

### Presentation

PL7 Micro/Junior/Pro software packages are designed for Windows 2000 Professional and Windows XP (1) operating systems and therefore benefit from all the facilities relating to these.

PL7 version 4.4 enables new TSX Micro/Premium modules to take advantage of additional functionalities:

- PL7 software registrations by the Internet, electronic mail, fax or phone (obligatory registration before 22nd day).
- Change in the Fipio catalogue for Momentum distributed I/O.
- Transfer of PL7 user rights between PCs via floppy disk or network.
- Enriched export files of the data application (FEF) for better compatibility with Unity software after migration.

PL7 Micro Junior/Pro software packages are offered according to two alternatives depending on the type of PLCs/PC cable: connection on the RS 232C port, or the USB port of a PC.

### Ergonomics of the software

More user-friendly and productive ergonomics thanks to:

- Access to contextual menus by right-clicking with the mouse for fast access to the services available for the selected object.
- Contextual help: direct access to help corresponding to the selected object.
- Tool tips: explanatory messages appear when skimming over the toolbar buttons.

Furthermore, some important functions make the software easier and safer to use:

### Multi-instance

The multi-instance function enables several applications to be worked on simultaneously.

This function enables:

- Several different applications present on the PC to be opened in offline mode in order to check or copy data.
- Debugging of two (or more) applications on two PLCs present on the same network in on-line mode. This is particularly useful when debugging inter-PLC communication functions.

### Management of access rights

Use of the various PL7 software functions can be limited and controlled by managing the access rights.

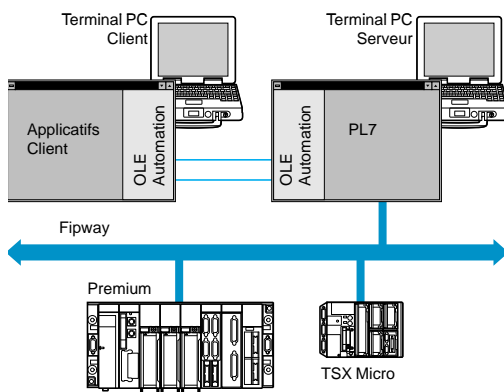
There are 5 user profiles (differentiated by passwords) which characterize the functions available to users on the programming terminal. The profiles range from read-only access to an application (lowest profile) to full programming (profile with the most rights).

### Application server

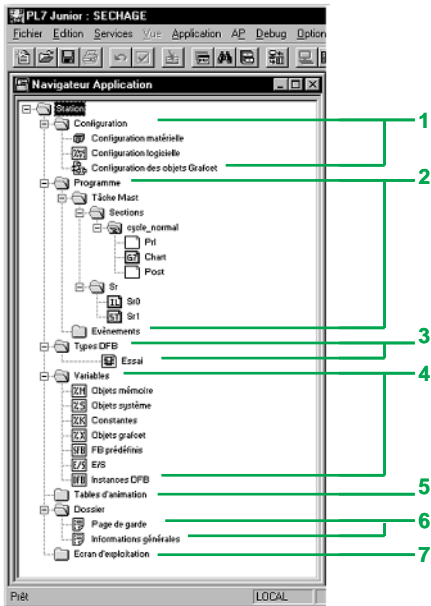
PL7 Pro software can be launched in OLE Automation server mode from a third-party client application. In this case, certain PL7 program functions can be executed following commands sent by an OLE client application. This instance of the program then no longer reacts to commands from the operator keyboard. Server mode can be launched in offline mode (COM) if both programs are on the same machine, or in remote mode (DCOM), if the programs are installed on different machines.

The commands available are as follows:

- Manage an execution context (open/close an application, modification of the address and driver of the connected PLC; PLC status).
- Control the PLC (connection/disconnection, send a RUN/STOP/INIT command, program uploading/downloading).
- Read data (application or symbol export only in source format, read symbol/comment associated with an address, read application identity).



(1) However, compatibility with the Windows 95 operating system is no longer provided and USB port is not compatible with Windows 95 and Windows NT 4.0.



### Application creation and debug tools

#### Application browser (conventional view)

Access to all programming and debug tools is gained via the application browser. This gives a global view of the program and enables all the application components to be accessed quickly via contextual menus.

- 1 Configuration editor.
- 2 Program editor.
- 3 DFB user function block editor.
- 4 Variable editor.
- 5 Animation table editor.
- 6 Documentation editor.
- 7 Operating screen editor.

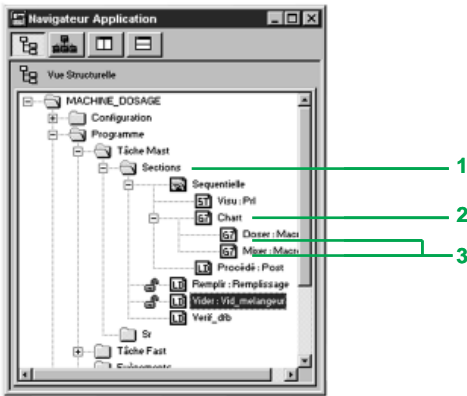
#### Concept of sections and Grafcet enhancement

In order to make programs more comprehensible, the FAST and MAST tasks are split into sections.

Each section 1 has a name, a comment and is programmed in one of the four languages available in PL7.

A section programmed in Grafcet language can contain a main chart 2 and macro-steps 3. Version ≥ V4.0 of PL7 allows comments to be added to each macro-step.

To protect intellectual property or avoid any unwarranted modification, each section can be write-protected or read/write-protected.



#### Function views of an application

PL7 Pro software can be used to structure an application for a Premium platform into functional modules that are broken down into sections (program code), animation tables and operating screens. Independently of the multitask structure of the PLC, the designer can define a multilevel tree structure of the automation application.

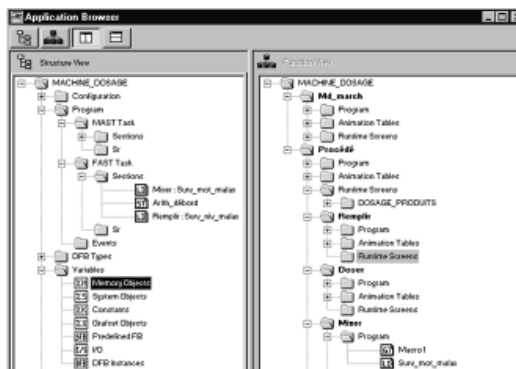
At each level, it is possible to attach program sections written in Ladder language (LD), Structured Text (ST), Instruction List (IL), Grafcet (SFC), and animation tables.

Two types of view are available at all times:

- A representation showing a tree structure of modules can provide a breakdown according to consistent functions in relation to the process to control.
- The classic representation of the application browser provides a view of the execution order of the program sections for each PLC.

The operation services associated with the functional view are available in one or the other view. In particular, a single command can be used to force whether or not a functional module is executed.

In this case, every section attached to the functional module is automatically forced.

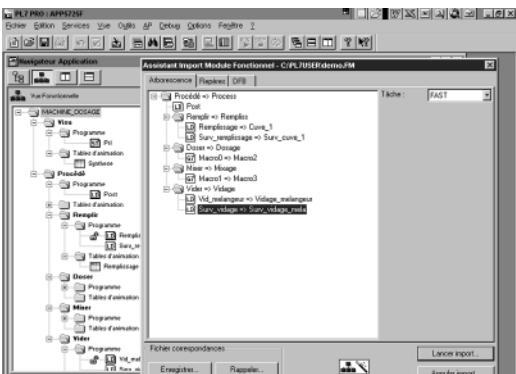


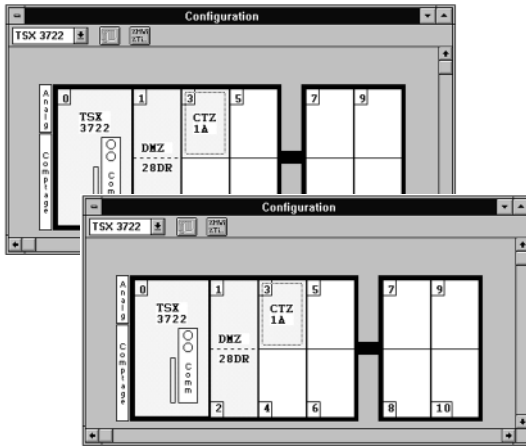
#### Exporting/importing functional modules

All or part of the tree structure can be exported into functional modules.

In this case, all program sections of the different module levels are exported.

During an import, an assistant can be used to reassign the data associated with the module in stages.





### Configuration editor

#### Hardware configuration

The configuration editor intuitively and graphically enables the declaration and configuration of the various components of the TSX Micro/Premium PLC application:

- Processor, coprocessor.
- Tasks.
- Application-specific I/O modules.
- Memory.
- ...

By clicking on an unconfigured position, the display of a dialog box shows the available I/O modules, classed according to family.

Once the various modules are positioned, selecting them accesses parameter entry for each module.

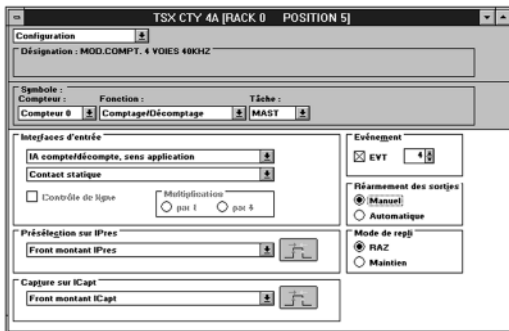
#### Software configuration

The configuration editor can also be used to set the software parameters of the application: choice of the number of constants, number of internal words and the number of each type of function block.

The configuration editor provides access to parameter entry for the function blocks. The copy/paste function for these parameters is available from version  $\geq$  V4.0 of the PL7 software onwards.

#### Configuration of Grafcet objects

When programming in Grafcet language, the configuration editor can define Grafcet objects (steps, macro-steps, etc.) and execution parameters (number of steps and active transitions).



#### Setup of application-specific functions

A number of tools are provided as standard for setting up the various applications: discrete I/O, analog I/O, counting, motion control (1), man-machine interface (MMI), communication, weighing (1), Warm Standby redundancy (2).

The parameter screens for the application-specific functions are accessed from the I/O configuration screen by clicking on the position in which the module has been defined.

The screens enable the main operating characteristics of the chosen application to be defined, for example:

- Filter values for discrete I/O.
- Voltage or current range for analog I/O.
- Threshold values for counting.
- Path of axes for position control.
- Calibration change during weighing.
- Transmission speed for communication.

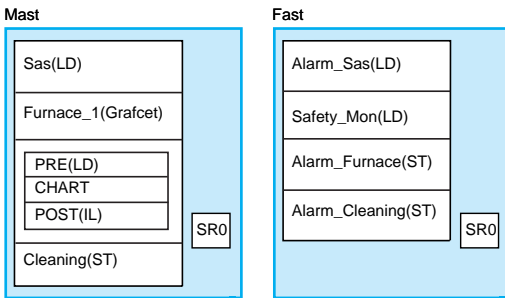
(1) PL7 Junior/Pro function available on a Premium platform.

(2) PL7 Junior/Pro function available on a Premium platform based on a TSX P57 353/453M processor (version with integrated Fipio link).

## Software structure

PL7 Micro/Junior/Pro software offers two types of structure:

- **Single task:** this is the simplified structure offered by default, where a single master task consisting of a main program, comprising several sections and subroutines, is executed.
- **Multitask:** this structure, which is better suited to high-performance real-time applications, consists of a master task, a fast task and event-triggered tasks, which have the highest priority. Master and fast tasks are divided into sections.



## Structured and modular programming

PL7 program tasks comprise several parts called sections and subroutines. Each section can be programmed in the appropriate language for the processing to be carried out.

Such division into sections enables a structured program to be created and program modules can easily be generated or added.

Subroutines can be called from any section of the task to which they belong or from other subroutines in the same task.

## Simple task software structure

There are two types of cyclic execution:

- **Normal cyclic execution.** This is the default option.
- **Periodic execution.** This type of execution, as well as the period, are selected by the user during configuration.

### Normal execution (cyclic)

At the end of each scan, the PLC system launches the execution of a new scan. The execution time of each scan is monitored by a software watchdog whose value is defined by the user.

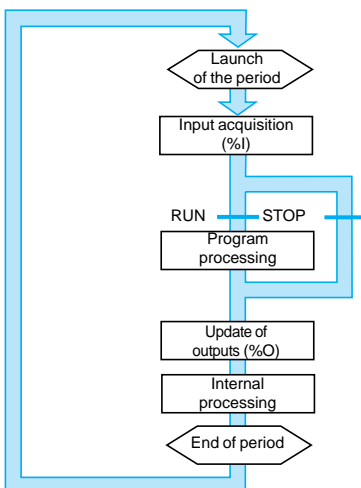
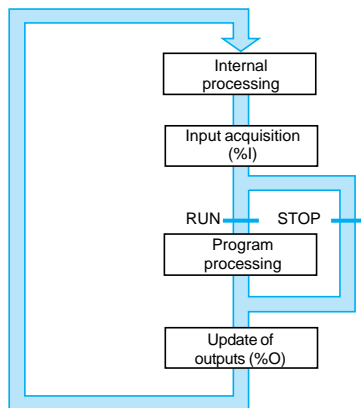
In the event of overrun, a fault occurs causing:

- The scan to stop immediately (STOP).
- A display on the front panel of the PLC.
- The alarm relay of the main rack power supply to be set to 0.

### Periodic execution

A new scan is executed at the end of each period. The execution time of the scan must be less than the time of the period defined (1 to 255 ms). In the event of overrun, the latter is stored in a system bit (%S19), which can be set to 0 by the user (by program or by the terminal).

A software watchdog which can be configured by the user monitors the scan time. In the event of overrun, an execution fault is signaled (see normal execution).



### Multitask software structure

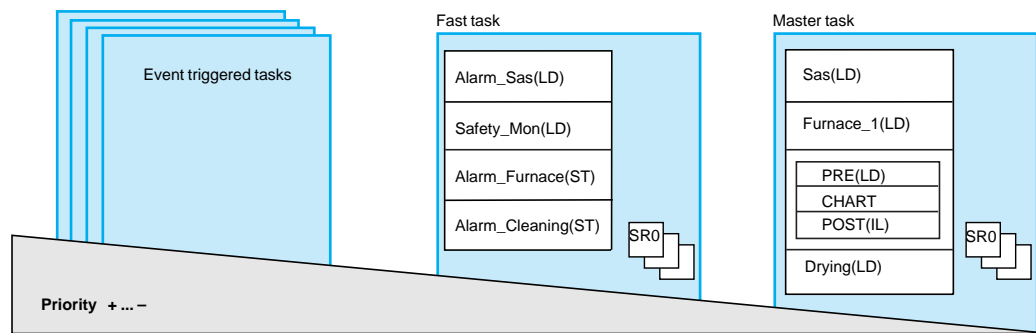
PL7 Micro/Junior/Pro software offers a Multitask software structure, consisting of:

- A master task (divided into sections, one of which may contain Grafcet).
- A fast task (divided into sections).

- One or more event-triggered tasks (only one section per task).

These tasks are independent and are executed in “parallel”, with the PLC processor managing the execution priority. When an event occurs, or at the start of the fast task cycle:

- The current execution of lower priority tasks is stopped.
- The event-triggered task or the fast task is executed.
- The interrupted task takes over again when processing of the priority task is completed.



This structure can optimize use of the processing power, and can be used to structure the application and simplify design and debugging, as it is possible to write and debug each task independently of the others.

### Master task

This compulsory task, which executes the main program, is periodic or cyclic (see single task structures). It is activated systematically. It is intended for sequential processing. Each section can be programmed in Ladder, Structured Text or Instruction List language. One section is dedicated to Grafcet language; when this language is chosen, 3 processing operations are proposed:

- Preliminary processing (PRE) is programmed in Ladder, Structured Text or Instruction List language and processes initializations on power return, operating mode modifications, input logic.
- Sequential processing (CHART) includes the graphic transcription and management of Grafcet charts. It provides access to processing of the actions and transition conditions.
- Post-processing (POST). This is programmed in Ladder, Structured Text or Instruction List language and is used to process all the instructions from the 2 preceding processing operations and the indirect safety functions specific to the outputs.

### Fast task

This task, which is higher priority than the master task, is periodic in order to leave time for execution of the lower priority task. Processing operations in this task must be as short as possible so as not to adversely affect the master task. It is useful when fast periodic changes in discrete inputs need to be monitored.

Each section of this task can be programmed in Ladder, Structured Text or Instruction List language.

### Event triggered tasks

Unlike the tasks described above, these tasks are not linked to a period. Their execution is triggered by an event occurring in an application-specific module (eg.: overrun of a counter threshold, change in state of a discrete input). These tasks have higher priority than all other tasks, and they are therefore suitable for processing operations requiring very short response times to the occurrence of an event.

They can be programmed in Ladder, Structured Text or Instruction List language.

#### Number of EVTi control events:

- TSX Micro PLCs: 8 events with TSX 37 10 and 16 events with TSX 37 21/22.
- Premium PLCs: 32 events with TSX 57 10 and 64 events with TSX 57 20/30/40 and PCX 57 20/30.

TSX Micro TSX 37 21/22 and Premium PLCs have 2 priority levels (EVT0 event has priority over other EVTi events).

**Ladder language (LD)**

**Program structure** (section, SR or event-triggered task)

Programs written in Ladder language consist of a series of rungs which are executed sequentially by the PLC. Each rung may be:

- Identified by a label.
- Completed by a comment of up to 222 characters.

A rung consists of 7 lines on TSX Micro and 16 lines on Premium, with 11 columns, allowing a maximum of 10 contacts and one coil per line.

**Program editor: Ladder language**

The Ladder language editor offers several tools for constructing rungs in a user-friendly way:

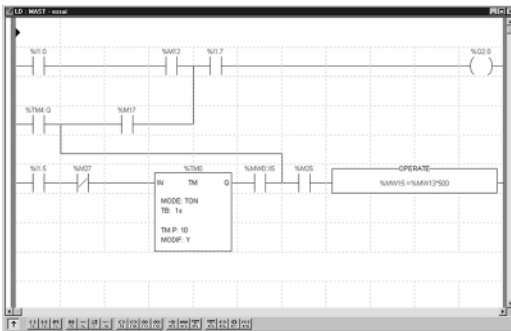
- A palette of graphic elements for direct access to the various graphic symbols of the language via the mouse or the keyboard: contacts, Boolean logic, coils, operation blocks, predefined function blocks...



- A rung can be drawn without having to fill in each element.
- The language objects can be entered and displayed in either symbol or address format.
- The symbol and address of each object can be displayed simultaneously.
- A rung is constructed simply by selecting the symbol from the graphic palette and placing it in the correct position in the grid on-screen.
- An automatic link line function optimizes the number of user actions.

The Ladder language editor is used to call up immediately the functions which assist data entry:

- Access to function libraries.
- Access to the variables editor.
- Cut, copy, paste.



**Structured Text language (ST)**

Structured Text language is a sophisticated algorithmic type language which is particularly suitable for programming complex arithmetic functions, table operations, message handling, etc.

**Program structure** (section, SR or event-triggered task)

Structured Text language enables direct transcription of a flowchart analysis and is organized into statements. Each statement consists of a label (1000 labels max), comments (256 characters max) and instructions.

There are four methods for controlling statements:

- Conditional action IF.
- Conditional iterative action WHILE (action repeated while a condition is true).
- Conditional iterative action REPEAT (action repeated until a condition is true).
- Repetitive action FOR (action repeated a certain number of times).

**Program editor: Structured Text language**

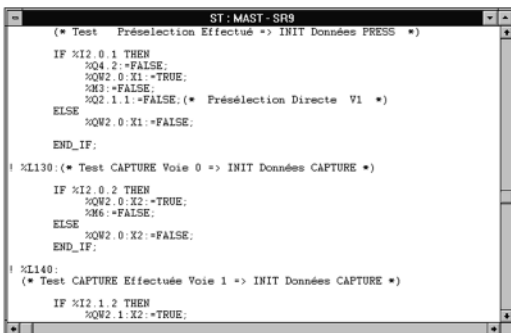
The editor enables statements to be entered one after another.

The editor provides help with entering:

- Modifications, insertion, etc.
- Cut, copy, paste.

Objects can be entered and displayed in either symbol or address format.

Different colors are used for the objects, language key words and program comments to make it easier to read.





TSX Micro PLC	Premium PLC
<ul style="list-style-type: none"> <li>■ 96 steps maximum (2) on 8 pages for TSX 37-10 and 128 steps for TSX 37-21/22</li> <li>■ 1024 transitions maximum (2)</li> <li>■ 11 elements maximum per divergence/convergence</li> </ul>	<ul style="list-style-type: none"> <li>■ maximum of 250 steps (2) on 8 pages</li> <li>■ 64 macro-steps of 250 steps.</li> <li>■ A "generic" comment can be associated with each macro-step</li> <li>■ 1024 transitions maximum (2)</li> <li>■ 11 elements maximum per divergence/convergence</li> <li>■ 1024 steps maximum in the application</li> </ul>

### Grafcet language (SFC)

Grafcet language is used to describe, in a simple and graphic manner, the sequential part of control systems. It corresponds to the SFC "sequential function chart" language described in standard IEC 61131-3.

#### Structure of the section in the master task

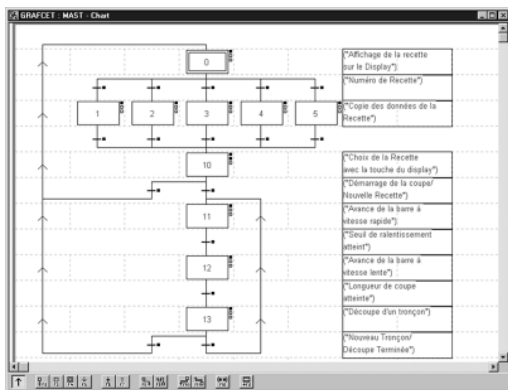
Grafcet SFC language is only used in one section of the master task. This is structured in three processing operations, see page 43100-EN/6.

Programs written in Grafcet SFC language consist of:

- Macro-steps (1) which are the only representation of a set of steps and a transition.
- Steps, with which the actions to be executed are associated.
- Transitions, with which the conditions are associated (transition conditions).
- Directed links, connecting the steps and transitions.

See characteristics page 43100-EN/14.

The actions (continual, pulsed on activation or deactivation) and transition conditions can be programmed in the desired language: Ladder, Structured Text or Instruction List.



#### Program editor: Grafcet SFC language

The editor offers 8 pages, each consisting of 11 columns and 14 lines, giving 154 cells per page.

A palette of graphic objects is used for direct access to each graphic symbol (macro-steps, steps, transitions, sequence selection, simultaneous activation/deactivation and connectors).

Programming of the transition conditions and actions is performed simply by clicking on the required chart element.

On a Grafcet page, comments of up to 64 characters can be entered in any cell. Functions which assist entry: cut, copy, paste, etc. are available to the user.

### Instruction List language (IL)

Instruction List language is a language representing, in the form of text, the equivalent of a Ladder diagram. It is used for writing Boolean equations and making use of the functions available in the language.

#### Program structure (section, SR or event-triggered tasks)

A program in Instruction List language comprises a sequence of instructions from the following different families:

- Bit instructions, for example read input n° 3: **LD %I1.3**.
  - Instructions on function blocks, for example start timer n° 0: **IN %TMO**.
  - Numerical instructions on single, double and floating point integers, for example, perform an addition: **[%MW10:= %MW50 + 100]**.
  - Instructions on word tables, character strings, for example, perform an assignment: **[%MW10:10:=%KW50:10]**.
  - Program instructions, for example, call subroutine n° 10: **SR10**.
- Each instruction is composed of an instruction code and a bit or word type operand.

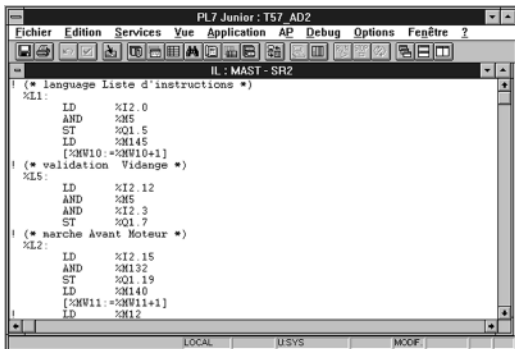
#### Program editor: Instruction List language

As in Ladder language, instructions are organized into sequences of instructions (equivalent to a rung). Each sequence of instructions can be identified by a label %Li, with i being from 0 to 999 and accompanied by a comment of 222 characters maximum.

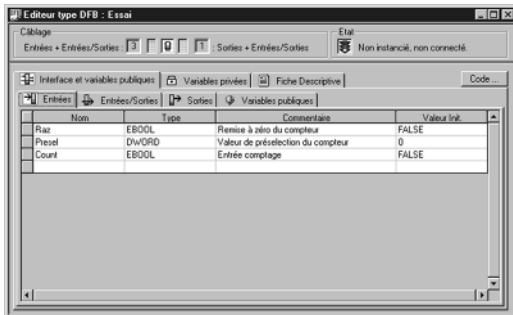
Each sequence of instructions is composed of one or more test instructions. The result of these instructions is applied to one or more action instructions.

Objects can be entered and displayed in either symbol or address format.

The editor provides help with entering data.



(1) With Premium PLCs only.



### Functions

#### User DFB function blocks

PL7 Pro software offers the user the possibility (for Premium PLCs) of creating their own function blocks which meet the particular needs of their applications. Once they have been created in the library, these function blocks can be used with PL7 Junior/Pro software.

These user function blocks enable an application to be structured. They are used as soon as a program sequence is repeated several times in the application or to freeze a standard type of programming. They can be exported to all other PL7 applications. Using a DFB function block in one or more applications enables:

- Simplification of program entry and design.
- Improved program readability.
- Easier debugging (all variables handled by the DFB function block are identified on its interface).
- Use of DFB-specific internal variables (independent of the application).

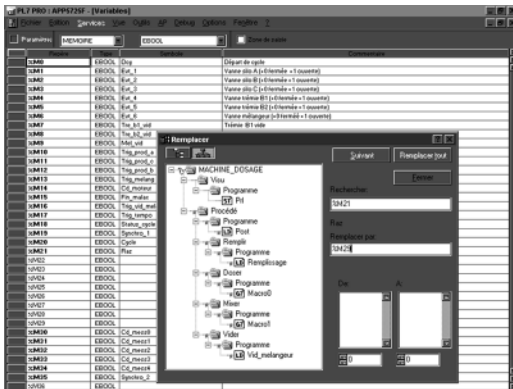
A DFB derived function block is set up in three phases:

- Design of the DFB which has a name, parameters (I/O), variables and code in Structured Text or ladder language.
- Creation of a DFB instance in the variables editor or when calling the function in the program editor.
- Using this instance in the program in the same way as a standard function block.

#### Main characteristics

<b>Inputs</b>	16 max. (1)
<b>Outputs</b>	16 max. (2)
<b>Inputs/outputs</b>	16 max. (1) (2)
<b>Public internal variables</b>	1000 (can be accessed via the application program)
<b>Private internal variables</b>	1000 (cannot be accessed via the application program)
<b>Comment</b>	322 characters max.
<b>Program sections</b>	One section in one of the 2 languages (ST, LD).

- (1) The maximum cumulative total of inputs and inputs/outputs is 16.  
 (2) The maximum cumulative total of outputs and inputs/outputs is 16.



#### Variables editor

The variables editor is used to:

- Symbolize the various application objects (bits, words, function blocks, I/O, ...).
- Define the parameters of the predefined function blocks (timers, counters, registers etc.).
- Enter the values of the constants and select the display base (decimal, binary, hexadecimal, floating point, message).

Each symbol (32 characters max, accented characters are permitted) can be accompanied by a comment (508 characters max).

Editing services are available in the editor:

- Find/replace an object in a part of the program or in a set of function modules (PL7 Pro).
- Find a character string in a list of symbols or comments.

Version ≥ V4.0 of PL7 offers enhanced functions due to:

- Copy/paste function for one or more symbol(s) and comments.
- Display in plain language of the overlap of different types of variable on a single memory address (for example, single and double format internal words, %MW0/%MD0).
- Highlighting of objects used by the application program.
- Opening the application variable database to third-party tools by importing/exporting text files (.txt). This new function makes it possible to create/modify application databases using a third-party software (for example TSX Microsoft Excel) that has extended edit functions.

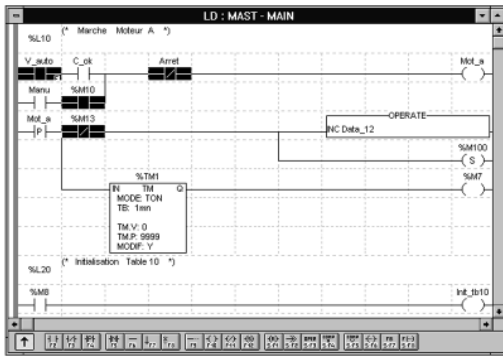
For each numerical variable, it is possible to select the display base (decimal, binary, hexadecimal, floating point, ASCII message).

Version ≥ V4.0 of PL7 offers new options for animation tables:

- Display of the comment associated with variables.
- Assignment of a single value to a number of variables.
- Change of display format for a number of variables.
- Display of the list of forced bits.







### Debugging tools

PL7 Micro/Junior/Pro software offers a complete set of tools for debugging applications. A tool palette provides direct access to the main functions:

- Setting stop points.
- Step by step program execution.
- Independent execution of the master (MAST), fast (FAST) and event-triggered (EVTi) tasks.



### Animation of program elements

Parts of the program are animated directly when the TSX Micro/Premium PLC is in RUN (rung, Structured Text statement or sequence of instructions in Instruction List language) by activating the PL7 animation function.

Animation is used to display the status of program variables, whatever the language used.

The animation can be frozen. Several windows can be displayed and animated simultaneously.

Modification	Référence	Symbole / Nom	Valeur courante	Nature
F3 Modifier	%M16	Trig_vid_melang	0	
	%M17	Trig_tempo	0	
F7 0	%M18	Status_cyclo	1	
	%M19	Synchro_1	0	
F8 1	%M20	Cycle	0	
	%M21	Raz	0	
	%M51	Trig_sect_remp	0	
	%M52	Trig_sect_vidan	1	
F4 Forcer 0	Gest_prod_silo_a	Gest_prod_silo_a	0	IN
F5 Forcer 1	Gest_prod_silo_a_Pressel	Gest_prod_silo_a_Pressel	2#0000 0000 0000 0000 00	IN
F6 DForcer	Gest_prod_silo_a_Couint	Gest_prod_silo_a_Couint	0	IN
	Gest_prod_silo_a_Done	Gest_prod_silo_a_Done	1	OUT
	Gest_prod_silo_a_Val_cou	Gest_prod_silo_a_Val_cou	16#00000007	PUBLIC

### Animation tables

Tables containing the application variables to be monitored or modified can be created by entering them or automatically initialized from the selected program part. Variables can then be modified, forced to 0 or to 1 for bit objects.

These tables can be stored in the application and therefore retrieved at a later date.

### Debugging the DFBs

■ Animation table: all public parameters and variables are displayed and animated in real time. It is possible to modify and force the desired objects.

■ As for the rest of the program, it is possible to use the following functions: breakpoint, step-by-step and program diagnostics.

### Grafcet debugging

In online mode, the browser gives a hierarchical view of the chart with CHART module and macro-step nesting. Animation is characterized by the presence or absence of indicator colors.

The Grafcet debug bar:

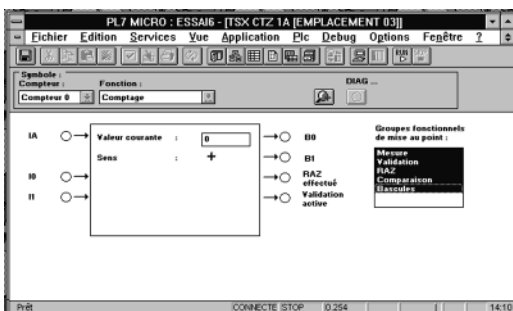
- Displays the state of the chart.
- Modifies the state of the chart.
- Gives information on the state of the Master task.

### Debugging the application-specific functions

The debug screens for the application-specific functions are accessed from the I/O configuration screen by clicking on the position in which the module has been defined, when the terminal is in online mode.

These screens are used for:

- Displaying and modifying the state of the I/O.
- Forcing the I/O.
- Displaying and modifying the current values.
- ...



### Diagnostics

The debug screens provide access to the general module or channel diagnostics.

These screens identify:

Internal module faults.

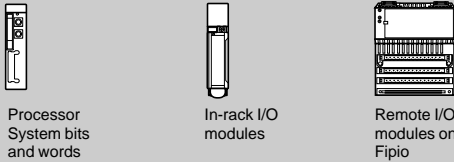
- External faults from the application.
- E.g.: range overrun fault for an analog module.

With version ≥ V4.0 of PL7 software, the Premium platform system diagnostics are extended. It is possible to monitor system bits and words as well as to display associated time-stamped messages automatically, without the need for additional programming. This monitoring applies to the system elements (processor, memory, tasks, ...), in-rack I/O and remote I/O on the Fipio bus.

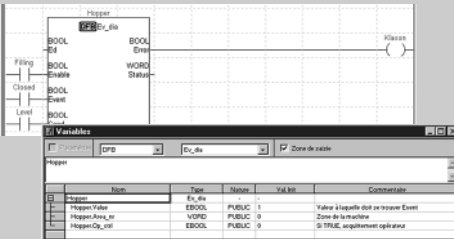
## Presentation

### Integrated diagnostics in Premium automation platforms

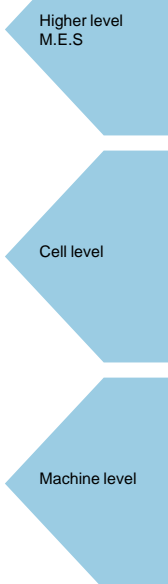
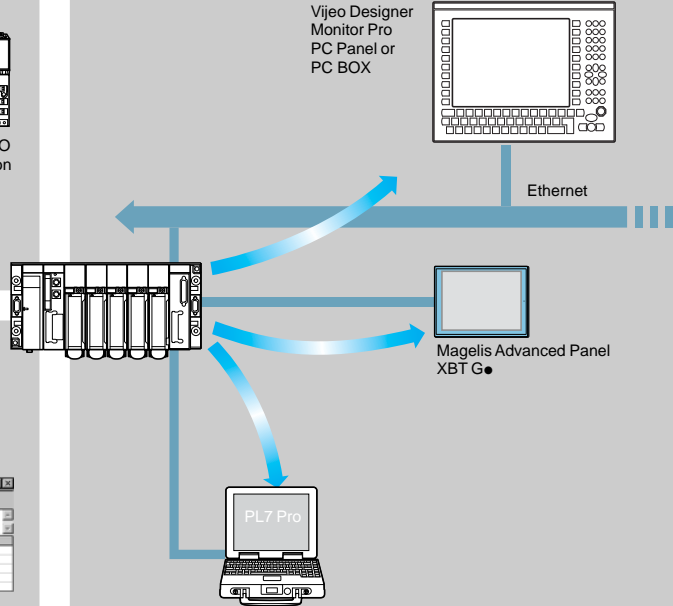
#### System diagnostics



#### Application diagnostics



#### Display viewers



The diagnostic offer of Premium platform is built on three elements:

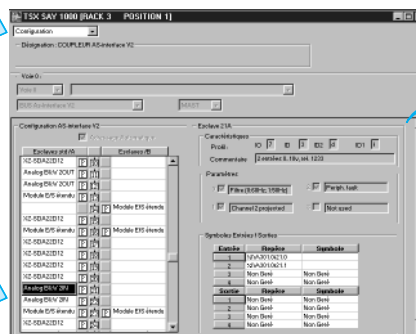
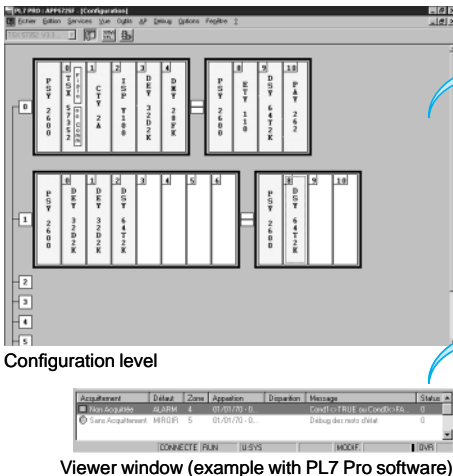
- System diagnostic.
- Diagnostic DFBs function blocks (system and application).
- Error message display system or viewers supplied as standard with Magelis XBT G● and Magelis iPC PC Panel/PC BOX, Vijeo Designer, Monitor Pro supervisory and PL7 Pro setup software

## Functions

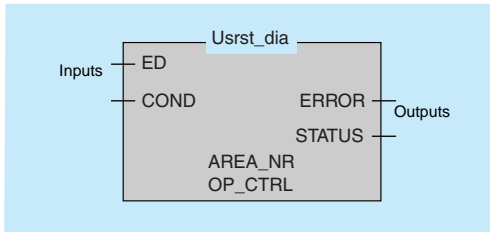
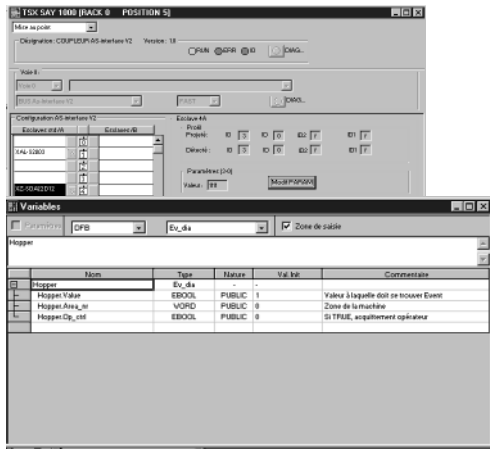
### System diagnostics

With version V4 of PL7 software, the Premium platform system diagnostics has been enhanced by the monitoring of system bits/words and I/O (in-rack and remote via the Fipio bus). All events automatically display time and date-stamped messages, without any additional programming, using one of the display units (1).

The diagnostic functions integrated in PL7 Junior/Pro software can be used for 1st level diagnostics of the configuration elements, and are effective for every module up to channel level.



(1) Diagnostics viewers are tools used to display and acknowledge error messages relating to diagnostics. They are supplied as standard with PL7 Pro, Vijeo Designer and Monitor Pro software, with Magelis terminals, and with the PLC web server which is accessible through a thin client Magelis PC Panel or PC BOX.



### Functions (continued)

#### Application diagnostics

PL7 Pro software has an integrated function block library containing monitoring function blocks known as Diagnostic DFBs. The Diagnostic DFB library comprises:

#### System diagnostic manufacturer blocks:

- IO\_DIA input/output fault. It can be used to monitor the state of the inputs/outputs.
- ASI\_DIA checks for the appearance of any errors on the AS-Interface bus (module or bus fault, slave missing, slave not configured or faulty).

#### Application diagnostic manufacturer blocks:

- EV\_DIA checks that an event (state of a bit) takes the correct value at the expected time (no notion of time).
- MV\_DIA checks the requested movement (change in the state of a bit) due to occur in a preset time period.
- ALRM\_DIA monitors the state of a bit (at 1 or 0).
- NEPO\_DIA and TEPO\_DIA can be used to monitor, control and diagnose the elements of the operating part created by wiring 2 preactuators and 2 sensors in conjunction.

#### Open diagnostic blocks:

These offer users the option of creating their own diagnostic function blocks to meet the specific requirements of their applications, thus complementing the manufacturer DFBs described above. They are created using 2 model blocks written in Ladder or Structured Text language.

The following can be created simultaneously in one application:

- a maximum of 26 types of system diagnostic DFBs
- a maximum of 26 types of application diagnostic DFBs.

#### Viewers

The diagnostic events processed by the Premium PLC using the diagnostic DFBs are stored in a buffer (memory space for data specific to the Premium PLC). The information contained in this PLC buffer is sent transparently for the user to the viewers and can be displayed automatically and used to handle errors and alarms. The viewer function is included as standard in:

- PL7 Pro programming software
- CCX 17 version > V 2.5 operator panels
- Magelis XBT-F graphic terminals
- Monitor Pro supervisor version V7.0.

The Premium platform has a multi-viewer option (link to a maximum of 15 viewers). A PC compatible station with the viewer function can be connected to several PLCs (link via X-Way communication to a maximum of 15 Premium platforms).

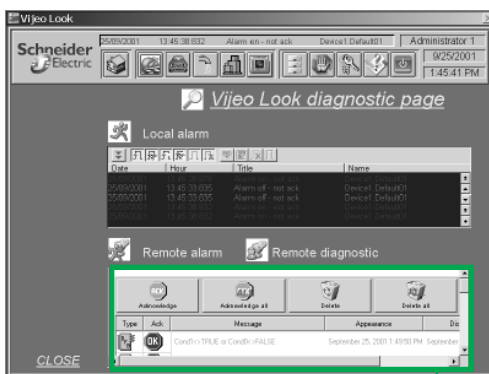
This buffer/viewer structure offers:

- A single point for managing faults per application
- Time and date-stamping at source of the appearance of faults
- Memorisation of transient faults
- Independence from the viewer functions. The frame transmitted from the PLC buffer is identical for all viewers.

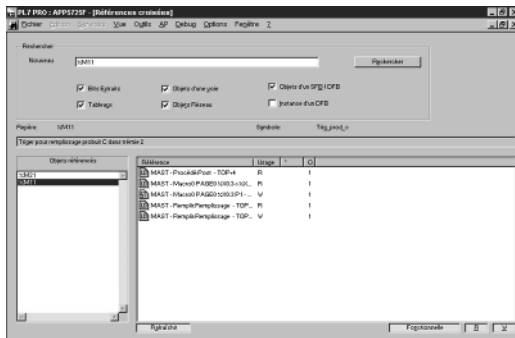
#### Composition of messages:

Each line displayed by the viewer represents a fault with, depending on the display capacity: state, type of DFB, geographical zone, dates and times of appearance/disappearance, associated message and status.

- 1 Viewer window on PL7 Pro screen.



1



### Variable cross-references

For every variable, this function can be used to:

- Search for program modules where this variable is used.
- Obtain the list of statements, rungs or expressions.
- Display and check activation conditions.

A log is used to keep track of this navigation.

Options relating to the variable, can be associated with the search (extract bit, table object, function block elements, network object, etc.).

This function can be initialized from the program or runtime screens.

### Application converters

PL7 Micro/Junior/Pro software includes application converters which make it possible to reuse in full or in part, applications already written in:

- PL7 2, applications for TSX 17, TSX 27 or TSX 47 10/20/25 PLCs.
- PL7 3 (1), applications for TSX/PMX 47...TSX/PMX 107 PLCs.
- ORPHEE (1), applications for April Series 1000 PLCs.

The converters offer the following utilities:

- Translation of language objects into the new PL7 syntax with retrieval of associated symbols and comments.
- Possibility of manually reassigning objects.
- Configuration check: the tool checks whether the configuration resources required by the program to be converted are compatible with the configuration of the destination application.
- Conversion (1) with generation of source files (Ladder, Structured Text or Grafset) in PL7 Junior/Pro format.
- Conversion ensures that instructions which are translated are functionally identical to the original program.
- A translation report gives a summary:
  - result of the conversion with the cause of non-translation where possible,
  - correspondence of variables in PL7 with original variables.

(1) Function or functionality requiring PL7 Junior/Pro software.

### X-Way communication drivers

The communication drivers (Uni-Telway COM port and USB port, PC side) are available in the PL7 software CD-Rom.

Depending on needs, another drivers can be installed from **TLX CD DRV20M** CD-Rom (to order separately). See following table.

Type of drivers	Windows XP Windows 2000	Windows NT	Windows 98 Millennium	Windows 95
Uni-Telway COM port	V1.8 IE19	V1.9 IE17	V1.7 IE18	V7.8 IE18
Uni-Telway TSX SCP 114	V1.1 IE04			
Modbus COM port	V1.5 IE25			
Fip ISA TSX FPC10 card	V1.4 IE06	V1.3 IE08	V1.4 IE06	V2.4 IE08
Fip TSX FPP 20 PCMCIA card	V1.2 IE03	V1.1 IE08	V1.2 IE04	
Ethway	V1.4 IE02	V1.1 IE03	V2.6 IE06	
ISAWay PCX 57, ISA card	V1.2 IE04	V1.5 IE06	V1.2 IE04	V1.2 IE09
PClway Atrium, PCI card	V1.1 IE09	-		
XIP X-Way on TCP/IP	V1.9 IE20			
USB for mini-DIN terminal port	PL7 included	-		
USB for USB terminal port	V1.0 IE14	-		

### References

Multilingual software packages (English, French, German, Spanish and Italian) for PC compatibles (1) equipped with Windows 98, Windows NT 4.0, Windows Millennium, Windows 2000 Professional or Windows XP operating systems.

■ **For one station**, these packages comprise:

- A CD-Rom supporting the PL7 multilingual software, the PL7 demonstration applications and the terminal link Uni-Telway driver,
- Two CD-Roms containing multilingual technical documentation,
- A CD-Rom containing the TSX Micro/Premium platform operating systems.

■ **For packages for 3 stations**, the above quantities are multiplied by three. the cordsets must be ordered separately, according to the required number of users.

The cordsets must be ordered separately, according to the required number of users and the type of connection (USB port or RS 232). See Separate elements page 43100-EN/17.

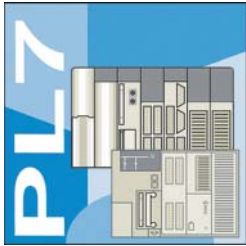


(1) Typical recommended configuration: Pentium processor, 266 MHz, 128 Mb of RAM memory, CD-Rom drive for installation of the PL7 program, VGA screen or above.



# Software

## PL7 software



### PL7 Micro software packages

PL7 Micro software enables programming in Instruction List, Ladder, Structured Text and Grafcet language. It can also be used to set up application-specific functions and perform maintenance and diagnostics of the developed applications. It includes the PL7 2 application converter.

Description	For PLC	Type of device and update	Reference	Weight kg
<b>PL7 Micro software packages</b>	TSX Micro	Single (1 station)	<b>TLX CD PL7M P45</b>	–
		Group (3 stations)	<b>TLX CD3 PL7M P45</b>	–
<b>PL7 Micro and SyCon V2.8 software package</b>	TSX Micro	Single (1 station)	<b>TLX CD PL7M PC45</b>	–
<b>Software updates for previous version of PL7 Micro</b>	TSX Micro	Single (1 station)	<b>TLX RCD PL7M P45</b>	–
		Group (3 stations)	<b>TLX RCD3 PL7M P45</b>	–
<b>Software update for previous version of PL7 Micro</b> supplied with SyCon V2.8	TSX Micro	Single (1 station)	<b>TLX RCD PL7M PC45M</b>	–

### PL7 Junior software packages

PL7 Junior software enables programming in Instruction List, Ladder, Structured Text and Grafcet languages. It can also be used to set up application-specific functions and perform maintenance and diagnostics of the developed applications. It includes the PL7 2, PL7 3 and ORPHEE application converters.

Description	For PLCs	Type	Reference	Weight kg
<b>PL7 Junior software packages</b>	TSX Micro, Premium, Atrium PCX	Single (1 station)	<b>TLX CD PL7J P45</b>	–
		Group (3 stations)	<b>TLX CD3 PL7J P45</b>	–
<b>Software updates for previous version of PL7 Junior</b>	TSX Micro, Premium, Atrium PCX	Single (1 station)	<b>TLX RCD PL7J P45M</b>	–
		Group (3 stations)	<b>TLX RCD3 PL7J P45M</b>	–
<b>Software upgrade packages from previous version of PL7 Micro</b>	TSX Micro, Premium, Atrium PCX	Single (1 station)	<b>TLX UCD PL7J P45M</b>	–
		Group (3 stations)	<b>TLX UCD3 PL7J P45M</b>	–



### Ensembles logiciels PL7 Pro

Functions on PL7 Pro software are identical to those on PL7 Junior software. It also offers the user the possibility of creating his own function blocks (DFBs) and graphic runtime screens.

Description	For PLCs	Type	Reference	Weight kg
<b>PL7 Pro software package</b>	TSX Micro, Premium, Atrium PCX	Single (1 station)	<b>TLX CD PL7P P45</b>	–
		Group (3 stations)	<b>TLX CD3 PL7P P45</b>	–
<b>Software updates for previous version of PL7 Pro</b>	TSX Micro, Premium, Atrium PCX	Single (1 station)	<b>TLX RCD PL7P P45M</b>	–
		Group (3 stations)	<b>TLX RCD3 PL7P P45M</b>	–
<b>Software upgrade packages from previous versions of PL7 Junior</b>	TSX Micro, Premium, Atrium PCX	Single (1 station)	<b>TLX UCD PL7P P45M</b>	–
		Group (3 stations)	<b>TLX UCD3 PL7P P45M</b>	–

### PL7 Micro/Pro software licenses

Description	For PLCs	Type	Reference	Weight kg
<b>PL7 TSX Micro Open Team software license</b>	TSX Micro	Team (10 stations) (1)	<b>TLX OT PL7M P45M</b>	–
<b>PL7 Pro Open Team software license</b>	TSX Micro, Premium, Atrium PCX	Team (10 stations) (1)	<b>TLX OT PL7P P45M</b>	–
<b>PL7 Pro Open Site software license</b>	TSX Micro, Premium, Atrium PCX	Site > 10 stations (1)	<b>TLX OS PL7P P45M</b>	–

### Separated elements

Description	Use	Reference	Weight kg
<b>X-Way drivers package for compatible PC</b>	Includes all X-Way drivers (voir page 43100-EN/15) Includes multilingual user documentation.	<b>TLX CD DRV20M</b>	–



TSX PCX 1031



TSX CUSB 485

Description	Used from processor	to PC port	Length	Reference	Weight kg
<b>Universal terminal port cable</b>	Mini-DIN port TSX Micro/Premium	RS 232D (15-way SUB-D connector)	2.5 m	<b>TSX PCX 1031</b>	0.170
		USB port (USB/RS 485 converter)	0.4 m	<b>TSX CUSB 485</b> (2)	0.144
		USB Port (mini-DIN/RJ45 cordset)	2.5 m	<b>TSX CRJMD 25</b> (2)	0.150

(1) Team user stations from the same geographical site.

(2) With **TSX CUSB 485** converter, use the **TSX CRJMD 25** cordset (equipped with 1 x mini-DIN and 1 x RJ45 connectors).