Note to users

This document refers to other software user manuals. See Appendix 2 in Divider E.

About this document

This document, reference TXT DM PL7 CMM V5●, describes the installation and use of PL7-COM V5 software installed under the X-TEL or MINI X-TEL Software Workshop.

PL7-COM V5 software is used for applications on TSX/PMX PLCs, versions V4 and/or V5. Under X-TEL, the TSX station containing the application must be selected before the PL7-COM software can be run. The PL7-COM software takes account of the model of the TSX "target" station and displays either the V5 screens and menus if the TSX/PMX selected is V5, or the V4 screens and menus if the TSX/PMX is V4.

Operating PL7-COM V5 software on a TSX/PMX station is the same as operating PL7-COM V4 software. The appendix in this document includes a summary of PL7-COM V4 software functions. In order to have a complete set of information relating to this software, the manual TXT DM PL7 CMM V42● should also be obtained.

Developments to PL7-COM version V5 compared to previous versions

The main developments to TXT L PL7 CMM V5E compared to TXT L PL7 CMM V42E are as follows :

Data exchanges between PL7-COM and XTEL-CONF

PL7-COM uses certain objects generated by the XTEL-CONF tool. These objects are :
- the type of processor
- the configuration of the rack modules
- the memory size of the cartridge
- the memory size reserved by XTEL-CONF.

The application structure (.APP) must be prepared with the XTEL-CONF tool before using PL7-COM software (see section 2.1 in divider A).

Uniqueness of file names

Only one configuration file is generated by PL7-COM and it contains the entire configuration for the communication module(s) in the TSX/PMX.
The configuration file name is COMM.BIN.

Simplifying the generation phase (STATION.APP)

It is no longer necessary in version V5 to generate the STATION.APP file after modifying a "BIN" file.

Terminology used

The names of certain function keys have been modified between version V4 and version V5. These modifications are :

- In connected mode

  [STORE] (V4) is now **[STA → DSK]**, used to transfer a TSX memory communication application to the COMM.BIN file. This transfer is executed using the TRANSFER tool.

  [RETRIEVE] (V4) is now **[DSK → STA]**, used to transfer the COMM.BIN file to the TSX memory. This transfer is executed using the TRANSFER tool.

- In local mode

  [•BIN] (V4) is now **[RETRIEVE]**, used to transfer any application file name from the hard disk to the COMM.BIN file in the X-TEL database.

  [STORE] (V4) remains as **[STORE]** and is used to transfer the COMM.BIN file to any application file name to the hard disk or to a diskette.

It is also possible to modify the configuration of the application I/O. To do this, the XTEL-CONF tool must be used, by pressing the **[XTELCONF]** soft key.

Connection to the FIPIO fieldbus

PL7-COM software can be used on an FTX 417/507 workstation connected to the FIPIO distributed I/O bus. In this case, the workstation uses the reserved connection point 63.

Compatibility between PL7-COM V4 and PL7-COM V5 applications

Any V4 application can be transformed into a V5 application, provided the following operations are performed :

① Import the V4 application under X-TEL V5, using **Save/Restore** or **Copy/Paste**.

② Create the initial window for a V5 station.

③ Start the **Import** function from the COMM icon in the initial window for a V5 station, and import the following files :

V4 station\COMM\APPLI\xxx.BIN (essential) : binary application,

to the V5 station directory \COMM\APPLI,

then

V4 station\COMM\MOD\xxx.C07 (optional),

to the V5 station directory \COMM\MOD.

④ Run PL7-COM in the V5 station and perform the following operations :

- select the **TSX/PMX** file under the item **local/working memory** (depending on the function) to display the RETRIEVE command
- activate the **[RETRIEVE]** command which provides access to the list of xxx.BIN station files
- activate the **[DIR BIN]** command and select the previously imported xxx.BIN file
- **<ENTER><ENTER>**, restores the xxx.BIN file under the V5 station.

⑤ Quit the PL7-COM function.

⑥ Launch the XTEL-CONF tool

- activate the **Generate and** the **enter application parameters** menu
- quit the XTEL-CONF tool.

⑦ Launch the PL7-3 function and activate the **[V5 CONF]** command to assign the new configuration defined under XTEL-CONF to the application program.

Section **Page**

1.1 General

1.1-1 Functions available with PL7-COM

PL7-COM software, reference TXT L PL7 CMM V5E, is a help program for programming and setting-up communication applications.

PL7-COM software comprises :

- a package for setting-up the TSX SCM 20/21/22 modules,
  - set-up (configuration, etc),
  - loading OFBs,
  - OFB diagnostics.
- a package for operating the TSX SCM modules,
  - communication OFB.

- **Functions associated with the communication modules**
  - assistance with entering configuration parameters using menus and on-line documentation,
  - assistance with diagnostics and debugging,
  - transferring the configuration between the PLC memory, the module memory and the disk,
  - documenting the configuration,
  - archiving the configuration to disk,
  - printing the configuration.

1.2 Configuration required for PL7-COM

To install PL7-COM requires an FTX 417/507 terminal or an IBM PS/2 microcomputer or compatible PC with :

- OS/2 operating system, version 1.3 or 2.1.
- The MINI X-TEL or X-TEL Software Workshop, reference TXT L BASE V5● or TXT L BJR V5●.
- PL7-3 software, reference TXT L PL7 3 V5●, TXT L PL7 3D V5● or TXT L PL7 3T V5●.

A minimum of 4 Mb of RAM memory and a 40 Mb hard disk.

Important

Telemecanique cannot guarantee correct operation of this software on all microcomputers or compatible PCs on the market with the above-mentioned characteristics.

1.3 Checking the hardware

The TXT L PL7 CMM V5 software package comprises :

- a 3" 1/2 diskette, reference TXT LF PL7 CMM V5,
- a 3" 1/2 diskette, reference TXT LF FB CMM V5,
- a software protection key,
- a licence agreement,
- this manual, reference TXT DM PL7 CMM V5.

To use PL7-COM, the following hardware should be used :

- An FTX 417/507 terminal or an IBM PS/2 microcomputer or compatible PC (see required configuration section 1.2).
- A terminal/PLC connection cable for an FTX 417/507 terminal.
- A terminal/PLC connection kit for an IBM PS/2 microcomputer or compatible PC, comprising :
  - an RS 232C/current loop converter,
  - converter/microcomputer connection cable with a 9-pin connector,
  - converter/microcomputer connection cable with a 25-pin connector,
  - converter/PLC connection cable,
  - a TSX SCC 02 software key support.

1.4 Connections

All connections specific to the terminal (monitor, keyboard, mouse, printer, software key support, etc) are assumed to be in place, this section only describes fitting the software key. To do this, place the key in the empty slot in the key support.

This operation must be carried out with the equipment switched off.

Note

This software key contains the access rights needed to access PL7-COM. The Key Manager tool, supplied with each Software Workshop, allows these rights to be transferred to the working key so that all rights are grouped on one key (the working key) so as to free a slot on the key support.

For further details about this tool, refer to the X-TEL or MINI X-TEL database manual.

The PL7-COM V5 software key is identical to that of PL7-COM V4.

1.5 Installing the software

1.5-1 Preliminary operations

Before installing PL7-COM on the hard disk it is advisable to :

- Read the licence agreement and guarantee concerning copying restrictions and installation of the software.
- Make a duplicate of the diskette required for installation to avoid any accidental damage to the original diskette and work only with the copy.

Important

The PL7-COM program disks are supplied in the write-locked position. Do not alter the position of the locking tabs.

1.5-2 Installation procedure

The following operations must be performed prior to installing PL7-COM :

- Check that the MINI X-TEL or X-TEL V5 Software Workshop is already installed :
  - if so, install PL7-COM according to the procedure described below,
  - otherwise, first install the MINI X-TEL or X-TEL Software Workshop (refer to the manual of the database concerned).
- Close all the current sessions ; to do this :
  - open the Task Manager window,
  - pull down the Shutdown menu and select the "Shutdown now" item,
  - reply to the questions and then Cancel.

Installing PL7-COM software

- Open an OS/2 full-screen session; to do this :
  - open the Start Programs window,
  - pull down the Group menu and select the Main Group item,
  - select the OS/2 full screen sessions item. The prompt [C:\] is displayed on the screen.
- Insert the TXT LF PL7 CMM V5 diskette in the drive.
- Enter the drive identifier (a: or b:), then confirm with <Enter>.
- From the new prompt (for example [A:\] or [B:\]), type **Install** then confirm with <Enter>.
- Follow the procedure displayed on the screen.
- When installation is complete, replace the diskette with the second diskette (reference TXT LF FB CMM V5).
- Type the **Install** command then confirm with <Enter>.
- Follow the procedure displayed on the screen.
- When the installation is complete and if it is the last one, check the configuration. Confirm with <Enter>.
- Remove the diskette from the drive and return to the Software Workshop using the <Ctrl><Esc> command.

1.6 Using the keyboard and the mouse

Using the keyboard

To use PL7-COM, Telemecanique recommends a 102 key QWERTY keyboard.

Certain PL7-3 function keys (CLEAR, ZOOM, QUIT, etc), which are also used by PL7-COM, are not printed as standard on the keyboard, but are accessed by another key or combination of keys.

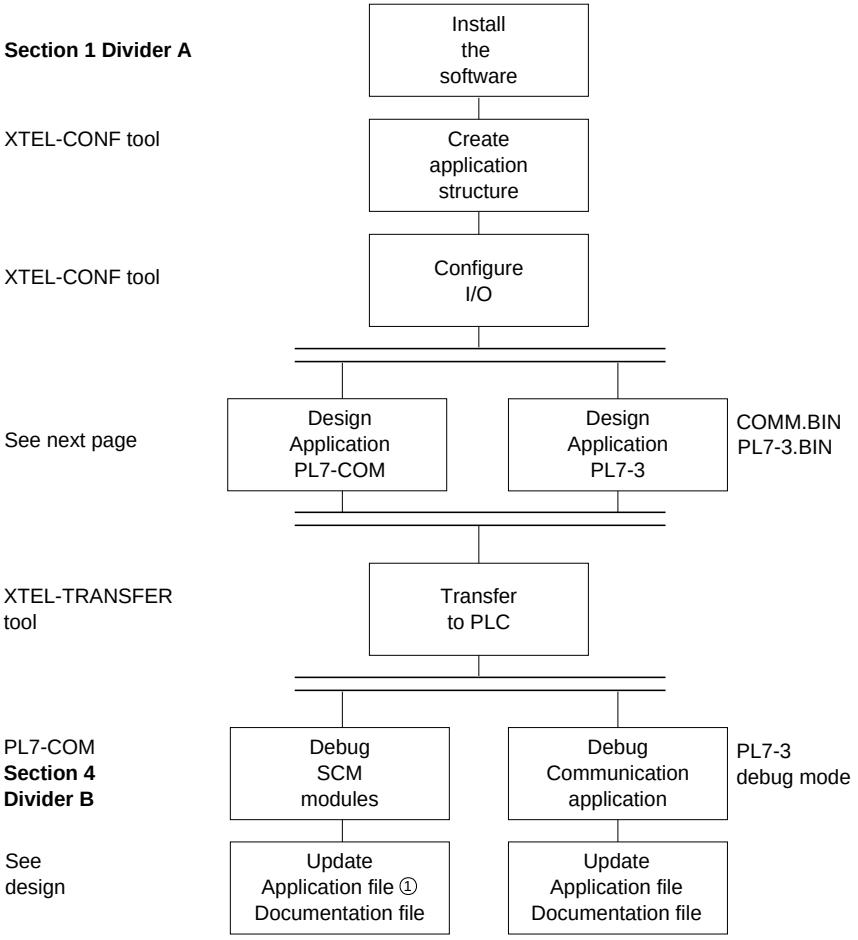
These keys, common to several programs, are described in the PL7-3 Operating modes manual, section 3.1 in divider A.

Using the mouse

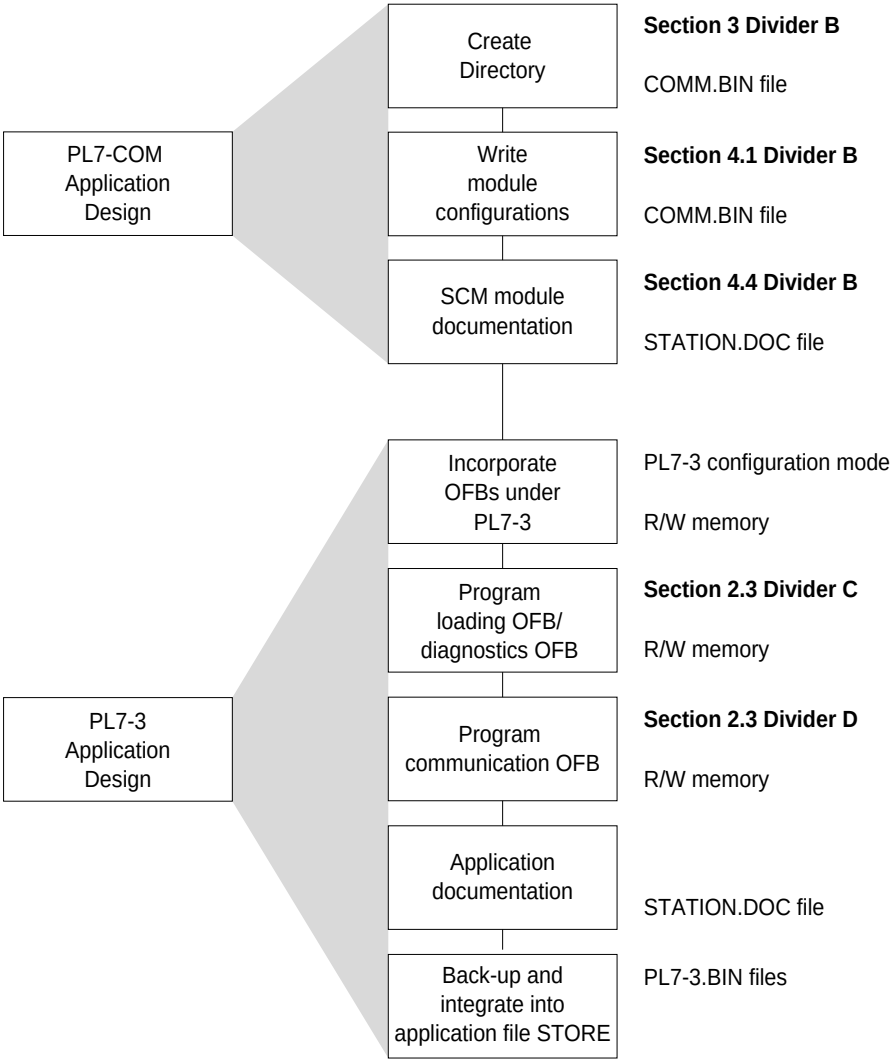
As for the keyboard, detailed use of the mouse is described in the PL7-3 Operating modes manual, section 3.2 in divider A.

2.1 Methodology for installing a communication application on a TSX/PMX programmable controller

The following methodology is intended as a guide to the user when creating, debugging, archiving and documenting a communication application. This methodology refers to each operation without going into detail about the operations required.



① It is not necessary to generate the application file after modifying a .BIN file.


PL7-COM Application Design
PL7-3 Application Design
Create Directory
Write module configurations
SCM module documentation
Incorporate OFBs under PL7-3
Program loading OFB/ diagnostics OFB
Program communication OFB
Application documentation
Back-up and integrate into application file STORE
Section 3 Divider B
COMM.BIN file
Section 4.1 Divider B
COMM.BIN file
Section 4.4 Divider B
STATION.DOC file
PL7-3 configuration mode
R/W memory
Section 2.3 Divider C
R/W memory
Section 2.3 Divider D
R/W memory
STATION.DOC file
PL7-3.BIN files

1.1 Accessing the configuration software

Configuration software for TSX SCM xxx communication modules is accessed by opening the corresponding COMM window. To do this :

① Open the Start programs window by double clicking on the corresponding item

② Pull down the Group menu and activate the Telemecanique item

③ Open the User window by double clicking on the XTEL item

④ Enter the user parameters (name and password) and then confirm to open the Volumes window

⑤ Open a volume by double clicking on the icon of the volume to be opened

⑥ Open a project by double clicking on the icon of the project to be opened

⑦ Open a station by double clicking on the icon of the station to be opened

⑧ Open the COMM function by double clicking on the corresponding icon. If this icon is not displayed in the secondary Functions window even though the software has been installed, this indicates that the function has not yet been defined. To do this :
- pull down the Define menu and activate the New item
- click on COMM then on OK.

⑨ For greater ease, open the COMM full screen window by clicking on the "arrow up" button of the window.

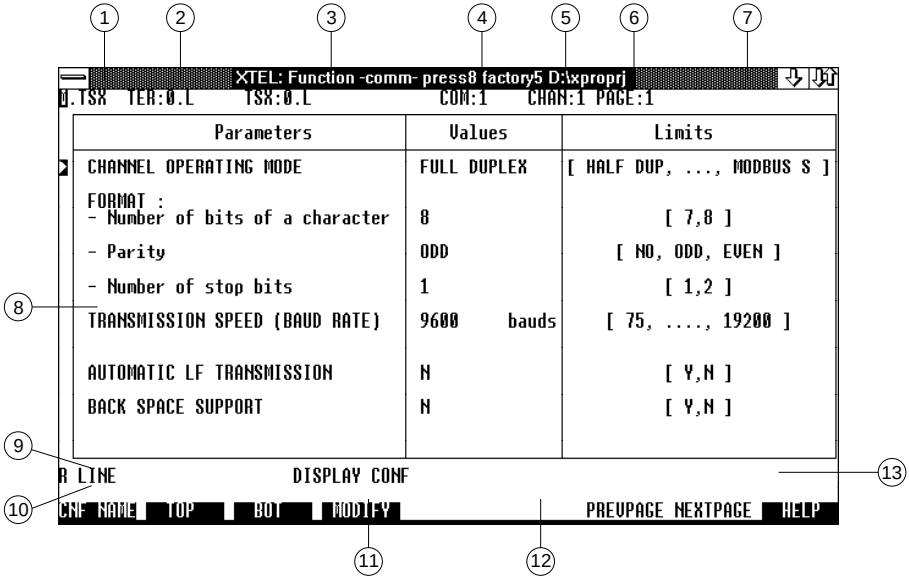
Notes

- If a PL7-COM session is already open (the corresponding icon appears on the screen outside the secondary Functions window), double click on this icon to open the corresponding window.
- To close a session, click on the corresponding icon to pull down a menu. Then click on the Shutdown/Close command.

1.2 Introduction to the display screen

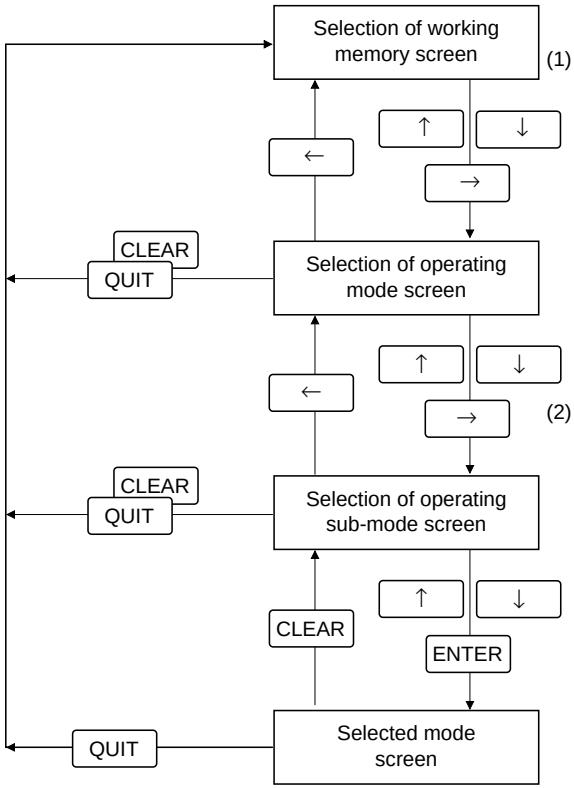
The window which displays the PL7-COM screens is known as the display screen. All items specific to the X-TEL Software Workshop (icons, window title, window commands, etc) are described in the Software Workshop manual.

Information displayed



① Working memory,
② Network address for the terminal,
③ Working memory and its address if COM file MEM or TSX MEM,
④ COM number or file name if COM file or TSX file working memory,
⑤ Channel number of module,
⑥ Current page number,
⑦ Name of application (only in TSX MEM, TSX file or COM file),
⑧ Display zone for configuration parameters,
⑨ Real time event zone, indicating PLC status,
⑩ Parameter entry line,
Zone indicating current operation (DISPLAY, MODIF, etc),
⑫ F1 to F9 soft key display line,
Message zone for syntax or manipulation errors.

Screen sequencing



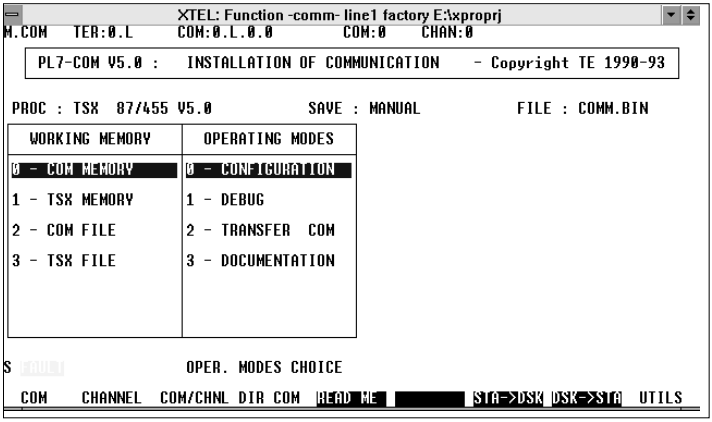
(1) To access COM MEMORY or TSX OR PMX MEMORY in connected mode requires :

- that a configuration memory file has previously been transferred to the PLC memory,
- that the XTEL-CONF I/O configuration declares at least one TSX SCM module or that the processor has an integral UNI-TELWAY port (SCI).

(2) only in DEBUG and TRANSFER modes.

1.3 Selecting modes

The choice of modes screen, the basic PL7-COM screen, provides access to all the functions available with this software.



This screen has two parts :

- a menu zone for selecting :
  - the working memory (module, PLC or disk),
  - the operating mode (configuration, debug, transfer and documentation),
  - an operating sub-mode for the debug mode or transfer mode.
- an information zone which indicates (in connected operation) :
  - the type of processor and its version,
  - the associated file name and type of store.

Role of the function keys

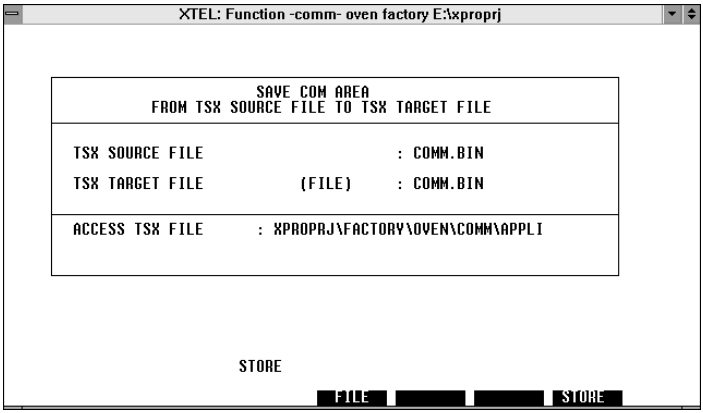
< ↑ > < ↓ >	used to move the cursor in the active column : working memory, operating modes or operating sub-modes. An item in a column can also be selected by entering its number.
< → > < ← >	used to move from one column to another.
<Enter>	confirms the selections made.

Role of the soft keys

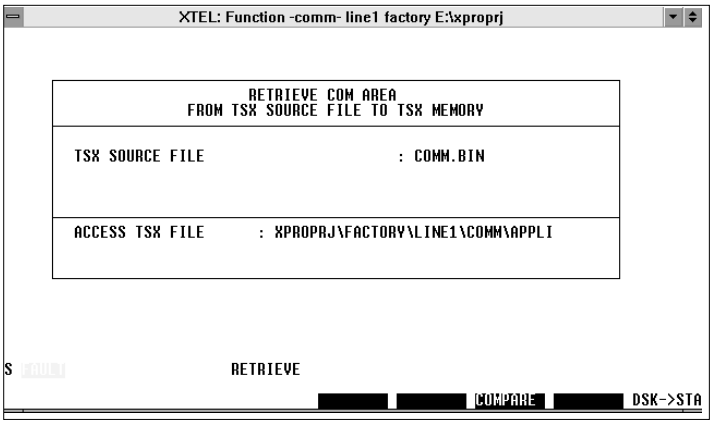
[EXIT] causes PL7-COM to quit with the possibility of saving and comparing.

[READ ME] provides access to on-line documentation.

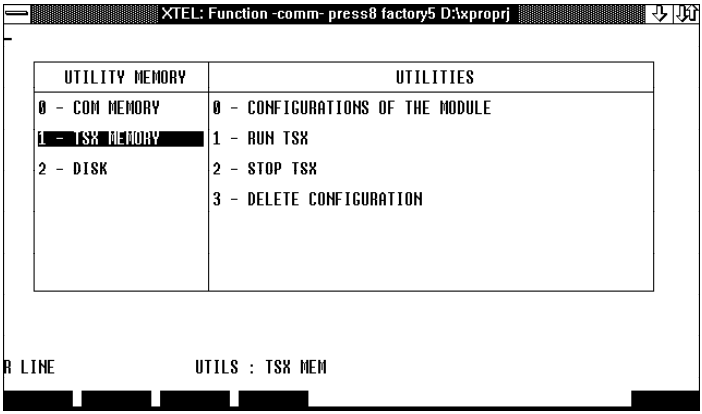
[STA→DSK] accesses a store function in the COM zone of the TSX memory in the COMM.BIN file.



[DSK→STA] enables the COMM.BIN file to be retrieved from the COM zone to the TSX memory.



[UTILS] accesses the utilities functions.

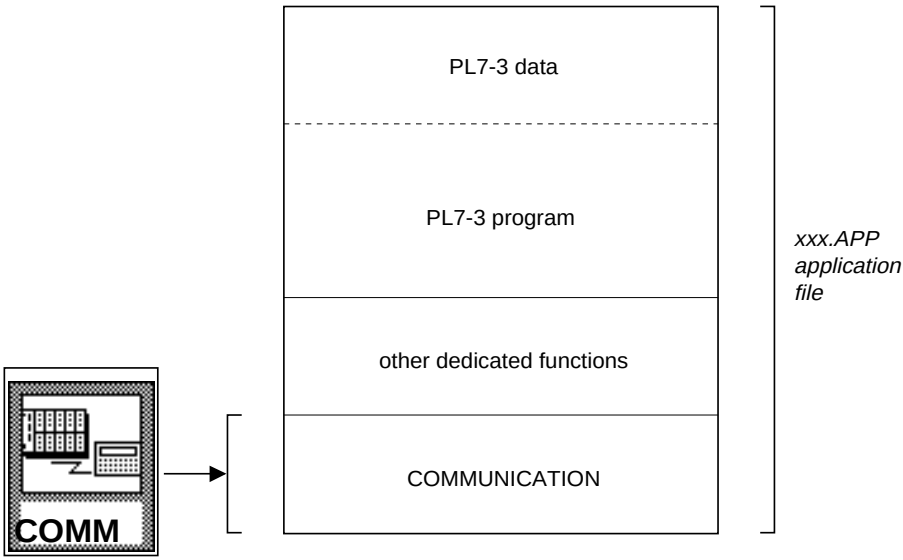


Other soft keys, specific to the selected mode, are described in section 2, selecting working memory.

1.4 Relationship with PLC memory

1.4-1 Dedicated communication zone in the PLC memory

If the COMM function is declared for a station, a dedicated communication zone is automatically created by the XTEL-MEM tool on creation of the xxx.APP file. The size of this zone, fixed by default by the XTEL-MEM tool, may be modified by the user. The position of this zone is determined by the size of the PL7-3 and other dedicated zones which it follows.



Contents of the communication zone

When the PLC memory image is created, the XTEL-CONF tool creates an empty zone. This can then be filled by PL7-COM (1). It comprises :

- the directory, consisting of :
  - a correspondence table between the logic numbers (0 to 63) and the physical positions of the modules in the racks. The program offers default assignments (2) which may be modified,
  - a table which gives the start address and the size of the configurations stored in the dedicated zone,
- the configurations, in ascending logic number order.

(1) *Providing the PLC memory image or the PL7-3 application contains at least the I/O configuration and the slots occupied by the TSX SCM modules.*

(2) *Ascending numbering from 0 to 63 in the order of the modules in the PL7-3 configuration.*

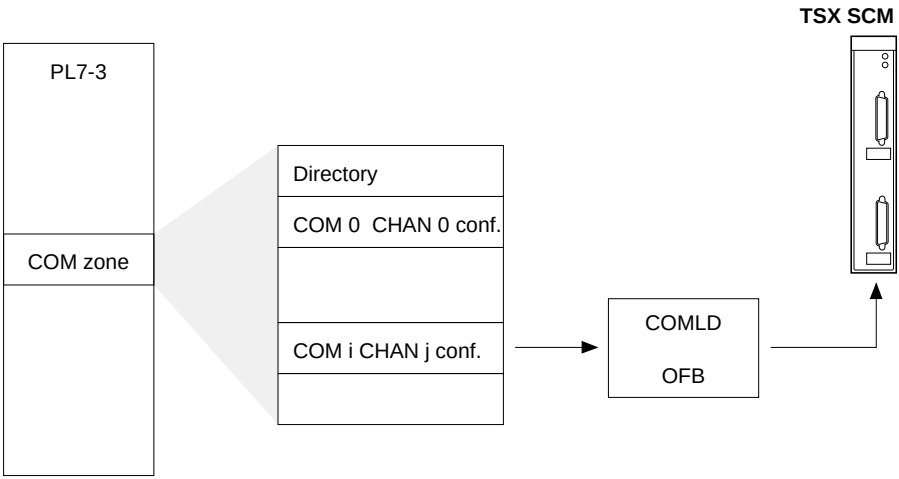
Dedicated COM zone

Correspondence table between the configuration n° and module location
Address and size of stored configurations
COM0 CHAN 0 configuration
COM0 CHAN 1 configuration
COM1 CHAN 0 configuration
COMn CHAN 0 configuration
COMn CHAN 1 configuration

Directory

This zone contains the information which can be accessed by the PL7-COM functions concerning its organization and by the communication OFB blocks. A compacting function is used to optimize the contents. A copy of this dedicated COM zone is stored in the xxx.BIN file under the COMM\APPLI directory on the hard disk (or diskette).

A configuration stored in this zone can be transferred to the TSX SCM xxx by the COM OFB.



The COMLD OFB is described in section 2, divider C.

1.4-2 Reservation while operating in connected mode

Any FTX 417/507 terminal or microcomputer can be physically connected to any TSX PLC station on the same MAPWAY/ETHWAY/FIPWAY/ETHERNET network. Because of this, several terminals can request to be logically connected to the same PLC station.

In order to avoid access or procedural conflicts, each terminal must request reservation of the entire dedicated COM zone. This reservation can only take place while reading from or writing to the directory or a configuration.

If the dedicated COM zone is not already reserved, the requester can then access this zone.
From this moment, any attempt by another terminal to access is refused and the message TSX ALREADY RESERVED appears. This reservation is cancelled when work has been completed.

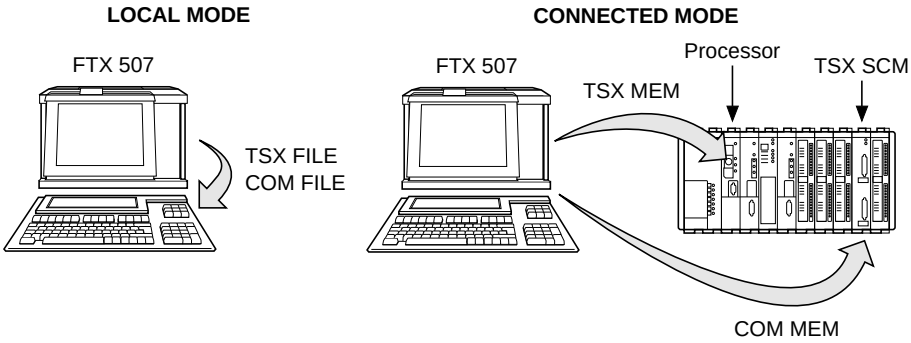
Caution

PL7-COM cannot be used to set up a remote station over a TELWAY network.

1.5 Methodology

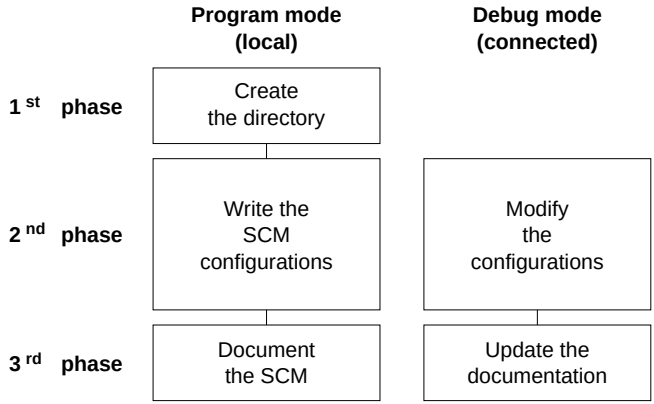
The SCM configuration software can be used :

- in local mode, working on the disk,
- in connected mode, working on the module memory (COM MEM) or the PLC memory (TSX/PMX MEM).



The use of Local mode is recommended for creating SCM configurations and the dedicated COM zone. Although there is nothing to stop a complete application being created in connected mode, it should only be used for modifications, corrections and debugging.

Implementation is in three phases :



B

1st phase : Creating the directory

- Open the COMM window.
- Select the TSX File memory.
- Select DIR COM (the software automatically recognizes STATION.APP files).
  Quit by pressing ENTER.

2nd phase : Creating the configuration

- In Program mode : select the TSX File (Local)
  - select the COM and COM CHNL (channel)
  - create the configurations (CONFIGURATION)
- In Debug mode : select COM-MEM (Connected)
  - modify the configuration
  - update TSX MEM using the Transfer function
    (COM.MEM TRANSFER TSX MEM)

3rd phase : Documentation

- In Program mode : select the TSX File
  - document each configuration (COM*CHNL*)
    output to printer or to STATION.DOC file (XTEL-DOC))
- In Debug mode : select TSX-MEM

Note

In TSX File mode, PL7-COM works directly on the COMM.BIN file. It is not necessary to store.

2.1 Introduction

B

The choice of working memory defines the PL7-COM operating mode : local or connected.

Local mode operation

In this case, the hard disk is selected as the working memory.
In local mode the user can :

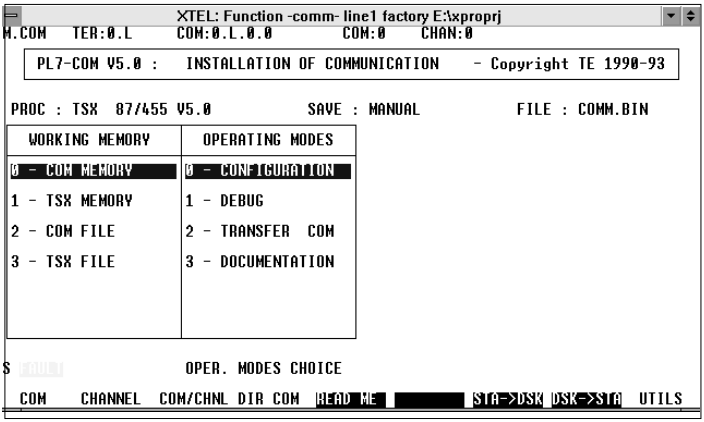
- define the configurations for each channel of the modules (COM file). Configurations created in this way are not associated with any module,
- generate the COMM.BIN file, image of the dedicated COM zone (TSX/PMX file).

Connected mode operation

In this case the COM memory (module memory) or TSX/PMX memory (PLC dedicated zone) is chosen as the working memory. In connected mode the user can :

- generate or modify a configuration,
- generate the COMM.BIN file, image of the dedicated COM zone,
- transfer configurations from the disk to the module or to the dedicated zone in the PLC memory.

When the terminal is connected to a TSX SCM module, PL7-COM can also be used for debugging.



2.1-1 Role of the common soft keys

Details of the soft keys common to the different modes are given below :

[COM] selects the number of the working module. In documentation mode, the "*" character confirms all the SCM modules which are configured,

[CHANNEL] selects the number of the working channel. In documentation mode, the "*" character confirms all the channels of the selected module. COM = * and CHANNEL = * enable documentation of all the channels stored in the TSX or PMX memory or in the TSX file,

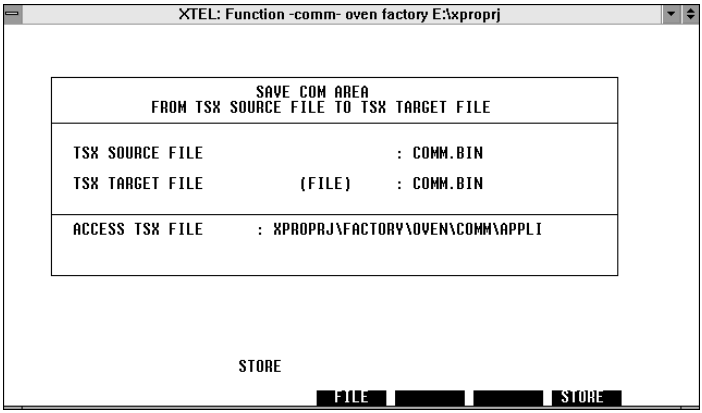
[COM/CHNL] selects the number of the module and of the working channel,

[DIR COM] provides access to the COM directory screen (see section 3.2 in divider B),

[READ ME] provides access to the PL7-COM help screens,

[R/S TSX] or **[R/S PMX]**
sets the PLC to RUN or to STOP.

[STA→DSK] displays a screen which allows the contents of the dedicated COM zone to be stored to disk, as a COMM.BIN file in the COMM\APPLI sub-directory :

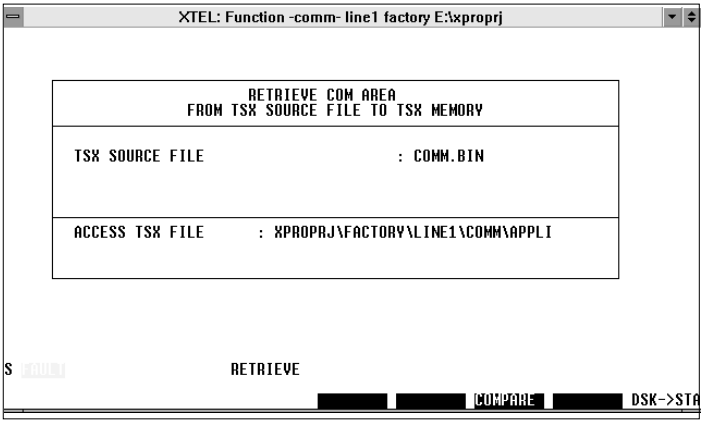


[AUTO/MAN] allows the type of store operation to be selected in connected mode. In automatic mode, all modifications are systematically stored. In manual mode, modifications are not stored unless the [STORE] key is pressed.

[COMPARE] starts the comparison between the source files and the target files.

[STA→DSK] stores a file and, after confirmation, starts the XTEL-CONF tool which updates the STATION.APP file corresponding to the configuration.

[DSK→STA] displays a screen which allows the contents of a COMM.BIN file, previously stored to disk, to be transferred to the dedicated COM zone of the PLC memory :



[COMPARE] starts the comparison between the source file and the dedicated COM zone of the PLC.

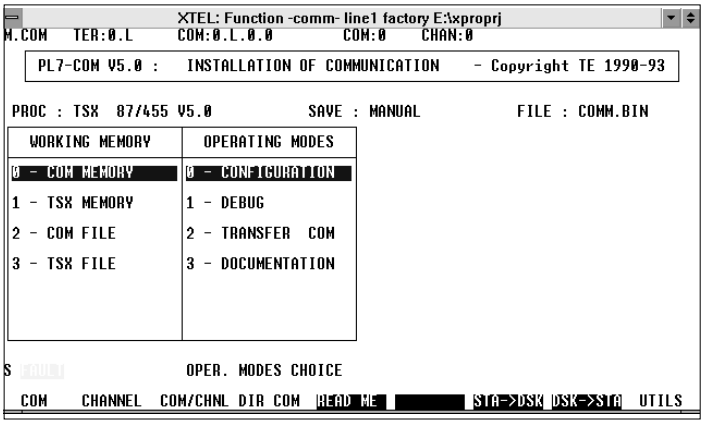
[DSK→STA] retrieves the selected COMM.BIN files to the dedicated COM zone in the PLC memory.

2.2 Selecting COM MEMORY

The COM memory is the only one which can be used for debugging and operating modules.

The configuration is stored directly in the module memory on each confirmation.

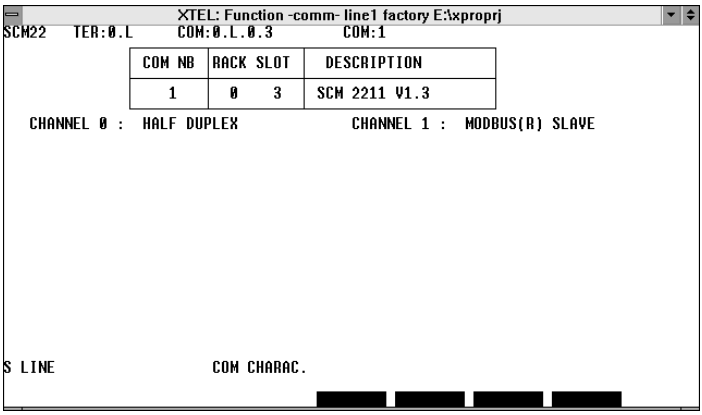
The COM memory can only be used if a STATION.APP configuration file, containing at least the I/O configuration performed under XTEL-CONF, has previously been transferred to the PLC memory. The PLC can be in STOP or in RUN.



[UTILS] accesses the utilities functions associated with the COM memory :

0 - MODULE CHARACTERISTICS : displays information relating to the module and to its configuration,

1 - DELETE CONFIGURATION : allows the configuration of the working channel to be cleared. This request must be confirmed.

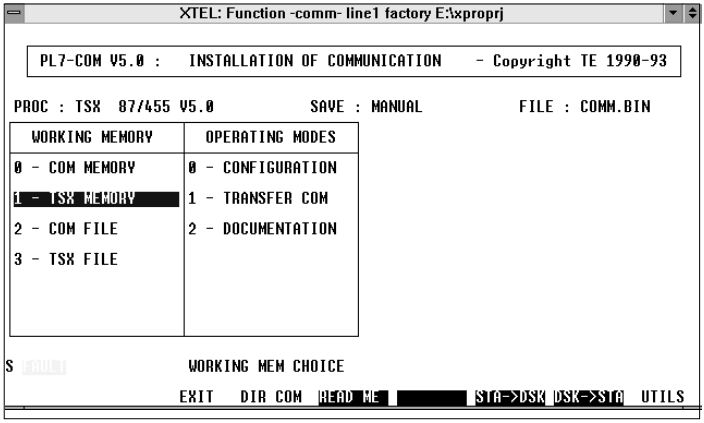


2.3 Selecting TSX/PMX MEMORY

The PLC memory is essentially for archiving. It allows the various configurations to be stored in the dedicated COM zone of the PLC memory.

This store operation allows the PLC program to reload the configurations, via the COMLD optional function block, if required. (The COMLD OFB is described in section 2 in divider C).

The TSX/PMX MEMORY can only be used if the I/O configuration performed under XTEL-CONF has previously been transferred to the PLC memory. The PLC can be in STOP or in RUN.



[UTILS] accesses the utilities functions associated with the TSX/PMX MEMORY :

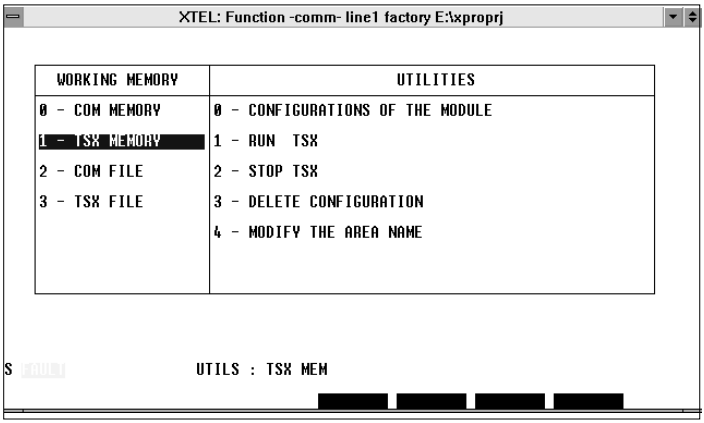
0 - CONFIGURATIONS OF THE MODULE : displays the list of configurations associated with a module :
the upper box indicates the number, the geographic address and the type of module,
the lower box indicates the channel number, the name, the date and time of creation or last modification, as well as the size of all the configurations stored in the PLC memory.

1 - RUN TSX or RUN PMX : sets the PLC to RUN,

2 - STOP TSX or STOP PMX : sets the PLC to STOP,

3 - DELETE CONFIGURATION : clears the configuration of the selected channel, after confirmation.

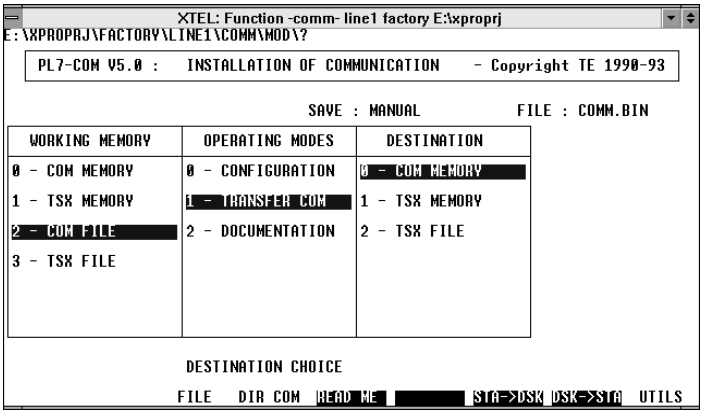
4 - MODIFY AREA NAME : associates a comment of up to 24 characters with the current COMM.BIN file.



2.4 Selecting a COM file

This is recommended for creating SCM configurations in the design office, or as a means of archiving. It does not require the PLC, the module, nor the PL7-3 configuration.

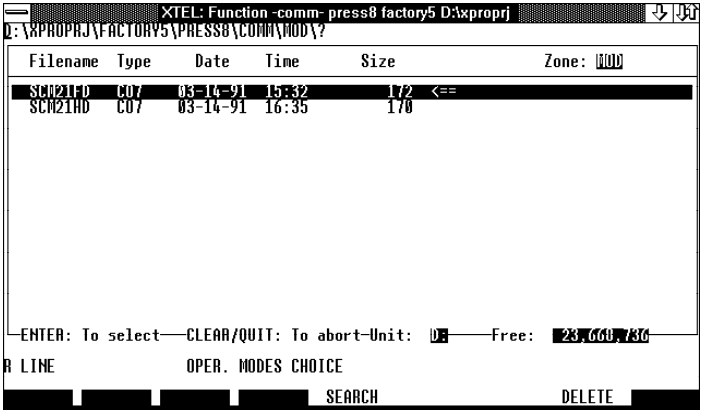
The configurations are stored on the hard disk or diskette (defined in the VOLUMES of the X-TEL Software Workshop) as they are entered.
The configurations created are "anonymous" : they are not associated with any module and are not dependent on any PL7-3 application (library function).



[UTILS] accesses the utilities functions associated with the DISK memory :

0 - MOD DIRECTORY: displays the list of files contained in the COMM\MOD directory.

1 - LIST OF CONFIGURATIONS : displays the list of xxx.C07 configuration files, contained in the COMM\MOD directory.



Whichever utility function is selected :

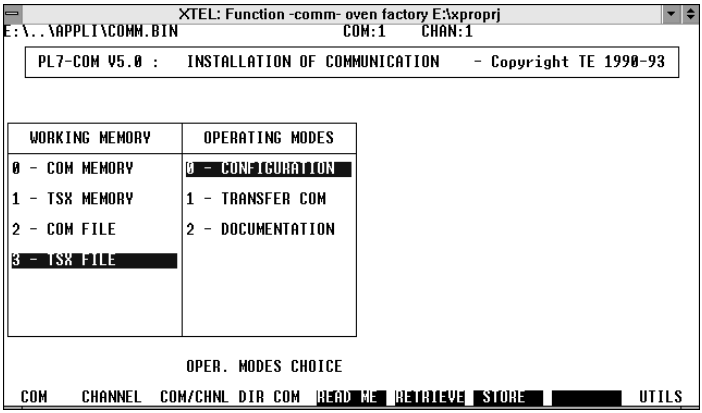
[SEARCH] searches for a file in a list.

[DELETE] deletes, after confirmation (YES), the file indicated by the cursor.

2.5 Selecting a TSX/PMX file

This mode allows an image of the PLC memory to be created in Local mode.

To use the TSX/PMX file, the station configuration must have previously been created using XTEL-CONF.



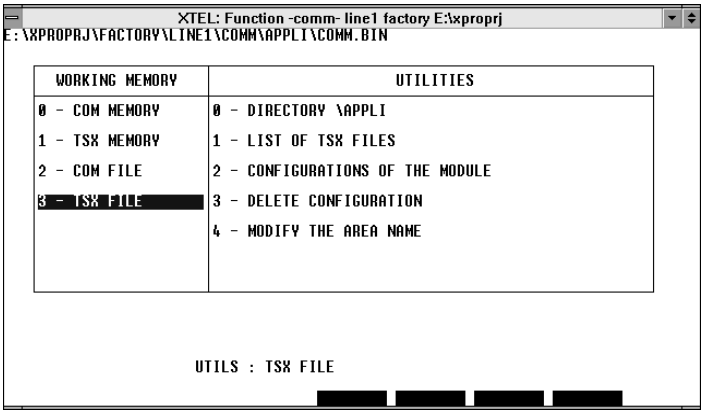
[STORE] Stores the configuration in a *.BIN file. By default, the store name is COMM.BIN. The screen displays two keys :

[FILE] enables selection of another name for the store file : *.BIN

[STORE] executes the store function.

[RETRIEVE] Enables a *.BIN file, previously stored using the STORE function, to be retrieved.
The file is restored in the XTEL zone under the name : COMM.BIN.

[UTILS] accesses the utilities functions associated with the TSX/PMX file :



0 - DIRECTORY\APPLI : displays the list of files contained in the directory :
COMM\APPLI (xxx.BIN, xxx.DOC files etc).

1 - LIST OF TSX FILES : displays the list of configuration files (xxx.BIN files).

2 - CONFIGURATIONS OF THE MODULE : displays the list of all the configurations associated with an SCM module in the current xxx.BIN file.

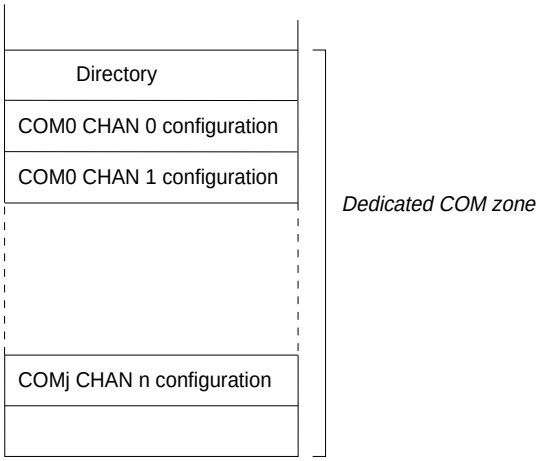
3 - DELETE CONFIGURATION : clears the configuration, specified by a COM number and a channel number, in the current xxx.BIN file.

4 - MODIFY AREA NAME : associates a comment of up to 24 characters with the current xxx.BIN file.

3.1 Dedicated COM zone

This PLC memory zone is used for storing the directory and the various configurations which may be loaded into the SCM modules. This zone is managed entirely by PL7-COM :

- The directory is created by PL7-COM.
- The configurations are entered :
  - either by direct entry to the TSX memory, from PL7-COM,
  - or by transferring a COM FILE to the TSX MEMORY,
  - or by transferring a COM MEMORY to the TSX MEMORY.



Any attempt to transfer a configuration to the PLC memory or to modify an existing configuration may be preceded by one of the following two messages :

- **Area full :** The size of the dedicated COM zone is insufficient to receive the new configuration. The size of the zone can be modified using the XTEL-CONF tool.
- **Area to be compacted :** The size of the dedicated COM zone is sufficient, provided that it is compacted. Optimizing the area in this way removes the "holes" created during transfer operations or when configurations are deleted. Compacting is performed by the [PACK] key accessible from the directory screen (see section 3.2).

3.2 Directory

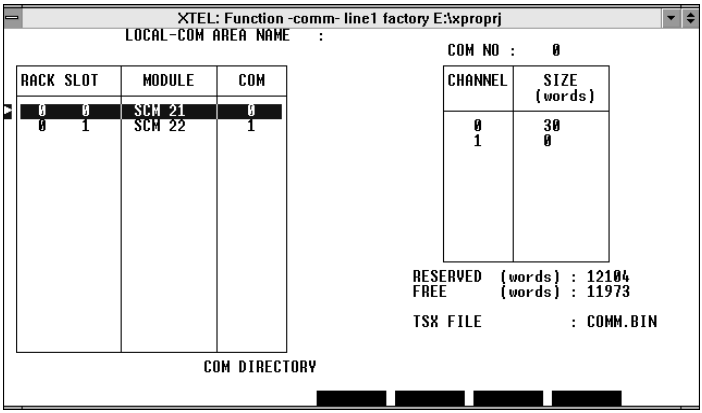
An SCM configuration is defined by :

- a module number from 0 to 63,
- a channel number 0 or 1.

It is the directory which defines the correspondence between the geographic position of the modules in the I/O configuration and the logic numbers.

Created by PL7-COM, the directory is stored initially in the dedicated COM zone of the PLC memory. The first 64 SCM modules in the I/O configuration are allocated a COM number from 0 to 63 in ascending order.
The allocation of these numbers may be modified by the user.

If PL7-COM is operating in connected mode (COM MEMORY or TSX MEMORY), the [DIR COM] soft key will display the COM DIRECTORY screen.



TSX-COM space

RESERVED The number of reserved words is fixed by XTEL-CONF. This number cannot be modified by PL7-COM.

FREE the number of free words represents the memory area not used.

TSX FILE COMM.BIN is the name under which the dedicated COM zone will be stored on the disk using the [STORE] command.

Soft keys

[ALL DIR] provides a detailed view of the COM directory, specific to each module :

- its geographic location : rack, module,
- its type (UNI-TELWAY, SCM 21, SCM 20, etc),
- the number of the associated COM,
- for each channel, the size of the configuration to which it is assigned.

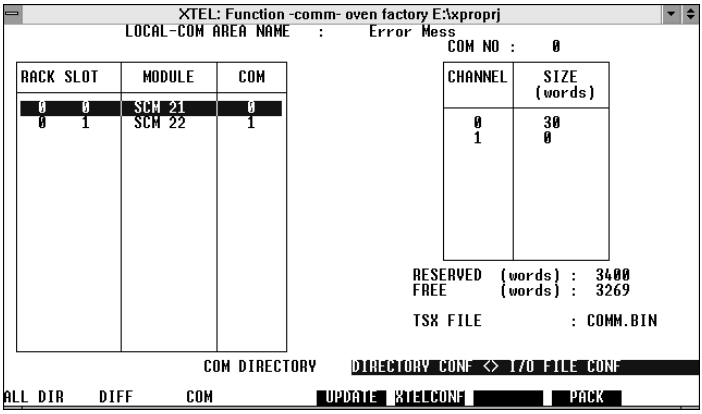
[TOP] displays the start of the directory,
[BOT] displays the end of the directory,
[PREVPAGE] displays the previous page,
[NEXTPAGE] displays the next page.

This is the view of the directory which will be provided in the documentation.

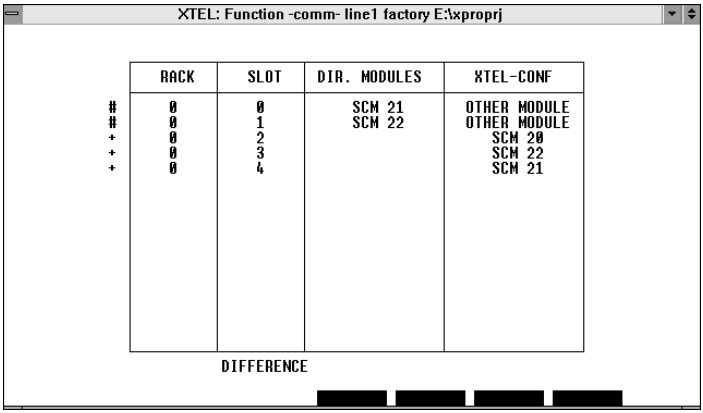
[COM] is used to modify the default assignment of the COM numbers. A number can only be assigned to one slot.

[PACK] compresses the dedicated COM zone. It is also used to recover empty spaces which have been created, for example, when configurations have been deleted.

[UPDATE] modification of a slot such as the addition or deletion of an SCM module in an I/O configuration using XTEL-CONF, is indicated in the COM directory by an asterisk which precedes each module concerned.
Displayed only in this case, the [UPDATE] key causes the directory to be updated following each new I/O configuration defined by XTEL-CONF.



[DIFF] displays the differences between the configuration of the SCM modules stored in the COM directory and the current configuration of the SCM modules.
In connected mode, the current configuration of the SCM modules corresponds to the configuration stored in the PLC.
In local mode, the current configuration of the SCM modules corresponds to the I/O configuration defined under XTEL-CONF.
This key is not displayed unless a difference is detected (a module added, removed or modified).



If a configuration is made up of more than 16 modules, the following soft keys are displayed :

[TOP] accesses the first module on the first page of the directory,

[BOT] accesses the first module on the last page of the directory,

[PREVPAGE] accesses the first module on the previous page of the directory,

[NEXTPAGE] accesses the first module on the next page of the directory.

B

4.1 CONFIGURATION mode

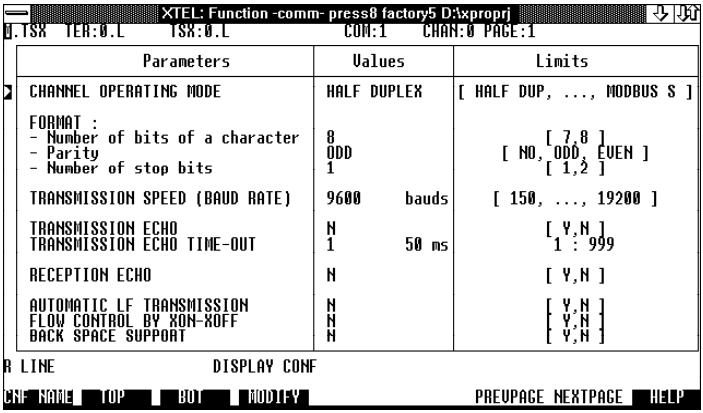
The CONFIGURATION mode is used for entering or modifying configuration parameters for a communication module channel. When a configuration is created, all the parameters take on a default value. These are not necessarily convenient for all applications. It is therefore necessary to check all the values before confirming the configuration.

The detailed description of each of the parameters is provided in the manuals accompanying the modules. On-line documentation also helps the user in making this selection.

The configuration screen is accessed from the basic PL7-COM screen :

- Select the working memory (see section 2).
- Use the < → > command to access the CHOICE OF MODES screen.
- Enter a COM number and a channel number using the [COM] and [CHANNEL] or [COM/CHNL] soft keys. If it is the COM file which is used, choose a file name using the [FILE] soft key.
- Select 0 - CONFIGURATION then confirm with <Enter>.

A screen specific to each type of SCM is offered. For example, the screen below corresponds to the configuration parameters of a TSX SCM 21.



Depending on the type of SCM, the configuration screen comprises 1 to 3 pages. Each of these pages has 3 columns :

Parameters designates the parameters,

Values indicates the value of each parameter. It is this zone which is completed by the user,

Limits indicates the choices available or the limits for each of the parameters. It is also the zone for help messages displayed by using the [HELP] soft key.

Soft keys

[CNF.NAME] allows the user to enter the name of the configuration (maximum of 16 alphanumeric characters),

[TOP] displays the first page and positions the cursor at the start of this page,

[BOT] displays the last page and positions the cursor at the start of this page,

[MODIFY] modifies the value of the parameter pointed to by the cursor or provides access to the modification (display of the soft keys or an entry line),

[PREVPAGE] displays the previous page,

[NEXTPAGE] displays the next page,

[HELP] displays in the third column, the help information for the parameter pointed to by the cursor.

4.2 DEBUG mode

4.2-1 Introduction to the mode

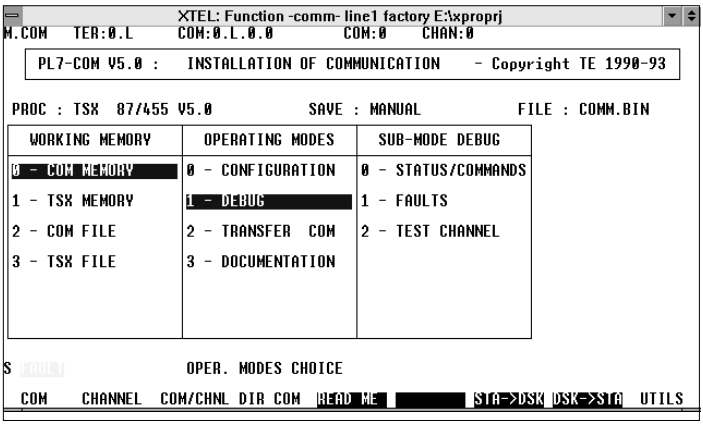
B

The DEBUG mode is used to display and modify the bits and words of the discrete interface or the communication module register. It can only be accessed if the COM memory has been selected.

A detailed description of each of the parameters is supplied in the accompanying documentation for the modules.

DEBUG mode is accessed from the basic PL7-COM screen :

- Choose the COM memory (see section 2).
- Use the < → > command to access the CHOICE OF MODES screen.
- Enter a COM number and a channel number using the [COM] and [CHANNEL] or [COM/CHNL] soft keys.
- Select 1 - DEBUG.
- Use the < → > command to access the choice of function.
- Choose the function required then confirm with <Enter>.



4.2-2 STATUS/COMMANDS screen

This screen groups the data and commands available at the discrete interfaces and registers of the module. These are displayed in a mnemonic form indicating the status of the bits. Bits at state 1 appear in reverse video (or highlighted or in green on a color screen).
A clear message is also displayed in the entry line providing a comment about the bit or word pointed to by the cursor.

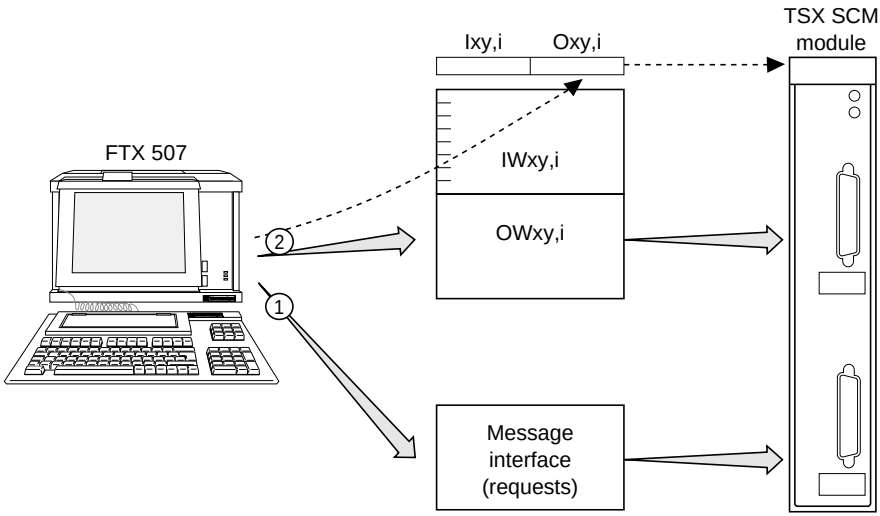
The left side of the screen displays the (STATUS) information provided by the module about the discrete interfaces (Ixy,i) and registers (IWxy,i).
The right side of the screen displays the commands to be sent to the module, via the discrete interfaces (Oxy,i) and registers (OWxy,i).

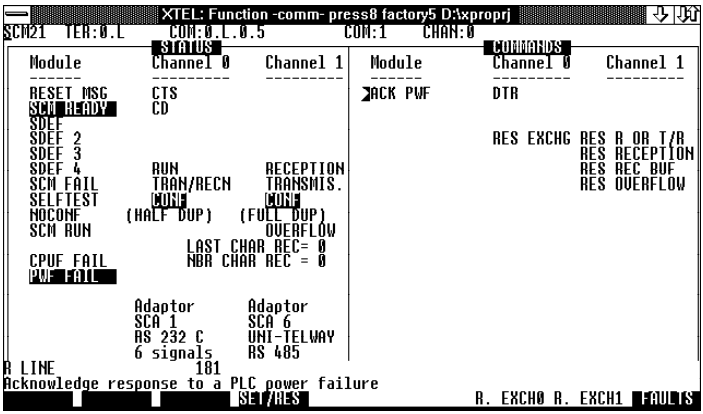
The information and commands available are divided into three columns :

- information or commands concerning the status of the module,
- information or commands concerning channel 0 of the module,
- information or commands concerning channel 1 of the module,

Modifying a command bit causes a request to be addressed directly to the module via the message interface (1). The program then updates the discrete and register command interfaces (2), to ensure that the information is consistent with the status of the module.

This mechanism enables the communication module to be set up with the PLC in STOP.





[SET/RES] enables the status of the command bit pointed to by the cursor to be modified.

[R. EXCH0] successively pressing this key :

- cancels the current request and stops sending the request on channel 0 (RST EXC command shown in reverse video or high-lighted),
- or authorizes the request to be sent on channel 0.

[R. EXCH1] successively pressing this key :

- cancels the current request and stops sending the request on channel 1 (RST EXC command shown in reverse video or high-lighted),
- or authorizes the request to be sent on channel 1.

[FAULTS] provides access to the fault display screen.

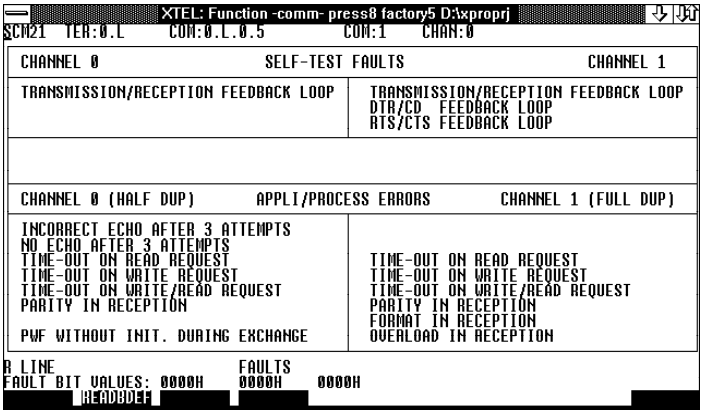
4.2-3 FAULT screen

This screen displays the list and state of non blocking fault bits of the module :

- module fault bits and results of self-tests in the upper part of the screen,
- application fault bits in the lower part of the screen.

For any type of fault, bits at state 1 are displayed in reverse video (or highlighted or in green on a color screen).

The fault screen also displays the fault bits string (BDEF) on the entry line.



[READBDEF] generates a request to read the fault bits string (BDEF), in order to refresh the display of these bits. The list of fault bits for the module is also refreshed.

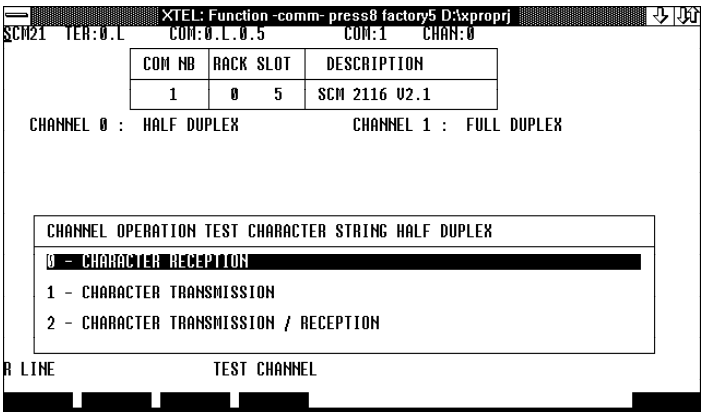
4.2-4 CHANNEL TEST screen

This screen enables the connection line to be tested according to its configuration :
- half-duplex,
- full-duplex,
- UNI-TELWAY,
- MODBUS®.

The type of module (COM number) and channel selected defines the type of connection to be tested and therefore the menu to be displayed :
- test a half-duplex character string channel,
- test a full-duplex character string channel,
- test a UNI-TELWAY master channel,
- test a UNI-TELWAY slave channel,
- test a MODBUS® master channel,
- test a MODBUS® slave channel.

1. Testing a half or full-duplex character string channel

The functions offered for testing a half or full-duplex character string connection are as follows :
- 0 - character reception,
- 1 - character transmission,
- 2 - character transmission/reception.



Character reception

This function enables up to 126 characters to be received, within a time envelope, on a line configured for half-duplex or full-duplex operation.
The corresponding screen has two parts :

- an upper part used for defining the request parameters :
  - number of characters to be received,
  - time envelope setting,
- a lower part which displays the response parameters on receipt of the number of characters or at the end of the time envelope :
  - reminder of the request code,
  - confirmation report for the exchange,
  - data received.

[MODIFY] enables the parameter pointed to by the cursor to be modified :
- number of characters to be received (1 to 126),
- time envelope setting (100ms to 60s).

[FAULTS] provides access to the fault display screen for the channel tested.

Character transmission

This function is used to transmit characters, within a defined time envelope, over a line configured for half-duplex or full-duplex operation.
The corresponding screen has two parts :

- an upper part used to define the request parameters :
  - time envelope setting,
  - data to be transmitted,
- a lower part which displays the response parameters on transmission of the characters or at the end of the time envelope :
  - reminder of the request code,
  - confirmation report for the exchange.

The characters transmitted are separated by commas and can be expressed :

- in ASCII (enclosed in single quotation marks),
- in hexadecimal (2 characters maximum from 0 to F).

Example : Write the message FAULT P1, skip two lines then write the message FAULT P2.

For this, the following data must be sent :

'FAULT P1',0A,0A,0D,'FAULT P2'

[MODIFY] enables the parameters pointed to by the cursor to be modified :
- time envelope setting (100ms to 60s),
- data to be transmitted.

[FAULTS] provides access to the fault display screen for the channel tested.

Transmission/reception of characters in half-duplex

This function is used to transmit and then to receive characters within a defined time envelope over a line configured for half-duplex operation.
The corresponding screen has two parts :

- an upper part used to define the request parameters :
  - number of characters to be received,
  - time envelope setting,
  - data to be transmitted,
- a lower part which displays the response parameters on transmission/reception of characters or at the end of the time envelope :
  - reminder of the request code,
  - confirmation report for the exchange,
  - data received.

The characters transmitted are separated by commas and can be expressed :

- in ASCII (enclosed in single quotation marks),
- in hexadecimal (2 characters maximum from 0 to F).

[MODIFY] enables the parameters pointed to by the cursor to be modified :
- number of characters to be received (1 to 126),
- time envelope setting (100ms to 60s),
- data to be transmitted.

[FAULTS] provides access to the fault display screen for the channel tested.

Transmission/reception of characters in full-duplex

This function allows simultaneous transmission and reception of characters within a time envelope over a line configured for full-duplex operation.
The corresponding screen has two parts :

- an upper part used to define the request parameters :
  - number of characters to be received,
  - time envelope setting,
  - data to be transmitted,
- a lower part which displays:
  - the transmission response parameters on transmission of the characters or at the end of the time envelope :
    . reminder of the request code,
    . confirmation report for the exchange,
  - the reception response parameters on reception of the characters or at the end of the time envelope :
    . reminder of the request code,
    . confirmation report for the exchange,
    . data received.

The characters transmitted are separated by commas and can be expressed :

- in ASCII (enclosed in single quotation marks),
- in hexadecimal (2 characters maximum from 0 to F).

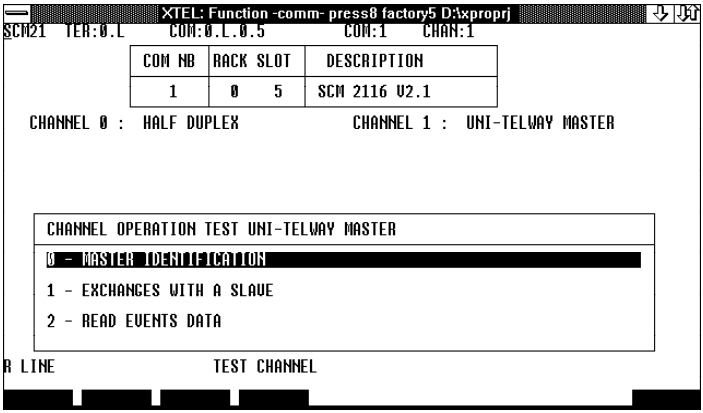
[MODIFY] enables the parameters pointed to by the cursor to be modified :
- number of characters to be received (1 to 126),
- time envelope setting (100ms to 60s),
- data to be transmitted.

[FAULTS] provides access to the fault display screen for the channel tested.

2. Testing a UNI-TELWAY master channel

The following functions are offered for testing a UNI-TELWAY master link :

- 0 - master identification,
- 1 - exchanges with a slave,
- 2 - read events data.



Identification of the master

The screen for this function displays three boxes which are used to :

- identify the line configured as a UNI-TELWAY master,
- continuously display all the slave stations connected and scanned by the master (up to 31 stations),
- display the error counters for the master:
  - number of messages sent and not acknowledged,
  - number of messages sent and refused,
  - number of messages received and not acknowledged,
  - number of messages received and refused.

[CLEAR CT] resets the error counters for the master to zero.

Exchanges with a slave

This function is used to send a UNI-TELWAY request from the master to a slave. The corresponding screen has 2 parts :

- an upper part used to define the request parameters :
  - connection address,
  - request code,
  - data to be transmitted,
- a lower part which displays the response parameters :
  - reminder of the request code,
  - confirmation report for the exchange,
  - data received.

General usage requests (identification of equipment, protocol version, status and mirror) are coded automatically by a soft key. Other requests must be entered by the user. Enter the request code and the data to be transmitted.

The data to be transmitted depends on the syntax of the request and is expressed :

- in ASCII (enclosed in single quotation marks),
- in hexadecimal, byte by byte (2 characters maximum from 0 to F).

A word is expressed as two bytes, the first byte being the least significant byte and the second the most significant.

Examples :

- read bit B100 (100 decimal = 0064 hexadecimal)
  - request code : 00
  - transmission data (number of the bit in a word) : 64,00
- write bit B10 (10 decimal = 000A hexadecimal) to 1
  - request code : 10
  - transmission data (number of the bit in a word and value of the bit in a byte) : 0A,00,01
- read word W1000 (1000 decimal = 03E8 hexadecimal)
  - request code : 04
  - transmission data (number of the word in a word) : E8,03
- write word W500 (500 decimal = 01F4 hexadecimal) to the value 54321 (54321 decimal = D431 hexadecimal)
  - request code : 14
  - transmission data (number of the word in a word and value of the word in a word) : F4,01,31,D4

The data received is coded in hexadecimal, byte by byte (refer to corresponding documentation : UNI-TELWAY reference manual).

A word is expressed as two bytes, the first byte being the least significant part of the word and the second byte the most significant.

[MODIFY] enables the parameters pointed to by the cursor to be modified :

- communication address (1 to 152),
- request code in hexadecimal,
- data to be transmitted respecting the request syntax.

[IDENT.] automatically codes the identification request. The corresponding parameters are displayed in the upper box.

[PROT VER] automatically codes the protocol version request. The corresponding parameters are displayed in the upper box.

[STATUS] automatically codes the status request. The corresponding parameters are displayed in the upper box.

[MIRROR] automatically codes the mirror request. The corresponding parameters are displayed in the upper box.

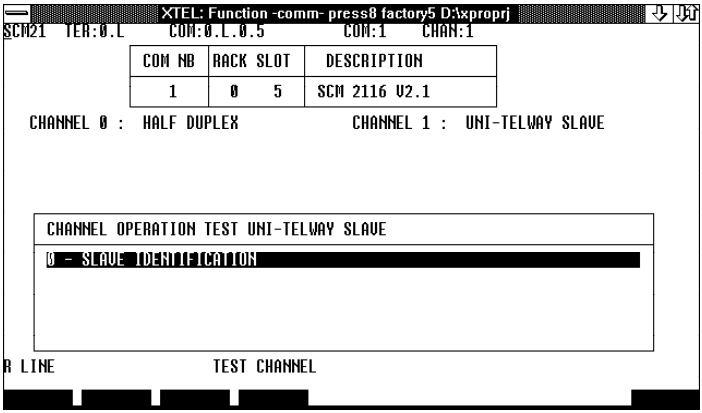
Reading event data

This function is used to read event data addressed by the 32 UNI-TELWAY stations. Only data which has changed state is read (see description of A8H request in the UNI-TELWAY reference manual).

3. Testing a UNI-TELWAY slave channel

The following function is offered for testing a UNI-TELWAY slave connection :

- 0 - slave identification.



Identification of the slave

This screen for this function displays two boxes which are used to :

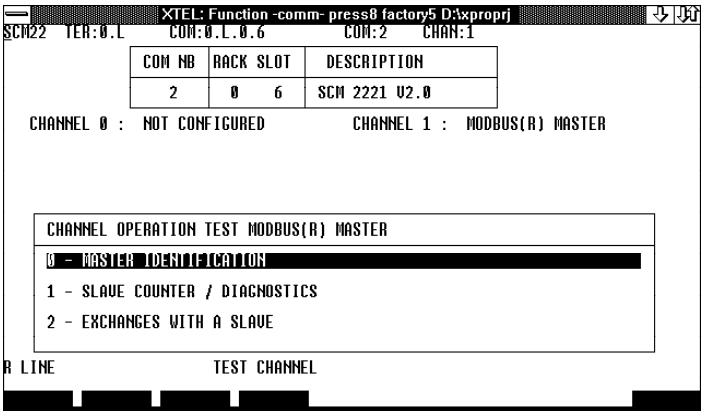
- identify the connection configured as a UNI-TELWAY slave,
- display the error counters for the slave :
  - number of messages sent and not acknowledged,
  - number of messages sent and refused,
  - number of messages received and not acknowledged,
  - number of messages received and refused.

[CLEAR CT] resets the error counters for the slave to zero.

4. Testing a MODBUS® master channel

The following functions are offered for testing a MODBUS® master connection :

- 0 - master identification ,
- 1 - slave counter / diagnostics,
- 2 - exchanges with a slave.



Note

When testing a MODBUS® master (channel) connection, this is successively :

- placed in STOP,
- placed in RUN during the test,
- placed in STOP when the test is completed.

Identification of the master

The screen for this function displays the counters for the line configured as a MODBUS® master :

- exchanges between the processor and the master module,
- exchanges between the master module and the slaves.

Slave counter / diagnostics

These functions are used for :

- sending an ECHO request to the slave,
- displaying the MODBUS® slave connection monitoring counters via the MODBUS® master connection.

Warning

This test initializes the master counters.

The corresponding screen has two parts :

- an upper part used to define the slave address,
- a lower part which displays the response parameters :
  - confirmation report for the ECHO request (Correct or incorrect),
  - value of the slave counters.

When the connection between the master and the slave is incorrect and/or after a time envelope (*), an error code is displayed in the message zone of the screen :

- 01H function number not defined at the slave,
- 02H MODBUS® address not recognized by the slave,
- 03H illegal value for MODBUS® address indicated,
- 04H fatal error of slave processor (CPUF),
- 05H acknowledgment : the slave processor has acknowledged and is processing the master's request,
- 06H slave processor occupied,
- 07H negative acknowledgment,
- 15H no correct response from the slave after several attempts,
- 17H request parameters incorrect,
- 18H channel in STOP,
- 19H slave in listen only mode (LOM),
- FDH MODBUS® function not recognized,
- FEH positive confirmation report.

Possible causes of error are :

- connection incorrect or absent,
- MODBUS® slave in STOP,
- PLC central processor supporting the slave is in STOP.

[MODIFY] used to define the address of the slave connection to be interrogated.

[CLEAR CT] initializes slave monitoring counters.

(*) The time envelope is calculated automatically according to the delay time for the response and the number of attempts defined in configuration. It is limited to 60 seconds.

Exchanges with a slave

This function is used to send a request from the MODBUS® master to a MODBUS® slave.

Warning

This test initializes the master counters.

The corresponding screen has two parts :

- an upper part used to define the request parameters :
  - the slave address,
  - request code,
  - data to be transmitted,
- a lower part which displays the response parameters :
  - reminder of the request code,
  - confirmation report for the exchange,
  - data received.

The ECHO request is automatically coded by a soft key. Other requests are entered by the user by typing the request code and the data to be transmitted.

The data to be transmitted depends on the request syntax and is expressed :

- in ASCII (enclosed in single quotation marks),
- in hexadecimal, byte by byte (2 characters maximum from 0 to F).

A word is expressed as two bytes, the first byte being the least significant part of the word and the second the most significant.

Examples :

- read connection events
  - request code : 0C
  - transmission data : no data
- read the number of messages received on the line
  - request code : 08
  - transmission data (diagnostic code) : 0B,00
- read the 10 (10 decimal = 000A hexadecimal) output words, the first word being the MODBUS® address 500 (500 decimal = 01F4 hexadecimal)
  - request code : 03
  - transmission data (MODBUS® address for the first word and number of words to read) : F4,01,0A,00

Data received is coded in either decimal or hexadecimal, word by word (refer to the corresponding documentation).

When the connection between the master and the slave is not correct and/or at the expiry of the time envelope (*) an error code is displayed in the message zone on the screen (refer to the diagnostic/slave counters function described earlier).

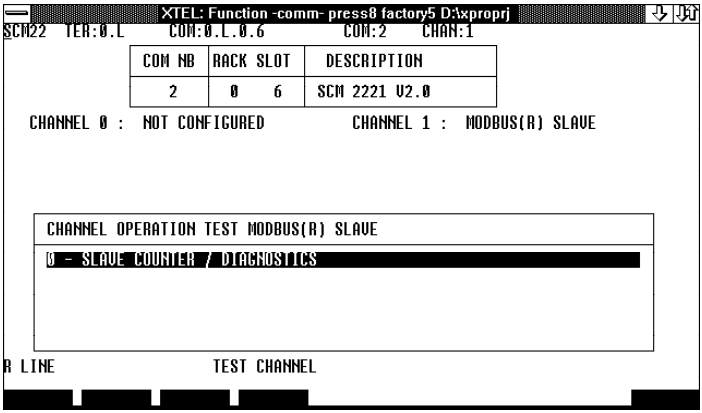
[ECHO] automatically codes the ECHO request. The corresponding parameters are displayed in the upper box.

[MODIFY] enables the parameters pointed to by the cursor to be modified :
- slave address,
- request code in hexadecimal,
- data to be transmitted respecting request syntax.

5. Testing a MODBUS® slave channel

The following functions are offered for testing a MODBUS® slave channel :

- 0 - slave counter / diagnostics.



Slave counter / diagnostics

The screen for this function displays :

- the station number,
- the monitoring counters for the slave connection.

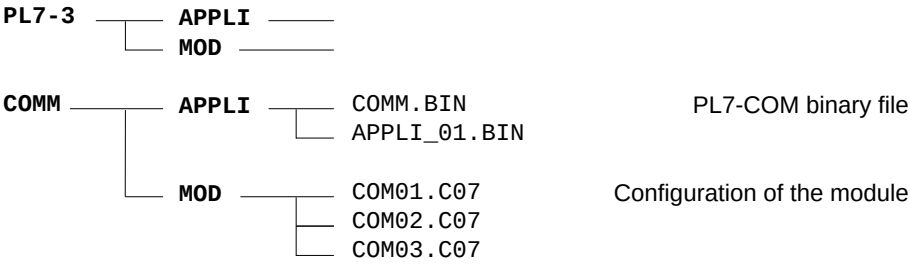
(*) The time envelope is calculated automatically according to the delay time for the response and the number of attempts defined in configuration. It is limited to 60 seconds.

4.3 TRANSFER mode

4.3-1 PL7-COM files

PL7-COM files are designated by a name with up to 8 characters, followed by a 3 character extension which indicates the type of file. They are stored in various sub-directories of the software workshop.

The application files are stored in the station, under the COMM sub-directory :



The directory accessible to the station for communication functions is the COMM directory :

- the COMM directory has two sub-directories :
  - the APPLI sub-directory which contains the xxx.BIN files in which the dedicated COMM zones are stored,
  - the MOD sub-directory which contains the files in which the SCM configurations of the module are stored. Each file contains a configuration which may require loading into a module. Each application generates a .C07 file.

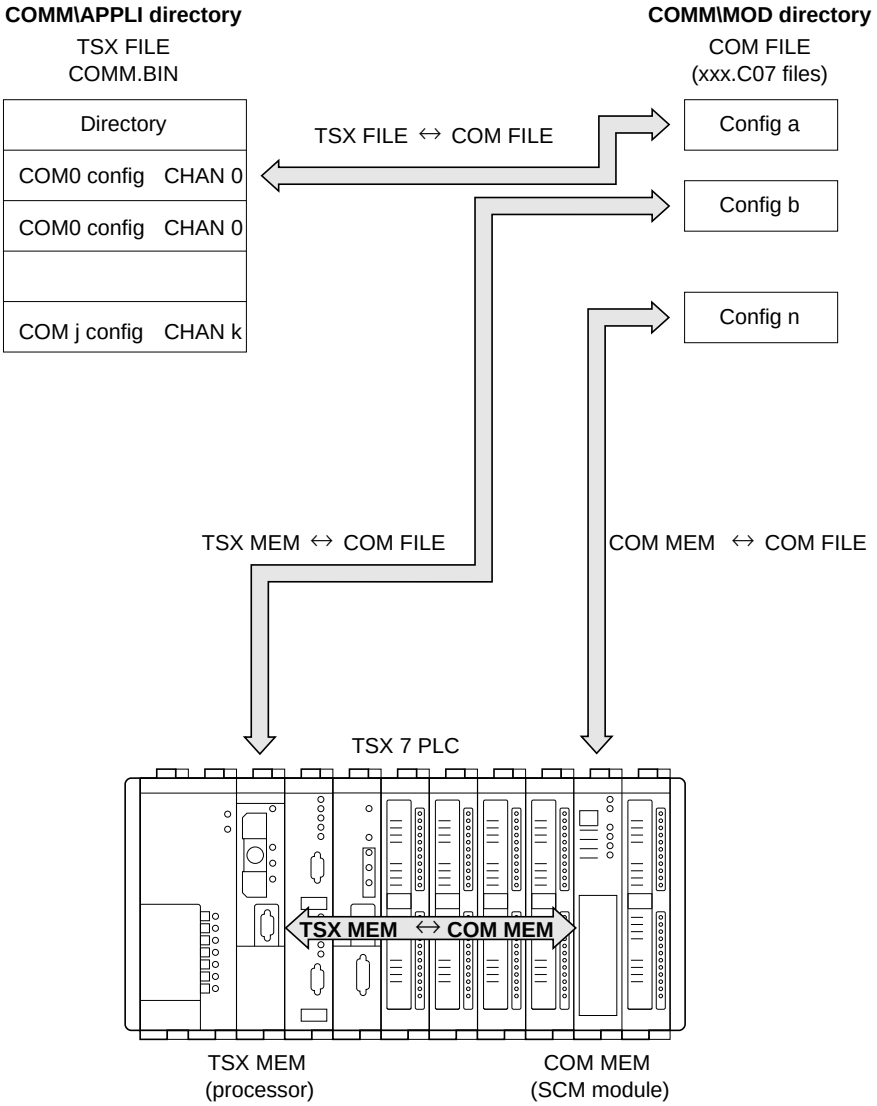
All the files in the COMM directory are generated by the PL7-COM function.

For the rest of this section :

- TSX FILE corresponds to files in the COMM\APPLI directory
- COM FILE corresponds to files in the COMM\MOD directory.

4.3-2 Possible transfers

The various possibilities offered by the TRANSFER mode are described below :

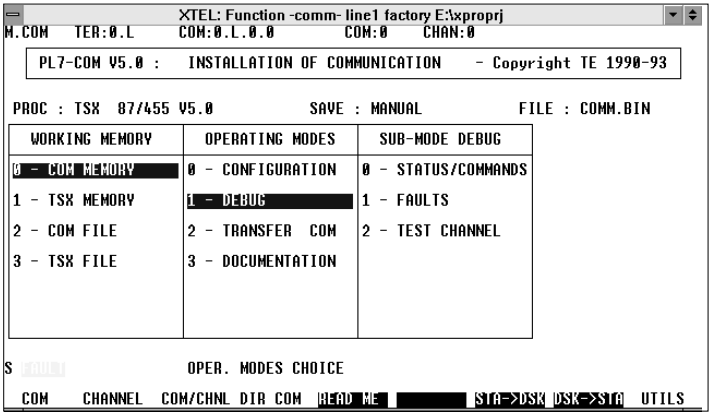


An application (directory and configurations) is transferred between the TSX file and the TSX memory using the STORE and RETRIEVE soft keys.

4.3-3 Using the TRANSFER mode

The TRANSFER mode is accessed from the basic PL7-COM screen :

- Select the source memory containing the application to be transferred (see section 2 - choice of working memory).
- Access the CHOICE OF MODES screen with the command < → >.
- Define the configuration to be transferred : enter a COM number and the channel number with the commands [COM] and [CHANNEL] (or [COM/CHNL]) or the name of a file with the command [FILE].
- Choose TRANSFER mode, which is displayed in the DESTINATION column.
- Access the destination memory selection with the command < → >.
- Select the destination memory, then confirm with <Enter>.



The soft key menu allows the source memory or the source file to be selected.

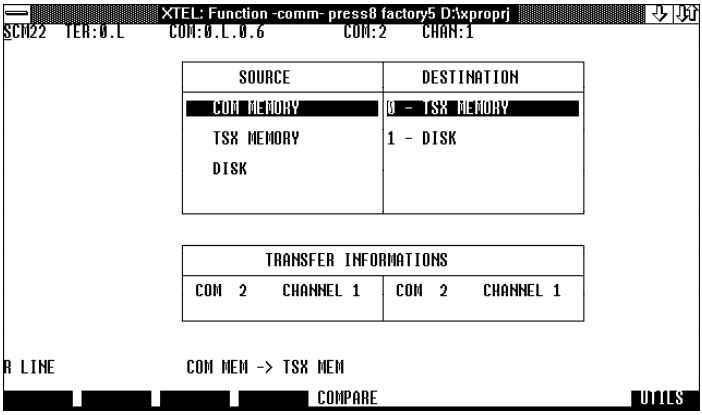
[FILE] enables the user to enter the name of the source file to be transferred from a COM file,

[COM] allows the user to choose the number of the communication module for transferring from the COM or TSX memory or from a TSX file,

[CHANNEL] allows the user to choose working channel number to be selected to transfer from the COM or TSX memory or a TSX file,

[COM/CHNL] allows the user to choose the number of the communication module and the channel for transferring from the COM or TSX memory or from a TSX file.

Example of a transfer screen :



The upper box displays the choice of source and target memories.
The destination memory can then be modified by the up and down arrows or by the numeric keys.

The lower box displays the address or name of the source application (in the left hand column) and destination application (in the right hand column).

[COMPARE] compares the contents of the source memory with that of the destination memory,

[UTILS] accesses the utilities functions associated with the source memory,

<Enter> initiates transfer between the source memory and the destination memory.

4.4 DOCUMENTATION mode

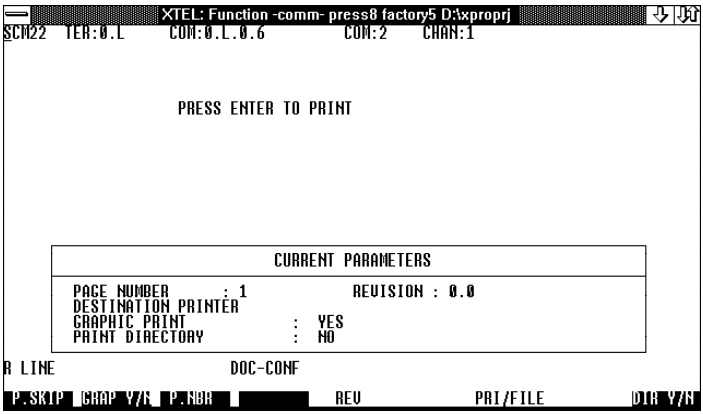
4.4-1 General

B

The DOCUMENTATION mode is used to print the configuration of a channel, of a TSX SCM module or of all the modules, or to save this to a xxx.DOC file for use by the XTEL-DOC tool.

The documentation screen is accessed from the CHOICE OF MODES screen :

- Select the working memory (see section 2).
- Access the CHOICE OF MODES screen with the command < → >.
- Enter a COM number (*) and channel number using the commands [COM] and [CHANNEL] (or [COM/CHNL]) or a file name using the command [FILE].
- Select DOCUMENTATION mode, then confirm with <ENTER>.



(*) To print or save all channels and modules to the xxx.DOC file, use an asterisk (*) in place of the channel and/or COM number :

- CHNL0, COM1 : channel 0 of module 1,
- CHNL*, COM1 : channels 0 and 1 of module 1,
- CHNL0, COM* : channel 0 of all the modules,
- CHNL*, COM* : channels 0 and 1 of all the modules.

[P.SKIP] skips the page.

[GRAP Y/N] enables the type of printer to be defined : graphic (yes) or ASCII (no). The selection appears in the CURRENT PARAMETERS box.

[P.NBR] allows the user to enter the number of the first page of the dossier. This number appears in the PAGE NUMBER item of the CURRENT PARAMETERS box.

[REV] allows the user to enter the revision index (optional) of the dossier. This number appears in the REVISION item of the CURRENT PARAMETERS box.

[PRI/FILE] enables the output to be defined : printer or xxx.DOC file.
The output chosen appears in the CURRENT PARAMETERS box.

The xxx.DOC file is placed under the directory XPROPRJ\PROJET\STATION\COMM\MOD, if the directory is not printed (command [DIR Y/N]). It is placed under the XPROPRJ\PROJET\STATION\COMM\APPLI directory, if the directory is printed (command [DIR Y/N]).

The xxx.DOC file is used by the XTEL-DOC tool, if the directory is printed (command [DIR Y/N]) and if graphic mode is chosen (command [GRAP Y/N]).

[FILE] allows the user to enter the name of a xxx.DOC documentation file, when the output selected is the file.

[DIR Y/N] allows the directory to be printed or not. The choice appears in the CURRENT PARAMETERS box.

<ENTER> starts printing the dossier, using the parameters in the CURRENT PARAMETERS box.

Important

When the output is to disk, the output file is located :

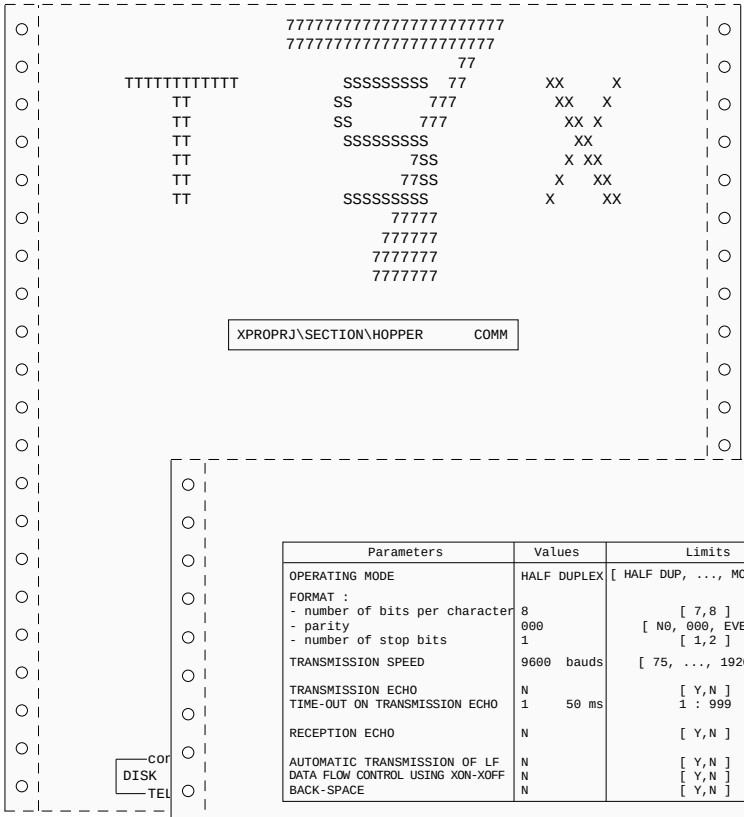
- in the COMM\APPLI directory if the print directory option has been requested (key [DIR Y/N],
- in the COMM\MOD directory otherwise.

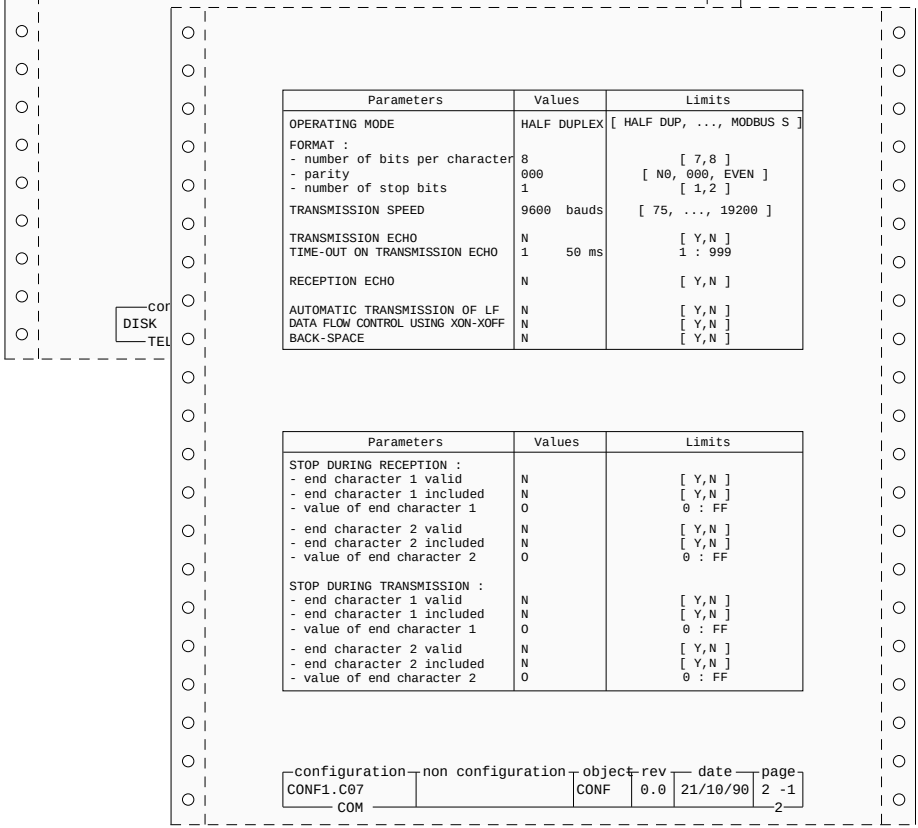
In both cases, the file name is followed by the extension .DOC.

For a file to be generated for use ultimately by the XTEL-DOC tool, the following two conditions must be respected :

- file must be under the COMM\APPLI directory,
- file must be created with the graphic printing option (key [GRAP Y/N].

4.4-2 Example of a listing page





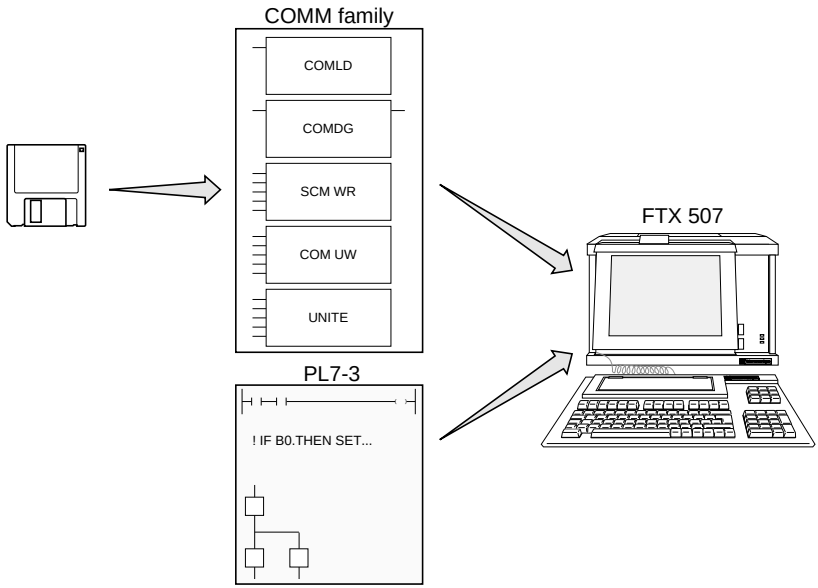
1.1 Introduction

The communication function blocks, **an extension of PL7-3 language**, are an extension of the PL7-COM setup software. This software is not sufficient to cover all the requirements encountered, for example configuration or reconfiguration of modules when an application is in use. For this a programming terminal must be used.
These requirements for loading a configuration, depending on the application (after a power break, changing a module, etc), can be performed by COMLDi Optional Function Blocks from a backup in the PLC memory.

The communication OFBs are supplied on a diskette, reference TXT LF FB CMM V5. The installation procedure is described in installing PL7-COM software (section 2.3 in divider A).

To use the communication OFBs in an application, it is necessary :

- For the COMM icon to be at station level. If this is not the case, then the software has not been installed in this station. Refer to accessing PL7-COM software (section 3.4 in divder A).
- Declare the type of OFB in the PL7-3 configuration.
- Define the number of OFBs to be used.
- Program the OFB in PL7-3.

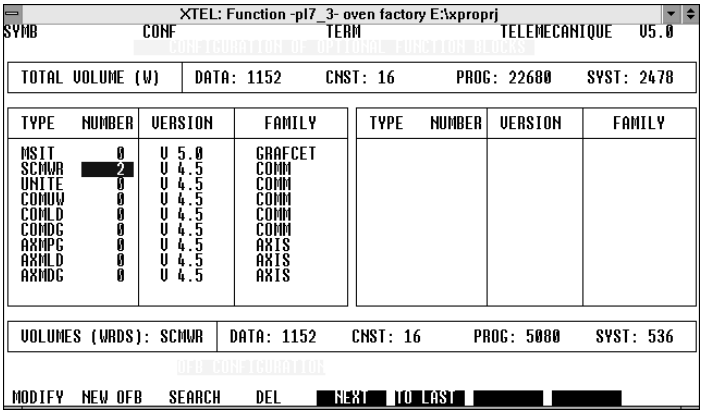


1.2 Configuring the OFBs

Before using an OFB in an application program, the type and number of OFBs must be declared in PL7-3 configuration mode.

Declaring the type and number of OFBs

From the CHOICE OF CONFIGURATION MODES screen, select item 5 - OPTIONAL FUNCTION BLOCKS. The screen then displays the list of OFBs already declared, and the number of OFBs by type.



If the COM family of OFBs is not present in this list, press the [NEW OFB] soft key. A new screen then displays the list of OFBs available on the hard disk. Use the commands [PREV FAM] and [NEXT FAM] to display the various families and the command [INS] to select the type of OFBs required for the application.

After confirming the new selection, the OPTIONAL FUNCTION BLOCKS CONFIGURATION screen is displayed again. The [MODIFY] command then allows the number of OFBs to be defined for each type declared.

For further information, refer to the PL7-3 operating modes manual.

1.3 Programming the OFBs

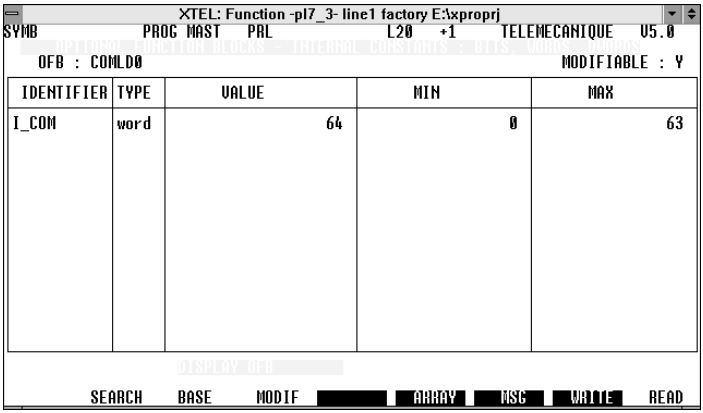
The COM family of OFBs can be programmed from any module in Ladder language (by means of an operation block) or in Literal language. The syntax is the same in both cases :

EXEC OFBi(Inp1;Inp2;Inp3;...;Inpn=>Outp1,Outp2;Outp[3;...;Outpm)

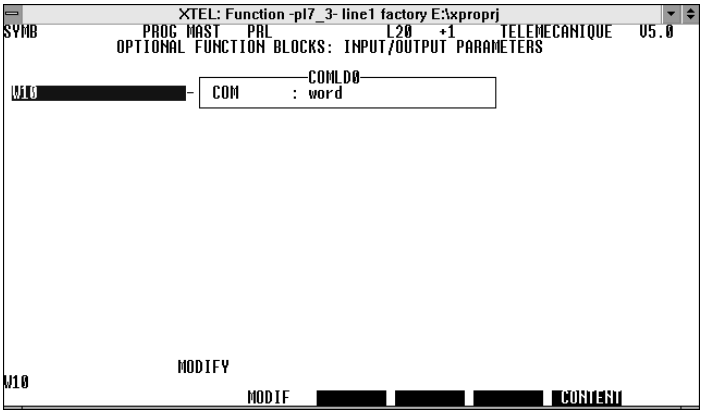
OFBi	type and number of the OFB,
Inp	input objects,
Outp	output objects,
=>	separator between the input and output parameters,
;	separator between parameters.

An OFB is programmed in PROGRAM mode.
The PL7-3 program displays the [EXEC], [CONTENT] and [PARAM] soft keys which respectively define all the values and the constants of the OFB and the I/O parameters to be specified. An instruction is entered as follows :

- Press the [EXEC] key.
- Enter the type and number of the OFB, (for example COMLD1).
- Press the [CONTENT] key to access the internal constants of the OFB :



- Initialize each constant then confirm with <Enter>.
- Press the [PARAM] key to display the parameters of the OFB :



- Assign a variable to the input and output parameters of the OFB.
- Confirm the screen then the equation with <Enter>.

It is not necessary to use (wire) all the I/O. The corresponding parameters take a default value.

The [IF], [THEN] and [ELSE] soft keys allow conditions to be placed on the execution of the OFB (for example after a warm or cold restart) :

IF (SY0+SY1) THEN RESET B0
IF NOT B0 THEN EXEC COMLD1(W12=>)
SET B0

1.4 Restrictions

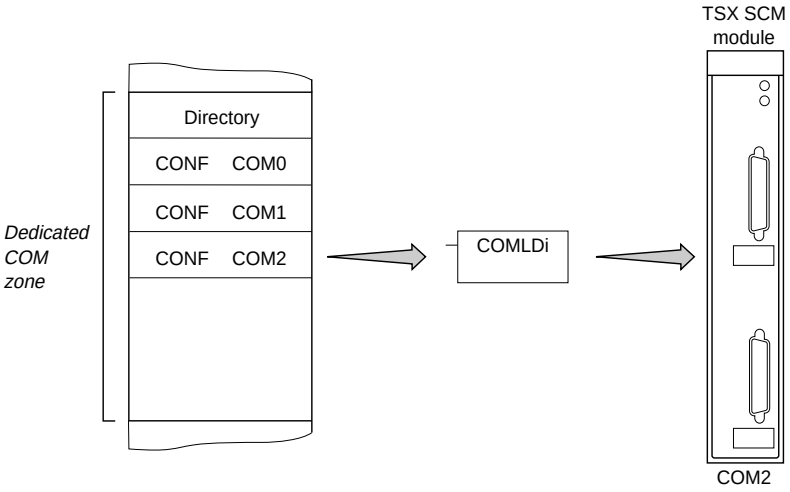
The COMLDi OFB can only by used with TSX SCM modules version greater than 1.4.

C

C

2.1 General

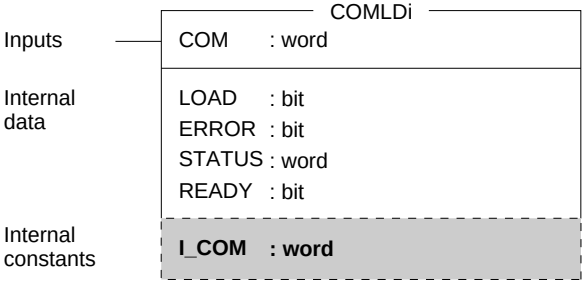
The COMLDi OFB enables the transfer, on request, of the contents of the configurations stored in the dedicated COM zone of the PLC memory, to the TSX SCM modules or to the UNI-TELWAY port incorporated in certain processors.



2.2 Introduction to COMLD

COMLD type OFBs operate in the same way as standard function blocks. Calling-up the OFB is the same as a load request.

The COMLD type OFB has an input parameter, internal constant and internal data. It does not have an output parameter.
The input parameters and the internal constant are defined during the programming phase. The internal data is used during execution.



2.3 Description of the parameters

Input parameters

Parameters	Type	Access	Description
COM	word	(2)	This word contains the number of the COM to which the OFB is assigned. On cold restart of the PLC, COM is automatically initialized with the contents of the I_COM internal constant.

Internal data

Parameter	Type	Access	Description
LOAD	bit	(2)	Setting this bit to 1 causes the configuration whose number is specified in the COM, to be transferred to the corresponding module. The effect is the same as executing the OFB with the EXEC command. It is used for forcing a configuration from a terminal.
ERROR	bit	(1)	This fault bit is set to 1, when the configuration is incorrectly transferred between the PLC and the module. It is reset to 0 after a new correct transfer.
STATUS	word	(1)	This word enables the cause of the incorrect transfer to be identified, by setting the corresponding bit to 1 (see section 2.4).
READY	bit	(1)	This bit indicates if the OFB has been executed or not. It is set to 0 during transfer of the configuration between the PLC and the module and returns to 1 when this is completed. This bit can be evaluated directly by reading COMLDi,READY.

(1) Read when using program and adjust modes (data mode, etc),
(2) Read when using program and adjust modes (data mode, etc)
Written to when using adjust mode (data mode, etc).

Internal constants

Parameter	Type	Access	Description
I_COM	word	(1)	This word indicates the logic number of the module on cold restart.

(1) Read when using program and adjust modes (data mode, etc).

C

2.4 List of faults

STATUS internal data

bit0 = 1 : TSX SCM module or UNI-TELWAY connection not functioning.
bit1 = 1 : transfer not possible : channel 0 adaptor fault.
bit2 = 1 : transfer not possible : channel 1 adaptor fault.
bit3 = 1 : module not available.
bit4 : not used.
bit5 = 1 : module absent.
bit6 = 1 : COM parameter missing in directory.
bit7 = 1 : COM module directories not defined or incoherent.
bit8 = 1 : type of module does not agree with I/O configuration.
bit9 = 1 : configurations missing from directory or incoherent.
bit10 = 1 : channel 0 configuration refused by module.
bit11 = 1 : channel 1 configuration refused by module.
bit12 = 1 : transfer cancelled by power break, software fault.
bit13 = 1 : OFB version incompatible with directory version.
bit14 = 1 : communication error (exchange not possible with module).
bit15 = 1 : system error (insufficient resources, etc).

2.5 Using the COMLDi function block

The COMLDi function block must be called-up after a cold or warm start. It operates in the same way as standard PL7-3 function blocks.

```
IF (SY0+SY1)  THEN RESET B0
IF NOT B0     THEN EXEC COMLD0 (⇒); SET B0
```

The internal data bit READY (COMLD0,READY bit) can be tested to determine if loading is complete.

The configuration can also be loaded using the SYSDIAG adjustment tool, by setting internal data bit LOAD to state 1.

It is advised to use the COM number to name the OFB block. For example, COMLD1 should be used for COM channel number 1.

2.6 Performance - Response time

Execution time and response time

Loading the configuration(s) from the TSX memory to the module memory requires :

- 6 master task cycles for the transfer of a single configuration (a single channel),
- 8 master task cycles for the transfer of 2 configurations (2 channels).

During loading, the COMLDi function block uses from 3 to 5 ms of CPU time, per cycle for a TSX 67.

Memory occupation

Program area	Data area	Constants area
2300 words for any number of uses	220 words for each use	8 words for each use

C

C

3.1 General

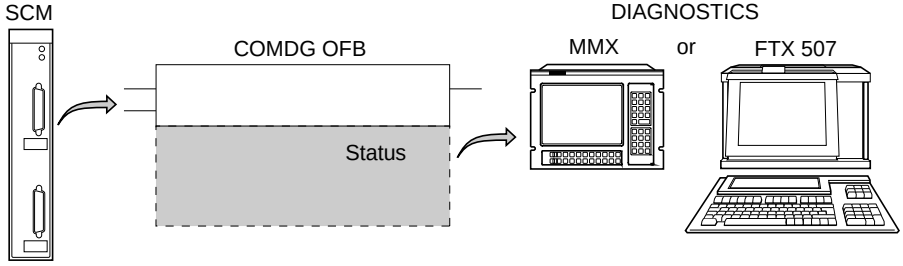
The COMDG OFB groups the fault information from the SCM module or the integrated UNI-TELWAY port :

- faults associated with the modules and detected during execution of the self-tests :
  - module absent or out of service,
  - module and I/O configuration codes are different,
  - terminal block fault.
- application faults associated with each channel :
  - communication fault on channel 0,
  - communication fault on channel 1.

The COMDG OFB is mainly intended to be used in conjunction with diagnostic software such as :

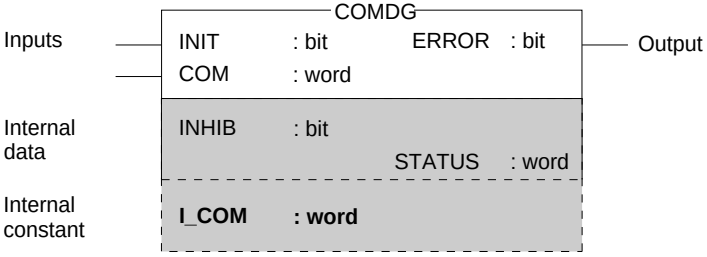
- APPLIDIAG under the Software Workshop,
- the DIAG function for PL7-MMI software.

For further information concerning these programs, see the corresponding manuals.



3.2 Introduction to the COMDG OFB

The COMDG OFB has 2 input parameters, an internal constant (defined in the programming phase) and internal data (used during execution). It has 1 output.



3.3 Description of the parameters

Inputs

Parameter	Type	Access	Description
INIT	bit	(2)	This bit reinitializes the OFB when it is set to 1. The ERROR output and the STATUS variables are set to 0. It can be read by the COMDGi,INIT mnemonic.
COM	word	(2)	It contains the logic number of the SCM module to which the OFB is assigned. On a cold restart or reconfiguration of the PLC, it is automatically initialized with the contents of the I_COM internal constant. It can be read or written to by the COMDGi,COM mnemonic.

Internal data

Parameter	Type	Access	Description
INHIB	bit	(4)	Monitoring of the module is stopped when this bit is at state 1, the ERROR output and the STATUS variables are set to 0. It can be read or written to by the COMDGi,INHIB mnemonic.
STATUS	word	(1)	This word contains the type of error associated with the module. It can be read by the COMDGi,STATUS mnemonic.

Output

Parameter	Type	Access	Description
ERROR	bit	(1)	This bit goes to 1 when an error is detected. It can be read by the COMDGi,ERROR mnemonic.

Internal constants

Parameter	Type	Access	Description
I_COM	word	(3)	This word specifies the number of the COM to which the OFB is assigned. It is between 0 and 63 (its default value is 64 which renders the OFB unusable if the parameter has not been initialized).

(1) Read when using program and adjust modes (debug mode, etc),
(2) Read when using program and adjust modes (debug mode, etc),
Written to when using adjust mode (debug mode, etc),
(3) Read when using adjust mode (debug mode, etc),
Written to when using program mode and by the [CONTENT] key,
(4) Read when using program and adjust modes (debug mode, etc),
Written to when using program and adjust modes (debug mode, etc).

3.4 List of faults

STATUS internal data

bit0	=	1	:	SCM module or UNI-TELWAY connection not functioning
bit1	=	1	:	C0 (SCM) adaptor fault or terminal block fault (UNI-TELWAY connection)
bit2	=	1	:	C1 adaptor fault
bit3	=	1	:	Communication fault - CHAN 0 -
bit4	=	1	:	Communication fault - CHAN 1 -
bit5			:	not used.
bit6			:	not used.
bit7			:	not used.
bit8			:	not used.
bit9			:	not used.
bit10	=	1	:	UNI-TELWAY module and TSX configuration are different
bit11	=	1	:	Type of module does not agree with I/O configuration
bit12	=	1	:	SCM module or UNI-TELWAY connection absent
bit13	=	1	:	Error accessing directory (missing, incoherent, etc.)
bit14	=	1	:	Communication error (exchange not possible with module)
bit15	=	1	:	System error (insufficient resources etc.)

3.5 Using the COMDG function block

Starting the OFB

The COMDG function block must be called-up just once after a cold start or reconfiguration of the PLC.

It executes automatically and cyclically. It is up to the user to organize that the OFB start instruction is scanned only once by the program, thus activating this instruction by an event which is only true during one cycle of the PLC.

Execution of the COMDG OFB

Once started, the OFB operates continuously in the monitoring task of the PLC. To stop this execution, the COMDGi,INHIB bit should be set to 1 by program.

Check during execution :

On a start command, the OFB checks:

- that the COM function is defined for the station,
- that the directory is present and correct (logic number of the module),
- that the module is present and ready to operate (module in start state and self-test complete),
- that the type of module is the same as that defined in the directory.

The COMDG OFB then reads the interface register to detect application faults.

> There is no need to assign a PL7-3 variable to the ERROR output, since this can be read directly using the COMDGi,ERROR mnemonic.
> However, a variable assigned to a block output is updated only once by the EXEC command.

Programming

The OFB must be started after a cold start.

```
! IF (SY0 + SY1) THEN RESET B0
! IF NOT B0 THEN EXEC COMDG1(;W0=>); SET B0
```

or
B0 = cold start detection bit,
W0 = COM number.

Acknowledgment of faults

> Faults arising in the module are memorized. The COMDGi,INIT input must be set to 1 to acknowledge the fault indicated by STATUS and to reinitialise the OFB. When a fault is detected, the OFB continues to monitor the module and to report the faults.

3.6 High performance - Response time

Execution time and response time

The modules are monitored about every 200ms. This time may extend to 1 second if the CPU of the programmable controller is heavily loaded.

Depending on the CPU load, execution time is between 500 µs and 4 ms.

The response time depends on the PL7 application. It varies from 1 to 5 cycles of the master task.

C

Memory occupation

Program area	Data area	Constant area
2000 words for any number of uses	150 words for each use	8 words for each use

1.1 Introduction

The optional communication function blocks are an **extension of the PL7-3 language**. The COMLDi and COMDGi communication function blocks (see divider C) enable the TSX SCMxxxx communication module to be set up : other functions blocks enable communication applications to be programmed. These are :

- SCMWRi : for exchanging formatted messages on the half-duplex or full-duplex channel of the SCM module,
- COMUWi : for request exchanges via UNI-TELWAY,
- UNITEi : for UNI-TE request exchanges via a MAPWAY or TELWAY network.

Important

You are advised to first read sections 1.2 and 1.3 in divider C. These two sections introduce the operating mode for configuring and programing OFBs.

D

On a TELWAY network, the size of a request is limited to 32 bytes.

1.2 Functions

The SCMWR function block simplifies the creation of communication programs for sending characters to a screen and/or a printer and receiving characters from a keyboard.
The SCMWR function block can also handle communication with another programmable system which has an asychronous serial communication line, for example a programmable controller or a microcomputer, etc.

The COMUW and UNI-TE function blocks are used for developing communication programs with equipment which supports UNI-TE protocol.

1.3 Example of an SCMWR communication application

The control system program shown in the following example is a program for controlling and regulating temperature; the PLC controls the cooking ovens.
A monitor screen is connected to the PLC via channel 1 on a TSX SCM 20 xxxx serial communication module.

The function required is as follows : if the temperature of one of the ovens exceeds a given threshold then the following message is displayed on the monitor screen :

Temperature oven N° : 3 = 120.35

The dynamic values 3 and 120.35 are the data words W10 and W11. These numerical values are converted and displayed in ASCII.

To display the fault messages automatically on the monitor screen simply requires :

- that application data words be initialized :

  W10 containing 12035; the temperature of the oven multiplied by 100.
  W11 containing the number of the oven at fault; in this case, W11 = 3.

  CW50 = M' "Temperature oven N°: "' : Display a character string.

  CW62 = M'I10 ' : Display a 1 character integer number.
  CW64 = M' "=" ': Display a character string.
  CW66 = M'I82e' : Display a decimal number as 8 digits with 2 digits after the decimal point.

  Note 1
  The syntax CWi = M'...' indicates that the word is displayed in "Message" (or character string) format.

  Note 2
  The character "e" acts as a separator at the end of the format.

- that the following instructions be programmed :

```
! SET SCMWR1,CHANNEL
! [W10 > 10000] --> B10
! IF RE(B10) THEN EXEC SCMWR1(;2;CW50;W11;W10.....=>)
```

Additional notes

Function block data :
The SCMWR1,CHANNEL variable is an internal data bit of the SCMWR function block. The instruction SET SCMWR1,CHANNEL transmits the message on channel N° 1 of the module.

Function block parameters :
They allow data exchanges between the function block itself and the rest of the application.
In order for the SCMWR function block to transmit the message, it is necessary to define :

- The number of the module selected, module N° 2 in the previous example,
- The format of the message (CW50),
- The number of the oven at fault, that is W11,
- The temperature of the oven at fault, that is W10.

D

1.4 Example of a COMUW communication application

The configuration of this new example resembles the previous configuration (cf 1.3). An additional monitoring PLC is connected to the process control PLC using channel 1 of a TSX SCM 21•• serial communication module. The control PLC acts as the UNI-TELWAY bus master, and the monitoring PLC is the slave at address 1.

The function required is as follows : if the temperature of the oven is greater than a maximum given threshold, then the control PLC must indicate to the monitoring PLC by writing the value 1 to word W100 of this PLC.

For this message to be transmitted on the UNI-TELWAY bus, a UNI-TE protocol request is required. To do this :

- Read the UTWREQ.SCY symbols file which defines a list of mnemonics associated with the names of the UNI-TELWAY requests, using the SDBASE software. For example :
  CW59 : Rd_w,
  CW95 : Wr_w.
- Read the initialization file of the UTWREQ.CST constant word in source format using PL7 software. This file allows the predefined UNI-TELWAY request codes to be initialized. For example : Wr_w = M'%1407%W01W01 e'.

Note
The UTWREQ.SCY and UTWREQ.CST files are supplied with PL7-COM. They are installed in the XPROSYS\OFB\COMM directory and must be imported using the IMPORT function in PL7-3 directory MOD.

Important

This import operation systematically reserves the constant words CW0 to CW114.

- Initialize the internal words of the control PLC application :
  W10 containing 12035. This is the temperature of the oven multiplied by 100,
  W11 containing the fault value. That is W11 = 1,
  W12 containing the address of the word of the monitoring PLC to which the fault is written, in this case W12 = 100.

Note 1
The syntax CWi = M'...' indicates that the word is displayed in "Message" (or character string) format.

Note 2
The character "e" acts as a separator at the end of the format.

- Program the instructions :

```
! 1 --> COMUW1,CHANNEL
! 1 --> COMUW1,SLAVE
! [W10 > 10000] --> B10
! IF RE(B10) THEN EXEC COMUW1(;1;Wr_w;W12;W11.....=>)
```

Additional commands

Function block data :

- The COMUW1,CHANNEL variable is an internal data bit of the COMUW function block. The instruction 1 → COMUW1,CHANNEL enables a message to be transmitted on channel N° 1 of the module,
- The COMUW1,SLAVE variable is an internal data bit of the COMUW function block. The instruction 1 → COMUW1,SLAVE enables a message to be transmitted to slave N° 1 on the UNI-TELWAY bus.

Function block parameters :

These allow data to be exchanged between the function block itself and the rest of the application.

In order for the COMUW function block to transmit the message, it is necessary to define :

- The number of the module selected, module N° 1 in the previous example,
- The message format (Wr_w),
- The address of the destination word in W12,
- The value to be written to the destination word contained in W11.

D

Sub-section **Page**

2.1 General

The SCMWR OFB enables communication programs and character exchange programs to be created with equipment having an asynchronous serial port. In the same way as other optional function blocks, the SCMWR OFB is an extension of PL7-3 language. It operates in transmission and/or reception mode with channels operating in Half or Full-duplex character string mode on TSX SCM 2●●● modules.

The principle of the SCMWR OFB is based on a simple one borrowed from high level languages such as BASIC, C, FORTRAN, etc.
The objective is to declare in a simple way a list of conversion codes to be applied to a list of variables. These conversion codes allow the variables to be converted to ASCII and to display the messages and the character strings.

The possible conversions and transfers are :

- Conversion of a word (16 bits) in decimal notation :[-]ddd.ddd,
- Conversion of a double word (32 bits) in decimal notation :[-]ddd.ddd,
- Transfer of characters in ASCII or hexadecimal format,
- Display of the system date and time.

The SCMWR function block is used to :

- transmit messages,
- receive a single message,
- transmit a message following the reception of characters.
  The size of the message in transmission mode is unlimited. The system is responsible for cutting a message into two or more parts and sending them successively. The size of the message is restricted to 255 characters in reception mode.

Notes
For transmitting a message, the communication format is a set of transmission codes.

Transmission Code 1	Transmission Code 2	•••••••	End of format

For a transmission followed by a reception, the format is a set of transmission codes, the code =, and then the character reception conversion code.

Transmission Code 1	Transmission Code 2	•••••••	Transmission Code n	=	Reception Code	End of format

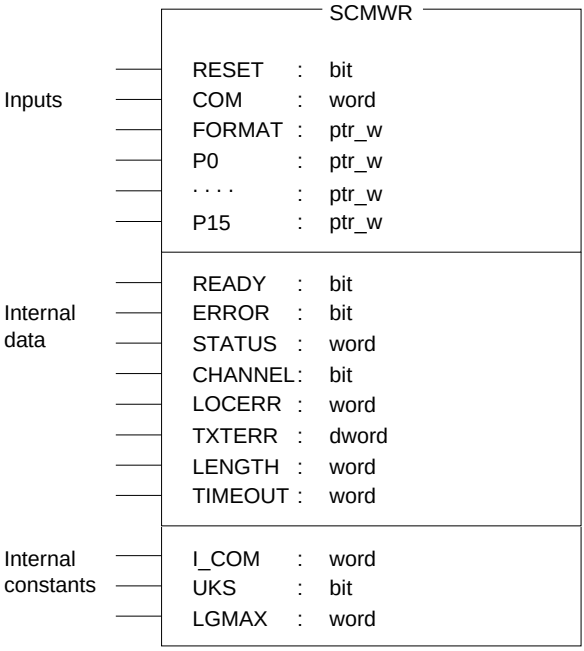
For simple character reception, the format is made up of the code = followed by a character reception conversion code.

=	Reception Code	End of format

2.2 Introduction to the SCMWRi OFB

The SCMWR type OFB has 19 input parameters, 3 internal constants and 8 internal data bits. It has no output parameters.
The input parameters, internal constants and internal data are defined during the programming phase.

2.2-1 General presentation



Bit or word type inputs may be values (binary for a bit, decimal for a word) or language data. The ptr_w type inputs are always words or constant words.
Example : EXEC SCMWR(;2;CW50;.....) or EXEC SCMWR(;W100;CW50;.....)

The SCMWR function block does not have an output parameter.

2.2-2 Input parameters

Parameter	Type	Access	Description
RESET	bit	(3)	This bit initializes the block and cancels the current exchange. Internal bits ERROR and STATUS are reset to 0.
COM	word	(3)	This word contains the COM word to which the OFB is assigned. On cold start of the PLC, COM is initialized with the contents of the internal constant I_COM.
FORMAT	word	(1)	This word contains the communication format for the conversions in the transmission buffer. Certain conversion codes do not require the input parameter Pi.
P0..P15	word	(1)	These words contain the input parameters associated with the communication format. Parameter P0 is associated with the 1st conversion code which requires a parameter Pi, and P1 with the 2nd code requiring a parameter Pi, etc.

D

(1) Read when using adjust mode (data mode, etc).
(2) Read when using program and adjust mode (data mode, etc).
(3) Read when using program and adjust mode (data mode, etc).
Written to when using adjust mode (data mode, etc).
(4) Read when using program and adjust mode (data mode, etc).
Written to when using program and adjust mode (data mode, etc).

To write in program mode, use the [PARAM] key.

2.2-3 Internal data

READY	bit	(2)	This bit indicates if the OFB is executed or not. It is set to 0 during execution of the OFB and returns to 1 when this is completed. This bit can be evaluated directly by reading SCMWRi, READY.
ERROR	bit	(2)	This fault bit is set to 1 when there is an error in the construction or transfer of the transmission buffer. It is reset to 0 after a RESET order for the SCMWRi block.
STATUS	word	(2)	This word enables the cause of the fault to be identified by setting the corresponding fault bit to 1 (see section 2.4).
CHANNEL	bit	(4)	This bit indicates the number of the channel to which the OFB is assigned. If this bit is at 1, it is channel 1; if this bit is at 0, it is channel 0. The default setting is 0.
LOCERR	word	(2)	This word contains the location of the first error encountered during analysis of the communication FORMAT. This data is used for debugging the application.
TXTERR	double word	(2)	This word contains the code number which caused the first FORMAT definition error.
LENGTH	word	(2)	This word contains the number of characters received during a character reception phase. If a parameter Pi cannot be read or written to, this word indicates the number of this parameter. This may be when character reception is into a CWi.
TIMEOUT	word	(4)	This word contains the maximum time allowed for execution of the OFB. That is to say to analyse the command and transmit the string to the module. It is expressed in multiples of 100 ms. The default setting is 0. A setting of 0 signifies an infinite transmission or reception time.

(1), (2), (3) and (4): refer to section 2.2-2.

D

2.2-4 Internal constants

Parameter	Type	Access	Description
I_COM	word		This word indicates the COM number to which the OFB is assigned on cold start or reconfiguration. Its value is from 0 to 63, the default value being 64 rendering the OFB unusable if the I_COM parameter has not been initialized.
UKS	bit		This bit indicates the display format for the date : UKS=1 01/31/1991 = 31 January 1991, UKS=0 31/01/1991 = 31 January 1991, UKS=1 3.14 UKS=0 3,14 The default value is UKS=1.
LGMAX	word		This word is used to declare the maximum size of a transmission line. It is also used for tabulation. The value of the word is between 10 and 140. The default value is 80.

To write in program mode, use the [CONTENT] key.

D

2.3 Communication format codes

List of conversion codes	
e	Marks end of format.
=	Code separator : Transmission and reception.
"....."	Display character string.
%.....%	Transmit a string of hexadecimal characters
t	Display system time.
d	Display system date.
I	Convert a word to decimal notation [-]ddd.ddd
F	Convert a double word to decimal notation [-]ddd.ddd
C	Transfer n characters.
c	Transfer a table of characters (ending with \).
A	Transfer index of n characters.
H	Transfer n characters - Hexadecimal display.
B	Transfer n bits - Display in bit format.
/	Skip line.
T	Tabulation from start of line.
P	Position cursor (VT100 protocol).
r	Clear screen (VT100 protocol).

Examples are given for the various communication codes. It is indicated for each code if a conversion input parameter is associated with that code, and if the exchange is authorized in transmission and reception or in transmission only.

e : Marks end of format

Message to be displayed	Application program
temperature fault	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'"temperature fault"' = M'e'

Definitions associated with the format

Structure of format-code

The code **e** marks the **end of format**. This code is used to stop analysis of the format.
All formats (single or a series of codes) **must end with the code e.**

Input parameters (Pi) : None.
Exchanges : Transmission and Reception.

= : Code separator : Transmission / Reception

Message to be displayed	Application program
Temperature acquisition	IF ... THEN EXEC SCMWR1 (; 2; CW50; W20; …) **Format** CW50 = M'"Temperature acquisition:"' = M'=I50' = M'e' **Input parameters :** The character string received is converted and stored in word W20.

Definitions associated with the format

Structure of format-code

The code **=** is used to indicate that the next code is a reception conversion code.
Only **one** reception conversion code is allowed.

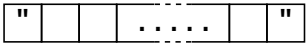
Where only reception of messages is required and not transmission, the first format code is **=** .

"..." : Display character string

Message to be displayed	Application program
temperature fault	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'"temperature fault"' = M'e'

Definitions associated with the format

Structure of format-code



The format contains a string of ASCII characters enclosed by **"..."** characters.
This is a string which is sent to the SCM module.

Input parameters (Pi) : None.
Exchanges : Transmission.

%...% : Transmit a string of hexadecimal characters

Message to be displayed	Application program
ALARMS	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'%DAC4%' = M'"ALARMS"' = M'%C4BF%' = M'e'

Definitions associated with the format

Structure of format-code

The format contains a string of hexadecimal characters enclosed between **%** characters.
Hexadecimal codes are sent to the SCM module.
Attention : the number of hexadecimal characters between the **%** characters **must be even.**

Input parameters (Pi) : None.
Exchanges : Transmission.

t : Display system time

Message to be displayed	Application program
09:10:05 temperature fault	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M't' = M'" temperature fault"' = M'e'

Definitions associated with the format

Structure of format-code

t

The system time in the PLC is converted to a string format

HH:MM:SS
- seconds
- minutes
- hours

Input parameters (Pi) : None.
Exchanges : Transmission.

d : Display system date

Message to be displayed	Application program
07/08/1991 start cycle	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'd' = M'" start cycle"' = M'e'

Definitions associated with the format

Structure of format-code

d

The system date in the PLC is converted to a string format

DD/MM/YY
- Year
- Month if the UKS constant = 0
- Day

MM/DD/YY
- Year
- Day if the UKS constant = 1
- Mois

Input parameters (Pi) : None.
Exchanges : Transmission.

I : Convert a word to a decimal value

Message to be displayed	Application program

`temperature = 123.45`
 	
8 digits with
2 after the decimal point
	`IF ... THEN`
	`EXEC SCMWR1 (; 2; CW50; W10;…)`
	Format
	`CW50 = M'"temperature="'`
	`= M'I82'`
	`= M'e'`

Definitions associated with the format

Structure of format-code

I	m	n

where m = total number of digits $1 \leq m \leq 8$.
n = number of digits after the decimal point $0 \leq n \leq 5$.

Input parameters (Pi) : This is the word to be converted to decimal.
Exchanges : Transmission and Reception.

F : Convert a double word to a decimal value

Message to be displayed	Application program

`temperature = 778.81`
 	
10 digits with
2 after the decimal point
	`IF ... THEN`
	`EXEC SCMWR1 (; 2; CW50; W10; …)`
	Format
	`CW50 = M'"temperature="'`
	`= M'F1002'`
	`= M'e'`
	Input parameters : Value is coded in words :
	W10 : Least significant, W11 : Most significant
	i.e. : W11 = 1 and W10 = 12345

Definitions associated with the format

Structure of format-code

F	m	n

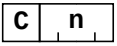
where m = total number of characters $1 \leq m \leq 13$.
n = number of digits after the decimal point $1 \leq n \leq 10$.
Input parameters (Pi) : This word contains the least significant bit of the item to be transcribed. The next word contains the most significant bit. For example if W10 = 12345 and W11 = 1 then the double word W11/W10 = 77881 and the display requested indicates : 778.81.
Exchanges : Transmission and reception.

C : Transfer n characters (Copy)

Message to be displayed	Application program
temperature fault	IF ... THEN EXEC SCMWR1 (; 2; CW50; W10; …) **Format** CW50 = M'C018' = M'e' **Input parameters:** W10 = M'temperature fault'

Definitions associated with the format

Structure of format-code



where n = number of bytes to read or write $1 \leq n \leq 999$.
Note : One reception is limited to 255 characters.

Input parameters (Pi) : This word contains the first two characters of the string to be transferred. The following characters are contained in the following words.
Exchanges : Transmission and Reception

C : Transfer a table of characters (copy)

Message to be displayed	Application program
temperature fault	IF ... THEN EXEC SCMWR1 (; 2; CW50; W10; …) **Format** CW50 = M'c' = M'e' **Input parameter :** W10 = M'temperature fault\'

Definitions associated with the format

Structure of format-code

This code causes the characters to be transferred starting with the first character of the input parameter to the end character : \.

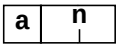
Input parameter (Pi) : This word contains the first two characters of the string to be transferred. The following characters are contained in the following words.
Exchanges : Transmission.

A : Transfer index of n characters

Message to be displayed	Application program
Tuesday	IF ... THEN EXEC SCMWR1 (; 2; CW50; W10; CW80; …) **Format** CW50 = M'A08e' **Input parameters(Pi) :** W10 is the access index to the table. In this case W10 = 1. CW80 is the message table. Each message has a fixed length of 8 characters. CW80 = M'MONDAY..' = M'TUESDAY.' = M'WEDNESDA' = M'THURSDAY' = M'FRIDAY..'

Definitions associated with the format

Structure of the format-code



where $1 \leq n \leq 80$ n defines the number of characters of each element of the table.

Input parameters (Pi) : The first associated parameter is the selection index of an element in a table. Index 0 points to the first element. The second parameter indicates the start of the table.
Exchanges : Transmission.

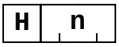
D

H : Transfer n characters - Hexadecimal display

Message to be displayed	Application program
Address fault = 01020F	IF ... THEN EXEC SCMWR1 (; 2; CW50; W10; …) **Format** CW50 = M'"Address fault="' = M'H003' = M'e' **Input parameters :** W10 = H'0201' W11 = H'000F'

Definitions associated with the format

Structure of format-code



where n = number of bytes to read or write $1 \leq n \leq 999$.
Note : One reception is limited to 255 characters.

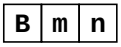
Input parameters (Pi) : This word contains the first two characters to be transferred. The following characters are contained in the following words. The characters are displayed in hexadecimal and start with the least significant bit. Example : W10 = H'0201' W11 = H'000F'.
A transfer of 3 characters causes 01020F to be displayed.
Exchanges : Transmission and Reception.

B : Transfer n bits - Display in bit format

Message to be displayed	Application program
Tool state = 101	IF ... THEN EXEC SCMWR1 (; 2; CW50; W10; …) **Format** CW50 = M'"Tool state="' = M'B031' = M'e' **Input parameters :** W10 = 5

Definitions associated with the format

Structure of format-code



where m = number of extracted bits to be displayed : $1 \leq m \leq 16$.
n = number of spaces displayed between each bit : $1 \leq n \leq 9$.

Input parameters (Pi) : The first extracted bit is the zero weighted bit or the input parameter data word. The extracted bits are the bits which follow in increasing order of significance. The bits are displayed in decreasing order of significance.
Exchanges : Transmission.

/ : Skip line - Position at start of next line

Message to be displayed	Application program
temperature fault pressure fault Each message is followed by a command : - Return to start of line (0D), - Jump line (0A).	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'"temperature fault"' = M'/' = M'"pressure fault"' = M'/' = M'e'

Definitions associated with the format

Structure of format-code

The characters 0D (carriage return) and 0A (jump line) are sent to the SCM module.

Input parameters (Pi) : None.
Exchanges : Transmission.

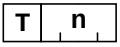
D

T : Tabulation from start of line

Message to be displayed	Application program
Alarm 12 Room 23 with the number of the alarm displayed in column 8 by 2 characters and the number of the room in column 17.	IF ... THEN EXEC SCMWR1 (; 2; CW50; W10; W11 …) **Format** CW50 = M'"Room Alarm"' = M'T008' = M'I20' = M'T017' = M'I20' = M'e'

Definitions associated with the format

Structure of format-code



where n = position relative to start of line : $1 \leq n \leq LGMAX$.
This code enables a display to be superimposed on a primary screen background.

Input parameters (Pi) : None.
Exchanges : Transmission.

P : Position cursor (VT100 protocol)

Message to be displayed	Application program
	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'P0111' = M'"Alarms"' = M'e'

Definitions associated with the format

Structure of format-code

P	l	c

where l = number of line to position cursor,
c = number of column to position cursor.
$1 \leq l \leq 25$ $1 \leq c \leq 80$

Input parameters (Pi) : None.
Exchanges : Transmission.

D

r : Clear screen (VT100 protocol)

Message to be displayed	Application program
	IF ... THEN EXEC SCMWR1 (; 2; CW50; …) **Format** CW50 = M'r' = M'e'

Definitions associated with the format

Structure of format-code

r

This code causes the transmission of a series of characters. The sequence is interpreted by a VT100 type terminal causing the screen to be cleared.

Input parameters (Pi) : None.
Exchanges : Transmission.

2.4 List of faults

Summary table of the OFB status as a function of the ERROR and READY bits

ERROR	READY	OFB status
0	0	Transmitting or receiving.
0	1	Transmission or reception completed correctly.
1	1	Transmission or reception completed incorrectly. Cause of fault is given in the STATUS word.
1	0	Normal status not possible.

STATUS internal data

bit0 = 1 : TSX SCM module not operating, absent or module type incoherent
bit1 = 1 : Module configuration is not half-duplex or full-duplex.
bit2 = 1 : Syntax error.
bit3 : not used.
bit4 : not used.
bit5 = 1 : Number of hexadecimal numbers even or zero.
bit6 = 1 : End of format character (e) absent.
bit7 = 1 : Error accessing PL7 objects: overflow, write not authorized
bit8 : not used.
bit9 : not used.
bit10 = 1 : Negative response.
bit11 = 1 : Error on maximum permissible transmission time (time-out).
bit12 = 1 : Exchange cancelled by power-break, a RESET command or software fault.
bit13 = 1 : Error accessing directory. The directory is not present or is incoherent, etc
bit14 = 1 : Communication error. Exchanges are not possible between the CPU of the programmable controller and the module.
bit15 = 1 : "System" error. Insufficient resources.

D

2.5 Using the SCMWR function block

To set up a communication application the following steps must be performed :

1 - Configure the module with the PL7-COM software tool
This tool enables the module to be configured in half-duplex or full-duplex mode.

When configuring the channel of the SCM module, it is recommended to define at least the reception mode stop character. The "ENTER" code (OD) is usually used.
Lastly, PL7-COM can check wiring and the quality of the communication using the TEST CHAN function.

2 - Define the FORMAT
The FORMAT is a word (Wi) or a constant word (CWi) which must be initialized. This data can be initialized using the adjust mode, the data mode or the constant mode. This data can be saved to disk. The display format used is the Message format.

Example : CW10 = M'"A text ..."'

To initialize constant words, it is practical to use a text editor.

Using a text editor (READ CW)
The various communication formats can be created automatically using a text editor. In effect, it is possible to create constant word files, (CW) in an ASCII type format.
Example : file TEXT.CST in the directory ...\STATION\PL7-3\MOD.

```
CW50 = M' "Temperature oven N°: " '
= M'I10 '
= M' "=" '
= M'I82e'
```

The Space or 0 characters in a FORMAT are non-significant characters.

The constant word file (TEXT.CST) can be re-read and inserted into an application using the read function "constant words in source format" (READ) in the ...\STATION\PL7-3\MOD directory. Thus formats defined once may be reused directly by other applications.

Using symbols or mnemonics

In the same way as other applications developed in the X-TEL Software Workshop environment, it is possible to define the symbols associated with data area words and constant words.

Using symbolic data improves legibility and flexibility of applications.

Declare in the symbol database :

Heat	B0
Tempe	W10
Num_Oven	W11
COM_2	CW40 Initialized with the value 2.
MSGDEF_1	CW50

The example becomes in this case :

```
! IF RE(Heat) THEN
      EXEC SCMWR1(;COM_2;MSGDEF_1;Num_Oven;Tempe;.....=>)
```

3 - Configure and program the OFB with the PL7 tool

PL7 is used to configure the SCM module, the SCMWR function block and the number of OFBs required by PL7.

The program mode is then used to define the input and internal constants of the OFB parameters.

Attention

It is recommended to make execution of the EXEC OFB conditional on a rising edge in order to avoid continuous transmission. For example :

```
! IF RE(Heat) THEN
      EXEC SCMWR1(;COM_2;MSGDEF_1;Num_Oven;Tempe;.....=>)
```

The fault message is then sent just once when the maximum authorized threshold is crossed.

2.6 Performance - Response time

Execution time and Response time

During execution of a function block (analysis of the format, transfer of data to the module, etc), i.e. when the bit READY = 0, the SCMWR function block uses 4 to 5 ms of CPU time per cycle.

Memory occupation

Program area	Data area	Constant area
5500 words for any number of uses	580 words per use	8 words per use

D

Sub-section **Page**

D

3.1 General

The COMUW and UNITE OFBs enable communication programs and data exchanges to be made with equipment having an asynchronous serial port and which support the UNI-TELWAY protocol. As with all other optional function blocks, the COMUW and UNITE OFBs are an extension of the PL7-3 language. They operate in reception and transmission mode using a TSX SCM 21•• communication module.

The COMUW OFB allows the exchange of messages over UNI-TELWAY using an SCM 21 module or by integrated UNI-TELWAY port. The UNITE OFB enables "UNITE" messages to be exchanged on a MAPWAY or TELWAY network.

The COMUW OFB provides **a simplified** address for designating a destination device on a "local" bus. In effect, only the number of the device need be indicated in the internal data SLAVE.

The UNITE OFB provides a complete address for designating a remote station on a single network or multinetwork architecture.

The principle of COMUW and UNITE OFBs is the same as that for the SCMWR OFB (see section 2).

D

The COMUW and UNITE function blocks are used to :

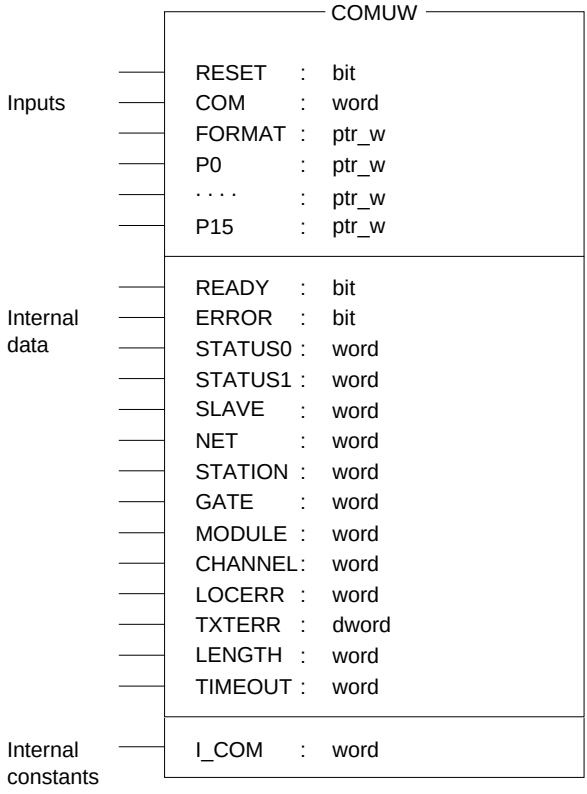
- transmit a UNI-TELWAY request,
- transmit a UNI-TELWAY request (question) and receive a response.

The size of messages in transmission and reception is limited to 128 bytes.

3.2 Introduction to the COMUW and UNITE OFBs

The COMUW type OFB has 19 input parameters, 1 internal constant and 14 internal data bits. It does not have an output parameter. The UNITE type OFB has 18 input parameters, no internal constant and 13 internal data bits. It does not have an output parameter. The input parameters, internal constants and the internal data are defined during the programming phase.

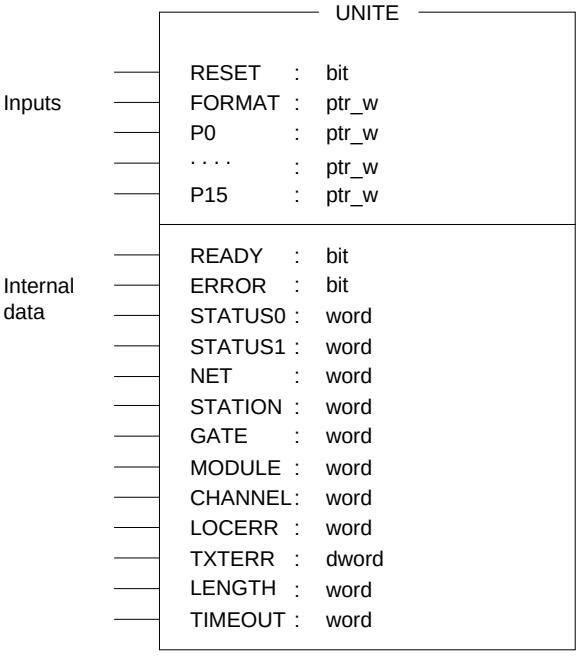
3.2-1 General presentation of the COMUW OFB



Bit or word type inputs can be values (binary for a bit, decimal for a word) or language data. The ptr_w type inputs are always words or constant words. Example : EXEC COMUW1(;1;CW50;.....) or EXEC COMUW1(;W100;CW50;.....)

The COMUW function block does not have an output parameter.

3.2-2 General presentation of the UNITE function block



The UNITE OFB does not have an input parameter to declare the logic number of the module (COM). The network station address is used for routing the message.

The UNITE OFB does not have an output parameter.

3.2-3 Input parameters

Parameter	Type	Access	Description
RESET	bit	(3)	This bit initializes the block and cancels the current exchange. Internal bits ERROR, STATUS0 and STATUS1 are reset to 0.
FORMAT	word	(1)	This word contains the communication format for making the conversion in the transmission buffer. Certain conversion codes do not required the input parameter Pi.
P0..P15	word	(1)	These words contain the input parameters associated with the communication format. The parameter P0 is associated with the 1st conversion code requiring a Pi parameter, and P1 with the 2nd code requiring a parameter Pi, etc.

Specific COMUW input parameters

Parameter	Type	Access	Description
COM	word	(3)	This word contains the COM number to which the OFB is assigned. On cold start of the PLC, COM is initialized with the contents of the internal constant I_COM.

(1) Read when using adjust mode (data mode, etc).
(2) Read when using program and adjust modes (data mode, etc).
(3) Read when using program and adjust modes (data mode, etc).
Write to when using adjust mode (data mode, etc).
(4) Read when using program and adjust modes (data mode, etc).
Write to when using program and adjust modes (data mode, etc).

To write in program mode, use the [PARAM] key.

D

3.2-4 Internal data

Parameter	Type	Access	Description
READY	bit	(2)	This bit indicates if the OFB is executed or not. It is set to 0 during execution of the OFB and returns to 1 when this is completed. This bit can be evaluated directly by reading COMUWi,READY or UNITEi,READY.
ERROR	bit	(2)	This fault bit is set to 1 when there is an error in the construction or transfer of the transmission buffer. It is reset to 0 after a RESET order for the COMUWi or UNITEi block.
STATUS0	word	(2)	This word enables the cause of a fault to be identified by setting the corresponding fault bit to 1 (see section 3.4).
STATUS1	word	(2)	This word enables the cause of a message being refused to be identified by setting the corresponding fault bit to 1 (see section 3.4).
LOCERR	word	(2)	This word contains the location of the first error encountered during analysis of the communication FORMAT. This data is used for debugging the application.
TXTERR	double word	(2)	This double word contains the value of the code which caused the first FORMAT definition error.
LENGTH	word	(2)	This word contains the number of characters received during a character reception phase.
TIMEOUT	word	(4)	This word contains the maximum time allowed for execution of the OFB, that is to say, for analyzing the command and transmitting the string to be sent to the module. It is expressed in multiples of 100 ms. The default setting is = 0. A setting of 0 signifies an infinite time for transmission or reception.

(1), (2), (3) and (4) refer to section 3.2-3.

D

Internal data specific to the COMUW OFB

1st case : UNI-TELWAY SLAVE configuration

Parameter	Type	Access	Description
SLAVE	word	(4)	This word contains the address of the destination device. SLAVE = 1..152 - Other BUS slave, SLAVE = 0 - BUS MASTER. Default setting = 153.

Note 1

When the SLAVE parameter is set to 0, there are five additional parameters for sending a message to a station on another network or to a device on another BUS connected to the MASTER equipment.

These additional parameters are :

NET : this parameter indicates the network number,
STATION : this parameter indicates the station number,
GATE : this parameter indicates the gate number,
MODULE : this parameter indicates the number of the destination module,
CHANNEL : this parameter indicates the number of the destination channel.

These are all Word type parameters and the access mode = 4.

Note 2

The parameter SLAVE = 254 allows access to the system gate of the slave SCM module.

2nd case : UNI-TELWAY MASTER configuration

SLAVE	word	(4)	This word contains the address of the destination device. SLAVE = 1..152 → A BUS slave SLAVE = 255 → This is a general broadcast message on the BUS. Default setting = 153.

Note 1

The parameter SLAVE = 254 allows access to the system gate of the master SCM module.

(1), (2), (3) and (4) refer to section 3.2-3.

Internal data specific to a UNITE OFB

Parameter	Type	Access	Description
NET	word	(4)	This word contains the network number of the destination application entity. Default setting = 0.
STATION	word	(4)	This word contains the station number of the destination application entity. Default setting = 254.
GATE	word	(4)	This word contains the gate number of the destination application entity. Default setting = 0.
MODULE	word	(4)	This word contains the module number of the destination application entity. Default setting = 0.
CHANNEL	word	(4)	This word contains the channel number of the destination application entity. Default setting = 1.

(1), (2), (3) and (4) refer to section 3.2-3.

D

3.2-5 Internal constants

Only the COMUW function block has an internal constant : I_COM.

I_COM	word		This word indicates the COM number to which the OFB is assigned on cold restart or reconfiguration. This setting is between 0 and 63. The default setting is 64, rendering the OFB unusable if the I_COM parameter has not been initialized.

To write in program mode, use the [CONTENT] key.

3.3 List of pre-defined request formats

Two files accompany PL7-COM. These are :

UTWREQ.SCY : This is a symbol definition file.
Each symbol is the name of a request format, for example CW100 : Wr_w.

UTWREQ.CST : This is an initialization file for constant words associated with these symbols.
Example : Wr_w = M'%1407%W01W01 e'

Reading these two files allows immediate access to the most recent UNI-TELWAY request formats. These pre-defined formats are described in the following table.

List of pre-defined request formats

Symbol	Function	Use	Size
Rd_w	Read word	EXEC COMUW1(;1;Rd_w;W10;W50...)	7 CW
Wr_w	Write word	EXEC COMUW1(;1;Wr_w;W10;W50...)	7 CW
Rd_obj	Read object	EXEC COMUW1(;1;Rd_obj;S_Wi;T_Wi; W10;W20;W30;W50....)	9 CW
Wr_obj	Write object	EXEC COMUW1(;1;Rd_obj;S_Wi;T_Wi; W10;W20;W30;W50....)	8 CW
Rd_sw	Read word SW	EXEC COMUW1(;1;Rd_sw;W10;W50...)	7 CW
Rd_dw	Read word DW	EXEC COMUW1(;1;Rd_dw;W10;W50...)	7 CW
Wr_dw	Write word DW	EXEC COMUW1(;1;Wr_dw;W10;W50...)	7 CW
Rd_comw	Read word COM	EXEC COMUW1(;1;Rd_comw;W1;W10;W30; W50;...)	10 CW
Rd_b	Read Bit Bi	EXEC COMUW1(;1;Rd_b;W10;W50;W51;…)	7 CW
Wr_b	Write Bit Bi	EXEC COMUW1(;1;Wr_b;W10;W50; ...)	6 CW
Rd_ioim	Read I/O	EXEC COMUW1(;1;Rd_ioim;W10;W11; W50;W51;W52;W53;…)	13 CW
Rd_sy	Read Bit SYi	EXEC COMUW1(;1;Rd_sy;W10;W50…)	5 CW
Nsoldata	Unsolicited data	EXEC COMUW1(;1;Nsoldata;W30;W50…)	4 CW
P_Run	RUN	EXEC COMUW1(;1;P_Run ...)	4 CW
P_Stop	STOP	EXEC COMUW1(;1;P_Stop ...)	4 CW

This list of formats may be extended.

The input parameters (Pi)

Pre-defined requests generally have input parameters to indicate the address of the object(s) manipulated, the number of objects manipulated, etc.

Rd_w
Wr_w
Rd_sw : EXEC COMUW1 (;1;Rd_w;W10;W50;…)
This request is used to read (or write) the word at the address given by W10. The result (or the value to be written) is indicated in W50.

Rd_obj
Wr_obj : EXEC COMUW1 (;1;Rd_obj;S_Wi;T_Wi;W10;W20;W30;W50;…)
This request is used to read (or write) a word or constant word at the address given by W10. The number of constant words exchanged is indicated in W20.
The size of the reception (or transmission) buffer is given by the parameter W30. Lastly, the result (or the values to be written) is indicated in W50 and the words which follow.

The UTWREQ.SCY and UTWREQ.CST files contain the initialized symbols associated with the constant words. These symbols are used by the Rd-obj and Wr-obj requests. The symbols are as follows :

S_com	CW0	Common words segment
S_Wi	CW1	Internal words segment
S_CWi	CW2	Constant words segment
S_Task	CW3	Tasks segment
S_sys	CW4	System objects segment
T_Wi	CW5	Signed integer type 16 bits
T_DWi	CW6	Signed integer type 32 bits
T_Task	CW7	Object type task

Rd_dw : EXEC COMUW1 (;1;Rd_dw;W10;W50;…)
This request is used to read the double word at the address given by W10. The result is indicated in W50 and W51.

Rd_comw : EXEC COMUW1 (;1;Rd_comw;W1;W10;W30;W50;…)
This request is used to read one or several words common to a station. The number of the station is given by W1, the number of the COM word to be read in W10 and the number of words in W30. The result is indicated in W50 and the words which follow.

Rd_b : EXEC COMUW1 (;1;Rd_b;W10;W50;W51;…)
This request is used to read the state of a bit Bi and to determine whether it is forced or not. The address of the bit Bi to be read is given by W10. The byte containing the requested bit is read and indicated in word W50. In the same way, the forcing state byte containing the requested bit is read and indicated in the word W51.

Wr_b : EXEC COMUW1 (;1;Wr_b;W10;W50;…)
The request is used to write the state of a bit. The address of the bit is given on W10. The state (0 or 1) is indicated in W50.

Rd_Sy : EXEC COMUW1 (;1;Rd_Sy;W10;W50;…)
This request is used to read the state of the system bit at the address given by W10. The byte containing the bit requested is read and indicated in word W50.

D

Rd_ioim : EXEC COMUW1 (;1;Rd_ioim;W10;W11;W50;W51;W52;W53;…)
This request is used to read the state of the I/O. The address of the I/O is given as : W10 = module number and W11 = rack number. The results are indicated in the parameters :
W50 = Fault status (status),
W51 = Configuration (Conf),
W52 = Values of inputs/outputs,
W53 = Values of forcing states.

Nsoldata : EXEC COMUW1 (;1;Nsoldata;W30;W50;…)
This request is used to transmit unsolicited data. The number of data bits to be transmitted is given by W30. The value of the data is indicated in word W50 and the words which follow.

3.4 List of faults

Summary table of the OFB status as a function of the ERROR and READY bits

ERROR	READY	OFB status
0	0	Transmitting or receiving
0	1	Transmission or reception completed correctly.
1	0	Normal status not possible.
1	1	Transmission or reception completed incorrectly The cause of the fault is given by the words STATUS0 and STATUS1.

Internal data STATUS0 for COMUW

bit0 = 1 : TSX SCM not operating, absent or module type incoherent.
bit1 = 1 : Configuration of channel 1 of the module is not UNI-TELWAY.
bit2 = 1 : Syntax error.
bit3 = 1 : Number of characters generated is greater than 128.
bit4 = 1 : Character received cannot be used by the FORMAT.
bit5 = 1 : Number of hexadecimal characters even or zero.
bit6 = 1 : End of format characters (e) absent.
bit7 = 1 : Error accessing PL7 objects : overflow, write not authorized.
bit8 = 1 : Address Ad1 or Ad2 not configured.
bit9 = 1 : Parameter SLAVE incorrect.
bit10 = 1 : Negative response.
bit11 = 1 : Message refused. See STATUS1.
bit12 = 1 : Exchange cancelled by power break, a RESET order or a software fault.
bit13 = 1 : Error accessing directory. The directory is not present or incoherent, etc,
bit14 = 1 : Communication error. Exchanges are not possible between the CPU of the programmable controller and the module.
bit15 = 1 : "System" error. Insufficient resources.

D

Internal data STATUS1 for COMUW

bit0			: not used.
bit1			: not used.
bit2	=	1	: Destination not accessible.
bit3	=	1	: Line error.
bit4	=	1	: Length error.
bit5			: not used.
bit6	=	1	: Address error.
bit7	=	1	: Request code unknown
bit8			: not used.
bit9	=	1	: Time-out exceeded.
bit10			: not used.
bit11			: not used.
bit12			: not used.
bit13			: not used.
bit14			: not used.
bit15			: not used.

D

Internal data STATUS0 for UNITE

bit0			: not used.
bit1			: not used.
bit2	=	1	: Syntax error.
bit3	=	1	: Number of characters generated greater than 128.
bit4	=	1	: Characters received cannot be used by FORMAT.
bit5	=	1	: Number of hexadecimal characters even or zero.
bit6	=	1	: End of format character (e) absent.
bit7	=	1	: Error accessing PL7 objects : overflow, writing not authorized.
bit8			: not used.
bit9			: not used.
bit10	=	1	: Negative response.
bit11	=	1	: Message refused. See STATUS1.
bit12	=	1	: Exchange cancelled by power break, a RESET command or software fault
bit13			: not used.
bit14	=	1	: Communication error.
bit15	=	1	: "System" error. Insufficient resources.

Internal data STATUS1 for UNITE

bit0	=	1	: Insufficient "Bus" resources.
bit1	=	1	: Insufficient "Line" resources.
bit2	=	1	: Destination not accessible.
bit3	=	1	: Line error.
bit4	=	1	: Length error.
bit5	=	1	: Network error.
bit6	=	1	: Address error.
bit7	=	1	: Request code unknown.
bit8	=	1	: Insufficient CPU resources.
bit9	=	1	: Time-out exceeded.
bit10			: not used.
bit11			: not used.
bit12			: not used.
bit13			: not used.
bit14			: not used.
bit15			: not used.

D

3.5 Using the COMUW and UNITE function blocks

To set up a communication application between devices or stations using UNI-TELWAY protocol, the following steps must be followed :

1 - Configure the module with the PL7-COM software tool (when using an SCM 21 xxx module) :

This tool can be used for configuration in UNI-TELWAY master or slave mode.

Note

A module connected to the bus takes its state, master or slave, and its slave number from the wiring in the connection box : TSX SCA xxx.
Lastly, PL7-COM enables the wiring and quality of the communication to be checked using the TEST CHANNEL function.

D

2 - Define the FORMAT :

The FORMAT is a word (Wi) or constant word (CWi) which must be initialized.

To create the predefined request formats, two files are supplied with PL7-COM. These are UTWREQ.SCY and UTWREQ.CST which are installed under the XPROSYS\OFB\COMM directory and must be imported using the IMPORT function in the PL7-3 directory MOD.

Important

This import operation systematically reserves the constant words CW0 to CW114.

The UTWREQ.CST file allows the set of predefined requests to be created.

When only one particular request is required, it is possible to create this one request using the associated file, for example : Rd_W.CST

Using the symbols or mnemonics
In the same way as for other applications developed in the X-TEL software environment, it is possible to define the symbols associated with words and constant words in the data area.

Using symbolic data improves readability and portability of the applications.

3 - Configure and program the OFB with the PL7 tool :

PL7 software is used to configure both the SCM or MAP communication modules, the COMUW or UNITE function blocks and the number of instances required by PL7.

Program mode is then used to define the input parameters and internal constants of the OFB.

Attention
It is advisable to make the execution of the EXEC OFB instruction conditional on a rising edge to avoid continuous transmission. For example :

Val_Def : W11 contains the default value, here W11 = 1.
Adr_Def : W12 contains the address of the word of the destination PLC to which the fault must be inscribed, here W12 = 100.

```
! IF RE(Heat) THEN
      EXEC COMUW1(;1;Wr_w;Adr_Def;Num_Def.....=>)
```

The fault message is then sent once when the maximum authorized threshold is crossed.

3.6 Performance - Response time

During execution of a function block (analysis of the format, transfer of data to the module, etc), i.e. when the bit READY = 0, the COMUW and UNITE function blocks use 4 to 5 ms of CPU time per cycle.

D

Memory occupation - COMUW

Program area	Data area	Constant area
3750 words for any number of uses	400 words per use	8 words per use

Memory occupation - UNITE

Program area	Data area	Constant area
2900 words for any number of uses	380 words per use	8 words per use

3.7 Creating new request formats

The format files UTWREQ.SCY and UTWREQ.CST are used to transmit predefined requests contained in these files.

These are the most recent UNI-TELWAY requests.

To create new requests, new formats must be defined. To do this, a list of conversion codes to be applied to a list of PL7 objects must be declared. The conversion codes enable the application variables to be put into the form of a UNI-TELWAY frame, to transmit this frame and, in the case of a question with response, to wait for a response.

The possible CONVERSIONS and TRANSFERS are :

- Transfer of the least significant byte of a PL7 object (Wi, CWi, DWi, CDWi) to a transmission buffer byte in the module and vice versa,
- Extraction of a set of PL7 object bits (Wi, CWi, DWi, CDWi) to a transmission buffer byte in the module,
- Transfer of a set of PL7 objects (Wi, CWi, DWi, CDWi) to a transmission buffer byte in the module and vice versa,
- Transfer of ASCII or hexadecimal characters.

D

Notes

When transmitting a message, the communication format is a series of transmission codes.

Transmission Code 1	Transmission Code 2	•••••••	End of format

For a transmission followed by a reception, the communication format is a series of transmission codes, followed by the code "=", and finally conversion codes "on reception of characters".

Transmission Code 1	•••••••	Transmission Code n	=	Reception Code 1	•••••••	Reception Code n	End of Format

3.7-1 List of conversion codes

List of conversion codes	
e	Marks end of format
b	Transfer the least significant byte of a word.
W	Transfer n words - n is coded in the format.
w	Transfer n words - n is an input parameter Pi.
L	Transfer n double words - n is coded in the format.
l	Transfer n double words - n is an input parameter Pi.
S	Transfer n characters - n is coded in the format.
s	Transfer n characters - n is an input parameter Pi.
X	Transfer a string of bits.
q	End of transfer of bits to a single word.
"....."	Transmit an ASCII character string.
%.....%	Transmit a hexadecimal character string.
=	Code separator : Transmission and Reception.
n	The nul format -skip a character in reception mode.

Examples are given for the various communication codes. It is indicated, for each code, if a conversion input parameter is associated with that code, and if the exchange is authorized for transmission and reception or for transmission only.

e : Marks end of format

Message to be exchanged	Application program
Request : RUN 24 | 07	IF ... THEN EXEC COMUW1 (; 1; CW50; …) **Format** CW50 = M' %2407% e'

Definitions associated with the format

Structure of format-code

e

The code **e** marks the **end of format**. This code is used to stop analysis of the format
All formats (single or a series of codes) **must end with the code e.**

Input parameters (Pi) : None.
Exchanges : Transmission and Reception.

b : Transfer the least significant byte of a word

Message to be exchanged	Application program
Request : RUN 24 | 07	IF ... THEN EXEC COMUW1 (; 1; CW50; W10; W11; …) **Format** CW50 = M'bbe' **Input parameters (Pi) :** W10 = H'0024' Question code W11 = H'0007' Sender category

Definitions associated with the format

Structure of format-code

b

This code copies the least significant byte of a word to the transmission buffer of the module.

Input parameters (Pi) : This is the word or constant word (transmission) from which the least significant byte is extracted and sent to the module.
Exchanges : Transmission and Reception.

W : Transfer n words - n is coded in the format

Message to be exchanged	Application program
Request : RUN 24 | 07	IF ... THEN EXEC COMUW1 (; 1; CW50; W10; …) **Format** CW50 = M' W01 e' **Input parameters Pi :** W10 = H'0724' Question code and sender category **Attention** : The least significant/most significant order is reversed.

Definitions associated with the format

Structure of format-code

W	n

Where 1 ≤ n ≤ 64, n defines the number of words to be exchanged with the module

Input parameters (Pi) : This is the first word of the exchange. The following words exchanged are the following words in memory.
Example : to exchange 31 words, M'W31'.
Exchanges : Transmission and Reception.

W : Transfer n words - n is an input parameters Pi

Message to be exchanged	Application program
Request : RUN 24 | 07	IF ... THEN EXEC COMUW1 (; 1; CW50; W10; W20; …) **Format** CW50 = M'we' **Input parameters Pi :** W10 = 1 Number of words W20 = H'2407' Value of word to be transferred **Attention** : The least significant/most significant order is reversed.

Definitions associated with the format

Structure of format-code

W

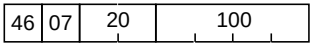
This code enables word or constant words (transmission) to be exchanged between the module and the PL7 object memory.

Input parameters (Pi) : The first associated parameter indicates the number of words to be exchanged. The second associated parameter is the first word or constant word to be exchanged.
Exchanges : Transmission and Reception

D

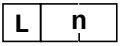
L : Transfer n double words - n is coded in the format

Message to be exchanged	Application program

Write the value 100 to double word DW20 of the destination device,
 
 	
Note : The value 100 is sent from DW80.
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10; w80; …)`
	Format
	`CW50 = M' %4607% W01L01 e'`
	Input parameters (Pi) :
	W10 = Address of destination double word, here W10 = 20
	W80 = Value to be transferred ; the value is coded on DW80 or W80 and W81.

Definitions associated with the format

Structure of format-code



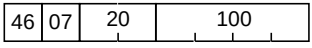
Where $1 \leq n \leq 32$ n defines the number of words to be exchanged with the module.

Input parameters (Pi) : The associated parameter indicates the address of the first word to be exchanged. For example, if the parameter is W80, the double word DW80 (that is W80 and W81) is transferred.

Exchanges : Transmission and Reception

l : Transfer n double words- n is an input parameter Pi

Message to be exchanged	Application program

Write the value 100 to double word DW20 of the destination device,
 
 	
Note : The value 100 is sent from DW80.
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10; W20; W80…)`
	Format
	`CW50 = M'%4607% W01 l e'`
	Input parameters (Pi) :
	W10 = Address of destination double word, here W10 = 20
	W20 = Number of double words to be exchanged, here W20 = 1
	W80 = value to be transferred, the value is coded on DW80, or W80 and W81.

Definitions associated with the format

Structure of format-code

This code enables the exchange (transmission) of double words or double constant words between the module and the PL7 object memory.

Input parameters (Pi) : The first associated parameter indicates the number of double words to be exchanged, the second associated parameter is the first double word, or double constant word, to be exchanged.

Exchanges : Transmission and Reception

S : Transfer n characters (bytes) - n is coded in the format

Message to be exchanged	Application program
Request : RUN 24 | 07	IF ... THEN EXEC COMUW1 (; 1; CW50; W10; …) **Format** CW50 = M' S002 e' **Input parameters (Pi) :** W10 = H'0724' Question code and sender category. **Attention** : The least signif/most significant order is reversed.

Definitions associated with the format

Structure of format-code

S	n

Where 1 ≤ n ≤ 128 n defines the number of characters (bytes) to be exchanged with the module.

Input parameters (Pi) : This is the first word from which the first byte to be exchanged is extracted. Bytes are transmitted in the order : least significant then most significant.
Exchanges : Transmission and Reception.

S : Transfer n characters (bytes) - n is an input parameters Pi

Message to be exchanged	Application program
Request : RUN 24 | 07	IF ... THEN EXEC COMUW1 (; 1; CW50; W10; W20 …) **Format** CW50 = M'se' **Input parameters (Pi) :** W10 = 2 Number of characters W20 = H'0724' Value of characters to be displayed **Attention** : The least signif/most significant order is reversed.

Definitions associated with the format

Structure of format-code

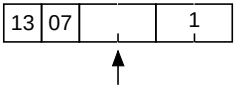
S

This code allows the characters to be exchanged between the modules and the PL7 object memory PL7.

Input parameters (Pi) : The first associated parameter indicates the number of characters (bytes) to be exchanged, the second associated parameter is the first word from which the first byte to be exchanged is extracted. Bytes are transmitted in the order, least significnt then most significant.
Exchanges : Transmission and Reception.

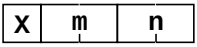
X : Transfer string of bits to a word

Message to be exchanged	Application program

The object is to activate output 0 of module 1 of the destination device. This is a write-io-module request.
 
 	
The module number is coded on bits : 0 … 2
The rack number is coded on bits : 3 … 5.
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10; W20; W30…)`
	Format
	`CW50 = M' %1307% X0002 X0305 W01 e'`
	Input parameters (Pi) :
	W10 = Module number coded on bits : 0 … 2
	W20 = Rack number coded on bits : 3 … 5
	W30 = Value of module bits, when W30 = 1 for output 0

Definitions associated with the format

Structure of format-code

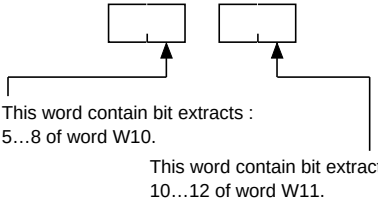


Where 0 ≤ m ≤ 15 and 0 ≤ n ≤ 15 and m defines the first bit to be transmitted to the module, and n defines the last bit to be transmitted to the module
The position of these bits is **unchanged**. This code makes a transfer **without shifting**.

Input parameters (Pi) : This is the word which contains the bits to be extracted
Exchanges : Transmission.

q : End of transfer of bits to a single word

Message to be exchanged	Application program


 	
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10; W11 …)`
	Format
	`CW50 = M'X0508 q X1012 e'`
	Input parameters (Pi) :
	W10 = Word containing bits : 5…8
	W11 = Word containing bits : 10…12.

Definitions associated with the format

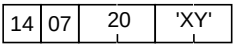
Structure of format-code

The code **X** transfer a string of bits to the module buffer. Several consecutive **X** codes allow transfer of bits to the same buffer word. The code **q** points to the **next word** in the buffer.The **X** codes following the code **q** causes the bit to be transferred to **the next word**.

Input parameters (Pi) : None.
Exchanges : Transmission and Reception.

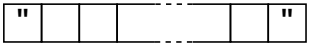
"..." : Transmit an ASCII character string

Message to be exchanged	Application program

Write message 'XY'
to word 20 of
the destination device
 
 	
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10 …)`
	Format
	`CW50 = M' %1407% W01 "XY" e'`
	
	`when W10 = 20`

Definitions associated with the format

Structure of format-code

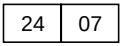


The format contains a string of ASCII characters enclosed by **"..."** characters.
This string is sent to the communication module.

Input parameters (Pi) : None.
Exchanges : Transmission.

%...% : Transmit a string of hexadecimal characters

Message to be exchanged	Application program

Request : RUN
 	
 
 	
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; …)`
	Format
	`CW50 = M'%2407% e'`

Definitions associated with the format

Structure of format-code

The format contains a string of hexadecimal characters enclosed in **%** characters .
The hexadecimal codes are sent to the communication module.
Attention : The number of hexadecimal characters between **%** characters **must be even**.

Input parameters (Pi) : None.
Exchanges : Transmission.

D

= : Code separator Transmission / Reception

Message to be exchanged	Application program

Read word W100
of destination device
 	
Question : 04 | 07 | 100
 	
Response : Value
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10; W20 …)`
	
	Format
	`CW50 = M' %0407% W01 = W01 e'`
	where
	W10 = Address of word to be read in destination device, in this case W10 = 100.
	W20 = Value read on return.

Definitions associated with the format

Structure of format-code

=

The code **=** is used to indicate that the codes which follow are reception conversion codes. The codes which follow enable the data in the module buffer to be extracted for arranging in the PL7 objects.

D

n : NUL Format

Message to be exchanged	Application program

Read object W100
of the destination device
 	
Question : 36 | 07 | 68 | 07 | 100 | 1
Object address
Number of objects
 	
Response : Value
Type of object
	`IF ... THEN`
	`EXEC COMUW1 (; 1; CW50; W10; W20; W30…)`
	
	Format
	`CW50 = M'%36076807% W01W01 = nW01 e'`
	where
	W10 = 100
	W20 = 1

Definitions associated with the format

Structure of format-code

n

The code **n** enables a character (byte) of the module buffer to be "skipped" during reception of messages.
During transmission this code sends a nul character to the buffer of the module.

3.7-2 Pre-defined request formats

The most recent request formats are given in the UTWREQ.CST file.

Rd_w	=	M'%0407%W01=W01e'	Read word
Wr_w	=	M'%1407%W01W01 e'	Write word
Rd_obj	=	M'%3607%bbW01W01=nse'	Read object
Wr_obj	=	M'%3707%bbW01W01se'	Write object
Rd_sw	=	M'%0607%W01=W01e'	Read system word
Rd_dw	=	M'%4007%W01=L01e'	Read double word
Wr_dw	=	M'%4607%W01L01 e'	Write double word
Rd_comw	=	M'%0707%W01W01=bW01e'	Read common word (COM)
Rd_b	=	M'%0007%W01=bbe'	Read bit
Wr_b	=	M'%1007%W01b e'	Write bit
			Read I/O module memory image
Rd_ioim	=	M'%0207%X0002X0306=bbW01W01e'	
Rd_sy	=	M'%0107%b=be'	Read system bit
Nsoldata	=	M'%FC07%se'	Transmit unsolicited data
P_Run	=	M'%2407%e'	Transmit RUN command
P_Stop	=	M'%2507%e'	Transmit STOP command

Using a text editor (READ CW)

The various request formats for the UTWREQ.CST file can be extended using a text editor.

The constant word file can be read and inserted into an application using the read function "constant words in source format" (READ).

D

1.1 Introduction

PL7-COM V5 software enables TSX SCM modules to be set up on TSX/PMX V4 PLC stations.

In this case, setting up a V4 station under PL7-COM V5 is the same as the setup procedure under PL7-COM V4.

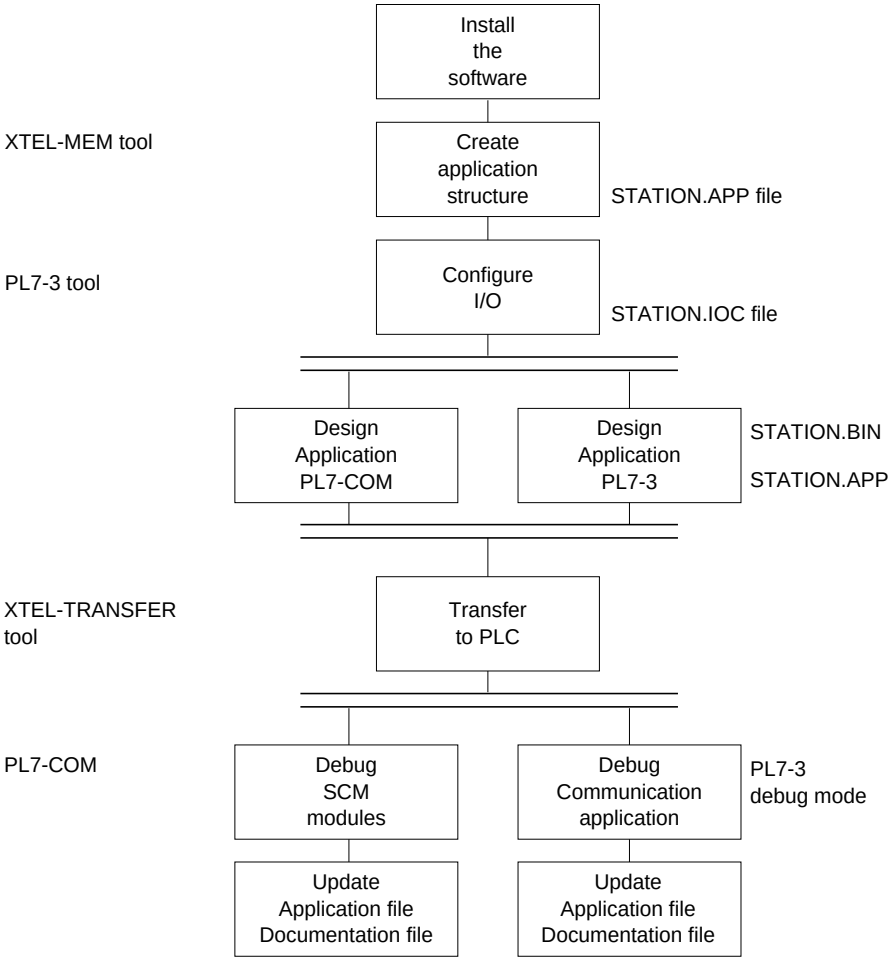
The operation corresponds to that described in the PL7-COM V4 programming manual, reference TXT DM PL7 CMM V4●.

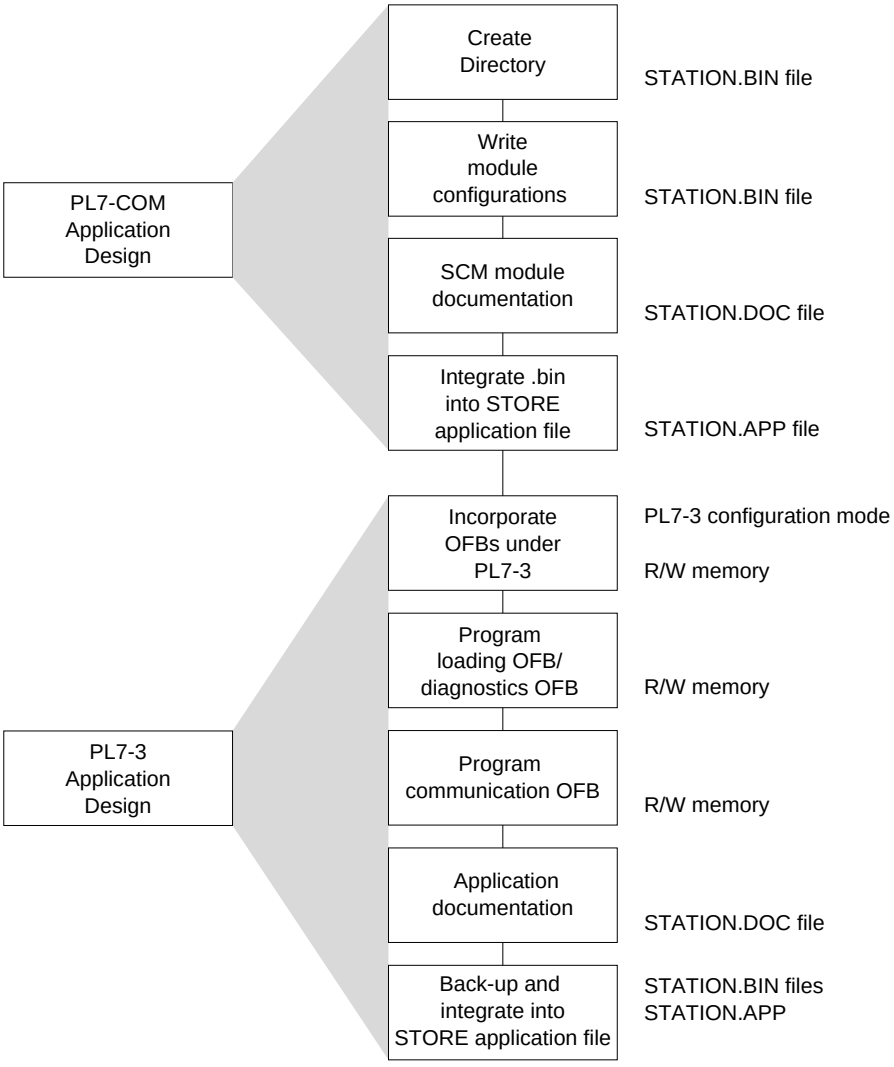
The purpose of this Appendix is to summarize the V4 setup and to point out the developments between a V4 station and a V5 station setup.

For further information, refer to the PL7-COM V4 programming manual, reference TXT DM PL7 CMM V4●.

1.2 Setup methodology

The following methodology is intended as a guide to the user when creating, debugging, archiving and documenting a communication application. This methodology refers to each operation without going into detail about the operations involved.




PL7-COM Application Design
PL7-3 Application Design
Create Directory
STATION.BIN file
Write module configurations
STATION.BIN file
SCM module documentation
STATION.DOC file
Integrate .bin into STORE application file
STATION.APP file
Incorporate OFBs under PL7-3
PL7-3 configuration mode
R/W memory
Program loading OFB/ diagnostics OFB
R/W memory
Program communication OFB
R/W memory
Application documentation
STATION.DOC file
Back-up and integrate into STORE application file
STATION.BIN files
STATION.APP

E

1.3 Screen sequences

Screen sequences and the choice of operating modes are the same in versions V4 and V5.

The first screen in the PL7-COM software enables the user to access all functions available in this software. The screen comprises one to three columns :

- The lefthand column selects the working memory : module, PLC or disk.
- The middle column selects the operating mode : configuration, debug, transfer and documentation.
- The righthand column is optional and is used to select a sub-operating mode in either transfer or debug mode.

Only one column contains the cursor and this is the active column.

The up and down cursor arrows **<↑>** and **<↓>** move the cursor within the active column. The left and right cursor arrows **<→>** and **<←>** move the cursor from one column to the next.

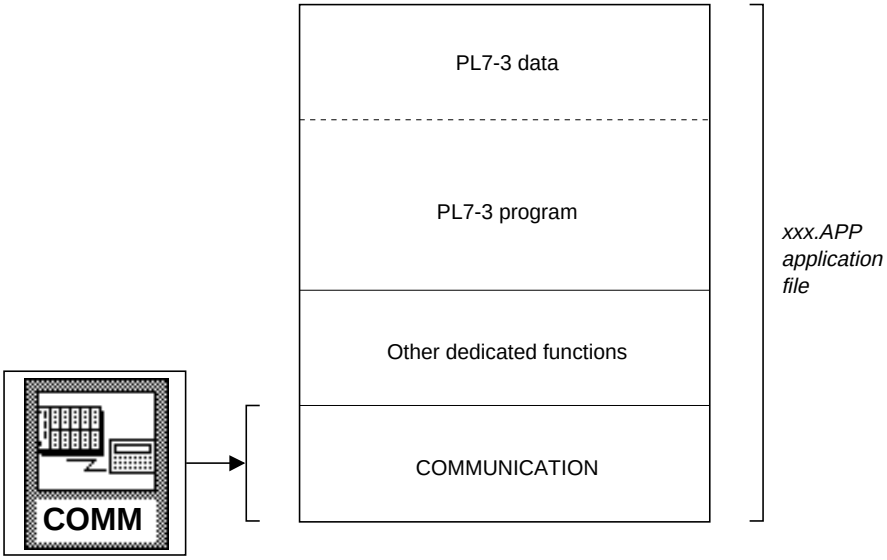
<Enter> confirms all selections.

E

1.4 Link with the PLC memory

Dedicated communication zone of the PLC memory

If the COMM function is declared at a station level, a dedicated communication zone is automatically created when the xxx.APP file is created by the XTEL-MEM tool. The size of this zone, fixed by default by the XTEL-MEM tool, can be modified by the user. The position of the zone is determined by the size of the adjoining dedicated PL7-3 zones and other dedicated functions.



Contents of the communication zone

When the xxx.APP file is created (PLC memory image) by the XTEL-MEM tool, this zone is empty, ready to be filled by PL7-COM (1). It comprises :

- the directory, consisting of :
  - a correspondance table between the logic numbers (0 to 63) and the physical positions of the modules in the racks. The software offers default assignments (2) which may be modified,
  - a table which gives the start address and the size of the configurations stored in the dedicated zone.
- the configurations, in ascending order of logic number.

(1) *Providing the xxx.APP file or the PL7-3 application contains at least the I/O configuration and the slots occupied by the TSX SCM modules.*

(2) *Ascending numbering from 0 to 63 in the order of the modules in the PL7-3 configuration.*

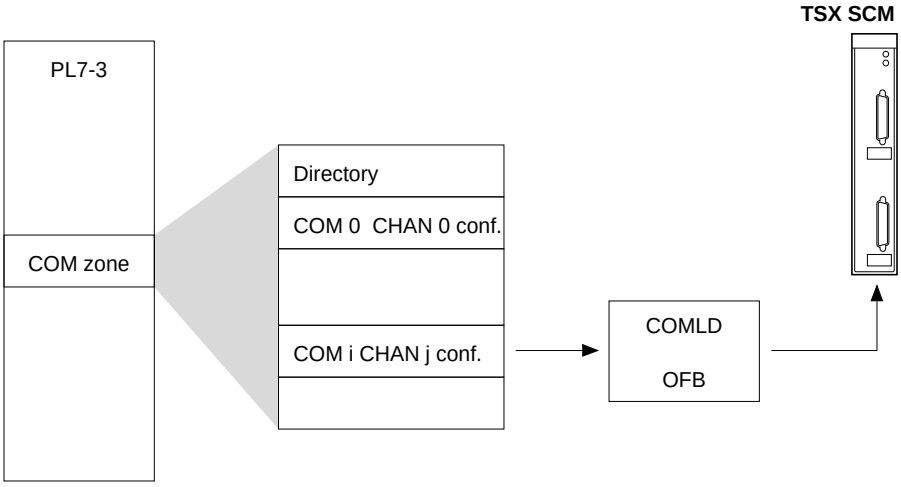
E

Dedicated COM zone

Correspondence table between the configuration n° and module location	*Directory*
Address and size of stored configurations	
COM0 CHAN 0 configuration	
COM0 CHAN 1 configuration	
COM1 CHAN 0 configuration	
COMn CHAN 0 configuration	
COMn CHAN 1 configuration	

This zone contains the information which can be accessed by the PL7-COM functions concerning its organization and by the communication OFB blocks. A compacting function is used to optimize the contents. A copy of this dedicated COM zone is stored in the xxx.BIN file under the COMM\APPLI directory on the hard disk (or diskette).

A configuration stored in this zone can be transferred to the TSX SCM xxx by the COM OFB.



1.5 Differences between PL7-COM V4 and PL7-COM V5

PL7-COM V4 software can control several xxxx.BIN applications in the X-TEL applications database.
Therefore, to store or load an application in the memory, the application name must be indicated.
PL7-COM V5 software uses a single application name : COMM.BIN.

In **connected mode**, PL7-COM V4 software has the following specific soft keys :

[STORE] displays a screen enabling storage of the contents of the dedicated COM zone to disk, in the form of an xxxx.BIN file stored in the communication applications zone.
With this store screen, the **[FILE]** key is offered and enables modification of the name of the destination file.

[RETRIEVE] displays a screen enabling the transfer of the contents of an xxxx.BIN file previously saved on disk, to the dedicated COM zone in the PLC memory.
With this transfer screen, the **[FILE]** key is also offered and enables modification of the name of the file containing the application.

Note :

The **[STORE]** and **[RETRIEVE]** keys in version V4 have been replaced by the **[STA→DSK]** and **[DSK → STA]** keys in version V5. The **[FILE]** key no longer exists.

In **local mode**, PL7-COM V4 software uses the **[.BIN]** and **[STORE]** keys to select an application or to save an application to disk.

[.BIN] enables selection of the working configuration (.BIN files). By default, this is the station name.
With this screen, the **[DIR.BIN]** key is also offered and accesses the list of .BIN files which can be selected.

[STORE] displays a screen enabling the storage to disk of the local application, in the form of an xxxx.BIN file stored in the communication applications zone.

Note:

The **[STORE]** and **[.BIN]** keys in version V4 have been replaced by the **[STORE]** and **[RETRIEVE]** keys in version V5. In V5, these keys allow a save to be performed using file names which have been chosen by the user.

When an xxxx.BIN file is being modified, PL7-COM V4 software uses the XTEL-MEM tool to update the xxxx.APP file. Thus, all modifications to a .BIN file must be integrated into an xxxx.APP file by the XTEL-MEM tool.

For example, after modifying a module configuration, and requesting a save using the **[STORE]** key, the XTEL-MEM software is run, after confirming the update of the xxxx.APP file.

This update is no longer necessary in V5.

To use PL7-COM V4 software in **local mode** with the TSX/PMX file, a PL7-3 application containing at least the I/O configuration must previously have been created. With a V5 station, this task is carried out implicitly by the XTEL-CONF tool.

Comment :

In V5, configuration of the I/O is performed using the XTEL-CONF tool.

2.1 List of manuals quoted in this document

When setting up PL7-COM software, the following manuals may be needed for reference :

- X-TEL reference manual, TXT DM XTEL V5E
- MINI X-TEL reference manual, TXT DM BJR V5E
- PL7-3 operating modes manual, TXT DM PL7 3 V5E
- TSX SCM 20/21/22 modules user's manual, Half/Full duplex character mode, TSX D23004E
- TSX SCM 22 module user's manual, Modbus protocol, TSX D24002E
- UNI-TELWAY reference manual, TSX 24004E
- UNI-TELWAY TSX SCM 21●6 modules user's manual, TSX D24005E
- UNI-TELWAY interface for model 40 TSX/PMX processors manual, TSX DM UTW E.