[BLOCK]</b>	Gives access to the function block instructions.
<b>[START]</b>	Starts execution.
<b>[STOP]</b>	Stops execution.
<b>[PRESET]</b>	Initialization.
<b>[RESET]</b>	Reset to zero.
<b>[UP]</b>	Upcounting.
<b>[DOWN]</b>	Downcounting.
<b>[SLCWORD]</b>	Circular shift to the left of 1 byte.
<b>[PUT]</b>	Storage.
<b>[GET]</b>	Retrieval.
<b>[INPUT]</b>	Reception.
<b>[OUTPUT]</b>	Transmission.
<b>[EXCHG]</b>	Transmission/reception.

**[EXEC]** Execution of an OFB.

- **Example** : Starting monostable M1 :

<b>[BLOCK]</b>	Gives access to the function block instructions.
<b>[START]</b>	Enters the start instruction.
<b>&lt;M&gt;&lt;1&gt;</b>	Enters the type and number of the block (monostable M1).

- **Reminder** : If the configuration parameters of the function block have not been modified, the function block is configured with its default values. The ZOOM mode enables the parameters of the block to be modified (see Section 9.3).

---

## Entering a conversion instruction

- [CONVER]** Gives access to the conversion instructions.
- [GTB]** Conversion of Gray to Binary.
  - [BTD]** Conversion of Binary to BCD.
  - [DTB]** Conversion of BCD to Binary.
  - [ATB]** Conversion of ASCII to Binary.
  - [BTA]** Conversion of Binary to ASCII.

## Entering a word table instruction

- [ARRAY]** Gives access to the word table instructions.
- [SEARCH]** Search for a word.
  - [EQUAL]** Comparison of words.

## Entering an interrupt instruction

- [INTERRUP]** Gives access to the interrupt instructions.
- [READTLG]** Reading of a telegram.
  - [READINT]** Reading of the interrupt bit.
  - [ACKINT]** Acknowledgment of an IT.
  - [MASKINT]** Masking of an IT.
  - [DMASKINT]** Demasking of an IT.
  - [SETIT]** Generation of an IT.

## Entering program instructions

- [OTHER]** Gives access to the program instructions.
- [JUMP]** Jumps the scan to another statement (identified by its label).
  - [CALL]** Subroutine call.
  - [RET]** Return from a subroutine.
  - [HALT]** Halts the program.
  - [READBIT]** Immediate reading of discrete inputs.
  - [WRITEBIT]** Immediate reading of discrete outputs.
  - [READREG]** Immediate reading of input registers.
  - [WRITEREG]** Immediate reading of output registers.
  - [READEXT]** Immediate reading of a message.
  - [WRITEEXT]** Immediate writing of a message.

### 9.4-5 Floating Point Type Instructions

These instructions are only available for PMX V5 processors which have a math coprocessor. They are not available on the TSX processors.

Floating point type instructions are entered or modified using the alphanumeric keyboard (enter entirely using letters) and cannot be accessed via the soft keys.

There are no floating point objects in PL7-3. These floating point type instructions manipulate DWi internal double words and CDWi constant double word objects or double word type OFB extract objects.

Immediate values in floating point format are only allowed for direct assignments, the values are entered with a fixed decimal point (for example 435.26) or with a floating point (for example 4.3526e+02 or 43.52e1). In all cases, the value is saved in the following format : 1 digit before the decimal point, 6 digits maximum after the decimal point and 2 digits maximum for the exponent. Taking into account a possible sign for the mantissa and the exponent, the 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 stored in floating point format. For example :

- if the value entered is 123456789.25
- the number 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

Designation	Example	Function
->	1.42 E+02->DW25	Assigning in floating point format of an immediate value in a double word.

### Calculation functions

Designation	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

Designation	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

Designation	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 greater than the second in floating point format.
INFF	INFF(DW10;DW20)->B3	Test that the first operand is less than the second in floating point format.

---

#### 9.4-6 Diagnostic Associated With a Statement

For V4 and V5 applications only, it is possible to associate a diagnostic with a statement, in order to monitor a Boolean condition or a transition. This is achieved by calling a diagnostic optional function block (OFB). Certain of these OFBs memorize the statement that is monitored. When the terminal is connected to a PLC that is running, it is not possible to delete the last diagnostic, or add the first diagnostic, of a statement that is memorized.

When a diagnostic is associated with a statement, the letter D appears in the bottom left of the statement or the bottom right for a transition.

The diagnostic code is executed after the statement with which it is associated.

**[DIAG]** Permits entry or modification of the code of the diagnostic associated with a statement. The following soft keys are displayed :

**[DEL DIAG]** deletes the diagnostic associated with a statement.

**[CONTENT]** permits entry or modification of the internal constants of the diagnostic OFB.

**[PARAM]** permits entry or modification of the I/O parameters of the diagnostic OFB.

**[IF] [THEN]** enters the keywords IF and THEN which condition the execution of the diagnostic OFB, associated with a Literal statement.

**[EXEC]** enters the keyword EXEC which commands the execution of the diagnostic OFB, associated with a Literal statement.

**<ENTER>** Validates the entries or modifications.



---

### 9.4-7 Validating a Statement

In the INSERT and MODIFY modes, after a statement is entered it must be validated so that it can be compiled and entered in memory.

**<ENTER>** Validates the statement that has just been entered and switches the terminal to the DISPLAY mode. (The statement can then be read or modified). In INSERT mode validates the statement and prompts a new insertion.

---

### 9.4-8 Deleting a Module or a Statement

#### Deleting a module

After selecting the module to be deleted on the Program Address screen :

**<DEL>** Deletes the selected module (after confirmation).

#### Deleting a statement in the DISPLAY mode

**<DEL>** Deletes the statement indicated by the pointer (after confirmation) including its label and comment. The terminal stays in the DISPLAY mode and displays the next statement.

#### Deleting a statement in the INSERT mode

**<CLEAR>** Deletes the statement being entered (after confirmation) including its label and comment. The terminal stays in the INSERT mode.

#### Deleting a statement in the MODIFY mode

**<CLEAR>** Deletes the statement (after confirmation). The terminal stays in the MODIFY mode. A second action on this key switches the terminal to the DISPLAY mode.

---

### 9.4-9 Duplicating a Statement

In the DISPLAY mode, this procedure enables a statement to be saved in a buffer memory so that it can be duplicated as often as required in any module programmed in Literal language.

#### Saving a statement

**[SAVE]** Saves the statement indicated by the pointer (after confirmation). The statement remains stored in the buffer memory until another SAVE request is received. If a diagnostic is associated with the statement, the diagnostic is also saved.

#### Restoring a statement

The statement concerned can be restored in the same program module or in any other Literal program module.

**[RESTORE]** Inserts the saved statement upstream of the one indicated by the pointer. The terminal is then in the INSERT mode so that :

- the statement can be modified (if necessary)
- and then validated by <ENTER> (whether modified or not).

The restored statement loses its label (if any).

---

## 9.5 Searching for the Program Addresses of a Variable

---

When a statement is shown on the screen in the Display mode, pressing <Alt><X> gives access to :

- The display of all the variables of the statement
- The selection of a variable and the display of all the program addresses relating to the selected variable
- The selection of a program address and the display of the corresponding program element.

The variable concerned can be a standard PL7-3 variable (Wi, Bi ...) or an OFB variable. For further information, refer to Divider H, Section 18.5.





<b>Section</b>	<b>Page</b>
<b>10 Constant Mode</b>	
Contents	10/1
10.1 Presentation of the Mode	
10.2 Constant Words	
10.3 Internal Constants of OFBs	
10.4 Handling Grafcet Situations	
<b>11 Adjust Mode</b>	
Contents	11/1
11.1 Presentation of the Mode	
<b>12 Data Mode</b>	
Contents	12/1
12.1 Presentation of the Mode	
12.2 Description of the Screen and the Keys	
12.3 Reading the Data	
12.4 Modifying and Forcing Data Objects	
<b>13 Debug Mode</b>	
Contents	13/1
13.1 Presentation of the Mode	
13.2 Predefining Screens	
13.3 Displaying Predefined Screens in Real Time	
13.4 Defining a Breakpoint	
13.5 Executing the Program	
13.6 Accessing the Other Modes	
13.7 Searching for the Program Addresses of a Variable	

---

F



<b>Sub-section</b>	<b>Page</b>
<b>10.1 Presentation of a Mode</b>	10/2
<b>10.2 Constant Words</b>	10/3
10.2-1 Displaying Constant Words	10/3
10.2-2 Modifying Constant Words	10/4
10.2-3 Storage and retrieval of Constant Words	10/6
<b>10.3 Internal Constants of OFBs</b>	10/7
10.3-1 Selection of the OFB	10/7
10.3-2 Displaying the Internal Constants of an OFB	10/8
10.3-3 Modifying the Internal Constants of an OFB	10/9
10.3-4 Storage and Retrieval of the Internal Constants of an OFB	10/10
<b>10.4 Handling Grafcet Situations</b>	10/11
10.4-1 Definition of the Descriptor of a Partial Grafcet Chart	10/12
10.4-2 Definition of a Forcing Situation	10/13

This section ends on page 10/14

---

## 10.1 Presentation of the Mode

---

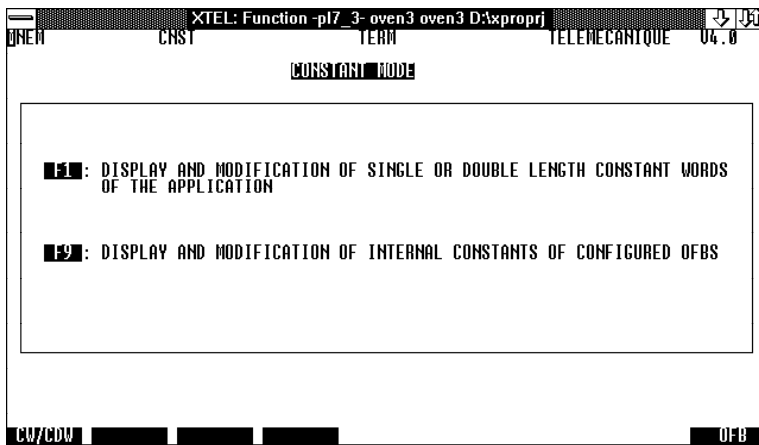
Constant mode permits the reading and modification of :

- The values of the single and double length constant words declared in the configuration. By default, the value of the constant words is 0.
- The values of the internal constants of the OFB declared in the configuration.

When the terminal is in Connected mode, constant words and OFB constant words can also be read in Adjust mode (except for the internal constants of OFBs), Data mode and Debug mode.

The number of constant words and the number and type of OFBs used must be declared in Configuration mode (see Section 6).

Constant mode is accessed from the Mode Selection display by pressing 3 - CONSTANT. The following screen is then displayed :



**[CW/CDW]** Gives access to the display and modification of the constant words, which can also be accessed by pressing <F1>.

**[OFB]** Gives access to the display and modification of the internal constants of OFBs, which can also be accessed by pressing <F9>.  
Gives access to the parameter selection required for handling Grafcet chart situations : Grafcet descriptor, forcing situations.

## 10.2 Constant Words

### 10.2-1 Displaying Constant Words

This screen shows the number of constant words configured and displays the value of the first 32 constant words.

The value of the 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 the single length words of the same rank : CDWi occupies the same location as the words CWi and CWi+1.

Each constant word is associated with a display notation (decimal by default). The possible notations are :

- Decimal (1322)
- Hexadecimal (H'3244')
- DCB (B'3244')
- Logic (L'0111 0001 0011 0000')
- Message (M'AB')
- Start of message (S'AB')
- End of message (E'AB')

The double length constant words can be framed on the CWi words with even addresses (CDW EVEN) or with odd addresses (CDW ODD). The display base is associated with 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 the CWi declared as floating in sdbase (with the letter F on the display screen) are automatically displayed in floating point base.**

CONSTANT	VALUE	CONSTANT	VALUE
CW0	=H'B104'	CW16	=0
CW1	=H'0005'	CW17	=0
CW2	=H'0005'	CW18	=0
CW3	=H'0000'	CW19	=H'B100'
CW4	=B'0000'	CW20	=100
CW5	=H'0000'	CW21	=100
CW6	=25	CW22	=0
CW7	=30	CW23	=0
CW8	=0	CW24	=0
CW9	=0	CW25	=H'0000'
CW10	=0	CW26	=0
CW11	=0	CW27	=0
CW12	=H'FFFF'	CW28	=0
CW13	=H'FFFF'	CW29	=0
CW14	=H'FFFF'	CW30	=0
CW15	=H'0000'	CW31	=0

**[CWi]** Permits the display of a constant word by entering its address in the operator input line. The word is displayed in the middle of a group of 32 words (for example for CW45, the words CW32 to CW63 are displayed).



- [CDWi]** Permits the display of a double length constant word by entering its address in the operator input line. The word is displayed in the middle of a group of 16 double words.
- [TOP]** Displays the first 32 constant words configured (or the first 16 double length words configured).
- [BOT]** Displays the last 32 constant words configured (or the last 16 double length words configured).
- [MODIF]** Permits the modification of constant words.
- [CDW EVEN]** Displays (in decimal or floating point) the double length constant words with even addresses.
- [CDW ODD]** Displays (in decimal or floating point) the double length constant words with odd addresses.
- [CW]** Returns to the display of single length constant words.
- <↑><↓>** Displays the previous or next screen, if it exists.

## 10.2-2 Modifying Constant Words

In the DISPLAY mode, access to the modification of constant words can be obtained by pressing the [MODIF] key.

The screenshot shows a terminal window with the title 'XTEL: Function -pl7\_3- oven3 oven3 D:\xpropj'. The main content is a table of constants. The table has two columns, each with 'CONSTANT' and 'VALUE' headers. The constants are listed from CW0 to CW31. The values are in hexadecimal or decimal. The 'MODIFY' key is highlighted at the bottom of the screen.

CONSTANT	VALUE	CONSTANT	VALUE
CW0	=H'B104'	CW16	=0
CW1	=H'0005'	CW17	=0
CW2	=H'0010'	CW18	=0
CW3	=H'0000'	CW19	=H'B100'
CW4	=B'0000'	CW20	=100
CW5	=H'0000'	CW21	=100
CW6	=25	CW22	=0
CW7	=30	CW23	=0
CW8	=0	CW24	=0
CW9	=0	CW25	=H'0000'
CW10	=0	CW26	=0
CW11	=0	CW27	=0
CW12	=H'FFFF'	CW28	=32
CW13	=H'FFFF'	CW29	=100
CW14	=H'FFFF'	CW30	=H'0000'
CW15	=H'0000'	CW31	=H'0000'

At the bottom of the screen, the 'MODIFY' key is highlighted. Other keys visible include 'BASE', 'NEXT', and 'TO LAST'.

- 
- [MODIFY]** Permits modification of the single or double length constant word displayed in reverse video.
- [BASE]** Modifies the notation (in succession) of the current constant word (single length word only). The possible notations are : Decimal, Hexadecimal (H), DCB (B), Logic (L), Message (M), Start of message (S) and End of message (E).
- [NEXT]** Copies the value and notation of the current constant word onto the next one.
- [TO LAST]** Copies the value and notation of the current constant word into all the subsequent words displayed on the screen.
- <↑><↓>** Move the reverse video cursor up or down the column.
- <←><→>** Move the reverse video cursor from one column to another.
- <ENTER>** Validates all the modifications made to the constant words displayed on the screen.
- <CLEAR>** Cancels the ongoing modification or entry.

### Remarks

Entries in the 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.

A 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 on the MSB side.

<b>Example :</b>	Value entered	H'8C'
	Value memorized	H'008C'

---

### 10.2-3 Storage and Retrieval of Constant Words

- [WRITE]** Permits the storage of the constant words in a file.
- [STR. NAME]** permits entry of the name of the file.
  - [DIR]** displays the list of existing files. During storage, the new file will overwrite an old file of the same name.
  - [MODULE]** assigns the name of the last module selected as the name of the file.
  - [ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.
- [FROM]** permits entry of the name or mnemonic of the first constant word to be stored.
- [TO]** permits entry of the name or mnemonic of the last constant word to be stored.
- [TOP → BOT]** permits storage of all the constant words.
- [READ]** Permits retrieval of the constant words previously stored in a xxx.CST file.
- [STR. NAME]** permits entry of the name of the file to be retrieved.
  - [DIR]** displays the list of existing files.
  - [MODULE]** assigns the name of the last module selected as the name of the file.
  - [ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.
- [FORMAT]** permits selection of the reading mode :
- SYMB : reading directed by the symbols
  - VAR : reading directed by the constant words
  - VAL : reading directed by the values
- [MODE]** defines the mode of retrieval :
- AUTO : global retrieval, with stopping on errors only
  - MANU : manual retrieval with confirmation of each element read.
- [FROM]** permits entry of the name or mnemonic of the single or double length word that will receive the first value retrieved (start of the retrieval zone).

**Caution :** If the file read contains mnemonics, these will define the zones to be retrieved. For example :

<b>File read</b>	<b>Table of mnemonics</b>
MAX = 50	MAX : CW5
CW4 = 52	MIN : CW12
MIN = 20	

Retrieval from MAX to MIN :	CW5 = 50
	CW6 = 52
	CW12 = 20

#### Note

The double words are stored and then retrieved in their display base : floating point or decimal (a CDW displayed in floating point is therefore saved in floating point).

### 10.3 Internal Constants of OFBs

#### 10.3-1 Selection of the OFB

This screen displays the internal constants of the application OFBs.

An OFB can be selected in two ways :

- 1 - By selecting the type of OFB on the current screen and then entering the number of the OFB
- 2 - By directly entering the full name of the OFB (type + number).

In both cases, validation of the entry by <ENTER> displays the internal constants of the OFB.

TYPE	NUMBER	VERSION	FAMILY	TYPE	NUMBER	VERSION	FAMILY
MUDGN :	2	U 4.0	PL7 DIAG				
EUDGS :	3	U 4.0	PL7 DIAG				
MUDGN :	4	U 4.0	PL7 DIAG				
MUDGS :	4	U 4.0	PL7 DIAG				
TRDGN :	4	U 4.0	PL7 DIAG				
TRDGS :	4	U 4.0	PL7 DIAG				
AXMLD :	4	U 0.7	AXIS				
COMLD :	4	U 0.5	COMM				

R LINEF      OFB CONSTANTS

SELECT

**[SELECT]** Permits the direct entry of an OFB (type + number), so that its internal constants can be displayed or modified.

F

### 10.3-2 Displaying the Internal Constants of an OFB

The internal constants of an OFB are presented on 3 types of screens which display respectively :

- The constants of the type bit, word or double word
- The constants of the type table of words or double word
- The message type constants.

If one of these types of constants does not exist in a given OFB, the corresponding screen remains empty.

IDENTIFIER	TYPE	VALUE	MIN	MAX
INI_T0	word	10	0	32767
INI_T1	word	20	0	32767
INI_MIN	word	0	0	32767
INI_MAX	word	20	0	32767
BASE	word	10	1	32767

**[SEARCH]** Permits entry of a new number of an OFB of the same type, so that its internal constants can be displayed.

**[BASE]** Displays the type of internal constant indicated by the cursor in the following notations in succession : Decimal, Hexadecimal, and Logic. With a PMX station, the double words can also be displayed in floating point. In this case, the MIN and MAX values are also displayed in floating point. If a value does not appear in the definition of floating point numbers, "???" is displayed.

**[MODIF]** Permits modification of the internal constants of the current OFB.

**[ARRAY]** Displays the table type internal constants of the current OFB.

**[MSG]** Displays the message type internal constants of the current OFB.

**[B,W,DW]** Displays the bit, word and double word type internal constants of the current OFB.

### 10.3-3 Modifying the Internal Constants of an OFB

This screen can be accessed from the DISPLAY OFB screen by pressing the [MODIF] key.

IDENTIFIER	TYPE	VALUE	MIN	MAX
INI_T0	word	10	0	32767
INI_T1	word	20	0	32767
INI_MIN	word	0	0	32767
INI_MAX	word	20	0	32767
BASE	word	10	1	32767

MODIFY OFB

ADJUST BASE MODIFY NEXT ARRAY MSG

**[ADJUST]** Authorizes or prohibits modification of the internal constants of the current OFB by an ADJUST software tool. Y or N appears at the top of the screen.

**[BASE]** Displays the constant indicated by the cursor in Decimal, Hexadecimal and Logic notation in succession. With a PMX station, the double words can also be displayed in floating point.

**If a double word is declared as floating in sdbase (with the letter F on the display screen) or if it is a floating type OFB element (for example an internal OFB constant from the PMS2 family), it is displayed in floating point base by default.**

**[MODIFY]** Permits modification of the internal constant indicated by the cursor. With a PMX station, the double words can also be entered as floating point values. In this case, if the immediate value entered is not in the definition of floating point numbers, a syntax error is indicated.

**[NEXT]** Copies the value of the current constant into the next one.

**[ARRAY]** Permits modification of the table type constants of the OFB.

**[MSG]** Permits modification of the message type constants of the OFB.

**[B,W,DW]** Permits modification of the bit, word or double word type constants of the OFB. With a PMX station, the double words can also be entered as floating point values. In this case, if the immediate value entered is not in the definition of floating point numbers, a syntax error is indicated.

**<ENTER>** Validates all the modifications to the constants of the current OFB.

**<CLEAR>** Cancels all the modifications to the current OFB.

---

### 10.3-4 Storage and Retrieval of the Internal Constants of an OFB

- [WRITE]** Permits the storage of the internal constants of an OFB in a file.
- [STR. NAME]** permits entry of the name of the file.
    - [DIR]** displays the list of existing files. During storage, the new file will overwrite an old file of the same name.
    - [MODULE]** assigns the name of the last module selected as the name of the file.
    - [ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.
- [READ]** Permits retrieval of the internal constants previously stored in a xxx.OFB file.
- [STR. NAME]** permits entry of the name of the file to be retrieved.
    - [DIR]** displays the list of existing files.
    - [MODULE]** assigns the name of the last module selected as the name of the file.
    - [ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.
  - [MODE]** defines the mode of retrieval :
    - AUTO : global retrieval, with stopping on errors only
    - MANU : manual retrieval with confirmation of each element read.

#### Note

The double words are stored and then retrieved in their display base : floating point, decimal or hexadecimal (a double word displayed in floating point is therefore saved in floating point).





## 10.4-1 Definition of the Descriptor of a Partial Grafcet Chart

This screen is accessible from the OPTIONAL FUNCTION BLOCK CONFIGURATION screen USING THE command [PART.] allowing the hierarchical level and the steps and macro-steps which comprise the partial Grafcet chart, associated with the selected OFB MSIT, to be defined.

The steps and macro-steps of the application are shown as a chart 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 cursor arrows or the direct access keys enable movement within the empty boxes (steps or macro-steps configured and called), so as to select the steps or macro-steps which make-up the partial Grafcet chart.

The "\*\*\*" boxes represent steps or macro-steps selected.

**[STEP]** Enables a step to be accessed directly by its number.

**[MACRO]** Enables a macro-step to be accessed directly by its number.

**[INS]** Selects a step or macro-step pointed to by the cursor.

**[DEL]** Deletes a step or macro-step pointed to by the cursor.

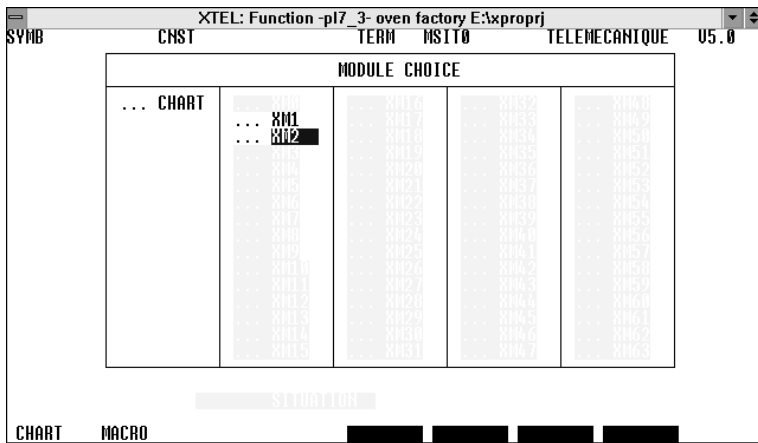
**[ERASE]** Cancels after confirmation, all the steps and macro-steps selected.

**[HIERA.]** Enables the hierarchy of a Partial Grafcet chart to be defined : 0 to 255. 0 corresponds to the highest hierarchical level, 255 to the lowest.

- <↑><↓> moves the cursor in the accessible boxes : empty or "\*\*\*" boxes.
- <←><→>
- <ENTER> Confirms the partition and updates the descriptor of the partial Grafcet chart, contained in the internal constants of the OFB MSIT.
- <CLEAR> Returns to the optional function block configuration screen.
- <QUIT> Returns to the operating modes selection screen.

### 10.4-2 Definition of a Forcing Situation

After 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, activate the command [SITUAT.] which allows the forcing situation for the partial Grafcet chart, associated with the OFB MSIT selected, to be defined. This situation will be saved in the internal constants (CW<sub>i</sub>) of the application, from the address selected. The program indicates the number of CW occupied by the new situation. After confirmation by <F9>, the module selection screen is displayed :



- [CHART]** Allows direct access to the selection of steps in the CHART module, in order to define the forcing situation.
- [MACRO]** Allows, in order to define the forcing situation, to access directly the selection of steps in a macro-step XM by its number.

<↑><↓> Move the cursor in the accessible modules.

<←><→>

<ENTER> Confirms the module selection and gives access to the forcing situation definition screen, for the module selected (CHART or XM). This screen represent the steps of the module as a table which contains :

- empty boxes : configured steps, which are called but do not appear in the partial Grafcet chart,
- "." boxes : configured module steps but not called,
- "/" boxes : non configured module steps,
- "0" boxes : steps configured, called, which are a part of the partial Grafcet chart, but not with a forcing situation,
- "1" boxes : steps configured, called, which are a part of the partial Grafcet chart, and with a forcing situation.

SYMB	0	1	2	3	4	5	6	7	8	9	TERM	MSIT0	0	1	2	3	4	5	6	7	8	9	U5.0	
X 0-				..	..	..	..	..	..	..		X10-	//	//	//	//	//	//	//	//	//	//	//	//
X 1-	1	..	0	0	0	..	..	..	..	..		X11-	//	//	//	//	//	//	//	//	//	//	//	//
X 2-	..	..	..	..	..	..	..	..	..	..		X12-	//	//	//	//	//	//	//	//	//	//	//	//
X 3-	..	..	..	..	..	..	..	..	..	..														
X 4-	..	..	..	..	..	..	..	..	..	..														
X 5-	..	..	..	..	..	..	..	..	..	..														
X 6-	..	..	..	..	//	//	//	//	//	//														
X 7-	//	//	//	//	//	//	//	//	//	//														
X 8-	//	//	//	//	//	//	//	//	//	//														
X 9-	//	//	//	//	//	//	//	//	//	//														

[STEP] Enables a step to be accessed directly by its number.

[INS] Selects the step pointed to by the cursor, so that it includes the forcing situation. The box then takes the value "1".

[DEL] Deletes selection of the step pointed to by the cursor. This no longer includes the forcing situation and the box takes the value "0".

[ERASE] Cancels after confirmation, all the steps selected.

<↑><↓> Move the cursor over the accessible steps.

<←><→>

<ENTER> Confirms after confirmation, the forcing situation for the module. The modules for which the forcing situation is defined, appear preceded by "\*\*\*\*" in the MODULE SELECTION screen.



<b>Sub-section</b>	<b>Page</b>
<b>11.1 Presentation of the mode</b>	11/2
11.1-1 Purpose	11/2
11.1-2 Objects Accessible in ADJUST Mode	11/3
 This section ends on page	 11/4

---

## 11.1 Presentation of the Mode

---

### 11.1-1 Purpose

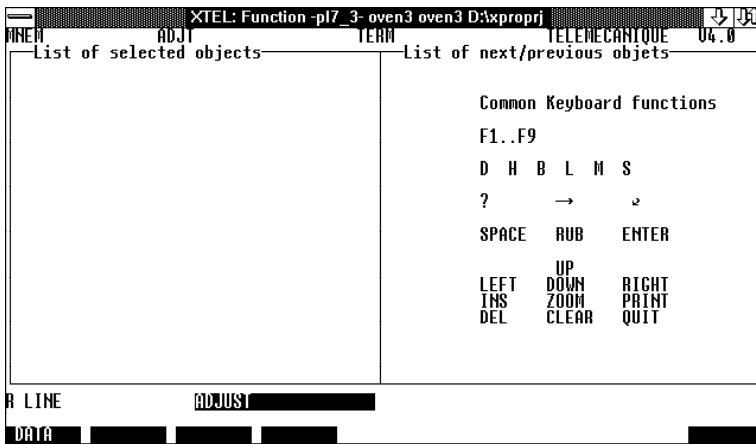
Adjust mode permits the display and modification of the PLC application data. It does not require the transfer of the application from the PLC to the terminal and does not reserve the PLC. In the Adjust mode, a terminal can be logically connected to a PLC which is already reserved.

Adjust mode has two main functions :

- The reading in real time of the values of bits and word objects (with the possibility of memorizing their "historical" values) and the search for forced bits
- The writing of the values of bits and word objects and the forcing of bits.

Adjust mode is a **submode of the Data mode**. It has all the same functions, only certain bits and words are not accessible in this mode. **The user should therefore refer to Section 12 for details concerning the use of the Adjust mode.**

To access the Adjust mode, select 4 - ADJUST on the Mode Selection display. The terminal must be in the Connected mode (PROCESSOR MEMORY) and can be either locally connected to the PLC or connected via a TELWAY, MAPWAY, ETHWAY or FIPWAY network (1).



The Adjust mode screen is exactly the same as the Data mode screen, and the keys have the same functions (see Section 12).

The screen is divided into two halves, each of which has its own specific functions and its own cursor which indicates the insertion position of a new data object. The objects are entered in the operator input line. After being entered, the object is displayed in reverse video in the half of the screen that is being used.

(1) PL7-3 only

### 11.1-2 Objects Accessible in the Adjust Mode

The tables below list the objects accessible in the Adjust mode, and the possibilities offered.

#### Bit objects

Object		Reading	Writing (RE)SET	Forcing (0/1)	Index (FD)
I/O module "in rack"	I/Oxy,i	yes	yes	yes	3(x-y-i)
I/O module "in rack"	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	Not accessible in Adjust mode			
Remote I/O module	RDx,y,i	Not accessible in Adjust mode			

#### Word objects

Object		Reading	Writing	Base	Index (FD)
Register word	I/OWxy,i	Not accessible in Adjust mode			
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 active time value	Xi,V - Xj,i,V	yes	no	yes	1(i)
Remote I/O module	STSx,y,i	Not accessible in Adjust mode			

## Bits extracted from words

Object		Reading	Writing (RESET)	Forcing (0/1)	Index (FD)
Register word	I/OW <sub>xy,i,j</sub>	Not accessible in Adjust mode			
Internal word	W <sub>i,j</sub>	yes	no	no	2(i-j)
Internal word	DW <sub>i,j</sub>	no	no	no	2(i-j)
Constant word	CW <sub>i,j</sub>	yes	no	no	2(i-j)
System word	SW <sub>i,j</sub>	yes	no	no	2(i-j)
Common word	COM(X) <sub>i,j,k</sub>	yes	no	no	3(i-j-k)
Remote I/O module	STS <sub>x,y,i,b</sub>	Not accessible in Adjust mode			

## Standard and optional function blocks

Object	Reading	Writing	Forcing (0/1)	Base	(RE) SET	Index (FD)
<b>Timer</b>						
Ti,P	yes	yes (*)	-	yes	-	1(i)
Ti,V	yes	no	-	yes	-	1(i)
Ti,R - Ti,D	yes	no	no	-	no	1(i)
<b>Monostable</b>						
Mi,P	yes	yes (*)	-	yes	-	1(i)
Mi,V	yes	no	-	yes	-	1(i)
Mi,R	yes	no	no	-	no	1(i)
<b>Counter</b>						
Ci,P	yes	yes (*)	-	yes	-	1(i)
Ci,V	yes	no	-	yes	-	1(i)
Ci,E - Ci,D - Ci,F	yes	no	no	-	no	1(i)
<b>Register</b>						
Ri,I	yes	yes	-	yes	-	1(i)
Ri,O	yes	no	-	yes	-	1(i)
Ri,E - Ri,F	yes	no	no	-	no	1(i)
<b>Text</b>						
TXTi,L/S/A/M/V/C/T	Not accessible in Adjust mode					
TXTi,D/E	Not accessible in Adjust mode					
<b>Control</b>						
CTRLi,R	Not accessible in Adjust mode					
<b>OFB</b>						
OFBi,input	Not accessible in Adjust mode					
OFBi,output	Not accessible in Adjust mode					
OFBi,constant	Not accessible in Adjust mode					
OFBi,data	Not accessible in Adjust mode					

(\*) If the block is configured with modification authorized (option YES).



<b>Sub-section</b>	<b>Page</b>
<b>12.1 Presentation of the Mode</b>	12/2
12.1-1 Purpose	12/2
12.1-2 Objects Accessible in DATA Mode	12/3
<b>12.2 Description of the Screen and the Keys</b>	12/6
12.2-1 Description of the Screen	12/6
12.2-2 Description of the Keys	12/7
<b>12.3 Reading the Data</b>	12/9
12.3-1 Reading an Object	12/9
12.3-2 Reading a list of Consecutive Objects (RH)	12/10
12.3-3 Reading the Historical List of an Object's Value	12/12
12.3-4 Searching for Forced or Blocked Objects	12/12
12.3-5 Using Soft Key [FD./.]	12/13
<b>12.4 Modifying and Forcing Data Objects</b>	12/14
12.4-1 Modifying and Forcing Bits	12/14
12.4-2 Writing a list of Consecutive Bits	12/15
12.4-3 Modifying and Blocking Grafcet Step Bits	12/16
12.4-4 Writing and Modifying Words	12/17
12.4-5 Writing a List of Consecutive Words	12/17
12.4-6 Saving and Restoring a Predefined Data Screen	12/18

This section ends on page 12/18





---

## 12.1 Presentation of the Mode

---

### 12.1-1 Purpose

The Data mode permits the display and modification of the application data of the PLC. It enables the operator to :

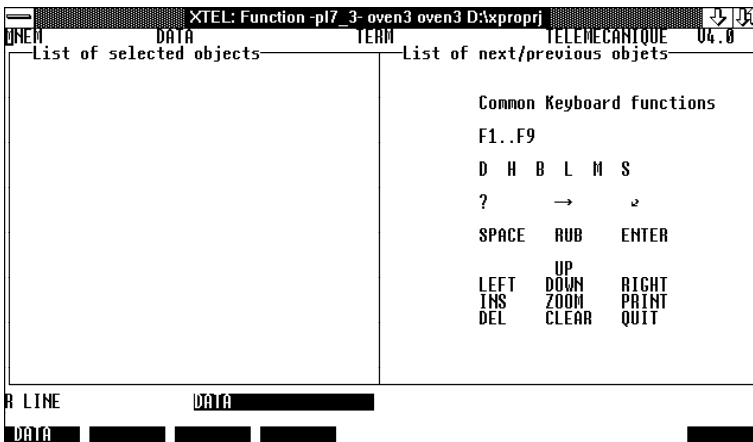
- Read the value of a bit or word in real time and memorize its "historical" values
- Write or force the state of a bit object
- Activate or block a Grafcet step
- Search for forced or blocked objects.

The Data mode has two main functions :

- **The display of memorized data.** This function is proposed by default. Only the object displayed in reverse video is shown in real time. The other objects listed on the screen are shown with their memorized values.
- **The display of continuously updated data.** All the objects are displayed in real time when the left half of the screen is used.  
When the right half of the screen is used, only the object displayed in reverse video is shown in real time, but with a list of its last 19 memorized values.

To access the Data mode, select 5 - DATA on the Mode Selection display. The terminal must be in Connected mode (PROCESSOR MEMORY) and can be either locally connected, or connected through a TELWAY, MAPWAY, ETHWAY or FIPWAY network (1).

When the terminal is logically connected to a PLC, it reserves that PLC for its exclusive use and the application is set up for use with PL7-3. In the Data mode, the terminal cannot be logically connected to a PLC which is already reserved by another terminal.



(1) PL7-3 only

If the terminal does not already have in its internal memory the application of the PLC to which it is connected, it proposes the transfer of the application from the PLC to the terminal (see Section 14).

**[YES]** Starts the transfer from the PLC to the terminal.

**[NO]** Cancels the transfer request.

### 12.1-2 Objects Accessible in the Data Mode

The tables below give the list of objects accessible in the Data mode, together with the possibilities offered.

#### Bit objects

Object		Reading	Writing (RE)SET	Forcing (0/1)	Index (FD)
I/O module "in rack"	I/Oxy,i	yes	yes	yes	3(x-y-i)
I/O module "in rack"	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(j)
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 active time value	Xi,V - Xj,i,V	yes	no	yes	1(i)
Remote I/O module	STSx,y,i	yes	no	no	1(i)

## Bits extracted from words

Object		Reading	Writing (RE)SET	Forcing (0/1)	Index (FD)
Register word	I/OWxy,i,j	yes	no	no	4(x-y-i-j)
Internal word	Wi,j	yes	no	no	2(i-j)
Internal word	DWi,j	no	no	no	2(i-j)
Constant word	CWi,j	yes	no	no	2(i-j)
System word	SWi,j	yes	no	no	2(i-j)
Common word	COM(X)i,j,k	yes	no	no	3(i-j-k)
Remote I/O module	STSx,y,l,b	yes	no	no	1(i)

## Standard function blocks

Object	Reading	Writing	Forcing (0/1)	Base	(RE) SET	Index (FD)
<b>Timer</b>						
Ti,P	yes	yes (1)	-	yes	-	1(i)
Ti,V	yes	no	-	yes	-	1(i)
Ti,R - Ti,D	yes	no	no	-	no	1(i)
<b>Monostable</b>						
Mi,P	yes	yes (1)	-	yes	-	1(i)
Mi,V	yes	no	-	yes	-	1(i)
Mi,R	yes	no	no	-	no	1(i)
<b>Counter</b>						
Ci,P	yes	yes (1)	-	yes	-	1(i)
Ci,V	yes	no	-	yes	-	1(i)
Ci,E - Ci,D - Ci,F	yes	no	no	-	no	1(i)
<b>Register</b>						
Ri,I	yes	yes	-	yes	-	1(i)
Ri,O	yes	yes	-	yes	-	1(i)
Ri,E - Ri,F	yes	no	no	-	no	1(i)
<b>Text</b>						
TXTi,L	yes	yes	-	no	-	1(i)
TXTi,S/V	yes	no	-	no	-	1(i)
TXTi,A/M/C/T	yes	yes	-	no	-	1(i)
TXTi,D/E	yes	no	no	-	no	1(i)
<b>Control</b>						
CTRLi,R	yes	no	no	-	no	1(i)

(1) If the block is configured with modification authorized (option YES).

---

**Optional function blocks**

<b>Object</b>	<b>Reading</b>	<b>Writing</b>	<b>Forcing (0/1)</b>	<b>Base</b>	<b>(RE) SET</b>	<b>Index (FD)</b>
<b>OFB</b>						
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, double words, bit tables, word tables and double word tables only.

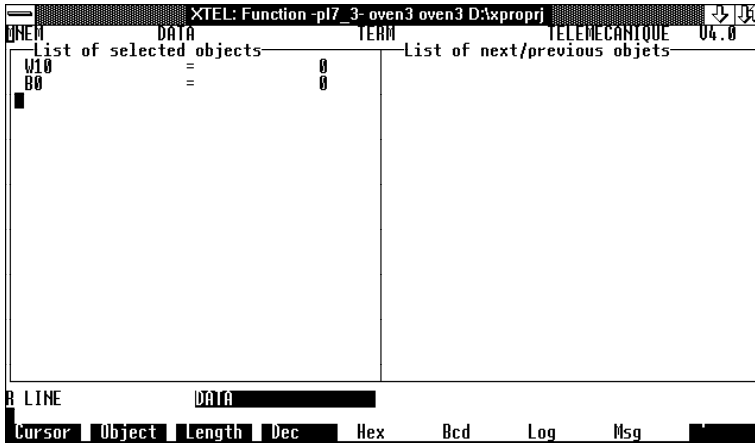
---

## 12.2 Description of the Screen and the Keys

---

### 12.2-1 Description of the Screen

The screen is divided into two halves, each of which has specific functions :



#### Right half (RH)

This half is used to display objects or lists of objects of different types, or of the same type but not in consecutive order.

The objects displayed in this half are saved when the mode is abandoned.

When the [REFRESH] key is pressed, the objects in this half are displayed in real time.

#### Left half (LH)

This half is used to display lists of objects of the same type in consecutive order.

The objects displayed in this half are not saved when the mode is abandoned.

When the [REFRESH] key is pressed, only the object displayed in reverse video is displayed in real time, but its previous states are displayed on the preceding lines.

---

### 12.2-2 Description of the Keys

On entering the mode, the right half of the screen displays the keys that can be used, under the heading "Common Keyboard Functions".

F1 to F9 are the soft keys,

D, H, B, L and M are the alphabetic keys used to display the notation in which the object will be displayed. The F key can also be accessed on the PMX station, for display in floating point notation.

#### Common keys

- <?>** Displays in the right half all the keys that can be used.
- <S>** Searches for all the forced or blocked objects of the same type as the object indicated by the reverse video cursor.
- <RUB>** Erases the contents of both halves of the screen.
- <ZOOM>** Lists the object indicated with the next 18 objects in the right half. Should be used with soft key F7 for objects with several indexes.
- <INS>** Gives access to the object indicated, so that its value can be modified in the operator input line.
- <DEL>** Gives access to the object indicated (and its value), so that its value can be modified in the operator input line.
- <CLEAR>** Returns to the previous work level of the ongoing operation. Depending on the context, can also be used to cancel an entry.
- <QUIT>** Returns to the Mode Selection display. Depending on the context, can also be used to cancel an ongoing operation.

---

## Specific keys

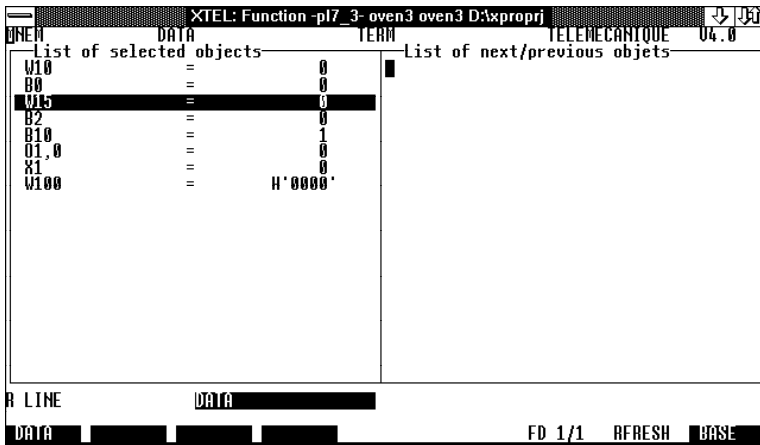
	<b>Left half (LH)</b>		<b>Right half (RH)</b>
<b>&lt;→&gt;</b>	Gives access to the RH by copying the object indicated in the LH.	<b>&lt;←&gt;</b>	Gives access to the LH at the level of the LH cursor
<b>&lt;↑&gt;</b>	Moves the cursor up.	<b>&lt;↑&gt;</b>	Displays the object having the previous address.
<b>&lt;↓&gt;</b>	Moves the cursor down.	<b>&lt;↓&gt;</b>	Displays the object having the next address.
<b>&lt;HOME &gt;</b>	Moves the cursor to the end of the list displayed.	<b>&lt;HOME &gt;</b>	Copies the object indicated in the RH at the end of the list in the LH.
<b>&lt;SPACE&gt;</b>	Erases the contents of the LH.	<b>&lt;ENTER&gt;</b>	Copies into the LH, the object indicated in the RH.
<b>[RFRESH]</b>	Displays in real time all the objects in the LH.	<b>&lt;SPACE&gt;</b>	Erases the contents of the RH except for the line indicated.
		<b>[RFRESH]</b>	Displays the historical list of the object indicated.

## 12.3 Reading the Data

### 12.3-1 Reading an Object

**[DATA]** Permits the entry of the desired object in the operator input line.

**<ENTER>** Validates the entry. The object and its value are displayed in reverse video in the left half.



**[DATA]** Permits entry of a new object in the place of the previous one.

**<↓>** Permits entry of a new object below the previous one, if it is the last one in the list. In this way a list of 19 objects can be displayed.

**[BASE]** By successive keystrokes, defines the notation in which the value of the word shown in reverse video is expressed :

- Decimal (by default)
- H = Hexadecimal
- B = Binary coded decimal
- L = Logic
- M = Message
- F = Floating point (only on PMX)

The notation can also be defined :

- By using the alphabetic keys D, H, B, L, M or F
- By using the soft keys F4 to F7 :  
[Dec], [Hex], [Bcd] and [Log]

Note : With a PMX station, if a double word is declared as floating in sdbase (with the letter F on the display screen) or it is a floating type OFB element (for example an OFB internal data item from the PMS2 family), it is displayed in floating point base by default.

As long as an object is indicated by the cursor, its state is shown in real time. When it no longer appears in reverse video, its value is no longer updated.



In addition to the usual function keys, two soft keys are proposed to facilitate the rapid entry of data objects :

- [Cursor]**      Positions the cursor at the start of the line to be entered.
- [Object]**      Displays the last object selected on the screen in the operator input line.

**Real-time display of a list of objects**

- [RFRESH]**    Displays in real time all the objects in a list of objects created in the LH. No entries are possible during the execution of this function.
- [CANCEL]**    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 real time.

**12.3-2 Reading a List of Consecutive Objects (RH)**

After having entered the first object of the list on the left half of the screen,

- [ZOOM]**      Displays the first object of the list in the right half of the screen, together with the next 18 objects. Only the object shown in reverse video is displayed in real time.

List of selected objects		List of next/previous objets	
Phot0	= 0	W100	= 0
Level	= 1	W101	= 0
Et0	= 0	W102	= 0
W100	= 0	W103	= 0
		W104	= 0
		W105	= 0
		W106	= 0
		W107	= 0
		W108	= 0
		W109	= 0
		W110	= 0
		W111	= 0
		W112	= 0
		W113	= 0
		W114	= 0
		W115	= 0
		W116	= 255
		W117	= 0
		W118	= 0

R LINE                      DATA

W100

DATA                      FD 1/1    REFRESH    BASE

The notation in which the value of word objects are displayed can be selected by using the [BASE] key. With a PMX station, this key also enables double word objects to be displayed in floating point notation.

### Remark

A list of consecutive objects can also be read by entering the length of the list at the same time that the first object is entered.

If the length requested is more than 19, as new objects are displayed at the bottom of the screen, the previous ones are erased at the top of the screen (modulo 19).

For word objects, the soft keys F4 to F7 can be used to define the notation in which they are displayed.

**Example :** Reading of bits B2 to B10 :

**[DATA]** Permits entry of the first word (B2) and the length of the list (9) as follows :

B2-9

**<ENTER>** Validates the entry.

XTEL: Function -pl7_3_ oven3 oven3 D:\xproprj					
LINE	DATA	TERM	TELEMECANIQUE		
	List of selected objects		List of next/previous objets		
B0	=	0	B2	=	0
W10	=	0	B3	=	0
W15	=	0	B4	=	0
B2	=	0	B5	=	0
B10	=	1	B6	=	0
01,0	=	0	B7	=	0
X1	=	0	B8	=	0
B2	=	0	B9	=	0
			B10	=	1

S LINE DATA

DATA SET RESET FORC 1 FORC 0 UNFORC FD 1/1 RFRESH BASE

**[Length]** Proposes a length of 1 by default, which can be modified as desired up to 32767 maximum.

### 12.3-3 Reading the Historical List of an Object's Values

This function permits the display, in the right half of the screen, the historical list of the real-time values of a data object.

- [DATA]** Permits entry of the object to be displayed.
- <ENTER>** Validates the entry and displays the object in the left half of the screen.
- <→>** Displays the object in the right half of the screen.
- [RFRESH]** Starts the display of the historical list. The real-time value of the object is updated on each scan and listed on 19 lines in the right half of the screen.
- [CANCEL]** Cancels the updating of the object on the current line.

### 12.3-4 Searching for Forced or Blocked Objects

Enter the address of the object that is the starting point of the search (by pressing the [DATA] key), or position the cursor on it if it is already displayed.

- <S>** Starts the search. When an object which is forced (indicated by F) or blocked (indicated by B) is found, the search stops and the object is displayed in the in the right half of the screen. A second action on this key restarts the search. All the objects found are displayed, so that a complete list of forced or blocked objects can be established.

XTEL: Function -pl7_3- oven3 oven3 D:\xpropj			
LINE#	DATA	TEAM	TELEMECANIQUE 04.0
List of selected objects		List of next/previous objets	
B0	=	0	B1 = 0
V10	=	0	B2 = 0
V15	=	0	B3 = 0
B2	=	0	B4 = 0
B10	=	1	B5 = 0
01.0	=	0	B6 = 0
01.1	=	1	B7 = 1 F
B100	=	0	B9 = 0 F
X1	=	0	B12 = 1 F
X2	=	1	<b>B13 = 1 F</b>
B8	=	0	
B1	=	0	

R LINE# DATA FORCED OBJECT FOUND

DATA SET RESET FORC 1 FORC 0 UNFORC FD 1/1 RFRESH BASE

**[CANCEL]** Cancels the search.

For objects with multiple indexes, the index on which the search is to be carried out is selected by using soft key F7 (see Section 12.3-5).

### 12.3-5 Using Soft Key [FD ./.]

Soft key **[FD ./.]** permits selection of the required index in the address of a data object, from which a list can be established (by pressing **[ZOOM]**) or from which a search can be made.

The screen display of soft key FD has two numbers :

- The number on the right is fixed by the terminal and depends on the number of indexes of the 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 I14,1.

**[FD1/4]** object with four indexes. Example OW22,1,3.

- The number on the left increases with successive pressures on F7 and indicates the incremented index (1 corresponds to the rightmost index, and the highest number corresponds to the leftmost index).

#### Examples

<b>[FD1/1]</b>	B1	B2	B3	....
<b>[FD1/2]</b>	W15,1	W15,2	W15,3	....
<b>[FD2/2]</b>	W15,1	W16,1	W17,1	....
<b>[FD2/3]</b>	O1,S	O2,S	I13,S	....
<b>[FD3/3]</b>	COM1,0,0	COM2,0,0	COM3,0,0	....
<b>[FD3/4]</b>	OW1,1,3	OW2,1,3	OW3,1,3	....
<b>[FD4/4]</b>	OW21,1,3	OW31,1,3	OW41,1,3	....

#### Remark

For discrete I/O bits, a list of input bits can continue with a list of output bits if there is an output module in the next slot after an input module, and vice versa.

## 12.4 Modifying and Forcing Data Objects

### 12.4-1 Modifying and Forcing Bits

To modify the state of a bit, enter the address of the bit and then use the soft keys to select the desired operation :

- [DATA]** Permits entry of the bit in the operator input line.
- <ENTER>** Validates the entry and gives access to the following soft keys :
- [SET]** Sets the bit indicated by the cursor to 1.
  - [RESET]** Resets the bit indicated by the cursor to 0.
  - [FORC 1]** Forces the bit indicated by the cursor to 1.
  - [FORC 0]** Forces the bit indicated by the cursor to 0.
  - [UNFORC]** Cancels the forced state of the bit.

LINE#	DATA	TEAM	TELEMECANIQUE	U4.0
List of selected objects		List of next/previous objets		
B0	=	0	B2	0
W10	=	0	B3	0
W15	=	0	B4	0
B2	=	0	B5	0
B10	=	1	B6	0
01,0	=	0	B7	1 F
01,1	=	1	B8	0
B100	=	1	B9	0 F
X1	=	0	B10	1 F
X2	=	1	B11	0
B8	=	0	B12	1 F
B9	=	0 F	B13	1 F
B1	=	1 F		

R LINE#      DATA

DATA    SET    RESET    FORC 1    FORC 0    UNFORC    FD 1/1    AFRESH    BASE

A forced bit is indicated by the letter F.

### Important

The forcing of all bits is cancelled by a cold restart (SY0 = 1).

### 12.4-2 Writing a List of Consecutive Bits

A list of consecutive bits can be set to 1 or 0 by adding the required length to the address of the first bit and then writing "=1" or "=0".

**Example :** Setting to 1 of bits B100 to B109 :

**[DATA]** Gives access to the operator input line, enter the bit address (B100), the length of the list (10) and "=1".

B100-10=1

**<ENTER>** Validates the entry. The list of bits is then displayed in the right half of the screen.

XTEL: Function -pl7_3-oven3 oven3 D:\xpropri				U4.0
OPEN	DATA	TERM	TELEMECANIQUE	
List of selected objects		List of next/previous objets		
B0	=	0	B2	=
W10	=	0	B3	=
W15	=	0	B4	=
B2	=	0	B5	=
B10	=	1	B6	=
O1_0	=	0	B7	=
O1_1	=	1	B8	=
B100	=	1	B9	=
X1	=	0	B12	=
X2	=	1	B100	=
B6	=	0	B101	=
B100	=	1	B102	=
			B103	=
			B104	=
			B105	=
			B106	=
			B107	=
			B108	=
			B109	=

R LINEF      DATA

DATA    SET    RESET    FORC 1    FORC 0    UNFORC    FD 1/1    REFRESH    BASE

The letter "W" after each bit indicates that the value displayed is the written value and not the real value.

### Remarks concerning I/O bits

- The forcing to 1 or 0 of an output bit while the program is running causes the activation or deactivation of the corresponding output.
- The setting to 1 or 0 of an output bit has no effect on the corresponding outputs if the program being executed also writes these bits (the program has priority).
- As long as the program being executed does not write these bits, the setting to 1 or 0 of these bits by the terminal is taken into account :
  - by the user program when it reads these bits
  - when the outputs are updated at the end of the scan cycle.

### 12.4-3 Modifying and Blocking Grafcet Step Bits

When a Grafcet step bit is displayed in the left half of the screen :

- [SET]** Activates the corresponding step. The activation of this step :
- Does not deactivate the upstream steps
  - Enables the downstream transition(s)
  - Causes execution of the actions associated with the step (actions on activation and continuous actions).
- [RESET]** Deactivates the corresponding step, but the actions on deactivation are not executed.
- [BLK]** Blocks the corresponding step, ie the downstream transition(s) are disabled.  
The Grafcet evolves up to the blocked step and stops on this step, whether the downstream transition conditions are true or false.  
Several steps can be blocked simultaneously.
- [UNBLK]** Unblocks the corresponding step, ie the downstream transition(s) are enabled.

XTEL: Function -p17 3- oven3 oven3 D:\xproj			
LINEF	DATA	TERM	TELEMECANIQUE 04.0
List of selected objects		List of next/previous objects	
B0	=	0	B2
W10	=	0	B3
W15	=	0	B4
B2	=	0	B5
B10	=	1	B6
O1_0	=	0	B7
O1_1	=	1	B8
B100	=	1	B9
X1	=	0	B12
X2	=	1	B100
B6	=	0	B101
X0	=	1 B	B102
X1	=	0	B103
			B104
			B105
			B106
			B107
			B108
			B109

DATA SET RESET BLK UNBLK FD 1/1 REFRESH BASE

- The blocking of a step has no effect on the evolution of other independent Grafcet charts.
- A blocked step is identified on the screen 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 will be activated.

### 12.4-4 Writing and Modifying Words

The value of a word can be modified directly by writing the word and its required value in the operator input line

**[DATA]** Permits entry of the word, for example W1=17217.

The following soft keys permit selection of the notation.

**[Dec]** Decimal W1 = 17217  
**[Hex]** Hexadecimal W1 = H'4341'  
**[Bcd]** Binary coded decimal W1 = B'4341'  
**[Log]** Logic (binary) W1 = L'0100 0011 0100 0001'  
**[Msg]** Message W1 = M'AC'  
**[ ]** Ends the entry of the value of the word in the selected notation.  
**<ENTER>** Validates the entry.

### 12.4-5 Writing a List of Consecutive Words

A list of consecutive words can be written by entering the length, the notation and the required value after the word address.

**Example :** Writing of the 10 constant words CW10 to CW19 with the hexadecimal value H'FA10'.

**[DATA]** Permits entry of the word address and the length of the list : CW10-10.

**[Hex]** Selects hexadecimal and permits entry of the value : FA10.

**[ ]** Ends the entry of the value.

**<ENTER>** Validates the entry. The list of constant words is displayed in the left half of the screen.

XTEL: Function -pl7_3- oven3 oven3 D:\xproprj		TELEMECANIQUE U4.0	
LINE#	DATA	TERM	TELEMECANIQUE
List of selected objects		List of next/previous objets	
B0	=	CW13	=
B9	=	CW14	=
W15	=	CW15	=
B2	=	CW16	=
B10	=	CW17	=
01_0	=	CW18	=
X0	=	0 B	H'FA10'
X1	=	CW11	=
01_1	=	CW12	=
B100	=	CW13	=
X2	=	CW14	=
B8	=	CW15	=
B1	=	CW16	=
B12	=	CW17	=
B13	=	CW18	=
B14	=	CW19	=
B14	=	CW10	=
CW10	=	H'FA10'	=
		CW11	=
		CW12	=

S LINE# DATA

FD 1/1 RFRESH BASE H



---

## Remarks

- The modification of output register words has no effect on the corresponding intelligent module if the user program also writes these words.
- Should the program 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 at the end of the scan cycle.

## Note

The preset words of function blocks cannot be modified unless modification of the function block parameters was authorized during the configuration procedure (option YES).

---

## 12.4-6 Saving and Restoring a Predefined Data Screen

**[Write]** Enables a list of objects displayed in the left half of the screen to be saved.

**[STR.NAME]** enables the name of the file to be defined.

**[DIR]** gives the list of existing files. When saving the new file will erase the old file with the same name.

**[STATION]** assigns the name of the station as the file name.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the file name.

**[Read]** Enables a list of data previously saved in a file to be restored.

**[STR.NAME]** defines the name of the file to be restored.

**[DIR]** gives the list of existing files.

**[STATION]** assigns the station name as the name of the file to be retrieved.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file to be retrieved.



<b>Sub-section</b>	<b>Page</b>
<b>13.1 Presentation of the Mode</b>	13/2
13.1-1 Purpose	13/2
13.1-2 Diagram of the Possibilities of the Mode	13/3
13.1-3 Debug Mode Screen	13/6
<b>13.2 Predefining Screens</b>	13/7
13.2-1 Purpose	13/7
13.2-2 Predefining a Program Screen	13/8
13.2-3 Predefining a Data Screen	13/9
13.2-4 Saving and restoring a Predefined Screen	13/11
<b>13.3 Displaying Predefined Screens in Real Time</b>	13/12
13.3-1 Purpose	13/12
13.3-2 Displaying a Statement or Ladder Network	13/14
13.3-3 Displaying a Predefined Grafcet Chart	13/16
<b>13.4 Defining a Breakpoint</b>	13/17
13.4-1 Inserting and Deleting a Breakpoint	13/17
<b>13.5 Executing the Program</b>	13/20
13.5-1 Execution	13/20
13.5-2 Influence of the Master Task on the Outputs	13/23
<b>13.6 Accessing the Other Modes</b>	13/24
13.6-1 Accessing the Operator Input Line	13/24
13.6-2 Accessing the Other Modes	13/24
<b>13.7 Searching for the Program Addresses of a Variable</b>	13/25
This section ends on page	13/26



---

## 13.1 Presentation of the Mode

---

### 13.1-1 Purpose

After the program has been entered on the terminal in the Program mode, the execution of the program by the PLC must be checked and debugged. The Debug mode offers a range of functions which allow the user to execute, display and modify the program.

Debug mode is only accessible when the terminal is operating in Connected mode, and enables the operator to :

#### **Start or stop execution :**

- Of the program
- Of one or more tasks

#### **Display in real-time :**

- Up to 6 predefined screens each containing :
  - A Ladder network (1)
  - A Literal statement (2)
  - A Grafcet page
  - A list of data objects
- The Ladder networks, Literal statements, or Grafcet pages upstream or downstream of the predefined screens
- A data object simultaneously with a predefined screen

#### **Modify (without leaving Debug mode) :**

- A program
- The function block parameters

#### **Insert a breakpoint in a task so that it can be executed :**

- Cycle by cycle
- Ladder network by Ladder network (1)
- Statement by statement (2)

#### **Access the Data mode so as to :**

- Read and write bit and word objects
- Force bit objects
- Block Grafcet steps
- Force the state of I/O bits

#### **Access the Search/Replace mode so as to :**

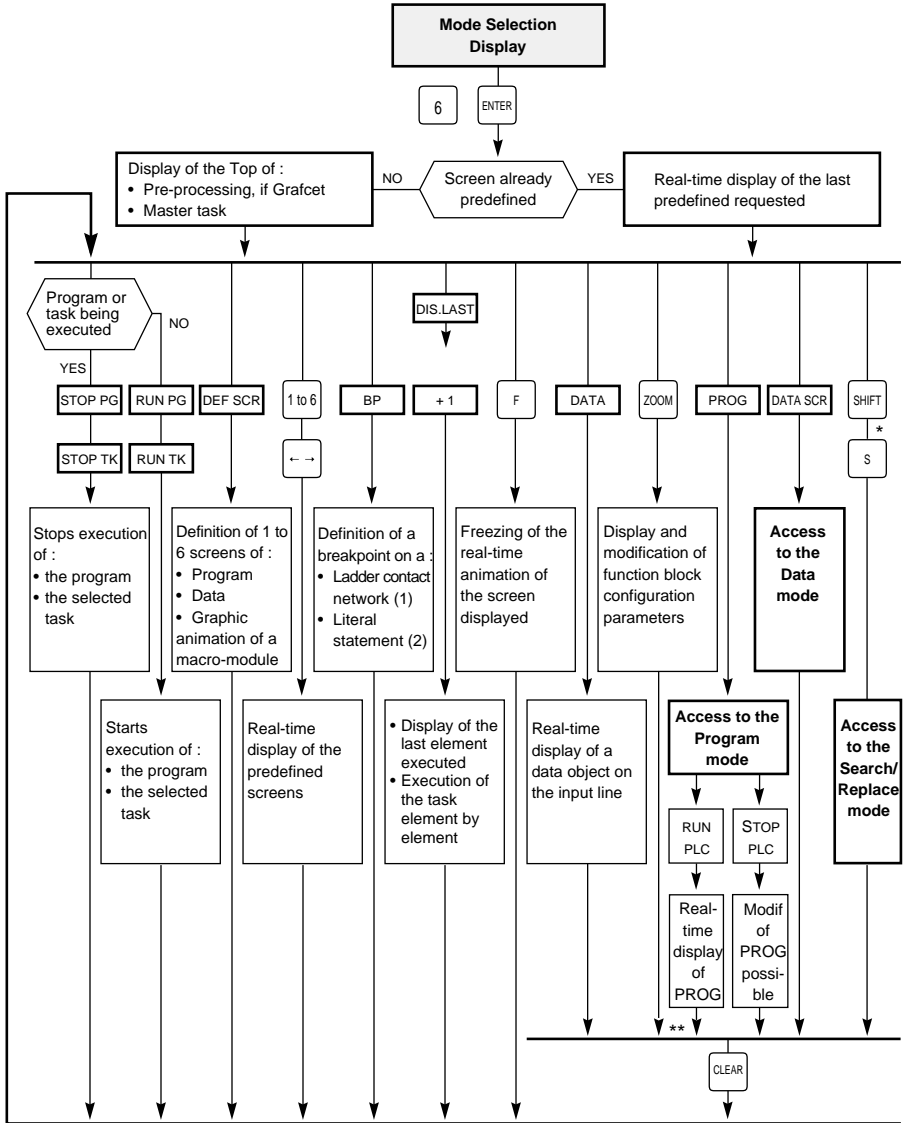
- Search for an object in all or part of the program
- Replace the objects by compatible objects in all or part of the program

#### **Freeze the real-time display on the screen**

(1) except PL7-3 GLT

(2) except PL7-3 GLD

13.1-2 Diagram of the Possibilities of the Mode



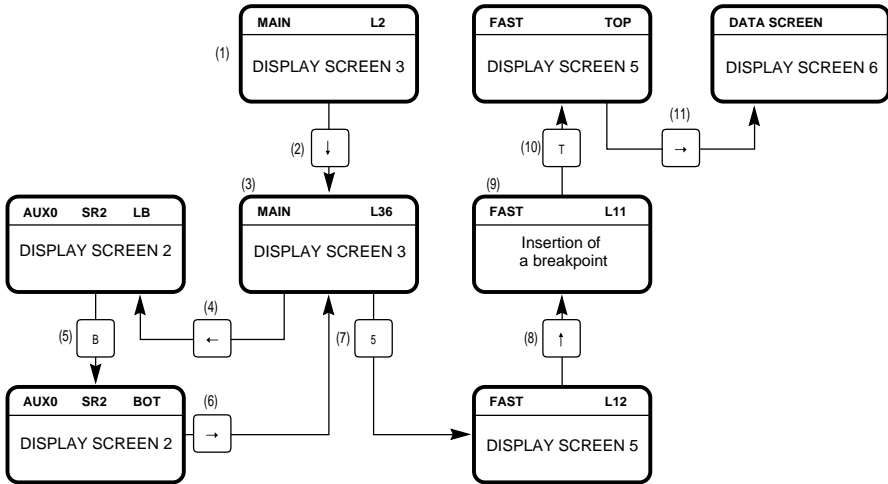
- \* <Shift><S> : Save
- <R> : Restore
- <S> : Search/Replace

\*\* Modification of program is possible

(1) except PL7-3 GLT

(2) except PL7-3 GLD

## Example of displaying predefined screens



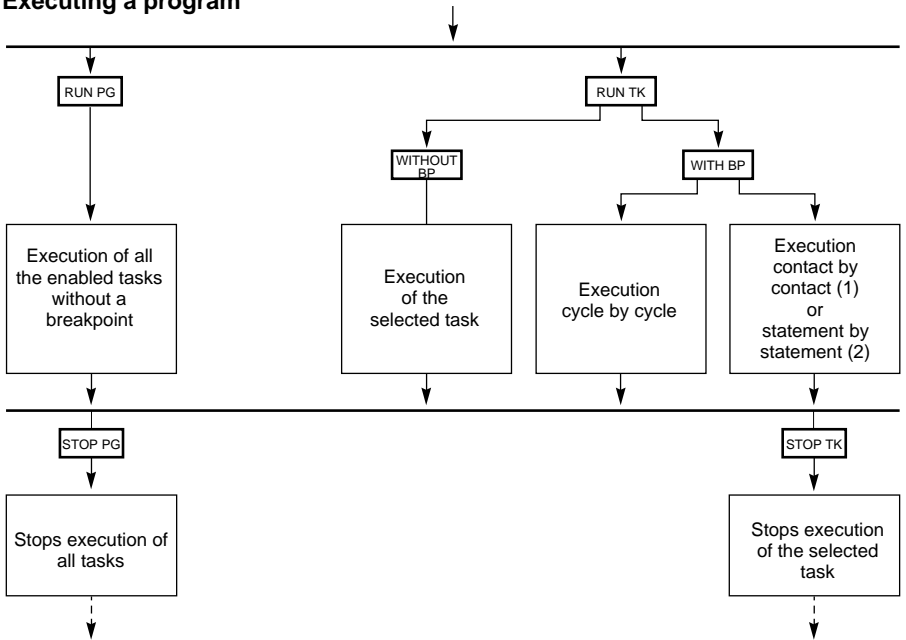
- (1) Real-time display of predefined screen No. 3 (DISPLAY SCREEN 3).  
 <↓> (2) Scrolls the program downwards with real-time display of each Ladder network, Literal statement, or Grafcet page.

### Note

Display with or without real-time animation is selected by pressing keys Y or N. If N is pressed, the program is scrolled up or down as fast as in Local mode.

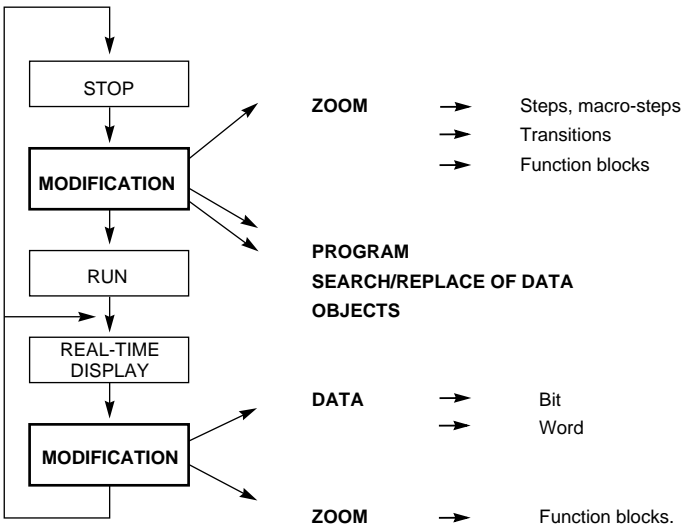
- (3) Real-time display of label 36 of DISPLAY SCREEN 3.  
 <←> (4) Displays the predefined screen with the previous number (DISPLAY SCREEN 2). In this case, label 8 of SR2 of AUX0.  
 <B> (5) Accesses the bottom of the module of DISPLAY SCREEN 2.  
 <→> (6) Displays the predefined screen with the immediately following number. In this case, the last screen that was displayed in DISPLAY SCREEN 3 (ie label 36) appears.  
 <5> (7) Keys 1 to 6 immediately access the corresponding predefined screens. In this case, DISPLAY SCREEN 5, Fast task, label 12.  
 <↑> (8) Scrolls the program upwards with real-time display of each Ladder network or Literal statement.  
 (9) Inserts a breakpoint on label 11 of the Fast task.  
 <T> (10) Accesses the top of the module (top of the Fast task).  
 <→> (11) Accesses (in this example) a data screen.

### Executing a program



### Modifying a program

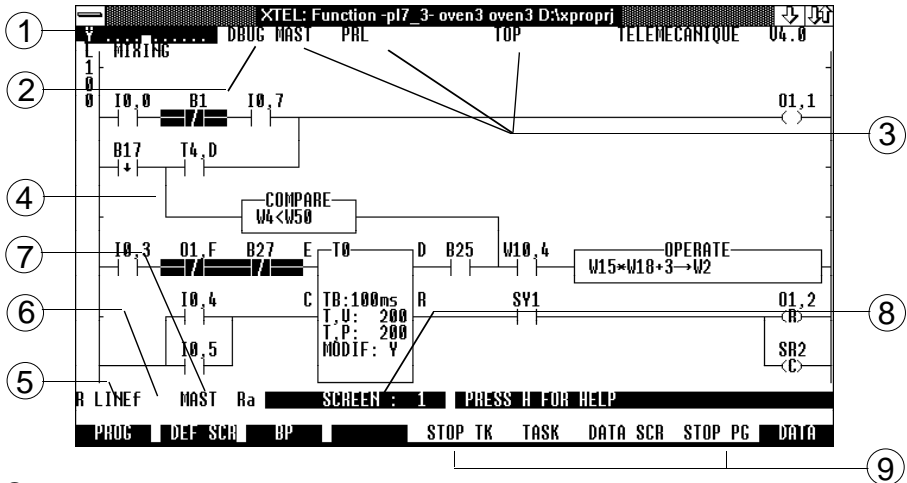
The Debug cycle time is very short.



(1) except PL7-3 GLT (2) except PL7-3 GLD

### 13.1-3 Debug Mode Screen

The information displayed in the Debug Mode screen is described below :



- ① Updating indicator of the screen, displayed in real time.
- ② Operating mode : DBU.
- ③ Program address of the element displayed :
  - Task : MAST, FAST, AUX
  - Module : MAIN, SR, CHART, XM, PRL, POST
  - Element :
    - Lx if the element is a Ladder network (1) or a Literal statement (2)
    - page x if the element is a Grafcet page.
- ④ Zone reserved for the element displayed in real time.
- ⑤ Type of logical connection to the PLC :
  - LINE displayed in steady state and coloured blue : Locally connected
  - LINE displayed blinking and coloured yellow : Network connected (3)
  - S before LINE indicates that the PLC is stopped
  - R before LINE indicates that at least one task is running
  - f after LINE indicates that at least one bit is forced.
- ⑥ BP indicates that a breakpoint has been defined in one of the tasks and /■ indicates that a step is blocked.
- ⑦ Task status zone, identifying the task selected on entering the mode and indicating its status :
  - R Task running
  - S Task stopped
  - B Task active but stopped on a breakpoint
  - s Task not active (not activated by a Control block CTRLi)
  - a Task active (activated by a Control block CTRLi).
- ⑧ Type of work mode (DEFINE SCREEN, DISPLAY SCREEN, etc).
- ⑨ Soft keys for stopping or starting the program or the selected task in the task status zone.

(1) except PL7-3 GLT (2) except PL7-3 GLD (3) PL7-3 only

---

## 13.2 Predefining Screens

---

### 13.2-1 Purpose

This function permits the predefinition, and storage in the internal memory of the terminal, of up to 6 screens. Each screen can be composed of :

- For a program screen, a Ladder network, a Literal statement, or a Grafcet page
- For a data screen, a list of any type of addressable data object.

The predefined screens remain in the terminal's memory as long as the terminal remains logically connected to the same PLC, and also when the terminal is switched from Connected to Local mode and vice versa, providing that no transfer of program is necessary.

If there is no predefined screen stored in memory, the terminal displays the first Ladder network, or the first Literal statement of the Master task, or of the Pre-processing module if Grafcet has been declared.

Otherwise, the last predefined screen that was displayed before changing the work mode is displayed on the screen.

**<INS>**           Accesses the first empty screen available.

**<1> to <6>**       Accesses the screen with the corresponding number. The cursor control keys (left and right) can also be used to access the required screen.

Two possibilities can then occur :

#### **The selected screen has not been defined**

The terminal displays the message "SCREEN NOT DEFINED, Press F2 or R"

**[DEF SCR]**       Accesses the type of screen to be defined.

**[PROG]** defines a program screen or graphic animation screen of a macro-module

**[DATA]** defines a data screen.

**<R>** restores a screen that has been saved.

#### **The selected screen has already been defined**

The corresponding program or data elements are displayed

**[DEF SCR]**       Permits modification of the predefined program or data screen.

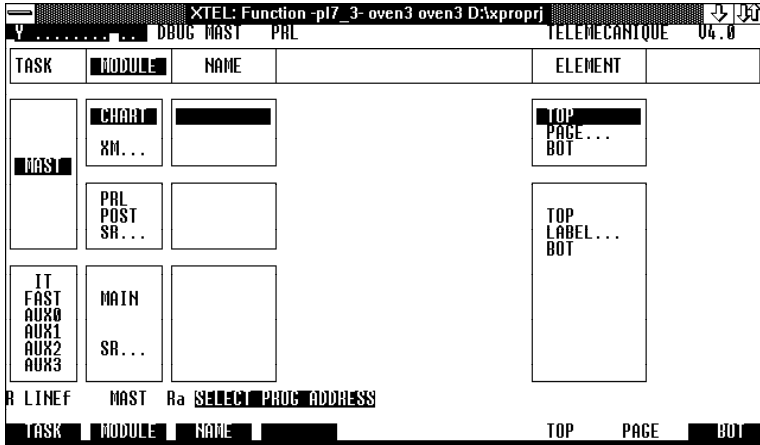
**<DEL>**           Erases the predefined screen, so that another program or data screen can be defined.



### 13.2-2 Predefining a Program Screen

After selecting the number of the screen to be defined,

**[PROG]** Calls up the Program Address selection screen which permits entry of the program element to be stored in memory as a predefined screen.



The address of the element is defined by the task and program module in which it is located, and by its label, or page number if it is a Grafcet element.

**[TASK]** enables selection of the task : MAST, IT, FAST, AUX0 to AUX3.

**[MODULE]** enables selection of the module in the current task : MAIN, PRL, SR, POST, CHART, XM.

**[NUMBER]** enables the current AUX, SR or XM module number to be defined

**[NAME]** gives access to the application List of Modules (see Section 7.1-3).

**[MOD SCR]** accesses the module Selection screen so as to give direct access to an elementary module (see Section 7.1-2).

**[PAGE]** permits a Grafcet page number to be defined as a program element (with a CHART or XM module).

**[LABEL]** permits a label number to be defined as a program element (with a MAIN or SR module).

**[TOP]** defines the start of the module (first label or Grafcet page number) as a program element (selected by default).

**[BOT]** defines the end of the module (last label or Grafcet page number) as a program element.

**[WRITE][READ]** enable the current program address to be saved/restored (see Section 13.2-4)

**<ENTER>** validates the above entries and displays the selected predefined screen (task, module, program element).

### 13.2-3 Predefining a Data Screen

This function enables the user to define a list of addressable data objects that can be stored in memory as a predefined screen.

This list is displayed in two columns of 21 lines. All types of addressable data objects can be mixed in the same list and the objects can be entered in any order, but the list can only contain objects that have been configured.

Refer to Section 12 for the list of accessible data objects and the notations in which word values, etc can be written.

After selecting the number of the screen to be defined,

- [DATA]** Permits entry in the operator input line of the required object(s). Each entry can be made either :
- Object by object (eg I14,0)
  - Or as a string of objects of the same type (eg W12-W18). The length of the string is limited to the capacity of one column (ie 21 objects).
- <ENTER>** Validates each entry. The cursor is then positioned at the left of the column to allow a new entry.
- <CLEAR>** Positions the cursor on the last object entered.
- <→>** Permits entry of objects in the right-hand column.

#### When the cursor is positioned on an object,

**[TOP]** positions the cursor at the top of the column.

**[BOT]** positions the cursor at the bottom of the column.

**[QUEUE]** positions the cursor under the last object entered and gives access to the input line so that a new object can be entered.

**[ERASE]** erases the entire column indicated by the cursor (after confirmation).

**<←><→>** move the cursor from one column to another.

**<↑><↓>** move the cursor up and down the column.

**<DEL>** deletes the object indicated by the cursor and moves the following objects up.

**<INS>** allows an object to be inserted at the point indicated by the cursor and moves the following objects downwards.

---

When the cursor is positioned at the top or bottom of a column :

<↑> displays at the top of the column, the previous object of the same type.

<↓> displays at the bottom of the column, the next object of the same type.

These keys display only the preceding or following objects of the same type.

**Example :** The last object in the RH column is text block TXT0,T

<↓> Completes the list of all the text block objects.



**<ENTER>** Validates all the entries and displays the objects in real time.

**[WRITE] [READ]** allows the list of objects to be saved/restored (see Section 13.2-4).

### Saving and restoring a predefined screen

This function allows a predefined program or data screen to be saved and restored as required :

**<S>** Saves the predefined screen that is displayed. This screen can then be restored wherever it is required by proceeding as follows :

---

**To restore on an empty screen**

- <INS>** Selects the first empty screen available.
- <R>** Restores the saved screen. The same screen can be restored several times.

**To restore on a screen that has already been defined**

- <R>** Requests erasure of the predefined screen and, after confirmation, replaces it with the saved screen.

**Deleting a predefined screen**

- <DEL>** Deletes the predefined screen after confirmation.
- <RUB>** This key saves the predefined screen and then erases it without requesting confirmation. It is equivalent to the keystrokes <S> for save, <DEL> and [YES].

---

**13.2-4 Saving and Restoring a Predefined Screen**

This function allows a program address or a data list to be saved to a file and then restored at a later time.

- [WRITE]** saves to a file :
- from the program address selection screen : the current program address,
  - from a data screen : the list of objects displayed on the screen.

- [READ]** allows the contents of a file previously saved to be restored :
- program address
  - list of objects displayed on the screen

**[STR.NAME]** allows the name of the file to be defined (saved or restored).

**[DIR]** gives the list of existing files. When a new file is saved it erases the old one with the same name.

**[STATION]** assigns the name of the station as the file name (saved or restored).

**[ARCHIV]** assigns the global archive name (PL7\_3) as the file name (saved or restored).

## 13.3 Displaying Predefined Screens in Real Time

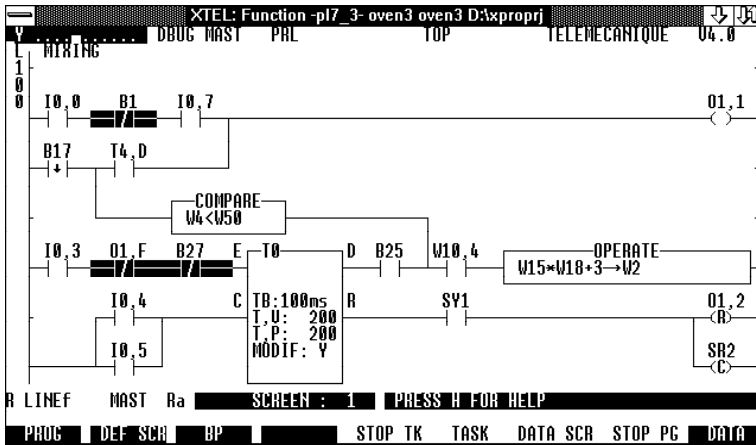
### 13.3-1 Purpose

This function permits the display in real time of the graphic elements of a Ladder (1) or Grafcet program. In Literal language (2), all the objects which make up a statement are displayed with their states and their contents.

The states or contents of the objects are read at the start of the scan cycle, during exchanges with the terminal. However, the updating of these real-time values on the screen depends on the language used :

- For a program element, the real-time values are updated when the Ladder network or Literal statement has been executed
- For a Grafcet page or a list of data objects, the real-time values are updated at the start of the scan cycle of the Master task.

An updating indicator, consisting of a progressive array of 11 dots, is displayed in the top left corner of the screen. The highlighting of each dot corresponds to the updating of the screen, providing that the task is being executed, that real-time animation has been selected (Y), and that the task is not frozen.



(1) except PL7-3 GLT

(2) except PL7-3 GLD

- 
- <?> or <H>** Gives access to a Help screen.
- <T> or <B>** (Top or Bottom) displays in real time the first or last element or page in the module that was selected during the predefinition of the screen.
- <Y> or <N>** (Yes or No) confirms or cancels the real-time animation of the screen. This option is particularly important since if <N> is pressed, the program can be scrolled up and down as fast as in Local mode by using the keys <↓> and <↑>.
- <F>** (Freeze) freezes the real-time updating of the screen displayed (program or data) to permit the operator to analyse the situation. Pressing this key a second time, or any other key of the Debug mode, unfreezes the screen.
- <D>** Displays the diagnostic program associated with the predefined program element.
- <SPACE>** Disables or enables the keyboard audio warning system that indicates keying errors.
- <QUIT>** Quits the Debug mode.

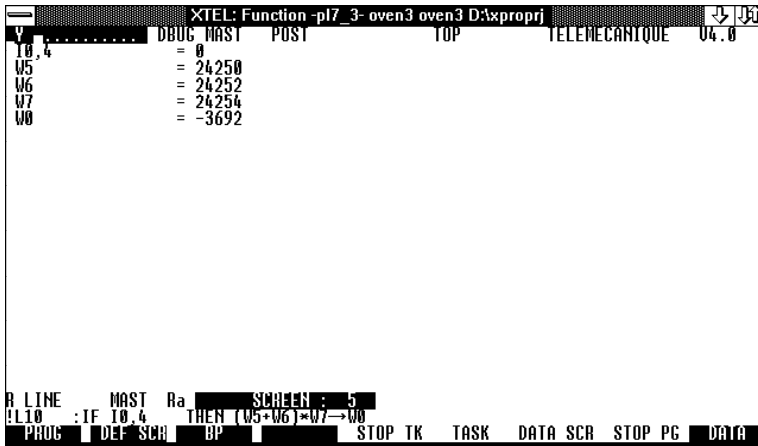
---

## 13.3-2 Displaying a Statement or Ladder Network

### Displaying a Literal statement (1)

The predefined Literal statement is displayed in the operator input line (just above the soft key menus).

The real-time values of all the data objects in the Literal statement are displayed on the screen in one or two vertical columns.



### Displaying a predefined Ladder network (2)

The elements of the Ladder network are displayed in real time as follows :

- **Contacts**

When the contact is conductive, the graphic symbol is shown in a different colour (in reverse video).

- **Coils**

When the associated bit is at state 1, the graphic symbol is shown in a different colour (in reverse video).

The JUMP and SR CALL output coils are not displayed in real time.

- **Function blocks**

When the output of the block is at state 1, it is shown in a different colour (in reverse video) with its values evolving in real time. The outputs of vertical and horizontal comparison blocks do not change colour.

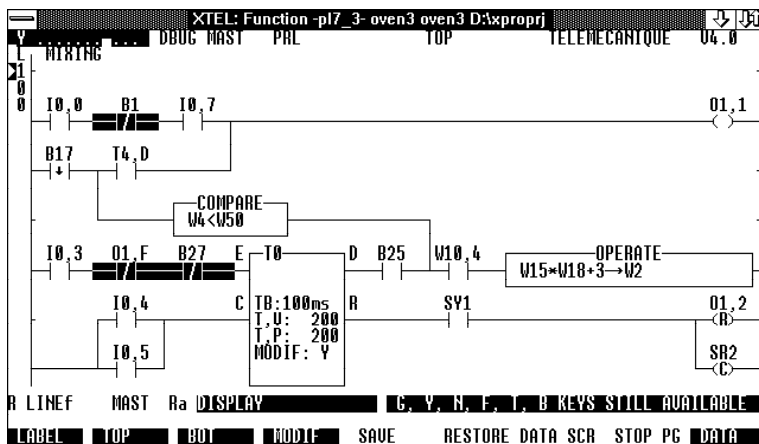
- **Operation blocks and horizontal comparison blocks**

The contents of these blocks are displayed in real time by using the ZOOM key.

(1) except PL7-3 GLD

(2) except PL7-3 GLT

Using the soft keys



- [LABEL]** Displays a program element by entering its label number.
- [TOP]** Displays the first program element of the module.
- [BOT]** Displays the last program element of the module.
- [MODIF]** Permits modification of the displayed program element.
- [SAVE]** Saves the current program element to a buffer memory.
- [RESTORE]** Restores the program element which was previously saved by SAVE.
- [DATA SCR]** Provides access to the Data mode (see Section 12).
- [STOP PG]** Stops the program.
- [RUN PG]** Starts the program.
- [DATA]** Displays a data object in the operator input line (see Section 13.6-1).
- <<-->>** Displays the preceding or following program element.
- <INS>** Permits insertion of a program element above the displayed Literal line (1) or Ladder network (2).
- <DEL>** Deletes the displayed program element.
- <ZOOM>** Provides access to the contents of function blocks and subroutines (see Section 8.3 or 9.3).
- <d> or <D>** Displays the diagnostic associated with a Ladder network.





(1) except PL7-3 GLD

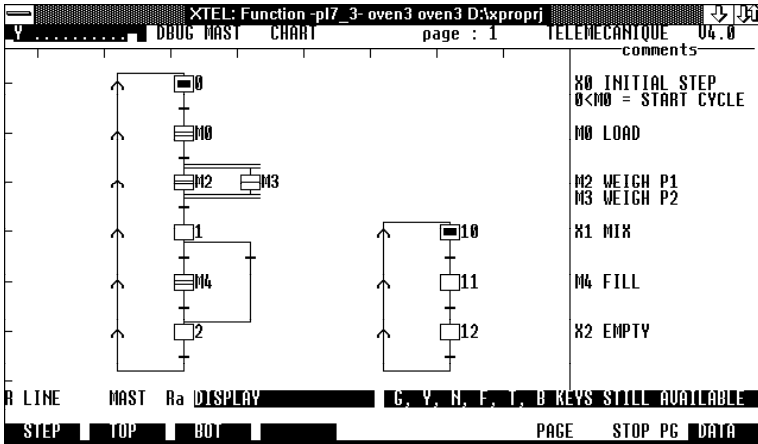
(2) except PL7-3 GLT



### 13.3-3 Displaying a Predefined Grafcet Screen

The graphic symbols for Grafcet steps and macro-steps are displayed in real time as follows :

-  Active step
-  Active initial step
-  Active macro-step with at least one step active other than the OUT step
-  Active macro-step in which only the OUT step is active



#### Note

The Grafcet pages are updated at the start of the scan cycle of the Master task, during exchanges with the terminal. The updating indicator is displayed whether the task is running or stopped.

- [STEP]** Gives access to the Grafcet page containing the specified step.
- [TOP]** Displays the first page of the Grafcet module.
- [BOT]** Displays the last Grafcet page configured.
- [MODIF]** Permits modification of the displayed Grafcet page.
- [DATA SCR]** Gives access to the Data mode (see Section 12).

---

<b>[STOP PG]</b>	Stops the program.
<b>[RUN PG]</b>	Starts the program.
<b>[DATA]</b>	Permits display of a data object in the operator input line (see Section 13.6-1).
<b>&lt;&lt;→&gt;&gt;</b>	Displays the preceding or following page.
<b>&lt;DEL&gt;</b>	Deletes the program element displayed.
<b>&lt;ZOOM&gt;</b>	Gives access to the actions and transition conditions, and also to the expansion of a macro-step (see Section 7.3).

---

## 13.4 Defining a Breakpoint

---

### 13.4-1 Inserting and Deleting a Breakpoint

A breakpoint can be defined in a Ladder network or a Literal statement in any program module of the tasks configured. The program module concerned can be :

- The Main program (MAIN)
- A subroutine (SR)
- The Pre- or Post-processing sections of the Master task (PRL or POST).

#### Defining a breakpoint

On any predefined screen :

- [BP]** Gives access to the following possibilities :
- [BP=SCR]** positions the breakpoint on the predefined screen that is displayed.
  - [DEF BP]** Gives access to the Program Address selection screen so that the task, module and element on which the breakpoint is to be defined can be selected.

If a breakpoint has already been defined, it will be deleted. The terminal then displays one of the following messages : "PREVIOUS BP HAS BEEN DELETED" or "BP = CURRENT SCREEN (OLD BP DELETED)".

Deletion of the previous breakpoint does not become effective until the task in which it was inserted is executed by pressing the [RUN TK] key.

```

XTEL: Function -pl7_3- oven3 oven3 D:\xproj
V ..... DBUG MAST POST TOP TELEMECANIQUE 04.0
I0.4 = 0
W5 = 24250
W6 = 24252
W7 = 24254
W8 = -3692

A LINEF // MAST Ra SCREEN : 3
L10 : IF I0.4 THEN (W5+W6)*0.7→W8
BP=SCR DEF BP ... DIS BP DEL BP

```

When a breakpoint has been defined, the letters "BP" are displayed in the Task status zone as follows :

- **Steady and coloured blue**, if the breakpoint is defined on a task that is stopped, or on an element that is not executed (eg an SR that is not called, or an element that is jumped). In this case the breakpoint is defined but not effective.
- **Blinking and coloured yellow**, if the breakpoint is defined on an element that is executed. In this case the breakpoint is effective and the execution of the task is stopped on the element on which the breakpoint is defined (but the timers and monostables continue to run).

The blocking of this task has no effect on the execution of the other tasks.

### Displaying a breakpoint

**[DIS BP]** Displays the element on which the breakpoint is defined.

### Deleting a breakpoint

**[DEL BP]** Requests deletion of the breakpoint (confirmed by YES). However, if the task is blocked, deletion of the breakpoint is not effective until the task is executed by pressing [RUN TK]. "BP" then disappears from the Task status zone.

The soft keys [RUN TK] and [STOP TK] are effective only on the task that is displayed in the Task status zone (just above the soft key menus).

If the operator tries to access the Program mode by pressing the [PROG] key, the terminal proposes the deletion of the breakpoint, the stopping of the program and the initialization of the tasks containing a breakpoint; since no modification can be made to the program while a breakpoint is effective.

**Note**

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 is deleted as soon as the task concerned is executed.

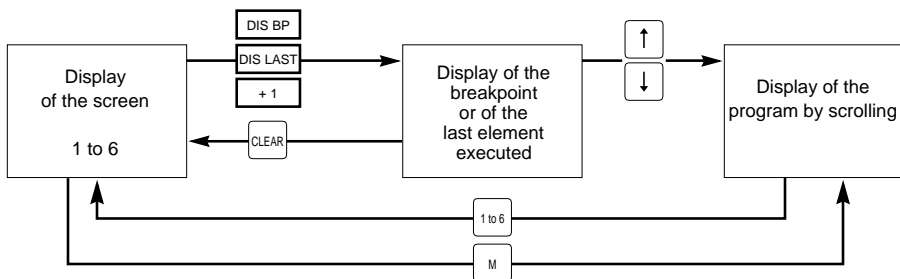
**Accessing a "Memo" screen**

**[DIS BP]** Displays the point in the program where the breakpoint is located.

**[DIS LAST]** Displays the last element that was executed.

The program can then be scrolled up and down to display all the elements upstream or downstream. But before scrolling starts, the terminal automatically memorizes the last screen that was displayed (called the "Memo" screen), so the user can return directly to it subsequently.

**<M>** Enables the operator to return to this Memo screen



Although the memorization of this screen is automatic, the Memo screen can be used as a seventh predefined screen, as follows :

**[DEF M]** Gives access to the predefinition of the Memo screen, by selection of the task, module and program element (label, Grafcet page, etc).

**<M>** Restores the Memo screen, which is displayed in real time.

**Remark**

The predefined Memo screen will be automatically erased and replaced by the last screen displayed if the keys [DIS BP], [DIS LAST] or [+1] are used.

## 13.5 Executing the Program

### 13.5-1 Execution

#### Possible states of a periodic task

The diagram opposite shows the possible states of a periodic task.

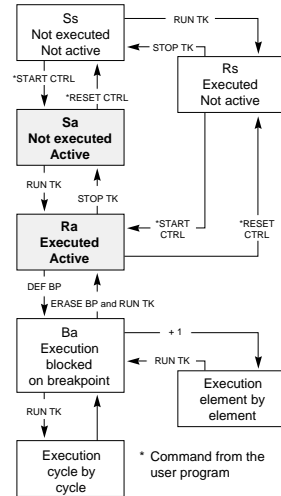
Since the Master task is always active, the first two states (Ss and Rs) concern only the Fast task and the Auxiliary tasks.

When the command STOP or RESET CTRL occurs, the task is stopped at the end of its scan (after the outputs are updated).

Only one breakpoint can be defined, irrespective of the number of tasks.

A task that is blocked on a breakpoint has no effect on the other tasks.

A breakpoint can be defined when the task is in state Ss or Rs, but only becomes effective when the task is in state Ra.



#### Scan cycle of a task

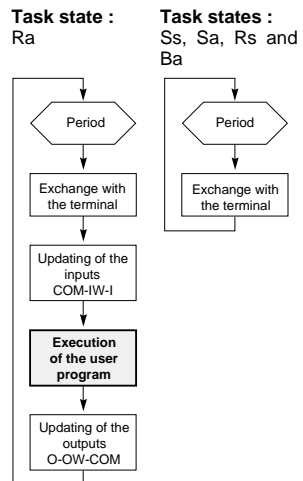
Each task is scanned with the frequency of its period, ie :

- The period defined by the configuration when the task is in states Rs and Ra
- A period of 50 ms defined by the system for states 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 I/O concerns the I/O modules declared in the task. By default, all the I/O are exchanged in the Master task. The common words (COM) are always exchanged in the Master task.

**The I/O are not updated and the program is not executed when the task is not executed, not active, or blocked on a breakpoint (ie in states Ss, Sa, Rs and Ba).**



**Executing the whole program**

Execution of the whole application program, composed of all the tasks configured, can be started and stopped from the mode selection display or in Debug mode, as follows :

- [RUN PG]** Starts execution of the program. The RUN light on the PLC comes on.
- [STOP PG]** Stops execution of the program. The RUN light on the PLC goes out.

The task status zone indicates the status of the program.

**Executing a task**

During the debugging of a application, it is useful to be able to execute a single task independently.

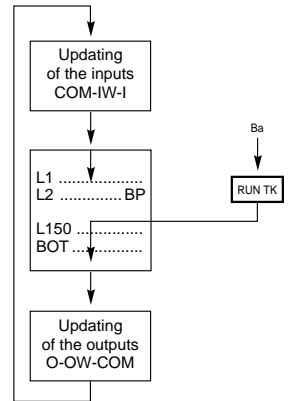
- [TASK]** Permits selection of the task by proposing one soft key for each task configured.
- [RUN TK]** Starts the selected task.
- [STOP TK]** Stops the selected task.

The task status zone indicates the status of the selected task.

**Executing a task cycle by cycle**

After selecting a task, a breakpoint must be defined in order to execute it cycle by cycle (see Section 13.4).

- [RUN TK]** or **<G>** executes one cycle of the task from the element (Ladder network or Literal statement) following the breakpoint, up to and including the element containing the breakpoint.



**Note**

The exchanges with the terminal continue normally and the other tasks that are running continue their cycles.



---

## Executing a module contact by contact (1) or statement by statement (2)

To execute a module (MAIN, PRL, POST or SR) element by element, a breakpoint must be defined in the module and the task containing this module must be selected.

**[DIS LAST]** Displays in real time the last element executed, which in this case is the element containing the breakpoint.

**[+1]** Then replaces the [DIS LAST] key, unless the breakpoint is on the last element of the module. Each time this key is pressed, the next element is executed.

### Notes

- If the element being executed contains an active jump instruction, the jump to the designated label is executed.
- If the element being executed contains an active SR CALL instruction, the subroutine is executed in its entirety.
- If the element being executed is the last element of a MAIN program module :  
**[+1]** causes updating of the outputs, reading of the inputs, and then the execution of the first element of the MAIN program module.
- If the element being executed is the last element in a PRL, POST or SR :  
**[+1]** is no longer proposed.

**[RUN TK]** or **<G>** then completes the cycle of the task and executes the next cycle up to the breakpoint, if it is effectively scanned.

### Remark

**<↓>** This key can be used to display the rest of the task after the breakpoint, before it is executed by pressing [RUN TK].

(1) except PL7-3 GLT

(2) except PL7-3 GLD

### 13.5-2 Influence of the Master Task on the Outputs

For safety reasons, the outputs of a PLC that is not executing its program normally are forced to zero by default. The modules whose outputs are forced to zero include :

- discrete output modules
- analog output modules
- positioning modules
- fast I/O coprocessor module.

This forcing of the outputs to zero can be inhibited by setting the system bit SY8 to 0 by the program or the terminal.

In Debug mode, a PLC is considered to be not executing its program normally when the following conditions occur :

- At least one task is not executed (state Ss : not executed and not active, or state Sa : not executed and active)
- A breakpoint is effective in a task (state Ba : execution blocked on a breakpoint).

#### Note

When system bit SY8 is set to 1 (by default), it resets all the outputs of the PLC configuration to zero if the program is not being executed normally. The complementary system bits SY32 to SY39 (one bit per double rack or per pair of single racks) permit a selective resetting to zero of the outputs. By default, these complementary system bits are set to 1.

When SY8 = 1, only the outputs in the racks whose system bits (SY32 to SY39) are at 1 are forced to zero 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 (general reset), SY32 (reset of rack 0/1), SY33 (reset of rack 2/3) and SY34 (reset of racks 4 and 5), are in the following states :

- SY8 = 1
- SY32 = 1
- SY33 = 0
- SY34 = 0

If the program is not being executed normally, only the outputs of rack 0/1 are forced to zero, the outputs of the other racks retain their states.



---

## 13.6 Accessing the Other Modes

---

### 13.6-1 Accessing the Operator Input Line

In the DISPLAY SCREEN work mode

- [DATA]** Gives access to the operator input line and allows the operator to :
- Read the value of an object (bit or word) in real time
  - Write the value of a word
  - Write or force the state of a bit
  - Activate or block a Grafcet step
  - Search for forced bits or blocked Grafcet steps.

The selected data object is displayed in real time in the operator input line, but this does not cancel the real-time animation of the predefined screen.

The functions then available are the same as those proposed in the Data mode (see Section 12), but only for a single data object - the one that is displayed in the operator input line.

---

### 13.6-2 Accessing the Other Modes

A particularly useful feature of the Debug mode is that it is possible, without leaving the Debug mode, to access the Program mode, the Search/Replace mode, and the Data mode.

#### Accessing the Program mode

- [PROG]** Gives access to the soft key menus of the Program mode so that the elements of the predefined screen can be displayed or modified as required.

#### Accessing the Search/Replace mode

- <S>** Gives access to all the functions of the Search/Replace mode (see Section 16), including :
- Searching for an object
  - Replacing an object by another compatible object
  - Searching for program elements with which a diagnostic code is associated.

---

### Accessing the Data mode

**[DATA SCR]** Gives access to the Data mode (see Section 12).

### Returning to the Debug mode

**[CLEAR]** Returns to the Debug mode, either :

- On the predefined screen that was displayed when the work mode was changed, in the case of accessing the Data or Search/Replace modes
- Or to the screen that was displayed in the Program mode.

### Remark

It is also possible to modify certain parameters of the configuration. For further details refer to Section 6.10.

---

## 13.7 Searching for the Program Addresses of a Variable

---

When a predefined screen is displayed, pressing <Alt><X> gives access to :

- 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 the selected variable
- The selection of a program address and the display of the corresponding program element in a predefined screen.

The variable concerned can be a standard PL7-3 variable (Wi, Bi, etc) or an OFB variable. For further information, refer to Divider H, Section 18.5.





<b>Section</b>	<b>Page</b>
<b>14 Transfer Mode</b>	
Contents	14/1
14.1 Presentation of the Mode	
14.2 Processor - Terminal Memory Transfers	
14.3 Save/Restore : Disk - Terminal or Processor Memory Transfers	
<b>15 Document / Print Mode</b>	
Contents	15/1
15.1 Presentation of the Mode	
15.2 Composition of the Dossier	
<b>16 Search / Replace Mode</b>	
Contents	16/1
16.1 Presentation of the Mode	
16.2 Search Function	
16.3 Replace Function	
16.4 Diagnostic Function	
16.5 Function for Moving FIP Equipment	
16.6 Linking the Searches	
16.7 Example of a Listing	
<b>17 Terminal Mode</b>	
Contents	17/1
17.1 Presentation of the Mode	

---

**G**



<b>Sub-section</b>	<b>Page</b>
<b>14.1 Presentation of the Mode</b>	14/2
14.1-1 Schematic Diagram	14/3
<b>14.2 Processor - Terminal Memory Transfers</b>	14/4
14.2-1 Memory Transfers from the PLC to the Terminal (Processor Terminal)	14/5
14.2-2 Memory Transfers from the Terminal to the PLC (Terminal Processor)	14/6
14.2-3 Terminal - PLC Memory Comparison (Terminal Processor)	14/7
14.2-4 Remote Leading of a Program	14/7
<b>14.3 Save/Restore : Disk - Terminal or Processor Memory Transfers</b>	14/8
14.3-1 Updating the xxx.APP File After Saving a xxx.BIN File	14/10
This section ends on page	14/10

---

## 14.1 Presentation of the Mode

---

This mode enables a PL7-3 application program to be transferred between :

- The internal memory of the terminal and the PLC to which it is connected
- The hard disk of the terminal and the PLC to which it is connected
- The hard disk and the internal memory of the terminal.

### Access to the various functions

All Transfer mode functions can be accessed in Local mode (Terminal Memory) or in Connected mode (Processor Memory).

If the terminal was in Local mode it changes automatically to Connected mode during transfer to the PLC and remains in Connected mode for the period of transfer or comparison.

A function is selected by entering its number and pressing <ENTER>.

### 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 of the OFBs
- The configuration parameters of the OFBs.

The constant words of the application (CWi and CDWi) and the internal constants of the OFBs are always transferred with the program.

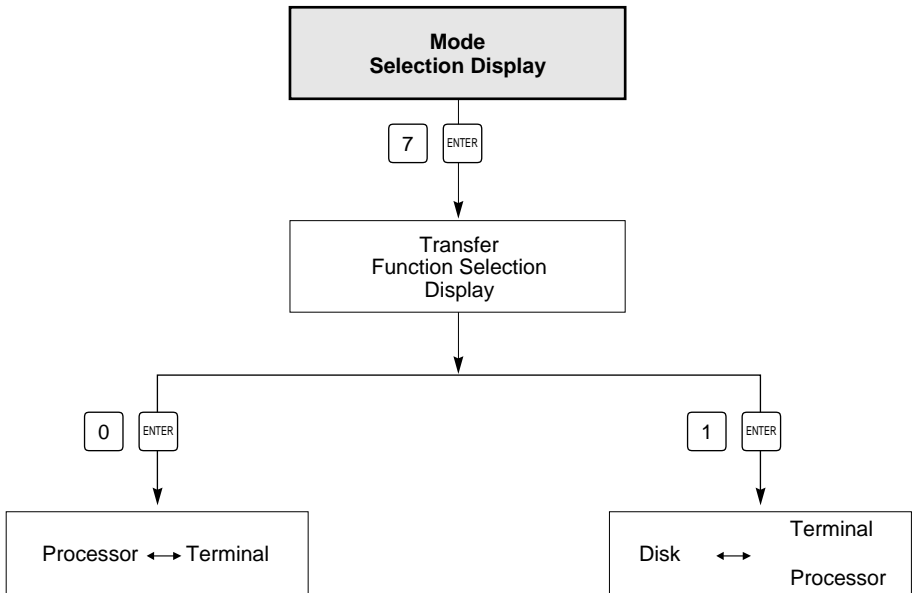
Dynamic objects such as I/O Bi, Xi, SYi bits and internal words Wi and DWi do not have a memory image in the terminal and are therefore not transferred during PLC-Terminal exchanges.

Only the internal words Wi and DWi are stored during PLC-Hard disk exchanges. The user can select an internal word table to be saved by specifying the start address of the table and its length.

### Note

Making a transfer causes no data loss in the source memory, but completely overwrites the destination memory, replacing all previous data with the newly transferred data.

## 14.1-1 Schematic Diagram

**Processor ↔ Terminal transfers**

**<0><ENTER>** Selects the function : Transfer Processor ↔ Terminal, which permits the transfer of the PL7-3 application between the memory of the PLC and internal memory of the terminal. For a V5 application, transfer is made via the hard disk : terminal to processor to hard disk or processor to hard disk to terminal and uses the X-TEL TRANSFER station, with the PL7\_3.BIN file.

**Disk ↔ Terminal/Processor transfers**

**<1><ENTER>** Selects the save/restore function : Transfer Disk ↔ Terminal/Processor which permits the transfer of the PL7-3 application between :

- The hard disk and the internal memory of the terminal (in Local mode)
- The hard disk and the PLC (in Connected mode).



---

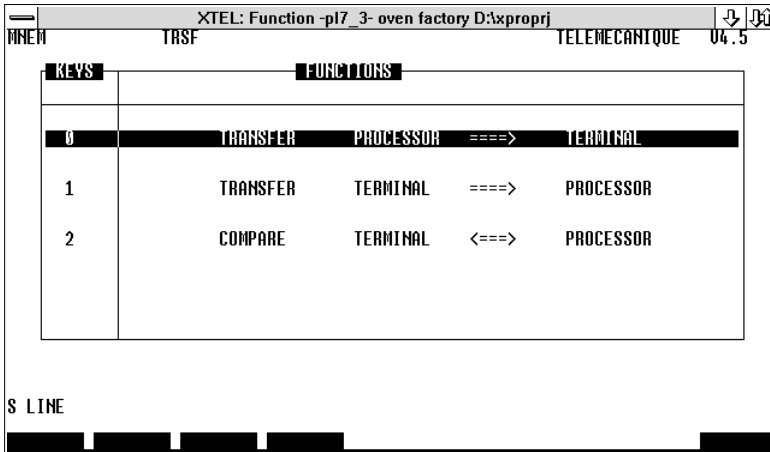
## 14.2 Processor - Terminal Memory Transfers

---

This function, accessible on the main Transfer mode screen, permits :

- The transfer of the PLC application to the internal memory of the terminal
- The transfer of the application in the internal memory of the terminal to a PLC
- The comparison of the application in the internal memory of the terminal with that in the PLC.

If the terminal is in Local mode, the connection with the PLC is automatic and lasts for the duration of the transfer.

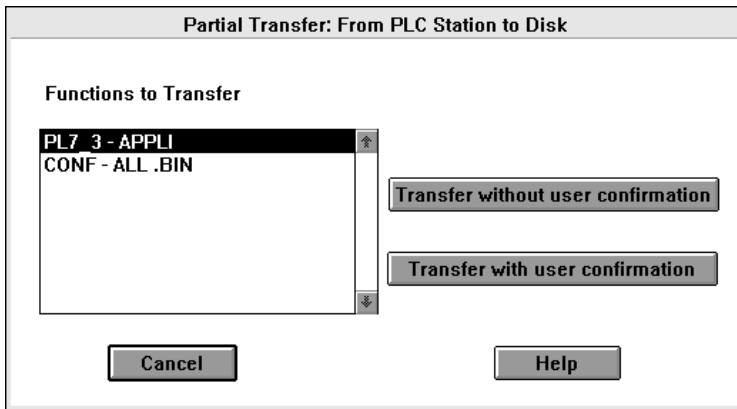


A Processor ↔ Terminal transfer, via the programming port of the PLC, can last several minutes. During transfer, messages indicating the following are displayed :

- transfer in progress
- the physical address of the zone being transferred and the progression of the transfer.

### 14.2-1 Memory Transfers from the PLC to the Terminal (Processor → Terminal)

- <0><ENTER>** Selects this transfer, which is executed after confirmation by [YES].
- <0><ENTER>** Starts 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 terminal is in local mode, it changes automatically to connected mode.

If the terminal is in RUN when the transfer is started, it will remain in RUN during the transfer and after.

If a fault occurs during the transfer, an error message appears on the screen. The <QUIT> key then allows the current transfer to be cancelled. A new transfer is then required.

#### Important

**In V4, this transfer causes the program and the constants previously contained in the terminal's local memory to be deleted. It is therefore essential, if the user wishes to retain the old application, to save it to disk before making the transfer.**

**In V5, this transfer causes the PL7\_3.BIN file to be overwritten. A confirmation request proposed, in order to read the file to local memory.**

#### Note

If the application comprises modules programmed in a language which is not supported, PL7-3 GLD and PL7-3 GLT software only authorize the Control, Adjust and Transfer modes.

---

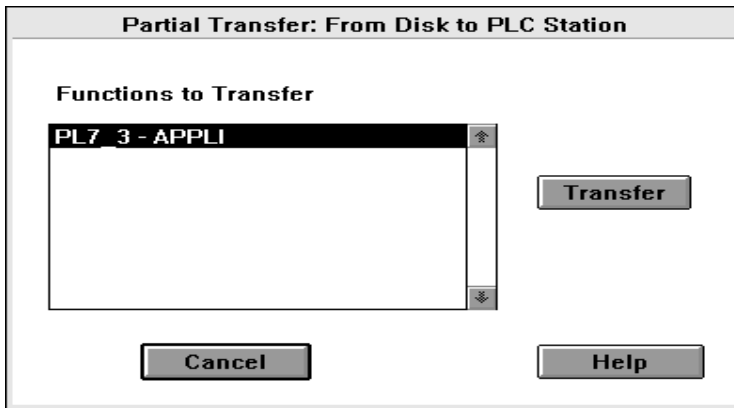
## 14.2-2 Memory Transfers from the Terminal to the PLC (Terminal → Processor)

Before making the transfer, check that :

- The number and size of the RAM cartridges in the PLC are sufficient
- The version of the application and the PLC are compatible.

<1><ENTER> selects this transfer, which will be executed after confirmation by [YES].

<1><ENTER> Saves the application contained in the local memory of the terminal to disk after confirmation by [YES]. This operation must be performed before the transfer is made. The XTEL-TRANSFER tool is then started in order to transfer the PL7-3 function to the PLC (refer to the software workshop documentation).  
If a file already exists on disk, replacing it must be selected in order to start XTEL-TRANSFER.



If the terminal was in Local mode, it switches automatically to Connected mode.

If the PLC is already in RUN, it changes automatically to STOP during the transfer and remains in STOP after the transfer is complete.

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.

If a fault occurs during the transfer, an error message appears on the screen. The <QUIT> key then allows the current transfer to be cancelled. A new transfer is then required.

---

**Important**

- **This transfer causes the program and the constants previously contained in the terminal's local memory to be deleted.**
- **Only V4 applications can be transferred to a V4 processor. In addition, for this transfer to be authorized, the PLC must already contain a description of its dedicated memory zones. This description is entered by using the XTEL-MEM and XTEL-TRANSFER tools. A PMX V4 application cannot be transferred to a TSX V4 PLC.**
- **Only V5 applications can be transferred to a V5 processor. In addition, for this transfer to be authorized, the PLC must already contain a description of its dedicated memory zones. This description is entered by using the XTEL-CONF and XTEL-TRANSFER tools. A PMX V5 application cannot be transferred to a TSX V5 PLC.**

---

**14.2-3 Terminal - PLC Memory Comparison (Terminal to Processor)**

**<2><ENTER>** Selects the comparison, which is executed after confirmation by [YES].

This function checks that the contents of the internal memory of the terminal and the contents of the PLC memory are identical.

The terminal automatically switches to Connected mode if it was in Local mode. At the end of the comparison, the result is displayed (PROCESSOR MEMORY = TERMINAL MEMORY or PROCESSOR MEMORY<> TERMINAL MEMORY)

---

**14.2-4 Remote Loading of a Program (1)**

This function downloads a program to a PLC other than the one to which the terminal is physically connected. Consequently, the PLC concerned must be connected to a TELWAY, MAPWAY, ETHWAY or FIPWAY network.

To perform the download, simply specify the number of the station to which the terminal is logically connected on the Mode Selection display (in the Network Address box), before entering the Transfer mode.

The transfer procedures are the same as those described above. The connection to the network is automatic. Depending on the load of the network, the transfer time may be considerably prolonged.

(1) PL7-3 only

### 14.3 Save/Restore : Disk - Terminal or Processor Memory Transfers

This function, accessible from the main Transfer mode screen, permits :

- In Connected mode, start execution of the XTEL-TRANSFER tool to :
  - transfer a PLC application to the hard disk of the terminal
  - transfer an application from the hard disk of the terminal to the PLC.
- In Local mode :
  - transfer an application from the internal memory of the terminal to the hard disk
  - transfer an application from the hard disk to the internal memory of the terminal.

Information transferred can consist of :

- The program and its constants : xxx.BIN files,
- The data (internal words Wi and DWi) ie an xxx.DAT. file

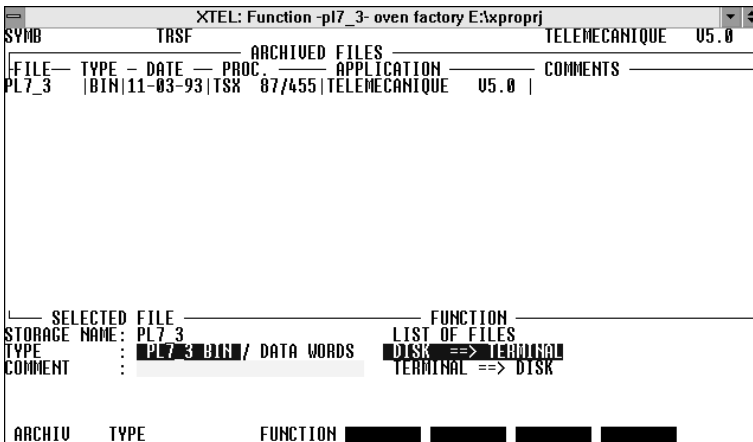
During a data save or restore procedure, the connection is established automatically at the moment of transfer.

Processor ↔ terminal disk transfer via the programming port of the PLC can take several minutes

During a Disk ↔ terminal/Processor transfer, messages indicating the following are displayed :

- transfer in progress
- the physical address of the zone being transferred and the progression of the transfer.

During a Disk → Processor transfer, the word FAULT indicates that the PLC application is temporarily no longer executable. This message disappears at the end of the transfer if it was correctly completed.



This screen is divided into two zones :

- **A zone for displaying the list of archived files, comprising :**

- FILE : the names of the files
- EXT : the type of extension (.BIN for program and constants, or .DAT for data)
- DATE : the date of storage
- PROC. : the type of PLC for which the program was written (TSX or PMX)
- APPLICATION : the name of the application
- COMMENTS : the associated comments.

The files are listed in the order in which they were stored.

This zone is filled in only if the user requests it (by using the List of Files function).

- **A zone for defining the procedure**

The user can define the procedure by using the following soft keys :

**[STR.NAME]** Permits entry of the name of the file to be transferred.

**[DIR]** lists the files stored in the PL7\_3\APPLI subdirectory.

**<↑> <↓>** moves the cursor onto the name of the file to be transferred.

**[SEARCH]** permits entry of the name of the file to be searched for.

**[DELETE]** deletes the selected file (after confirmation).

**[STATION]** assigns the name of the current station as the name of the transferred file.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the transferred file.

**[TYPE]** Permits selection of the type of file to be transferred :

- PL7.3 BIN : selects a xxx.BIN file (program and constants)
- DATA WORDS : selects a xxx.DAT file (internal words WI and DWi).

**[FUNCTION]** Permits selection of the function to be executed :

- LIST OF FILES : displays the list of files
- DISK → TERMINAL : retrieval of a .BIN file
- TERMINAL → DISK : storage of a .BIN file
- DISK → PROC. : retrieval of a .DAT file
- PROC. → DISK : storage of a .DAT file.

**[COMMENT]** Permits entry of a comment (of 22 characters maximum) associated with the file (V2 or V3 applications only). For V4 or V5 applications, comments can be entered in the Program mode by pressing <Alt><K>.

### Remark

The storage and retrieval of an application on a diskette is no longer done by using PL7-3 software, but by using the X-TEL software workshop (see Section 18.5).

### Note

If the application includes modules programmed in a language which is not supported, the PL7-3 GLD and PL7-3 GLT software only authorize the Control, Adjust and Transfer modes.

### 14.3-1 Updating the xxx.APP File After Saving a xxx.BIN File

After saving a xxx.BIN file to disk (TERMINAL → DISK transfer), the updating of the corresponding xxx.APP file (by calling the MEM tool) is proposed.  
The procedure is the same as that described in Divider A, Section 3.8.

```
===== XTEL: Function -pl7_3- oven2 factory3 D:\xproprj
MEM          TASF          ARCHIVED FILES          Odiag Opc102 Onum
FILE-- TYPE - DATE -- PROC. -- APPLICATION -- COMMENTS --
OVEN2  |BIN|10-31-91|TSX 67/410|Odiag Opc102 Onum
PL7_U&L2|BIN|07-08-91|TSX 67/410|Odiag Opc102 Onum

-----
SELECTED FILE          FUNCTION
STORAGE NAME: OVEN2   LIST OF FILES
TYPE          : PL7.3 BIN / DATA WORDS  DISK ==> TERMINAL
COMMENT      :                               TERMINAL ==> DISK

                                WER LAUCH TO UPDATE .APP FILE ?
NO          YES
```

G



<b>Sub-section</b>	<b>Page</b>
<b>15.1 Presentation of the Mode</b>	<b>15/2</b>
15.1-1 General Description	15/2
15.1-2 Schematic Diagram	15/4
<b>15.2 Composition of the Dossier</b>	<b>15/5</b>
15.2-1 General	15/5
15.2-2 Title Page	15/7
15.2-3 General Information	15/9
15.2-4 Wiring the I/O	15/11
15.2-5 Program Comments	15/14
15.2-6 Network Information	15/18
15.2-7 Entering the Footer	15/19
15.2-8 Selecting the Print Functions	15/20
15.2-9 Description of the Listings	15/26

This section ends on page 15/30



---

## 15.1 Presentation of the Mode

---

### 15.1-1 General Description

The Document/Print mode has two main functions :

- **Documentation of an application**, which allows the user to enter all the necessary information concerning an application, in the form of a complete and structured dossier, including :
  - The title page
  - Up to 16 pages of general information
  - The I/O wiring
  - The program comments
  - Up to 16 pages of information concerning the networks
  - The footer at the bottom of each page.
- **Printing of a dossier**, which allows the user to print a complete or partial dossier for this application.

The Document/Print mode can be accessed from the Mode Selection display by selecting 8 - DOCUMENT/PRINT. The terminal can be in Local or Connected mode (PLC running or stopped). The screen below is displayed :

The screenshot shows a terminal window with the title bar 'XTEL: Function -pl7\_3- oven factory D:\xproprj'. The main display area is divided into several sections:

- KEYS** (left column) and **PRINT** (right column) section:
  - 0 SELECTION OF CONTENTS
  - 1 COMPLETE PRINTOUT
  - 2 PARTIAL PRINTOUT
- KEYS** (left column) and **ENTRIES** (right column) section:
  - 3 ENTER TITLE PAGE
  - 4 ENTER INFORMATION PAGES
  - 5 ENTER MNEMONICS
  - 6 ENTER I/O WIRING
  - 7 ENTER PROGRAM COMMENTS
  - 8 ENTER NETWORK INFORMATION
  - 9 ENTER FOOTER
- CURRENT PARAMETERS** section:
  - F4:TYPE OF LISTING :NO
  - F5:PAGE NUMBER :L
  - F6:ARCHIVE NAME :OVEN
- S LINE** section:
  - PAGE
  - LISTING
  - PAGE NB
  - ARCHIV

#### Documentation function

This function is accessed by numeric keys 3 to 8, which permit selection of the various types of entries required.

#### Print function

This function is accessed by numeric keys 0 to 2, which permit a complete or partial dossier to be printed.

For LL type printing (Long Listing) or SL type printing (Short Listing), the following options are proposed :

- [NO]** Printing of the dossier using the parameters defined in the Document/Print and Terminal modes.
- [YES]** Creation and printing of a file that can be used by the X-TEL DOC tool. The 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 that allows the user to replace the old file with a new one [YES], or not [NO], as required.

The general storage parameters are used implicitly to access the following files :

- xxx.TIT (Title page)
- xxx.DES (General information)
- xxx.CA1 to xxx.CA8 (I/O wiring)
- xxx.COM (Program comments)
- xxx.NET (Network information).

But the table of modules and the description of the footer are never read implicitly. In order to produce a dossier containing this information, it is therefore necessary to read these two files by using the functions :

- 6 - ENTER PROGRAM COMMENTS, with the MODULE and READ keys
- 8 - ENTER FOOTER, with the READ key.

For a V5 application, a fixed comment file name is defined for each task/module duo. The table, accessible by the function 6- ENTER PROGRAM COMMENTS, lists all the task/module duos configured and a predefined non-modifiable file name associated with each. The file containing the application program comments no longer exists.

The printing of a dossier can be interrupted by <Ctrl><Z> or by the [ABORT] soft key.

### Remark

- Depending on the selection made in the Terminal mode (FILE or PRT) : creation of a file or printing of the dossier.

### Specific commands

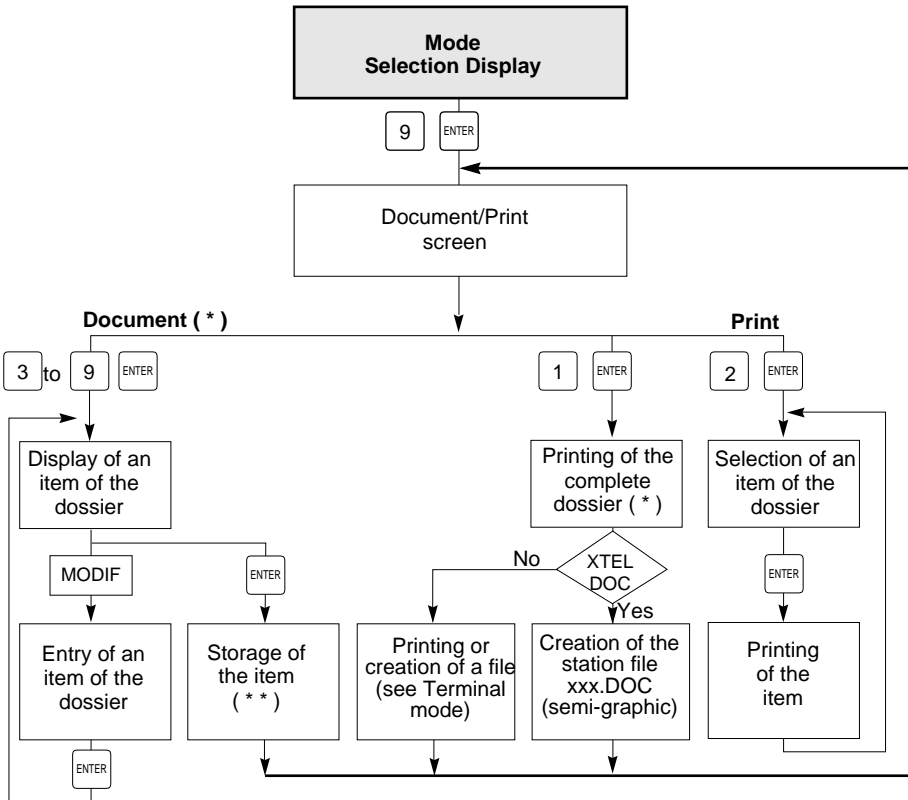
These commands, which are associated with the documentation and printing functions, are described in the CURRENT PARAMETERS zone of the screen.

### Note

When the selection of the contents of the dossier has been defined, it can be stored in a XXX.CDE file by the command [WRITE].

The table of contents of a dossier that has been stored in a xxx.CDE file can be read by the command [READ].

## 15.1-2 Schematic Diagram



(\*) The function 0 - SELECTION OF CONTENTS permits the selection of the items composing the complete dossier.

(\*\*) Refer to the organization of files in Section 2.4.

---

## 15.2 Composition of the Dossier

---

### 15.2-1 General

#### Entry and storage

The information that is entered is stored on the hard disk. Therefore each item that is entered corresponds to a file. The table below shows the correspondence between the various items and the files :

---

3 - ENTER TITLE PAGE	xxx.TIT
4 - ENTER GENERAL INFORMATION	xxx.DES
5 - ENTER I/O WIRING	xxx.CA1 to CA8
6 - ENTER PROGRAM COMMENTS	xxx.COM
7 - ENTER NETWORK INFORMATION	xxx.NET
8 - ENTER FOOTER	xxx.CRT

---

The configurations of the I/O, software, function blocks, OFBs, constant words and program are stored in the xxx.BIN application file. The structure and table of contents of the program, and the cross references, are generated automatically.

In addition to these files are the files reserved for the system, xxx.TMP and xxx.BAK.

#### Printing

The various items and sub-items contained in the complete printed dossier are listed below :

---

Items	Entry (operating mode)
1 Title page	DOCUMENT/PRINT
2 General information	DOCUMENT/PRINT
3 I/O configuration	CONFIGURATION
4 I/O wiring	DOCUMENT/PRINT
6 Configuration of the software	CONFIGURATION
Configuration of the function blocks	CONFIGURATION
Configuration of the OFBs	CONFIGURATION
Constant words	CONSTANT
7 Network information	DOCUMENT/PRINT
8 Program	PROGRAM
Table of contents of the program	Automatic
Structure of the program	Automatic
Grafcet structure	Automatic
9 Cross references of the variable	Automatic
Cross references of the mnemonics	Automatic
Cross references of the program	Automatic
10 Table of Bits Used	Automatic

---

If the application is not documented, the dossier comprises only items 1, 3, 5, 7, 8, and 9.

---

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 dossier.

### Documentation

Before creating or modifying a file it is necessary to :

- Define an archive name. By default this name is the name of the station for a V4 application and the global archive name (PL7\_3) for a V5 application.

**[ARCHIV]** the name (8 characters maximum) under which the files will be stored. It is recommended to use the same name as that of the program file (ie the name declared during storage of the program on the hard disk).

**[DIR]** gives the list of all the previously stored files.

**[SEARCH]** allows entry of the name of the file being searched for.

**[DELETE]** deletes selected file after confirmation.

**[STATION]** assigns the name of the station as the archive name.

**[ARCHIV]** gives global archive name (PL7\_3) as the archive name.

- Enter the revision index number (REV) in the item "ENTER FOOTER". The index number appears in the footer when the dossier is printed.

The documentation is entered item by item. The items 6 - ENTER PROGRAM COMMENTS (at MODULE level), and 8 - ENTER FOOTER can be saved to hard disk by the command [WRITE]. All the other items are automatically saved each time they are validated.

### Caution

Pressing the <CLEAR> key before returning to the Document/Print selection screen causes the loss of all the information that has been modified.

### Recovering from errors

Although storage of most of the documentation items is automatic, it is useful to understand the storage principle in order to recover from a failure situation such as a power break.

During the entry or modification of an item, a temporary xxx.BAK file is created by the system. This file is erased when the item is validated.

If an error or failure occurs during entry, the current file may be erased. However it is possible to recover the information that was entered before the last modifications were made, by immediately copying the xxx.BAK file (without opening a new item) in place of the file that was being entered when the incident occurred.

### Recommendations

When entering documentation information, make frequent saves to the hard disk. When entry of the documentation dossier is completed, copy it immediately to diskette by using the X-TEL Export function.

## 15.2-2 Title Page

The title page permits entry of the name, origin and revision history of the dossier.

Access to the title page file (xxx.TIT) is automatic and uses the global storage parameters.

```

XTEL: Function -pl7_3_ oven3 oven3 D:\xproj
-----
Title      : 0UEH 3
-----
Designer   : _____company_____  department_____  manager_____
User       : TELEMECANIQUE      DIV.4      T.B.
Maintenance :
-----
REV.  DATE      CREATION  REVISION  DESIGNER  EXECUTED by
 01   210391
-----
A LINE
MODIFY
  
```

Access to any of the predefined zones on the screen can be obtained by using the <Tab> or <Shift><Tab> keys.

The cursor can then be positioned within the predefined zone by using the cursor control keys.

**[MODIFY]** Enables entry or modification of the title page information.

**[TITLE]** Title (63 characters maximum).

**[COMPANY]** Names of the companies (16 characters maximum) involved in designing, using and maintaining the application.

**[DEPT.]** Names of the company departments (16 characters maximum) involved in designing, using and maintaining the application.

**[MANAG.]** Names of the company managers (16 characters maximum) involved in designing, using and maintaining the application.

**[REV]** Revision index number (3 characters maximum).

---

**[DATE]** Date of creation or modification of the dossier (8 characters maximum).

**[REVIS.]** Comment (32 characters maximum) explaining the nature of the revision.

**[DESIGN]** Name of the designer (12 characters maximum).

**[EXECUTED]** Operator's name (12 characters maximum).

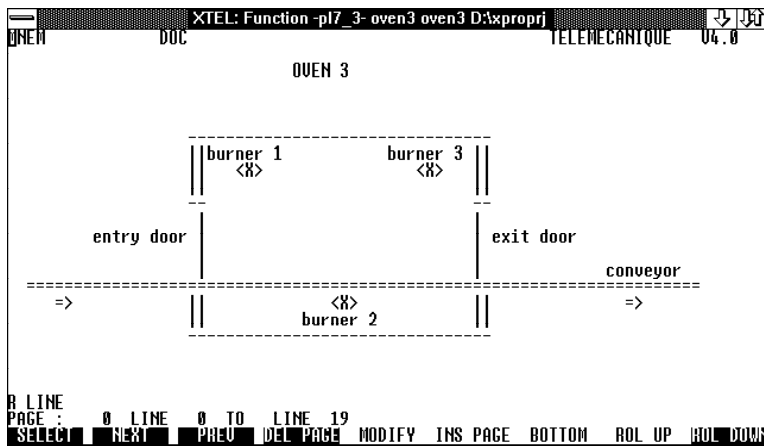
### 15.2-3 General Information

A full screen text editor enables entry of general information concerning the application, which should include :

- A general description of the global operation of the application
- Any specific procedures required for its operation and maintenance
- A description of the program
- A list of documents pertaining to the application.

The text editor permits entry of up to 16 pages of free text (pages 0 to 15). Each page, comprising 56 lines of 80 columns, is displayed on 4 screens of 20 lines.

Access to the general information file (xxx.DES) is automatic and uses the global storage parameters.



The parameter display line at the bottom of the screen indicates the current page, column and line numbers when text is being entered or modified.

#### Reading the text

- [SELECT]** Permits direct access to a page by entering its number (0 to 15).
- [NEXT]** Accesses the next page. After page 15, page 0 is displayed.
- [PREV]** Accesses the previous page. After page 0, page 15 is displayed.
- [DEL PAGE]** Deletes the current page, after confirmation.
- [INS PAGE]** Inserts a blank page upstream of the current page.



- 
- [BOTTOM]** Displays the last screen of the current page (lines 36 to 55).
- [TOP]** Displays the first screen of the current page (lines 0 to 19).
- [ROL UP]** Scrolls up screen by screen onto the previous page.
- [ROL DOWN]** Scrolls down screen by screen onto the next page.
- [MODIFY]** Permits access to the page displayed to allow entry or modification. Moving to another page is blocked until the current page is validated.

### Entering or modifying the text

- [BOTTOM]** Accesses the last line of text on the page.
- [TOP]** Accesses the first line of text on the page.
- [INS LINE]** Inserts a line above the current one.
- [DEL LINE]** Deletes the current line and stores it in a buffer memory.
- [UNDEL]** Restores the line previously deleted by [DEL LINE] on the line indicated by the cursor.
- [CLEAN]** Deletes the contents of the current page, after confirmation.

### Moving or copying a block of text

- [SET]** Permits selection of a block of text (20 lines maximum) by using the cursor control keys. The lines concerned are displayed in reverse video.
- [RESET]** Cancels the selection.
- [CUT]** Deletes and saves the selected block of text, so that it can be moved elsewhere on the same page or another page.
- [PUT]** Restores the saved block of text upstream of the current line.

### Remark

Moving or copying a block of text can be compared to cutting and pasting a sheet of paper :

- Selection of the part of the sheet to be cut ([SET])
- Cutting ([CUT])
- Pasting ([PUT]).

The selected block of text remains in memory (even after it is restored) until [SET] is used again.

### 15.2-4 Wiring the I/O

This function enables a printout of the I/O configuration and the wiring diagrams of the I/O modules and the terminals blocks.

The racks are always represented as double racks.

If racks 1 and 3 are virtual racks, they are left empty.

After selecting function 6 - ENTER I/O WIRING, the number of the double rack to be documented must be entered. The type of processor, previously selected on the Mode Selection display, appears as a reminder at the bottom of the screen.

#### Entering the I/O configuration

The screen displays the configuration of the selected rack and the type of module indicated by the cursor.

XTEL: Function -p17_3- oven3 oven3 D:\xproj								
MEM		DOC		***** I/O WIRING *****			TELEMECANIQUE 04.0	
Rack1	-	-	-	-	-	-	-	-
Rack0	632	9	56	52	-	-	-	-
Module	0	1	2	3	4	5	6	7
PC = TSX 67/410								
CONF		INI 0		MODIFY			CLR CAB HELP	

- [CONF]** Automatically restores the application configuration, if it is stored in the working memory.
- [INI O]** Initializes all configuration codes to 0 (modules absent) on the displayed double rack.
- [MODIFY]** Enables modification of the code number of the module indicated by the cursor.
- [REST COD]** Initializes the module indicated by the cursor, with the code stored by <Enter> in the Help menu.
- [CLR CAB]** Clears all the wiring of I/O modules in the displayed double rack, after confirmation.
- [HELP]** Accesses the list of I/O modules and intelligent modules available.
- <DEL>** Deletes the module indicated by the cursor (but not the wiring).

---

## Entering the I/O wiring

**<ZOOM>** accesses the wiring of the module indicated by the cursor. Each module has one page of 56 lines for entering the wiring diagram for the module.

```
-----
XTEL: Function -pl7_3_oven3 oven3 D:\xproj
-----
Rack 0      Module: DET 16 12      Code : 56
Mod 2       16 Entrees regroup. 24 UCC
Task: MAST  16 Inputs regroup. 24 UCC
-----
wire        block   I/O Bit   mnemonic   comment
-----
          BLK1     | DET 16 12 |
          1       |-----|
          2       | Ix,0 - |   S_Cyc..   Start cycle.....
          3       | Ix,1 - |   Stop....  Stop oven request.....
          4       | Ix,2 - |   Mat       Material present in oven
          5       |-----|
          6       | Ix,3 - |
          7       |-----|
          8       | Ix,4 - |   Req_mat   Request for material
          9       | Ix,5 - |   En_clsd   Entry door closed
         10       | Ix,6 - |   En_open   Entry door open
         11       |-----|
-----
          LINE  0 TO LINE 19
          MODIFY          BOTTOM  ROL UP  ROL DOWNS
```

One screen displays 20 lines divided into 4 zones :

- **Wire** For showing the wiring of the terminal block to the sensor
- **Block** For showing the layout of the terminal block
- **I/O bit** For showing the layout of the module with its I/O bits
- **Mnemonic and comment**  
For showing the mnemonics and comments, associated with I/O bits.

## Reading an existing page

- [BOTTOM]** Displays the last screen of the current page (lines 36 to 55)
- [TOP]** Displays the first screen of the current page (lines 0 to 19)
- [ROL UP]** Scrolls the page up from bottom to top, screen by screen
- [ROL DOWN]** Scrolls the page down from top to bottom, screen by screen
- [MODIFY]** Accesses entry or modification of the wiring.

---

### Entering or modifying a page

- [SEL.BLK]** Enables selection of the type of terminal block. After validation, a drawing of the block appears in the terminal block zone.
- [INI MOD]** Shows the drawing of the module.
- [INI SYMB]** Shows the mnemonic and comment zones, with the mnemonics and comments associated with the I/O bits of the module.
- [DRW LOW]**
- [DRW HIGH]** Enables vertical lines to be drawn upwards or downwards respectively from the position of the cursor.
- [IMPORT]** Copies the page showing the module wiring to the page displayed on the screen. This copy is only possible within the same double rack.  
**Example :** [IMPORT], <0>, <3>, <ENTER>.  
 This sequence of keystrokes copies the page showing the module wiring in rack 0 slot 3, onto the current page.
- [DEL WIRE]** Deletes the contents of the Wiring zone from the line indicated by the cursor.
- [BOTTOM]** Moves the cursor onto the last line that has been written.
- [TOP]** Moves the cursor onto the first line that has been written.
- [././.]** Access the next series of soft keys.
- [INS LINE]** Inserts a blank line above the current line.
- [DEL LINE]** Deletes the current line.
- [CPY LINE]** Copies the current line to the next line.
- [UNDEL]** Restores the line previously deleted by [DEL LINE] on the line indicated by the cursor.

### Note

The association of the keys [DEL LINE] and [UNDEL] allows a line to be moved.

### Moving a text block

The 3 keys [SET], [CUT] and [PUT] have similar functions to those described in Section 15.2-3.

- [SET]** Selects a block of text.
- [RESET]** Cancels the selection.
- [CUT]** Deletes and saves the selected block.
- [PUT]** Restores the stored block.

---

### 15.2-5 Program comments

This function enables a comment of 10 lines, 80 characters maximum to be associated with a program element. This comments appears printed above the associated program element (Ladder network (1), Literal statement (2), program module, etc).

#### V4 application

Allows a choice between access to application program comments and access to module program comments.

Program comments at the application level can be associated with any program Labels. Access to the comments files uses the global parameters (archive name and archive zone).

Program comments at module level can only be associated with Labels of named modules, described in the module table stored in the terminal memory. There is one file per module. The archive zone is the MOD zone and the files have the same names as the modules.

Program comments at module level are used to extract an application module, and to insert one into the module database. It can also be part of a module extracted from a module database.

A name can also be associated with a task, purely to access the comments at task level.

#### Note

If a comment is defined at several levels for the same Label, priority is given to :

- Program comments at module level, if the module is named and if the file exists in the MOD zone
- Program comments at task level, if the task is named and if the file exists in the MOD zone
- Program comments at application level, if the global archive name has been defined and if the file exists in the global archive zone (APPLI).

**[APPLI]**      Accesses the comments at application level.

**[MODULE]**    Accesses the comments at module level.

(1) Except PL7-3 GLT

(2) Except PL7-3 GLD

## V5 application

The table of modules no longer exists, as the module names are defined at XTEL-MOD level and cannot be modified under PL7-3. For each task/module duo, a fixed comment file name is defined. This file is stored in the MOD zone as soon as it contains a comment. The table created lists all the task/module duos configured. This table cannot be modified by the user and the task/module duos are each associated with a predefined comment file name which cannot be modified. The file containing the program comments at application level no longer exists.

TASK	MODULE	FILE	MACRO MOD.	BAS. MOD
MAST	CHAR1	MCHAR1	MOTOR2	MOTOR
MAST	POST	MPOST		
MAST	SM0	MSR0		
MAST	SM1	MSR1		
MAST	SM2	MSR2		
MAST	SM3	MSR3		
MAST	SM4	MSR4		
MAST	SM5	MSR5		
MAST	SM6	MSR6		
MAST	SM7	MSR7		
MAST	SR0	MSR0		
MAST	SR1	MSR1		
MAST	SR2	MSR2		
MAST	SR3	MSR3		
MAST	SR4	MSR4		

PREV SCR NEXT SCR RETURN

**[PREV SCR]** Displays the previous screen.

**[NEXT SCR]** Displays the next screen.

**[MOD INFO]** Lists, for information, the elementary modules and macro-modules contained in the selected task/module duo.

**[RETURN]** Replaces the [MOD INFO] key when it is active. Returns to the task/module duo selection screen.

**<↑><↓>** Move the cursor in the table.

**<ENTER>** Gives access for entry or modification of the comment associated with the task/module duo.

XTEL: Function -pl7_3- oven3 oven3 D:\xproj		TELEMECANIQUE	U4.0
ENTER PROGRAM COMMENT	FILE	: OUVEN3	
CHECK TEMPERATURE BURNER 1			
Error occurs if temperature is inferior to lower setpoint or superior to upper setpoint.			
Error monitoring by event diagnostic OFB: EUDGS0			
Comment Number : 0	MAST	PRL	L100
[SELECT]	[NEXT]	[PREV]	[SEARCH] [MODIFY] [ADDRESS] [G7 ADDR] [PUT COM] [GET COM]

Correspondence between the comment entered in the Document/Print mode and the program element is established by the program address, defined by the [ADDRESS] and [G7-ADDR] soft keys.

### Reading the comments

- [SELECT]** Permits direct access to a comment by entering its number (0 to 499).
- [NEXT] or <↓>** Gives access to the next comment; after comment 499, comment 0 is displayed.
- [PREV] or <↑>** Gives access to the previous comment; after comment 0, comment 499 is displayed.
- [SEARCH]** Permits access to a comment by entering its program address.
- [ADDRESS]** Permits entry of the address of the program element associated with the current comment (*task, module, label*).
- [G7-ADDR]** Permits entry of the address of the program element (action or transition condition) associated with the current comment (*task, module, step reference or transition*).
- [PUT COM]** Saves the current comment so that it can be copied elsewhere.
- [GET COM]** Restores the comment previously saved by [PUT COM] and permits its modification.
- <DEL>** Deletes the current comment and its associated address (after confirmation).
- [MODIFY]** Permits entry or modification of the comment.

**Entering or modifying a comment**

- [INS LINE]** Inserts a blank line above the current line.
- [CPY LINE]** Copies the current line onto the next one.
- [CLEAN]** Deletes the displayed comment, after confirmation.
- [BOTTOM]** Positions the cursor on the last line of the comments zone.
- [TOP]** Positions the cursor on the first line of the comments zone.

**Moving a line**

- [DEL LINE]** Deletes and saves the current line.
- [UNDEL]** Restores the line previously deleted by [DEL LINE] onto the line indicated by the cursor.



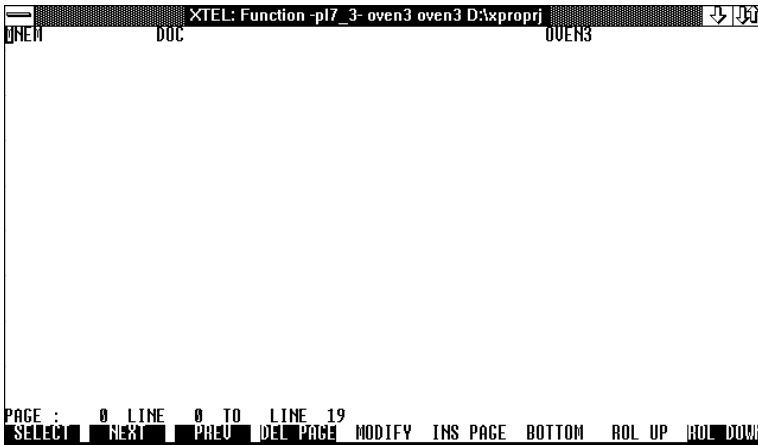
---

### 15.2-6 Network Information

A full screen text editor allows entry of data onto the application's PLC networks, including :

- General description and global operation of the application
- Any special procedures or precautions required for its operation and maintenance
- The description of the networks and their structure
- The names of any related documents.

The text editor permits entry of up to 16 pages of free text (pages 0 to 15). Each page, comprising 56 lines of 80 columns, is displayed on 4 screens of 20 lines.



The commands of this text editor are the same as those of the text editor used to enter the General Information (see Section 15.2-3).

### 15.2-7 Entering the Footer

This function allows the entry of 3 character fields defining the footer that will be printed at the bottom of each page :

- 1 field of 25 characters maximum
- 1 field of 45 characters maximum
- 1 field of 3 characters maximum (rev), revision index number.

This footer is saved at the explicit request of the user. It is neither read nor written automatically.

The screenshot shows a terminal window titled "XTEL: Function -pl7\_3- oven factory D:\xpropri". The main display area is titled "FOOTER ENTRY". Below this title is a form with several input fields. The first field is labeled "application" and contains the text "TSX Telemecanique". The second field is labeled "oven factory". The third field is labeled "rev" and contains the number "1". The fourth field is labeled "date" and the fifth is labeled "page". The "rev" field is highlighted in reverse video. At the bottom of the screen, there are three buttons: "MODIF", "WRITE", and "READ". To the left of these buttons, the text "S LINE" is visible.

<↑><↓>

Select the field to be entered or modified.

[MODIF]

Permits entry or modification of the selected field (shown in reverse video).

[WRITE]

Saves the footer to a xxx.CRT file in the PL7\_3/APPLI subdirectory.

[STR. NAME] permits entry of the name of the archive file.

[DIR] lists existing files.

[ARCHIV] assigns the global archive name (PL7\_3) as the name of the file.

[READ]

Permits a previously saved file to be read.

[STR. NAME] permits entry of the name of the file to be retrieved.

[DIR] lists existing files.

[ARCHIV] assigns the global archive name (PL7\_3) as the name of the file to be retrieved.

---

## 15.2-8 Selecting the Print Functions

The Print function permits printing of :

- The complete application dossier, with options for its contents
- Any item of the dossier.

Before a program can be printed, it must be in the internal memory of the terminal (in Local mode), or in the user memory of the PLC (in Connected mode).

The general storage parameters are used automatically to access the following files :

- Title page
- General information
- I/O wiring
- Program comments at application level
- Network information.

The table of modules and the description of the footer are never read implicitly. In order to produce the dossier containing this information, it is therefore necessary to load the corresponding files into the terminal memory :

- 6 - ENTER PROGRAM COMMENTS, with the MODULE and READ keys
- 8 - ENTER FOOTER, with the READ key.

### Selection of the current printing parameters

- [LISTING]** Successively pressing this key selects the type of printout :
- SL : Short listing (Ladder language and the configuration of the OFBs are more compact),
  - LL : Long listing,
  - NO : No listing. The entire dossier is scrolled through on the screen.
  - /1 : No listing. The dossier is displayed on the screen page by page controlled by the operator.

The choice selected is displayed in the TYPE OF LISTING item in the CURRENT PARAMETERS zone of the main Document/Print mode screen.

- [PAGE NB]** Enables entry of the number of the first page of a partial printout. The number appears in the PAGE NUMBER item in the CURRENT PARAMETERS zone.

- [ARCHIV]** Enables modification of the global archive name of the files to be printed. This is then displayed in the ARCHIVE NAME item in the CURRENT PARAMETERS zone.

### Complementary functions

- [PAGE]** Immediately feeds a new page of paper into the printer.
- <CTRL><Z>** Cancels the ongoing printout.
- [+1] or <Space>** Displays the next page during page by page display.
- [WAIT]** Freezes the display during continuous scrolling.
- [CONTINUE]** Continues scrolling.
- [ABORT]** Cancels the current display mode (continuous scrolling, or page by page).

### Composition and printing of the entire dossier

- <0><ENTER>** Permits selection of the contents of the dossier.

```

XTEL: Function -pl7_3- oven3 oven3 D:\xprojj
-----
ONEM          DOC          SELECT ITEM FOR PRINTOUT          OUEMS
-----
Type of printer      : ASCII
* Title Page
* General Information
* I/O Configuration
* I/O Wiring
* Mnemonics Table
* Software Configuration
* Function Block Configuration
* Configuration of OFBs
* Constant Words ( TOP BOT )
* Network Information

* Program
* OFB listing
* Object listing
* With Mnemonics
* Program Contents
* Program Structure
* Grafcet Structure

* Cross-Refs
* Object
* Address
* Cross-Refs : Variables
* Cross-Refs : Mnemonics
* Cross-Refs : Program
* Table of Bits used
  Bi : TOP BOT
  I/O : 0 2 4 6 8 A C E
        1 3 5 7 9 B D F

NOLIST          WRITE          READ

```

By default, all the items are active.

- [LIST]** Selects the item indicated by the cursor. An asterisk then appears in front of the selected item.
- [NOLIST]** Cancels selection of the item indicated by the cursor.

### Type of printer

- [PRT]** Selects the printer : IBM semi-graphic or ASCII.

---

## Constant words

- [FROM]** Selects the first constant word to be printed.  
**[TO]** Selects the last constant word to be printed.  
**[TOP → BOT]** Selects all the constant words.

## Program

- [ALL]** Prints all the program.  
**[PARTIAL]** Selects part of the program.  
**[OFB]** Prints with or without the graphic form of the OFBs used.  
**[OBJECT]** Prints data objects and mnemonics, with or without comments.  
**[SYMBOLIC]** Prints data objects in mnemonic or variable format.

## Cross references

- [OLD-NEW]** Systematically recalculates the cross reference tables, or prints the current tables.

### Variables

- [ALL]** Cross references of all the variables.  
**[PARTIAL]** Selects the variables.

### Program

- [ALL]** Cross references of all the program.  
**[PARTIAL]** Selects part of the program.

## Table of bits used

### Bi

- [FROM]** Selects the first bit or word to be printed.  
**[TO]** Selects the last bit or word to be printed.  
**[TOP → BOT]** Prints all the bits or words.

### I/O

- [IOLIST]** Selects or excludes the double rack indicated by the cursor.  
**<←> <→>** Move the cursor.

- [WRITE]** Enables the customized table of contents of the dossier to be saved to a xxx.CDE file.

**[STR. NAME]** permits entry of the name of the file to be stored.

**[DIR]** lists existing files.

**[STATION]** assigns the name of the current station as the name of the file.

**[MODULE]** assigns the name of the last module selected as the name of the file.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file.

**[READ]** Allows a previously stored xxx.CDE file to be read.

**[STR. NAME]** permits entry of the name of the file to be retrieved.

**[DIR]** lists existing files.

**[STATION]** assigns the name of the current station as the name of the file to be retrieved.

**[MODULE]** assigns the name of the last module selected as the name of the file to be retrieved.

**[ARCHIV]** assigns the global archive name (PL7\_3) as the name of the file to be retrieved.

**<1><ENTER>** Prints the "complete dossier" composed of the selected items (see Print function in Section 15.1-1).

### Printing of part of the dossier

**<2><ENTER>** Gives access to the partial printing function (Partial Listing).

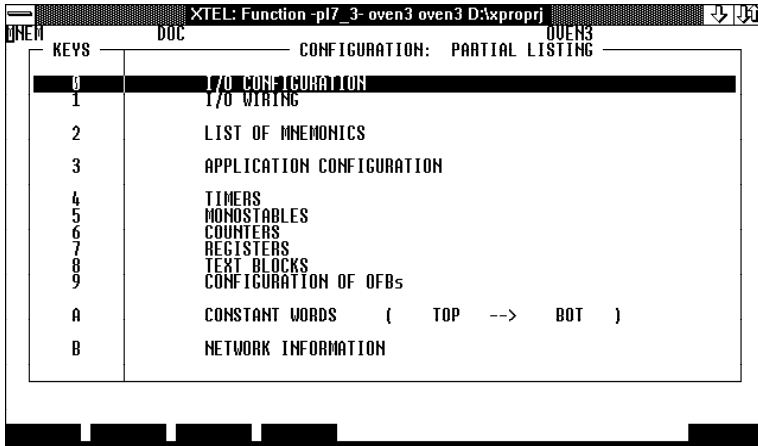
KEYS	PARTIAL PRINTOUT
0	TITLE PAGE
1	INFORMATION PAGES
2	COMPLETE CONFIGURATION
3	PARTIAL CONFIGURATION
4	PROGRAM
5	COMPLETE CROSS-REFS
6	PARTIAL CROSS-REFS
7	TABLE OF BITS USED

Select the item to be printed and press **<ENTER>** to start the printout.

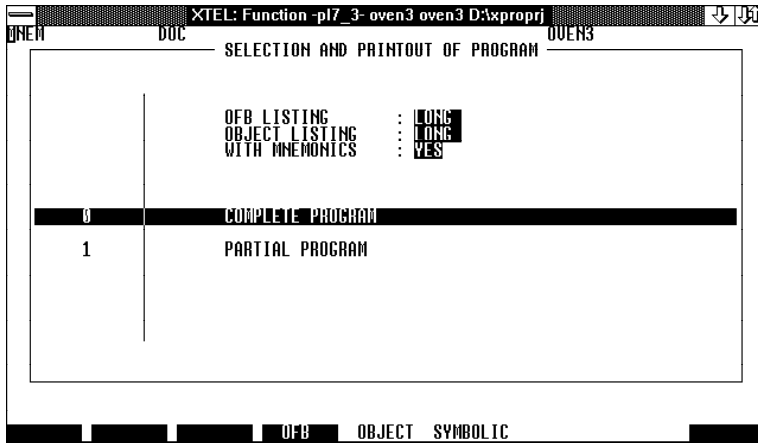
The following items propose a new menu :

- Partial configuration
- Program
- Partial cross references
- Tables of bits used.

<3><ENTER> Displays the Configuration : Partial Listing screen :

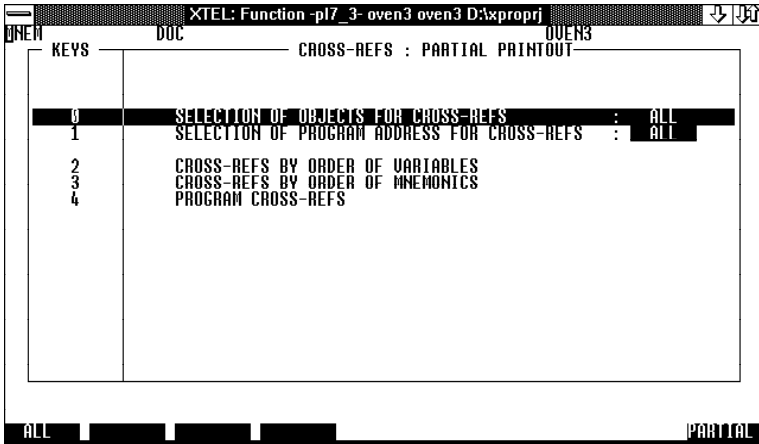


<4><ENTER> Displays the Selection and Printout of Program screen



- [OFB] Printing with or without the graphic form of the OFBs called.
- [OBJECT] Printing of data objects and mnemonics, with or without comments.
- [SYMBOLIC] Printing of data objects in mnemonic or variable format.

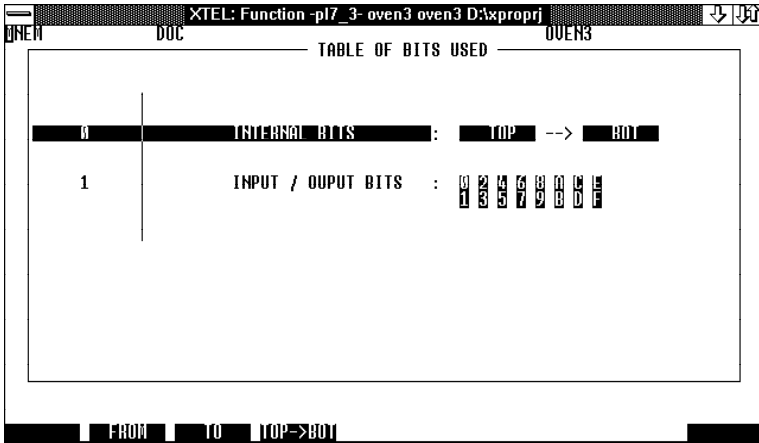
<6><ENTER> displays the Cross Refs : Partial Printout screen



**[ALL]** Cross references all the variables or all the program.

**[PARTIAL]** Selects the variables or part of the program.

<7><ENTER> displays the Table of Bits Used screen :



**[FROM]** Selects the first bit or word to be printed.

**[TO]** Selects the last bit or word to be printed.

**[TOP → BOT]** Prints all the bits or all the words.



---

## 15.2-10 Description of the Listings

Detailed composition of the complete listing of an application :

- 1 - Title page
- 2 - General information
- 3 - I/O configuration
- 4 - I/O wiring :
  - Wiring diagrams of the modules and terminal blocks
- 5 - Software configuration :
  - Configuration of the application
  - Configuration of the function blocks
  - Constant words
- 6 - Network information
- 7 - Program :
  - IT, Fast, Master and Auxiliary tasks
  - Program structure, table of modules and Grafcet structure
  - Table of contents of the program
- 8 - Cross references :
  - By variables
  - By mnemonics
  - By subroutines
- 9 - Table of bits used
  - Internal bits
  - I/O bits
  - Internal words
  - Constant words
- 10 - Table of contents of the dossier

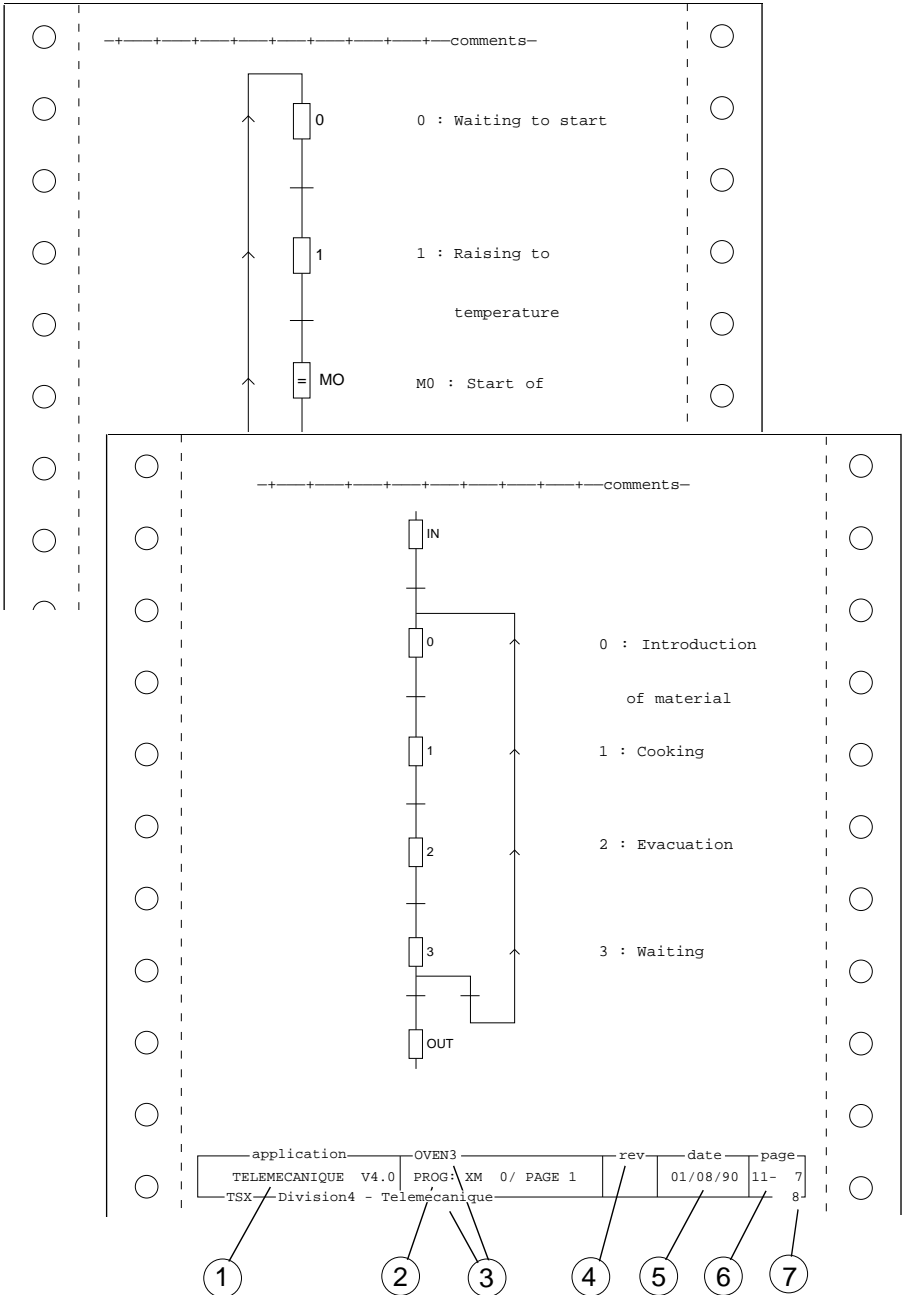
### Description of a listing page

Each A4 listing page comprises :

- The information (Ladder networks, Grafcet or Literal pages , I/O wiring, etc)
- A footer at the bottom of each page giving :

- ① The name of the application entered in Configuration mode
- ② The name of the item being printed
- ③ The personalization of the footer
- ④ The document version number (entered by using the <REV> key)
- ⑤ The date of printing
- ⑥ The relative page number by item
- ⑦ The absolute page number.

Example of listing pages printed in semi-graphic mode

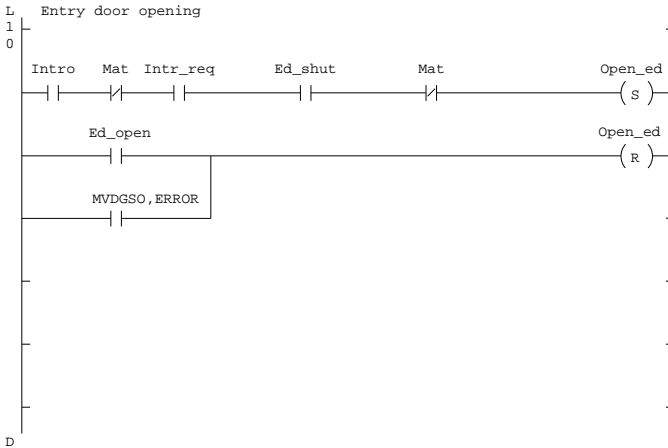


## Example of program elements with comments

### 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  
 MVDGS0.



```
EXEC MVDGS0 ( ; Open_ed ; Ed_open ; NOT ( Ed_open . Ed_shut ) ; NOT Ed_shut ; Nul_ofb => ; ; )
```

### OPTIONAL FUNCTION BLOCK I/O PARAMETERS

	MVDGS0			
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_of	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 gives the use of the variable in the application :

- **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 of a table
- **WT** : Writing of a table
- **RT-X** : Reading of an indexed table
- **WT-X** : Writing of 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	MNEMONIC	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 - Telemecanique				24

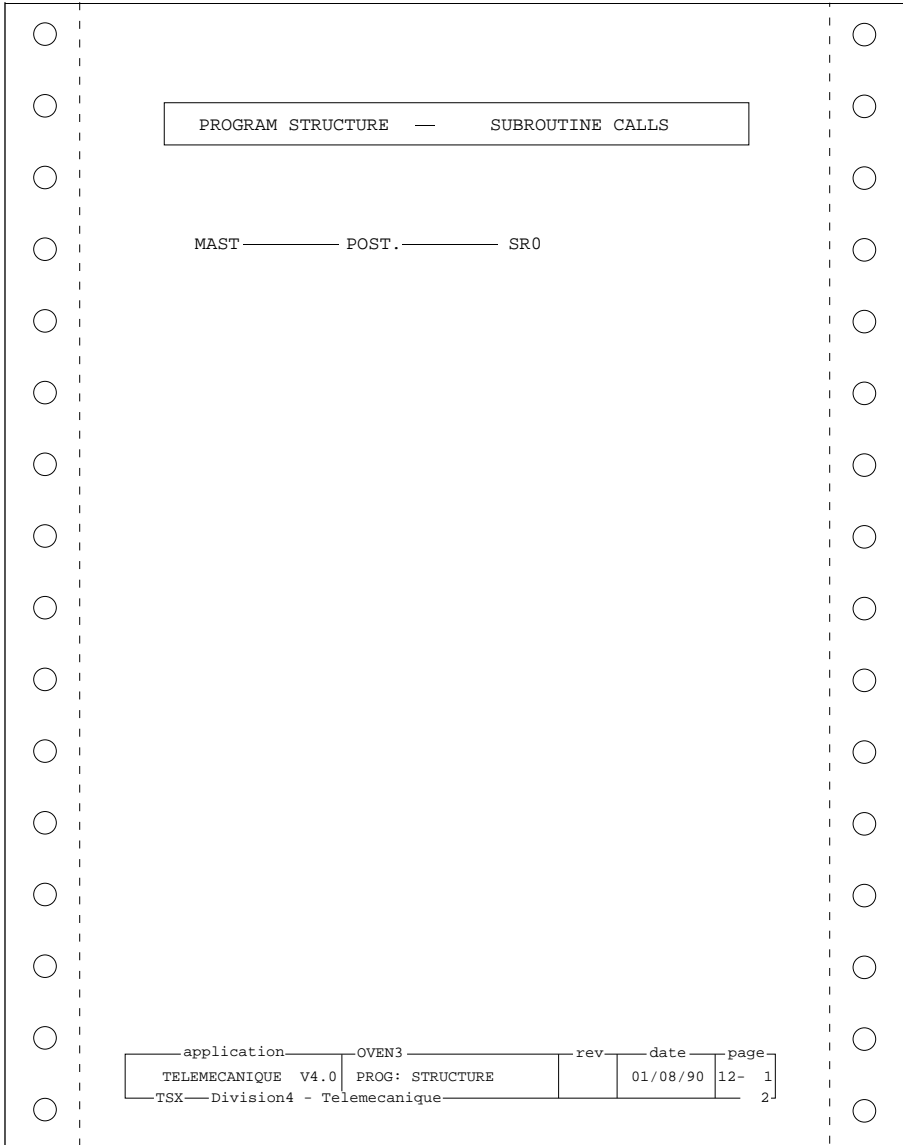


---

## Example of a program structure (file storage tree of the program)

### Note

\*\*\* Indicates that the corresponding program element calls one or more subroutines.





<b>Sub-section</b>	<b>Page</b>
<b>16.1 Presentation of the Mode</b>	16/2
<b>16.2 Search Function</b>	16/4
<b>16.3 Replace Function</b>	16/7
<b>16.4 Diagnostic Function</b>	16/9
<b>16.5 Function for Moving FIP Equipment</b>	16/11
<b>16.6 Linking the Searches</b>	16/12
<b>16.7 Example of a Listing</b>	16/13
This section ends on page	16/14

---

## 16.1 Presentation of the mode

---

The SEARCH/REPLACE mode uses either the variable or the mnemonic symbol mode and has 3 functions :

### Search

This function allows an object to be searched for in :

- all or part of an application program (PLC in RUN or in STOP),
- one or several elementary modules and/or macro-modules (V5 only).

### Replace

This function enables an object to be searched for and replaced by a **compatible** object in :

- all or part of an application program (PLC in RUN or in STOP),
- one or several elementary modules and/or macro-modules (V5 only).

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. The RAM bits do not have an edge,
- An internal word  $W_i$ , modified during operation 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 the  $X_i$  step. This is also true for an  $X_{Mi}$  macro-step.

The Replace function also enables FIP equipment to be moved in :

- all or part of an application program,
- one or several elementary modules and/or macro-modules.

### Diagnostics

This function enables a search for program elements with an associated diagnostic code in :

- all or part of an application program (PLC in RUN or in STOP),
- one or several elementary modules and/or macro-modules (V5 only).

(1) Except PL7-3GLD

(2) Except PL7-3GLT

---

The object searched for or replaced can be :

- A bit
- A word or double word
- A function block (standard or optional)
- A standard or optional function block element
- A remote I/O element (for FIPIO equipment)
- A bit extracted from a word, a function block or from remote I/O

Searches can be performed on objects which are read only, write only or read/ write in program modules written in Literal (1) or Ladder (2) language. When replacing, the objects are forced to a read/write criteria.

The search/replace function is not authorized for bits and words which are 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.

The SEARCH/REPLACE mode is accessed from :

- the MODE SELECTION display, in local or connected mode,
- DEBUG mode, in connected mode, by pressing <S>.



## 16.2 Search function

The Search/Replace screen is divided into three zones displaying the options selected by default or by the operator :

- ① Display of the part of the application program or list of elementary modules and /or macro-modules concerned by the search, replacement or diagnostic (by default the entire program or all the elementary modules and macro-modules).
- ② Display of the function required :
  - Search or replace, with display of the object concerned
  - Diagnostics.
- ③ Display of :
  - The use criteria (read or write) of the object to be searched for or replaced
  - The operating mode : manual or automatic, with or without listing.

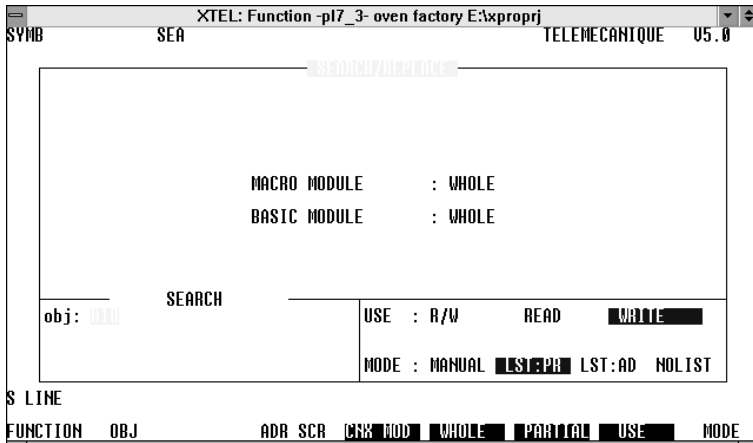
TASK	MODULE	FROM	TO
FAST	CHART	TOP	TOP
MAST	XM	P	P
	PR	BOT	BOT
	POST		
AUX0	MAIN	TOP	TOP
AUX1	SR	L	L
AUX2		BOT	BOT
AUX3			

obj: [REDACTED]      USE : R/W    READ    [REDACTED]

MODE : [REDACTED]    LST:PR    LST:AD    NOLIST

FUNCTION    OBJ      MOD SCR    [REDACTED]    WHOLE    PARTIAL    USE    MODE

**[MOD SCR]** Accesses the screen, to search for an object in the elementary modules and/or macro-modules.



**[ADR SCR]** Accesses the screen to search for an object in the application program.

**[FUNCTION]** Successively pressing this key selects the required function :

- SEARCH,
- REPLACE,
- or DIAGNOSTICS.

**[OBJ]** Permits entry of the object to be searched for.

**[CNX MOD]** Accesses the move function for FIP equipment, in the application program or in the elementary modules and/or macro-modules.

**[WHOLE]** Permits a search for the object in the entire application program or in all the elementary module and macro-modules.

**[PARTIAL]** When searching in the application program, this key provides access to the PROGRAM ADDRESS SELECTION screen, so as to select the part of the program (task, module, program element) on which to perform the search.

When searching in the elementary modules and/or macro-modules, this key provides access to the MODULE SELECTION screen, so as to select the elementary module or the macro-module on which to perform the search. A search on a macro-module actually means performing a search on all the elementary modules which make up the macro-module.



- 
- [USE]** Successively pressing this key selects the use of the object concerned :
- REA/WRI all objects,
  - READ read only objects,
  - WRITE write only objects.
- [MODE]** Successively pressing this key selects the operating mode of the object concerned :
- MANUAL selects manual mode. The terminal waits for a command from the user to continue.
  - LST:PR selects automatic mode with printout of a complete listing giving the labels and program elements containing the object searched for.
  - LST:AD selects automatic mode with printout of a partial listing giving only the labels of the elements containing the object searched for.
  - NOLIST selects automatic mode without printout of a listing.
- [ABORT]** Cancels the operation in progress.

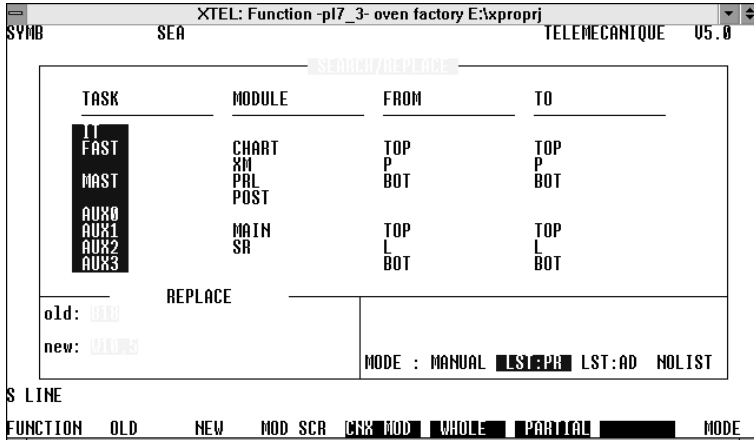
### 16.3 Replace Function

Replacing an object by another object can be performed either in local or connected mode.

When connected to a V2 or V3 PLC, replacement is possible only when the PLC is stopped. When connected to a V4 or V5 PLC, replacement is possible when the PLC is running or stopped.

An object can only be replaced by another **compatible** object.

For example, B1 can be replaced by B50 or by W5,3.



**[FUNCTION]** Successively pressing this key selects the function required : search, replace or diagnostic.

**[OLD]** Permits entry of the object to be replaced.

**[NEW]** Permits entry of the new object.

**[MOD SCR]** Accesses the screen which enables an object in the elementary modules and /or macro-modules to be replaced.

**[ADR SCR]** Accesses the screen which enables replacement of an object in the application program.

**[CNX MOD]** Accesses the function for moving FIP equipment, in the application program or in the elementary modules and /or the macro-modules.

**[WHOLE]** Permits search and replacement throughout the entire application program or in all the elementary modules and macro-modules.

---

**[PARTIAL]** When searching in the application program, this key provides access to the PROGRAM ADDRESS SELECTION screen, so as to select the part of the program (task, module, program element) on which to perform the Search/Replace.

When searching in the elementary modules and/or macro-modules, this key provides access to the MODULE SELECTION screen, so as to select the elementary module or the macro-module on which to perform the search. A search on a macro-module actually means performing a Search/Replace on all the elementary modules which make up the macro-module.

**[MODE]** Successively pressing this key selects the replace operating mode :  
MANUAL selects manual mode.

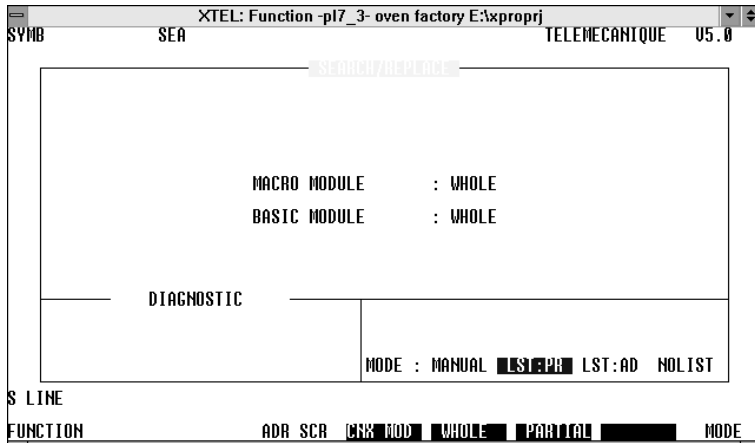
LST:PR selects automatic mode with printout of a complete listing giving the labels and the elements containing the object to be replaced.

LST:AD selects automatic mode with printout of a partial listing giving only the labels of the elements containing the object to be replaced.

NOLIST selects automatic mode without printout of a listing.

**[ABORT]** Cancels the operation in progress.

## 16.4 Diagnostic function



- [FUNCTION]** Successively pressing this key selects the function required : search, replace or diagnostic.
- [MOD SCR]** Accesses the screen which enables diagnostic search for an object in the elementary modules and/or macro-modules.
- [ADR SCR]** Accesses the screen which enables diagnostic search, in the application program.
- [CNX MOD]** Accesses the function for moving FIP equipment, in the application program or in the elementary modules and/or macro-modules.
- [WHOLE]** Permits diagnostic search throughout the entire application program or in all the elementary modules and/or macro-modules.
- [PARTIAL]** When performing a diagnostic search in the application program, this key provides access to the PROGRAM ADDRESS SELECTION screen, so as to select the part of the program (task, module, program element) on which to perform the diagnostic search.  
When performing a diagnostic search in the elementary modules and/or macro-modules, this key provides access to the MODULE SELECTION screen, so as to select the elementary module or the macro-module on which to perform the diagnostic search. A diagnostic search on a macro-module actually means performing a diagnostic search on all the elementary modules which make up the macro-module.

- 
- [MODE]** Successively pressing this key selects the diagnostic search operating mode :
- MANUAL selects manual mode.
  - LST:PR selects automatic mode with printout of a complete listing giving the labels and the elements diagnosed.
  - LST:AD selects automatic mode with printout of a partial listing giving only the labels diagnosed.
  - NOLIST selects automatic mode without printout of a listing (showing the number of elements diagnosed).
- [ABORT]** Cancels the operation in progress.

## 16.5 Function for Moving FIP Equipment

This function, accessed by the command [CNX MOD], enables all the objects of a FIP device to be replaced in the application program or in the elementary modules and/or the 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 :

**n°connection[,n°module].**

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 confirming with <Enter>, the software checks 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 replacements found and performed is not displayed.

### Use of soft keys

#### [MOD SCR], [ADR SCR], [WHOLE], [PARTIAL]

The role of these 4 soft keys is identical to the replace function described in section 16.3.

#### [MODE]

The role of this key is identical to that of the replace function described in section 16.3. Nevertheless, in order to avoid interrupting the current replacement procedure (ABORT), which will make the application incoherent, this key does not allow change to MANUAL mode.

#### [OLD]

Allows the source equipment address to be entered : connection n° [,module n°].

#### [NEW]

Allows the target equipment address to be entered : connection n° [,module n°].

**[VARIABLE]** Returns to the object replace function (see section 16.3).

#### Note

Depending of the size of the program, moving the FIP equipment may take considerable time.



---

## 16.6 Linking the Searches

---

### Automatic search, replace or diagnostic

**[MODE]** Selects one of the automatic modes, which appears in reverse video.

**<ENTER>** Starts the selected function :

- Automatic search, replacement or diagnostic,
- With or without a 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 replaced (in replace mode),
- The number of program elements diagnosed (in diagnostic mode).

### Manual search or diagnostic

**[MODE]** Selects the manual mode, which appears in reverse video.

**<ENTER>** Starts the search for the first object (in search mode) or the first program element (in diagnostic mode).

**[NEXT]** Searches for the next object or program element.

A message indicates the number of times the object or program element has been found.

### Manual replacement

**[MODE]** Selects manual mode, which appears in reverse video.

**<ENTER>** Starts the search for the first object to be replaced. The element which contains it is displayed.

**[YES]** Confirms replacement of the object displayed, by the new object.

**[NO]** Cancels replacement of the object.

**[NEXT]** Searches for the next object.

A message indicates the number of times the object has been replaced.

### 16.7 Example of a Listing

**LST:PR** Enables printout of a listing indicating the labels and the elements containing the object before and after replacement.

```

○      Function           :   REPLACE           ○
      Object(s)          :   W100 : W113
○      Application name   :   TELEMECANIQUE   V4.0   ○
      Date                :   01/08/90
○
      *****
      Literal statement:
○      !                 IF [Temp_b1<Cons_b1]       ○
                        THEN SET Ma_b1
○
      New Literal statement :
○      !                 IF [Delta<Cons_b1]         ○
                        THEN SET Ma_b1
  
```

**LST:AD** Enables printout of a listing indicating the labels of the elements containing the object replaced.

```

○      Function           :   REPLACE           ○
      Object(s)          :   W113 : W100
○      Application name   :   TELEMECANIQUE       V4.0   ○
      Date                :   01/08/90
○
      Program address    :   MAST  CHART  S(C) 1   TOP  +1   ○
      Occurrence(s)     :   1      OCCURRENCE(S)
○      Program address    :   MAST  CHART  S(C) 1   TOP  +3   ○
      Occurrence(s)     :   1      OCCURRENCE(S)
○      Program address    :   MAST  CHART  S(C) 1   TOP  +5   ○
      Occurrence(s)     :   1      OCCURRENCE(S)
○      Program address    :   MAST  SR0           TOP  +2   ○
      Occurrence(s)     :   1      OCCURRENCE(S)
○      Program address    :   MAST  SR0           TOP  +4   ○
      Occurrence(s)     :   1      OCCURRENCE(S)
○      Program address    :   MAST  SR0           TOP  +6   ○
      Occurrence(s)     :   1      OCCURRENCE(S)
  
```



---

G



<b>Sub-section</b>	<b>Page</b>
<b>17.1 Presentation of a Mode</b>	<b>17/2</b>
This section ends on page	17/2

---

## 17.1 Presentation of the Mode

---

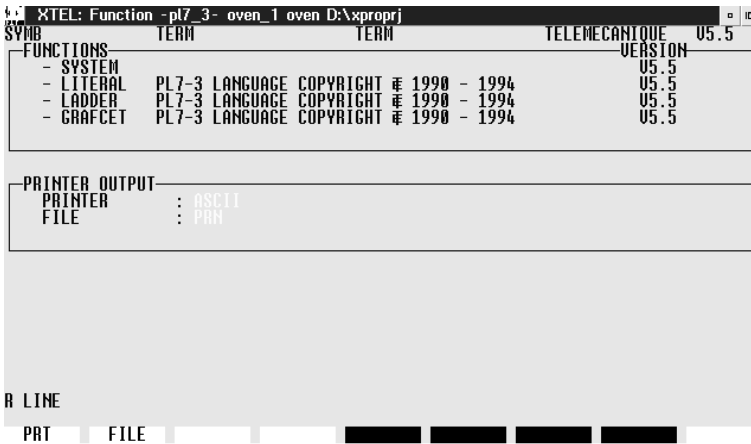
Terminal mode allows :

- Selection of the type of printer : IBM semi-graphic or ASCII.
- The printer output to be redirected to a PL7\_3.DOC file on the hard disk. This redirection concerns all printings of PL7-3 software, including printing of screens, printing of the application dossier, etc.

The PL7\_3.DOC file will include all the printing operations performed (copies of the screen, complete or partial application dossier, etc.).

This mode also displays the languages available and their software versions.

After Terminal mode is selected on the Mode Selection display, the following screen appears :



**[PRT]** Selects the type of printer : IBM semi-graphic or ASCII.

**[FILE]** Redirects the printer output to a hard disk file.

**[PRN]** restores the printer output.

**[MODIF]** enables modification of the characteristics of the redirection file.

**[FILENAME]** enables modification of the name of the file.

**[ZONE]** successively pressing on this key defines the storage zone of the file : APPLI or MOD.

**[PATH]** allows the directory to which the file is to be archived to be defined (storage pathway of the file).



<b>Section</b>	<b>Page</b>
<b>18 Appendices</b>	
Contents	18/1
18.1 Memory Allocation Overview	
18.2 Entering the Application Comment	
18.3 Selection of a V4 Processor	
18.4 Selection of a V5 Processor	
18.5 Storage and Retrieval of Files	
18.6 Search for the Program Addresses of a Variable using <Alt><X>	
18.7 Possibilities of the Different Rights of Access	
18.8 Converting a V3 Application to V4	
18.9 Converting a V4 Application to V5	
18.10 PL7-3 Help	
18.11 Modular V5	
18.12 List of Mnemonic Symbols	





<b>Sub-section</b>	<b>Page</b>
<b>18.1 Memory Allocation Overview</b>	18/3
<b>18.2 Entering the Application Comment</b>	18/4
<b>18.3 Selection of a V4 Processor</b>	18/5
<b>18.4 Selection of a V5 Processor</b>	18/6
<b>18.5 Storage and Retrieval of Files</b>	18/7
<b>18.6 Search for the Program Addresses of a Variable using &lt;Alt&gt;&lt;X&gt;</b>	18/8
18.6-1 General	18/8
18.6-2 Use in PROGRAM mode	18/8
18.6-3 Use in DEBUG mode	18/10
18.6-4 Example of Using the Command <Alt><X> in PROGRAM mode	18/12
<b>18.7 Possibilities of the Different Rights of Access</b>	18/16
<b>18.8 Converting a V3 Application to V4</b>	18/17
<b>18.9 Converting a V4 Application to V5</b>	18/18
<b>18.10 PL7-3 Help</b>	18/20
18.10-1 Note on the Help Screens	18/20



---

<b>Sub-section</b>	<b>Page</b>
<b>18.11 Modular V5</b>	<b>18/22</b>
18.11-1 Principle of Operation	18/22
18.11-2 Direct Access to an Elementary Module	18/22
18.11-3 Labelling an Elementary Module	18/24
18.11-4 Protection of an Elementary Module	18/25
18.11-5 View and Graphic Animation of a Macro-Module	18/27
<b>18.12 List of Mnemonic Symbols</b>	<b>18/30</b>
This section ends on page	18/30

---

## 18.1 Memory Allocation Overview

This screen, accessible by the command <Alt><T>, gives in detail the occupation of the PL7-3 standard program memory zone.

STANDARD PROG SPACE

CODE+GRAPH	VOLUME (WORDS)		
	RESERVED	USED	FREE
	94472	3250	91222
SYSTEM	1560	1560	0
TOTAL	96032	4810	91222

TASK SIZE	CODE		GRAPH	
	SEG1	SEG2	SEG1	SEG2
IT	0	0	0	0
FAST	0	0	0	0
MAST	2676	6	568	0
AUX0	0	0	0	0
AUX1	0	0	0	0
AUX2	0	0	0	0
AUX3	0	0	0	0

### Left-hand table

This table indicates the volume reserved and the volume occupied in the Memory 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 :
  - First segment; code objects of the MAIN program
  - Second segment; code objects of the SRs.

- 
- 2 graphic segments :
    - First segment; graphic objects for the task,
    - Second segment; objects for the diagnostic source program.

**Special case :**

When Grafcet is configured, the segments of the code for the Master task are as follows :

- First segment; Grafcet interpreter, code objects relating to Grafcet, CHART, XM, SR and MAIN.
- Second segment; action and transition condition code objects, PRL and POST.

---

## 18.2 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.

### V2 or V3 applications

The application comment can only be entered in Transfer mode (Terminal ↔ Disk/TSX) store/retrieve a program. It is accessed by using the [COMMENT] key and is entered in the operator input line. All characters are authorized, including a space, and the entry is limited to 24 characters maximum.

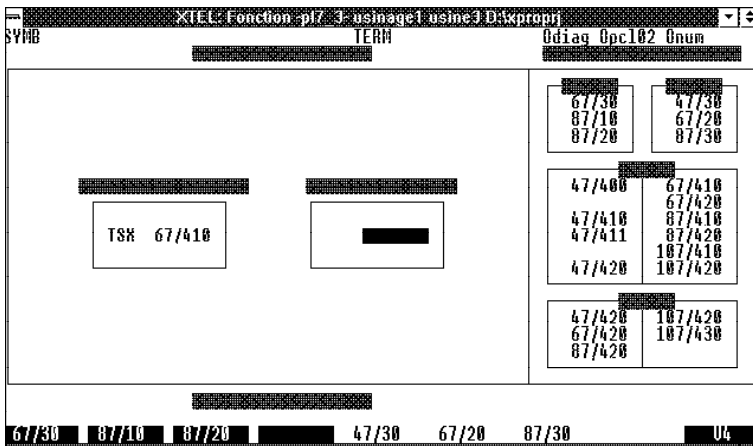
### V4 or V5 applications

Application comment entry is accessed by using the <Alt><K> keys in the Configuration, Program, Constant, Data, Debug and Document/Print modes. As for a V2 or V3 application, the comment is entered in the operator input line and is limited to 24 characters maximum.

### 18.3 Selection of a V4 Processor

For V4 applications, selection of the processor is not made at PL7-3 level, but by the MEM station tool of the software workshop, which defines the type of processor and the physical and logical memory spaces. To do this, proceed as follows :

- In the Functions window, define the various functions required for the application : PL7\_3, bridge, comm, axis, etc. The corresponding icons appear.
- Open a session of the MEM tool by using the Definition/Creation commands.
- Enter the name of a xxx.APP output file and validate it, which gives access to the selection of the processor and of the physical and logical memory spaces.
- Validate the selections so as to create a xxx.APP file without the associated binary file.
- Open a session of the PL7-3 function and, on the Mode Selection display, press [PROC], which causes initialization by default. The screen below appears :



**[V4]** Enables entry of the name of the xxx. APP file previously defined by the MEM tool. After validation, the type of processor memorized by the xxx.APP file appears in the Target Processor box on the screen. Validate the selection by [YES].

**[DIR]** Gives the list of existing xxx.APP files.

## 18.4 Selection of a V5 Processor

For V5 applications, selection of the processor is not made at PL7-3 level, but by the XTEL-CONF station tool, which is used to define the hardware resources required for the application. To do this, proceed as follows :

- In the Functions window, define the various functions required for the application : PL7\_3, comm, etc. The corresponding icons appear.
- Open a session of the CONF tool.
- Pull down the Definition menu and select the item I/O Config. in rack.
- Click twice on rack 0/1, to open a dialogue box to configure it : select the type of rack, processor, power supply and I/O modules.
- Select the various modules which are in the rack, including the processor ; confirm the configuration then exit the configuration function for in-rack modules.
- Pull down the Generation menu to generate the application file.
- Quit XTEL-CONF.
- When launching 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 appears in the PROCESSOR TYPE field in the Mode Selection display. After loading, PL7-3 automatically loads the PL7\_3.BIN file, which determines the type of processor.

In connected mode, the connected PLC is recognized automatically, and thus the processor. The type and version appears in the PROCESSOR TYPE field of the MODE SELECTION display.

---

## 18.5 Storage and Retrieval of 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 service required.

### Retrieval of one or more files

This function uses the Import service, which permits :

- The importation under PL7-3 of files generated under DOS
- The importation under PL7-3 of V5 files stored on hard disk (in another station) or on diskette.

To do this, the operator must :

- Progressively define the source address of the files, 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 operator (see Section 2.4).

The validation of the selections automatically copies the selected source files into the target zone.

### Storage of one or more files

This function uses the Export service which permits the transfer, to the exterior of the project 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 operator only needs to select :

- The zone containing the files (APPLI or MOD), by double clicking on the zone
- The file or files to be stored, by clicking on the corresponding names.

The destination address is selected by ascending the destination tree structure by double clicking on the various directories.

Validation of the selections automatically stores the selected file(s).

### Warning

In order to receive the files, the diskette must be formatted and not write protected.

---

## 18.6 Search for the Program Addresses of a Variable using <Alt> <X>

---

### 18.6-1 General

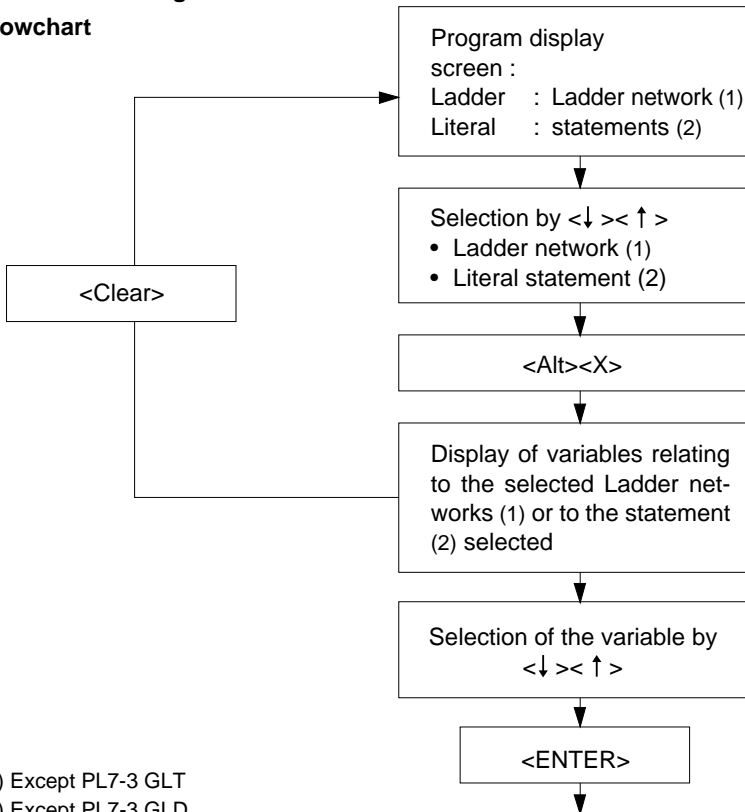
The command <Alt> <X> accesses cross references of a variable. When a Ladder network (1) or Literal statement (2) is displayed in the Program or Debug modes, this function permits :

- 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.

---

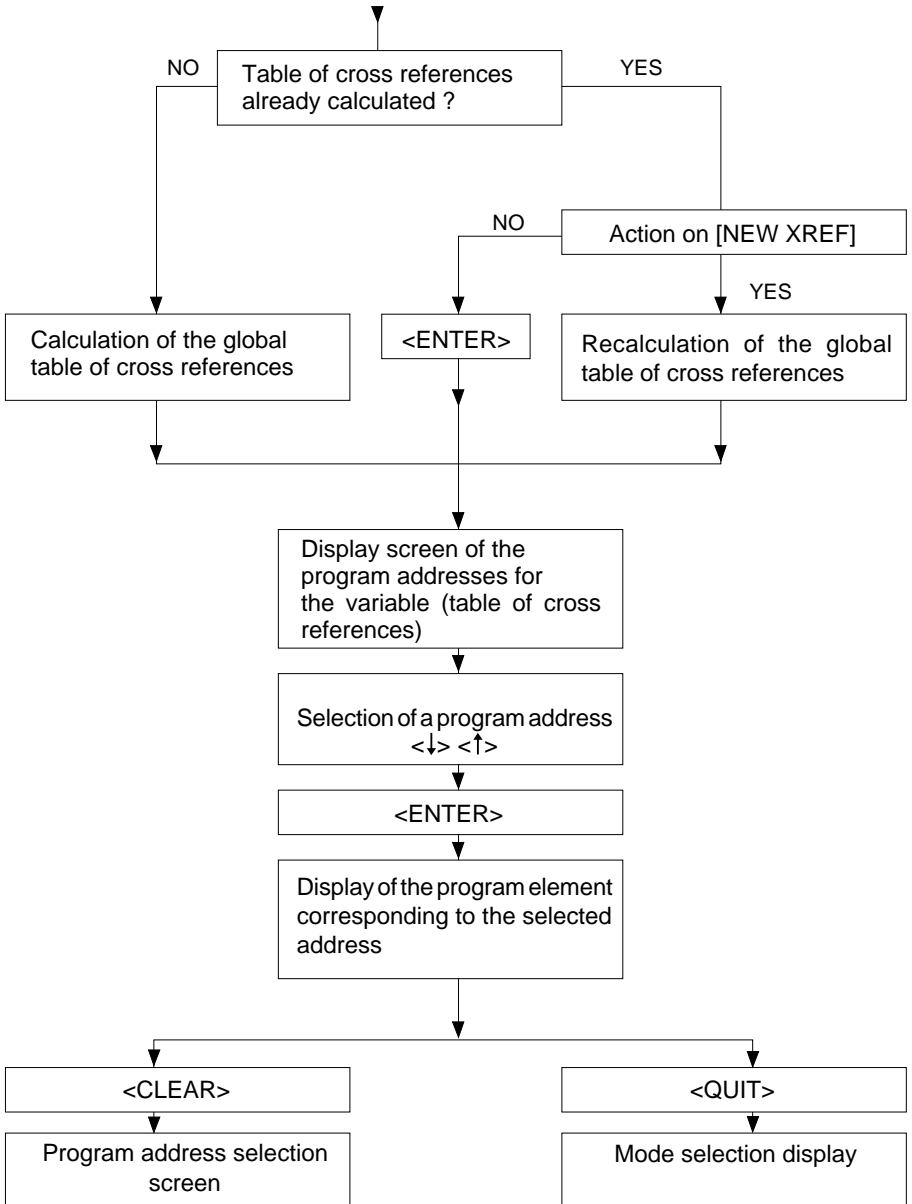
### 18.6-2 Use in Program Mode

#### Flowchart



(1) Except PL7-3 GLT

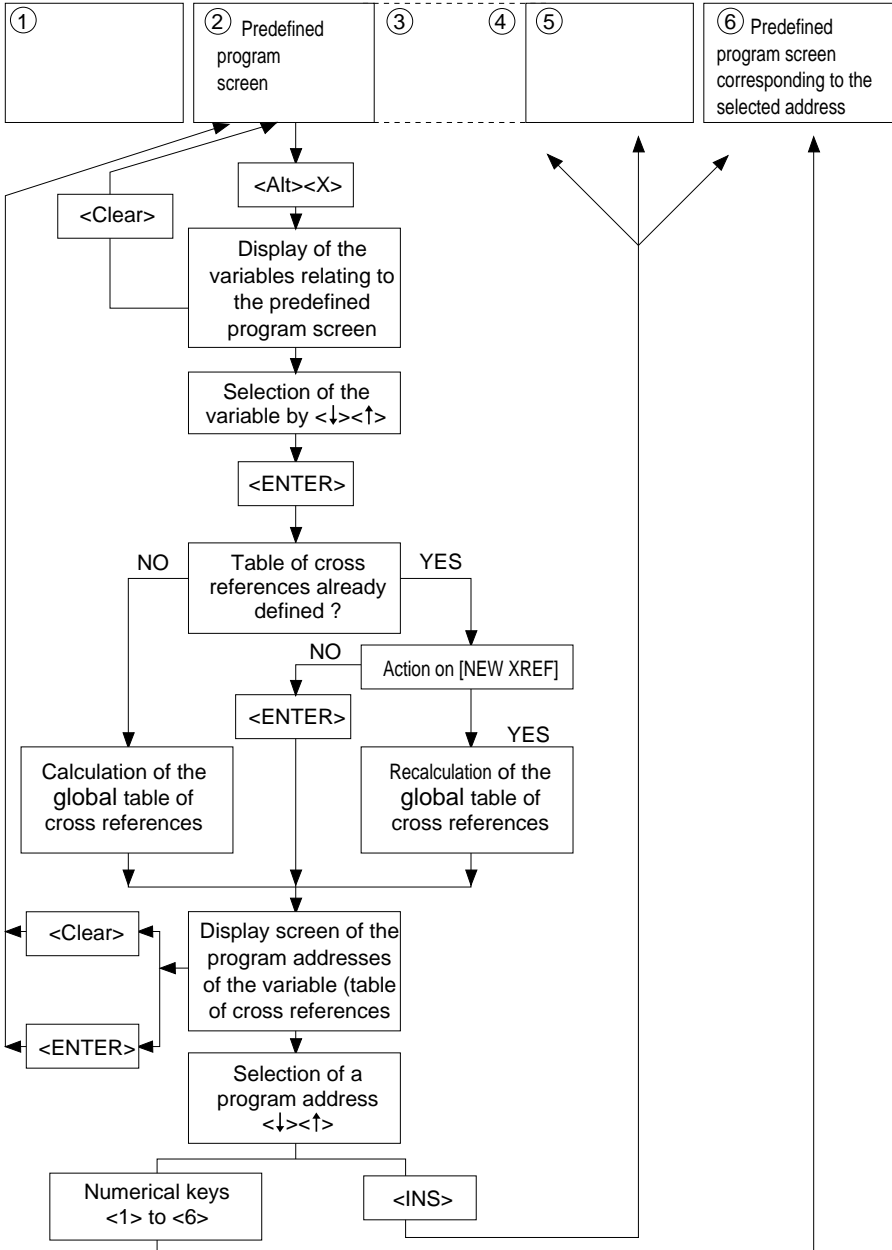
(2) Except PL7-3 GLD





### 18.6-3 Use in DEBUG mode

#### Flowchart



- 
- <INS>** Enables the program element relating to the selected variable to be inserted into the first available predefined screen. An error message appears if no predefined screen is available.
- <1> to <6>** Enables the program element relating to the selected variable to be inserted into the screen (the number must correspond to the selected key).

With this function, it is therefore possible to memorize up to 6 uses for the same variable and to move from one to another by means of the predefined screens.

## 18.6-4 Example of Using the Command <Alt><X> in PROGRAM mode

- **Display of the program element**

Position the cursor on the network (1) containing the variable to be analysed, as shown on the screen below.

```

XTEL: Function -pl7_3- oven factory E:\xproj
SYMB      PROG MAST PRL      TOP +2      TELEMECANIQUE U5.0
!         RESET B0; RESET B1; 0 -> W0; 100 -> W1; 100 -> W2; 10 -> W3
!         X1 -> B2
▶!        IF RE(B2)+FE(B2) THEN EXEC MSIT0(B0;B1;X1;W0;W1=>B3;W3)

[VARIABLE]

LABEL    TOP    BOT    MODIF    SAVE    RESTORE    DELETE    READ
  
```

- **Display of the list of variables of the selected program element**

When the above screen is displayed, press <Alt><X> to display all the standard PL7-3 variables of the selected network (1) (see the following screens).

```

XTEL: Function -pl7_3- oven_1 oven D:\xproj
SYMB      PROG      TELEMECANIQUE U5.5
VARIABLE  SYMBOL  COMMENT
10.4      Level
X0.1      .....
X0.3      .....
12.2      Senrev  Sensor of reverse position
12.4      Securev Security of reverse position
12.5      Secufwd Security of forward position
12.0      Amtrev  Ask for reverse motion
12.6      Security General security
12.7      .....
10.2      Cond    Monitored condition
10.3      Event   Monitored event
10.1      Ackn    Acknowledgement for diag OFBs
01.8      Defevn  Default of EUDGNO function block
01.9      Defevs  Default of EUDGSO function block

R LINE

VARIABLE  NEXT  XRF
  
```

**[VARIABLE]** Provides access for entering a PL7-3 object, in the form of a variable or mnemonic symbol, in order to display its cross references.

(1) Except PL7-3 GLT

**[NEW XREF]** Enables the new cross references to be calculated. If this key does not appear, it means that the cross references have not been calculated, in which case <ENTER> must be pressed to start the calculation before the cross references of the variable are displayed.

**[BACK]** Enables a return to the address previously selected by <Enter> in the cross reference screen of the variable (refer to screen below).

**[NEXT XRF] [PREV XRF]**  
 Accesses the previous or next page of cross references for different types of objects (standard PL7-3 variables, OFB elements, objects associated with remote I/O).

**[NEXT SCR] [PREV SCR]**  
 Accesses the previous or next page of cross references for objects of the same type.

<↓> <↑> Selects the variable.

**[NEXT]** Accesses the next cross references.

<ENTER> Displays the cross references of the selected variable (see the screen below).

- **Display of the uses of the variable in the program (cross references of the selected variable).**

VARIABLE	SYMBOL	TASK	MODULE	SUB MODULE	LABEL	OFFSET	USE
B2	.....	MAST	PRL		TOP	+1	W
		MAST	PRL		TOP	+2	R
		MAST	PRL		TOP	+2	R

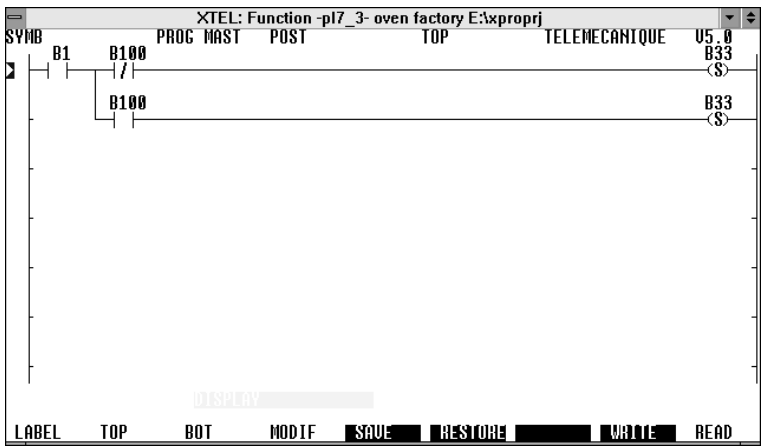
<↓> <↑> Enables the program address to be selected.

<ENTER> Displays the program element corresponding to the selected program address (see next screen).

**Note**

The USE column indicates the use of the variable in the program (see Section 15, Description of listings). The number that may be present in the OFFSET column indicates the position of the variable in the program in relation to the label shown in the LABEL column.

(1) Except PL7-3 GLT

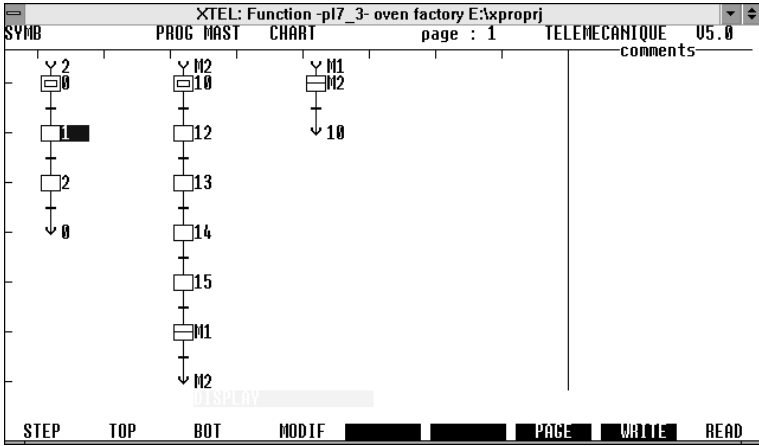


• **Special case where the program address is a Grafacet address**

In the case of a Grafacet address, as shown in the example below, the system does not automatically «ZOOM» on the sub-module indicated, but selects the Grafacet page concerned with the cursor placed on the step indicated on the left-hand side of the SUB-MODULE column.

XTEL: Function -pl7_3- oven factory E:\xproj								
SYMB		BITS			TELEMECANIQUE			U5.0
VARIABLE	SYMBOL	TASK	MODULE	SUB MODULE	LABEL	OFFSET	USE	
B1	.....	MAST	PRL		TOP		W	
		MAST	PRL		TOP	+2	R	
		MAST	CHART	X1 (C)	TOP		R	
		MAST	POST		TOP		R	

**<Enter>** Displays the Grafacet pages relating to the selected address (see the next screen).



**<ZOOM><ZOOM>** Displays the program element concerning the action associated with the step (see the next screen).

```

XTEL: Function -pl7_3- oven factory E:\xproprj
SYMB      PROC MAST  CHART S(C) 2  TOP      TELEMECANIQUE  U5.0
!         IF NOT M1,R.[W100>W102].[W100<W103] THEN SET O3,1 ELSE RESET B1
    
```

## 18.7 Possibilities of the Different Rights of Access

To program max.	_____					
To program min.	_____					
To adjust max.	_____					
To adjust min.	_____					
To operate max.	_____					
To operate min.	_____					
Access to ADJUST mode		yes	yes	yes	yes	yes
Access to CONTROL mode (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 constants of OFBs), PLC in STOP			yes	yes	yes	yes
Modification of the constants (CW + internal constants of OFBs), PLC in RUN				yes	yes	yes
Access to CONFIGURATION mode (display)		yes	yes	yes	yes	yes
Modification of the configuration in local mode					yes	yes
Modification of the configuration in connected mode PLC in STOP		yes	yes	yes	yes	yes
Modification of the configuration in connected 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 <sup>(1)</sup>	yes	yes
Replace, PLC in STOP					yes	yes
Replace, PLC in RUN						yes
Access to TERMINAL mode				yes	yes	yes

<sup>(1)</sup> Only in SEARCH

---

## 18.8 Converting a V3 Application to V4

---

This operation consists of retrieving a PL7-3 binary file (xxx.BIN) and transferring it to a V4 station. To do this, the operator must :

- Create (if it is not already done) the V4 target station :
  - Define in this station the PL7\_3 function, plus the other functions required
  - By using the PL7\_3 icon, start the Import service (using the Import command) which enables the V3 xxx.BIN file to be imported into the APPLI directory of the PL7\_3 function,
  - Start the MEM tool and create a xxx.APP file without the associated binary file (selection of the processor and the physical and logical memory spaces). Check that the PL7\_3 DATA and PL7\_3 PROG spaces are sufficient to receive the binary file.
- If not :
  - Adjust the size of the spaces for each of the TSX functions as required
  - Modify the choice of the processor or of the memory cartridge.
- Start PL7-3 in the V4 target station :
  - Read the xxx.BIN file stored in the APPLI directory of the function
  - Access the Type of Processor on the Mode Selection display by pressing [PROC]
  - Select the required V4 processor, then read the xxx.APP file previously defined by the MEM tool.

After validation, if the memory description for PL7-3 in the xxx.APP file is sufficient, and if the original binary file is compatible with the selected processor (in terms of NUM or network modules), the application is converted to V4 format.

To transfer this V4 application to the PLC, two solutions are possible :

- Complete the construction of the xxx.APP file by including the PL7-3 binary file, since this is now in the V4 format.  
The resulting xxx.APP file will be transferred to the PLC by the Station transfer tool.
- Transfer the xxx.APP file (without binary file) to the PLC by the Station transfer tool then, using PL7-3, transfer the binary file to the PLC in the Transfer mode.

### Note

If the physical spaces defined in the xxx.APP file are not sufficient to receive the binary file, the MEM tool checks whether it can satisfy the request by recuperating the space available in the PL7\_3 PROG zone.

- If this space is sufficient, the MEM tool proposes to construct a xxx.APP file by reducing the space reserved for PL7\_3, after checking the size required
- If this space is not sufficient, the MEM tool abandons the construction of the xxx.APP file and the following message appears : Insufficient physical memory.



---

## 18.9 Converting a V4 Application to V5

---

All V4 applications can be converted to V5 applications, provided the following operations are followed :

- ① Retrieve the V4 application under X-TEL V5, by **Store/Retrieve** or **Copy/Paste**.
- ② Create a V5 target station.
- ③ By using the PL7\_3 icon of the V5 target station, start the **Import** function and import the following files :

StationV4\PL7\_3\APPLI\xxx.BIN (compulsory) : binary application

StationV4\PL7\_3\APPLI\xxx.COM (optional) : program comments

to the StationV5\PL7\_3\APPLI directory

then

StationV4\PL7\_3\MOD\xxx.SCY (optional) : table of V4 mnemonic 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) : source constant words

to the StationV5\PL7\_3\MOD directory.

- ④ 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 target station.
  - Retrieve using the [RETRIEVE] command.
  - Return to the mode selection display (CLEAR).
  - Access the store screen with the command [STORE].
  - Activate the [STR.NAME] command and enter the name of the files to be stored to 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\APPLI\PL7\_3.BIN
    - StationV5\APP\STATION.IOC.
  - Quit PL7-3.

- 
- ⑤ Start the CONF station tool and perform the following operations :
    - From the **File** menu, activate **CASC to CONF** item (retrieves the V4 station configuration, contained in the STATION.IOC file). Reply "Yes" to the question posed.
    - In the **Definition** menu, activate the **Config. in rack I/O** item and if necessary modify the V5 processor proposed by default.
    - In the **Generation** menu, activate the **with task period entry** item and configure the period of the various tasks.
    - In the **Generation** menu activate the **with application parameters entry** item and if required modify the memory configuration.
    - Quit the CONF tool.
  - ⑥ 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.
  - ⑦ 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 retrieved : enter the name of the xxx.BIN file, previously imported under the V5 target 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 DIAGNOSTIC function then RECONFIGURATION function.
  - ⑧ Save the new application to disk using the [STORE] command.

---

## 18.10 PL7-3 Help

---

PL7-3 Help uses QuickHelp, to associate a help screen to each PL7-3 language screen. For further 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, by using the keywords or the pull down menus.

### <Help> or <F10>

Provides access to PL7-3 Help function. The Help screen associated with the current PL7-3 screen is displayed

### <X>

Returns to the current PL7-3 screen.

---

### 18.10-1 Note on the Help Screens

Each help screen has three main parts :

- A zone used to display the explanatory text for the associated PL7-3 screen
- The action bar, at the top of the screen, which accesses the pull down menus
- Vertical scroll bar, on the right of the screen, which allows the text to be scrolled.

#### Action bar

Each of the 6 items on the action bar accesses a pull down menu. To select an item the mouse can be used or the keyboard by entering the first letter (for example F for File).

**File :** To print a help screen, quit Help PL7-3, etc.

**View :** Enables a screen to be searched for, to access the previous and next screens, view the screen sequences, etc.

**Categories :** Accesses the list of screens. Selection of a screen in the list to be displayed.

**References :** Provides access to the list of keywords of the help screen displayed. Selection of a keyword giving access to the corresponding screen.

**Paste :** Enables the current window or current screen to be stored in the paste file (paste.qh by default).

**Options :** Enables modification of the size of the window or of the paste mode (information stored by the paste command is inserted or overtyped).

---

**Text zone**

A PL7-3 help screen always begins with a title (in green underlined in white on a colour 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 colour screen). Each region deals with a different subject : general description, possible action, etc, and may be divided into sub-regions (shown by a yellow asterisk on a colour screen).

The keywords, which permit direct access to a corresponding help screen, are highlighted (in white on a colour screen). The words shown in green on a colour screen (apart from the titles) are the important words in the text.

**Scroll bars**

By using the mouse on this bar, the text can be scrolled up or down on the screen. The arrow at the top and bottom of this bar can be used for the same purpose.

---

---

## 18.11 Modular V5

---

### 18.11-1 Principle of Operation

From the sources generated under PL7-3, XTEL-MOD enables the creation of elementary modules and macro-modules. Each elementary 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 mnemonic symbols and a comments file. The macro-modules comprise a set of elementary modules.

PL7-3 enables elementary modules and macro-modules created under XTEL-MOD to be recuperated and integrated into the application.

In V4, the modules are not accessible until they are integrated into the application. However, PL7-3 V5 allows the following functions :

- Direct access to the first line of code of an elementary module
- Labelling of an elementary module in the program and the documentation : name of the associated macro-module, name of the elementary module, module protection
- Write protection of an elementary module
- View and graphic animation of a macro-module
- Search/Replace in an elementary module or macro-module.

---

### 18.11-2 Direct Access to an Elementary Module

To read the elementary modules using the "READ XTEL-MOD/PL7-PMS2" function in the RETRIEVE screen, PL7-3 stores their physical addresses. Once this is done, the PL7-3 program can be accessed from the PROGRAM ADDRESS SELECTION screen and from a table containing all the macro-modules and elementary modules for the application : MODULE SELECTION screen (see section 7.1-2). This screen itself can be accessed :

- from the PROGRAM ADDRESS SELECTION screen in Program mode and Debug mode, using the [MOD SCR] soft key (refer to the corresponding modes)
- from the SEARCH/REPLACE mode screens using the [PARTIAL] command.

XTEL: Function -pl7\_3- oven factory E:\xproprj

SYMB PROC TELEMECANIQUE U5.0

MACRO MOD.	BASIC MOD.	R/RW	TASK	MODULE	FROM	VERS.	DATE
	HEAT	RW	MAST	PRL	MOD	U0	11-03-93
	MOTOR	RW	MAST	PRL	MOD	U0	11-03-93
FACTORY	PAINT	RW	MAST	POST	MOD	U0	11-03-93
	PAINT1	RW	MAST	PRL	MOD	U0	11-04-93
	PAINT2	RW	MAST	PRL	MOD	U0	11-04-93
INIT2	OVEN1	RW	MAST	PRL	MOD	U1	11-04-93
MOTOR2	MOTOR	RW	MAST	PRL	MOD	U0	11-03-93

MODUC SELECTOR

PREV SCR NEXT SCR ADR SCR

**In Program mode**

- <↓> <↑> Enables an elementary module to be selected.
- <ENTER> Accesses the first line of code in the program, of the selected elementary module.

**In Debug mode**

- <↓> <↑> Enables an elementary module to be selected.
- <ENTER> Accesses an animated program screen, as the first line of code is the first line of the selected elementary module.

**Note**

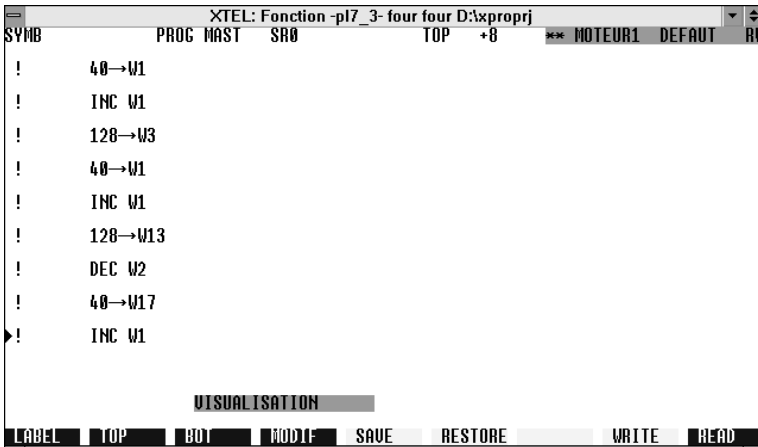
In Search/Replace mode the command [MOD SCR] also performs all the search and replace functions in an elementary module, in all the elementary modules in a macro-module or in all the elementary modules of an application program.

---

### 18.11-3 Labelling an Elementary Module

Labelling of elementary modules only applies to program modules. Constants, mnemonic symbols and program comments are not labelled. Labelling consists of displaying certain information, to show that a program line is part of an elementary module. This information : name of the macro-module associated with the elementary module if one exists (8 characters), name of the elementary module (8 characters) and protection linked to the module (2 characters), are displayed in the "application name" field in the Program, Debug and Search/Replace screen (in manual mode only).

In Documentation/Print mode, each elementary module is preceded by a line which indicates the start and label of the module. At the end of the module, a new line indicates the end and also the label of the module. This information appears on screen as well as on paper when the dossier is printed.



The screenshot shows a debugger window titled "XTEL: Fonction -pl7\_3- four four D:\xproj". The window displays assembly code with labels. The code is as follows:

```
SYMB      PROG  MAST  SR0      TOP  +8  ** MOTEUR1  DEFAULT  RM
!         40→W1
!         INC W1
!         128→W3
!         40→W1
!         INC W1
!         128→W13
!         DEC W2
!         40→W17
!         INC W1
```

At the bottom of the window, there is a "VISUALISATION" bar and a menu bar with the following options: LABEL, TOP, BOT, MODIF, SAVE, RESTORE, WRITE, READ.

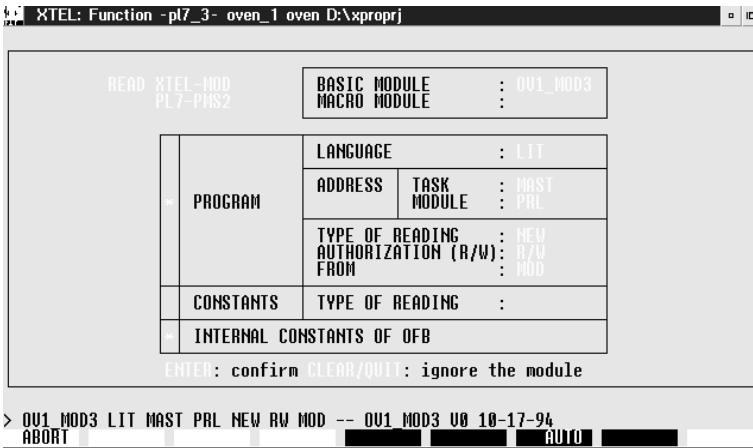
#### Note

If no macro-module is associated with an elementary module, the name of the macro-module is replaced by the characters "." (period).

### 18.11-4 Protection of an Elementary Module

An elementary 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 RETRIEVE function, maintains the functions associated with the module, including protection, which appear in all the modular V5 screens.

- RW : the module is accessible in read and write
- R : the module is read only accessible (write protected).



Consequences of write protecting an elementary module are as follows :

- It is impossible to modify or erase a write protected module whilst it is being inserted into the application (RETRIEVE). If a problem occurs during this operation, it can only be abandoned : [ABORT] key.
- If a new configuration deletes an elementary module which was write protected, a reconfiguration error will occur.
- In Program mode, it is not possible to modify a line of code of a write protected module :
  - The name of the variables or of the constants cannot be modified, even within a function block (MODIFY mode)
  - A line of code cannot be inserted or deleted (INS and DEL operations)
  - "Read" and "restore" operations are not available
  - Modifications cannot be performed in ZOOM mode
  - Windows cannot be moved or positioned.



- 
- In Program mode it is not possible to erase the write protected elementary modules which are "attached" to a macro-module, nor the program modules which contain a write protected elementary module. However, using the DEL command in the MODULE SELECTION screen, it is possible to erase :

- the macro-modules and thus at the same time all the elementary modules (write protected or not) which are associated with them
- the elementary modules (write protected or not) which are not associated with a macro-module.

By the DEL command in the PROGRAM ADDRESS SELECTION screen :

- the program modules not containing any write protected elementary modules.
- In Search/Replace mode, replacement is refused each time it is requested on a part of the program which contains a write protected module.
- In Debug mode, the write protected code cannot be modified. However, the values of variables may be modified.

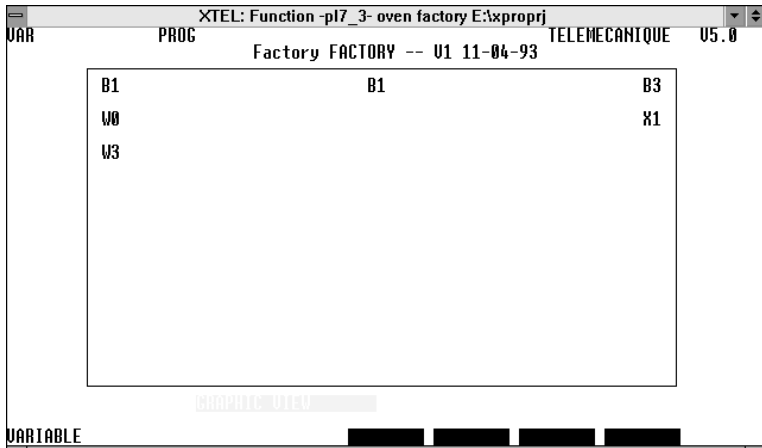
### 18.11-5 View and Graphic Animation of a Macro-Module

Under XTEL-MOD, a graphic representation can be associated with each macro-module which shows the macro-module as a block comprising the input variables, the internal variables and the output variables. These variables are freely defined by the user from the referenced variables in the elementary modules which make up the macro-module and cannot be modified in PL7-3. The PL7-3 MODULE SELECTION screen enables a macro-module to be selected and accessed using <ENTER> :

- with a graphic view of the macro-module, in Program mode
- with graphic animation of the macro-module, in Debug mode.

If the macro-module does not have an associated graphic (optional under XTEL-MOD), the message "18060 - VIGNETTE FILE NOT EXISTING" 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.

**[VARIABLE]** Displays the objects in the form of mnemonic symbols or variables.

<ALT><V> displays the objects in the form of variables.

<ALT><S> displays the objects in the form of mnemonic symbols.

**[ASS NAME]** Displays the objects in the form of associated names (defined under XTEL-MOD). The objects which do not have an associated name are displayed as variables.

#### Note

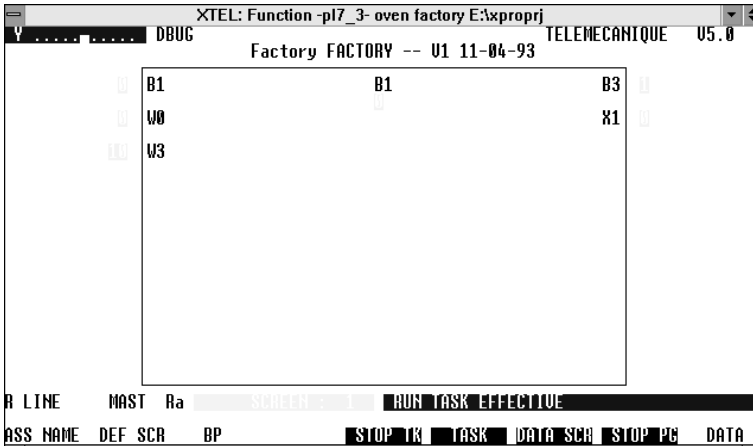
If an object is defined in the graphic representation associated with a macro-module, but is not configured in the application, the graphic view of the macro-module in VARIABLE mode displays the character string "\*\*\*" in place of the object. However, in ASSOCIATED NAME mode, the object is displayed normally, in order to allow the user to know the name of the variable which has not been configured.

---

## 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 an elementary module, which can be selected under XTEL-MOD (Select module photo), or selected arbitrarily by PL7-3 : first elementary module of a macro-module.

- If the selected elementary 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 module selected is a Grafcet module, the "photo" is taken at the start of the master task cycle.



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.

**[VARIABLE]** Displays the objects in the form of mnemonic symbols or variables.

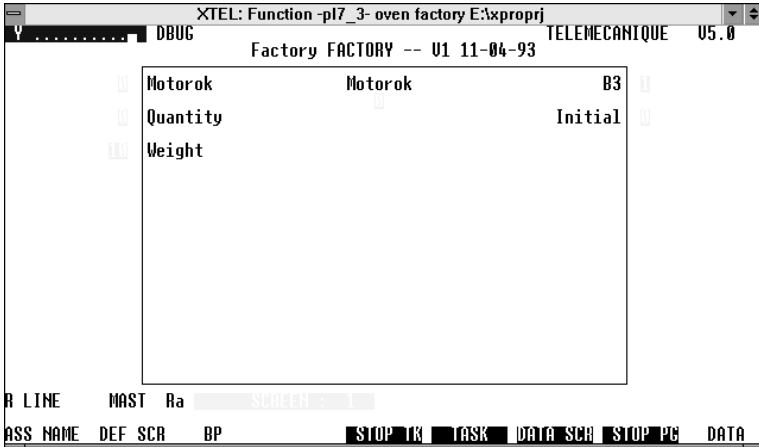
**<ALT><V>** displays the objects in the form of variables.

**<ALT><S>** displays the objects in the form of mnemonic symbols.

**[ASS NAME]** Displays the objects in the form of associated names (defined under XTEL-MOD). The objects which do not have an associated name remain displayed as variables.

**[DEF SCR]** Accesses the modification screen for the notation base for the display.

The function of the other soft keys remains the same as described in section 13 - Debug Mode.



This screen displays in the display band, the variable, the mnemonic symbol or the name associated with the selected variable. This allows variables which have more than 16 characters to be displayed in their entirety.

- [TOP]** Positions the cursor on the first variable in the column in which it is positioned : first input, internal or output variable.
- [BOT]** Positions the cursor on the last variable in the column in which it is positioned : last input, internal or output variable.
- [BASE]** Used to modify the notation base for the display of the variable selected by the cursor : decimal, hexadecimal, BCD, message (ASCII) or floating point (PMX only).
- <ENTER>** Confirms the modifications made to the base and accesses the previous screen.
- <CLEAR>** Returns to the previous screen without confirming the modifications.

**Note**

If an object is defined in the graphic representation 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. If this object is configured, but is not to be animated, the character string "\*\*\*\*" is displayed in place of the object in VARIABLE mode.

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.

---

## 18.12 List of Mnemonic Symbols

---

This screen is accessed by the command <Ctrl><F5>, 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 network or Literal statement).

VARIABLE	PROC	SYMBOL	COMMENT
M		Motorok	Motor running

SDBASE

**[SDBASE]** Accesses the XTEL-SDBASE tool symbol editor.

**<CLEAR><QUIT>**

Returns to the program display screen.

**PL7\_3 SHEET****PL7\_3 EXECUTION TIME ; V5 Model 40 PROCESSORS****1. SUMMARY**

The processor power ratios given below are average figures measured on typical programs. They only affect execution of instructions.

TSX 4730 and TSX 6720 V3 processors are used as examples.

Type of processing	V3	V4			V5				
	TSX 6720	TSX PMX 674x0	TSX PMX 874x0	TSX PMX 1074x0	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
Boolean on I/O or internal bits	1.0	1.08	1.08	1.68	1.08	1.58	1.08	1.68	1.50
Boolean on word extract bits	1.0	1.2	2.3	4.4	1.2	1.4	2.3	2.8	4.9
Numeric (arithmetic)	1.0	1.3	2.7	5.5	1.3	1.6	2.6	3.4	5.5

**2. "OVER-HEAD" TASK**

(Time in microseconds):	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>MAST MASTER task</b>	3500	3050	1900	1680	1030
<b>FAST FAST task</b>	1300	1070	550	470	260
<b>AUX0 AUXILIARY task</b>	1300	1200	610	520	300

**3. NORMAL I/O EXCHANGES**

Base racks and direct I/O extension racks (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Discrete modules with 4I or 4O</b>	N/A	N/A	N/A	N/A	N/A
<b>Discrete modules with 8I or 8O</b>	60	55	37	31	31
<b>Discrete modules with 16I or 16O</b>	80	74	48	42	40
<b>Module with 8 REG IN + 8 REG OUT</b>	920	850	370	300	270

Local I/O extension racks (LES 20) (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Discrete modules with 4I or 4O</b>	N/A	N/A	N/A	N/A	N/A
<b>Discrete modules with 8I or 8O</b>	80	71	70	62	62
<b>Discrete modules with 16I or 16O</b>	100	89	86	75	75
<b>Module with 8 REG IN + 8 REG OUT</b>	1140	1070	784	700	680

Fiber optic remote I/O extension racks (LFS 200) (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Discrete modules with 4I or 4O</b>	N/A	N/A	N/A	N/A	N/A
<b>Discrete modules with 8I or 8O</b>	104	92	91	81	81
<b>Discrete modules with 16I or 16O</b>	130	116	112	98	98
<b>Module with 8 REG IN + 8 REG OUT</b>	1480	1390	1020	910	884

#### 4. PROGRAM PROCESSING

GRAF CET development (master task).

Time taken by the Grafcet interpreter to calculate the new stable state :

$$TG7 = TGF + ( NEA \times TEA ) + ( NTP \times TTP )$$

TG7 = Grafcet time,  
 TGF = Set time,  
 NEA = Number of simultaneous Active Steps,  
 TEA = Step Activation Time,  
 NTP = Number of Validated Transitions,  
 TTP = Calculation time for a Validated Transition.

(Time in microseconds):	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
TGF	1010	900	430	320	170
TEA	100	88	44	40	21
TTP	215	190	90	80	38

#### 5. INSTRUCTION EXECUTION TIME

Phrase structure (Time in microseconds):	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Start of literal phrase</b> ! or !Label	0.50	0.32	0.50	0.32	0.32
<b>Action separator</b> (except SET;RESET if test=0)	0.50	0.32	0.50	0.32	0.32
<b>Start of Ladder network</b> (Network + Label)	0.50	0.32	0.50	0.32	0.32
<b>LITERAL structure</b> IF Bi THEN SET Bj {Bi=0}	9.5	8.5	7	5	4
IF Bi THEN SET Bj {Bi=1}	8	7	6	4.5	3.5
IF Bi THEN SET Bj ELSE SET Bk {Bi=0}	11	10	8	5.5	4.5
IF Bi THEN SET Bj ELSE SET Bk {Bi=1}	18	17	13	9	6.5
<b>LADDER structure</b>	0	0	0	0	0



**5. INSTRUCTION EXECUTION TIME (continued)**

Bit instructions (I/O;Bi;Xi;Sy) (Time in microseconds):	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
Bit	0.5	0.32	0.5	0.32	0.35
NOT Bit	0.5	0.32	0.5	0.32	0.35
RE (Bi)	0.5	0.32	0.5	0.32	0.35
FE (Bi)	0.5	0.32	0.5	0.32	0.35
-> Bit	1.0	0.64	1.0	0.64	0.7
SET Bit	1.0	0.64	1.0	0.64	0.7
RESET Bit	1.0	0.64	1.0	0.64	0.7
+	0.5	0.32	0.5	0.32	0.35
.	0.5	0.32	0.5	0.32	0.35
()	0.0	0.0	0.0	0.0	0.0
Indexing	+32.0	+28.0	+18.0	+15.0	+7.00
NOT Bi -> Bi	2.2	1.8	2.3	1.5	1.7
RE (Bi) -> Bi	2.2	1.8	2.3	1.5	1.7
FE (Bi) -> Bi	2.2	1.8	2.3	1.5	1.7
Bi + Bj -> Bk	2.7	1.8	2.9	1.8	2
Bi (Wj) -> Bk	34	30	20	16	9

Bit string instructions (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
imm -> Bi[n]	24 +0.55n	20 +0.3n	15 +0.5n	11.5 +0.3n	8 +0.3n
Wi->Bj[n]	26 +0.55n	21 +0.3n	16 +0.5n	12.3 +0.3n	8.3 +0.3n
Bi[n] -> B[n]	40 +0.55n	30 +0.3n	23 +0.5n	16 +0.3n	12 +0.3n

Word bit instructions (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
Wi,j	19	15	11	8	5
Wi,j -> Bk	20	16	12	9	6
SET Wi,j	14	11	8	6	3.5
RESET Wi,J	14	11	8	6	3.5
Wi,j -> Wk,l	31	27	18	14	8
CWi,j -> Bk	56	48	30	23	14

Operations on words Time increases (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
by CW instead of W object	42.5	30	19	14	7
by CDW instead of DW object	44	32	20	15	8
by indexed object	23	20	12	9.5	5

**5. INSTRUCTION EXECUTION TIME (continued)**

Operations on words Arithmetic (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
imm --> Wi	14	12	9	6.5	4
imm --> DWi	20	18	13	9	6
Wi + Wj --> Wk	37	30	20	14	9
Wi * Wj --> Wk	37	30	20	14	9
Wi * Wj --> Wk(overflow)	78	67	42	31	21
Wi / Wj --> Wk	37	30	20	14.5	9.5
Wi / Wj --> Wk{division by 0}	86	73	47	36	23.5
Wi REM Wj --> Wk	38	31	20	15	9.5
Wi REM Wj --> Wk {div by 0}	87	74	48	36	24
SQRT ( Wi ) --> Wj	98	80	44	34	17
INC Wi	9	8	6	4	2.5
DEC Wi	9	8	6	4	2.5

Operations on words Logic (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
imm AND Wi --> Wj	30	25	18	13	8
imm OR Wi --> Wj	30	25	18	13	8
imm XOR Wi --> Wj	30	25	18	13	8
CPL ( Wi ) --> Wj	21	19	14	10	6

Operations on words Comparisons (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
[ Wi = imm ] --> Bj	25	20	17	12	8
[ Wi >= imm ] --> Bj	25	20	17	12	8
[ Wi = Wj ] --> Bk	25	20	17	12	8
[ Wi >= Wj ] --> Bk	25	20	17	12	8
[ DWi = imm ] --> Bj	30	26	19	14	9
[ DWi >= imm ] --> Bj	30	26	19	14	9
[ DWi = DWj ] --> Bk	30	26	19	14	9
[ DWi >= DWj ] --> Bk	30	26	19	14	9

Operations on words Shifts (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
SHL n (Wi) --> Wi	42	36	24	18	11
SHR n (Wi) --> Wj	42	36	24	18	11
SLC n (Wi) --> Wi	42	36	24	18	11
SRC n (Wi) --> Wj	42	36	24	18	11
SLCWORD (Wi;Wj)	19	16	11	8	4

**5. INSTRUCTION EXECUTION TIME (continued)**

Operations on words Conversions (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
DTB (Wi) --> Wj	58	46	23	18	9
BTD (Wi) --> Wj	52	41	21	16	9
GTB (Wi) --> Wj	81	74	37	33	11.5
GTB (Wi) --> Wj	81	74	37	33	11.5
BTA (Wi) --> Wj[3]	80	67	35	30	13
BTA (DWi) --> Wj[6]	460	377	152	125	65
ATB (Wi[3]) --> Wj	120	103	53	43	18
ATB (Wi[6]) --> DWj	2000	1666	655	570	300

Operations on tables (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
imm --> Wi[10]	26	23	15	10	7
imm --> DWi[10]	80	63	36	26	15
Wi[10] --> Wj[10]	56	51	27	22	13
CWi[10] --> Wj[10]	95	83	44	36	22
Wi[10] + Wj[10] --> Wk[10]	188	166	79	68	32
EQUAL(Wi[10];Wj[10])-->Wk	69	60	34	27	15
EQUAL(Wi[10];Wj[10];Wk)-->WI	70	60	35	28	15
SEARCH (Wi[10];Wj) --> Wk	80	70	40	31	18

Operations on FLOATING POINT words Arithmetic (Time in microseconds) :	PMX 474x5	PMX 674x5	PMX 874x5	PMX 1074x5
Imm floating point --> DWi	17	10	8	6
ADDF(DWi;DWj) --> DWk	135	58	46	28
ADDF(DWi;DWj) --> DWk {error SY18=1}	145	70	53	34
SUBF(DWi;DWj) --> DWk	137	58	46	28
SUBF(DWi;DWj) --> DWk {error SY18=1}	145	70	53	34
MULF(DWi;DWj) --> DWk	135	58	46	28
MULF(DWi;DWj) --> DWk {error SY18=1}	145	70	53	34
DIVF (DWi;DWj) --> DWk	149	62	49	30
DIVF (DWi;DWj) --> DWk {division by 0}	150	72	56	35
SQRTF (DWi) --> DWj	124	53	42	25
SQRTF (DWi) --> DWj {negative number}	118	61	48	29

Operations on FLOATING POINT words Comparisons (Time in microseconds) :	PMX 474x5	PMX 674x5	PMX 874x5	PMX 1074x5
EQUF (DWi;DWj) --> Bk	112	53	42	26
EQUF (DWi;DWj) --> Bk {error SY18=0}	118	57	45	27
SUPF (DWi;DWj) --> Bk	112	52	41	25
SUPF (DWi;DWj) --> Bk {error SY18=0}	118	56	43	27
INFF (DWi;DWj) --> Bk	112	52	41	25
INFF (DWi;DWj) --> Bk {error SY18=0}	118	56	43	26

**5. INSTRUCTION EXECUTION TIME (continued)**

Operations on FLOATING POINT words Conversions (Time in microseconds) :	PMX 474x5	PMX 674x5	PMX 874x5	PMX 1074x5
BTF (DWi) → DWj	99	44	34	21
FTA (DWi) → Wj[5]	2150	820	665	360
FTB (DWi) → DWj	120	59	46	28
FTB (DWi) → DWj { error SY18=1}	116	60	46	28
ATF (Wi[7]) → DWj	1710	660	555	266
ATF (Wi[7]) → DWj {error SY18=1}	125	55	50	25
FTD (DWi) → Wj[5]	140	46	35	22
FTD (DWi) → Wj[5] {error SY18=1}	100	49	37	24
DTF (Wi[5]) → DWj	127	46	35	22

Instructions on function blocks (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
START Ti	40	36	22	18	7
PRESET Ti	32	28	18	14,5	6
STOP	31	27	18	14	6
STOP Ti {Ti in stop}	29	26	18	13,5	5,5
START Ti {Ti in run}	35	30	18	15,5	6,5
PRESET Ti {Ti in run}	35	30	18	15,5	6,5
START Mi	30	26	17	14	6
START Mi {Mi in run}	30	27	17	14	6
RESET Ci	45	40	23	19,5	9
PRESET Ci	48	41	25	21	9
UP Ci	38	34	20	17	7
DOWN Ci	40	34	20	17	7
RESET Ri	32	29	16	13,5	6
PUT Ri	42	36	21	17	8
GET Ri	42	36	21	17	8

Instructions on function blocks Task control (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
START CTRLi {inactive task}	48	43	22	18	9
START CTRLi {active task}	48	43	22	18	9
RESET CTRLi {active task}	52	47	25	21	10
RESET CTRLi {inactive task}	52	47	25	21	10

**5. INSTRUCTION EXECUTION TIME (continued)**

Program structure instructions (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
WHILE ... DO ... {condition false}	8	7	5	4	3
WHILE..DO.. {condition true n times}	8+23n	7+20n	5+12n	4+10n	3+6n
JUMP Li	20	14	10	9	5
CALL SRI	63	56	32	26	15
RET	36	32	18	15	10

Instructions on function blocks TEXT (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Local, direct addressing</b>					
OUTPUT TXTi	820	710	315	280	120
INPUT TXTi	780	685	310	270	115
EXCHG TXTi	1550	1340	610	530	240
RESET TXTi (input-output) (exchg)	762 1372	670 1206	300 540	266 480	115 208
<b>Local, indirect addressing</b>					
OUTPUT TXTi	1210	1050	480	410	190
INPUT TXTi	1180	1030	460	400	185
EXCHG TXTi	2000	1700	760	675	310
RESET TXTi (input-output) (exchg)	856	752	337	399	130
<b>Network, direct addressing</b>					
OUTPUT TXTi	1085	933	412	366	159
INPUT TXTi	1085	908	401	356	154
EXCHG TXTi	1718	1479	653	582	251
RESET TXTi (input-output) (exchg)	793	697	312	277	120
<b>Network, indirect addressing</b>					
OUTPUT TXTi	1410	1240	555	493	214
INPUT TXTi	1426	1228	541	482	154
EXCHG TXTi	2144	1846	815	723	314
RESET TXTi (input-output) (exchg)	887	780	345	310	135

Immediate I/O exchange instructions (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
WRITEBIT ( Bi ; lx ) {16 bits}	150	122	83	65	44
WRITEBIT ( Bi ; lx ) {8 bits}	135	112	68	55	34
READBIT ( lx ; Bi ) {16 bits}	135	112	80	61	42
READBIT ( lx ; Bi ) {8 bits}	123	105	69	55	34
WRITEREG ( Wi ; lx )	300	238	186	136	108
READREG ( lx ; Wi )	310	250	190	144	116

**5. INSTRUCTION EXECUTION TIME (continued)**

Interrupt processing instructions (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>MASKINT (Ix)</b>	110	100	52	44	25
<b>DMASKINT (Ix)</b>	109	98	51	42	24
<b>ACKINT (Ix)</b>	110	100	52	44	25
<b>READINT (Ix;Bi)</b>	130	120	65	54	31
<b>SETIT (Ix)</b>	110	98	52	42	25

**6. IT TASK SPEED OF REACTION TO INTERRUPTIONS**

Position of module generating the IT (Time in milliseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Base rack</b>	1.2	1	0.7	0.5	0.3
<b>Local I/O extension rack (LES20)</b>	1.3	1.1	0.8	0.6	0.4
<b>Electrical remote I/O extension rack (LES 200)</b>	2.0	1.8	1.5	1.3	0.9
<b>Fiber optic remote I/O extension rack (LFS 200)</b>	2.0	1.8	1.5	1.3	0.9

**7. MESSAGE HANDLING**

Incoming message Time per byte (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Base rack</b>	320 +15n	260 +10n	226 +8.6n	214 +8n	148 +8n
<b>Local I/O extension rack (LES20)</b>	320 +30n	260 +20n	226 +17n	214 +16n	148 +16n
<b>Electrical remote I/O extension rack (LES 200)</b>	320 +40n	260 +26n	226 +23n	214 +22n	148 +22n
<b>Fiber optic remote I/O extension rack (LFS 200)</b>	320 +40n	260 +26n	226 +23n	214 +22n	148 +22n

Outgoing message Time per byte (Time in microseconds) :	PMX 474x5	TSX 474x5	TSX PMX 674x5	TSX PMX 874x5	TSX PMX 1074x5
<b>Base rack</b>	840 +15n	668 +10n	424 +8.6n	338 +8n	204 +8n
<b>Local I/O extension rack (LES20)</b>	840 +30n	668 +20n	424 +17n	338 +16n	204 +16n
<b>Electrical remote I/O extension rack (LES 200)</b>	840 +40n	668 +26n	424 +23n	338 +22n	204 +22n
<b>Fiber optic remote I/O extension rack (LFS 200)</b>	840 +40n	668 +26n	424 +23n	338 +22n	204 +22n

---

## Explanatory note for software upgrade kit TXT RK PL7 3 D / T V52 •

---

### Composition of products for updating PL7 3 Junior to PL7 3 V52

Two versions are available :

- TXT RK PL7 3 D V52 : PL7-3 D V43 or V5 to PL7-3 V52 upgrade kit,
- TXT RK PL7 3 T V52 : PL7-3 T V43 or V5 to PL7-3 V52 upgrade kit.

These products consist of :

- 1 incremental key, reference TXT RK PL7 3 D (or T) V5, to upgrade PL7-3 D (or T) V43 to PL7-3 D (or T) V5,
- 1 incremental key, reference TXT RK PL7 3 D (or T) V52, to upgrade PL7-3 D (or T) V5 to PL7-3 V52,
- this service instruction.

#### Note

The diskettes and documentation are provided in the XTEL PACK kit.

### Description of the operations to be performed

1. Install the V52 software from the diskettes provided in the XTEL PACK kit on the FTX, IBM PC or compatible microcomputer, which is equipped with the OS/2 operating system, version 1.3, 2.1 or WARP 3.0.

**Note** : for upgrade kits, it is not necessary to uninstall the earlier versions (refer to the general rules for compatibility between the various versions, at the end of the document).

2. Prepare the software keys using the XTEL : KEY MANAGER function, described in the X-TEL V52 software workshop documentation, reference TXT DM XTEL V52•.

This product contains 2 software protection keys, containing the PL7-3 D (or T) V5 rights (key reference TXT RK PL7 3 D (or T) V5) and the PL7-3 V52 rights (key reference TXT RK PL7 3 D (or T) V52) respectively. It is also necessary to have the work key for the terminal which was previously equipped with V4 or V5 software and its backup key.

An updated work key must be created (with its backup). To do this :

- insert the work key and the backup key in slots A and B in the terminal or the key support,
- restore the rights (Restore function) from the terminal backup key to the work key.  
Result : the work key becomes the original key for the terminal again, and the backup is empty,
- leave the original key in position,

- 
- insert the V4 to V5 upgrade key, reference TXT RK PL7 3 D (or T) V5 (only if your terminal was previously equipped with V43 software). Select the **Increment** function and confirm. Result : the upgrade key becomes empty and the original key contains the PL7-3 D (or T) V5 right,
  - insert the V5 to V52 upgrade key, reference TXT RK PL7 3 D (or T) V52. Select the **Increment** function and confirm. Result : the upgrade key becomes empty and the original key contains the PL7-3 V52 right,
  - back up the rights (**Backup** function) from the original key onto one of the empty keys,
  - keep the backup key in a safe place,
  - **all the software can now be launched from the terminal.**