
TXBT Manual

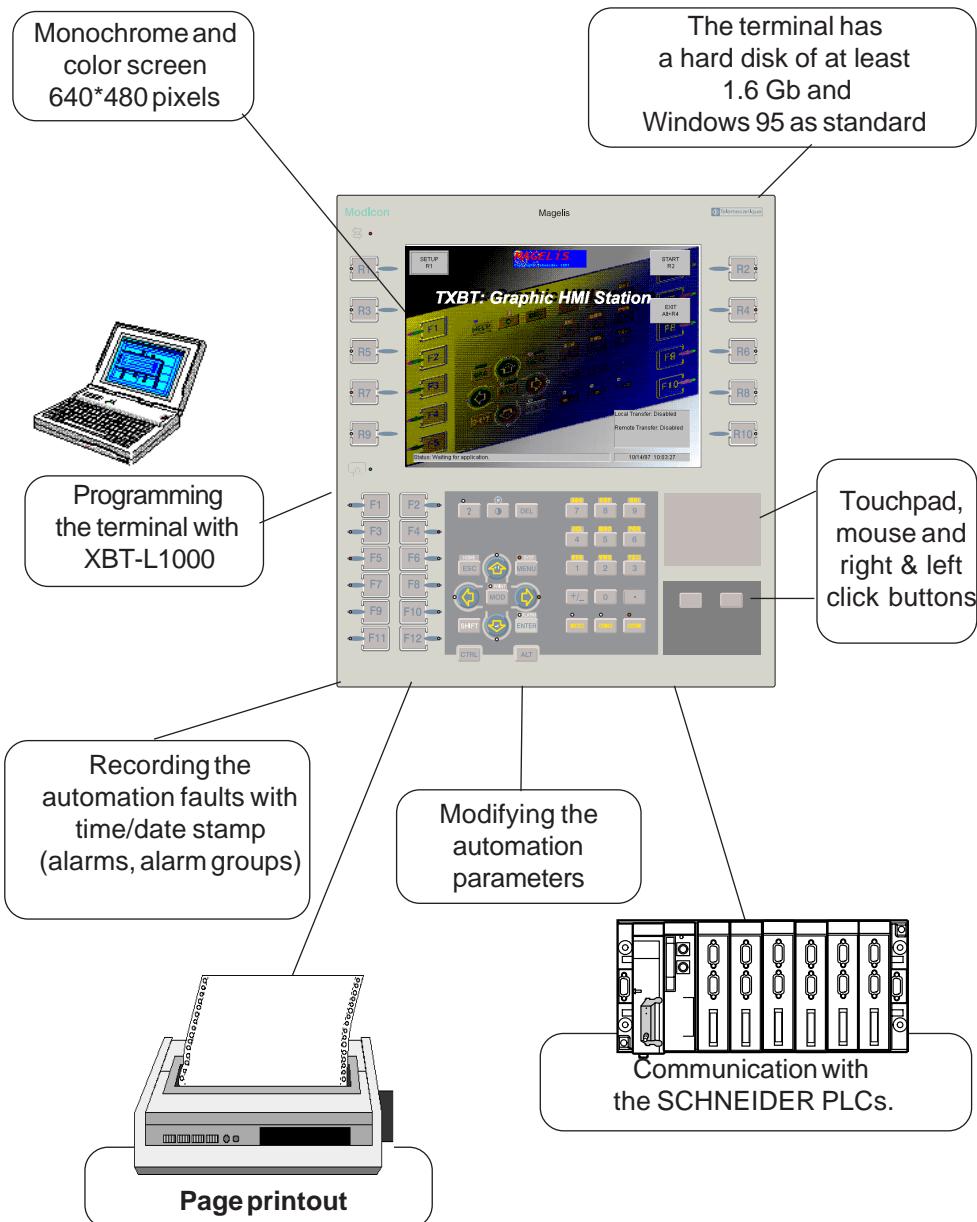
Contents

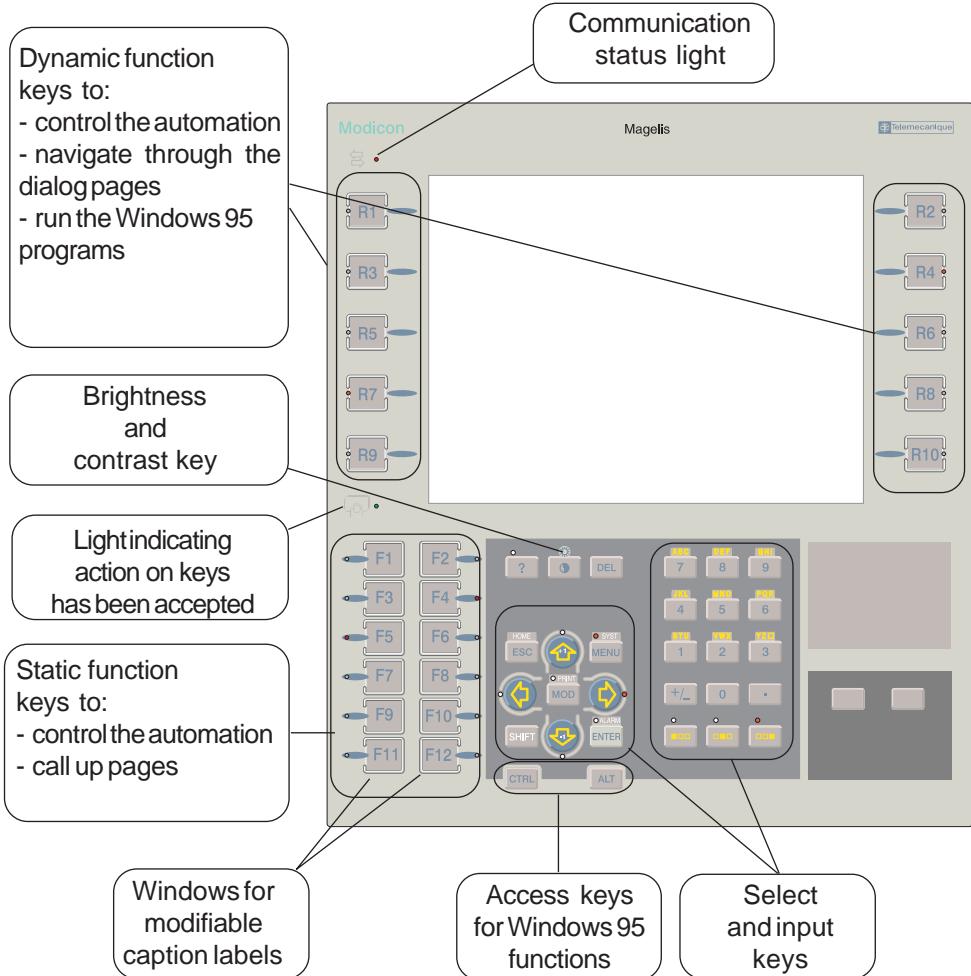
1. Hardware Description of the TXBT Terminal	7
2. Installation	12
2.1. Installing the Precut Labels	12
2.2. Overall Dimensions	13
2.3. Assembly	14
2.4. Installing the Memory Card	16
2.5. Connections	17
2.5.1. Power Supply	17
2.5.2. Relay Outputs	17
2.5.3. LPT1 Printer Port	17
2.5.4. Port for External Keyboard	18
2.5.5. COM1 Serial Port	19
2.5.6. COM2 Serial Port for Programmable Logic Controllers (PLC)	22
2.5.7. Installing the ISA Cards	24
2.5.8. Installing a PCX	25
2.6. Brightness and Contrast Control	27
3. Software Description	28
3.1. Software Architecture	28
3.2. Initial screen	30
3.3. Setting Up the Transfer	31
3.4. Configuring the Printer	32
4. Operation	33
4.1. Powering Up	33
4.2. Loading an Application	36
4.3. Changing Protocol and Defining the Addresses	38
4.3.1. Addressing with the ModbusProtocol	38
4.3.2. Addressing with the Modbus + Protocol	39
4.3.3. Addressing with the UNI-TELWAY, Fipway, ISAway Protocols	39
4.3.4. Configuring the Address on the TXBT	40
4.4. Input of a Remote Address in XBT-L1000	43
4.5. Loading Additional Files During the Application Transfer	46
4.6. Running Applications	48
4.7. Adjust Function	51

5. Troubleshooting	55
5.1. The Local Transfer from XBT- L1000 does not function	55
5.2. Impossible to communicate with the PLCs	55
5.3. Diagnostic Messages	57
5.3.1. No Startup Application	57
5.3.2. Impossible To Open the Keyboard Driver	57
5.3.3. Impossible To Use the Specified Protocol	57
5.3.4. Terminal Is Not a TXBT	58
5.3.5. Operator-Dialog Software Is Already Running	58
5.3.6. OLE Function Will Not Be Available During Running	58
5.3.7. Wait Message For Connection To PLC	58
5.4. Remote Transfer Does Not Function	59
5.5. Damaged Disk: Windows Does Not Start.....	59
5.6. A Monochrome Screen Is Black	60
5.7. Other Cases.....	60
6. Connecting Equipment to the TXBT	62
7. Installing Software on the TXBT.....	64
7.1. Installation Methods	64
7.2. Installing by the Add-on Transfer Integrated in XBT-L1000	66
7.3. Installing by Direct Cable Connection	68
7.4. Installing by an External Drive	71
8. Utilization Tutorial for the OLE Server of the Magelis Applications	72
8.1. Introduction	72
8.1.1. Needs Provided	72
8.1.2. Applications supported	73
8.2. Development of a Client Application	74
8.2.1. Example of a Visual Basic 5.0 Application	74
8.2.2. Adding Extra Functions	85
8.2.3. Example of an Excel 97 Application	89
8.3. Manipulated MAGELIS Objects	97
8.3.1. Types of Variables Processed by MAGELIS	97
8.3.2. Format of the Automation Variables Processed by MAGELIS	98
8.3.3. Using Symbols	98
8.3.4. Address	99
8.3.5. Call Syntax	99

8.4. Description of the Methods Available	101
8.4.1. GetEquipmentList: Gets the Equipment and Symbol Files Lists	102
8.4.2. GetLangueSystem : Gets the Language of the System Pages in Progress	103
8.4.3. GetRepere : Gets the Address	104
8.4.4. ReadVariable: Continuous Unitary Read	105
8.4.5. AddCallbackReference / ReadVariableContinue / ActiveReadContinue	106
8.4.6. AddCallbackReference	108
8.4.7. ReadVariableContinue	108
8.4.8. ActiveReadContinue	109
8.4.9. StopReadContinue	110
8.4.10. WriteVariable: Write	111
8.5. Debugging an OLE Automation Client Application	112
9. Connecting the TXBT to an Intranet	113
9.1. Selecting the Network Card, and Installing	113
9.2. Setting the Network Parameters	113
Appendix: Interpreting the Error Codes	114

1. Hardware Description of the TXBT Terminal





Access to the Windows 95 Function Keys

A keyboard for a PC with Windows 95 as standard has two extra keys: the START and MENU keys. To access these functions on the TXBT, use the following equivalents:



- For the **Start** key, press Ctrl+Esc on the TXBT to run an application.



- For the **Menu** key, press the right mouse button on the TXBT to access the most common icon-activated functions.

- For **TAB** key, press F2.

- For **BACK TAB** key, press SHIFT + F12.

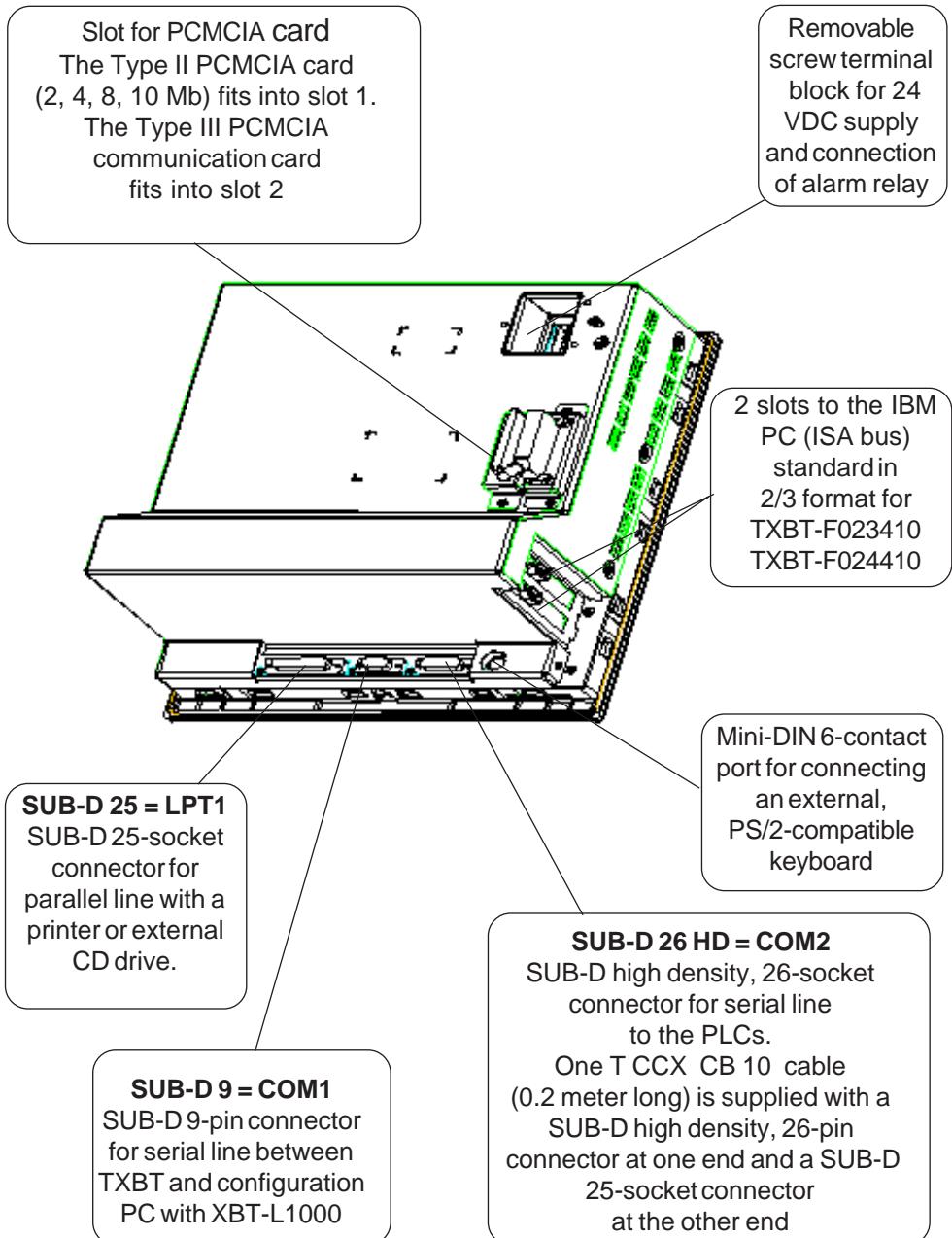
Note:

Pressing the TAB key on an external keyboard enables the F12 key.



Pressing the keys Ctrl, Alt and F1 in sequence on an external keyboard is equivalent to pressing the Static Function key i on the TXBT keyboard.

CAUTION: After use, this key sequence leaves the image of Static Function Key i at 1 in the PLC, even after the keys have been released. The only way to make the image of Static Function Key i disappear from the PLC is to press and release Static Function Key i on the TXBT keyboard.



Environment

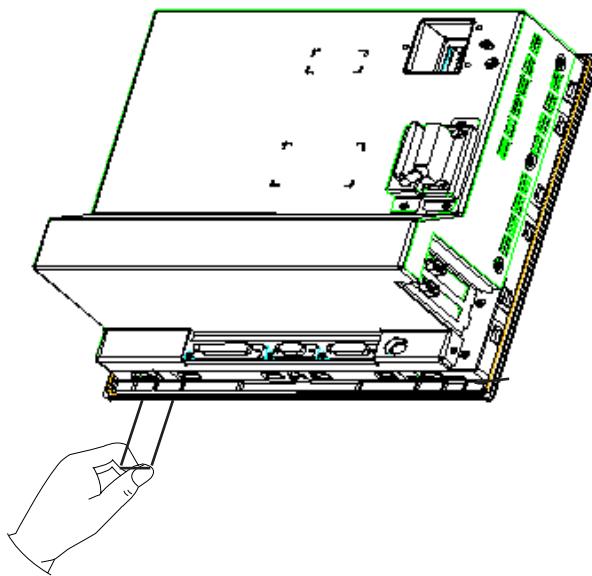
Operating temperature:	0 °C... +45 °C
Storage temperature:	- 20 °C...+60 °C
Relative humidity (no condensation):	0 ... 85 %
Protection (front panel):	IP 65, Nema 4 (IEC 529 - NF C 20010 - UL Type 4,4x indoor)
Protection (rear panel):	IP 20
Electrostatic discharge resistance:	IEC 1000 - 4 - 2 level 3
Electromagnetic interference:	IEC 1000 - 4 - 3 10 V / m
Electrical interference:	IEC 1000- 4 - 4 level 3
Shocks:	IEC 68-2-27: half-sinewave pulse of 11ms, 15 gn, along the 3 axes
Vibrations	IEC 68 - 2 - 6 0.075 mm 10 Hz - 57 Hz 1 g 57 Hz - 150 Hz

2. Installation

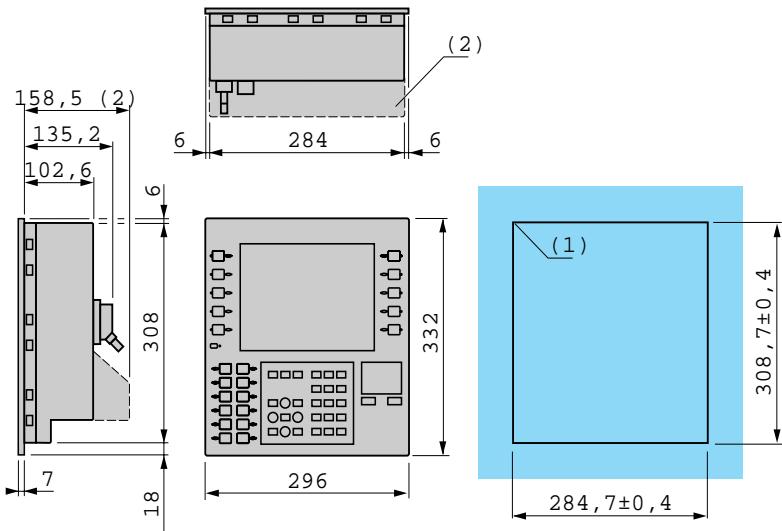
2.1. Installing the Precut Labels

Set of precut labels (Reference XBLYF10).

These sets of labels allow a text to be associated with the function keys Fx.



2.2. Overall Dimensions



**Cutting out the mounting panel
Panel thickness $1.5\dots6$ mm**

(1) $r = 2$ min., 3.5 max.
(2) Only TXBT-F02

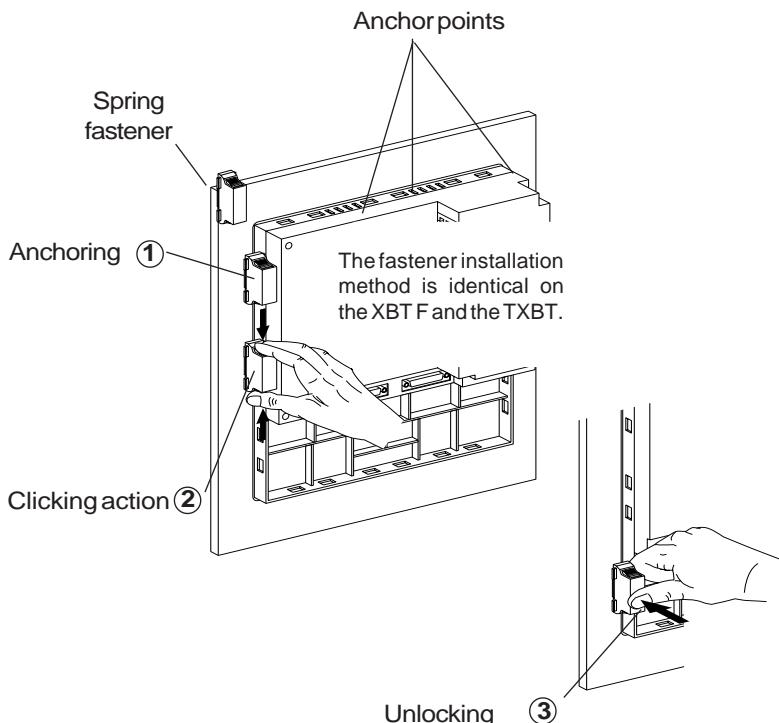
WEIGHT = 5.1 kg

2.3. Assembly

The TXBT terminals are attached to the mounting panel by spring fasteners (12 for the TXBT-F02 and 10 for the TXBT-F03)

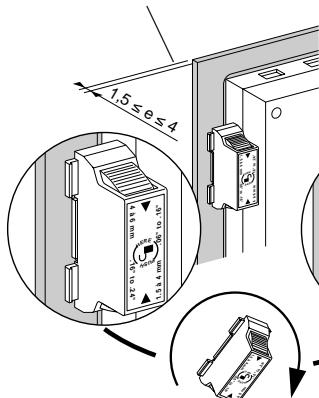
 Pay attention to the direction of assembly, which depends on the mounting panel thickness.

For perfect sealing, the panel surface must be flat and rigid.

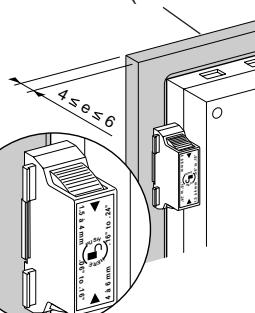


- **Spring fastener position versus mounting panel thickness.**

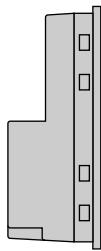
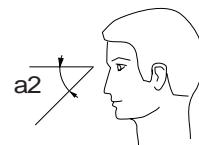
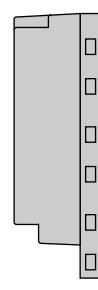
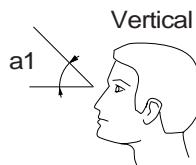
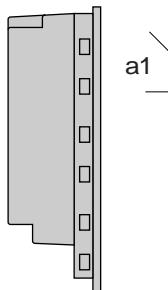
$1.5 < \text{panel thickness} < 4 \text{ mm}$
 $(0.06 < \text{thick} < 0.16 \text{ inch})$



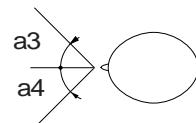
$4 < \text{panel thickness} < 6 \text{ mm}$
 $(0.16 < \text{thick} < 0.24 \text{ inch})$



- **Optimum viewing angles (in degrees)**



Horizontal



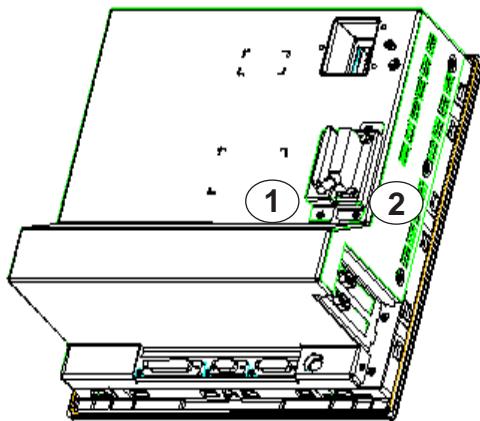
Screen type	a1	a2	a3	a4
Monochrome	20	10	30	30
Color	30	20	45	45

- **Cleaning**

Clean with water to which a neutral (non-acid) product can be added.

2.4. Installing the Memory Card

The Type II PCMCIA memory card (2, 4, 8, 10 Mb) fits into slot **1**.
The Type III PCMCIA communication card fits into slot **2**.



• Handling Recommendations for PCMCIA Cards

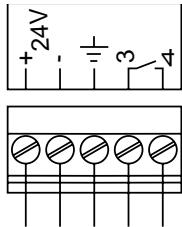
- Protect the cards from humidity,
- Never use force to insert a card as this could damage the locating device.
- The memory card can be inserted and removed with the power on. To extract the card, click the PCMCIA icon on the Windows 95 task bar, and select "Stop Sundisk ATA FlashDisk".
- You can also use the configuration panel to stop the PCMCIA card.
- The network interface card must be inserted and extracted with the power off.



When a Magelis application is running, you must stop it before inserting or removing a PCMCIA card.

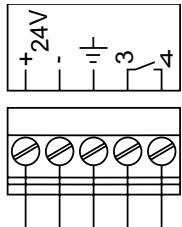
2.5. Connections

2.5.1. Power Supply



- **Nominal voltage:** 24 V DC
- **Voltage limits:** 18... 30 V (including ripple)
- **Consumption:** 80 W
- **Inrush current on power-up** (for capacity of the external protection):
 - I **peak:** 108 A (for zero line impedance)
 - $I^2 t = 1.2 \text{ A}^2/\text{s}$ (for capacity of line fuse)
 - I. t = 11 mC (for capacity of line circuit breaker)

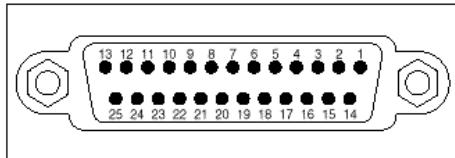
2.5.2. Relay Outputs



- **Operating limits:** 1.0 mA / 5V min. AC/DC
- **Max. resistive load:** 1A / 24V - 220 V AC/ DC
- **Max. inductive load:** 0.5 A / 24V - 220 V AC/ DC

2.5.3. LPT1 Printer Port

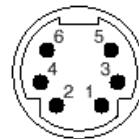
- **Type:** Centronics
- **Connector:** SubD 25 sockets



Pin No.	Signal	I/O	Description
1	-STROBE	O	Strobe
2	D0	I/O	Data bit 0
3	D1	I/O	Data bit 1
4	D2	I/O	Data bit 2
5	D3	I/O	Data bit 3
6	D4	I/O	Data bit 4
7	D5	I/O	Data bit 5
8	D6	I/O	Data bit 6
9	D7	I/O	Data bit 7
10	-ACK	I	Acknowledge
11	BUSY	I	Busy
12	PE	I	End of paper
13	SLCT	I	Select
14	-AUTOFED	O	Automatic line feed
15	-ERROR	I	Error
16	-INIT	O	Printer initialisation
17	-SLCTIN	O	Select input
18/19/20	0V	-	Signal ground
21/22/23	0V	-	Signal ground
24/25	0V	-	Signal ground

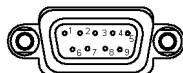
2.5.4. Port for External Keyboard

- **Connector:** Mini DIN 6 connector



Pin No.	Signal	I/O	Description
1	KDATA	I/O	Data
2	NC	-	Not connected
3	0V	-	Ground
4	5V	-	5 VDC
5	KCLOCK	O	Clock
6	NC	-	Not connected

2.5.5. COM1 Serial Port

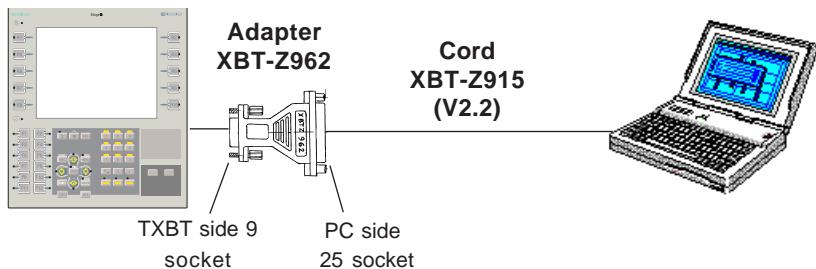


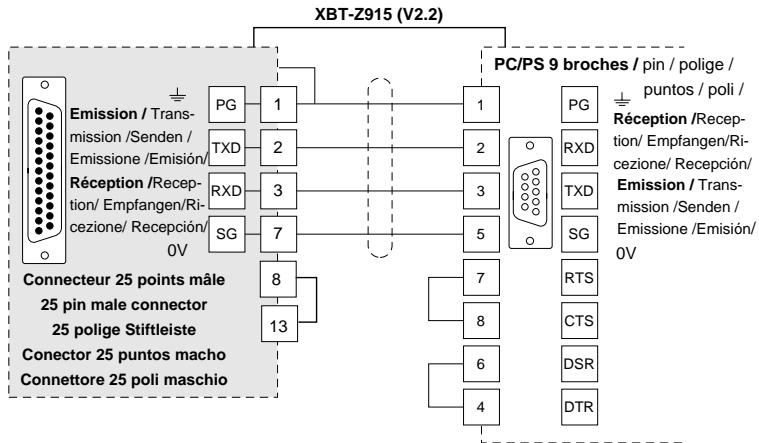
- **Type:** RS232
- **Connector:** SubD 9 pins

Pin No.	Signal	I/O	Description
1	DCD	I	Data carrier detect
2	RXD	I	Data received
3	TXD	O	Data transmitted
4	DTR	O	Data terminal ready
5	0V	-	Signal ground
6	DSR	I	Data set ready
7	RTS	O	Request to send
8	CTS	I	Clear to send
9	RI	I	Ring indicator

- **Configuration for local transfer of applications**

The TXBT/PC link is made with cord XBT-Z915 (V2.2.) fitted with the adapter XBT-Z962.





Connect the serial port with the power off.
Only connect one port at a time.
Tighten the 2 connector lock screws.

- **Configuration for remote transfer of applications**
Refer to § 4.2.

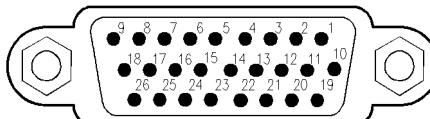
- **Connecting a Mouse to COM1**

A serial mouse can be connected to COM1 by the following procedure:

- stop the Magelis application.
- on the initial screen, use Setup to disable the local transfer which monopolises the COM1 port.
- from Start/Settings/Control Panel/Add New Hardware, reply "No" when asked if you want automatic detection of the device, select "Mouse" as type of device, then "Standard Mouse Types" and "Standard Serial Mouse".
(You may make another choice when the mouse manufacturer and model are listed on the screen).
- it is next advisable to delete the mouse corresponding to the touch pad, using the Delete button of the System Properties box.

2.5.6. COM2 Serial Port for Programmable Logic Controllers (PLC)

- **Type:** RS232/422/485 isolated
- **Connector:** SubD high density, 26 sockets



Connect the serial port with the power off.
Only connect one type of port at a time.

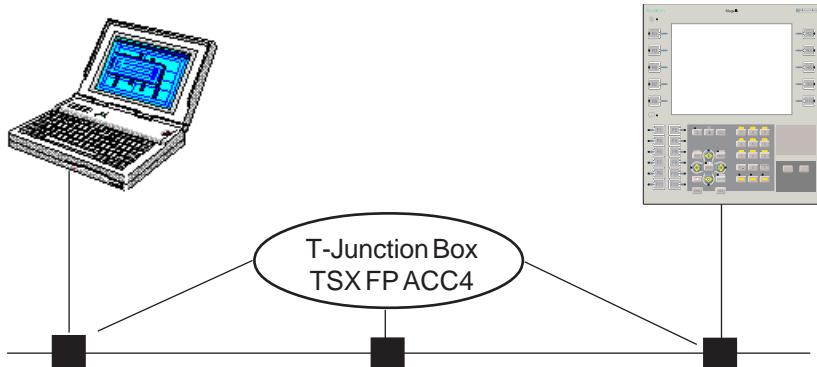
Pin No.	Signal	I/O	Description
1	0V iso	-	RS422 ISO common
2	A'		RS422 RXD + receive
3	COM	-	Multipoint addressing common
4	B3		Multipoint address code
5	-UTW		
6	RXD		RS232 ISO receive
7			Reserved
8			Reserved
9			Reserved
10	B	O	RS422 TXD - send
11	A	O	RS422 TXD + send
12	B'		RS422 RXD - receive
13	B1		Multipoint address code
14	B4		Multipoint address code
15	NC	-	Not connected
16	0V Iso		RS232 ISO common
17			Reserved
18			Reserved
19	0V iso	-	RS422 ISO common
20	NC	-	Not connected
21	B0		Multipoint address code
22	B2		Multipoint address code
23	PAR		Multipoint address parity
24	TXD	O	RS232 ISO send
25			Reserved
26			Reserved

- **Method of Connection**

UNI-TELWAY: The TXBT is connected to the network via cable XBT-Z968 and adaptor T CCX CB20 002, coupled to TXBT COM2 (high density 15 sockets).

Fipway: The TSXFPP20 PCMCIA card must be inserted in slot 2; the TXBT is connected to the unit TSXFPACC4 via cable TSXFPCG010.

The Plug and Play function automatically identifies the PCMCIA cards. There is nothing to configure, except for the TXBT address on the network (use the Magelis Tools->Fipway tool)



Other Connections

Refer to the service instructions supplied with each protocol to determine the references of the cables and the cable connections.

Typical Connections

Refer to § 4.4.

2.5.7. Installing the ISA Cards

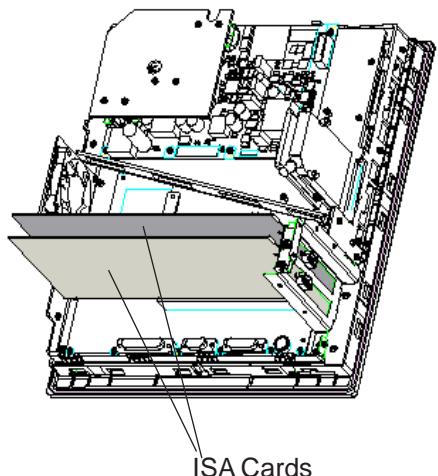
The TXBT-F023410 and TXBT-F024410 feature two slots for add-on cards to the IBM PC-AT (ISA bus) 2/3 format standard.

- **Procedure**

Before starting to remove the rear cover, switch off the terminal power and disconnect all the power and connecting cables.

As the ISA standard does not define the length of the external connections, certain ISA cards may have to be inserted in the TXBT after it has been secured on its mounting panel.

1. Remove the rear cover,
2. Locate the two ISA bus connectors in the bottom of the terminal,
3. Remove the face plate from each insertion slot,
4. Make any switch adjustments required by the card manufacturer's instructions; then fit the card by plugging it into its connector,
5. Lock the card in position,
6. Install the rear cover.



- **Maximum Consumption**

The overall consumption of the two cards on the ISA bus must not exceed:

Maximum consumption at 5 V: 2 A max.,

Maximum consumption at 12 V: 1.2 A max.,

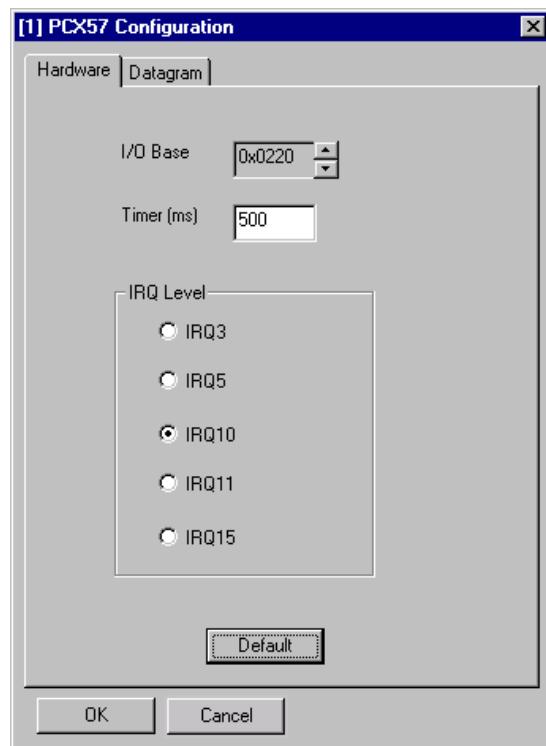
- **Installing the Software**

Refer to the instructions supplied with the card.

2.5.8. Installing a PCX

Install the PCX card as shown in § 2.4.7 and by following the instructions supplied with the card .

The ISAWAY driver is preinstalled on the TXBT and it must not be reinstalled; it is however necessary to use Start/Programs/Magelis/Tools/ISAWAY to configure the memory address and the IRQ according to the PCX card switches.



The next step is to allocate the resources (selected I/O address and IRQ) so that Windows does not assign them to another card:

- Start/Settings/Control Panel/System
- Device Manager; Select: Computer then click Properties.
- Select: Reserve Resources.
- Allocate the IRQ and I/O address assigned to the PCX.

2.6. Brightness and Contrast Control



The controls only work with monochrome screens.

Contrast control: While keeping the control key depressed, use the UP and DOWN arrow keys to adjust the contrast.



Brightness control: while keeping the SHIFT key and the control key depressed together, use the UP and DOWN arrow keys to adjust the brightness.



This adjustment is not possible with an external keyboard.

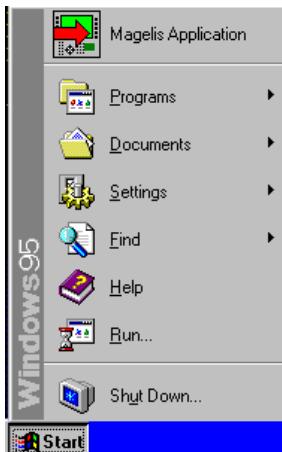
3. Software Description

3.1. Software Architecture

The TXBT includes:

- **A disk (C)** containing Windows 95 and the Magelis graphic startup application (for using Magelis graphic applications created with the software XBT-L1000).
- **A disk (D)** for use as required. The D disk space also contains two special directories with:
 - a copy of Windows 95,
 - a copy of the startup application installation for use if there are problems with disk C (see § Troubleshooting).
- **A disk (E)** for the type II PCMCIA memory card (inserted in slot 1). This is the PCMCIA card that will receive the downloaded application, i.e. the basic executable files and the application data. The card can also store the **add-on tasks** (see § 4.6) and the **external executable files** (see § 4.5).

When this cartridge is not installed, the application is downloaded into a dedicated directory on disk C: (TXBT\appli).



The Start menu features a Magelis Application icon for running the Magelis graphic application.

The Programs menu accesses the Magelis Tools Group containing the following utilities:

- The configuration utilities for the networks UNI-TELWAY, Fipway, Isaway, Modbus+
- The test utility for the console front panel (Testleds)
- A test utility for the xway networks (Xwin32)

As a guide, the available space on a 1.6 Gb disk will be:

- 390 Mbytes on C.
- 830 Mbytes on D.



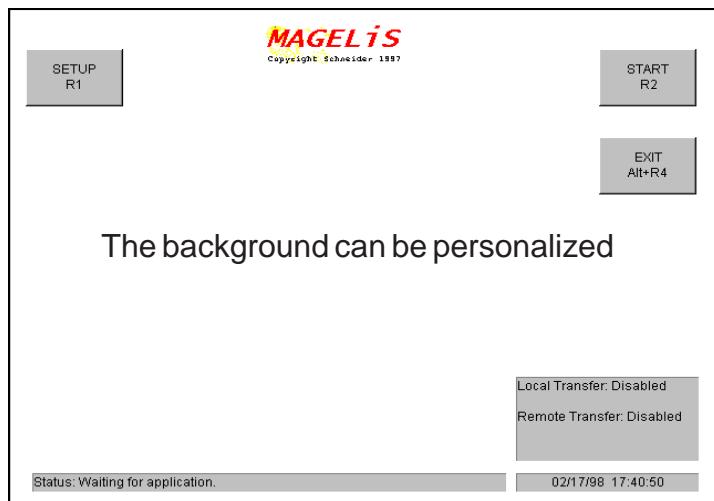
Bootstrap PCMCIA

A bootstrap PCMCIA could be needed if problems occur on the TXBT (see § 5); it is therefore advisable to do this the first time it is used.

Use a PCMCIA card of at least 4Mb.

1. Insert the PCMCIA card,
2. Open Windows Explorer,
3. Use the right mouse button to select drive E and then «Format ...»
4. In the box displayed, select "**Copy only the system files**",
5. Perform formatting,
6. Copy the contents of the directory TXBT\Bootcard on the XBT-L1000 CD-ROM to the PCMCIA card.

3.2. Initial screen



This screen appears when the terminal has no Magelis graphic application or when the application has been stopped.

- **Background of the Initial Screen**

The screen background can be changed by simply replacing the file TXBT.BMP in the directory C:\TXBT\TOOLS. The file must have a resolution of 640*480, 256 colors).

The initial screen functions are activated by **the dynamic keys**.

- **Running the Magelis Graphic Application R2**

Switchover to operation of the Magelis graphic application.

- **Access to Windows 95, Running an Application**

Pressing the CTRL+ESC keys runs an application.

- **Closing the Initial Screen**

As the initial screen cannot be iconized, the ALT+R4 keys are used to close the screen.

- **Setup:** The initial screen indicates the type of transfer, whose parameters can be set: see § 3.3

3.3. Setting Up the Transfer

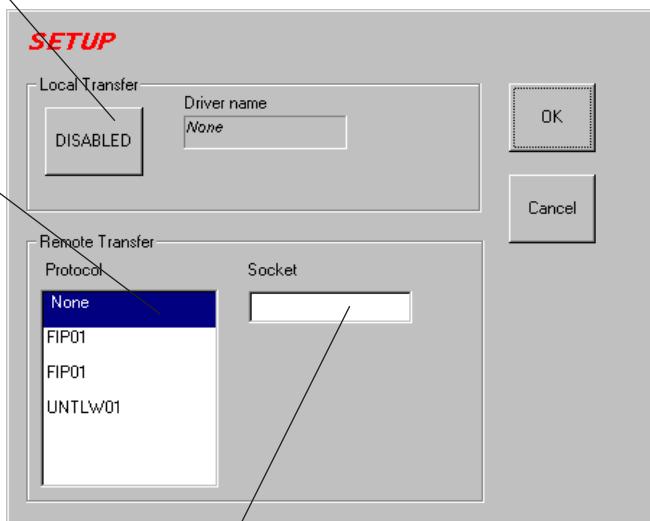
This function is used to set the transfer parameters:

- **Local transfer:** the Magelis graphic application is loaded into the TXBT via the direct PC link (PC with XBT-L1000 <-> TXBT software); the COM1 port is used.
- **Remote transfer:** if the terminal and the PC containing XBT-L1000 are connected to the PLC network and if this network is **UNI-TELWAY**, **Fipway** or **Ethway**, the transfer can be made over this network.

The R1 Setup function accesses the parameter-setting screen.

Pressing this button enables or disables the local transfer. The Driver Name field indicates UNTLW02 until the transfer is enabled and displays None when the transfer is disabled.

To enable the remote transfer, you must select the protocol for the network.



Socket (channel):

- For protocols other than UNI-TELWAY, leave the value of -1.
- For UNI-TELWAY, enter the number of the desired slave.

The address shown on the initial screen is the address to be utilized for a remote transfer by XBT-L1000.

Note

Generally there is no need to disable the local transfer.

This is compulsory when:

- connecting a printer on the COM1 serial port,
- connecting an external mouse on the COM1 serial port.
- connecting any other device on the COM1 serial port.

3.4. Configuring the Printer

The printer is configured on the TXBT using the Windows 95 printer configuration.

The printer configuration screen can be opened:

- either directly from Windows (Start menu ...),
- or under the Magelis application via the "System menu" (SHIFT-MENU), then "Printer parameters" (R5).

4. Operation

4.1. Powering Up

First Power-Up

Before being fully operational, Windows 95 must be personalized by proceeding as follows the first time your machine is switched on:

- Give your **Name** and the name of your **company**,
- Read through the Windows 95 license contract and click "I accept the contract conditions",
- Enter the Windows 95 **serial number** given on the documentation,
- Click **Done** to confirm this information,
- Select the relevant time zone for the machine's place of use and confirm by **OK**,
- Select the type of printer if installed on your machine, or click **Cancel** if there are none installed.

Windows 95 is now ready for use.

Note: To rerun the machine under Windows 95, click on **Start** then **Shut Down** and select **Restart** the computer on the next panel.



Functions performed by the TXBT on power-up

On power-up, the terminal searches for the Magelis graphic application on the **PCMCIA** card and then on the hard disk in the directory **C:\TXBT\APPLI**.

If the terminal finds an application, the application is automatically run. If the terminal is unable to find an application, the initial screen is displayed; it can be closed using the keys Alt-F4.

The following functions are available on power-up:-

- Switching to Windows 95 (Alt-F4/Alt-R4).
- Automatic running of the Magelis application when a valid cartridge is inserted.
- Automatic running of the application on completion of a transfer.
- Manual running of the Magelis graphic application via the R2 key.
- Setting up the TXBT via the R1 key (see § 3.3).
- Cartridge formatting when a Transfer/Export command is received from XBTL1000.

Return to the Magelis Graphic Initial Screen

To rerun a stopped Magelis application, select the **Start/Magelis Application** command.

- 
- If the startup utility has not been run, no applications can be loaded.
 - To stop the TXBT, use the Windows Start/Stop command.
- If the TXBT is switched off without using this command, the operator will be requested to supply some information on the next power-up.



If you do not want to run a Magelis application automatically on startup:

- Press **SHIFT** continuously during the Windows startup, or
- Modify the **Programs/Start group** by **deleting** the Magelis application from it.

To do this, go to the properties of the Windows taskbar (right-click → Properties).

And in the menu 'Programs of the Start menu', select the Advanced button.

Now delete the Magelis application from the Programs/Start directory.

4.2. Loading an Application

An application is loaded via XBT-L1000 and the Transfer/Export command.

Method of Loading

1. Storing the application on the TXBT

The Magelis graphic application can be stored on the terminal:

- either on the type 1 PCMCIA card in slot 1,
- or on the hard disk in directory C:\TXBT\APPLI.

2. During the Transfer

- If the PCMCIA card is installed in the terminal, the software transfers the application,
- If the PCMCIA card is not installed, the software transfers the application into the directory C:\TXBT\APPLI.

If the transfer is interrupted, the application is invalid and must be transferred again.

Note:

- when a PCMCIA card is installed, only the PCMCIA application is disabled and the hard disk application becomes the active one.

3. Current Application

Only one Magelis application can be executed on a terminal; this application is called the **Current Application**.

4. On Power-Up

- If an application is resident on the PCMCIA card,
- and if another application is on the hard disk in the directory C:\TXBT\APPLI, the application on the PCMCIA card is copied onto the hard disk in the directory C:\TXBT\APPLI and writes over the previous application.

5. Inserting a PCMCIA Card during Operation

- If a PCMCIA cartridge containing an application is inserted while an application is running (application in C:\TXBT\APPLI),
- and if the user quits the current application,
the next time the initial screen is accessed, the application on the PCMCIA card is copied onto the hard disk in the directory C:\TXBT\APPLI.

6. No PCMCIA Card or No Application on the Card

When there is no cartridge or no application on the cartridge, the current application is the one in the directory C:\TXBT\APPLI

7. Point-to-Point Transfers

These transfers transit through the serial port SUB-D 9-pin connector (see COM1 pin-out in § 2.4.5) and are controlled by the UNI-TELWAY driver configured as the master (installed as standard on the terminal).

8. Remote Transfers

If the terminal and the PC containing XBT-L1000 are connected to the PLC network and if this network is **UNI-TELWAY, Fipway or Isaway**, the transfer can be made over this network.

The transfer network and the serial link are selected by setup via the initial screen.

Refer to § 4.3. for the terminal addressing on the network.

9. When a Printer is connected to the serial port

When a printer is connected to the serial port, the local transfer must be disabled (see § 3.3).

4.3. Changing Protocol and Defining the Addresses

TXBT terminals can be used with TSX Micro, Premium and Quantum PLCs. The terminals do not support the UniTE V1 protocol but only UniTE V2

Communication Protocols

The following protocols are available:

FIPO, FIPWAY, ISAWAY, UNI-TELWAY, Modbus and Modbus+. These protocols can be downloaded from the XBT-L1000 CD-ROM using the XBT-L1000 software (menu File/Install protocol). The protocols are then transferred on the TXBT while the application is being transferred.

Network Connection

The network connection is via the SubD 26-socket connector (COM2 see § 2.4.5.) or via a PCMCIA communication card (Modbus+, FIPIO, FIPWAY, refer to the Instructions supplied with the PCMCIA card)

4.3.1. Addressing with the Modbus Protocol

The TXBT terminal is necessarily the Master. The message "Connecting..." is shown on the service line until a PLC answers a request.

4.3.2. Addressing with the Modbus + Protocol

The network connection is via a TSX MBP 100 PCMCIA card



- The address is coded on the TBXT: refer to § 4.3.4
- If entered in XBT-L1000, the network address is ignored.

4.3.3. Addressing with the UNI-TELWAY, Fipway, ISAway Protocols

- **Slave UNI-TELWAY:** A pictogram (see § 5.3.7.) is displayed until the terminal is seen by the master (On a TXBT, a cable has no effect).
- **Fipway, ISAway (Communications with PCX via the ISA bus):** The message Connecting... is shown in the service bar until a PLC answers a request.

4.3.4. Configuring the Address on the TXBT

The TXBT's address on the network is entered locally on the TXBT using the network configuration utilities provided (**Magelis Tools: UNI-TELWAY, Fipway, Isaway, ModBus+**).

The Magelis Tools Group containing these utilities is accessed via the Program menu. The Xwin32 utility is used to test the XWAY network.

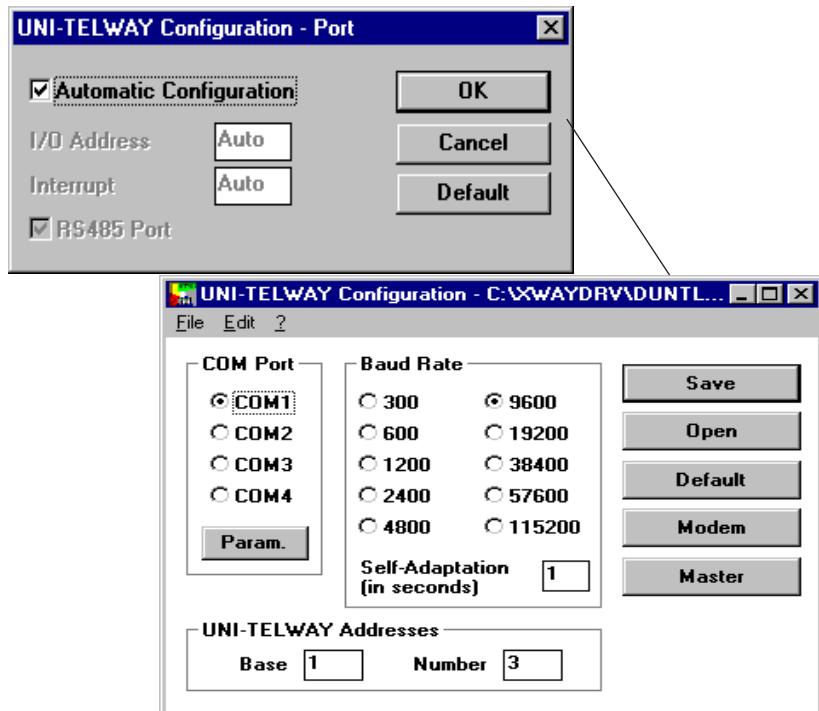


- Any configuration produced under XBT-L1000 is ignored.
- The FPP20 thumbwheels are ignored.
- A code in a UNI-TELWAY unit is ignored (SCA62).



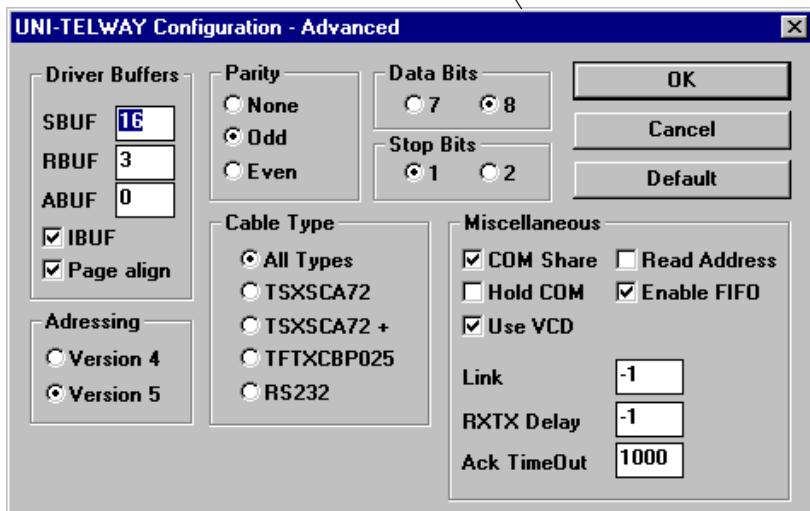
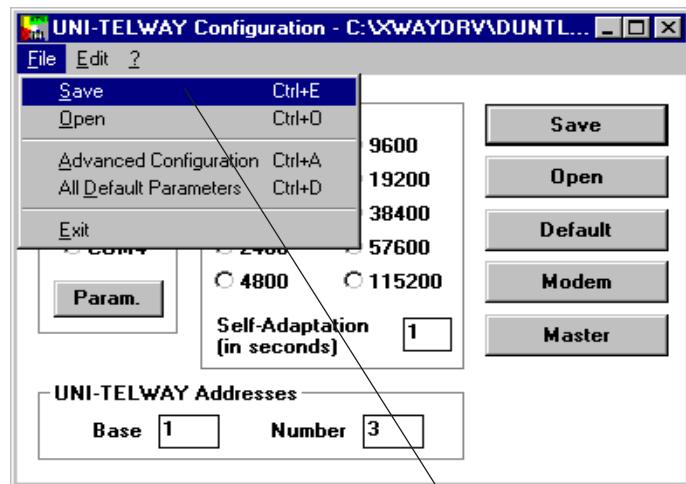
Configuring the UNI-TELWAY Address

The TXBT occupies the default slave addresses 4, 5 and 6. To modify them, select the menu Start/Program/Magelis Tools then the UTW Configuration tool.



Magelis Graphic: TXBT

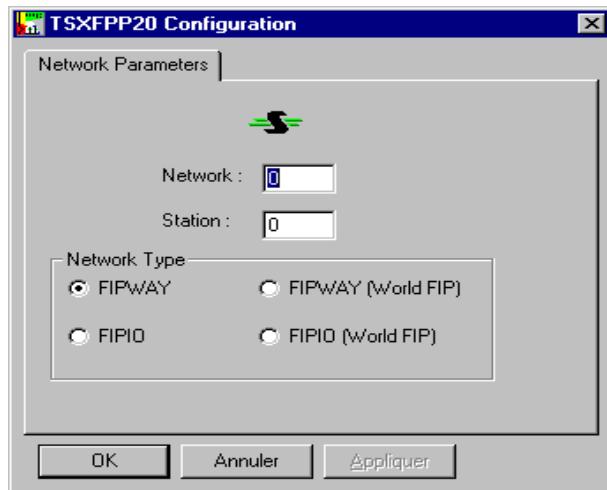
Enter the parameters as indicated by the protocol's service instruction (supplied with the XBT-L1000 software).





Configuring the FIPWAY-FIPIO Address

Select the Start/Program/Magelis Tools menu and then the FIPWAY-FIPIO configuration tool.



Enter the parameters as indicated by the protocol's service instruction.



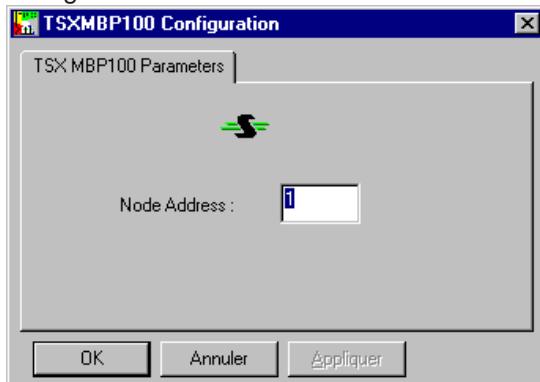
Configuring the ISAWAY Address

The address of the TXBT on ISAWAY cannot be configured; the TXBT is seen as a slave of the PCX, with a number of 200 or higher.



Configuring the MODBUS + Address

Select the Start/Program/Magelis Tools menu and then the MODBUS + configuration tool.

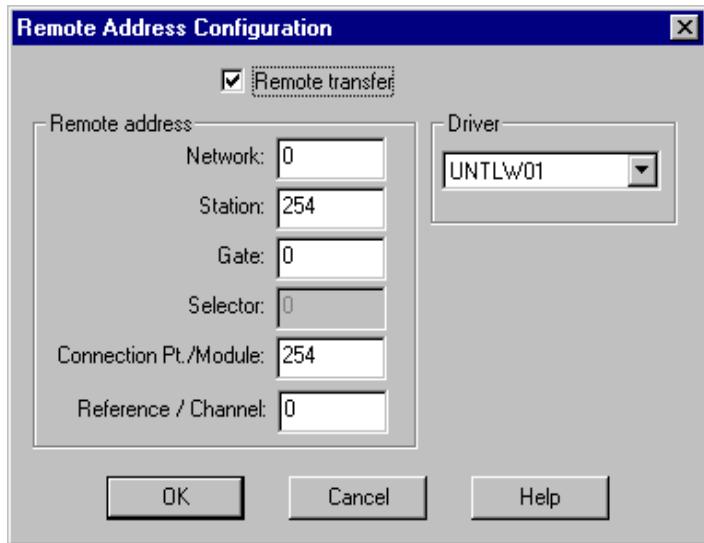


Enter the parameters as indicated by the protocol's service instruction.

4.4. Input of a Remote Address in XBT-L1000

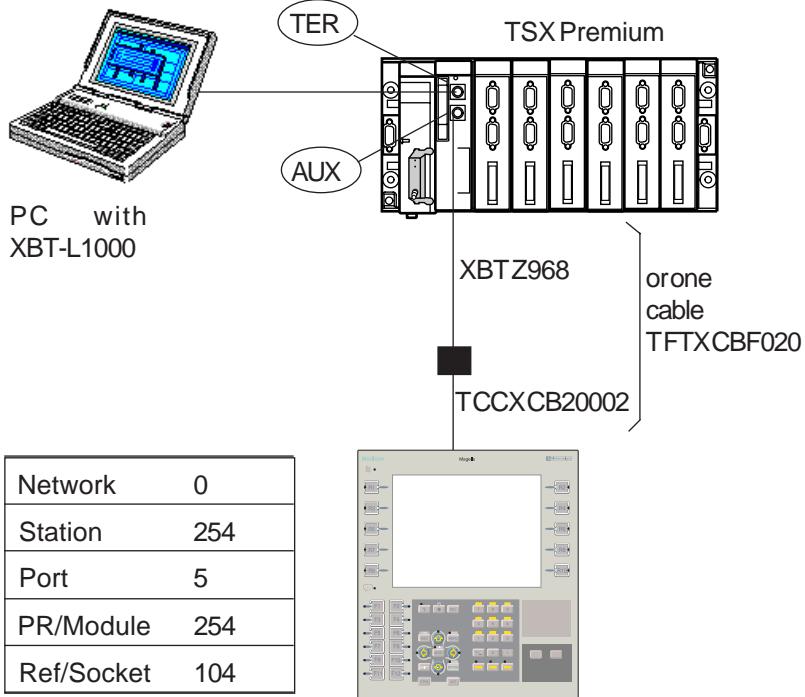
This address is used to export an application to a TXBT via the existing XWAY network, instead of via the transfer cable XBTZ915 + XBTZ962. When the address is configured in the TXBT, the address for the remote transfer must be declared in XBT-L1000. To do this, proceed as follows:

1. Select the **Transfer/Remote address** menu.
2. Define the remote address using:
 - the address configured locally on the TXBT,
 - the socket (channel) number selected or allocated dynamically during the setup of the transfer.

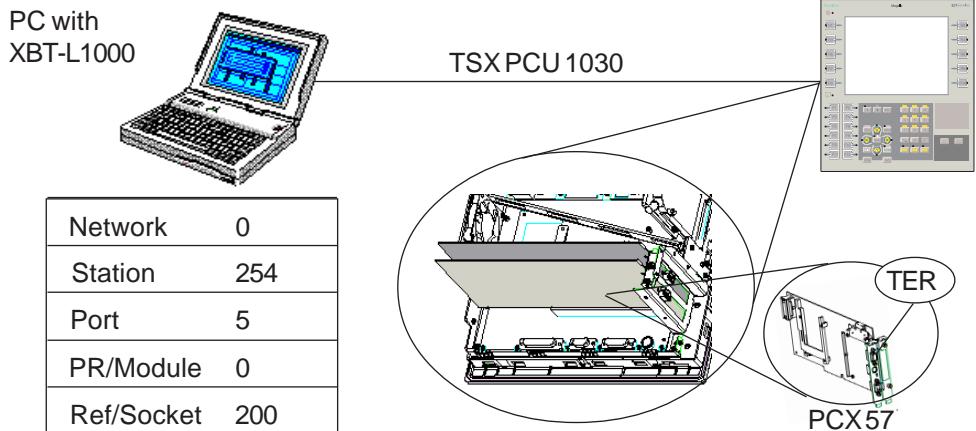


Note: Remote transfer is not possible with FIPIO.

- Example of addressing with UNI-TELWAY

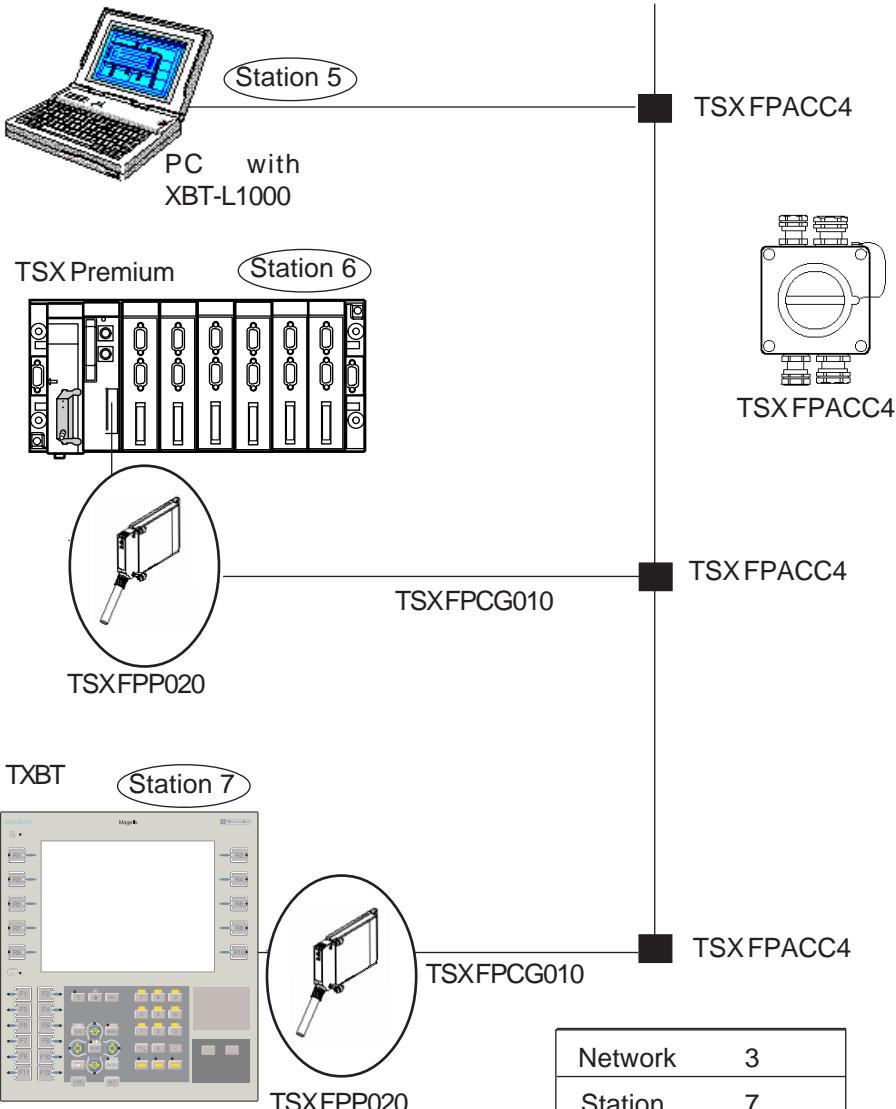


- Example of addressing with ISAWAY



Magelis Graphic: TXBT

- Example of addressing with FIPWAY



*Indicate the port given on the initial screen.

Network	3
Station	7
Port	1*
PR/Module	
Ref/Socket	

Remote Transfer

Tick this box to input the address of the target terminal. When the box is not ticked, the transfers are local.

Remote Address

The 5 fields are used to input the XWAY address at 5 levels, i.e. Network, Station, Port, Module, Socket (Channel). The proposed default address is the address of the bus master PLC.

The configuration of this address is specific to the XBT-L1000 software . It is memorized so that it will be used for the transfers made in subsequent sessions. It is not associated with the application.

Driver

If several drivers are installed in the PC, the driver to be used must be selected. This choice applies only for remote transfers.

N.B.

1. In remote transfers, the transfer performance depends on the various networks involved.
2. In local transfers, only the DUNTLW (UNI-TELWAY) driver can be used.

4.5. Loading Additional Files During the Application Transfer

During the PC -> TXBT transfer, additional files can be loaded into the TXBT.

Procedure

1. With the XBT-L1000 software, select the **Configure/Attach items...** menu then select **Files**.
2. Select the files to be loaded into the TXBT.
Use the upper part of the screen and the **Browse** button to find and select the files to be added.
The lower list shows the files already added as attached items.

Add is used to add the file selected in the upper list to the list of attached files.

The **memory size** required by the application in the terminal is updated to allow for the size of the files in the list.

Delete: deletes a file from the list.

Delete all: deletes all the files in the list.

Export

The files are exported to the application directory C:\TXBT\APPLI.

Import

During importing, only the application's source files are imported.

Modifying the Files

If files already selected are modified on the PC with XBT-L1000, they must be deleted (**Delete**) from the list and then added (**Add**) before transfer.



Also refer to Running Applications in § 4.7.

N.B.

When a file previously transferred as an attached item is deleted from the selection, it will no longer be transferred and will moreover be deleted on the TXBT during the export.

4.6. Running Applications

The TXBT terminal runs the Windows 95 applications installed on it (e.g. the calculator). If the Magelis application is started, it continues to operate in the background, the pages are refreshed and the alarms are monitored. The software can be run by two methods.

Running by Windows 95 commands

- **To Start a Program**

1. Click the Start button (or press the CTRL+ESC keys), then select Program.
2. Select the program to be run.

- **To Start a Program using the Run Command**

1. Click the Start button (or press the CTRL+ESC keys), then Run.
2. In the Open field, enter the location and name of the program you want to run.
If you have forgotten the location or name of the program file, just click Browse.

Running by the Dynamic Function Keys Rx

An executable program can be run by pressing an Rx key (e.g. the calculator).

• Configuration with XBT-L1000

Example: start the calculator with the key R4 once the measurement is displayed.

1. Select the page associated with the dynamic key in the page Tree-structure window.
2. Select the **Configure/Function keys** menu to display the terminal front panel.
3. Double-click the dynamic function key to be allocated.
4. Select **Run Application**.
5. Click **Open** and enter the executable file path.

Executable File Path

This is the path (disk, directory/subdirectory) on the TXBT terminal and not the path on the PC with XBT-L1000.

If the location of the executable file on TXBT is modified, it will not operate when run.

Example: D:\\WINDOWS95\\CALC.EXE



Running from the PLC

The protocols UNI-TELWAY, Fipway, FIPIO and ISAWAY have a function allowing an executable file on the TXBT to be run from the PLC, without any specific configuration in the Magelis application.

For this, the TXBT must receive an "FC" request and then the name of the executable file to be run, e.g. "FC" "calc" will run the Windows calculator.

The executable file name is limited to 8 characters and must be followed by a word at 0x0000.

The executable file must be in the current directory (Appli) or in a Windows system directory (C:\Windows, C:\Windows\system...).

The PLC is not sent any meaningful 'executable file running OK' report.

Typical implementation of the PLC program so that it sends this command to the TXBT:

```
RESET%MW100:X0
%MW229:=30          (* timeout *)
%MW230:=10          (* length of data *)

(* data: *)
%MW220:=16#4143;      (* "CA" *)
%MW221:=16#434C;      (* "LC" *)
%MW222:=16#0000;
%MW223:=16#0000;
%MW224:=16#0000;

(*Send request FC to the TBXT (refer to the PL7 Manual)*)
SEND_REQ(ADR#0.0.200, 16#00FC, %MW220:5, %MW225:1,
%MW227:4);
```

N.B.

Address 0.0.200 corresponds here to a PCX sending the FC request to the TXBT which accommodates it.

More generally the destination address must be the first address of the TXBT on the network used (refer to the service instructions for the protocol).

4.7. Adjust Function

- **Principle**

The adjust function is used to modify on a TXBT the variables of the connected PLCs.

Adjustment is available for the following communication protocols:

- . UNI-TELWAY · Fipway
- . IsaWay · Fipio
- . Modbus · Modbus+

- **Accessible items**

XWAY Protocols (UNI-TELWAY, ISAWAY, FIPWAY, FPIO)

- internal word
- constant word
- internal word bit
- internal double word
- internal bit
- local input/output bit
- remote input/output bit
- system bit
- step bit
- macrostep bit
- macrostep input step bit
- macrostep output step bit

Modbus and Modbus+

The accessible items are the input and output registers, the input and output bits, and the bits extracted from registers.

- **Loading the PL7 Symbols File**

The adjustment can be symbolic if the address/symbol association file has been loaded from XBT1000 to the TXBT.

Creating the Symbols File

To create a symbols file with PL7:

1. Open the variable editor.
2. Select the File/Export command,
3. Select the disk and/or directory for storage.

-
4. Enter a file name, e.g. Appli.SCY
 5. Select the export mode and confirm the procedure.

Transferring the Symbols File to the TXBT

The resulting symbols file (e.g. Appli.SCY) must be loaded on the TXBT terminal as follows:

1. Select the **Configure/Attach items...** menu, then select **Files**.
2. Select the file(s) (example Appli.SCY) for loading on the TXBT: the file is now part of the Magelis application and will be exported in each export operation.
3. Select the **Transfer/Export** menu. The application and the selected file(s) are then exported.



A modification to this file will not be automatically recorded; for this, the file must be deleted and then added again.

The terminal does not check for consistency between the PLC application and the loaded symbols file.

- **Accessing the Adjust Function**

The adjustment function can be accessed with the dynamic function keys using two methods:

From the System Page

Display this page by pressing the **Menu** key and then the dynamic function key associated with the Adjust item.

From an Application Page

The adjust function can be run from an application page by allocating a dynamic function key associated with this page to the running of the variable adjust tool.

Configure this function with XBT-L1000 as follows:

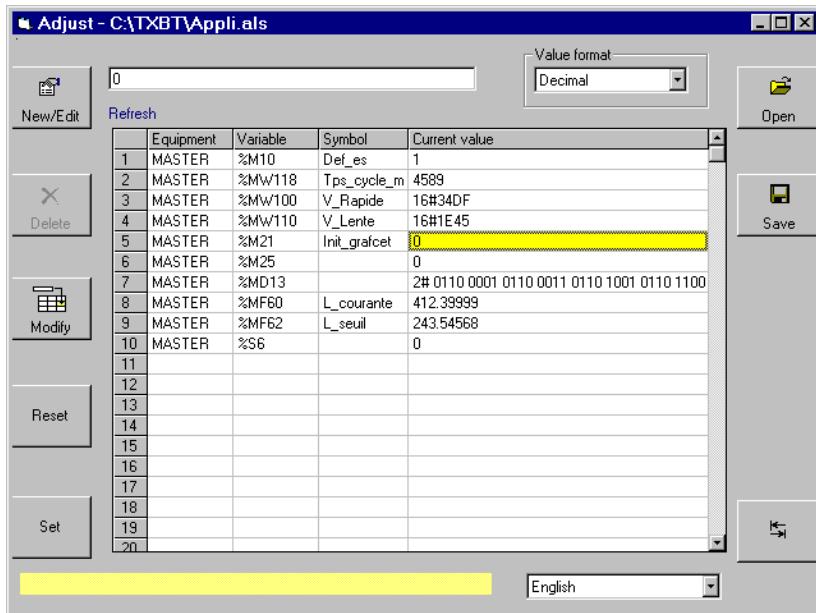
1. Select the page associated with the dynamic key in the Page tree-structure window.
2. Select the **Configure/Function keys** menu to display the terminal front panel.
3. Double-click the dynamic function key to be allocated.
4. Select **Run Application**,
5. Click **Open** and enter **REGLAGE.EXE**.

N.B.

The file's complete path must be given when the executable file is not in the APPLI directory.

• Procedure

1. Select the equipment,
2. Enter a variable (New/Edit R1),
3. Start nonstop refreshing (Refresh R6).



Modifying a Value

1. Select the cell to be modified,
2. Enter the new value,
3. Confirm the entered value (Modify R3).

Deleting a Variable

1. Select the line,
2. Delete the line (Delete R10).

Display Format

The data read in the PLC can be displayed in the following formats:

- Signed decimal integers
- Hexadecimal values (0 to 0xFFFFFFFF)
- Bit string
- Single precision floating format
- ASCII character string with 1 to 4 characters; non-ASCII characters are represented by " . ".

List of Variables

The list of displayed variables can be saved via the key R4 in a file .ALS then retrieved by the key R2. The retrieved list overwrites the current list.

N.B.

1. When "???" appears, it was not possible to access the variable.
2. Pressing **ESC** returns to the previous page of the Magelis application. The adjust function is not stopped but its window is relegated to the background.

Alt-R4 (or clicking X) stops the adjust function.

5. Troubleshooting

5.1. The Local Transfer from XBT- L1000 does not function

Check that TXBT shows the initial screen with "Local Transfer : Enabled" displayed at the bottom right.

- If the local transfer is not active, attempt to activate it using the Setup procedure (R1: see § 3.3.)
- If the local transfer is still not active, check the DUNTLW2 driver is initiated in the Config.sys file by the following line:
DEVICE=C:\TXBT\DRIVERS\UNTLW2.EXE
PROFILE=C:\TXBT\DRIVERS\UNTLW.002
- If the local transfer is activated but does not function, a possible cause is the speed performance of the PC with XBT-L1000. In this case, lower the speed on the TXBT by editing the file C:\TXBT\DRIVERS\UNTLW.002 (set the SPEED value to 19200), reboot and try again.

5.2. Impossible to communicate with the PLCs

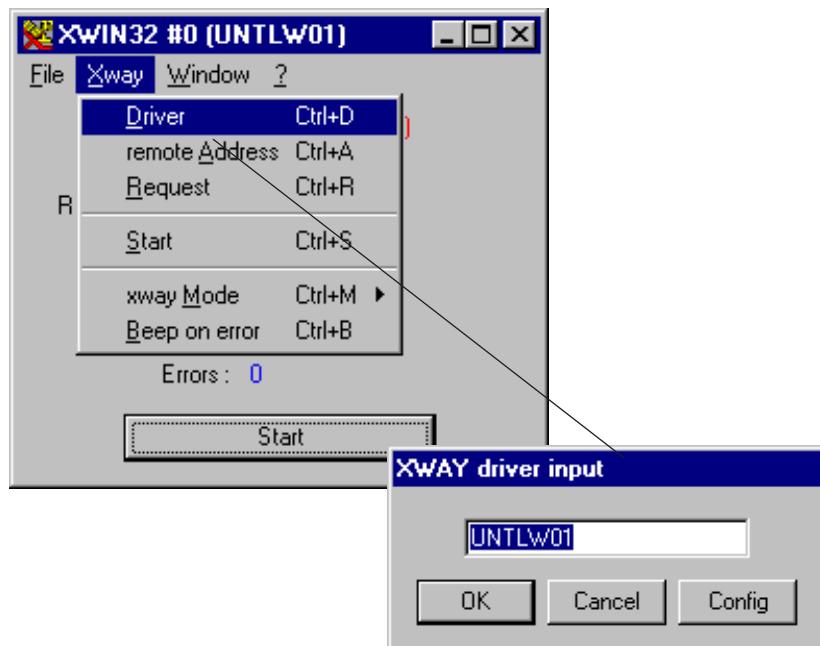
Diagnostic messages are displayed in case of problems; see § 5.5 for details of these messages.

1. Check the PLC addresses input under XBT-L1000
2. Use the XWIN32 tool to check for access to these PLCs via their addresses. Proceed as follows:



Select the Start/Program/Magelis Tools Group menu and then the network test utility XWAY (Xwin32).

Select the driver and address of the PLC in Xwin32.



3. Check the network is physically connected.
4. The address entered under XBT-L1000 of a PCX or a UNI-TELWAY master must be 0.254.0, and not its address on the network if this PLC is hooked into the network.

5.3. Diagnostic Messages

The TXBT can display a series of visual messages for first-level fault finding by the user.

All the messages are animated and consist of the repeated display of images depicting the problem.

5.3.1. No Startup Application



The application has not started up normally; rerun it via the Magelis Application menu.

5.3.2. Impossible To Open the Keyboard Driver

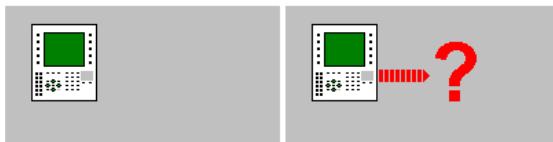


Black

Red

Check the pre-installation by the command (under DOS): What C:\txbt -C

5.3.3. Impossible To Use the Specified Protocol

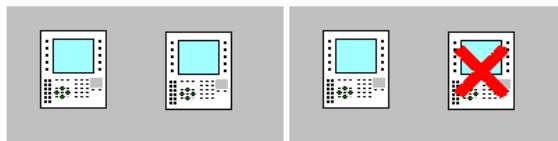


Check the pre-installation by the command (under DOS): What C:\txbt -C

5.3.4. Terminal Is Not a TXBT



5.3.5. Operator-Dialog Software Is Already Running



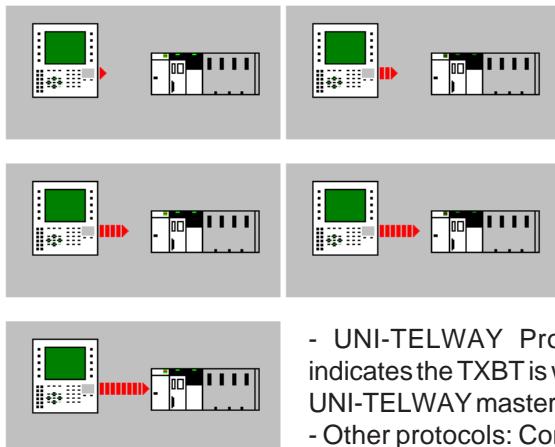
Stop and reboot the TXBT terminal.

5.3.6. OLE Function Will Not Be Available During Running



The OLE services are not available.

5.3.7. Wait Message For Connection To PLC



- UNI-TELWAY Protocol: The message indicates the TXBT is waiting for polling by the UNI-TELWAY master.
- Other protocols: Connection in progress.

5.4. Remote Transfer Does Not Function

Check the address input via XBT-L1000; it must be the one read on the initial screen (Bottom right). If a problems occurs, correct the address.

5.5. Damaged Disk: Windows Does Not Start

The disk D contains a copy of the Windows 95 CD and a copy of the start-up installation. If disk C is defective, all the software programs (Windows 95 and Magelis graphics) can be reinstalled. This requires a **bootstrap PCMCIA**.

1. Power down the TXBT
 2. In slot 1, install the **bootstrap PCMCIA** supplied with the TXBT.
 3. On setup, select to boot via the PCMCIA (press the key R2 during booting).
- Before rebooting the TXBT, the configuration was as follows:

C	D	E
Applications Operating Disk	Saving	PCMCIA

- After rebooting the TXBT, the configuration becomes:

D	E	C
Applications Operating Disk	Saving	PCMCIA

It is therefore necessary to format **D:**

4. Oformat **D:**
5. Reboot the TXBT via the PCMCIA card. Windows will start to reinstall automatically. Follow the instructions.
6. Once Windows 95 is reinstalled, reboot the TXBT via the hard disk (by pressing the key R2 during booting)
7. Install the startup application under Windows : D :\Txbt.ins\disk1\Setup
8. Check the settings of the Windows parameters:
 - 256 colors
 - US keyboard
 - Install the PCMCIA support
 - Delete the HDD stop

5.6. A Monochrome Screen Is Black

The brightness may be set to minimum. Adjust the settings.

5.7. Other Cases

Using TXBT with an Azerty keyboard:

Keyboard can be switched from US (QWERTY) to French (AZERTY) by the CTRL-SHIFT key.

This action is local to the focused application.

Avoid remanent messages indicating that a job is in the printer queue:
Select the default printer. Set it in "Hold" state.

Change FIP connexion from FPP20 to FPC10 ISA card:

It is mandatory to transfer an application with another protocol, and then to transfer again the FIP protocol application.

When it starts, you will be prompted to select your connection card.

Alarms defined on "Other Bit"

These alarms are numbered with a number from the 1st number after Master's dialog table last alarm bit (i.e. 4 Alarms words in master's dialog table means that 1st alarm on "Other Bit" is numbered 65)

Instead of equipment name, "Oth. Bit" appears in the Alarm list status bar.

A Trend object does not occupy the whole object surface.

The useful surface of a trend is limited to 256 pixels

Fatal Error 2 occurs when launching the Magelis Application:

A PCX PLC or the master PLC on UNI-TELWAY must be defined as addresses 0.254.0

Alarm group names are truncated to 8 characters:
Even if in buildTime you can enter more than 8 characters, only the 8 first
will be available in runtime

Shutting Down TXBT Safely

TXBT must be Shutdown safely by Start/Shutdown in Windows95 Taskbar before powering off.

You can directly Shutdown Windows by executing the provided tool: Shutdown.exe

Shutdown.exe can be run via a Dynamic Function Key on a Magelis application page.

Unable to access one of the 8 last %Mi configured in PL7

TXBT accesses internal bits by groups of 8 bits. If some in the group are not configured, TXBT will receive an error for the group.

To solve this problem, modify PL7 configuration, and add 8 internal bits.

Message "File not found" during export of an extension task:

This may be because of a Carriage return missing in end of .lst file

6. Connecting Equipment to the TXBT

Always make the connections with the TXBT powered down.
To connect hardware to the COM1 serial port, you must disable the local transfer (see § 3.3).

- **Printer**

A printer can be connected to the parallel port (LPT1 SubD 25 sockets) or to the serial port (COM1 9 pins).

Use the configuration panel to configure the printer:

1. Press the **CTRL+ESC** keys to run the Start menu.
2. Select **Parameters/Configuration panel** command.
3. Click the printer icon. To add a printer, click the Add printer icon.
4. Select port LPT1

- **Mouse**

A mouse can be connected to the serial port (COM1 9 pins).

Use the configuration panel to configure the mouse:

1. Press the **CTRL+ESC** keys to run the Start menu.
2. Select the **Parameters/Configuration panel** command.
3. Click the **Add peripheral** icon and then **Mouse**; allocate the communication port to the mouse.

- **External Keyboard**

An external keyboard can be connected to the DIN 6 socket port; both the keyboards are active.

The TXBT front panel is a QWERTY keyboard and it is therefore preferable for the external keyboard to be QWERTY as well, otherwise one of the keyboards will not operate correctly.

Pressing **CTRL+SHIFT** converts a keyboard from QWERTY to AZERTY and vice-versa.

Connecting an external keyboard does not require rebooting or powering down.



The external keyboard must not be used for machine commands (risk of disconnection causing loss of control, impossible to enter the TFS, etc.).

- **Diskette Drive**

A diskette drive can be connected to the parallel port (LPT1 SubD 25 sockets).

- **CD ROM Drive**

A CD ROM drive can be connected to the parallel port (LPT1 SubD 25 sockets).

- **Network Card**

PCMCIA card: refer to § 2.3

ISA card: refer to § 2.4.7

N.B.

If the equipment has to be installed with an installation diskette, it may be worthwhile recopying the diskette contents onto a PCMCIA memory card and installing via the PCMCIA card.

7. Installing Software on the TXBT

7.1. Installation Methods

Various methods can be employed for installing software on the TXBT. Select the method best suited to the size and form of the software to be installed.

Method	Type of use	Software Size	Speed	Required Hardware
Add-on transfer, integrated in XBT-L1000	Software of medium size, no specific hardware. Files only transferred to the TXBT application directory. Well integrated in the application transfer. Well suited to transferring the data created on the XBT-L1000 PC and used by the TXBT.	< 10 Mb	250 Kb/min.	XBTZ915+ XBTZ962
Remote installation by "Direct cable connection". Interconnects 2 PCs with Windows 95. Not available on Windows NT.	Performs a complete installation, can update directories and change the registry (data base describing the PC configuration). More difficult to install than the add-on transfer integrated in the XBT-L1000. Well suited to installing a product once and for all. Requires that the XBTL-L1000 PC has Windows 95/98.	< 4 Mb if a serial link is used. 30 Mb if parallel cable is used.	50 Kb/min. on serial link	XBTZ915+ XBTZ962 Parallel cable or Laplink type

Method	Type of use	Software Size	Speed	Required Hardware
Use an external diskette or CD-ROM drive on a parallel port.	All cases when this drive is available. To install the drive, its drivers must have been installed beforehand, generally via a diskette using one of the other methods.	Any size	Speed of the parallel port (250 Kb/sec.)	External drive
Use a company Intranet.	The TXBT is on an intranet (See § 9 : Connecting the TXBT to an intranet)	Any size	Depends on the network	Ethernet card in the TXBT
Use a PCMCIA card.	Small-size software, when a portable computer is available for creating the PCMCIA.	PCMCIA cards > 100 Mb are available, their price being the limiting factor		Portable, computer, PCMCIA card

7.2. Installing by the Add-on Transfer Integrated in XBT-L1000

To Define an Add-on Task :

The purpose of an add-on task is to enhance the TXBT with functions provided by an external executable file.

Proceed as follows to create an add-on task:

Create a text file (with extension **.opt**) containing a designation and a file name in the following syntax: (example of sample.opt)

[General]

name=Option Sample 1.0

list=Sample1.lst

Create a file with the name indicated in the "list" input, containing the paths of all the files to be exported to the TXBT (full paths must be given):

example of sample1.lst

d:\Opt\sample\abc.exe

d:\Opt\sample\def.dll

d:\Windows\ghi.ini

Place these 2 files in the XBT-L1000 installation directory; the option is then displayed in the Configure/Attached items/Add-on tasks menu.

The purpose of the add-on task are installed independently on the PC where XBT-L1000 is run.

Task Selection and Transfer

Procedure

1. Select the **Configure/Attach items...** menu then select **Add-on tasks** to be exported.

The list gives the designation of all the add-on tasks installed on the PC, which runs XBT-L1000.

2. Select or deselect the add-on tasks to be exported. The list status is recorded with the application.

N.B.

1. If an application with add-on tasks is opened on a PC and these add-on tasks have not been installed on the PC, they are ignored.
2. An add-on task on the TXBT cannot be imported to the PC.
3. The memory required by the application on the terminal is updated to allow for the size of the files referenced by each of the selected add-on tasks.

Running the Add-on Tasks

1. Certain tasks can be accessed on the system page (e.g. adjust function)
2. An add-on task may also be run by a dynamic function key from any page (see § 4.6. Running an executable file).

N.B.

The transferred files are in the TXBT application directory. If in the next transfert, the add-on task is not selected, the files will be deleted from the TXBT application directory.

7.3. Installing by Direct Cable Connection

Configuration



- **Connections**

The TXBT is connected to a PC with Windows 95 via the Grafiti loading cable (cord XBT-Z915 V2.2.) fitted with adapter XBT-Z962 , or a 9-contact serial cable (female/female) as per the following pinout:

Signal	Pin	Pin	Signal	
DCD	1	<->	4	DTR
RXD	2	<->	3	TXD
TXD	3	<->	2	RXD
DTR	4	<->	6	DSR
DTR	4	<->	1	DCD
GND	5	<->	5	GND
DSR	6	<->	4	DTR
RTS	7	<->	8	CTS
CTS	8	<->	7	RTS
RI	9		not connected.	

- **Procedure**

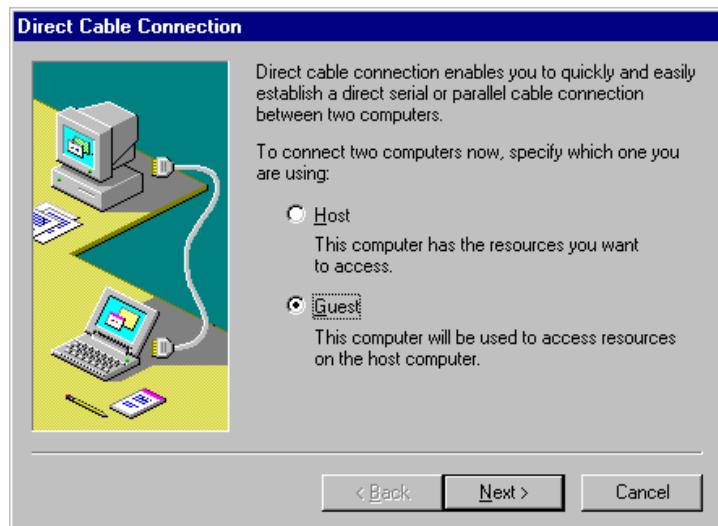
Both units contains the Windows accessory "Direct cable connection"

1. On the PC, use the **Programs\Accessories\Direct Cable Connection** menu to open Direct cable connection and configure as "**Host**" on Com1.
2. Run Direct cable connection on the TXBT, and configure as **Guest**, on Com1.
3. Define the diskette as shared space on the PC; to do this, right-click Diskette A on Explorer and select Properties.
- select the Share tab and tick Shared as
4. Display the host PC on the TXBT.
5. The diskette can now be used as if it were a local diskette on the TXBT.

Remark on the configuration (see diagram on preceding page)

Check the network configuration of the host PC:

- The **Direct cable connection** function must be installed on the host PC (see Windows Help or the documentation).



-
- The network must be installed on the host and the guest as follows:
 - Client for the Microsoft networks,
 - NETBEUI protocol,
 - Access control: control at the resource level.
 - The direct communication by cable can also be made by a parallel cable running between the LPT1 ports of the PC and TXBT. This parallel link has a higher throughput.

The Direct Cable Connection utility is not available on Windows NT.

7.4. Installing by an External Drive

For example, by copying the manufacturer's installation diskettes for the external drive onto a PCMCIA memory card.

Note: An external drive must have been installed in accordance with instructions of its manufacturer.

8. Utilization Tutorial for the OLE Server of the Magelis Applications

8.1. Introduction

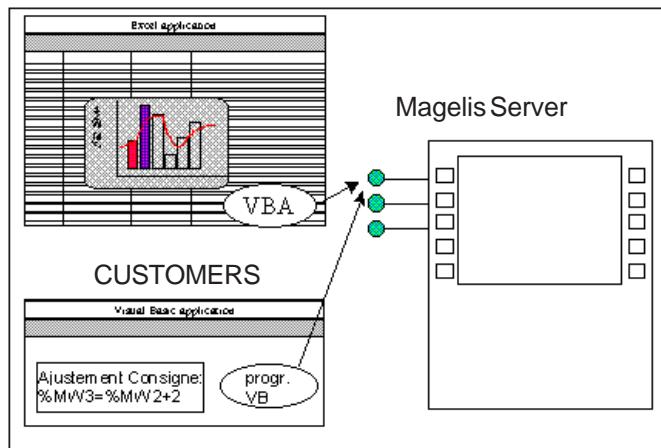
8.1.1. Needs Provided

The principle of this server is to make it possible for applications based on OLE Automation technology to access variables controlled by the MAGELIS products.

The purpose of the OLE Automation server is to open up the MAGELIS data to other Windows applications via, for example, the Microsoft script language VBA (Visual Basic for Applications).

Thanks to this possibility, it is also possible to handle the MAGELIS data in customer applications developed in Visual Basic by the final customer or in any other language supporting the OLE Automation protocol.

Operating Diagram



The user runs the customer application:

- either conventionally using Explorer (the Windows 95 File Manager)
- or by the MAGELIS graphic task which has shortcuts to third-party applications.

N.B.: All the examples described below are provided on the XBT-L1000 installation CD ROM.

8.1.2. Applications supported

The table below lists some of the Microsoft products that support OLE Automation.

Application	OLE Automation (client)	OLE Automation (server)
Microsoft Access version 2.0	Yes	Yes
Microsoft Word version 6.0	No	Yes
Microsoft Excel version 5.0	Yes	Yes
Microsoft FoxPro version 2.5	No	No
Microsoft Mail version 3.2	No	No
Microsoft PowerPoint version 3.0	No	No
Microsoft Project version 4.0	Yes	Yes
Microsoft Publisher version 2.0	No	No
Microsoft Visual Basic V 3.0, V4.0, 5.0	Yes	Yes



A TXBT with 32 Mb of RAM is recommended when using the Client OLE Automation executable files.

8.2. Development of a Client Application

We will now show you how to use the OLE Magelis server in the context of an application running on Visual Basic 5.0 and on Excel 97 (VBA). All the examples described below can be found on the XBT-L1000 CD ROM.

8.2.1. Example of a Visual Basic 5.0 Application

To start, it is necessary to have a basic knowledge of programming Visual Basic in order to create an application. And for this, it is useful to consult a Visual Basic programming guide.

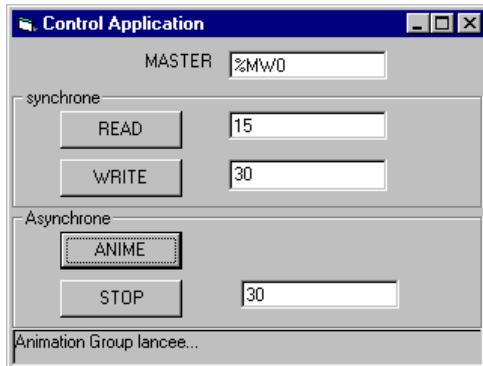
Nonetheless the OLE functions are easy to implement and we will attempt to give you as complete a method as possible for developing an application.

8.2.1.1. Description of the Control Application

Control is an application that only shows all the functions possible with the MAGELIS OLE server. The aim is not to show what a Visual Basic application can do, but to indicate the interactions that must be implemented in order to use the MAGELIS OLE server.

This application will therefore have the capability of:

- reading a system or PLC variable
- writing a system or PLC variable
- continuously accessing a system or PLC variable (asynchronous mode)
- and adding other useful data by the server (This is developed in section 8.3).



Read Function of a Variable

1. Input the variable address in a field.
2. Press the READ button to read the MASTER PLC variable via the server and MAGELIS.
3. The value is returned and displayed in the relevant field.
4. If a problem occurs, a fault is displayed in the status bar.

Write Function of a Variable

1. Input the variable address in a field.
2. Enter the value to be written.
3. Press the WRITE button to write the MASTER PLC variable via the server and MAGELIS.
4. If a problem occurs, a fault is displayed in the status bar.

Continuous Read Function of a variable

1. Input the variable address in a field.
- Press the ANIMATED button to start continuous reading of the MASTER PLC variable via the server and MAGELIS.

-
2. The value is returned and displayed continuously in the relevant field
 3. If a problem occurs, a fault is displayed in the status bar.
 4. Press the STOP button to halt continuous reading of the MASTER PLC variable via the server and MAGELIS.
 5. The variable can be read or written while it is animated, which means the operation of the animation can be checked (e.g. by writing different values and seeing them being refreshed immediately in the relevant field).

8.2.1.2. Creating a New Project

We will first of all create a new project in Visual Basic 5.0. This project is of the Standard EXE type.

When you select a standard EXE type project, it consists of a form (a window containing the application), which will contain the code required for our MAGELIS OLE customer.

To finalize this project creation, go to the "Properties" box of form 1 and to the property "Caption" and replace the "Form 1" text by "Control Application".

It is now time to construct our application; we will therefore use figure 1.1 as a basis and construct the Man/Machine Interface (MMI) using the "control panel" objects.

The useful controls are briefly described below:

Type of control	Name	Caption
Frame	Frame2	Synchronous
Frame	Frame1	Asynchronous
CommandButton	CommandButtonREAD	READ
CommandButton	CommandButtonWRITE	WRITE
CommandButton	CommandButtonREADCONTINUE	ANIME
CommandButton	CommandButtonSTOPREADCONTINUE	STOP
TextBox	ClientVariableName	
TextBox	ClientVariableResult	
TextBox	ClientVariableValue	
TextBox	ClientVariableAnim	
Label	Label1	MASTER
Label	Status	

8.2.1.3. Application Architecture

- Declaration of the global variables
- An initialization procedure for connecting to the MAGELIS form_load server
- A procedure via function buttons
 - CommandButtonREAD_Click()
 - CommandButtonWRITE_Click()
 - CommandButtonREADCONTINUE_Click()
 - CommandButtonSTOPREADCONTINUE_Click()
- A procedure to quit the application
 - form_queryUnload
- A display function for the asynchronous mode

8.2.1.4. Initialization

We must now connect to the server; to do this, we will declare a variable of the object type that we are going to initialize.

This variable must be declared to the application as global (i.e. declare it before the procedures and functions).

Explicit Option

" declare the server object

Dim ptrMagelis As Object

Private Sub Form_Load()

" Initialize the objects

Set ptrMagelis = Nothing

" Get Magelis object server

Set ptrMagelis = CreateObject("grafrsrv.SrvDll")

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

Set ptrMagelis = Nothing

End Sub

8.2.1.5. Using the Synchronous Functions

Initially we will only use the synchronous functions (READ and WRITE). As we have seen in the last chapter, we now have an object of our server (ptrMagelis). This object illustrates methods which can be called as follows: ptrMagelis.ReadVariable(Equip,Address,Value) where the parameter is the equipment name (Refer to XBT-L1000 for the equipment name), the address or this address symbol.

CAUTION: For the symbol-based call to work, the symbol files must be associated in the equipment under XBT-L1000 and these files must have been transferred (see XBT-L1000 document: 'Transfer of Additional Files').

The first step is therefore to implement the READ button; to do this, take the method ReadVariable, which uses three parameters that must be declared, plus one for the return code (Refer to section 8.5 for further information).

```
Sub CommandButtonREAD_Click() " Code associated with clicking "READ"  
button
```

```
Dim equip$ As String  
Dim repere$ As String  
Dim Value As Variant
```

```
equip$ = «MASTER» "Equipment name  
repere$ = ClientVariableName.Text "get the address or the symbol  
Result = ptrMagelis.ReadVariable(equip$,repere$, Value)  
If Result <> 0 Then  
    Status.Caption = «Read error...»  
Else  
    Status.Caption = «Read Ok»  
End If  
ClientVariableResult.Text = Value "direct display of the value
```

```
End Sub
```

We can now move on to the method WriteVariable, which also uses three parameters that must be declared, plus one for the return code (Refer to section 8.5 for further information).

Sub CommandButtonWRITE_Click() " Code associated with clicking the "WRITE"button

```
Dim equip$ As String
Dim repere$ As String
Dim Value As Variant

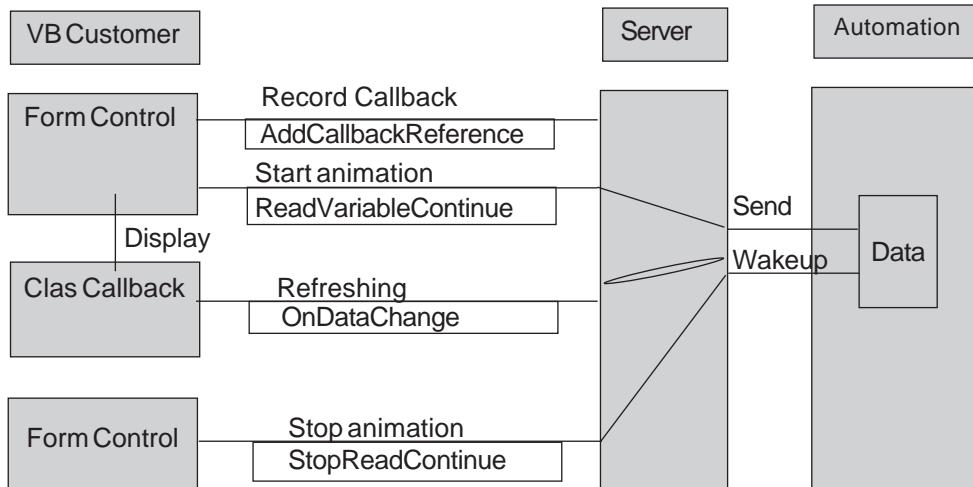
equip$ = «MASTER» "Equipment name
repere$ = ClientVariableName.Text "get the address or the symbol
Value = ClientVariableValue.Text
Result = ptrMagelis.WriteVariable(equip$,repere$, Value)
If Result <> 0 Then
    Status.Caption = «Write error ...»
Else
    Status.Caption = «Write Ok»
End If
```

End Sub

The two functions are now implemented and you can test them.

8.2.1.6. Using the Asynchronous Functions

The asynchronous principle for MAGELIS OLE is to be able to request a variable and to request its automatic refreshing.
To perform this request, we apply the following principle.



N.B. The Callback class record function need only be performed once; afterwards, you can start and stop your animation of the variables as many times as you wish. The only way of removing this recording is to invalidate the class (which we will explain later).

You can see that a class type component must be added to our project. This component is in fact the link the server uses to send the reading values.

This class is an object that will have to be initialized; it only contains one procedure called "OnDataChange". This procedure, whose code we will give you below, presents you with two choices: either to perform the processing in the callback, or to do it by a function call in the Form. The last solution is advantageous in that the variables do not have to be shared between the two objects (the group number).

The latter solution will therefore be used in the example.

We will now add this class; for this, go to the Insert/Class menu , rename the class in **CallBack** using properties/Name

The following code must be inserted (CAUTION: the function declaration must be exactly as below or the server will not be able to call you back).

```
Sub OnDataChange(hVar As Long, Value As Variant)
    " called when a new value is available
    " note that the link between the request and event is made by hVar
    Form1.AfficheResult hVar, Value "call the display function in the Form
End Sub
```

The Callback class is now created and we must declare an object of this class that we must initialize before sending the server its reference.

We will now modify the variables declaration and the procedure Form_Load:

Explicit Option

```
" declare the objects  
Dim ptrMagelis As Object  
Dim ptrCallback As Object
```

Private Sub Form_Load()

```
" Declare the local variables  
Dim Result As Long  
  
" Initialize the objects  
Set ptrMagelis = Nothing  
Set ptrCallback = Nothing  
  
" Getthe Magelis object server  
Set ptrMagelis = CreateObject("grafrsrv.SrvDll")  
  
" Create the callback object internal to the customer  
Set ptrCallback = New ProjTest.Callback  
  
" Create the Callback link between the customer and the server  
Result = ptrMagelis.AddCallbackReference(ptrCallback)  
If Result <> 0 Then  
    Unload Me  
End  
End If  
  
End Sub
```

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

```
Set ptrMagelis = Nothing  
Set ptrCallback = Nothing  
  
End Sub
```

As the return link with the server is now established, we can set up the start and stop mechanism for the animation of a variable.

First of all we will explain the principle involved. The method ReadVariableContinue contains three parameters, which for the first two are tables. These two tables are the breakdown of the variables you wish to animate. An example is given below:

	Equip(*)	Address()
0	MASTER	%MW1
1	SLAVE1	%MD50
2	MASTER	%MW55
3	XBT	SYS_TIME
4	MASTER	%MW10
5	MASTER	%Q5.0
6	MASTER	%MW10
7	XBT	SYS_DATE
8	MASTER	TEMP_FOUR

* The names of the equipment items are the names that you will give in XBT-L1000, except for XBT which corresponds to the systems variables managed by MAGELIS.

SYS_TIME and SYS_DATE are the names of the time and date systems variables.

TEMP_FOUR is a symbol of a variable which you declare in PL7jr and whose file scy you transfer by XBT-L1000.

Once you have completed your tables, use the method ReadVariableContinue, which will send you back a value representing the number of the animated group, followed by the method ActiveVariableContinue to initiate refreshing.

The third parameter of the method ReadVariableContinue represents the refresh rate you want. This rate is an integer number, representing multiples of a second. Entering this parameter as 0 provides the fastest response (depending on the communication). In fact this value depends on the processing capability of your client application.

If you find you can no longer access your application (Mouse click not accepted, press a keyboard key, move the window), your application is not fast enough to process the messages sent back by the Magelis OLE server. Improve this situation by increasing the parameter value.

After refresh initiation, the server will call the method OnDataChange of your callback class once it has all the variables; it will automatically renew the request at the requested rate. Note that, when the group number is 0, there is a problem with the variables you wish to animate (Syntax error, Equipment does not exist, etc.)

In this method, we make use of a procedure, which we have called Display and which must be created by using the Tools/AddProcedure menu and by declaring it as public sub.

We will now complete the code for the Animate and Stop buttons, and the display procedure and the declaration of the global variables (i.e. the variable which designates the animated group).

Option Explicit

" declare the objects

Dim ptrMagelis As Object

Dim ptrCallback As Object

Dim hGVar As Long

Sub CommandButtonREADCONTINUE_Click() " Code associated with clicking the "READCONTINUE" button

```
Dim Result As Long  
Dim Value As Variant  
Dim TabEquip(1) As String  
Dim TabRep(1) As String  
Dim Refresh As integer
```

```
TabEquip(1) = «MASTER»  
TabRep(1) = ClientVariableName.Text  
Refresh = 0 ' is the refresh rate in seconds  
          " 0 is the fastest for the communication
```

```
hGVar = ptrMagelis.ReadVariableContinue(TabEquip, TabRep, Refresh)  
If hGVar <= 0 Then  
    Status.Caption = «Animation fault...»  
Else  
    Result = ptrMagelis.ActiveReadContinue(hGVar)  
    Status.Caption = «Group animation started...»  
End If  
" there is still nothing displayed, the read request is only recorded and will  
generate events for us, processed by the procedure AfficheResult.
```

End Sub " CommandButtonREADCONTINUE

Sub CommandButtonSTOPREADCONTINUE_Click() " Code associated with clicking the "STOPREADCONTINUE" button

```
Dim Result As Long  
  
If hGVar <> 0 Then  
    Result = ptrMagelis.StopReadContinue(hGVar)  
    hGVar = 0  
    Status.Caption = «Animation stopped...»  
End If
```

End Sub " CommandButtonREADCONTINUE

Public Sub AfficheResult(hVar As Long, Value As Variant)

```
If hVar = hGVar Then  
    Form1.TextResultAnim.Text = Value(0)  
End If
```

End Sub

And now your application should be ready to run. First run the server application (via the start-up), then your Visual Basic customer. You can test its operation.

8.2.2. Adding Extra Functions

These other methods are available on the MAGELIS OLE server. They include conversion of a symbol into an address, the list of the equipment items with their related symbols files and the language of the systems pages (for a multi-language application). We will show you how to use these functions in a project. For this we will start from the Control project that we will name ExtendControl.

8.2.2.1. Getting the Equipment List

One of the advantages of the method **GetEquipmentList** is that it not only lets you form a ComboBox object containing the equipment list (and therefore in an application, the possibility of selecting an equipment item without input mistakes), but also provides the list of symbols files associated with these equipment items. If you have this list, all you need is these symbol files (*.scy) so that when you select an equipment item, you will load the list of associated symbols in a new Combobox .

Thus in your Control application, we will replace the MASTER label by a Combobox and then use the method **GetEquipmentList**.

Our application becomes:

```
Private Sub Form_Load()
Dim Result As Long
Dim NbEquip As Long
Dim EquipList As Variant
Dim SymbList As Variant

" Initialize the objects
Set ptrMagelis = Nothing
Set ptrCallback = Nothing
" Initialize the data handles in autorefresh
hGVar = 0
" Get the Magelis object server
Set ptrMagelis = CreateObject("grafrsrv.SrvDll")

" Create the callback object internal to the client
Set ptrCallback = New ProjTest.Callback

" Create the link between the client and the server
Result = ptrMagelis.AddCallbackReference(ptrCallback)
If Result <> 0 Then
    Unload Me
    End
End If

" Get the list of equipment and the associated symbol files
Result = objCbSvr1.GetEquipmentList(NbEquip, EquipList, SymbList)
If Result = 0 Then
    If (IsArray(EquipList)) Then
        For i = 1 To NbEquip
            ComboEquip.AddItem EquipList(i - 1)
        Next i
        ComboEquip.ListIndex = 0
    End If
End If

End Sub
```

It is then necessary to modify the functions which the equipment utilizes, i.e. ReadVariable, WriteVariable and ReadVariableContinue.

Sub CommandButtonREAD_Click() " Code associated with clicking the "READ" button

```
Dim equip$ As String  
Dim repere$ As String  
Dim Value As Variant
```

```
equip$ = ComboEquip.Text "Equipment name
```

8.2.2.2. Getting a Variable address from its Symbol

You can obtain the address of a variable via its symbol to check, for instance, the symbol validity before processing. We will indicate one possible use here but you should be aware that the read and write methods can be employed directly with a symbol.

For example, the function below checks during the input of a variable name whether it is not an address; if it is, the function calls the method GetRepere to display the address of the variable corresponding to the symbol.

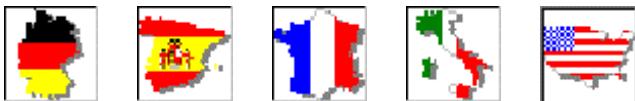
PrivateSub CommandButtonCONVERT_Click()

```
Dim Equip$  
Dim Symbols$  
Dim Value As Variant
```

```
Equip$ = ComboEquip.Text "Equipment name  
If Left$(ClientSymbolName.Text, 1) = «%» Then  
    Symbols$ = ClientSymbolName.Text " Convention CEI 1131-3.  
Else " The variable name is a symbol (an identifier)  
    Result = ptrMagelis.GetRepere(Equip$, Symbols$, Value)  
    If Result<>0 Then
```

```
repere$ = «???? »
Else
    repere$ = Value
End If
End If
EndSub
```

8.2.2.3. Getting the Systems Pages Language



You can also develop a multi-language OLE Automation client and, so that it matches your XBTL1000 applications, you can retrieve the systems pages language data element and adapt your resources accordingly.

To do this, you use the method GetLangueSystem, which sends you back the value of the language of the systems pages in progress, i.e.: German (D), Spanish (S), English (E), French (F), Italian (I)

For this you can add this method at the start level of the application, i.e. in the procedure Form_Load(). The collected information is simply displayed in a label located at the bottom of our Excontrol application.

```
Private Sub Form_Load()
Dim Result As Long
Dim NbEquip As Long
Dim EquipList As Variant
Dim SymbList As Variant
//..
.../
Dim L As Integer
Result = ptrMagelis.GetLangueSystem(L)
If Result <> 0 Then
End If
Langue.Caption = Chr(L)
End Sub
```

8.2.3. Example of an Excel 97 Application

We will now show you how to use this server with Excel. You will see for yourself that the implementation is very similar to the Visual Basic application. We have selected the 1997 version of Excel as it is more advanced in terms of Visual Basic for Applications. Indeed to be able to use the asynchronous part of our server, the client must be able to reference itself to the server and therefore offer an object class, a capability that Excel 97 provides.

Our tutorial is not a training course in Excel or in Visual Basic for Applications but it does enable a customer application of our Magelis OLE server to be implemented simply. Should you wish to develop a major Excel application, you must refer to a reference manual.

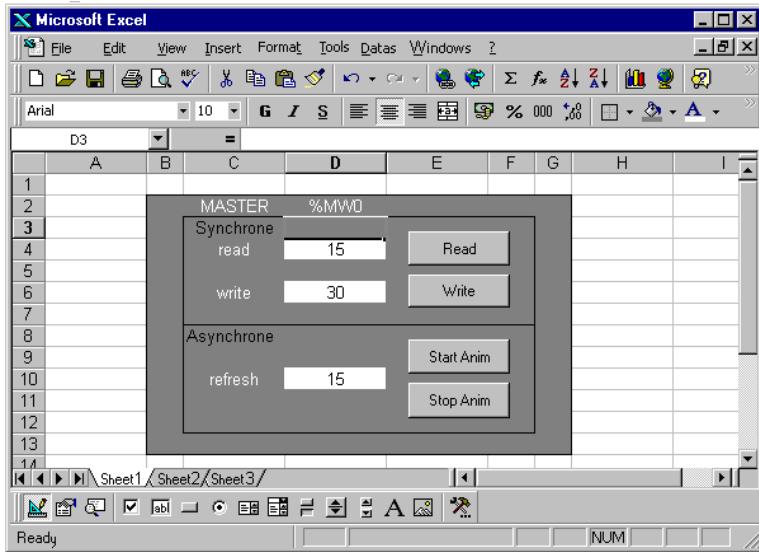
The application we are going to develop here is based on the principle used previously for Visual Basic application, which considerably simplifies the explanation. For a better understanding of the operation, we recommend you to read the description of the Control example in Visual Basic 5.0. VB and VBA are in fact very similar, the only difference being the Excel-specific procedures (accessing a cell).

This application will thus allow use of the synchronous read and write functions and the asynchronous read function.

To benefit from Excel, the read data can be, for instance, displayed as graphs, pie charts, etc.

8.2.3.1. Defining Our Application

Our application will be capable of reading and writing a PLC data element and of continuously monitoring the data element.



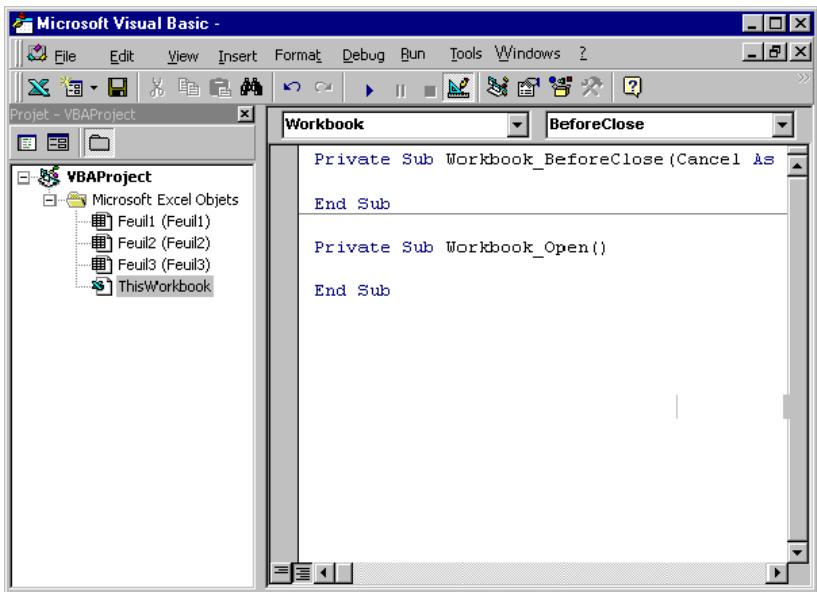
We are going to create the above sheet by adding the four buttons ("command" tool bar) and by changing their names and the labels via the properties menu linked with the button (click the button with the right mouse button).

Once this is done, the code linked with this sheet must be prepared. For this we will make use of macros. These command macros consist of Visual Basic for Applications code and perform actions on your sheets. To access Visual Basic, use the Tools\Macro\VisualBasicEditor menu. This opens the VisualBasicEditor application .

Once in the VisualBasicEditor, you will see that the bottom right of the window contains the description of your Excel directory, together with its three sheets and a module called ThisWorkbook, which is the Excel application containing your sheets. In this module, you can insert macros for opening and closing your directory, for accessing the Excel status bar, etc.

Moreover we will use this module to open and close the Magelis OLE server on opening and closing the directory, and we will indicate the changes in our actions in the status bar.

For this, double-click ThisWorkbook, then select the open and before close functions that we are going to complete.



Let us now begin with the global declarations of interest to us, i.e. the server and the callback class. For these variables to be global for the application, they must be declared inside a module. We will therefore create this module via the Insert/Module menu.

Application Public Variables

Public server As Object

Public callback As Object

After declaring our public variables, we must now deal with the server connection and disconnection, which requires entering the code in the This Workbook file.

Private Sub Workbook_Open()

```
Const progID$ << " Identify the MAGELIS server.
```

```
Const S_OK = 0 " Status OK returned by OLE Automation interfaces.
```

```
Dim result As Long
```

```
Application.StatusBar = «Connection to the Server in progress...»  
Set serveur = Nothing  
Set callback = Nothing
```

On Error Resume Next

```
Set serveur = CreateObject(«Grafsrv.srvDll») " Create the Automation Server  
object  
If (Err.Number <> S_OK Or serveur Is Nothing) Then  
    Application.StatusBar = «Connection to the Server Failed...»  
    End  
End If  
Set callback = New ClassCallback  
  
result = serveur.AddCallbackReference(callback)  
If result <> 0 Then  
    Application.StatusBar = «Error during Callback...»  
    End  
End If  
Application.StatusBar = «Connection to the Server success...»
```

End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)

```
Set callback = Nothing:  
Set serveur = Nothing  
Application.StatusBar = «You're disconnected from the Server»
```

End Sub

You can see that we have initialized a Callback object with a ClassCallback type. As this class must exist in your application, we are going to add it. For this, use the Insert/Class module menu and change the name in the properties window by "ClassCallback".

We will then add to this class the code for the server return to the asynchronous case.

Sub OnDataChange(hVar As Long, Value As Variant)

```
" called when a new value is available  
" note that the link between the request and the event is made by hVar  
Feuil1.AfficheResult hVar, Value
```

End Sub

In this code you can see that we will call up a procedure, which will be declared in Feuil1 and which is called AfficheResult.

All we have to do now is to define the button-related macros contained in Feuil1 and the AfficheResult procedure. Just double-click Feuil1 in the Project window; then add the following procedures:

- Private Sub CommandRead_Click()
- Private Sub CommandWrite_Click()
- Private Sub CommandStartAnim_Click()
- Private Sub CommandStopAnim_Click()
- Public Sub AfficheResult()

The Read command performs unitary reading of a PLC data element and utilizes the ReadVariable method of the Magelis server.

The Write command performs unitary writing of a value in the PLC data element and utilizes the WriteVariable method of the Magelis server.

The StartAnim command provides continuous reading of one or more PLC data elements and utilizes the ReadVariableContinue and ActiveReadContinue methods. These methods yield a number for the group of variables processed.

The StopAnim command completely halts the animation of a given group. It utilizes the StopReadContinue method of the Magelis server.

The AfficheResult procedure displays the data sent back by the server in the asynchronous case.

The related code is given below:

"global variable

Dim hGVar As Long "number of the refreshed group

Private Sub CommandRead_Click()

Dim result As Long

Dim Equip As String

Dim Rep As String

Dim Value As Variant

Equip = Feuil1.Cells(2, 3)

Rep = Feuil1.Cells(2, 4)

result = serveur.ReadVariable(Equip, Rep, Value)

If result < 0 Then

 Application.Statusbar = «Read error»

Else

 Application.Statusbar = «Read variable»

End If

Feuil1.Cells(4, 4) = Value

End Sub

Private Sub CommandStartAnim_Click()

Dim result As Long

Dim TabEquip(0) As String

Dim TabRep(0) As String

Dim Freq As Integer

TabEquip(0) = Feuil1.Cells(2, 3)

TabRep(0) = Feuil1.Cells(2, 4)

Freq = 1 "indicates refreshing every second

hGVar = serveur.ReadVariableContinue(TabEquip, TabRep, Freq)

If hGVar <= 0 Then

 Application.Statusbar = «Animation fault»

```
Else
    result = serveur.ActiveReadContinue(hGVar)
End If
```

```
End Sub
```

```
Private Sub CommandStopAnim_Click()
```

```
Dim result As Long

If hGVar <> 0 Then
    result = serveur.StopReadContinue(hGVar)
    hGVar = 0
End If
```

```
End Sub
```

```
Private Sub CommandWrite_Click()
```

```
Dim result As Long
Dim Equip As String
Dim Rep As String
Dim Value As Variant
```

```
Equip = Feuil1.Cells(2, 3)
Rep = Feuil1.Cells(2, 4)
Value = Feuil1.Cells(6, 4)
result = serveur.WriteVariable(Equip, Rep, Value)
```

```
If result < 0 Then
    Application.Statusbar = «Write error»
Else
    Application.Statusbar = «Write Ok»
End If
```

```
End Sub
```

Public Sub AfficheResult(hVar As Long, Value As Variant)

```
If hVar = hGVar Then  
    Feuil1.Cells(8, 4) = Value(0)  
End If
```

End Sub

8.3. Manipulated MAGELIS Objects

A review of the various needs indicates that the only manipulated MAGELIS "objects" (exposed to third-party applications) are as follows:

- variables used in communications (PLC variables),
- local variables belonging to the internal database.

These are the "Magelis variable objects".

These variable can be communicated between processes (applications) via the **MAGELIS OLE Automation server**.

8.3.1. Types of Variables Processed by MAGELIS

- **List of the Variables used in Communication**

The short list is available on the protocols, excepting XWAY. The full list is available on XWAY.



Short List

Addressable Object	Address	Range
internal word	%MW <i>i</i>	*
internal word table	%MW <i>i:j</i>	j<40
internal double word	%MD <i>i</i>	*
constant word	%KWi	*
internal bit	%Mi	0-255
internal bit table	%Mi: <i>j</i>	j<128
local input/output bit	%I <i>i.j</i>	* j=0-15
	%Q <i>i.j</i>	
remote input/output bit	%I <i>i.j</i>	* j=0-15
	%Q <i>i.j</i>	
system bit	%Si	0-127
step bit	%Xi	*

- **List of Local Variables**

- Last alarm,
- Number of alarms
- Page number.

The exhaustive list still has to be defined (it will be determined using the internal MAGELIS database).

8.3.2. Format of the Automation Variables Processed by MAGELIS

- Boolean
- word (integer over 16 bits)
- double word (integer over 32 bits)
- floating format (in 32-bit IEEE format)
- character string (maximum 80 ASCII characters)

8.3.3. Using Symbols

A variable can be associated with a symbol on the MAGELIS terminals associated with a Schneider protocol. These variable symbols come from an existing symbol base created by one of the Schneider PLC programming software applications (PL7, Concept...).

By importing a neutral address/symbol association file (.SCY for PL7, .TXT for ConCept) the client application, which will use the services of the MAGELIS OLE Automation Server, could work directly with the symbols, e.g. read the value of a variable symbolized by "SENSOR".

Refer to the GetRepere function.

A .SCY file is only associated with a single PLC. An equipment table is generated by XBTL1000 and it enables an equipment number to be cross-referenced with a .SCY file.

A symbol is therefore: <Num. Equipment>. <symbole>

8.3.4. Address

The variable accessing methods involve the use of an address parameter. An address is an equipment number followed by a PL7 or Modbus syntax address. The equipment number must be known (declared under XBT-L1000) and must be in the equipment table (Refer to application architecture).

8.3.5. Call Syntax

- The equipment items can be designated by their number or by their name.
- The variables can be designated by their address or by their symbol if an address/symbol association file has been loaded.

Examples of correct syntax:

Equipment	Variable	Comment
1	%MW55	Always accepted
MASTER	%MW12	Always accepted
1	Toto	Accepted if a file .SCY has been associated with equipment 1
MASTER	Toto	Accepted if a file .SCY has been associated with the Master equipment
0	SystemDate	Internal variable: Identified by Equipment 0. Accepted if the name is part of the internal MAGELIS variables (See list below)

- **List of Internal Variables**

SystemDate SYS_DATE

SystemHour SYS_TIME

DateFormat SYS_CURRENT_DATE_FORMAT

HourFormat SYS_CURRENT_TIME_FORMAT

VeilleTempo

Luminosity

Contrast

Language

LineSpeed

LineParity

LineStopBitNumber
SlaveNumber
CurrentPage
AlarmAcquittement
AlarmPresence
AlarmNumber
PasswordLevel

- **Programming on XBTL1000**

The "Symbols Files" will be transferred as supplementary files by XBTL1000.

8.4. Description of the Methods Available

The following 9 methods are available on the MAGELIS server:

- **GetEquipmentList**

Gets the list of the equipment items configured in the application as well as the list of the symbols files (scy) associated with the equipment.

- **GetLangueSystem**

Gets the language code of the systems pages in progress
" F ", " E ", " D ", " S ", " I " (French, English, German, Spanish, Italian).
The returned code is the ASCII value of the letter.

- **GetReper**

Associates the symbol/equipment item and the corresponding address.
This method is only valid when a symbol/address database has been imported into the application. The method must be applied each time symbols are used.

- **ReadVariable**

Reads the value of a variable in the PLC. Synchronous function.

- **AddCallbackReference**

Initializes the callback system of the client application

- **ReadVariableContinue**

Initializes the variable group for continuous reading (refreshed) and gets a handle on this group. To be used with ActiveVariableContinue and StopReadContinue.

- **ActiveVariableContinue**

Initiates reading of the variable group initialized in refreshing.

- **StopReadContinue**

Stops continuous refreshing initialized with ReadVarContinue and destroys the group handle.

- **WriteVariable**

Writes the value of a variable in the PLC. Synchronous function.

8.4.1. GetEquipmentList: Gets the Equipment and Symbol Files Lists

GetEquipmentList() is a synchronous function that gets the list of equipment items configured in the application and the list of associated scy files.

Definition of the Method

```
rc=GetEquipmentList(OUT ListEquipSymbole As string(), OUT  
ListFileSymb As string())
```

Declaration of the Variables

```
Dim ListEquip() As String  
Dim ListFileSymb() As String
```

Calling the Method

```
rc = TheGrafSvr.GetEquipmentList(ListEquipSymbole, ListFileSymb) ;
```

Return Codes

. S_OK	0	Read OK
· E_FAIL	0x80004005	When an error occurs (equipment file not found)
· other<0		Resulting value of IDispatch::Invoke

8.4.2. GetLangueSystem : Gets the Language of the System Pages in Progress

GetLangueSystem() is a synchronous function that gets the language code of the system pages in order to harmonize the language of a multi-language customer application.

Definition of the Method

GetLangueSystem (OUT Langue As integer) AS Long

Declaration of the Variables

```
Dim CodeLangue as integer  
Dim rc As Long
```

Calling the Method

```
rc=MagelisServer.GetLangueSystem(CodeLangue) ;
```

Return Codes

- | | | |
|----------|------------|--------------------------------------|
| · S_OK | 0 | OK |
| · E_FAIL | 0x80004005 | If the data is not available |
| · other | <0 | Resulting value of IDispatch::Invoke |

8.4.3. GetRepere : Gets the Address

GetRepere() is a synchronous function that gets the data (address) to send to ReadVariable, based on a symbol and an equipment number.

Definition of the Method

GetRepere(IN Equipement As String, IN Symbole As String, OUT & Repere As Variant) As Long

Declaration of the Variables

```
Dim Equipement As String  
Dim Symbole As String  
Dim Repere As Variant  
Dim Result As String
```

Calling the Method

```
rc=MagelisServer.GetRepere(Equipment,Symbol,Address)
```

Return Codes

- | | | |
|----------|------------|--------------------------------------|
| · S_OK | 0 | If it exists |
| · E_FAIL | 0x80004005 | If the address does not exist |
| · other | <0 | Resulting value of IDispatch::Invoke |

8.4.4. ReadVariable: Continuous Unitary Read

ReadVariable is a synchronous function that sends the read request to the PLC and waits for the reply.

Definition of the Method

```
ReadVariable(IN Equipement As String,IN Address or symbol As String,OUT &Data As Variant) ;
```

Declaration of the Variables

```
Dim Equipement As String  
Dim Symbole As String  
Dim Repere As String  
Dim Data As Variant
```

Calling the Method

```
rc = TheGrafSrv.ReadVariable(Equipement, Symbole, Data) ; // the value is selected by Data
```

Return Codes

· S_OK	0	Read OK
· E_FAIL	0x80004005	Read error
· E_INVALID_EQUIP	-1001	Equipment name invalid
· E_INVALID_REPERE	-1002	Address invalid
· E_INVALID_PROTOCOLE	-1003	Protocol not processed
· E_INVALID_VARSYSTEM	-1004	System variable not defined
· other		Resulting value of IDispatch::Invoke

N.B.

In the case of a word table (%Mwi:j) read, the result is given as a character string.

In the case of a bit table request, the result is given in a table.

8.4.5. AddCallbackReference / ReadVariableContinue / ActiveReadContinue

The asynchronous management system introduces three functions:

- AddCallbackReference allows the Client to transfer the reference of a callback class including a callback function OnDataChange. This action is performed at the startup of the customer application and it creates the Callback link of the entire application.
- ReadVarContinue is an asynchronous function that returns a group number on the operation so initialized.
- ActiveReadContinue is the startup function for the animation. The OLE automation server then maintains the communication and informs its client of the variable state change via the OnDataChange function of the Callback class.

AddCallbackReference(IN Callback As object)

hGroup=ReadVariableContinue(IN Equipement() As variant, IN Repere() As variant)

ActiveReadContinue(IN hGroup As Long)

Example:

TheGrafSrv.AddCallbackReference(Callback)

'For each asynchronous read

hGroup(n)=TheGrafSrv.ReadVariableContinue(Equipement,Repere)

ActiveReadContinue(hGroup(n))

The handle will be used to get the value of the read variable when the client is notified that this value has changed.

The client notification mechanism is as follows:

- The client's application contains a "callback object" (which we supply)
- Transfer the "callback object" address into the parameters of the AddCallbackReference.
- The OnDataChange procedure for this object will take over control when the value changes, with the continuous read handle involved and the value of the variable as parameters.

N.B.

- 1: The returned values are ASCII strings representing the data according to their formats (e.g.: for a floating format, %MF100 will for instance give "5.235E12")
- 2: The returned values are signed.
- 3: It is preferable to use groups of data to be read rather than one group per data element as this substantially increases the performance of the client application which ought to process only one callback message for all the variables.

```
OnDataChange(hRead As long, Value As Variant)
{
Select Case hRead

Case hGroup(0):
    For i=0 to UBound(Value)
        Form1.TextResultAnim(0).Text = Value(i)
    Next i

Case hGroup(1):
    For i=0 to UBound(Value)

        Form1.TextResultAnim(1).Text = Value(i)
    Next i

End Select
}
```

8.4.6. AddCallbackReference

Definition of the Method

AddCallbackReference(IN Callback) As Long

Declaration of the Variables

Dim Callback As Object

Dim rc As Long

Calling the Method

rc=TheGrafSrv.AddCallbackReference(Callback)

Return Codes

- | | | |
|-----------|------------|--------------------------------------|
| · S_OK | 0 | OK |
| · E_FAIL | 0x80004005 | initialize error |
| · other<0 | | Resulting value of IDispatch::Invoke |

8.4.7. ReadVariableContinue

Definition of the Method

ReadVariableContinue(IN Equipement(), IN Repere(), IN Freq) As Long

Declaration of the Variables

Dim Equipement(1) As String

Dim Repere(1) As String

Dim Freq As integer

Dim hGroup As Long

Calling the Method

hGroup=Magelis.ReadVariableContinue (Equipment, Repere, Freq) ;

Return Codes

· hGroup	>0	OK
· Group create error	0	Readerror
· E_INVALID_EQUIP	-1001	Equipment name invalid
· E_INVALID_REPERE	-1002	Address invalid
· E_INVALID_PROTOCOLE	-1003	Protocol not processed
· E_INVALID_VARSYSTEM	-1004	System variable not defined
· E_ARRAY_DEF	-1005	Table incorrectly defined
· Other	<0	Resulting value of IDispatch::Invoke

8.4.8. ActiveReadContinue

Definition of the Method

ActiveReadContinue (IN hGroup) As Long

Declaration of the Variables

Dim rc As Long

Dim hGroup As Long 'value already obtained by ReadVariableContinue method

Calling the Method

rc = MagelisServer.ActiveReadContinue (hGroup) ;

Return Codes

· S_OK	0	Read OK
· E_FAIL	0x80004005	Run error
· other	<0	Resulting value of IDispatch::Invoke

8.4.9. StopReadContinue

This method must be called to stop the refreshing of the variables requested by ReadVariableContinue/ActiveReadContinue

Definition of the Method

StopReadContinue(hGroup) As Long

Declaration of the Variables

Dim hGroup As Long 'value declared as global, obtained by ReadVariableContinue

Dim rc As Long

Calling the Method

rc=MagelisServer.StopReadContinue (hGroup) ;

Return Codes

- | | | |
|----------|---|--------------------------------------|
| · S_OK | 0 | ReadOK |
| · E_FAIL | | Readerror |
| · other | | Resulting value of IDispatch::Invoke |

8.4.10. WriteVariable: Write

WriteVariable() is a synchronous function that writes the value of a variable.

Definition of the Method

WriteVariable(IN Equipement,IN Repere ou symbole, OUT &Data) As Long

Declaration of the Variables

```
Dim ListEquip() As String  
Dim ListFileSymb() As String  
Dim rc As Long
```

Calling the Method

```
rc=MagelisServer.WriteVariable(Equipement,Repere,&Data) ;
```

Return Codes

· S_OK	0	Read OK
· E_FAIL		Readererror
· other		Resulting value of IDispatch::Invoke

8.5. Debugging an OLE Automation Client Application

The OLE automation client application can be debugged directly on the TXBT provided the development environment is on the TXBT.

To debug the OLE automation client application on the development PC on which XBT-L1000 runs, the following are required:

- XBT-L1000 version 3.30 or up
- Windows NT (4.0 or up)

In this case, proceed as follows:

- Create an application for the TXBT on XBT-L1000 by defining the required equipment.
- There is no need to create application pages.
- Run the simulation.
- The Magelis OLE automation server is now enabled and can execute the methods called by the client application being debugged.
- A trace file c:\OLETrace.dbg is created; it contains the trace of all the calls of methods and gives the cause if they fail. This file can only be looked up when the simulation is shut down.

N.B. The simulation window is not representative of the variables accessed by the OLE automation client. Only those variables accessed by a ReadVariableContinue function will be displayed in the window, with a variable name always equal to "%MW1111".

9. Connecting the TXBT to an Intranet

9.1. Selecting the Network Card, and Installing

Select a "Plug and Play" card for easy installation.

The example below covers the installation of an "Etherlink III 3C509 Combo" card.

- 1) Power down the TXBT.
- 2) Insert the card.
- 3) Power up the TXBT.
- 4) Reboot Windows. The card will be detected and the relevant software installed without operator action.
- 5) Reboot the TXBT after completion of the installation.

9.2. Setting the Network Parameters

The network parameters are set via Start/Parameters/Configuration panel/Network parameters.

This function is used to select the protocol (usually Microsoft-TCP/IP), assign the TXBT a name on the network and to share files with other computers.

You are recommended to refer to Chapter 3 in Microsoft's Windows publication and to seek information from the manager of the network on which the TXBT is to be connected.

Appendix: Interpreting the Error Codes

Code 1

Memory allocation error: reboot the TXBT.

Code 2

Equipment not known: check the address of the equipment item configured in XBT-L1000, use Magelis Tools/Xwin32.

Code 3

Variable not known in XBT-L1000: retransfer the application.

Codes 258/514

Socket number invalid: check the BASE and NOMBRE (number) parameters in the case of the UNI-TELWAY protocol (Magelis Tools/UNI-TELWAY).

Code 513

Communication driver not installed: check CONFIG.SYS or the registry as per type of Driver; reboot the TXBT.

Code 523

All the communication buffers have been used: increase the value of the parameters RBUF and SBUF in the configuration files of the drivers.

Code 525

Time-out on request emission, equipment involved is not connected, or the address configured in XBT-L1000 does not match the true address.