

# MAC Operator Terminal

Human-Machine-Interface

User's Manual

## **MAC E900**

## Manual E900

### Foreword

This manual is a programming and function description for the operator terminals E900 and E910 and the programming software MAC Programmer+. MAC Programmer+ and SW-MTA-WIN are the same programming software. In the manual we assume that a PLC system from Mitsubishi Electric is connected to the terminal. When using other PLC brands we refer to the respective driver manual for the PLC for specific details. The functions in the terminals are the same, regardless of the terminal used. Beside this manual, the following manuals are also available:

- MELSEC FX, Programming manual
- MELSEC FX, System manual
- A-CPU, Programming manual
- E900/E910, Installation
- IFC PBDP, Manual
- IFC 128E
- IFC MC
- IFC PI
- IFC ETTP, IFC ETCX

© MITSUBISHI ELECTRIC EUROPE B.V. 2000

All examples in this manual are used solely to promote understanding of how the equipment works and its operation. Mitsubishi Electric Europe b.v. take no responsibility if these examples are used in real applications.

Because of the great many application areas for this equipment, the user himself must acquire the appropriate knowledge needed to use the equipment correctly for particular applications.

Mitsubishi Electric Europe b.v. absolves itself of all responsibilities for damage and injuries that may occur during installation or use of this equipment.

Mitsubishi Electric Europe b.v. absolves itself of all responsibilities for any damage caused by modifications made to the equipment.



# **Safety precautions**

## **General**

- Check the delivery for transport damage. If damage is found, advise your supplier.
- The product fulfils the requirements of article 4 of EMC directive 89/336/EEC.
- Do not use the product in an environment with high explosive hazard.
- Modifications, changes, additions and/or alterations to the product are forbidden.
- Use with this product only parts and accessories manufactured according to specifications of Mitsubishi Electric Europe b.v.
- Read the installation and operating instructions completely and carefully prior to installation, use, or repair of this product.
- This equipment should only be operated by qualified personnel.

## **At installation**

- The product is designed for stationary installation.
- Install the product according to the accompanying installation instructions.
- The product must be grounded according to the accompanying installation instructions.
- This equipment must be installed by qualified personnel.
- High voltage, signal and supply cables must be separated.
- The product should not be installed in direct sunlight.

## **In use**

- Keep the equipment clean.
- Emergency stop and other safety functions should not be controlled from the terminal.
- Do not touch the keys, displays, etc. with sharp objects.

### **Service and maintenance**

- The agreed warranty applies.
- Clean the display and face with a soft cloth and mild detergent.
- Use batteries specified by Mitsubishi Electric Europe b.v. Batteries should be changed by qualified personnel. The person changing the batteries should be grounded during the operation; e.g. with a grounded wrist strap.
- Repairs should only be made by qualified personnel.

### **At disassembly and scrapping**

- Local regulations apply concerning recycling of products or part.
- Please note that the electrolyte condenser and display contain hazardous substances.

# Content

<b>1 Introduction .....</b>	<b>1-1</b>
1.1 Programming .....	1-3
1.2 Connecting the terminal to the PLC system .....	1-4
1.3 Status display and control .....	1-4
1.4 Location of the terminal .....	1-5
1.5 Compact solutions .....	1-6
<b>2 Installation of the software .....</b>	<b>2-1</b>
2.1 MAC Programmer+ .....	2-1
2.2 Connecting the terminal to a personal computer .....	2-3
2.3 Downloading the system program .....	2-3
<b>3 Terminal functions .....</b>	<b>3-1</b>
3.1 The operation modes, RUN, PROG and SETUP .....	3-1
3.2 The terminal's keyboard .....	3-6
3.3 Setting the real time clock .....	3-9
3.4 Information page .....	3-10
3.5 "Joystick" function .....	3-11
<b>4 General .....</b>	<b>4-1</b>
4.1 Method for programming a project .....	4-1
4.2 Efficient communication .....	4-2
4.3 Menu structure .....	4-5
4.4 Blocks .....	4-6
4.5 Signal types i MELSEC .....	4-7
4.6 Signal format .....	4-9
<b>5 Programming using the PC software .....</b>	<b>5-1</b>
5.1 Creating a project .....	5-1
5.2 Changing project settings .....	5-4
5.3 Creating a block via the Block Manager .....	5-5
5.4 Programming blocks .....	5-12

5.5 The graphic block editor .....	5-13
5.6 The textblock editor .....	5-23
5.7 The symbol manager .....	5-25
5.8 I/O change .....	5-28
5.9 Change BDTP station .....	5-29
5.10 I/O cross reference .....	5-30
5.11 Other editors.....	5-31
5.12 The File Menu.....	5-32
5.13 The Edit Menu .....	5-33
5.14 The View Menu .....	5-33
5.15 The Functions Menu.....	5-36
5.16 The Setup Menu .....	5-37
5.17 The Block Manager Menu .....	5-50
5.18 The Transfer Menu .....	5-50
5.19 The Window Menu.....	5-51
5.20 The Help Menu.....	5-52
<b>6 Graphic presentation and maneuvering .....</b>	<b>6-1</b>
6.1 General parameters .....	6-1
6.2 Graphic objects .....	6-2
6.3 Maneuvering Graphic Blocks.....	6-32
6.4 Using the touch screen .....	6-35
<b>7 Text-based presentation and maneuvering.....</b>	<b>7-1</b>
7.1 General parameters .....	7-1
7.2 Text object.....	7-2
7.3 Maneuvering Textblocks .....	7-11
<b>8 Historical trends .....</b>	<b>8-1</b>
8.1 Calculate the trend size .....	8-2
8.2 Defining trend objects .....	8-3
8.3 Transferring trend data .....	8-5
<b>9 Message library .....</b>	<b>9-1</b>

<b>10 Alarm handling</b> .....	10-1
10.1 Alarm grouping .....	10-2
10.2 Alarm messages .....	10-3
10.3 Configuring the alarm list.....	10-4
10.4 Alarms in run-time mode .....	10-6
<b>11 Recipes</b> .....	11-1
11.1 Calculate the recipe size.....	11-2
11.2 Creating a recipe with the terminal .....	11-2
11.3 Transferring a recipe to the PLC system.....	11-3
11.4 Erasing a recipe .....	11-4
11.5 Using recipes in a personal computer .....	11-4
11.6 Loading and saving recipes via the PLC program .....	11-5
<b>12 Passwords</b> .....	12-1
12.1 Defining security levels.....	12-1
12.2 Defining passwords .....	12-2
12.3 Logging in .....	12-3
12.4 Password for transfer .....	12-3
12.5 Password for all terminal functions.....	12-3
12.6 Altering passwords in run-time mode .....	12-4
<b>13 Report print-outs</b> .....	13-1
13.1 Connection to a printer .....	13-2
13.2 Text block printouts .....	13-2
13.3 Graphic block printout.....	13-3
13.4 Defining the print-out .....	13-4
13.5 Printer settings .....	13-4
13.6 Setting a communication port.....	13-6
13.7 Control codes to printer .....	13-6
<b>14 Time control</b> .....	14-1
14.1 Defining the time channels .....	14-1
14.2 Presentation in run-time mode .....	14-2



<b>15 System monitor</b> .....	15-1
<b>16 Index addressing</b> .....	16-1
<b>17 Communication</b> .....	17-1
17.1 Transparent mode .....	17-1
17.2 The terminal as communication interface (No Protocol Mode).....	17-5
17.3 Modem connection .....	17-13
<b>18 Network communication</b> .....	18-1
18.1 Examples of possible networks .....	18-2
18.2 Network communication via ETHERNET .....	18-11
18.3 Serial network communication .....	18-16
18.4 Network services .....	18-28
18.5 Network accounts.....	18-43
<b>19 Network functions in the terminal</b> .....	19-1
19.1 FTP server.....	19-1
19.2 SMTP client.....	19-5
19.3 Terminal reflection – applet .....	19-6
19.4 Web server.....	19-7
19.5 Recommendations and limitations for network communication .....	19-18
<b>20 LEDs</b> .....	20-1
20.1 External LEDs in E900 VT .....	20-1
<b>21 Function keys</b> .....	21-1
21.1 Definitions .....	21-1
21.2 Jump to block with the function keys .....	21-5
21.3 External function keys in E900 VT .....	21-5

<b>22 Expansion cards</b> .....	22-1
22.1 IFC PBDP .....	22-1
22.2 IFC MC .....	22-1
22.3 IFC 128E .....	22-1
22.4 IFC ETTP and IFC ETCX.....	22-1
22.5 IFC PI.....	22-2
22.6 IFC GA .....	22-2
<b>23 Project transfer</b> .....	23-1
23.1 Terminal settings.....	23-2
23.2 Transfer settings.....	23-2
23.3 TCP/IP transfer.....	23-5
23.4 Serial transfer .....	23-6
23.5 Transfer via modem.....	23-7
<b>24 Technical data</b> .....	24-1
24.1 Membrane keyboard .....	24-4
24.2 Drawings .....	24-7
24.3 Character tables.....	24-11
<b>25 Appendix</b> .....	25-1
<b>Index</b> .....	I



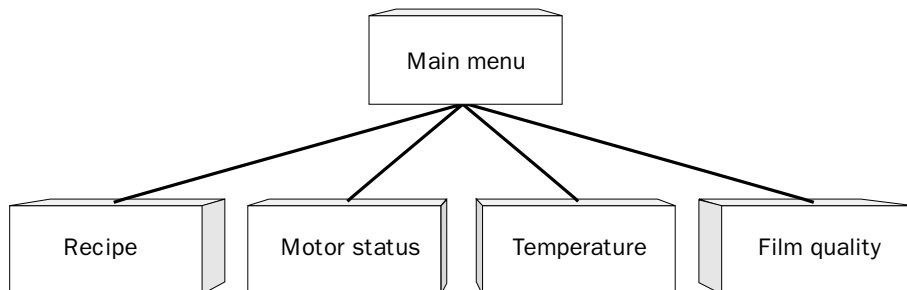
# 1 Introduction

The pace of production in today's industrial work environment increases steadily. At the same time, the operator's job responsibilities increase and become more comprehensive. Quickly and simply, the operator must get machine status information and, equally simply, must be able to control the machine's operation. The PLC system's functions increase and become more advanced, making it possible to effectively and efficiently control complicated processes. With the *operator terminal* from Mitsubishi Electric Europe b.v., *human-machine* communication is made simple and safe – even for the most advanced processes.

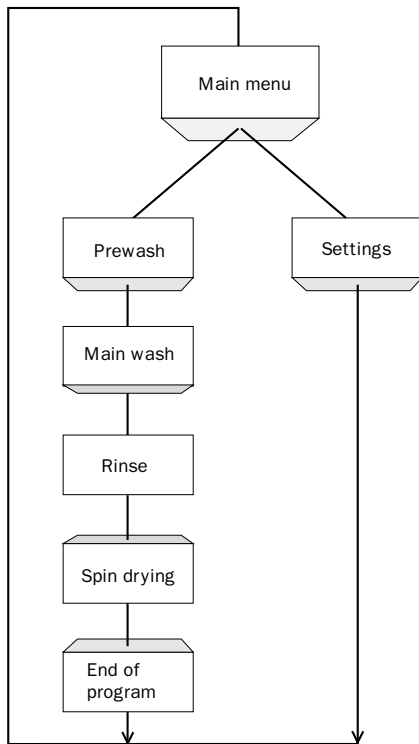
The graphical operator panels were developed to meet the requirements of man-machine communication for supervision and control in a variety of usage areas in manufacturing and process industries, etc. The terminal simplifies the operator's work because it is easily adapted to the operator's work environment. This means that the operator can continue to use the terms and protocols he or she is familiar with.

In the terminal, *projects* can be built up as *menu hierarchies*, or as *sequences*. A menu hierarchy comprises a main screen (with, for example, an overview) and a number of subsequent screens with more detailed information for a particular section. The operator can select which screen is displayed.

In the operator terminal, a screen is called a *block*.



A sequence starts with a main menu, from which the operator chooses a sequence where the blocks are displayed in a pre-set arrangement. Normally, the blocks are controlled by the PLC program.

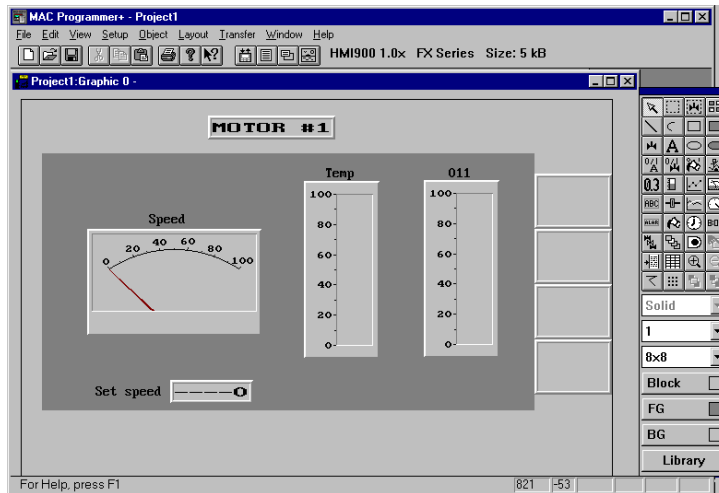


The terminal's functions enable either graphical or alphanumeric *presentation* and *control*. Additionally, there are functions for *alarm handling*, *report printout*, *realtime trend*, *recipe handling*, *time control*, and more.

The terminal's functions are not only simple to manage, they also provide clear cost advantages compared to conventional systems with buttons, indicator lamps, time relays, preset counters, and seven-day clocks. The terminal also offers functions which facilitate a more comprehensive use of the PLC system.

## 1.1 Programming

The terminal is programmed through a PC using the *MAC Programmer+* software. Hereafter this product is referred to as the *PC software*. It is also possible to program the terminal through the built-in keyboard. The E910 is a terminal with a touch screen and has no built-in keyboard which means that the E910 must be programmed using the PC software. Functionally the terminals are identical.

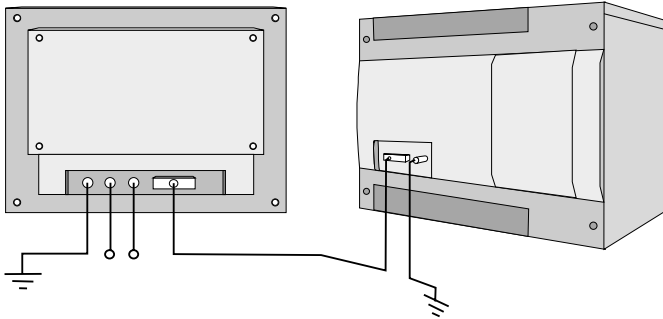


For the most part the terminal is *object oriented*, that is its work is based on an object, and the functions this is to have are determined afterwards. All types of signals are defined on this principle.

The programmed project is stored in the terminal.

## 1.2 Connecting the terminal to the PLC system

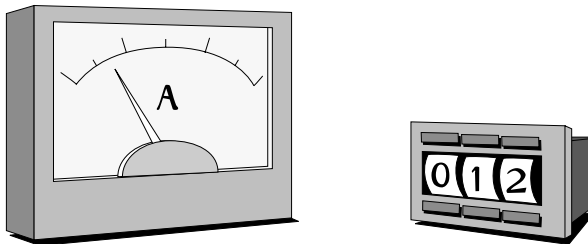
A terminal couple to the PLC system offers many benefits: the user does not need to make changes or additions to existing PLC programs; the terminal does not block any of the PLC system's inputs or outputs; and it improves the clarity and accessibility of such PLC system functions as time control and alarm handling.



## 1.3 Status display and control

The operator is very familiar with indicator lamps, as well as analog and digital displays for showing status as they appear today in the vast majority of applications. The same applies to push buttons, rotary and thumbwheel switches for controlling a system. Replacing them with a terminal provides the operator with all status display and control systems in one unit.

The operator can see and control all the information in the PLC system using the nomenclature of the machine. By doing so the operation of the complete system is not only enhanced but also simplified.



This is possible with a terminal thanks to the fact that the interchange of all information takes place through so-called blocks in the terminal. A block can be defined as a Textblock with only text information, or a Graphic block with full graphics.

The terminals have function keys for direct control. When an operator presses/energizes a function key during run, the digital signal in the PLC system linked to the function key will be switched on.

If several blocks are used, the operator can use jump objects to move between different blocks. In this way, a menu tree is created, allowing a structured application.

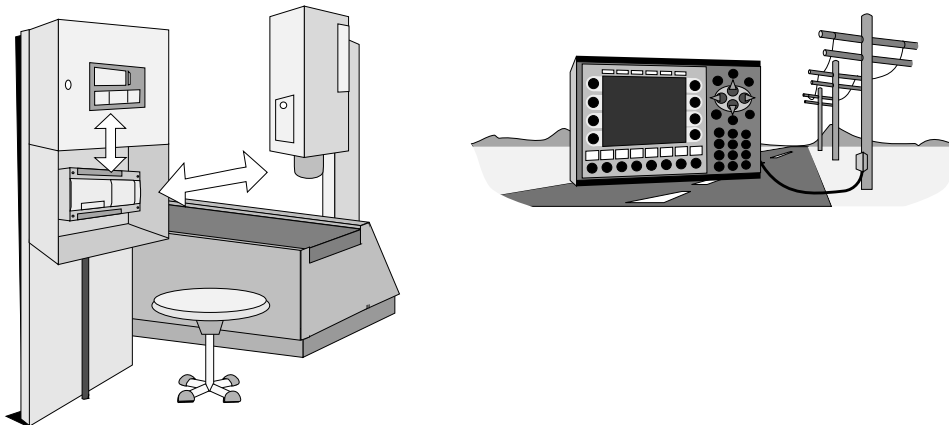
## 1.4 Location of the terminal

The terminal should be located close to the machine, to have maximum usability. The operator then receives the correct information directly and can react quickly and efficiently.

In some cases, such as monitoring/running or maintenance, the operators workplace can be several kilometres from the process. Communication is via modem.

Several terminals can be installed, if the production line is long with a large number of work stations.

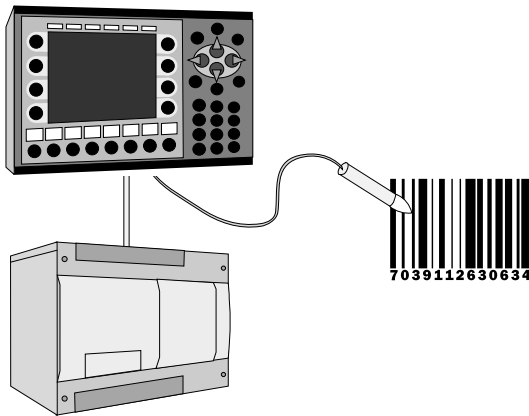
A unit working in parallel can also be connected. This could be another terminal. It could also be a PC with the programming tool MELSEC MEDOC. It is then possible to program the PLC system via the terminal at the same time as the terminal communicates with the PLC system.





## 1.5 Compact solutions

External units such as barcode readers, weighing machines, telephone modems and so on can be connected to the PLC system via the terminal. All the connection requires is that the unit can be connected to an RS-232 interface and that the communications is by ASCII-protocol. Data entering the terminal is written directly to the PLC register.



## **2 Installation of the software**

### **2.1 MAC Programmer+**

MAC Programmer+ is a program package used to develop projects for operator terminals from Mitsubishi Electric Europe B.V. The functions in the MAC Programmer+ depend on the terminal selected.

We recommend you use the mouse when using the MAC Programmer+. For short commands we refer to the Windows User's Guide.

In the MAC Programmer+ you create your project with graphic blocks and textblocks and then transfer them to the terminal. The programming procedure is described in the chapter "Programming using the PC software".

Help texts are available for all functions. By pressing the F1 key you will get assistance for the current function. You can also press the help button in the toolbar and the shape of the cursor changes, then click on the function you want information about.

### **System requirements**

MAC Programmer+ should be running on a computer with 50 Mbytes of free space and Microsoft Windows 95/98/NT/2000. The MAC Programmer+ can be used with either color or monochrome monitors.

### **Installing the MAC Programmer+**

The MAC Programmer+ is supplied on a CD. Place the CD in your CD ROM drive and the setup of the programs will start automatically. If not, call up **RUN** from the **Start** menu and type D:\setup.exe (assuming D: is your CD drive name). Choose to install MAC Programmer+ by clicking on the name and follow the instructions.

The setup will create an icon for MAC Programmer+ and the manual in the MAC Programmer+ group. You will find MAC Programmer+ and the manual by clicking on Start and choosing Program/MAC Programmer+. The manual can also be read directly from the CD by clicking on Manuals.

## The menu bar

The menu bar contains a number of pull-down menus; **File**, **Edit**, **View**, **Functions**, **Setup**, **Object**, **Transfer**, **Window** and **Help**. The File menu contains functions which affect the whole project. In the Edit menu you find functions for cut, copy and paste. The View menu contains the different editors, such as the **block editor** and **symbol manager**. In the menu **Functions** LEDs, passwords etc are defined. The basic settings for the terminals are made in the Setup menu. The Object menu is only available in the editors and contains all the objects. The objects can also be found in the toolbox. The Transfer menu is used for transferring projects between the MAC Programmer+ and the terminal. In the Window menu you will find the windows functions e.g. zoom in/out, show/hide the toolbox, status bar etc. Furthermore you make the grid settings and state the path to a external program e.g. Paint. In the Help menu you will find help functions for the program.

## The status bar

The status bar is displayed at the bottom of the MAC Programmer+ window. To display or hide the status bar, use the Status Bar command in the Window menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of the toolbox button that the cursor points at.

The right areas of the status bar indicate which of the following keys are locked on:

**OVR**    Overwrite (Insert key)

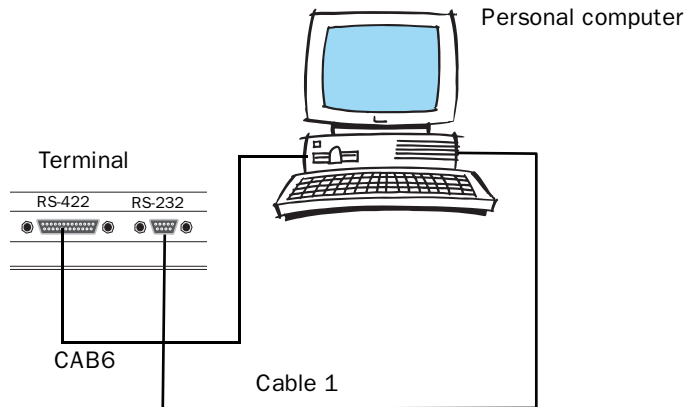
**CAP**    The Caps Lock

**NUM**    The Num Lock

Coordinates, line and column, in the block editor are also displayed.



## 2.2 Connecting the terminal to a personal computer



Name	Continentaleurope	GB	Rest of the world
Cable 1	MAC-PC-CAB-R2	MAC-PROG/9-CAB	MAC-PROG/9-CAB

Use the cable recommended in the driver manual for the selected PLC system for connection to the terminal.

### Note!

The power should be switched off during the connecting procedure.

## 2.3 Downloading the system program

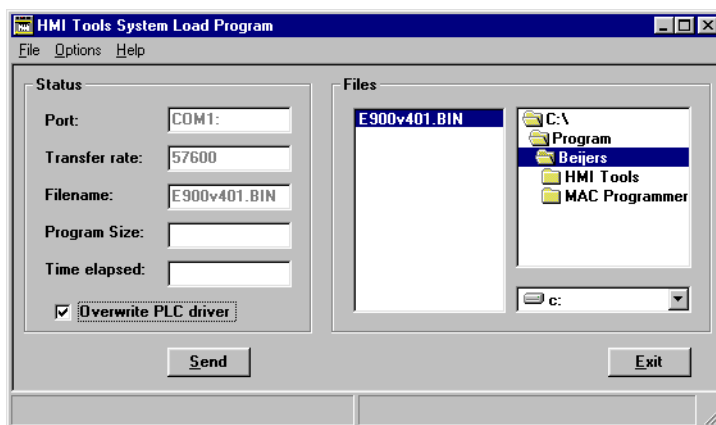
The terminal has a system program (operative system) which is stored in the terminal's memory when delivered. The system program can be exchanged, for example when updating to newer versions. The following is required to transfer the system program to the terminal:

- Personal computer
- Transfer cable between the PC and the terminal.
- The System Loader personal computer program (available as an icon in the program group HMI Tools). See the HMI Tools manual for installation of the program group.
- File with the new system program (e.g. E900V500.BIN).

## Transfer in the following steps:

1. Connect the cable between the personal computer and the terminal.
2. On the E-terminal you hold the [←] and [F1] keys pressed on start up, when the main cable (voltage feed is connected, to set the terminal in sysload-position. An LED should then blink green and the screen should be off. On the touch terminal E910 just make sure that the switch on the back of the terminal is in position 1.
3. Selecting Program/HMI Tools/System Loader in the Start menu starts the personal computer program. No settings need to be made on the E-terminal.

No settings need to be made on the terminal.



Under **Options, Settings** you set the communication port and transfer speed for the communication.

Down-loading of the system program can also take place through a modem for remote updating of terminals. Then the baudrate must be 2400 baud.

---

### Note!

When downgrading a terminal it is important to make sure that the square "**Overwrite PLC driver**" is selected.

---

## 3 Terminal functions

This chapter describes the different modes in the terminal, the keyboard and the information page in the terminal.

### 3.1 The operation modes, RUN, PROG and SETUP

The E900 has three operating modes. The E910 do not have programming mode, PROG.

- *SETUP mode*, where the basic settings are made such as choice of PLC system, menu language and printer settings.
- *Programming mode*, where the terminal can be programmed and changes can be easily made at system set-up.
- *Run-time mode*, when the terminal is running with a PLC.

#### Switching between modes

##### Switch between PROG and RUN

Switch between programming, PROG, and run-time, RUN, by simultaneously pressing the [←] and [MAIN]. The possibility to switch between PROG and RUN from the terminal can be controlled via a password. For more information we refer to the chapter “Programming using the PC software”.

##### Switch between PROG and SETUP

To switch from PROG to SETUP you press [LEAVE] until the start-up menu is displayed, then press any key. Press [LEAVE] to return to PROG.

##### Switch between RUN and SETUP

Press [←] and [MAIN] simultaneously to enter programming mode, PROG. Then press any key when the start-up menu is displayed to enter the SETUP mode. To switch from SETUP mode to RUN you press [←] and [MAIN].

Put the switch on the back of the E910 in position 4 to enter the SETUP mode. When the switch is in position 0 the E910 is in Run-time mode. See also the section The switch on E910.

## The SETUP mode

In this section we describe functions that cannot be made via MAC Programmer+. For settings via MAC Programmer+ we refer to the chapter “Programming using the PC software”.

### Erasing the memory

In the SETUP menu in the terminal you will find the **Erase memory** function. This function erases the terminal’s application memory. All blocks and definitions of alarms, time channels, function keys and system signals are erased.

Parameter	Description
<b>Enter</b>	The memory is erased. The configuration menu is shown automatically when the erasure is completed.
<b>Prev</b>	Return to the previous level without erasing the memory.

---

#### Note!

All data stored in the terminal is lost when the memory is erased. The language parameter are not affected by this function. Other parameters are erased or allocated the default setting values.

---

### Memcard maintenance

Under **Memcard maintenance** you select to format an installed PCMCIA memory card or to transfer to and from the memory card. You can format the memory card as a backup card or as an expansion card. Under **Mem-Card** transfer you can select to transfer: project to memcard, project from memcard, delete memcard project, recipe to memcard, recipe from memcard and delete memcard recipes. For more information see the manual for IFC MC.

### Contrast setting

The contrast settings can be made in setup mode in the menu **Setup, Contrast settings** or in run-time mode via a jump to the system block 997. Pressing the function key marked + makes the display brighter and pressing the function key marked - makes the display darker. Return to the previous level by pressing EXIT.

## The Prog mode

The programming mode is the mode where the application can be modified. It is here you can change textblocks and graphic blocks, modify the time channels, alarms, recipes and function keys. We recommend that the terminal is programmed using the MAC Programmer+ software.

---

**Note!**

Block 0 must be defined in the project in order to be able to run the application.

---

### Edit

#### Editing graphic objects

The arrow keys are used when editing a graphic object to change the size of or move the object in 8-pixel steps. Changing/ moving pixelwise is done by pressing [SNAP OFF] in the help line.

To move stepwise through all the objects in the editing mode press the [SELECT] key followed by [NEXTOBJ] in the help line. The object marked is inverted and the object type is shown in a square.

When an object is marked press [EDITOBJ] to edit the object's parameters, [DELOBJ] to erase the object and [LEAVE] to return. Point to an object and press [MOVE] or [SIZE] and make the changes with the arrow keys to move or change the size of the object.

#### Editing text objects

Mark an object by moving the cursor to the object with the arrow keys and press [Next] in the help line. You can scroll the text with the arrow keys if the Text-block holds more information than can be shown on the display. Press [LEAVE] in the help line to exit from a menu.



**Help line**

There is a help line in the terminal which contains functions linked to the function keys located directly under the line. The functions of the keys change depending on which level is active. The help lines are only shown in the programming mode.

Function	Description
<b>NEXT</b>	Used to go to the next level. Settings made are saved.
<b>LEAVE</b>	Return to the previous level.
<b>TOGGLE</b>	Switch alternately.
<b>LIST</b>	Show option list.
<b>INSERT</b>	Insert.
<b>DELETE</b>	Delete.
<b>HELP OFF</b>	Switch between showing and removing help line.
<b>SELECT</b>	Mark an object.
<b>SNAP OFF/ON</b>	SNAP ON means moving and size change in 8-pixel steps. OFF = 1-pixel steps.
<b>NEWOBJ</b>	Create a new object.
<b>PREVOBJ</b>	Show previous object.
<b>NEXTOBJ</b>	Show the next object.
<b>EDITOBJ</b>	Edit an object.
<b>DELOBJ</b>	Delete an object.
<b>MOVE</b>	Move an object.
<b>SIZE</b>	Change the size of an object.
<b>ENLARGE</b>	Increase the character size in an object.
<b>REDUCE</b>	Decrease the character size in an object.

**Transfer**

Here you manually put the terminal in transfer mode. When the terminal is in transfer mode it is possible to transfer the project between the terminal and the MAC Programmer+. Using the function Automatic terminal RUN/TRANSFER switching in MAC Programmer+ the software automatically puts the terminal in transfer mode.

## **Run**

Run-time mode is where the application is running. When switching to run-time mode, block 0 is automatically displayed.

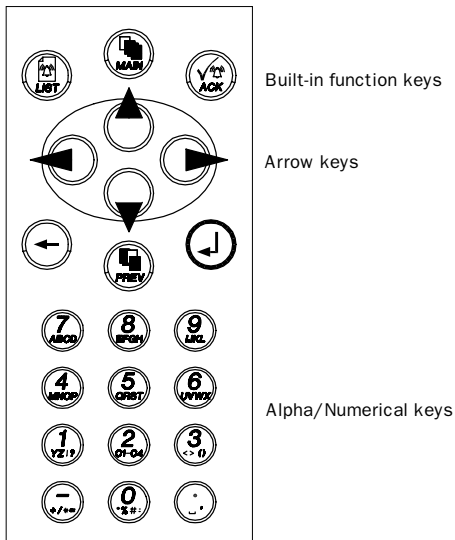
In the terminal the built-in keyboard is used to select and change values in run-time mode.

An error message is shown on the display if a communication error should occur between the terminal and the PLC system, See "Appendix". The terminal starts automatically when communication restarts. Key sequences affecting I/O that have been pressed during a communication error are stored in the terminal buffer and will be sent to the PLC system when the communication is running again.

To get a watch-dog function, where the PLC system can detect if a communication error has occurred, the terminal clock can be continuously transferred to a register in the PLC system (see the section Date/Time format in the chapter "Programming using the PC software"). The PLC system checks if the register is updated and if not, an alarm indicating communication error can be set in the PLC system.

How the different objects/functions work in run-time mode are described in the respective sections of this manual.

## 3.2 The terminal's keyboard



### Alpha/numerical keys

From the alpha/numerical keyboard the following characters can be entered in dynamic text and numerical objects during run-time in the terminal.

0-9

A-Z

a-z

! ? < > ( ) + / \* = ° % # : ' @

National characters

---

#### Note!

The @ character can only be entered in terminals with the system program version 5.0 or higher.

---

Numeric values are entered by pressing the respective key once. A to Z are entered by pressing the respective key 2 to 5 times and the lower case (small) letters a to z are entered by pressing the respective key 6 to 9 times. The time interval between pressing can be set (see the section Key delay time in the chapter “Programming using the PC software”). If the key is not pressed within the time interval, the cursor is moved to the next position.

By pressing the key C1-C4 two to nine times different national characters are entered. This makes frequently used characters that are not standard on the alpha/numerical keyboard available in the terminal (see the section Country settings in the chapter “Programming using the PC software”).

---

**Note!**

In MAC Programmer+ all characters in the used font table (see the section Country settings in the chapter Programming via MAC Programmer+) except reserved characters can be used as a static text. Write the wanted character by pressing the ALT key and at the same time write 0 (zero) on the numerical keyboard on the PC followed by the character code (see Font tables in the “Appendix”).

---

## Reserved characters

The ASCII characters 0-32 (Hex 0-1F) and 127 are reserved for internal functions in the terminal and may not be used in any project or file in the terminal. The characters are used as control characters.

## Arrow keys

The arrow keys are used to move the cursor within a block, menu or dialog.

## Built-in function keys

All keys are not available on all terminals.

The [↵] ([ENTER]) key is used to confirm settings made and go on to the next line/level.

The [PREV] key is used to return to the previous block. If the security level is changed the [PREV] key is disabled.

The [LIST] key is used to bring up the alarm list.

The [ACK] key is used to acknowledge an alarm in the alarm list.

The [MAIN] key is used to jump to block 0 in run-time mode.

The [←] key is used to delete the character to the left of the cursor.

---

**Note!**

When the main block (block number 0) has been shown the [PREV] key does not work because the historical blocks are deleted when the main block is shown.

---

---

**Note!**

With the IFC 128E expansion card it is possible to put all the keys on an external keyboard. For more information see the IFC 128E manual.

---

## Key sequences

There are some key sequences in the terminal with the following functions.

Key sequence	Function
[←] [MAIN]	Switch between PROG and RUN.
[←] [F1]	Hold this combination pressed during start up to activate the mode for downloading of system programs, see the Download the system program section.
[←] [PREV]	Calls up the information page in run-time mode, see the “Information page” section.
◀ ▶	Hold this combination pressed during start up to activate the self-test function.

## The switch on E910

To enter the different modes in the E910 you set the switch on the back of the terminal, the switch has the following functions.

Switch position	Function
0	Run-time mode (regular running).
1	Sysload
2	Calibrate touch.
3	Dragging cross.
4	Setup mode.
8	Activates the self test function.
9	Clear the clock RAM.

## 3.3 Setting the real time clock

The real time clock in the terminal is set under **SETUP, Date/Time**.

Select the alternative Set Terminal clock. The date and time are now displayed. Press [SET] to change the settings. Enter required date and time. In the editing mode you move the cursor with the arrow keys. Press [NEXT] to return or interrupt the editing before [ENTER] is pressed.

The real time clock can also be set in run-time mode via a maneuverable clock object.

### 3.4 Information page

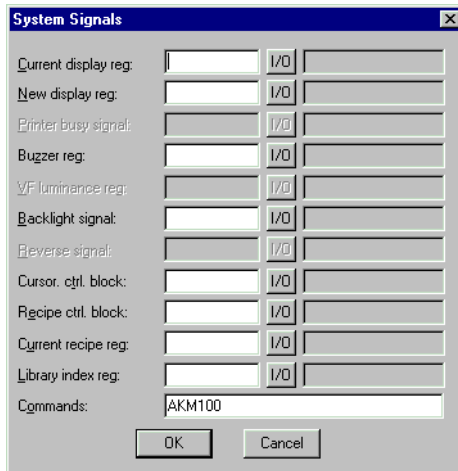
The terminal contains an information page. The information page is activated by simultaneously pressing the [←] and [PREV] keys in run mode. On the touch terminal a touch key can be used/configured to call up the information page.

Parameter	Description
<b>STARTS</b>	Number of times the terminal has been started.
<b>RUN</b>	Number of hours the terminal has been in operation.
<b>CFL</b>	Number of hours the backlighting has been switched on.
<b>DYNAMIC MEMORY</b>	Free RAM memory (working memory) in number of bytes.
<b>FLASH MEMORY</b>	Free Flash memory (project memory) in number of bytes.
<b>FLASH ERASE CNT</b>	Number of times each sector of the Flash memory has been erased. The current spare block always shows zero.
<b>FLASH CACHEHITS</b>	Percentage of Block/Allocation cachehits in the file system.
<b>FLASH ALLOCS</b>	Max percentage of used/active allocations per block in the file system.
<b>ALARMS</b>	Number of alarms and alarm groups in the application.
<b>TRENDS</b>	Number of trend IO and trend objects in the application.
<b>RECIPES</b>	Number of production methods in the application.
<b>DIGITAL IOs</b>	Number of digital IOs monitored continuously (static) and number in the current block (monitor).
<b>ANALOG IOs</b>	Number of analog IOs monitored continuously (static) and number in the current block (monitor)
<b>IO POLL</b>	Time in ms between two readings of the same IO.
<b>TIMEOUT</b>	Number of time-outs in the communication.
<b>CHKSUM</b>	Number of check sum errors in the communication.
<b>BYTEERR</b>	Number of byte errors in the communication.
<b>FRAME</b>	Number of framing errors in each port.
<b>OVERRUN</b>	Number of overrun errors in each port.
<b>PARITY</b>	Number of parity errors in each port. 1=RS-422 port and 2=RS-232 port.

### 3.5 “Joystick” function

Not valid for the E910.

With this function it is possible to use the arrow keys as function keys. On the command line under System signals you enter the command AK and an address. For example you can enter AKM100 (the command AK and memory cell M100).



M100 then is the enable signal and the following 4 memory cells are according to the following control block.

Memory cell	Description
Mn0	Enable. Arrow key detection. Disable=normal function.
Mn1	Left key
Mn2	Down key
Mn3	Up key
Mn4	Right key

If you press an arrow key when the enable signal is set, the memory cell corresponding to the key pressed will be set. When the enable signal is set the normal functions of the arrow keys are disabled.



## **Example**

The following example can be used to exit the joystick function and return to normal function. Make the following steps.

- create a block with 1:st man obj defined.
- enter the static text “JOYSTICK”.
- create a digital object with the following settings.

D.Signal: M100

Text 0 : OFF

Text 1 : ON

Maneuver: YES

The textblock will look like this.

JOYSTICK #—

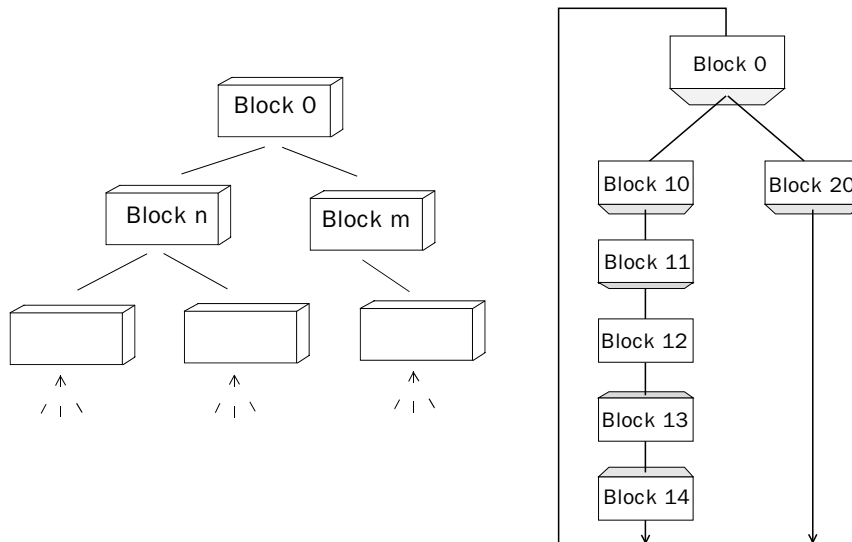
## 4 General

This section describes how the terminal is designed as well as the basics for using it. Furthermore there is an explanation of the general rules and object parameters as well as common functions which apply to the terminal.

### 4.1 Method for programming a project

Building up the application in the terminal graphically provides the operator with concise information on the operation of the system. It is important to organize your application well. Think about which functions are required. Begin at the introductory level and then work down through the levels of detail. When an object is programmed, base this on the functions in your application. Each function then becomes one or more blocks depending on how complex the function is. A project can contain both Graphic blocks and Textblocks and each block can contain static and dynamic objects. To achieve a structured application the block should be created in a hierarchy which makes working natural for the operator.

In the run-time mode it is possible to test all or parts of the application before running.



## 4.2 Efficient communication

To make the communication between the terminal and the PLC system quick and efficient, the following should be noted about how the signals are read and what that can be done to optimize the reading.

### Signals affecting the communication time

It is only signals to objects in the current block that are read continuously. Signals to objects in other blocks are not read, that is the number of blocks does not affect the communication time.

Besides the signals to objects in the current block, the terminal is continuously reading the following signals from the PLC system:

Display signals

Block print-out signals

LED registers

Alarm signals

Remote acknowledge signals on alarms and alarm groups

Login signal

Logout signal

Trend registers at the sample points

Bargraph registers if using min/max indicators

New display register

Buzzer register

Backlight signal

Cursor control block

Recipe control block

Library index register

Index registers

PLC clock register if the PLC clock is used in the terminal

List erase signal

No protocol control register

No protocol on signal

**Signals not affecting the communication time**

The following signals do not affect the communication time:

- Signals linked to function keys
- Time channels
- Objects in the alarm messages

**How to make the communication more efficient****Group PLC signals consecutively**

The signals from the PLC system are read most rapidly if all signals in the list above are consecutive. If, for example, 100 signals are defined, it is quickest to read these if they are linked to, for example, M0.0-M11.7. If the signals are spread out (e.g. I0.4, Q30.0, T45.3 etc.) the updating is slower.

**Efficient block changes**

Block changes are carried out most rapidly and efficiently through the block jump function on the function keys or through a jump object. "Display signals" in the block header should only be used when the PLC system is to force the presentation of another block. The "New Display" register can also be used if the PLC system is to change the block. This does not affect communication as much as a larger number of "Display signals".

**Using the terminal clock**

An extra load is put on communication if the clock of the PLC system is used since the clock register must be read up to the terminal. Downloading of the clock to the PLC system also creates an extra load. The interval between downloadings should therefore be as long as possible.

## Packaging of signals

When the signals are transferred between the terminal and the PLC system, all signals are not transferred simultaneously. Instead they are divided into packages with a number of signals in each package. To decrease the number of packages and make the communication faster this number has to be considered. The number of signals in each package depends on the driver used.

PLC system	Number of signals in each package	
	Analog signals	Digital signals
FX CPU	32	200
A CPU	116	800
QnA CPU	60	200
A C24	58	200
QnA C24	51	864

To make the communication as fast as possible the number of packages has to be minimized. Consecutive signals require a minimum of used packages but it is not always possible to have consecutive signals. In such cases the so-called waste between two signals has to be considered. The waste is the maximum distance between two signals you can have and still keep them in the same package. The waste depends on the driver used.

PLC system	Waste	
	Analog signals	Digital signals
FX CPU	20	100
A CPU	20	100
QnA CPU	20	40
A C24	20	100
QnA C24	54	864

Signal	1	2	3	4	5	6	7	8	9	10
Used	X	X					X	X	X	

Waste

## Operator's pictures

### Use graphic blocks for operator communication

The textblock is designed in the first place for report print-outs. They are slower and demand more memory than graphic blocks.

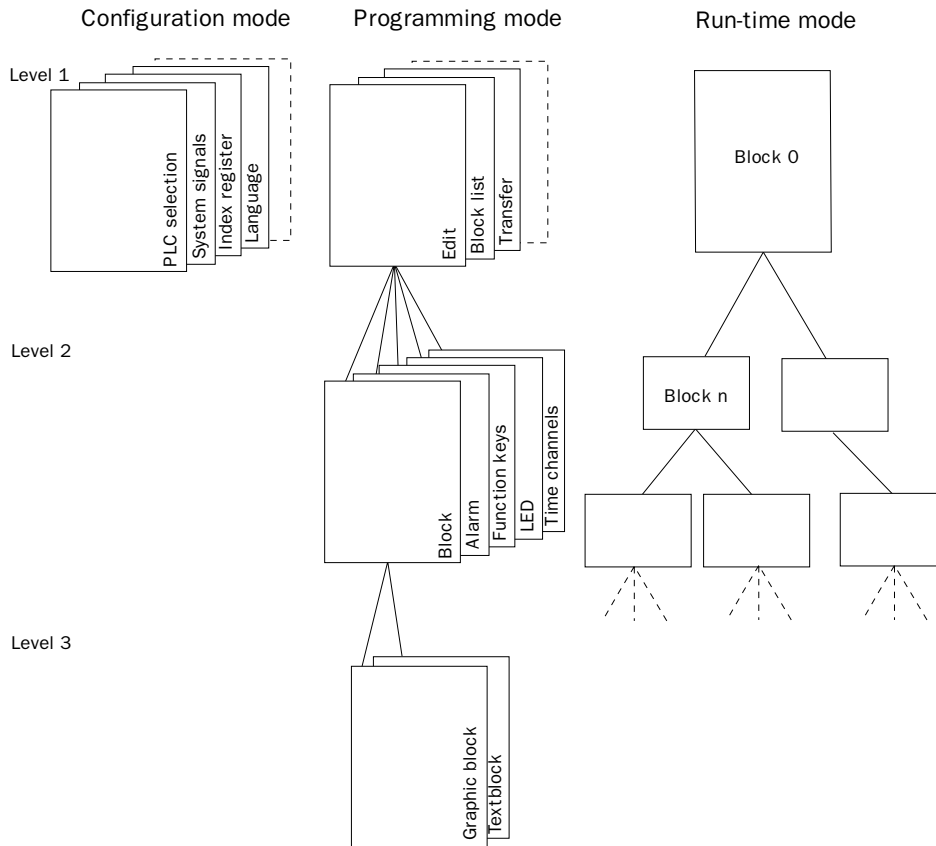
### Use the 3-D effects for elegant operator pictures

Combinations of objects with frames and 3-D rectangles can be very effective. In this context the "lighting" appears to come from the top left-hand corner, that is the "shadowing" is on the under and right-hand edges of raised objects and on the upper right-hand edges of inset objects.

## 4.3 Menu structure

The terminal is divided into three modes: *configuration*, *programming* and *run-time mode*. Normally configuration and programming are carried out with the PC software. Each mode consists of a different number of menu levels depending on the function. Each level consists of a menu where you make a choice or state a parameter to go on to the next level (menu).

The application is built up of *blocks*, *Graphic blocks* and/or *Textblocks* (primarily for report print-outs). Values from the PLC system are shown and altered in the blocks. Each block has a block number between 0 and 989 allocated by the program. Blocks 990-999 are reserved for special purposes, so-called *system blocks*. The terminal is object-oriented which means that a block can contain all the signals linked to an object for controlling and monitoring, a pump for example.



## 4.4 Blocks

A block header is defined for each block. The block number, type of block, status word etc. are contained in the block header. The alarm, time channels and contrast settings functions can also be called up as blocks. These are called system blocks.

A maximum of 990 blocks can be defined in a project.

---

### Note!

The block type cannot be altered for a defined block.

---

## 4.5 Signal types i MELSEC

The PLC system's signal types are allocated according to the table below. See the PLC system user manual for the complete address area.

### Digital signals (ON/OFF)

The terminal can handle digital signals of the following types: For addressing other PLC systems refer to the manual for the drivers.

Signal	Function	A series	FX series	QnA series
<b>X</b>	Input Relay	•	•	•
<b>Y</b>	Output Relay	•	•	•
<b>M</b>	Internal Relay	•	•	•
<b>Dn.x</b>	Bit addressing in Data Registers. Bit x in Data Register n. Hexadecimal addressing of the bits, 0-F.			•
<b>L</b>	Latch Relay	•		•
<b>B</b>	Link Relay	•		•
<b>F</b>	Error Relay	•		•
<b>SB</b>	Link Special Relay			•
<b>V</b>	Edge Relay			•
<b>S</b>	Step Relay		•	•
<b>T</b>	Timer	•	•	•
<b>ST</b>	Accumulated Timer			•
<b>C</b>	Counter	•	•	•
<b>SM</b>	Special Internal Relay			•



## Analog signals (Values)

The terminal can handle analog signals of the following types.  
For addressing other PLC systems refer to the manual for the drivers.

Signal	Function	A series	FX series	QnA series
<b>T</b>	Timer	•	•	•
<b>ST</b>	Accumulated Timer			•
<b>C</b>	Counter	•	•	•
<b>D</b>	Data Register	•	•	•
<b>SD</b>	Special Register			•
<b>R</b>	File Register (Bank addressing)	•		•
<b>W</b>	Link Register	•		•
<b>SW</b>	Link Special Register			•
<b>Z</b>	Index Register			•
<b>ZR</b>	File register (Continuous numbering)			•

---

### Note!

32-bit counters (C200-C255) in FX systems cannot be monitored.

---

## 4.6 Signal format

The following signal formats are available in the dialog for each object.

Format type	Range
Signed 16-bit	-32768 – +32767
Unsigned 32-bit	0 – +65535
Signed 32-bit	-2147483648 – +2147483647
Unsigned 32-bit	0 – +4294967295
Float with exponent, 32-bit	±3,4E38 Numbers larger than 1000000 are shown with exponent.
Float without exponent, 32-bit	The Positions (including the decimal point and sign) and Decimals state the valid range. E.g. 8 positions and 3 decimals lead to ±999,999.
BCD Float, 32-bit	0-9999,9999. Only MELSEC AnA PLC system.
BCD 16-bit	0-9999
BCD 32-bit	0-99999999
HEX 16-bit	0-FFFF
HEX 32-bit	0-FFFF FFFF



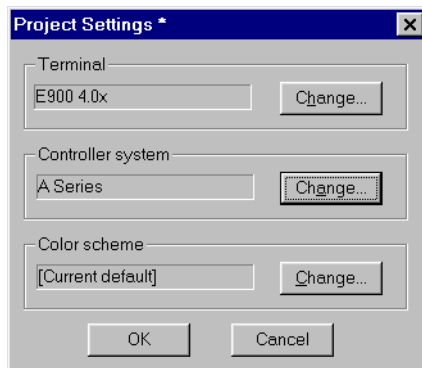
## 5 Programming using the PC software

This section describes how the terminal is programmed using the PC software MAC Programmer+. All the functions can be reached with the PC software from the menu:

File Edit View Functions Setup Block Manager Transfer Window Help

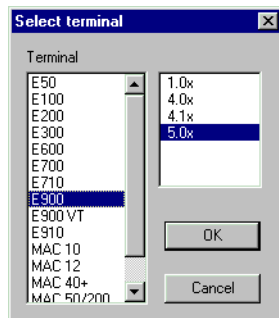
### 5.1 Creating a project

A new project is created by selecting **New** in the **File** menu. The **Project Settings** dialog window is now shown. Here you can select the PLC system, type of terminal and color scheme.



#### Terminal:

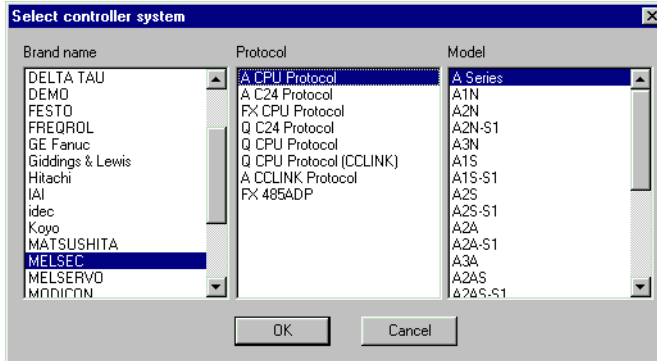
Press **Change...**



Select terminal type and version (system program) for current terminal type.

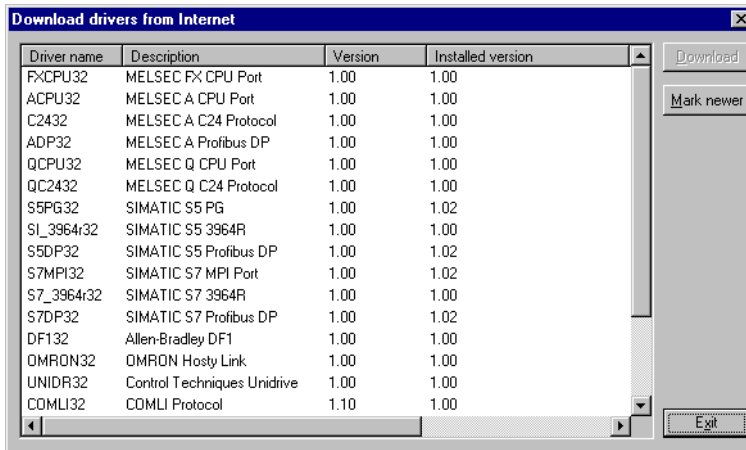
### Controller system:

Select here the controller system to which the terminal is to be connected. When you press the **Change...** button the following dialog is shown. The list shows installed drivers. Here you select brand name, protocol and controller model.



### Update drivers from Internet

To update available drivers to the latest version or to install new drivers from the Internet you use the **Update terminal drivers from Internet** function in the **File** menu. All project must be closed before this function is used. The computer must be able to make an Internet connection. You do not need a browser. When the connection is established a list shows all drivers that can be downloaded from the Internet to the computer.



The list shows the version number of available drivers and the version number of installed drivers. Mark the driver/drivers you want to install in the MAC Programmer+. The function **Mark Newer** will mark all drivers that are available in a newer version than the one installed and the drivers not installed. You then select **Download**. Each driver is approximately 500 kb and it is ready to use when the download is completed.

**Update drivers from disk**

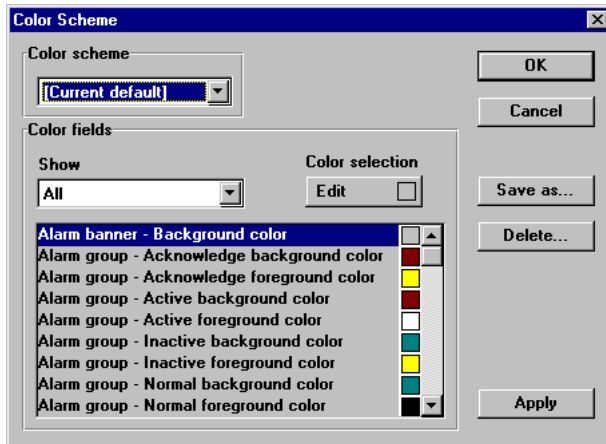
To update available drivers to the latest version or to install new drivers from disk, use the **Update terminal drivers from Disk** function in the **File** menu. All projects must be closed before this function is used. Open the mpd file in the map where you find the drivers. Now a list shows available drivers is displayed.

The list shows the version number of available drivers and the version number of installed drivers. Mark the driver/drivers you want to install in the MAC Programmer+. The function **Mark Newer** will mark all drivers that are available in a newer version than the one installed and the drivers not installed. You then select **Download**. Each driver is approximately 500 kb and it is ready to use when the download is completed.

**Color scheme:**

In terminals supporting colors, you can create your own color scheme and store it under a unique name. The color scheme decides the default color of the background, menus, dialogues, objects etc. When you select an object in the toolbox or in the menu the colors of the object are shown as defined in the color scheme.

When you select **Change** the following dialog is displayed. Here you change the current color scheme or create a new one.



The **Apply** button update all colors in the project, except lines, circles, rectangles and arcs.

## 5.2 Changing project settings

Both the selected terminal and the controller system can be changed in the project. Choose **Project settings** in the **File** menu and press **Change** at the parameter Terminal and/or controller system.

If you update the system program in the terminal to a newer version you must change to the correct terminal version in the Project settings dialog. If this is not done the new functionality in the new terminal version cannot be used.

If you change the controller system in the project to a system with other names of the devices used, these must be changed. It is easily done by using the internal name list, see the section Name list.

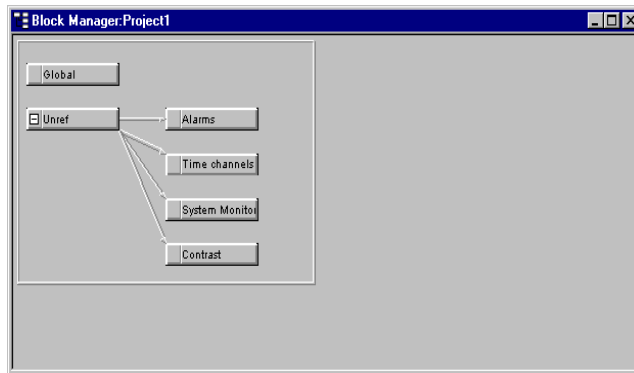
1. Choose **Name list** in the **View** menu.
2. Select **Undefined** to insert all I/Os used in the project to the name list.
3. Select **Export** to export the name list to a text file. State a name and press **Save**. Define how the text file will be separated.
4. Pick up the text file in a text editor, e.g. Wordpad.
5. Change all the I/Os to devices used in the new controller system and save the file as a text file.

6. Select **Import** in the Name list dialog and press **No** on the question if invalid I/Os should be removed.
7. Select **Rebind** to update all new I/Os in the project with the new device names.
8. Choose **Project settings** in the **File** menu and press **Change**.
9. Select the new controller system you want to change to and press **OK** twice.

All I/Os in the projects have now the correct devices and the change of PLC system is completed.

### 5.3 Creating a block via the Block Manager

The block manager is shown when you have created a project. Which blocks are included in an application are shown here.



In the block manager there is a toolbox with the following functions.



Mark block and create block.



Add a new block.



Define block header for selected block.



Make a block jump to the block to be created with the cursor.





Make function key block jump to the block to be created with the cursor.



Delete selected block.



Edit selected block.



Block manager settings.



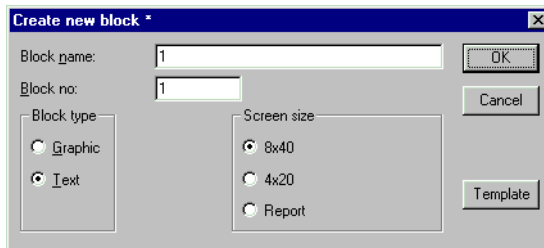
Zoom in.



Zoom out.

## Define block

When you add a block in the Block Manager the following dialog is shown. It is a simplification of the complete block header.



### Block name:

A name for the block can be written here. The block name is shown in the block manager and in the block list.

### Block no:

The number of the block. The defined values are entered automatically if a block with assigned number already exists. The block number 0 is shown automatically on starting up and must always be used in a project.

### Block type:

Select block type, graphic or text.

**Screen size:**

Select character size for textblock. The character size can not be changed for a defined object.

**Template:**

Here you can copy a template to the block or save the block as a template.

**Define the complete block header**

Basic parameters valid for each single block are defined in the block header. Select **Header** in the **Block Manager** menu. The following parameters can be defined in the block header:

**Block no:**

The number of the block. The defined values are entered automatically if a block with assigned number already exists. The block number 0 is shown automatically on starting up and must always be used in a project.

**Block name:**

A name for the block can be written here. Block name is optional.

**Display signal:**

A digital signal which, when activated, results in the block being shown on the display. If display signals are used they should be ordered consecutively, for the different blocks to get the best performance. This field is not filled in if another method for changing blocks is used.

**Print signal:**

A digital signal which when activated results in the block being printed out on a connected printer. The display signal and printer signal can be the same. The printer signals should be used consecutively to printout as quickly as possible.

**Completion signal:**

A digital signal which is activated when the print out is completed. Normally the signal is set. If the function Reset in Block Header is marked then the signal is set to zero when the printout is completed.

---

**Note!**

Only textblock can be sent as email message.

---

**Send mail signal:**

The message is sent when the stated digital signal is set.

**Mail completion signal:**

A digital signal that is activated by the terminal when the message is sent. Normally the signal is set. If you mark the function Reset the signal will be reset when the message is sent.

**Mail to address:**

Enter the email address to the receiver. Select up to eight receivers from the list shown when you click on the ... button. The list of email addresses is defined in the **SMTP Client Service** dialog under **Network Services** in the **Setup** menu.

**Attach file:**

Enter the name of a trend or recipe file to be attached to the message. If there is a trend file and a recipe file with the same name the trend file is sent.

**Security level:**

A security level (0-8) for the block is stated here. The operator must enter a password equivalent to a given or higher security level if a security level > 0 is stated.

**Background block no:**

Here you have the possibility to load another block as a background in current block. For example, if you want to use the same background in more than one block. When the graphic block editor is active you can select if the background block shall be shown in the current block with the function **Show Background Block** in the **Windows** menu.

**Cursor color:**

Select the color of the cursor in graphic blocks.

**Cursor thickness:**

Select between three different alternatives.

**Block type:**

Select block type, graphic or text. It is not possible to change block type for a defined block.

**Screen size:**

Select the character size in the textblock. It is not possible to alter the type of a defined block.

**Fkeys:**

The local function keys (F1-F22) for the block are defined here. Enter a signal that will be activated when the function key is pressed. See the chapter on "Function keys".

**Template:**

Here you can copy a template to the graphic block or save the block as a template.

**Status:**

States the display's appearance in the run-time mode. The status word has no function in the system blocks. The parameters in the status word have the following functions.

**Cursor off:** States if the cursor will not be visible in the block in the run-time mode.

**Put cursor at first man. object:** States if the cursor will be on the first maneuverable object instead of in the top left-hand corner.

**Disable main menu key:** States if the [MAIN] key is to be blocked in the run-time mode when the block is shown on the display.

**Disable alarm list key:** States if the [LIST] key is to be blocked in the run-time mode when the block is shown on the display.

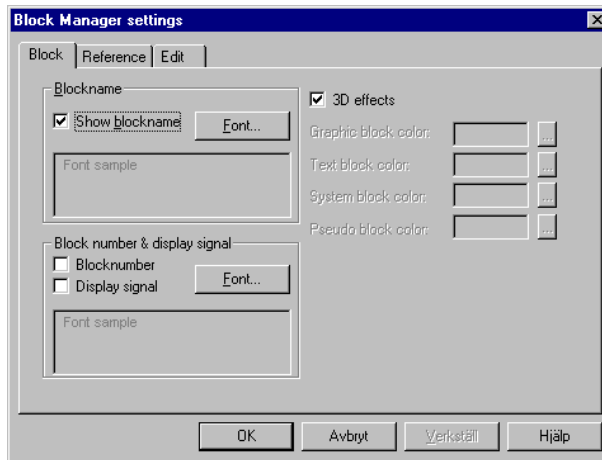
**MORE indication:** States if the + sign is to be shown in the bottom right-hand corner and the top right-hand corner when there is more of the block below and above respectively than that shown on the display.

**Auto data entry:** States if the cursor will automatically move to the next maneuverable object after an entry. The cursor can only be placed on maneuverable objects in this state.

**Disable Prev Key:** States if the [PREV] key and the "Return to previous block" function for the function keys is to be blocked in the run-time mode when the block is shown on the display.

## Settings for the Block Manager

Here you make the settings for the presentation in the Block Manager.



### Block:

Select how data should be presented in the block.

### Reference:

Select the appearance of the Block Manager.

### Edit:

Select special functions for the presentation in the Block Manager.

## Show terminal around the working area

To graphic blocks there is a function in the **Window** menu called **Show terminal**. When this function is active a picture of the current terminal is drawn around the working area. The picture of the terminal has pressure sensitive surfaces on function keys, LEDs and lead text fields.

### Define function keys

When you double click on a function key you can select to define a local or global function key. The editor for selected function is displayed.

### **Define LEDs**

When you double click on a LED the editor for defining LEDs is displayed.

### **Create insertable strip**

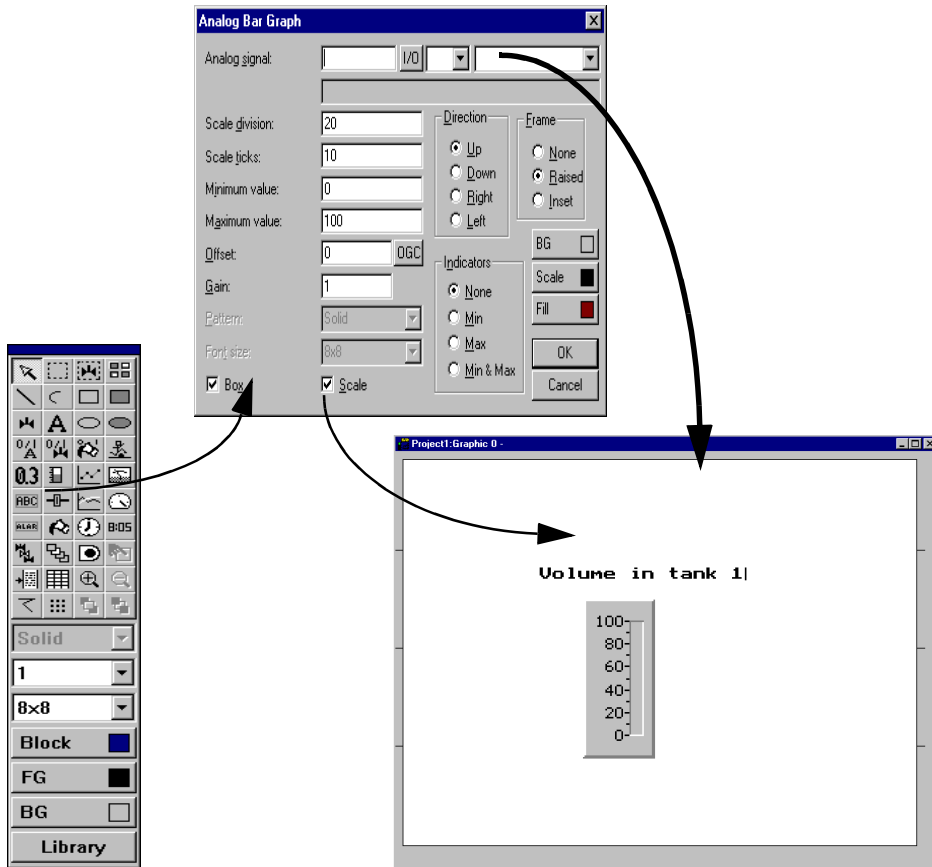
When you double click on a field on the insertable strip, a dialog is displayed where you can enter a text, select adjustment of the text and select font. This function makes it possible to create the whole insertable strip and print on plastic film and then use it in the terminal.

## **5.4 Programming blocks**

Double click on current block or select Edit for marked block. The working area for the block and the tool box are now shown. The objects that can be created in the block are in the tool box.

Select an object by clicking on the object in the tool box, and place the cursor in the working area where you want the object to end up. The dialog window for selected objects is shown when you click. The object is now shown on the working area. If you have selected static text or graphics, it is written directly on the working area.

General parameters for the objects are described in the chapter "General". Each type of object is explained in the chapters "Graphic presentation and maneuvering" and "Text-based presentation and maneuvering".



## 5.5 The graphic block editor

This section describes the graphic block editor in the MAC Programmer+. The editor is based on Windows standards.

In the graphic block editor you create graphic blocks with static and dynamic graphics.

### How to open the graphic block editor

The graphic block editor is opened when you double click on the current block in the Block Manager or from the block list.



## Mouse, keys and the cursor

The following section describes how to use mouse and keys in the graphic block editor. Furthermore we describe the different shapes of the cursor.

### Mouse

Use the floating toolbox to create an object.

Click on an object to mark it.

Mark several objects by drawing a frame around the objects or by holding the Shift key down when you click on the objects.

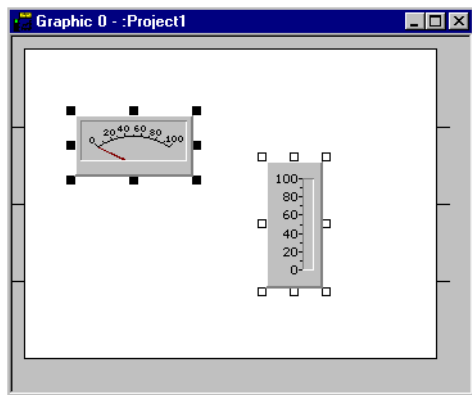
Keep the mouse button down on object and drag the mouse to move it.

Keep the mouse button down on object and press the Ctrl key and drag the mouse to copy it.

Keep the mouse button down on object handle and drag the mouse to size it.

Double click on object to view parameters.

The following picture shows marked objects.



**Keys**

Use Object menu to create an object.

Use arrow keys to move cursor.

Use [Ctrl] + [arrow keys] to move on pixel level.

Press [SPACE] on object to mark it.

Mark several objects by selecting Block Select in the Object menu and drag a frame around the objects using [SPACE] and arrow keys.

Keep [SPACE] down on object and use arrow keys to move it.

Keep [SPACE] down on object handle and use arrow keys to size it.

Press [ENTER] to view parameters on marked object.

**Cursor**

The cursor has four different shapes



within an object



change size of an object



in graphic working area



selection in menu or toolbox

**The toolbox**

In the editor you use the toolbox to create objects in the block. All functions are also available in the menus. The following table describes the functions in the toolbox. See the chapter “Graphic presentation and maneuvering” for the different objects that can be used in graphic blocks.



Selection pointer mode



Area selection mode



Symbol creation mode



Zoom in



Zoom out



Grid



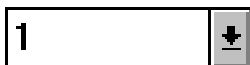
Put on top



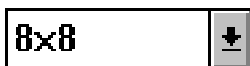
Put to background



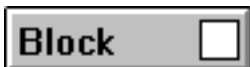
Line pattern



Line thickness



Character size



Background color for the block



Foreground color for the object



Background color for the object



Import/export symbols.

## Create object

Select the type of object in the toolbox and drag the object. Static graphics will be displayed when you click in the editor. Dynamic objects are defined in a dialog box before they are shown on the display.

When an object is drawn it is marked with handles and the function selection pointer mode is activated.

## Recommendations

When the graphic block editor is active there is a command in the **Window** menu to turn **Draft Move & Size** on/off. It is recommended to turn this option on when using a slow computer. This turns off the immediate updating of the sized/moved object. A check mark appears next to the menu item when Draft Move/Size is on.

In the **File** menu you will find the **3D Controls** function. This functions means that the forms in the program will be displayed with 3D effects. It is recommended to turn this function off when using a slow computer.

## Static graphics

The static graphic objects are line, arc, ellipse, rectangle, polygon, symbol, text and decorations and are used to create background graphics.

## Dynamic objects

Select the type of object in the toolbox and place it in the block. Before the object can be shown on the display you must define the object in the displayed dialog. See the chapter "Graphic presentation and maneuvering" for further details about object settings.

## Select several objects

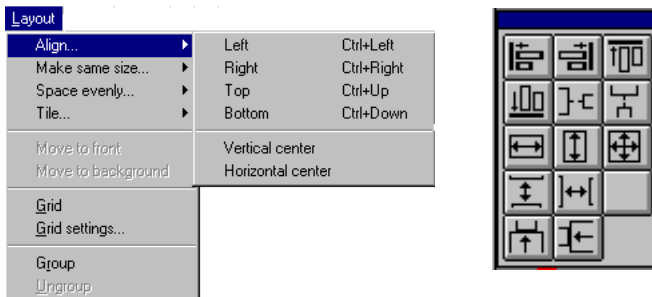
In the graphic block editor you can select several objects in two different ways.

- Click outside the object, keep the left mouse key pressed and draw a frame around the objects you want to select. The latest created object will be shown with filled handles.
- Select the selection cursor in the toolbox. Keep the [Ctrl] key pressed while you select the objects. The latest selected object will be shown with filled handles.

## Place object

To easily adjust the placing of the objects in graphic blocks there are a number of functions in the Layout menu. Under this menu you will find the functions **Align**, **Make same size**, **Space evenly** and **Tile**. The functions are also available in a separate toolbox.

To use the functions two or more objects must be selected. The functions calculate the placing from one or two reference object. For the functions Align, Make same size and Tile it is the last selected/created object that is the reference object, see the section "Select several objects". For the Space evenly function it is the uppermost and the lowermost objects that are the reference objects. The reference objects are not affected by the function.



## Align

Under Align you have six alternatives for adjusting the objects horizontally and vertically.

### Left

This left-aligns the marked objects with the reference object.

### Right

This right-aligns the marked objects with the reference object.

### Top

This top-aligns the marked objects with the reference object.

### Bottom

This bottom-aligns the marked objects with the reference object.

### **Make same size**

Under Make same size there are three alternatives for making the marked objects the same size.

#### **Width**

Adjust the marked objects so that their width is the same as the reference object.

#### **Height**

Adjust the marked objects so that their height is the same as the reference object.

#### **Both**

Adjust the marked objects to exactly the same size as the reference object.

### **Space evenly**

Under Space evenly there are two alternatives for adjusting the space between marked objects

#### **Vertical**

Adjusts marked objects with an even vertical spacing. The uppermost and lowermost objects retain their positions. At least three objects must be marked.

#### **Horizontal**

Adjusts marked objects with a horizontal spacing. The leftmost and rightmost objects retain their positions. At least three objects must be marked.

### **Tile**

Under Tile there are two alternatives for adjusting objects so that they are edge-to-edge.

#### **Vertical**

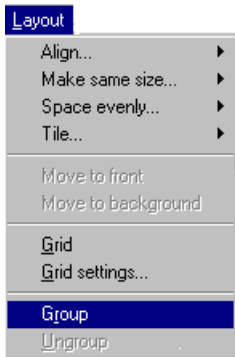
Adjusts marked objects so that they align vertically edge-to-edge with the reference object.

#### **Horizontal**

Adjusts marked objects so that they align horizontally edge-to-edge with the reference object.

## How to group objects

In the Layout menu you can select to group several objects. The grouped objects are handled as an object that can be resized. Color and font can be changed individually for the object in the group object. When you click on an object in the group object the dialog for the object is displayed. Mark the objects you want to group and select the function **Group** in the **Layout** menu.



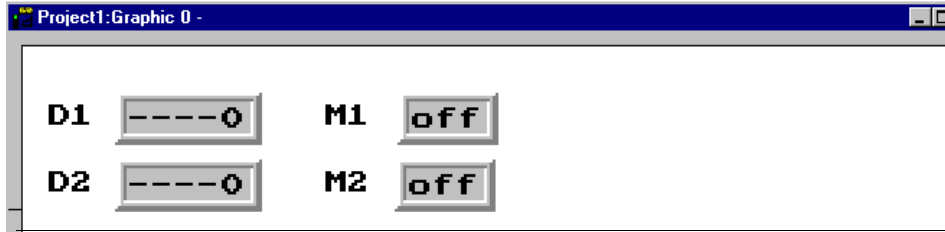
With the **Ungroup** function in the **Layout** menu you release the group object.

## Save and load group object

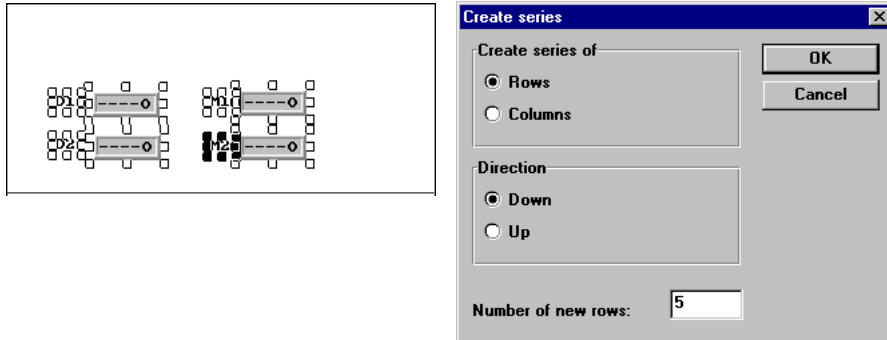
If you mark a grouped object and press the **Library** button in the toolbox in the graphic block editor you can save the object. When you press the button library without any object being marked you can load grouped objects stored on the computer. MAC Programmer+ is delivered with several grouped objects. The objects are stored in c:\Program\Beijers\MAC Programmer+\lib\library\.

## Create series

In the graphic block editor you can create series of objects. First create two rows and/or columns with the same object.

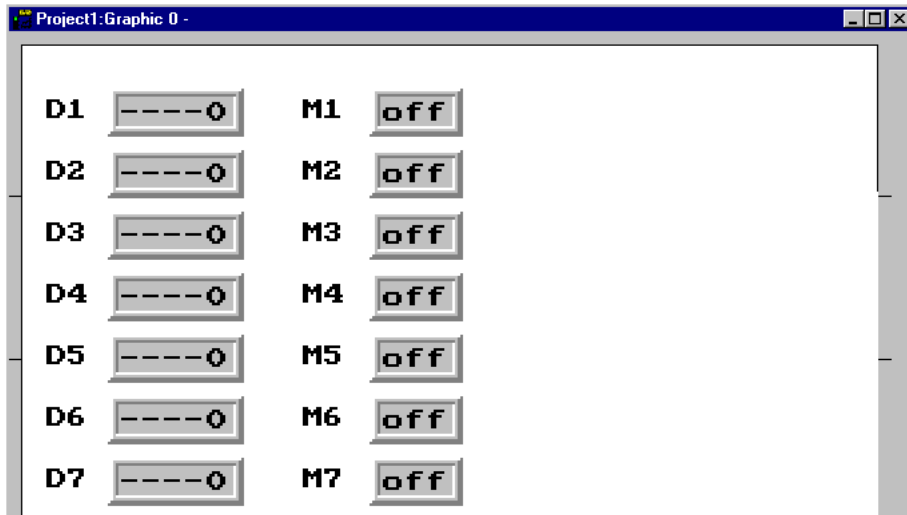


Thereafter mark the objects and select the **Create series** function in the **Object** menu.



A dialog is now displayed where you select to create rows or columns, the number of rows/columns and in which direction the series shall expand. MAC Programmer+ now creates a table with the assigned number of rows or columns.





---

**Note!**

To be able to create the table the last character in the lead text must be a figure.  
The object Alarm banner is not allowed in a table.

---

## Symbols

You can create symbols in three ways, using the Symbol manager, with the functions Create symbol or copy graphics from other Windows programs using the clipboard. See separate chapter how to use the Symbol manager

### The function Create symbol

- Select the function Create symbol.
- Drag a marking frame around the graphic you want to save as a symbol.
- State the name of the symbol, max 8 characters.

The symbol is now stored in the symbol directory with the stated name.

### **Copy graphics from other programs**

- Use the copy function in another program (e.g. Paint).
- Make graphic block active in the MAC Programmer+ and select Paste.
- State the name of the symbol, max 8 characters.

The symbol is now stored in the symbol directory with the stated name.

With the functions copy and paste in the MAC Programmer+ it is possible to copy graphics and symbols between blocks and between projects.

---

#### **Note!**

When you copy user-defined symbols between projects, the symbol itself will only be copied if the symbol does not already exist in the target project.

---

## **5.6 The textblock editor**

This chapter describes the textblock editor in the MAC Programmer+. The editor is based on Windows standards.

In textblock editor you build up dialogs and reports. A textblock can consist of static text and dynamic objects. Static text is text which is not changed when executing the program. Dynamic objects are objects that are linked to the signals in the PLC system.

### **How to open the editor**

The text editor is opened by double clicking on the current block in the Block Manager or from the block list. In the block list you select a defined textblock or create a new textblock.

### **Mouse and keys**

Click at the beginning of the text you want to mark and drag the mouse pointer over the text. If you are using the keyboard press on shift and arrow keys.

The function Cut will erase marked text.

The character End of line is stated when you press the key combination [Ctrl] [Enter].

Double click or press [F4] on object to view parameters.

## The toolbox

In the textblock editor there is a toolbox with the following functions. All functions are also available in the menus. See the chapter “Text-based presentation and maneuvering” for the different objects that can be used in textblocks.



Zoom in



Zoom out



ASCII choice list. Selection of characters not available on the keyboard

## Define textblock

### Static text

The textblock editor is a text editor in which you type the static text. You can also use the Windows functions copy/paste in or between blocks, or from other Windows programs e.g. Microsoft Word. This makes it easy to create documentation of your application.

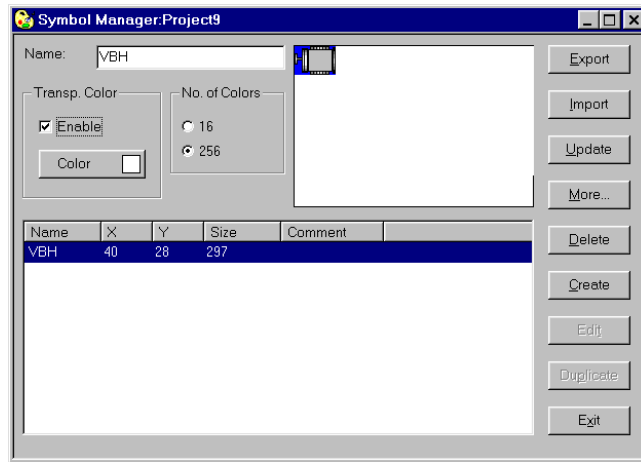
### Dynamic object

You can define a dynamic object anywhere in the text. Select the type of object in the toolbox or in the object menu and then you define the object in the dialog displayed.

The dynamic objects are signed with the character # followed by one or several '-' depending on the number of positions in the object. For further details about definition of dynamic objects, see the chapter “Text-based presentation and maneuvering”.

## 5.7 The symbol manager

In the symbol manager you have the possibility of importing and exporting symbols. You can also create and delete user-defined symbols from the symbol directory. In the symbol list you can see the user-defined symbols. The predefined symbols are not shown, since they are not changeable. How to create a symbol is described in the section “Symbols” in this chapter.



### Export symbols

Symbols can be exported in BMP format to be used in other applications.

### Import symbols

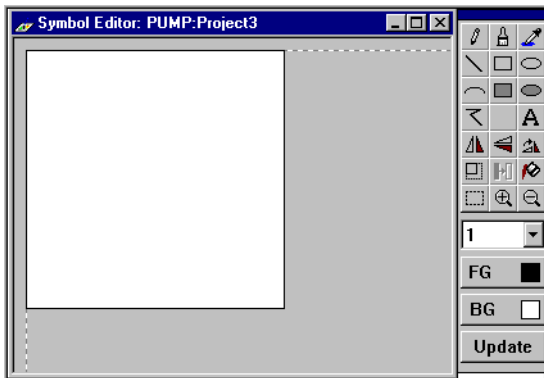
The import function makes it possible to reuse symbols from other applications.

Picture files in bmp, cmp, dex, fpx, jpg, mpt, pcd, png, tga, tif and pcx formats can be imported into the symbol directory from other Windows applications e.g. Paint.

MAC Programmer+ is delivered with several symbol libraries containing different symbols, e.g. pump symbols. The symbols are stored in c:\Program\Beijers\MAC Programmer+\lib\bitmap\.

## Create

The Create function is used to create new symbols. When you select Create, you first have to enter a name of the symbol. When you click on OK the bitmap editor opens. The bitmap editor works as a normal drawing program with the limitation according to current terminal.



## Delete

When you press the right mouse button and draw with the pen, you draw with the background color. If the background color is white it looks like you delete.

## Edit

The function Edit activates the bitmap editor for a defined symbol.

## Duplicate

This function is used to create a copy of current symbol with a different name.

## The More function

The More function is used to add more information to the symbol:

**File name:** States the file name if the symbol was imported from a file.

**Date created:** States the date the symbol was created.

**Source:** States the source of the symbol.

**None:** Unknown source

**Bitmap file:** Imported from bitmap file

**Clipboard:** Imported from clipboard  
(Copy & Paste function)

**Graph block:** Created in a graphic block

**Comment:** Here you add your own comment to the symbol.

## The Delete function

The Delete function is used to delete a symbol from the project.

## The Transparent color function

When you import a symbol you can select one color in the symbol to be transparent.

## The function No. of Colors

In the E900 and E910 supporting colors you can select to handle symbols in 16 or 256 colors.

## 5.8 I/O change

With the I/O change function you can change a single I/O or move a whole I/O area. You can change I/O in the whole project or only in the object that was marked when you selected the function.

You can use the function on blocks in the block list, objects in the graphic block and textblock editor, lines in the alarm list, lines in the function key editor, lines in the LED editor or lines in the cross reference list.

Select the **I/O change** function in the **Edit** menu.

### Change I/O in

Select if you want to change I/O in the whole project or for selected objects.

### Change type

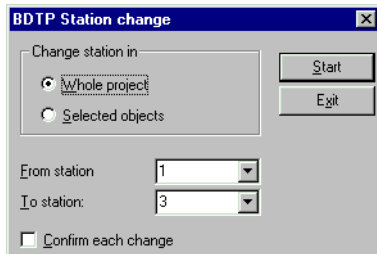
Select if an I/O shall change place or if you want to move a whole I/O area.

### From I/O, End I/O, To I/O

State the I/O to be changed and what it should be changed to or which I/O area you want to move.

## 5.9 Change BDTP station

With this function you can change the index numbering in a BDTP client project in a BDTP network. For example from station 1 to station 3. Select the **BDTP Station Change** function in the **File** menu.



### Change station in

Select if you want to change the index numbering in the whole project or for selected objects.

### From station, To station

Enter the index number to be changed and to which BDTP station the index number is to be changed.

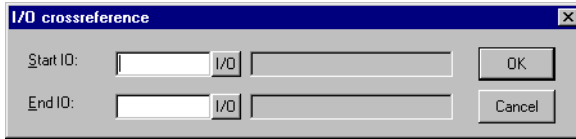
### Confirm each change

Select if you want to confirm the change of BDTP station for each object.



## 5.10 I/O cross reference

The cross reference function is used to document I/O in a clear way. Select the **I/O Cross reference** function in the **View** menu.

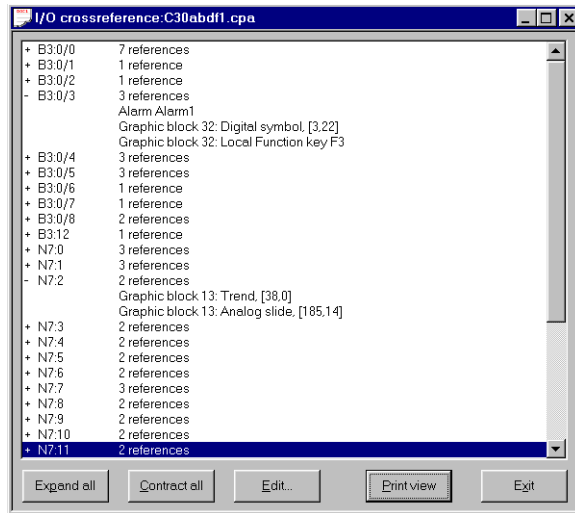


In the dialog displayed you state the Start I/O and End I/O. If you leave the Start I/O field empty, all I/O up to the value in the End I/O field will be included. If you leave the End I/O field empty, all I/O from the value in the Start I/O field will be included. If you leave both the fields empty all I/O in the list will be included.

### Presentation

The result of the function will be presented in a list with two levels. The first level shows used I/Os and how many objects that refer to respective I/O.

You reach the second level when you double click on the + character to the left of the I/O. Now the objects containing the I/O you clicked on are displayed. The + character will become a - character. The Print View button opens the print dialog and you can select to print the cross reference list as it looks on the display to a connected printer.



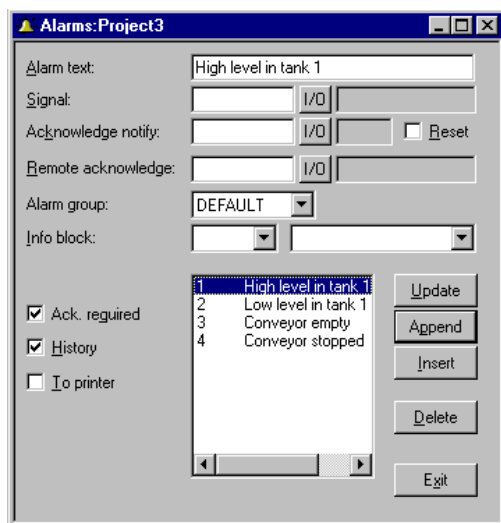
One or several lines in the list can be copied and pasted into e.g. a Microsoft Word document.

## 5.11 Other editors

The MAC Programmer+ also has editors for handling Function keys, LEDs, Alarms, Alarm groups, Passwords and Time channels. These editors are activated from the **Functions** menu. The editors work in a similar way. The parameters are described in separate chapters in this manual.

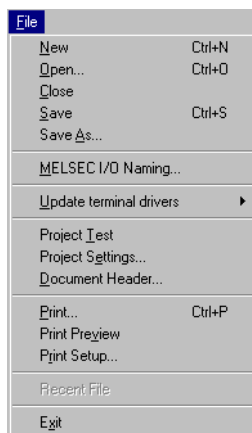
Defined Function keys, LEDs, Alarms, Alarm groups and Time channels are displayed in a list in the editors respectively. Add a new definition with **Append** or **Insert**. If you want to change a definition, mark the alarm, make the changes and press **Update**. If you want to update or append more than one definition to the list, press Update or Append once and then press Enter. Thereafter Update or Append remain operational until changed. Erase a marked definition with **Delete** or **Clear**. End the editor with **Exit**. The following example describes the Alarms editor.

The alarm is numbered automatically. When you press Append the alarm definitions are listed last, and when you press Insert the new definition will always be inserted before the marked definition in the list. When you press Update you confirm changes made on a definition.



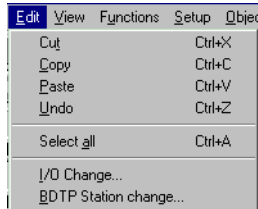
## 5.12 The File Menu

There are functions for creating, opening, saving and closing projects in the **File** menu. Printer settings, previewing print out, creating documentation headers and activating print out are also selected from here. Furthermore there are functions for testing a project and altering project settings. With the function **MELSEC I/O Naming** you select if the internal name list should be used or if a name file from MELSEC MEDOC or MELSEC MEDOC *plus* should be imported.



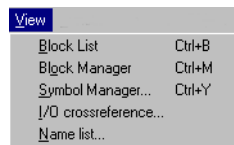
## 5.13 The Edit Menu

Functions for cutting, copying, pasting, undo and select all are found in the **Edit** menu.



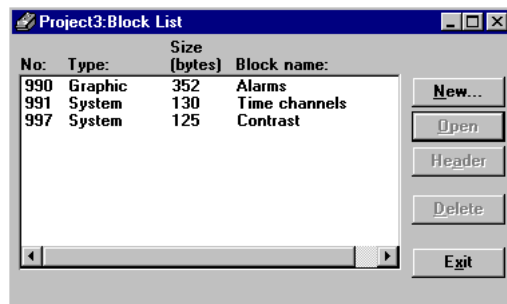
## 5.14 The View Menu

Function editor for block editing, symbol handling, cross reference and name list are to be found in the **View** menu.



## Block List

Blocks included in the application are shown in the block list. In the block list you create new blocks by pressing the **New** button. When you select New the block header is displayed. Basic parameters for the block are defined in the block header.



## **Block Manager**

Blocks included in the application are shown in the Block Manager graphically. In the Block Manager you create new blocks, define the block header, define block jump with the functions in the toolbox.

## **Symbol Manager**

Activates the symbol manager. In the symbol manager you create a library of symbols in BMP format. The symbols in the symbol manager are then available in the symbol list when you create static and dynamic symbol objects. It is also possible to duplicate and edit defined symbols.

## **I/O Cross reference**

Under I/O Cross reference you document I/O in a clear way. See the section “I/O cross reference” in this chapter.

## **Name list**

Under **Name list** you define a local name list for devices used. Devices used in the project without defined name can be inserted in the name list with the function Undefined. New devices can be added and existing names can be edited and updated. With the Rebind function you update the project with the changes done in the name list.

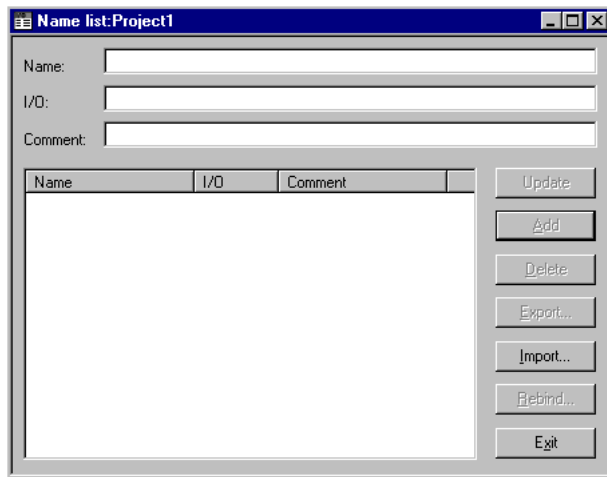
A name list can be exported to a text file and a text file can be imported to a name list. The items in the file are either separated with the character →, ; or ,. An internal name list can be sorted. The text file may not include national characters.

---

### **Note!**

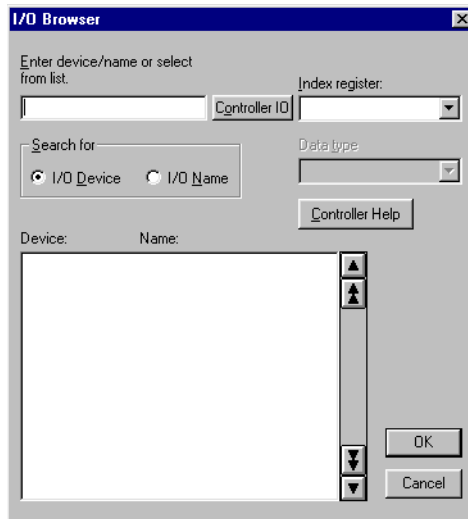
In the MELSEC I/O Naming function in the File menu you must choose Internal name list to use the name list.

---



If you have created a local name list in your project you can choose I/O signals from this name list via the I/O button. Click on the I/O button in all instances where you want to enter an address. The I/O browser has incremental searching. Incremental searching means that searching begins immediately while you are still entering characters in the name/device.

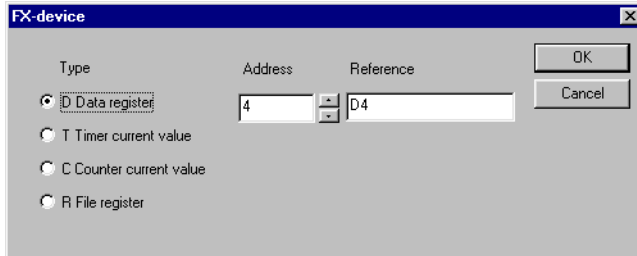
The I/O list is stated in device order or name order.



If you have linked a driver specific name file to your project you can choose I/O signals from this name file via the I/O button.

## Controller I/O

The button Controller I/O in the I/O browser can give you help about the addressing for different PLC-types.



---

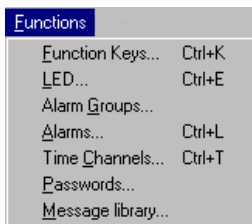
### Note!

The Controller I/O function is not supported by all drivers.

---

## 5.15 The Functions Menu

Function editor for function keys, LED, alarm, time channels, passwords and message libraries are to be found in the **Function** menu.



### Function keys

Under Function keys you define global and local function keys. See the chapter "Function keys".

### LED

Under LED you define the function for the LEDs. See the chapter "LEDs".

## Alarm Group

Under Alarm Group you can group alarms; for example, by danger level - so that they can be more realistically identified and dealt with. See the chapter “Alarm handling”.

## Alarms

Under Alarms you define alarm messages and the signal to be connected which will activate the alarm. See the chapter “Alarm handling”.

## Time Channels

Under Time Channels you define the time channels used to control events in the process at specified times. See the chapter “Time control”.

## Password

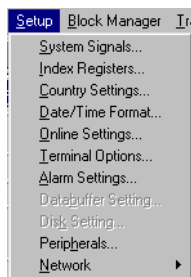
Under Password the password is defined for the application at different access levels. See the chapter “Passwords”.

## Message library

Under Message library you create message tables where values between 0 and 65535 are linked to texts. See the chapter “Message library”.

## 5.16 The Setup Menu

Functions for configuration of the terminal are to be found in the **Setup** menu.





## System signals

Setting of the handshaking signals between the terminal and the PLC system.

### Current display reg:

Data register in the PLC system which contains the number of the block shown on the display in the run-time mode. The data register is updated automatically by the terminal on changing block. This register does not affect the choice of textblock.

### New display reg:

A PLC data register that determines which block will be shown on the display. D100 can be stated for example. The value 34 is entered into D100 from the PLC program or the terminal if block 34 is to be shown on the display.

### Buzzer reg:

A register whose value decides the sound of the buzzer. See the table below. If the value is 0 then the buzzer is silent. The unit in the table is Hz.

	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>A</b>	<b>B</b>
<b>Contra</b>	33	37	41	44	49	55	62
<b>Large</b>	65	73	82	87	98	110	123
<b>Small</b>	131	147	165	175	196	220	247
<b>One</b>	262	294	330	349	392	440	494
<b>Two</b>	523	587	659	698	784	880	988
<b>Three</b>	1046	1174	1318	1397	1568	1760	1975
<b>Four</b>	2093	2348	2636	2794	3136	3520	3950
<b>Five</b>	4186						

### Backlight signal:

Digital signal which determines if the display is to be switched on or off.

**Cursor ctrl. block:**

Start register to a control block in the terminal which writes the current cursor position to the data register in the PLC system.

Register	Description
<b>Dn0</b>	Current cursor position X (in pixels), 0-319.
<b>Dn1</b>	Current cursor position Y (in pixels), 0-239
<b>Dn2</b>	Status register 0 Normal 1 User trying to move down but there is no object there. 2 User trying to move up but there is no object there. 3 User trying to move left but there is no object there. 4 User trying to move right but there is no object there.

Start register to a control block in E910 which writes the current touch position to the data register in the PLC system.

Register	Description
<b>Dn0</b>	X-coordinate (in pixel), 0-319.
<b>Dn1</b>	Y-coordinate (in pixel), 0-239.
<b>Dn2</b>	Status register 0 Not pressed 1 Pressed

### Recipe ctrl.block:

Control block for loading/saving/deleting off recipe files via the PLC system

Register	Content	Description
<b>Dn0</b>	Command	Command register set by the PLC Available commands: 0: No command. 1: Save recipe to the terminal. 2: Read recipe from the terminal. 3: Delete recipe from the terminal.
<b>Dn1</b>	Result code	Handshake register set by the terminal 0: Ready for new command. 1: OK. 2: Recipe file write error. 3: Recipe file not found
<b>Dn2</b>	File name char. 1-2	Recipe file name in the terminal
<b>Dn3</b>	File name char. 3-4	
<b>Dn4</b>	File name char. 5-6	
<b>Dn5</b>	File name char. 7-8	
<b>Dn6</b>	Start data register	First data register to be read/written to/from recipe file.
<b>Dn7</b>	Number of registers	Number of registers to read/write to/from recipe file.

### Current recipe reg:

States the first of four registers where the terminal put the name of the last loaded recipe from the system. The name can be presented in an ASCII object. The function takes four registers (eight characters) independently of the recipe name.

### Library index reg:

Used for indexing message libraries. In the Message object you state the number of the library from which the texts are loaded. If an index register is defined, the content of the index register is added to the stated number in the object. This makes it possible to control from which library the texts are loaded.

## Commands

One or more of the following commands can be stated on the command line. The commands are separated by a space.

Command	Description
<b>DGP</b>	Delete the alarm group name from the alarm print out.
<b>NTx</b>	Timeout in ms for messages in the No protocol mode.
<b>PDxxxxxxx</b>	Password xxxxxxxx that protects the transfer menu.
<b>PSxxxxxxx</b>	Password xxxxxxxx that will give access to all functions in the terminal.
<b>RPD</b>	RUN/PROG disable. Turns off the possibility to switch between RUN/PROG with the keys [BACKSPACE] and [MAIN]. If RPD is used you switch mode via the MAC Programmer+.
<b>BR</b>	Batch Read. Puts the terminal in transparent mode communication when using MELSEC PLC systems.
<b>DD</b>	Disable Delete alarms in the alarm list. When stated it is not possible to delete inactive and acknowledged alarms from the alarm list.
<b>DIMDx</b>	Data register containing a value between -63 and +63 controlling the display brightness. -63 means that the display is as dark as possible and +63 means that the display is as light as possible. Normal value is 0.
<b>Rx</b>	Maximum number of retransmissions. x = number of attempts.
<b>BSD</b>	Disable Baud Switching when communicating via transparent mode with FX2N and Q. BSD must be set in the terminal not connected to the PLC-system.
<b>SW</b>	Converts text with Swedish ASCII (7 bits) characters to 8 bits IBM PC extended ASCII on printing.
<b>Tx</b>	Global timeout for x milliseconds.
<b>BFF</b>	Block Form Feed. Puts a form feed after each block at print-out.
<b>AKMx</b>	Activates the joystick function, see the section “Joystick” function” in the chapter “Terminal functions”.
<b>NHD</b>	No header. Possible to printout graphic blocks with no header (containing block name, block number, date and time) on a laser printer.

## Index register

Index addressing of dynamic objects. For more information see the chapter “Index addressing”.

## Country settings

### Language:

Choice of system message language; *British English*, *German*, *Swedish* or *American English*. The default setting is British English menu texts.

### Character set:

The choice of character set affects the font table used in the terminal and which national characters that will be used.

Character set	Font table in graphic based terminals
Swedish	437
German	437
French	850
Spanish	850
Norwegian/Danish	850
Russian	866
Slavic	852

Different national characters (see the section “The terminal’s keyboard” in the chapter “Terminal functions”) are used depending on the chosen character set.

	Swedish	German	French	Spanish	Nor/Dan
<b>C1</b>	Å	Ü	È	Ñ	Å
<b>C2</b>	Ä	Ä	É	É	Æ
<b>C3</b>	Ö	Ö	Ê	Ó	Ö
<b>C4</b>	å	ß	è	Á	Ø
<b>C5</b>	ä	ü	é	ñ	å
<b>C6</b>	ö	ä	ê	é	æ
<b>C7</b>		ö		ó	ö
<b>C8</b>		ß		á	ø

When Slavic or Russian are chosen the national characters are not used.

## **Date/Time format**

Setting of date and time format.

### **Date format:**

The default format can be YY-MM-DD, YYMMDD, DD.M.YY, DD/MM/YY or MM/DD/YY where Y=year, M=month and D=day.

### **Time format:**

The default format can be HH:MM:SS or HH:MM where H=hour, M=minute and S=second.

### **Clock used:**

Select Terminal or Controller. Terminal means that the terminal's built-in clock is used and controller means that the clock in the controller system is used in the terminal.

### **Clock → Controller:**

Only valid if the clock in the terminal is used. Mark if the clock in the terminal is to be transferred to the controller system.

---

#### **Note!**

If the controller system has a real-time clock that is activated and the terminal clock is transferred to the same registers in the controller system is using, the clock in the controller system is superior.

---

### **Update interval:**

State how often the terminal will transfer the clock data to the controller system. The value is entered in seconds and 60 is recommended. A shorter updating interval makes the communication between the terminal and the controller system slower.

### **Controller register:**

Enter the first address of the registers that will be used to store date and time in the controller system.

### Controller:

Settings controller system MELSEC A series.

Data register	8 highest bits	8 lowest bits
<b>D9025</b>	year	month
<b>D9026</b>	day	hour
<b>D9027</b>	minute	second
<b>D9028</b>	–	weekday

Settings PLC system MELSEC FX series.

Data register	Presentation
<b>D8013</b>	second
<b>D8014</b>	minute
<b>D8015</b>	hour
<b>D8016</b>	day
<b>D8017</b>	month
<b>D8018</b>	year
<b>D8019</b>	weekday

If another driver than MELSEC is used, see the manual of the PLC system how the date and time are stored. The terminal stores the information in the same order as the default settings for the PLC system.

### Daylight saving:

The beginning and end of summer or Daylight saving time are defined here. State the day of the week, week in the month, month or hour. The default setting in the PC software is selectable between Europe, the US and User defined. If you do not want to use Daylight Saving Time, leave the two entry fields for month empty.

### Online settings:

Allow change of selected function in run-time mode.

## Terminal options

### Screen save time:

The time in minutes after the last operation on the display when it will switch off. The default setting is 10 minutes. This function is used to optimize the display's working life. See also the section Backlight action in the chapter "Alarm handling".

### Key delay time:

The time interval between pressing the same key before the cursor moves to the next position. Used for input of ASCII characters (A-Z etc.). See the section Alpha/numerical keys in the chapter "Terminal functions".

### Key beep:

States whether the terminal will beep when a key is pressed.

### Key repeat:

States whether the numerical keys, arrow keys and built-in function keys are to be repeated as long as the key is pressed. Function keys and input of ASCII characters (A-Z etc) cannot be repeated.

### Background:

Select terminal background color in the list of choices by clicking BG.

### Foreground:

Select terminal text color in the list of choices by clicking FG.

### Window:

Select the color of the windows shown on the display by clicking Windows.

## Trend settings

Common settings for all trends.

### Store changed samples

Select to store samples in trends only if the value has changed since the last sample.

### Store all samples

Select to store all samples in trends even if the value has not change since the last sampling. The parameters are valid for all defined trends.

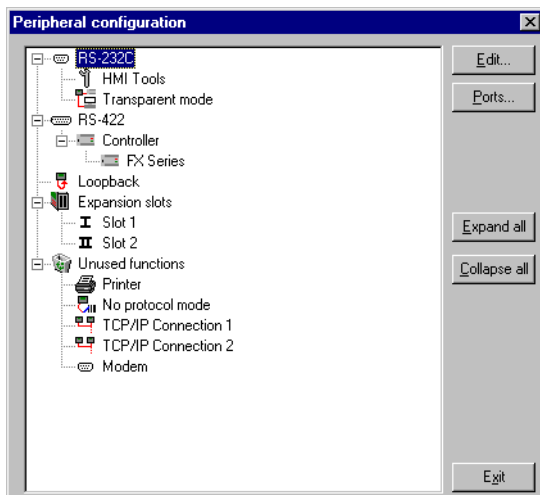


## Alarm settings

General setting of the alarm list. For more information see the section on “Alarm handling”.

## Peripherals

All communication configurations are made under **Peripherals**. When you select Peripherals the following is displayed. The display shows the units defined for the system. You can drag and drop the units.



### Edit...

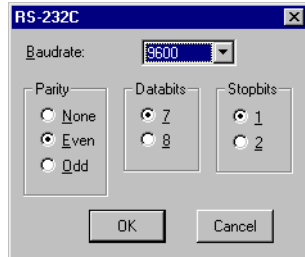
You can mark a unit or a port on the display and press **Edit...** or click the right mouse button. Then the dialog for the selected unit/port is displayed.

### Ports...

When you select **Ports...** a dialog is displayed where you can see current configurations. The configurations can also be changed here.

## RS-232C

When you select the **RS-232C** port and press **Edit...** or click the right mouse button the following dialog is displayed. You select baudrate, parity, number of data bits and stop bits for the port.



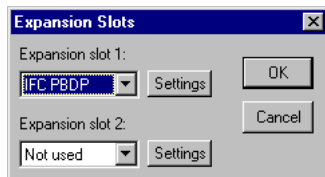
## RS-422

When you select the **RS-422** port and press **Edit...** or click the right mouse button the following dialog is displayed. You select baudrate, parity, number of data bits and stop bits for the port.



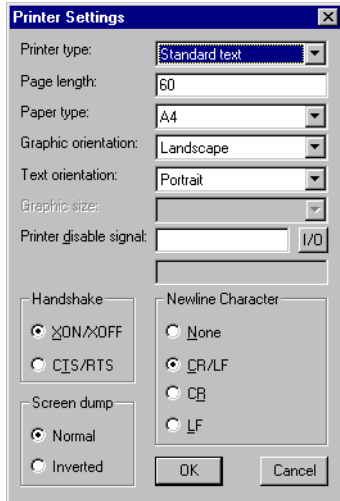
## Expansion slot

When you select an **Expansion slot** and press **Edit...** or click the right mouse button you can define which expansion card you want to use and then make the settings for the expansion cards. For further information, see the manuals for the respective cards.



## Printer

When you select **Printer** and press **Edit...** the dialog for printer settings is displayed.



### Printer type:

Select printer, None or e.g. HP PCL5.

### Page length:

The number of lines which will be printed out before form feed. If the page length is set to 0 form feed never occurs. Default setting is 60.

### Paper Type:

Select type of paper.

### Graphic orientation:

Select the graphics orientation, portrait or landscape

### Text orientation:

Select text orientation, portrait or landscape.

### Graphic size:

Select size of the graphic printout.

### Printer disable signal:

Digital signal for stopping the print out

### New line character:

Marks the required new line character; CR/LF, CR, LF or none.

**Handshake:**

Marks the required type of handshaking between the printer and the terminal; XON/XOFF or CTS/RTS. Refer to the printer manual for information about suitable handshaking.

**Screen dump:**

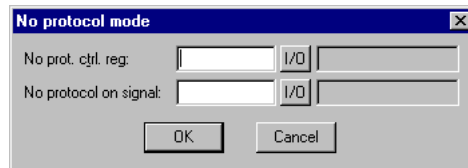
Printout of the screen. Select normal or inverted printout.

**Note!**

The settings on the printer are used for parameters such as character set, character size and margins.

**No protocol mode**

When you select **No protocol mode** and press **Edit...** or click the right mouse button the following dialog is displayed.

**No prot. ctrl. reg.:**

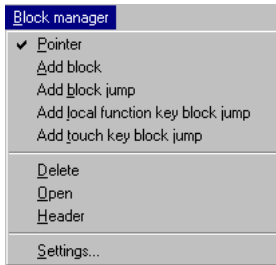
The first control register used in the No protocol mode. The No protocol mode is described in chapter "Communication".

**No protocol on signal:**

A digital signal for switching between No protocol and Transparent modes. Used for switching between No protocol and Transparent modes in run-time mode, for example, to ring up a computer and send a message.

## 5.17 The Block Manager Menu

The **Block manager** menu has functions for creating blocks and block jumps.



## 5.18 The Transfer Menu

The Transfer menu has functions for the transfer of projects as well as communication settings for transferring between a PC and the terminal. See the chapter “Project transfer”.



---

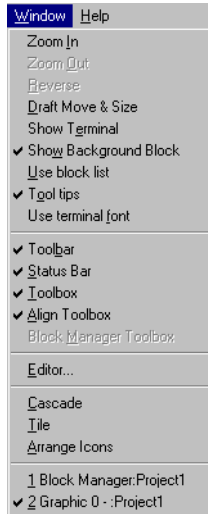
### Note!

The communication settings must be the same in the PC software as in the terminal.

---

## 5.19 The Window Menu

In the window menu there is basic functions for windows and functions unique for MAC Programmer+. Here we only describe the functions special for MAC Programmer+.



### Draft Move & Size

The function turns off the immediate updating of the sized/moved object.

### Show Terminal

A terminal is drawn around the working area in the graphic block editor. From this terminal you can reach the editors for defining LEDs, function keys and insertable text strings. When you double click on the respective function the corresponding dialog is displayed.

### Show Background block

Only valid for graphic blocks. The background block is shown when editing in the graphic block editor.

### Use Block list

You can select if the software shall activate the block manager or the block list when you create a new project.

### Tool tips

Show explanatory text for the function the cursor are pointing at.

### **Use terminal font**

The character set defined in Country settings in the Setup menu is used in MAC Programmer+. This makes it possible to edit, for example, Russian projects without a russian version of Windows.

### **Toolbar**

Show the toolbar.

### **Status Bar**

Show the status bar

### **Toolbox**

Show the toolbox.

### **Align Toolbox**

Show the align toolbox.

### **Block Manager toolbox**

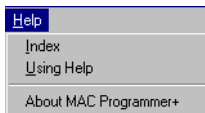
Show the toolbox for the block manager.

### **Editor**

Open stated graphic editor.

## **5.20 The Help Menu**

The **Help** menu contains help texts.



## 6 Graphic presentation and maneuvering

In this section the graphic objects are first presented in tabular form and each object is then described.

### 6.1 General parameters

#### Colors

In terminals supporting colors there are 16 colors available for objects, black, red, prussian blue, magenta, yellow, gray, light gray, light green, light red, light prussian blue, light magenta, light yellow and white. Bit-maps can be shown in 256 colors

The colors allow the creation of more realistic objects with 3-D effects and shadowing. Apart from background and foreground colors for the blocks, the colors can also be selected for the graphic objects.

The colors for the background, text and windows are determined when the terminal is configured. Colors can be chosen, for example, for the scales and curves for graphic objects.

#### Scaling

The *Offset* and *Gain* parameters are used to change the scale of the register value to a shown value according to the following equation:

$$\text{DisplayValue} = \text{Offset} + \text{Gain} * \text{RegisterValue}$$

When the object is altered from the terminal in the run-time mode, the scale is changed for the shown value according to the following equation:

$$\text{RegisterValue} = (\text{DisplayValue} - \text{Offset}) / \text{Gain}$$

Scaling does not affect the defined max and min values nor the number of decimal places.

---

#### Note!

The functions Increment and Decrement affect the RegisterValue of the maneuverable object, and not the DisplayValue.

---



## Offset and Gain calculation

The **OGC** function in MAC Programmer+ is an aid for calculating the Offset and Gain parameters. Enter the max and min values for the PLC register and the max and min values for the presented value. The function will then calculate the correct value of the parameters Offset and Gain.

## 6.2 Graphic objects

### Static graphic objects









Static graphic objects are used to draw graphics that are not affected by the process.

---

#### Note!

Static object will always be drawn behind dynamic objects in run-time mode.

---




Symbol	Object
	<b>Arc</b>
	<b>Ellipse, filled</b>
	<b>Line</b>
	<b>Polygon line</b>
	<b>Rectangle, filled, 3-D</b>
	<b>Key field object</b>
	<b>Symbol</b>
	<b>Text</b>

## Dynamic digital graphic objects

Dynamic digital graphic objects are linked to digital signals in the PLC system.

### Note!

It is not allowed to place dynamic objects on top of each other.






Symbol	Object	Description
	<b>Digital symbol</b>	Switch between two symbols depending on the state of a digital signal.
	<b>Digital text</b>	Switch between two texts depending on the state of a digital signal.
	<b>Filled object</b>	Used to fill in a framed area in one of two colors depending on the state of a digital signal.










## Dynamic analog graphic objects

Dynamic analog graphic objects are linked to the registers in the PLC system.






### Note!

It is not allowed to place dynamic objects on top of each other.

Symbol	Object	Description
	<b>ASCII</b>	Handle ASCII strings in Graphic blocks.
	<b>Analog fill</b>	Used to fill in a framed area in one of sixteen colors depending on the value of a register.
	<b>Diagram</b>	Used to draw X/Y diagrams based on the content of the PLC registers.
	<b>Multichoice</b>	Linked to a PLC register which can assume up to eight different states. Text of up to 30 characters can be linked to each state.
	<b>Multisymbol</b>	Shows one of up to eight symbols depending on the value in the PLC register. Also allows moving of a symbol across the screen.

Symbol	Object	Description
	<b>Numeric</b>	Entering and presentation of values as numbers
	<b>Slide</b>	Makes possible increasing and decreasing the value of an analog signal.
	<b>S meter</b>	Creates a graphic speedometer on the display.
	<b>Bar</b>	Presents values as a bar chart.
	<b>Trend</b>	Presents values collected from the registers as a curve.
	<b>VU meter</b>	Creates a graphic VU meter on the display.
	<b>Message</b>	Used to display texts from a Message library.
	<b>Analog numeric table</b>	Creates a table with numeric objects.
	<b>Touch key</b>	Only E910. Used to create a pressure-sensitive surface on the screen.

## Other objects

Symbol	Object	Description
	<b>Jump</b>	Jump to another block.
	<b>Alarm banner</b>	Used to show a line in the alarm list.
	<b>Digital clock</b>	Object to show a digital clock.
	<b>Analog clock</b>	Object to show an analog clock.
	<b>TCP/IP Command prompt</b>	Object to send TCP/IP commands to other units. Only valid if the terminal is used in a TCP/IP network.



## Alarm banner

Object used to show a line in the alarm list.



The 'Alarmbanner' dialog box contains the following settings:

- List line number: 1
- Positions: 30
- Font size: 8x8
- Alarm group: ALL
- Security Level: 0
- Show Day of the week: ☐
- Show Date: ☐
- Show Time: ☐
- Show Symbol: ☐
- Ack possible: ☒
- Frame:
  - ☐ None
  - ☒ Raised
  - ☐ Inset
- BG: ☐
- Buttons: OK, Cancel

Parameter	Description
<b>List line number</b>	States the number of the line in the alarm list from which information is to be collected (1=first line, 2=second line etc.) if the stated alarm group is shown in the alarm list.
<b>Positions</b>	The number of positions to be displayed.
<b>Font size</b>	Select font size for the text in the object.
<b>Alarm group</b>	States which alarm group will be shown on the alarm line. The object will be shown in the color specified for the alarm group.
<b>Security level</b>	Select security level for the object.
<b>Show day of the week</b>	Select whether the day of the week will be shown in the alarm line.
<b>Show date</b>	Select whether the date will be shown in the alarm line.
<b>Show time</b>	Select whether the time will be shown in the alarm line.
<b>Show symbol</b>	Select whether the alarm status symbols will be shown in the alarm line. See the chapter "Alarm handling".
<b>Ack possible</b>	Select whether it will be possible to acknowledge the alarm with the ACK key on the terminal.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color for the object.

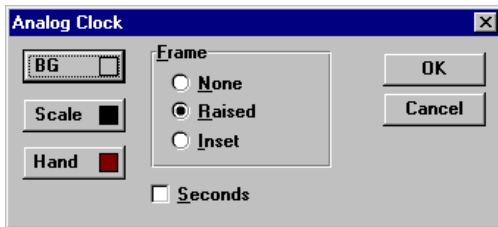
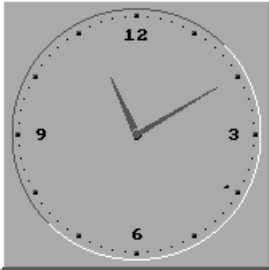
### Note!

The foreground color of the alarm text is defined in the alarm group definition.



## Analog clock

Time object for presenting an analog clock.



Parameter	Description
<b>Seconds</b>	Select whether the clock will be shown with a second hand.
<b>Frame</b>	Select whether the clock shall be shown with a frame.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select the color of the scale in the object.
<b>Hand</b>	Select the color of the second hand in the object.

---

### Note!

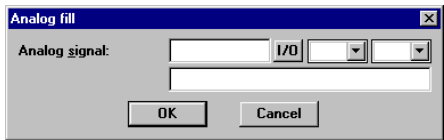
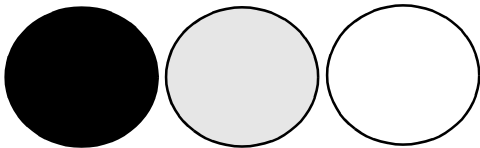
A maneuverable Date/Time object must be defined to set the clock in the terminal during run-time.

---



## Analog fill

Objects used for filling framed areas with one of sixteen colors depending on the value in a register. The colors are used according to the following table.



Parameter	Description
<b>Analog signal</b>	State the register whose value decides the color of the object. See the following table.

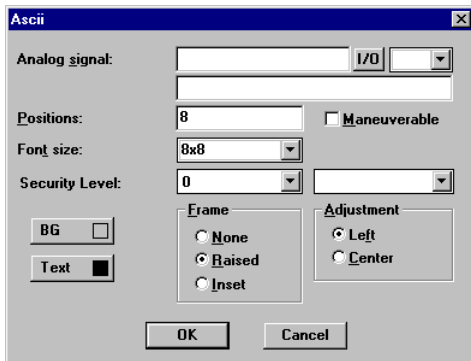
Register value	Color	Register value	Color
<b>0</b>	Black	<b>8</b>	Gray
<b>1</b>	Blue	<b>9</b>	Light blue
<b>2</b>	Green	<b>10</b>	Light green
<b>3</b>	Cyan	<b>11</b>	Light cyan
<b>4</b>	Red	<b>12</b>	Light red
<b>5</b>	Magenta	<b>13</b>	Light magenta
<b>6</b>	Yellow	<b>14</b>	Light yellow
<b>7</b>	Light gray	<b>15</b>	White

For limitations and how to place an object, see “Digital fill”.



## ASCII

Object for handling ASCII strings in Graphic blocks. ASCII objects provide the opportunity to present texts stored in the CPU's data registers. The texts should be in IBM extended ASCII format. The text is converted from 8-bit IBM extended ASCII to Swedish 7-bits ASCII if SW is stated on the command line in System Signals.

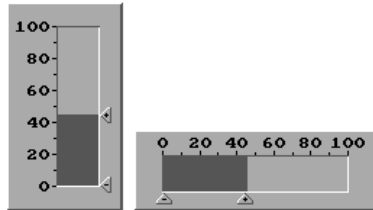


Parameter	Description
<b>Analog signal</b>	State the register where the text for the first position is to be stored.
<b>Positions</b>	State the number of positions the text will occupy on the display. Each register contains two characters (occupying two positions).
<b>Font size</b>	Select font size.
<b>Security Level</b>	Select security level for the object.
<b>Maneuverable</b>	When selected the object can be forced from the terminal in run-time mode.
<b>Adjustment</b>	Select whether the object is to be left aligned or centred.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color of the object.
<b>Text</b>	Select the color of the text in the object.



## Bar

Objects which present whole or decimal numbers in the form of a bar graph.



**Analog Bar Graph**

Analog signal:  I/O

Scale division:  20

Scale ticks:  10

Minimum value:  0

Maximum value:  100

Offset:  0 OGC

Gain:  1

Pattern:  Solid

Font size:  8x8

☒ Box ☒ Scale

BG ☐ Scale ☐ Fill ☐

OK Cancel

**Direction**

☐ Up ☐ Down ☐ Right ☐ Left

**Indicators**

☐ None ☐ Min ☐ Max ☐ Min & Max

**Frame**

☐ None ☐ Raised ☐ Inset

Parameter	Description
<b>Analog signal</b>	The address of the signal.
<b>Minimum value</b>	The minimum value the signal can have.
<b>Maximum value</b>	The maximum value the signal can have.
<b>Direction</b>	Select whether the bar is to be shown to the right, left, up or down.
<b>Scale</b>	Select whether the scale is to be shown.
<b>Box</b>	Select whether a box is to be drawn around the bar.
<b>Indicators</b>	Select whether the highest and/or the lowest value for the signal shall be shown on the axis. Indicators are set to zero when the terminal is switched on. Zeroing can also be carried out in the run-time mode, see the Maneuvering Graphic blocks section. The indicators only function for 16-bit numbers.
<b>Scale division</b>	State the interval between the scale numbers to be drawn.
<b>Scale ticks</b>	State the interval between the scale ticks displayed.
<b>Fill</b>	Select fill color.
<b>Offset and Gain</b>	Used to scale register value. See the section "General parameters".
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select the color of the scale in the object.

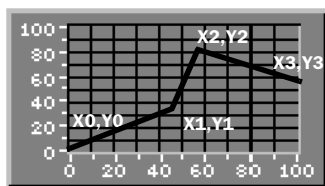




## Diagram

Objects used to draw X/Y diagrams or bar graphs based on the content of the PLC register. This is a real-time function. The object is normally used for non-time-based presentations. Time-based display with a sample time of <1 second can be shown if the PLC creates data collection. As an example we shall say the value in data register 100 is the first X-coordinate and the value in data register 200 the first Y-coordinate. The number of register pairs will be 4. The table and the figure below illustrate our example.

X-coordinate	Register	Value	Y-coordinate	Register	Value
X0	100	0	Y0	200	11
X1	101	41	Y1	201	40
X2	102	51	Y2	202	85
X3	103	92	Y3	203	62



**Diagram**

Curve:

Register pair count:

☐ Maneuverable

X Analog signal:  I/O

Y Analog signal:  I/O

Value form: ☒ Line ☐ Bar

Frame: ☐ None ☒ Raised ☐ Inset

Font size:

Y-axis: Scale division:  Scale ticks:  Minimum value:  Maximum value:

X-axis: Scale division:  Scale ticks:  Minimum value:  Maximum value:

Thickness: ☐ Thin ☒ Thick

Diagram maneuver:

Current graph signal:  I/O

Current cursor signal:  I/O

X Edit step:

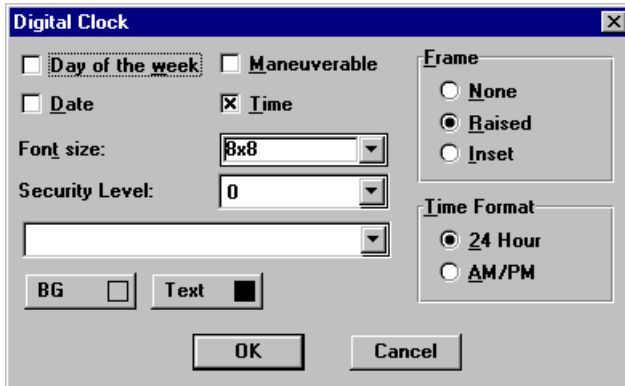
Y Edit step:

Parameter	Description
<b>Register pair count</b>	The number of register pairs to be drawn as dots or bars for the curve.
<b>Value form</b>	Select whether the diagram shall be a bar graph or linear graph for the curve. A bar is drawn for each register pair in the bar graph. The X/Y coordinates are marked in as dots between which a line is drawn in a linear graph.
<b>Maneuverable</b>	Select if the curve respective can be altered from the terminal in run-time mode.
<b>Curve</b>	Select the color of the curve.
<b>X Analog signal</b>	The PLC register which contains the first X-coordinate for the curve.
<b>Y Analog signal</b>	The PLC register which contains the first Y-coordinate for the curve.
<b>Y Scale division</b>	The interval between the scale numbers on the Y-axis.
<b>Y Scale ticks</b>	The interval between the scale ticks displayed on the Y-axis.
<b>Y Minimum value</b>	The minimum value of the Y-coordinate.
<b>Y Maximum value</b>	The maximum value of the Y-coordinate.
<b>X Scale division</b>	The interval between the scale numbers on the X-axis.
<b>X Scale ticks</b>	The interval between the scale ticks displayed on the X-axis.
<b>X Minimum value</b>	The minimum value of the X-coordinate.
<b>X Maximum value</b>	The maximum value of the X-coordinate.
<b>Current graph signal</b>	The register which contents decides what curve to be edited in run-time mode.
<b>Current cursor signal</b>	The register which contents decides which point on the curve to be edited in run-time mode.
<b>X Edit step</b>	The size of the step when you press an arrow key in run-time.
<b>Y Edit step</b>	The size of the step when you press an arrow key in run-time.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>Grid</b>	When selected a grid is drawn in the diagram.
<b>BG</b>	Select background color of the object.
<b>Scale</b>	Select the color of the scale for the object.
<b>Grid color</b>	Select the color of the grid in the diagram.



## Digital Clock

Time object for presenting a digital clock.



Parameter	Description
<b>Day of the week</b>	Select whether the day of the week will be shown.
<b>Date</b>	Select whether the date will be shown.
<b>Time</b>	Select whether the time will be shown.
<b>Time format</b>	Time can be shown either as AM/PM or a 24-hour clock.
<b>Maneuverable</b>	Select whether the clock can be reset in run-time mode.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>Font size</b>	Select font size for the text in the object.
<b>Security level</b>	Select security level for the object. 0 means no security level.
<b>BG</b>	Select background color.
<b>Text</b>	Select the color of the text in the object.

---

### Note!

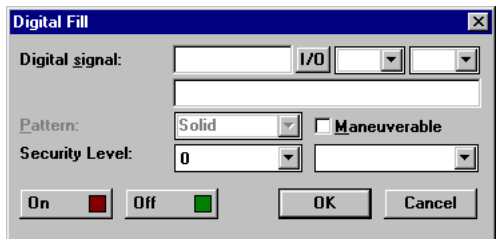
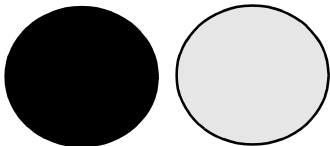
A maneuverable Date/Time object must be defined to set the clock in the terminal during run-time.

---



## Digital fill

Objects used for filling framed areas with a selected color.



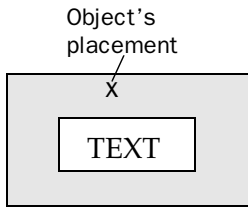
### Note!

Filling areas that are too irregular can cause system failure in run-time mode. Filling can, in certain cases, make the screen slow.

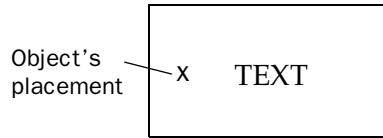
Parameter	Description
<b>Digital signal</b>	The address of the digital signal.
<b>Maneuverable</b>	When selected the object can be forced from the terminal in run-time mode.
<b>Security Level</b>	Select security level for the object.
<b>On</b>	Select the color of the object when the signal is 1.
<b>Off</b>	Select the color of the object when the signal is 0.

## Placing an object

The program calculates what is to be filled in the specified area. It is therefore important that the object is correctly placed. Incorrect placed fillers may cause errors in the application during run-time. The filled area is limited only by static objects and static parts of dynamic objects. Fill objects can be replaced with Digital Symbol or Multi Symbol to give better performance to the project.



*Good! To give the display better response, frame the text so it does not need to be filled.*

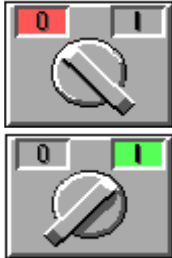


*Bad! The display will be slow, because the program must make many calculations to fill in the irregular areas between letters.*



## Digital symbol

Object that is used to switch between two chosen symbols depending on the state of a digital signal.



**Digital Symbol**

Digital signal:  I/O

Symbol OFF:   ☐ Maneuverable

Symbol ON:

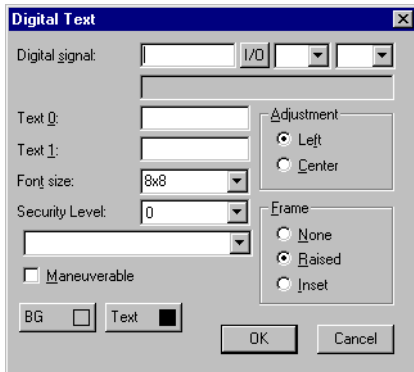
Security Level:  0

Parameter	Description
<b>Digital signal</b>	The address of the signal.
<b>Symbol OFF</b>	Select the symbol that is to be shown when the signal is 0.
<b>Symbol ON</b>	Select the symbol that is to be shown when the signal is 1.
<b>Maneuverable</b>	When selected the object can be forced from the terminal in run-time mode.
<b>Security Level</b>	Select security level for the object. 0 means no security level.



## Digital text

Text object which is used to switch between two entered texts depending on the state of a digital signal. The maximum text length is 30 characters.



Parameter	Description
<b>Digital signal</b>	The address of the digital signal.
<b>Text 0</b>	The text that is to be shown when the signal is 0.
<b>Text 1</b>	The text that is to be shown when the signal is 1.
<b>Font size</b>	Select font size.
<b>Security Level</b>	Select security level. 0 means no security level.
<b>Maneuverable</b>	When selected the object can be forced from the terminal in run-time mode.
<b>Adjustment</b>	Select whether the text is to be left aligned or centred.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color of the object.
<b>Text</b>	Select text color in the object.



## Jump

Object for jumping to another block. Makes it possible to build up a hierarchy of menus in the application. Jumping back to the previous block is done by pressing the [PREV] key (up to nine levels back). See chapter “Function keys”.



Parameter	Description
<b>Jump to block</b>	States the number of the block to which the jump is to be made.
<b>Text</b>	Enter any text that will be displayed.
<b>Positions</b>	The number of positions the text will take up on the display.
<b>Font size</b>	Select font size.
<b>Adjustment</b>	Select whether the text shall be left aligned or centred.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color for the object.
<b>Text</b>	Select the color of the text in the object.

### Note!

If a jump is generated to a block that does not exist in run-time mode, an error message is shown.





## Message

Message object presenting texts from message libraries.



The screenshot shows a 'Message' dialog box with the following controls:

- Analog signal:** A text input field followed by an 'I/O' button and a dropdown arrow.
- Library:** Two dropdown menus.
- Font size:** A dropdown menu showing '8x8'.
- Positions:** A text input field showing '0'.
- Security Level:** Two dropdown menus.
- Maneuverable range:** Two text input fields labeled '0' and 'to: 0'.
- Adjustment:** Radio buttons for 'Left' and 'Center' (selected).
- Frame:** Radio buttons for 'None', 'Raised' (selected), and 'Inset'.
- Buttons:** 'BG' (with a checkbox), 'Text' (with a black square icon), 'OK', and 'Cancel'.

Parameter	Description
<b>Analog signal</b>	Analog signal controlling which text from message chosen library is to be shown.
<b>Library</b>	Select the number of a message library. Defined under Message Library in the Function menu.
<b>Font size</b>	Select font size of the text.
<b>Positions</b>	Number of positions to be displayed. 0=automatic adjustment
<b>Security level</b>	Select security level for the object. 0 means no security level.
<b>Maneuverable range</b>	A range of maximum 64 texts can be maneuverable in run-time mode. State the numbers of the first and the last texts of the range. See the section "Maneuvering Graphic Blocks".
<b>Adjustment</b>	Select if the text shall be left aligned or centered.
<b>Frame</b>	Select if the text shall be displayed with a frame.

### Note!

When using Library index register the number of positions should not be 0 since the automatic length adjustment does not work.

For more detailed information, see the chapter "Message library".



## Multiple choice

Objects that can have several states. The object is linked to a register which can assume a maximum of eight states. Each state can be allocated a text, maximum 30 characters long.



**Multiple Choice**

Analog signal:  I/O

Text:

0:  ☐ Man. ☐ Left ☒ Center

1:  ☐ Man. ☐ Left ☒ Center

2:  ☐ Man. ☐ Left ☒ Center

3:  ☐ Man. ☐ Left ☒ Center

4:  ☐ Man. ☐ Left ☒ Center

5:  ☐ Man. ☐ Left ☒ Center

6:  ☐ Man. ☐ Left ☒ Center

7:  ☐ Man. ☐ Left ☒ Center

Font size:

Security Level:

Parameter	Description
<b>Analog signal</b>	The register that decides which text is to be displayed.
<b>Text 0-7</b>	The texts that are to be shown for each state of the object.
<b>Man 0-7</b>	When selected the object can be maneuvered to this state in run-time mode. See the section "Maneuvering Graphic Blocks".
<b>Adjustment</b>	Select whether the text shall be left aligned or centred.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>Font size</b>	Select font size in the object.
<b>Security Level</b>	Select security level for the object.
<b>BG</b>	Select background color for the object.
<b>Text</b>	Select the color of the text in the object.



## Multisymbol

Objects that can show one of a maximum of eight different symbols depending on the value in a data register. The PLC register can also control the symbol's location on the display.



Parameter	Description
<b>Analog signal</b>	The register that decides which symbol to be displayed. If the register contains the value 1 symbol 1 is displayed etc.
<b>Move X</b>	The value which contains the X-coordinate for the object. Permissible values are 0-639.
<b>Move Y</b>	The value which contains the Y-coordinate for the object. Permissible values are 0-479.
<b>Symbol 0-7</b>	Select the symbol to be displayed. If the value of the register is 0, then symbol 0, is displayed etc.

### Note!

Move X and Move Y only need to be stated if the object is to be moved around on the display with the help of the PLC.

### 0.3 Numeric

Objects for entry and presentation of values in numerical form. It is used, for example, to create entry fields.



**Analog Numeric**

Analog signal: [ ] I/O [ ] [ ]

Positions: [5]

Decimals: [0]

Minimum value: [-32768]

Maximum value: [32767]

Offset: [0] OGC [ ]

Gain: [1]

Font size: [8x8]

Security Level: [0]

Adjustment:  
☒ Right  
☐ Center

Frame:  
☐ None  
☒ Raised  
☐ Inset

Parameter	Description
<b>Analog signal</b>	The address of the signal.
<b>Positions</b>	The number of positions the entered value will take up including comma and minus sign.
<b>Decimals</b>	The number of decimals with which the value shall be entered.
<b>Minimum value</b>	The minimum limit for maneuvering.
<b>Maximum value</b>	The maximum limit for maneuvering.
<b>Offset and Gain</b>	Used to scale register value. See the section “General parameters”.
<b>Font size</b>	Select font size in the object.
<b>Security Level</b>	Select security level for the object.
<b>Adjustment</b>	Select if the input field shall be left aligned or centred.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>Maneuverable</b>	Select if the object can be altered from the terminal in the run-time mode.
<b>Zero fill</b>	Select if zeros are to be printed out in empty positions.
<b>BG</b>	Select background color for the object.
<b>Text</b>	Select the color of the text in the object.



## Numeric table

Object for creating a table of analog numeric objects.



Parameter	Description
<b>Analog signal</b>	The first signal to be presented in the table. Only 16-bit registers can be used if version 4.0 or higher on the terminal are used.
<b>X-size</b>	State the number of signals to be presented horizontally.
<b>Y-size</b>	State the number of signals to be presented vertically.
<b>Positions</b>	The number of positions you should enter the value with including comma and minus signs.
<b>Decimals</b>	The number of decimals with which you should enter the value.
<b>Minimum value</b>	The minimum limit for maneuvering.
<b>Maximum value</b>	The maximum limit for maneuvering.
<b>Offset and Gain</b>	Used to scale register value. See the section "General parameters".
<b>Font size</b>	Select the size of the text in the object.
<b>Security level</b>	Select the security level for the object.
<b>Maneuverable</b>	Select whether the object can be altered in the run-time mode.

<b>Parameter</b>	<b>Description</b>
<b>Zero fill</b>	Select if zeros are to be printed out in empty positions.
<b>Direction</b>	Select direction for the object, horizontal or vertical. The signals in the table are calculated after this direction.
<b>Table frame</b>	Select whether the whole table shall be shown with a frame.
<b>Adjustment</b>	Select whether the entry field shall be left aligned or centred.
<b>BG</b>	Select background color for the object.
<b>Text</b>	Select the color of the text in the object.
<b>Item frame</b>	Select if every single item in the table shall be shown with a frame.

---

**Note!**

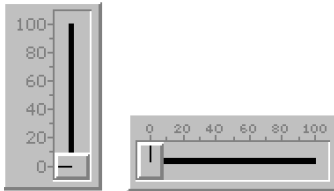
Minimum value and maximum value are only active when maneuvering the object.

---



## Slide

Objects which make it possible to increase or decrease the value of an analog signal.



Parameter	Description
<b>Analog signal</b>	The address of the analog signal.
<b>Scale division</b>	State the interval between the scale numbers to be drawn.
<b>Scale ticks</b>	State the interval between the scale ticks displayed.
<b>Minimum value</b>	The minimum value the object can have.
<b>Maximum value</b>	The maximum value the object can have.
<b>Offset and Gain</b>	Used to scale register value. See the section "General parameters".
<b>Security level</b>	Select security level for the object.
<b>Maneuverable</b>	Select whether the object can be altered from the terminal in the run-time mode.
<b>Direction</b>	Select whether the object shall be presented vertically or to the right.
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select color of the scale in the object.
<b>Button</b>	Select color of the indication button in the object.

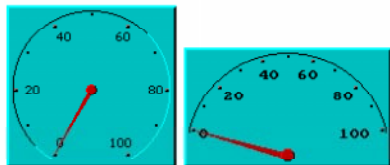
### Note!

The data type formats BCD float and Float without exponent do not work for slide objects.



## S-meter

Objects for creating a graphic speedometer on the display.



**Analog Speedometer**

Analog signal:  I/O

Scale division:

Scale ticks:

Minimum value:

Maximum value:

Angle:

Offset:  OGC

Gain:

Font size:

Frame  
☐ None  
☒ Raised  
☐ Inset

BG ☐

Scale ☐

Needle ☐

OK Cancel

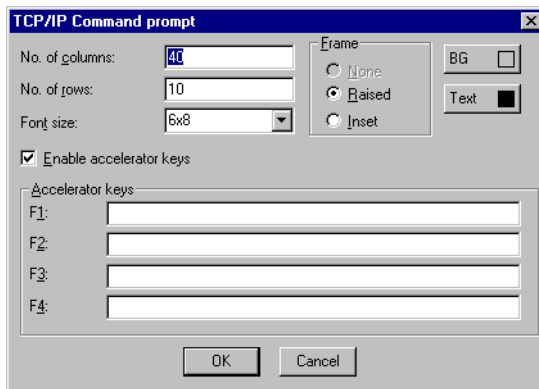
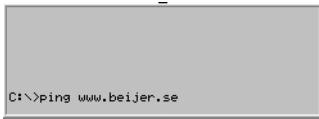
Parameter	Description
<b>Analog signal</b>	The address of the signal.
<b>Scale division</b>	State the interval between the scale numbers to be drawn.
<b>Scale ticks</b>	State the interval between the scale ticks displayed.
<b>Minimum value</b>	The minimum value the object can display.
<b>Maximum value</b>	The maximum value the object can display.
<b>Angle</b>	State the angle, working area, for the object. 10-360 degrees.
<b>Offset and Gain</b>	Used to scale register value. See the section "General parameters".
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select the color of the scale in the object.
<b>Needle</b>	Select the color of the needle in the object.





## TCP/IP Command prompt

A window in which TCP/IP commands can be entered and sent between terminals and personal computers in a TCP/IP network. In run-time, previous commands can be called up with the arrow keys [Up] and [Down].



Parameter	Description
<b>No. of columns</b>	The width of the window.
<b>No. of rows</b>	The height of the window.
<b>Font size</b>	Select font size in the object.
<b>Enable accelerator keys</b>	Select whether the function keys F1-F4 should write predefined commands in the window.
<b>Accelerator keys</b>	Enter any command that will be written at the command prompt by pressing respective function key.
<b>Frame</b>	Select the type of frame with which the window will be shown.
<b>BG</b>	Select background color for the window.
<b>Text</b>	Select the color of the text in the window.

Commands for the TCP/IP Command prompt.

<b>Command</b>	<b>Description</b>
IPCONFIG	Gets network parameters.
PING	Testing if a host is available.
ROUTE	Show, add, delete routes.
ARP	Show, add, delete IP/Hardware address.

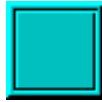
For further information about the TCP/IP network we refer to the chapter "Network communication".



## Touch key

Only valid for the E910. See the section “Maneuvering Graphic Blocks”.

The object is a pressure-sensitive surface whose function corresponds to a function key. The object is used to change pictures, control digital signals and so on.



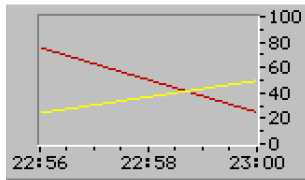
Parameter	Description
<b>Action</b>	The type of signal the object shall affect. Predefined functions are described in the chapter “Function keys”. Increment, Decrement and Set are not available.
<b>Digital signal</b>	The digital signal affected by the object if I/O is selected under Action.
<b>Block</b>	The block that will be displayed if the object is defined as a jump object if block type is selected under Action.
<b>Security level</b>	Select security level for the object.
<b>Button type</b>	Select required type of button, circular, rectangular or invisible.
<b>IO event</b>	States how the object shall affect the signal. Momentary means that the signal is affected as long as the object is touched. Toggle means that the signal is set to one or zero alternately when the object is affected. On means that the signal is set when the object is affected. Off means that the signal is reset when the object is affected.
<b>BG</b>	Select background color for the object.

An invisible, pressure-sensitive “virtual” surface can be used to define the area for block changes in an overview display of, for example, a machine. Detailed images are linked to defined parts of the overview display. When the pressure-sensitive areas are pressed, the detailed images appear.



## Trend

Objects which graphically present collected values from analog signals.



**Trend**

Name:

Curve:

Analog signal:  I/O

Offset signal:  0

Gain signal:  1

Enable:  I/O

Y Min value reg:  I/O

Y Minimum value:  0

Y Max value reg:  I/O

Y Maximum value:  100

Y Scale division:  20

Y Scale ticks:  10

Font size:  5x5

Y Value form: ☒ Linear ☐ Logarithm

Y Scale: ☐ Off ☐ Left ☒ Right ☐ Both

Frame: ☐ None ☒ Raised ☐ Inset

X Sample interval:  00:00:10

X Stored samples:  100

X Time range:  00:04:00

X Time scale div.:  00:02:00

X Time scale ticks:  00:00:30

Sample full limit:  0

Sample full signal:  I/O

☒ Grid

☐ Scale

Parameter	Description
<b>Name</b>	State the name of the object. The name must be unique for each object.
<b>Curve</b>	Select the curve you want to edit (1-6).
<b>Analog signal</b>	The analog signals the object shall sample and present the values for. Only 16-bit numbers can be used. For valid devices see the driver manual.
<b>Color</b>	Select a color for each curve.
<b>Offset and Gain</b>	Used to scale register value. See the section "General parameters".

<b>Parameter</b>	<b>Description</b>
<b>Enable</b>	The digital signal which when set starts the collection of data. Data collection ceases if the signal is set to zero. The parameter does not need to be stated.
<b>Y Min value reg.</b>	The minimum value on the Y-axis loaded from the stated PLC register.
<b>Y Minimum value</b>	The minimum value on the Y-axis. (If Y Min value reg. is defined this is the minimum value that can be loaded from the register.)
<b>Y Max value reg.</b>	The maximum value on the Y-axis loaded from the stated PLC register.
<b>Y Maximum value</b>	The maximum value on the Y-axis. (If Y Max value reg. is defined this is the maximum value that can be loaded from the register.)
<b>Y Scale</b>	States which side of the Y-axis shall be shown, to the left, right, both or neither.
<b>Y Scale division</b>	The interval between the scale numbers on the Y-axis.
<b>Y Scale ticks</b>	The interval between the scale ticks displayed.
<b>X Sample interval</b>	States the time interval between the collection of data. Minimum value is 1 second and maximum value is 100 hours.
<b>X Stored samples</b>	States how many values shall be stored. The maximum number of values are 4800.
<b>X Time range</b>	States the time interval to be shown in the trend diagram.
<b>X Time scale division</b>	The interval between the scale numbers on the X-axis.
<b>X Time scale ticks</b>	The interval between the scale ticks displayed.
<b>Sample full limit</b>	State the number of samples when the samples full signal shall be set.
<b>Sample full signal</b>	Select a signal which will be set when the number of samples reach the value stated in the parameter Sample full limit.
<b>Frame</b>	Select if a frame shall be printed around the object.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select the color of the scale in the object.
<b>Erase trend data signal</b>	The digital signal which when set will erase all historic trend data.
<b>Grid</b>	Select if grid shall be displayed in the object.
<b>Grid color</b>	Select a suitable color for the grid.

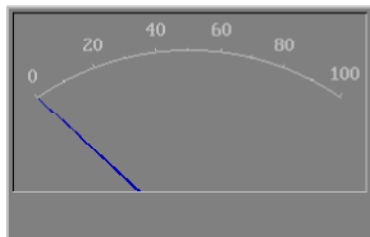
**Note!**

If you copy a block containing a trend object you must change the name of the trend object. Two trend objects may not have the same name.



## VU-meter

Object for creating a graphic VU meter on the display.



Parameter	Description
<b>Analog signal</b>	The address of the signal.
<b>Scale division</b>	State the interval between the scale numbers to be drawn.
<b>Scale ticks</b>	State the interval between the scale ticks displayed.
<b>Minimum value</b>	The minimum value the signal can have.
<b>Maximum value</b>	The maximum value the signal can have.
<b>Offset and Gain</b>	Used to scale register value. See the section "General parameters".
<b>Frame</b>	Select if a frame shall be drawn around the object.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select the color of the scale in the object.
<b>Needle</b>	Select the color of the needle in the object.

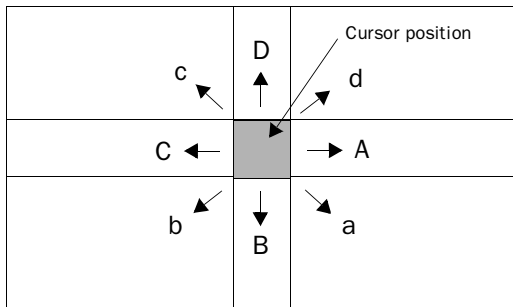
## 6.3 Maneuvering Graphic Blocks

Not valid for the E910.

Move between the maneuverable objects with the arrow keys. Objects selected are highlighted with a blinking frame.

### Selecting maneuverable objects

The arrow keys are used to jump between maneuverable objects in the terminal during run-time. The choice of object is according to the following principle: The cursor's position is assumed to be the center of a cross. Press the right arrow and the first object found in area A according to the diagram below is selected. Should the system not find an object in the n arrow field on the right, it searches in area a. The down arrow searches in areas B and b, the left arrow searches in areas C and c and the up arrow searches in areas D and d.



### Digital objects

Digital objects, text, symbol and filled objects, switch status when [ENTER] is pressed. If the INC and DEC functions have been linked to function keys, the signal linked to the object with these keys is set and re-set respectively.

## **Analog objects**

### **ASCII objects**

Mark the object and enter the required text. Finish writing with [ENTER].

### **Message objects**

Place the cursor on the object and press [ENTER]. An option list is now shown with all the maneuverable states. Move stepwise to the required state and press [ENTER] and the analog signal linked to the object is forcibly set.

### **Multichoice objects**

Place the cursor on the object and press [ENTER]. An option list is now shown with all the maneuverable states. Move stepwise to the required state and press [ENTER] and the analog signal linked to the object is forcibly set.

### **Numerical objects**

To maneuver numerical objects write in the value and press [ENTER]. If a value too high or too low is entered the max or min value for the object is shown. This information will also be shown if you press [ENTER] when the object is maneuverable.

### **Numerical table objects**

When the table object is marked, press [ENTER] to select the first item in the table. Use the arrow keys to move the cursor between the items in the table. Change the value in the marked item and press [ENTER].

### **Potentiometer objects**

The object is maneuvered with the arrow keys by placing the cursor on the object and pressing [ENTER]. The value can now be increased/decreased with the number that corresponds to the setting "Scale Ticks" in the object. End maneuvering with [ENTER].

The object can also be maneuvered with the Increment and Decrement functions on the condition that they are linked to the function keys, see the "Function keys" chapter.



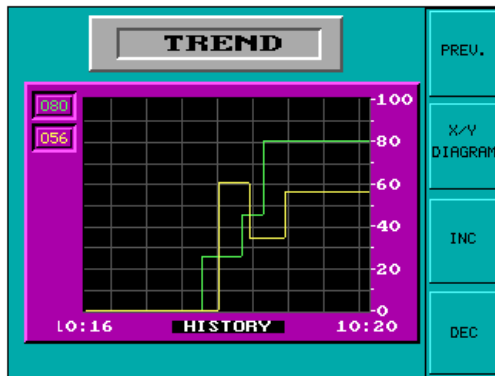
## Bar object

The min/max indicators for bar objects can be set at the current value by pressing [ENTER].

The max/min indicators are reset when you touch the bar.

## Trend objects

Trend curves can show historic data in the run-time mode. Mark the required trend object and press [ENTER]. A dialog square is now shown and the date for the data to be presented can be chosen. "History" is shown at the bottom of the square. Press [ENTER] again to return to real-time display. Trend data is stored in the files whose names are stated when the trend object is defined.



Trend History			
Left	time	:	10:32:00
Right	time	:	10:36:00
Left	date	:	97-05-21
Right	date	:	97-05-21
OK			

## **Other objects**

### **Digital clock**

The digital clock can be altered by marking the object and giving the required time. Finish with [ENTER].

---

**Note!**

If you use the clock in the PLC system and you want to change it, this must be done in run-time mode.

---

### **Jump object**

Mark the object and press [ENTER].

### **TCP/IP Command prompt**

TCP/IP Commands can be written on the highlighted row. Previous commands can be called up by pressing the arrow keys [Up] and [Down].

## **6.4 Using the touch screen**

This section is only valid for the E910.

The E910 does not have a built-in keyboard. All maneuvering takes place using the touch screen. Because of its sensitivity, you should only press one point on the screen at any given time. Otherwise, the terminal will interpret the midpoint of two points pressed.

### **Digital objects**

Digital objects, text, symbol and filled object change status when you touch them.

### **Analog objects**

#### **ASCII objects**

When you touch the object an alphanumeric keyboard will be shown on the screen. Touch in the required text and end with [ENTER].

#### **Multichoice objects**

When you touch the object a choice list appears. Select the required object by touching it.

### Numerical objects

When you touch the object a numeric keyboard appears. Touch the new value and end with [ENTER].

### Numerical table objects

When you touch the item in the table object, a numeric keyboard appears. Touch the new value and end with [ENTER].

### Potentiometer objects

Touch the button in the object and drag to maneuver the object.

### Bar object

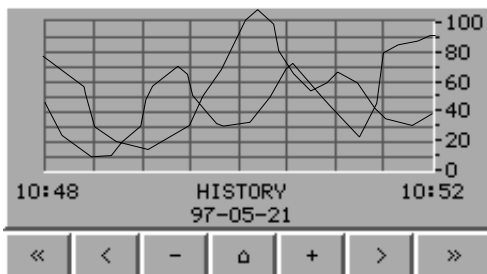
The max/min indicators are reset when you touch the bar.

### Trend objects

Trend curves can show historic data in run-time. When you touch the object, a dialog box appears where you can choose the time interval and date for presented data.

Double arrow	Move the trend one window.
Single arrows	Move the trend 1/4 window.
-	Zoom up the trend.
+	Zoom down the trend.
^	Return to default settings.

Return to real-time trend by touching the object again.



## Other objects

### Digital clock

When you touch the object a numeric keyboard appears. Touch the required time and end with [ENTER].

**Note!**

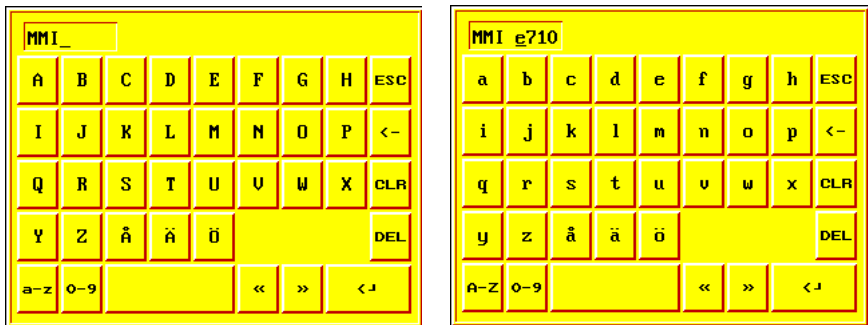
If you use the clock in the PLC system and you want to change it, this must be done in run-time mode.

### Jump object

Touch the object to activate the jump.

### Alphanumeric keyboard

The alphanumeric keyboard appears when you maneuver, for example, an ASCII object.

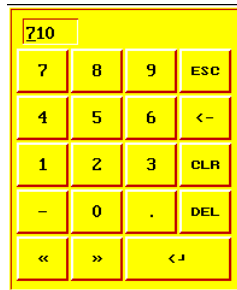
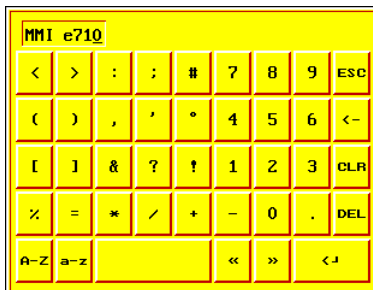


Key	Description
<b>A-Z</b>	Used to enter required text.
<b>ESC</b>	Close down the keyboard and return to previous level.
←	Delete one character to the left.
<b>CLR</b>	Delete all characters.
<b>DEL</b>	Delete the character on cursor position.
↵	Confirm the settings and close down the keyboard.
»	Move the cursor to the right.
«	Move the cursor to the left.

Key	Description
<b>a-z</b>	Switch between upper case and lower case letters.
<b>0-9</b>	Switch between lower case letters and digits and special characters.

## Numeric keyboard

The numeric keyboard appears when you maneuver, for example, a numeric object.



Key	Description
<b>0-9</b>	Used to enter required text.
<b>ESC</b>	Close down the keyboard and return to previous level.
<-	Delete one character to the left.
<b>CLR</b>	Delete all characters.
<b>DEL</b>	Delete the character at cursor position.
<J	Confirm the settings and close down the keyboard.
>>	Move the cursor to the right.
<<	Move the cursor to the left.

## **Choice list**

Choice lists will also appear on the screen. In the choice list you use the arrows to show the uppermost or lowermost choice.

[CANCEL] is used to close down the list without making a selection.

Empty	
Level 1	
Level 2	
Level 3	
Level 4	
Full	↓
CANCEL	



## 7 Text-based presentation and maneuvering

Text-based presentation and maneuvering are used to create operator dialogs in text-based terminals or are suitable for generating report print-outs of various kinds. These can be daily reports, status reports etc. For an example of what a report looks like see the “Report print-outs” chapter. Text objects in tabular form are shown in this section. Every object is then described separately.

### 7.1 General parameters

#### Scaling

The *Offset* and *Gain* parameters are used to change the scale of the register value to a shown value according to the following equation:

$$\text{DisplayValue} = \text{Offset} + \text{Gain} * \text{RegisterValue}$$

When the object is altered from the terminal in the run-time mode, the scale is changed for the shown value according to the following equation:

$$\text{RegisterValue} = (\text{DisplayValue} - \text{Offset}) / \text{Gain}$$

Scaling does not affect the defined max and min values nor the number of decimal places.

---

#### Note!

The functions Increment and Decrement affect the RegisterValue of the maneuverable object, and not the DisplayValue.

---









#### Offset and Gain calculation

The function **OGC** in MAC Programmer+ is an aid for calculating the Offset and Gain parameters. Enter the max and min values for the PLC register and the max and min values for the presented value. The function will then calculate the correct value of the parameters Offset and Gain.



## 7.2 Text object

### Dynamic text object

Symbol	Object	Description
	<b>Analog</b>	Presents values in numerical form.
	<b>Date/time</b>	Setting of date and time.
	<b>Digital</b>	Switch between two texts depending on the state of a digital signal.
	<b>Multiple-choice</b>	Links to a PLC register which can assume up to eight different states. A text of 30 characters can be linked to each state.
	<b>Jump</b>	Jump to another block.
	<b>Bar graph</b>	Presents values as a bar graph.
	<b>Text</b>	Handles ASCII strings.
	<b>Message</b>	Object showing texts in a message library.

## 0.3 Analog object

Object which presents values in numerical form as whole or decimal numbers.

Parameter	Description
<b>Analog signal</b>	The address for the signal.
<b>Positions</b>	The number of positions used by the object. Characters such as minus sign or decimal points must also be counted.
<b>Decimals</b>	The number of decimal places which represent the object.
<b>Maneuverable</b>	States if the object can be altered in run-time.
<b>Minimum value</b>	The minimum value you can enter for an object.
<b>Maximum value</b>	The maximum value you can enter for an object.
<b>Offset and Gain</b>	Used to change scale of the register value. See the section General parameters.

---

### Note!

Minimum value and maximum value are only valid when maneuvering the object.

---



## Bar graph object

Presents values in the form of a bar graph. The bar will be horizontal so that the maximum number of positions are marked when the value has reached its maximum level.

Parameter	Description
<b>Analog signal</b>	The address for the signal.
<b>Positions</b>	The number of positions which the signal uses.
<b>Direction</b>	States whether the bar will be presented to the right or left.
<b>Minimum value</b>	The minimum value for the object.
<b>Maximum value</b>	The maximum value for the object.
<b>Offset and Gain</b>	Used to change scale of the register value. See the section "General parameters".



## Date/time object

Object for setting date and time. Date and time objects collect data from the real-time clocks in the terminal or the controller system.

The screenshot shows a 'Date and Time' dialog box with a title bar and a close button. Inside, there are four checkboxes: 'Day of the week' (unchecked), 'Date' (unchecked), 'Time' (checked), and 'Maneuverable' (unchecked). Below these are two dropdown menus labeled 'Security Level:'. At the bottom are 'OK' and 'Cancel' buttons.

Parameter	Description
Day of the week	States if the day of the week will be shown. Requires two positions.
Date	States if the date will be shown. Use the form given in Date/time format in configuration.
Time	States if the time will be shown. Use the form given in Date/time format in configuration.
Maneuverable	Select if the object can be altered in run-time mode.

---

**Note!**

A maneuverable Date/Time object must be defined to set the clock in the terminal during run-time.

---



## Digital text

Object that can assume two states, zero or one.

Digital Text

Digital signal:  I/O

Text 0:

Text 1:

Security Level:

☐ Maneuverable

OK Cancel

Parameter	Description
Digital signal	The address for the signal.
Text 0	The text shown when the signal is zero.
Text 1	The text shown when the signal is one.
Maneuverable	Select if the object can be altered in run-time mode.



## Jump object

Object for jumping to another block. Makes it possible to build up a hierarchy of menus in the application. Jumping back to the previous block is done by pressing the [PREV] key (up to nine levels back). See chapter “Function keys”.

Parameter	Description
<b>Current block</b>	States the number of the block left.
<b>Jump to block</b>	States the number of the block to which the jump is to be made.

---

### Note!

If a jump is generated to a block that does not exist in run-time mode an error message is shown.

---



## Message object

Message object presenting texts from message libraries.

Parameter	Description
<b>Analog signal</b>	Analog signal controlling which text from chosen message library to be shown. For valid devices see the driver manual.
<b>Library</b>	Select message library. Defined under Message Library in the Function menu.
<b>Positions</b>	Number of positions to be displayed. 0=automatic adjustment.
<b>Maneuverable range</b>	A range of maximum 64 texts can be maneuverable in run-time mode. State the numbers of the first and the last texts of the range. See the section "Maneuvering Textblocks".

---

### Note!

When using indexed message libraries the number of positions should not be 0 since the automatic length adjustment does not work.

---

For more detailed information, see the chapter "Message library".



## Multiple choice object

The object is linked to up to three digital signals which together can assume eight different states. A text of 30 characters can be linked to each state.

**Multiple Choice**

Analog signal:

Text:

0:  ☐ Man.

1:  ☐ Man.

2:  ☐ Man.

3:  ☐ Man.

4:  ☐ Man.

5:  ☐ Man.

6:  ☐ Man.

7:  ☐ Man.

Security Level:

OK Cancel

Parameter	Description
<b>Analog signal</b>	This register decides which text is to be presented. For valid devices see the driver manual.
<b>Text 0 - Text 7</b>	The texts that will be shown for each state of the object.
<b>Man 0 - Man 7</b>	States whether it is possible to maneuver this status in run-time mode. See the section "Maneuvering Textblocks".





## Text object

Object for handling ASCII strings. It is possible to present the text stored in the registers of the controller system. Texts should be in ASCII format. The texts can be altered in run-time mode from the terminal by opening and closing the entry field with [ENTER].

The text is converted from 8-bit IBM extended ASCII to Swedish 7-bits ASCII if SW is stated on the command line in System Signals.

Parameter	Description
<b>Analog signal</b>	The register where text for the first position is stored.
<b>Positions</b>	Each register contains two characters (occupying two positions). The basic setting is 8 positions which is equivalent to four registers.
<b>Maneuverable</b>	Select if the object can be forcibly set in the run-time mode.

## **7.3 Maneuvering Textblocks**

textblocks consist of text lines with static and dynamic objects. The dynamic objects show the current status for those signals to which the object is linked. Certain dynamic objects are maneuverable which means that you can change their status and values in run-time mode.

To alter a maneuverable object, use the arrow keys to move the cursor to the right position so that the object is marked. Textblocks can be scrolled vertically but not horizontally

### **Digital objects**

Digital objects are maneuvered by marking the required object. Then press [ENTER] to alter the status of the object.

### **Analog objects**

#### **Analog and Date/Time**

The objects are maneuvered by placing the cursor on the right object. After this the new value is written in. Finish with [ENTER].

Before [ENTER] is pressed, new value entry can be aborted by leaving the field with [↑] or [↓]. The old value will then be retained.

#### **Text objects**

Maneuver a text object by moving to the object and pressing [ENTER]. An entry field is then shown. The entry field is either on the first or last line depending where the object is on the display. Scrolling of the entry field takes place if the text is longer than the width of the display. Press [ENTER] when writing is complete.

#### **Message objects**

Maneuver a message object by moving to the object with the arrow keys and press [ENTER]. An option list is now shown with all maneuverable states on the display. Move to the required state, press [ENTER] and the analog signal linked to this state is forcibly set.

### **Multichoice objects**

Maneuver a multichoice object by moving to the object with the arrow keys and press [ENTER]. An option list is now shown with all maneuverable states on the display. Move to the required state, press [ENTER] and the analog signal linked to this state is forcibly set.

### **Jump object**

Move to the object and press [ENTER].

### **Bargraph object**

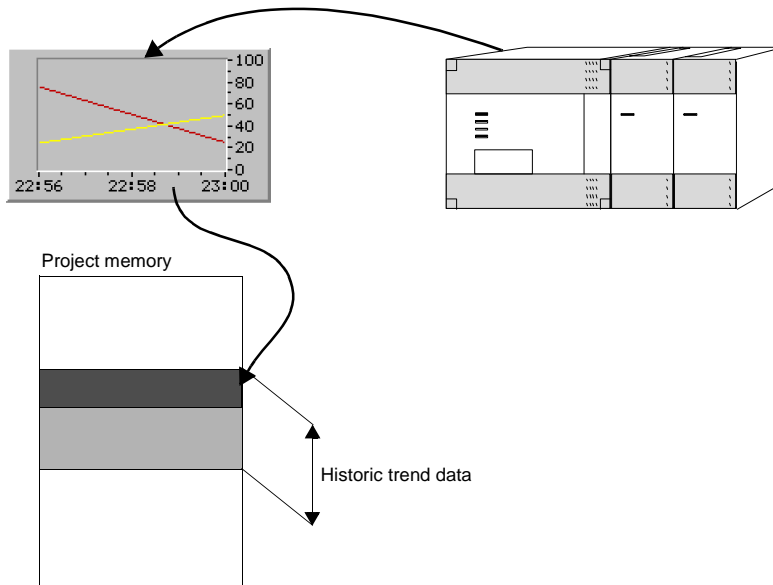
The indicators for min and max value can be set to the actual value for the bargraph object by first selecting the object and then pressing [ENTER].

## 8 Historical trends

This section describes historical trends. This function means that analog values from the PLC system are collected and presented in a trend object while running. The presentation is in the form of a curve. The values collected are stored in the terminal's project memory.

It is possible to define several independent trend curves in the same or different blocks. The total number is limited by the project memory and the RAM of the terminal. The use of other objects and functions add to the total number of trends in a project, for example, the number of alarms.

Such things as the time interval between collection of data and how many values are to be stored are defined in the trend object.



## 8.1 Calculate the trend size

If you want to know how much of the project memory the trend data will occupy the following calculation must be made.

$$S = TOS + AK (28 + (645 * ((AS / 100) + 1)))$$

S=The number of bytes.

TOS=TrendObjectSize, if all parameters for a trend object are changed the TOS will be 320 bytes.

AK=The number of defined curves in the trend object.

AS=The number of Samples is rounded up to the nearest hundred.

---

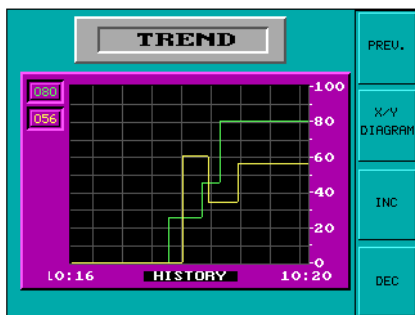
### Note!

Even the RAM of the terminal can cause a limit to the number of trends in a project. The limit is dependent on other objects and functions used in the project.

---

## Presentation in run-time mode

Trend curves can show historic data in the run-time mode. Mark the required trend object and press [ENTER]. A dialog square is now shown and the date for the data to be presented can be chosen. "History" is shown at the bottom of the square. Press [ENTER] again to return to real-time display. Trend data is stored in the files whose names are stated when the trend object is defined.



Trend History			
Left	time	:	10:32:00
Right	time	:	10:36:00
Left	date	:	97-05-21
Right	date	:	97-05-21
OK			

## 8.2 Defining trend objects

Trend objects are defined as other dynamic objects in a block. The object can be linked to one to six analog signals. Unlike other objects, trend objects must be given a name with 1 to 8 characters. The first character must be a letter or a number. The standards for naming files in MS DOS are applied. The following parameters are defined for a trend object. Valid characters are 0-9, A-Z and \_. The standards for naming files in MS DOS are applied. The following parameters are defined for a trend object. Under Terminal options in the Setup menu you can decide if trends are going to be stored when changed or if all samples shall be stored. For more information see “The Setup Menu”.

Parameter	Description
<b>Name</b>	State the name of the object. The name must be unique for each object.
<b>Curve</b>	Select the curve you want to edit (1-6).
<b>Analog signal</b>	The analog signals for which the object shall sample and present the values. Only 16-bit numbers can be used.
<b>Color</b>	Select a color for each curve.
<b>Offset and Gain</b>	Used to scale register value. See the chapter “General”.

Parameter	Description
<b>Enable</b>	The digital signal which when set starts the collection of data. Data collection ceases if the signal is set to zero. The parameter needs not be stated.
<b>Y Min value reg.</b>	The minimum value on the Y-axis loaded from the stated PLC register.
<b>Y Minimum value</b>	The minimum value on the Y-axis. (Ignored if Y Min value reg. is defined).
<b>Y Max value reg.</b>	The maximum value on the Y-axis loaded from the stated PLC register.
<b>Y Maximum value</b>	The maximum value on the Y-axis. (Ignored if Y Max value reg. is defined.)
<b>Y Scale</b>	States which side of the Y-axis shall be shown, to the left, right, both or neither.
<b>Y Scale division</b>	The interval between the scale number on the Y-axis.
<b>Y Scale ticks</b>	The interval between the scale ticks displayed.
<b>X Sample interval</b>	States the time interval between the collection of data. Minimum value is 1 second.
<b>X Stored samples</b>	States how many values shall be stored. The maximum number of values are 13000.
<b>X Time range</b>	States the time interval to be shown in the trend diagram.
<b>X Time scale division</b>	The interval between the scale number on the X-axis.
<b>X Time scale ticks</b>	The interval between the scale ticks displayed.
<b>Sample full limit</b>	State the number of samples when the samples full signal shall be set.
<b>Sample full signal</b>	Select a signal which will be set when the number of samples reaches the value stated in the parameter Sample full limit.
<b>Frame</b>	Select if a frame shall be prited around the object.
<b>BG</b>	Select background color for the object.
<b>Scale</b>	Select the color of the scale in the object.
<b>Erase trend data signal</b>	The digital signal which when set will erase all historic trend data.
<b>Grid</b>	Select if grid shall be displayed in the object.
<b>Grid color</b>	Select a suitable color for the grid.

**Note!**

If you copy a block containing a trend object you must change the name of the trend object. Two trend objects may not have the same name.

## 8.3 Transferring trend data

Trend data and recipes can be transferred to and from a personal computer with the *File Transfer* program (an icon in the program group HMI Tools) for calculating statistics and other types of presentation and storage.

Trend files and production methods can for example be opened directly in Excel for statistical calculations. Values can also be changed and thereafter transferred by copying or saving files in the unit. See the manual for HMI Tools.

### Trend files

The name of the trend files is given for each trend when the trend objects are defined. The file will be allocated the extension .SKV. The format for each line in the trend file is as follows;

DDDD;TTTT;AAAA;BBBB;CCCC;DDDD;EEEE;FFFF:

Format	Description
<b>DDDD</b>	Date format given in Setup.
<b>TTTT</b>	Time format given in Setup.
<b>AAAA</b>	Trend curve 1.
<b>BBBB</b>	Trend curve 2 (if defined).
<b>CCCC</b>	Trend curve 3 (if defined).
<b>DDDD</b>	Trend curve 4 (if defined).
<b>EEEE</b>	Trend curve 5 (if defined).
<b>FFFF</b>	Trend curve 6 (if defined).

The oldest value is shown on the first line of the file and the latest on the last line. The .SKV format can be imported directly to Microsoft Excel. You can use the diagram manager to create statistical diagrams in Excel. It is not possible to alter something in the file and then send it back to the terminal. The picture below shows an example of how it can appear in Excel.



Edit Recipe: TREND1.skv

File Help

1996-01-04    Insert Row    Delete Row    Exit

Device	Value					
1996-01-04	21:08:16	1 6	61	46	31	16
1996-01-04	21:08:18	5 0	55	40	25	10
1996-01-04	21:08:20	7 2	57	42	27	12
1996-01-04	21:08:21	8 3	58	43	28	13
1996-01-04	21:08:22	9 4	59	44	29	14
1996-01-04	21:08:23	0 5	60	45	30	15
1996-01-04	21:08:24	1 6	61	46	31	16
1996-01-04	21:08:25	2 7	62	47	32	17
1996-01-04	21:08:26	3 8	63	48	33	18
1996-01-04	21:08:27	4 9	64	49	34	19
1996-01-04	21:08:28	5 0	65	50	35	20
1996-01-04	21:08:29	6 1	66	51	36	21
1996-01-04	21:08:30	7 2	67	52	37	22
1996-01-04	21:08:31	8 3	68	53	38	23
1996-01-04	21:08:32	9 4	69	54	39	24
1996-01-04	21:08:33	0 5	70	55	40	25
1996-01-04	21:08:34	1 6	71	56	41	26
1996-01-04	21:08:35	2 7	72	57	42	27
1996-01-04	21:08:36	3 8	73	58	43	28
1996-01-04	21:08:37	4 9	74	59	44	29
1996-01-04	21:08:38	5 0	75	60	45	30
1996-01-04	21:08:39	6 1	76	61	46	31
1996-01-04	21:08:40	7 2	77	62	47	32
1996-01-04	21:08:41	8 3	78	63	48	33
1996-01-04	21:08:42	9 4	79	64	49	34
1996-01-04	21:08:43	0 5	80	65	50	35
1996-01-04	21:08:44	1 6	81	66	51	36
1996-01-04	21:08:45	2 7	82	67	52	37

## 9 Message library

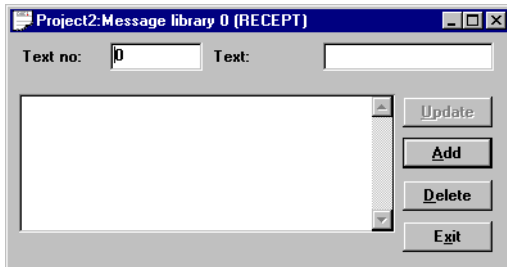
With the message library function you create message tables where values between 0 and 65535 are linked to texts. The function is used for presentation of each step in a sequence control. Another use is presenting error codes. An analog signal generates the error codes which are linked to texts in a message library. The function can also be used for operator communication.

The message library is one or several message tables containing up to 512 text strings each. Each text string can have a length of 40 characters.



Parameter	Description
Library	State a number of the message library.
Description	State a name of the message library.

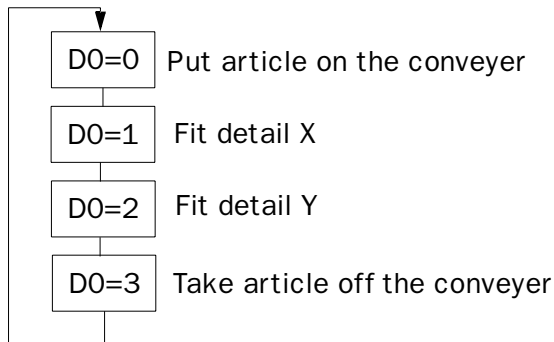
You edit a message library by marking a library and click on Edit. Several editing windows can be displayed at the same time.



Parameter	Description
<b>Text no</b>	State a number of the text, 0-65535.
<b>Text</b>	Freely defined text which is displayed when current signal includes the text number for the text.

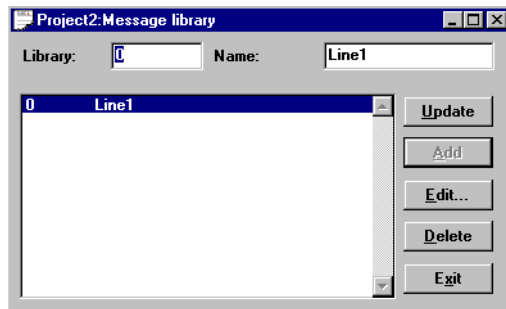
## Example

The following example shows how the function may be used. In a sequence control we want to display a text for each step in the sequence.



Start by creating a message library named Line1.

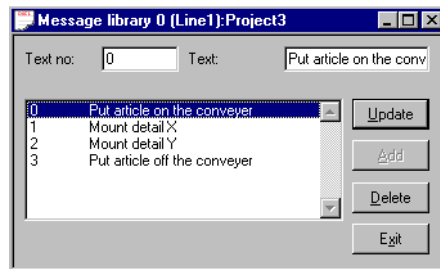
1. Select **Message library** in the **View** menu.
2. Define a number, in this case 0, and a name of the library, in this case Line1.
3. Click on Add.



A message library named Line1 has now been created. Thereafter you define the texts in the library.

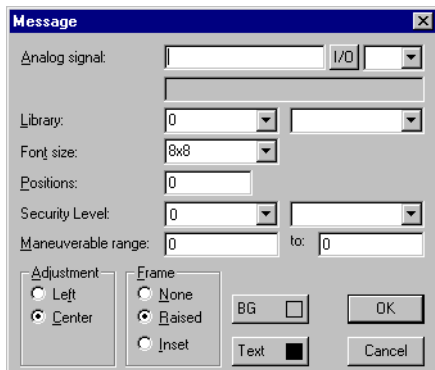
4. Mark the library, line1, and click on Edit.

In this dialog you define text number and text. In the field **Text no** you state a value of the analog signal linked to the message object. In the field **Text** state the text displayed in the message object.



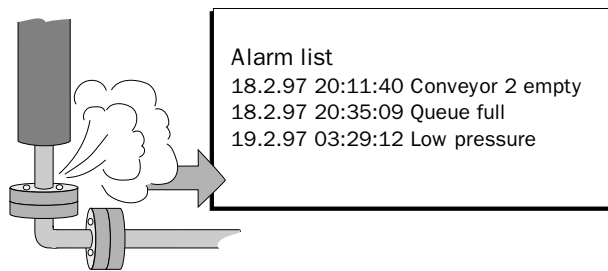
When the message library is defined, you create a message object in the project. Message objects can be defined in both text and graphic blocks.

5. Select the **Message object** in the toolbox. Place the cursor where you want to place the object and click.
6. State the analog signal to control which text is to be displayed.
7. In the field library you select message library.
8. Select if the object should be maneuverable and between which texts it may be changed.



## 10 Alarm handling

This section describes alarm management. The function is used to call the operators attention to events in the process that need immediate attention. This function is divided into alarm groups, alarm messages and an alarm list. The alarms can be divided into groups, for example, to create an order of priorities. The alarm messages are definitions of the signals that will activate an alarm as well as the alarm message for each signal. The alarm list contains alarms caused in the run-time mode.

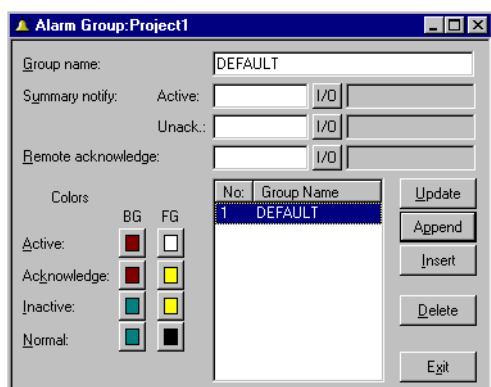


## 10.1 Alarm grouping

Alarms in E900 can be divided into 16 groups and i E910 into 7-11 groups depending on selected font size. Each group can be allocated a certain color. Alarms can be sorted into groups in the alarm block. It is not necessary to define the alarm groups.

### Defining alarm groups

Alarm groups are defined under **Alarm groups** in the **Functions** menu. The attribute for alarms in this group are decided in the following dialog box.



Parameter	Description
<b>Group name</b>	Any name for the alarm group.
<b>Summary notify</b>	Active - Digital signal which is set to unity when there is an active alarm in the group. Unack - Digital signal which is set to unity when there is an unacknowledged alarm in the group.
<b>Remote acknowledge</b>	Digital signal which acknowledges all the alarms in the group simultaneously.
<b>Colors</b>	States the colors for active, acknowledged, inactive alarms as well as the normal state for the alarm.

## 10.2 Alarm messages

Alarm messages are defined under **Alarms** in the **Functions** menu. The alarm message is entered here with a maximum of 60 characters. More characters can be shown if a smaller font size is used in the alarm list. The message can contain dynamic data (just like a Textblock) which can be digital or analog. You can define 100-300 alarms depending on the application.

The screenshot shows a software window titled "Alarms:Project3". It contains several input fields and checkboxes for configuring an alarm. The fields include "Alarm text", "Signal" (with an "I/O" indicator), "Alarm when signal is" (with radio buttons for "On" and "Off"), "Acknowledge notify" (with an "I/O" indicator and a "Reset" checkbox), "Remote acknowledge" (with an "I/O" indicator), "Alarm group" (a dropdown menu), and "Info block" (two dropdown menus). There are also three checkboxes: "Ack. required", "History", and "To printer". At the bottom, there is a table with three columns: "No.", "Signal", and "Alarm Text". To the right of the table are five buttons: "Update", "Append", "Insert", "Delete", and "Exit".

Parameter	Description
<b>Alarm text</b>	An alarm text.
<b>Signal</b>	The digital signal which when activated generates the alarm is stated here.
<b>Alarm when signal is</b>	Select On if a set signal should cause an alarm. Select Off if a reset signal should cause an alarm.
<b>Acknowledge notify</b>	Digital signal affected when the alarm is acknowledged. Normally the signal is set.
<b>Reset</b>	The above signal is set to zero when the alarm is acknowledged if Reset is marked.
<b>Remote acknowledge</b>	Digital signal which acknowledges the alarm when activated.
<b>Alarm group</b>	States the alarm group for the definition.



Parameter	Description
<b>Info block</b>	State a block number of a text or graphic block. This makes it possible to give the operator a “help page” containing information about the alarm and possible actions. 0 means that no blocks are linked to the alarm. If the information block is a textblock it will be included if the alarm is sent as an email message. See the section “Alarms in run-time mode”.
<b>Mail to address</b>	When an alarm is activated an email can be sent to a predefined receiver. The alarm text is then the subject. See the section “SMTP Client” in the chapter “Network functions in the terminal”.
<b>Ack. required</b>	States if the alarm should be acknowledged or not. A cross in the square means that the alarm must be acknowledged and no cross means that the alarm functions as an indicator alarm, that is as a form of information.
<b>History</b>	States when the alarm shall be taken out of the alarm list. A cross means that the alarm remains in the list until the alarm list is full. No cross means that the alarm is taken out of the list when it is acknowledged and is no longer active. If the <b>Acknowledge</b> parameter is not marked then the alarm is taken out of the list as soon as it is not active.
<b>To printer</b>	Whether the alarm message should be printed out directly if the alarm status changes is stated here.

## 10.3 Configuring the alarm list

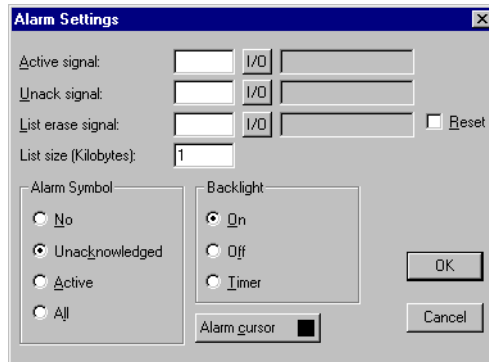
General settings for the alarm list are made in **Alarm settings** in the **Setup** menu. Alarms take different amounts of space in the alarm list depending on the length of the alarm text and the number of objects in the alarm message.

$$S = 19 * AO + NC$$

S = The number of bytes.


AO = The number of objects in the alarm message.

NC = The number of characters in the alarm message.



The 'Alarm Settings' dialog box contains the following controls:

- Active signal:** A text input field followed by an 'I/O' button and another text input field.
- Unack signal:** A text input field followed by an 'I/O' button and another text input field.
- List erase signal:** A text input field followed by an 'I/O' button and another text input field, with a ☐ **Reset** checkbox to its right.
- List size (Kilobytes):** A text input field containing the value '1'.
- Alarm Symbol:** A group box containing four radio buttons: **No**, **Unacknowledged** (selected), **Active**, and **All**.
- Backlight:** A group box containing three radio buttons: **On** (selected), **Off**, and **Timer**.
- Alarm cursor:** A checkbox with a black square icon next to it.
- Buttons:** **OK** and **Cancel** buttons are located at the bottom right.

Parameter	Description
<b>Active signal</b>	States the digital signal which the terminal set when there is an active alarm.
<b>Unack signal</b>	States the digital signal which the terminal set when there is an unacknowledged alarm.
<b>List erase signal</b>	States the digital signal which when set erases all not active alarms in the alarm list.
<b>Reset</b>	Means that the signal stated on List erase is set to zero when the alarm list is erased.
<b>List size</b>	States the list size in number of kbytes. Note that the real size of the alarm list is twice the stated size. The performance of the project is effected negatively if list sizes over 10 kbytes is used.
<b>Alarm symbol</b>	States when the alarm symbol should be shown, see table below. <b>■ALARM■</b> is shown in the Textblock and a red bell,  , in the top right-hand corner of the display is shown in the Graphic block.
<b>Backlight action</b>	State when the background light shall be turned on for an alarm. <b>On</b> means that the backlight is turned on when the alarm symbol is shown. Default setting. <b>Off</b> means that the backlight is not affected by alarms. <b>Timer</b> means that the backlight is on when a new alarm is activated and turns off when the time for the screen save is out.
<b>Alarm cursor</b>	In the E900 and E910 the cursor color can be changed in the alarm list.

Option	Alarm symbol shows when	
	active	unacknowledged
NO		
UNACK		•
ACTIVE	•	
ALL	•	•

## 10.4 Alarms in run-time mode

Alarms are shown in the Textblock with the text **■ALARM■** in the upper right-hand corner of the display (you can select not to have this during configuration). A red bell blinks in the upper right-hand corner of the display for alarms in Graphic blocks.

The alarm is presented in an alarm list with the predefined alarm texts. The alarm list contains the latest alarms which are listed by alarm group according to definition and with the latest alarm at the top of the list. The alarm list's size in number of kbytes is selected during configuration. The alarm list is shown on jumping to the alarm block. The status for the alarm is shown for each alarm as well as the time when the alarm was activated, deactivated or acknowledged depending on the display mode selected.

Alarms can have the following status.

Symbol	Status
*	Active, Not acknowledged
\$	Not active, Not acknowledged
-	Active, Acknowledged
<blank>	Not active, Acknowledged

Alarms can be shown in the following format.

Display format	Description
<b>S</b>	The time when the alarm was activated
<b>E</b>	The time when the alarm was inactivated
<b>A</b>	The time when the alarm was acknowledged

To get to the alarm block you can either press [LIST], define a jump to system block 990 or allow the controller system to produce a list via the Display signal for block 990.

To acknowledge an alarm place the cursor on the line where the alarm is and press [ACK]. Alarms can be printed out directly as they occur or change status if a printer is connected. This should be stated when the alarm is defined. The alarm is printed out with date, time, status and the alarm text.

Return to the previous block by pressing [LIST] on the terminal.

The current content of the alarm list can be printed out by giving the Printer signal for block 990.

### **Block linked to alarms**

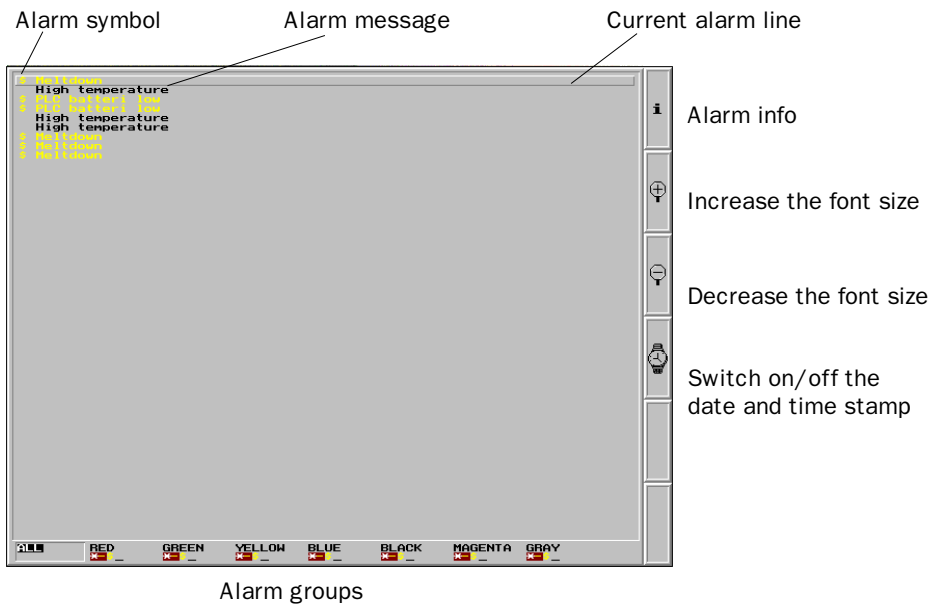
Text or graphic blocks can be linked to alarms. This means that when the operator presses the INFO key at an alarm in the alarm list, a block linked to the alarm is shown. This block can contain information about the alarm and recommended actions. It is only possible to press the INFO key if current alarm is linked to a block. Return to the alarm list by pressing [PREV] on the terminal.

## Graphic alarm page in run-time mode

The page is graphic and can be modified by the user. The function keys can be linked to the functions in order to enlarge or reduce the text on the alarm page as well as the function for flipping through the pages. Furthermore there is the possibility to select the display of date and time.

Alarms can be sorted into groups and you can select which group is to be shown. Status is shown with different colors, the color defined when setting the alarm group. You select alarm group with the left and right arrow keys.

### E900

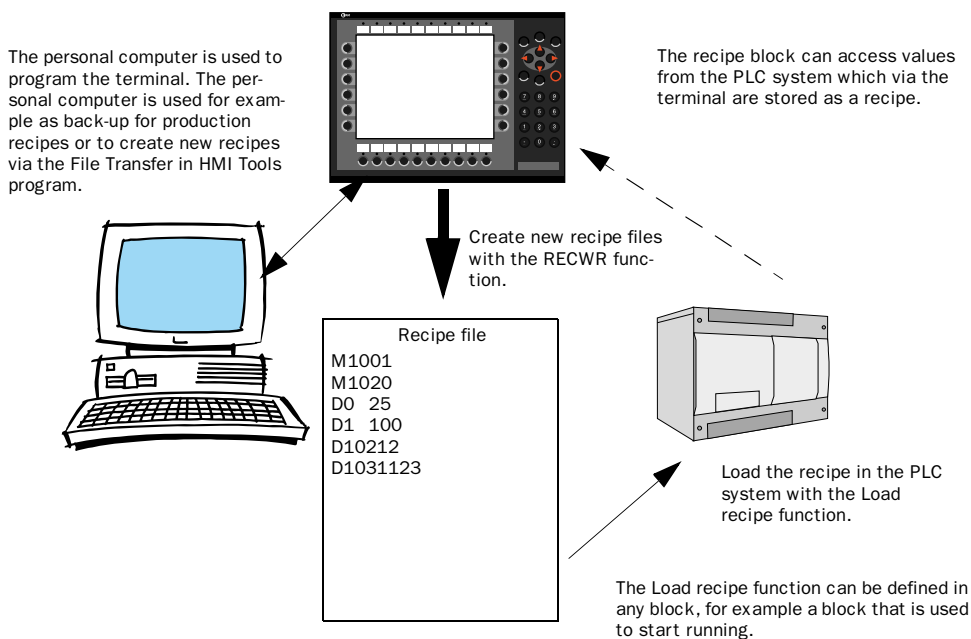


# 11 Recipes

This chapter describes the function recipe handling. The recipes function means that all dynamic data in a block, that is signals and their values, can be saved in the file in the run-time mode. The operator can then download the file to the PLC system. The PLC system then starts to work with the loaded values. By using the recipe handling function, a large number of parameter settings can easily be reused. The user creates a recipe library containing files with different parameter settings. This function makes time-critical production more efficient. An example is when you often shift between similar products such as when producing a unit in different colors. The recipe files can be created from the terminal or from the PLC system or from a PC with the HMI Tools software.

Recipe files are stored in the terminal. To use recipes, the functions for storing, loading and erasing recipes must be linked to function keys, see the chapter “Function keys”.

The diagram below shows the principle for managing recipes.



## 11.1 Calculate the recipe size

Use the following equation to calculate how much of the project memory the recipe will occupy.

$$S = 90 + n (2 * IOG_i + 28)$$

S = number of bytes. If S calculated will be less than 360, S will be set to 360.

n = the number of IO series

$IOG_i$  = the number of IO in each IO series

### Example

Our recipe consists of three I/O series, D0-D109 (110), D200-D499 (300) and M0-M99 (100).

The calculation will be:

Calculate the sums of the different series:

$$1104 = (2 * 110 + 28) + (2 * 300 + 28) + (2 * 100 + 28)$$

Then calculate the sum:

$$S = 90 + 1104$$

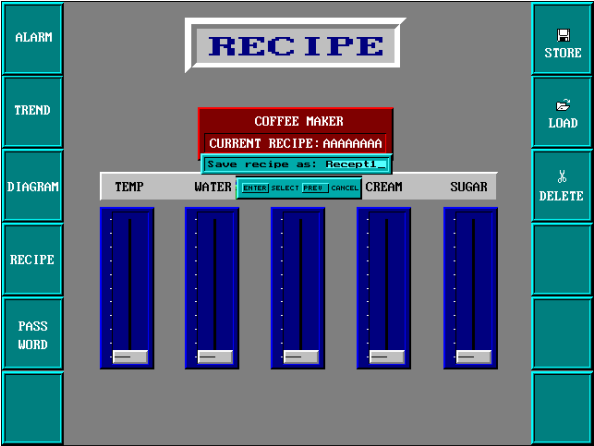
$$S = 1194 \text{ bytes}$$

## 11.2 Creating a recipe with the terminal

Which block or blocks are to be used to save the recipe is decided when the application is programmed. All those signals that are to be included in the recipe are defined in the block that will be saved as a recipe. All dynamic values in the block are stored in the recipe file. All digital and all analog objects except trend objects can be used as recipe parameters.

A jump is made in run-time mode to the block which contains the recipe parameters. Enter the required values in the dynamic object and press the function key that is linked to RECWR. The name of the file is given in the entry field shown. The recipe name may not contain national characters. The recipe file is stored in the terminal.

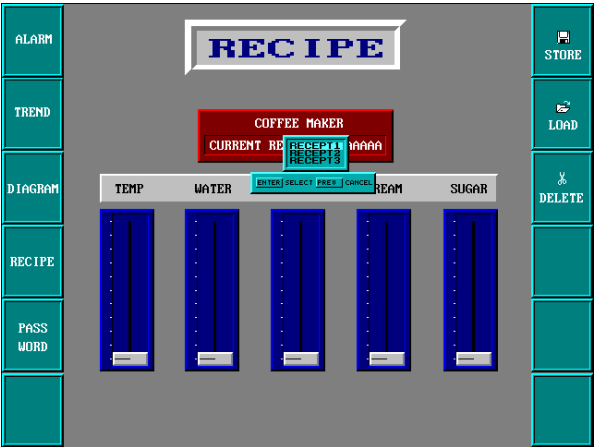
E900



### 11.3 Transferring a recipe to the PLC system

The recipe is transferred to the PLC system in run-time mode with the RECRD function. This means that the signals and their values stored in the file are transferred to the PLC system. When the function key for RECRD is pressed a list of available recipe files to choose from is shown. Select one and press [ENTER]. The PLC system will then run with the downloaded values.

E900





## 11.4 Erasing a recipe

Any named recipe can be erased in run-time mode from the terminal memory with the RECDEL function. Press the function key connected to RECDEL. Available recipe files to choose from are now shown. Select the file to be erased and press [ENTER]. Confirm the erasure by pressing [ENTER] or abort by pressing [PREV].

## 11.5 Using recipes in a personal computer

The terminal is treated as a unit in the personal computer with the Transfer program (in the HMI Tools program group). This means that the personal computer can be used for making safety copies of files in the terminal for example recipe files. New recipes can be created in the personal computer and be copied to the terminal.

The files are stored in the .SKV format in the personal computer which can be brought up in Excel. Files can be edited in Excel and then used in the plant again.

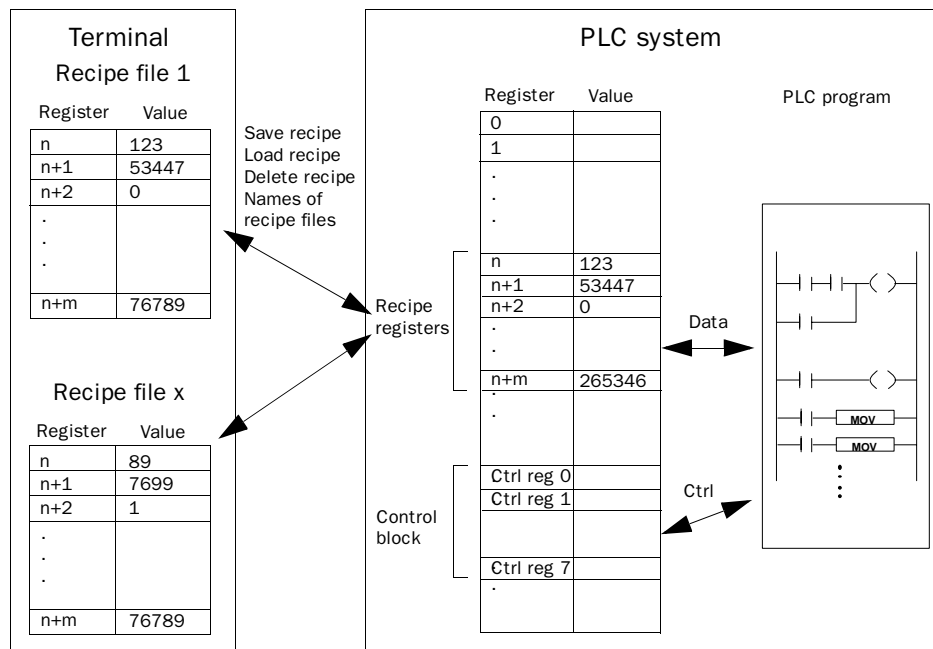
The file must end with the instruction END. See the example below.

```
M100    ;1
M102    ;0
D0       ;25
D50      ;12
END
```

For further information we refer to the manual for HMI Tools.

## 11.6 Loading and saving recipes via the PLC program

By using a control block in the PLC system you can load, save and delete recipe files in the terminal. The files created by the PLC program are compatible with the recipe files saved in the terminal. This means that recipe files saved by the PLC program can be loaded in the terminal and vice versa. The control block is described in the following table.



In the **Setup** menu, **System signals** you define the first register in the control register area. This register and the six following registers are used as a control register.

Register	Content	Description
<b>Dn0</b>	Command	Command register set by the PLC Available commands: 0: No command 1: Save recipe to the terminal. 2: Read recipe from the terminal. 3: Delete recipe from the terminal.
<b>Dn1</b>	Result code	Handshake register set by the terminal. 0: Ready for new command. 1: OK. 2: Recipe file write error. 3.: Recipe file not found.
<b>Dn2</b>	File name char. 1-2	Recipe file name in the terminal.
<b>Dn3</b>	File name char. 3-4	
<b>Dn4</b>	File name char. 5-6	
<b>Dn5</b>	File name char.7-8	
<b>Dn6</b>	Start data register	The first data register to be read/written to/from recipe file.
<b>Dn7</b>	Number of registers	Number of registers to read/write to/from recipe file.

To perform commands, the following handshake procedure must be carried out:

- 10.The handshake register must be 0. If it is not, make sure that the command register is set to 0.
11. Write the command in the command register.
- 12.Wait for completion or error code in the handshake register.
- 13.Set the command register to 0. After this, the terminal will set the handshake register to 0.

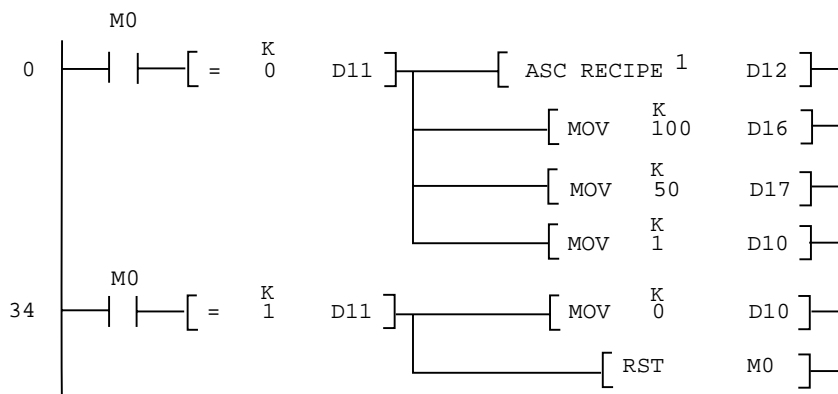
### Recipe names

The following characters can not be used in recipe names:

! ? < > ( ) + / \* = ° % # : . [space] , -

### Example

The PLC wants to save D100 to D149 as recipe name “RECIPE1” when M0 is activated. The control block starts at D10. The PLC program should be as follows.



### Limitations

Recipe created in the PLC system can include maximum 1000 registers. Only data registers can be used.



## 12 Passwords

This section describes how to use security levels and passwords in the terminal. This function makes it possible to create a security system for the terminal. The operators can be given different accessibility to the system.

Blocks, function keys and maneuverable objects can be allocated a security level. Each security level is given a password. The user must log in with a password for the security level concerned or a higher one to gain access to the various levels. It is not essential to use this function.

### 12.1 Defining security levels

Blocks, function keys and maneuverable objects can, when they are defined, be allocated a security level. A security level (0 to 8) is given in the dialog box for the object. Everyone will have access if the 0 security level is used.

## 12.2 Defining passwords

The password is defined for security levels 1 to 8 under **Passwords** in the **Functions** menu.

Security level:	Password	Confirm question	Comment
1:			
2:			
3:			
4:			
5:			
6:			
7:			
8:			

Login signal:  I/O

Logout signal:  I/O

Current level reg:  I/O

Login timeout:  0 minutes

Password run/prog:

OK Cancel

Parameter	Description
<b>Password sec level 1-8</b>	Enter the password and any comments for security levels 1-8.
<b>Confirm question</b>	Enter a question that should be confirmed by the operator before maneuvering is enabled. The function does not work when a function key or touch key is defined with a password.
<b>Login signal</b>	State here the digital signal which when set generates the entry field for logging in. The entry field for logging in can also be linked to a function key, see the "Function keys" chapter.
<b>Logout signal</b>	State here a signal which when set logs out the current user. This function can also be linked to a function key, see the "Function keys" chapter.
<b>Current level reg</b>	State a register where the terminal put the current password level (0-8).

Parameter	Description
<b>Login timeout</b>	State here the time in minutes during which the terminal can remain unused before the user is automatically logged out. If 0 is used no logging out takes place.
<b>Password run→ prog</b>	Here you can state a password which must be entered when switching from RUN to PROG manually. The function is not used when switching from PROG to RUN or when Auto RUN/TRANSFER switching from the MAC Programmer+ is used.

## 12.3 Logging in

Logging in is either controlled from a function key or through a digital signal from the PLC system. The entry field for logging in is shown if the function key linked to the function Login to specified security level is pressed or the digital signal is activated. State the password here. The password is linked to the security level, see above. If the security level is changed to a higher level the [PREV] key will be disabled.

## 12.4 Password for transfer

On the command line under System signals in the Setup menu you can state the command PDxxxxxxx where xxxxxxxx is the password. The operator then have to enter a password to have access to the functions in the Transfer menu.

## 12.5 Password for all terminal functions

On the command line under System signals in the Setup menu you can state the command PSxxxxxxx where xxxxxxxx is the password. When the operator enter the password he has access to all functions in the terminal, even the functions protected with other passwords.



## 12.6 Altering passwords in run-time mode

Passwords can be altered in run-time mode with the Change login password function. When the function key linked to Change login password is pressed a dialog is shown where the passwords for each security level can be altered.

---

**Note!**

It is not possible to state the security level for block 0.

---

---

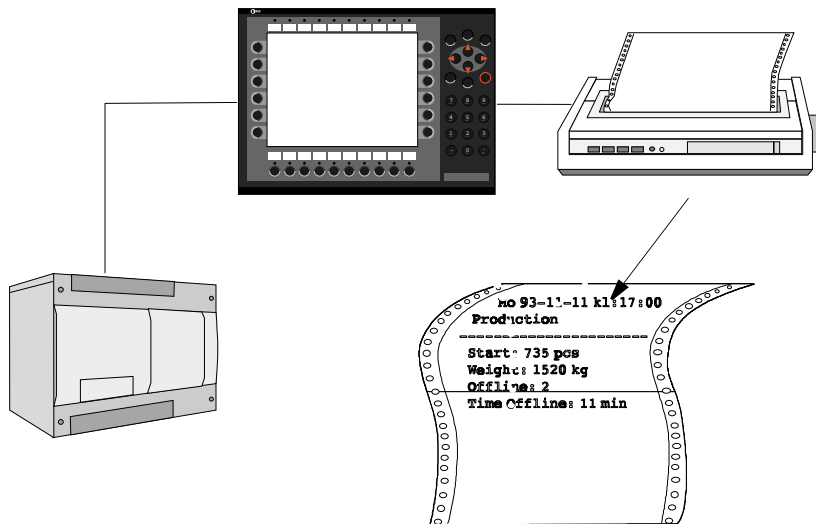
**Note!**

After logging out the [PREV] key and the “Return to previous block” function for the function keys is ignored to provide unauthorized to enter block that have password.

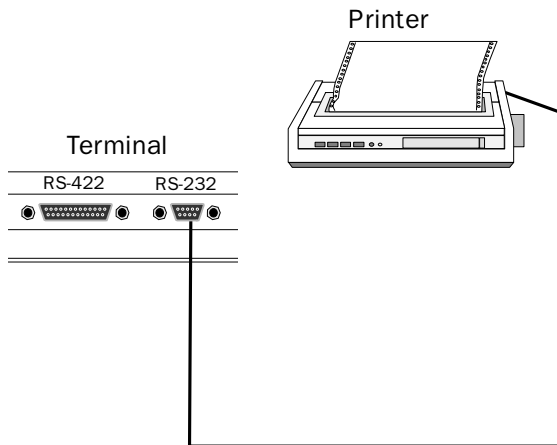
---

## 13 Report print-outs

This section describes how to make a report print-out from the terminal. For production follow-up it is simple to create different types of reports such as daily reports and event reports. The illustration below shows the principle for generating a daily report.



## 13.1 Connection to a printer



The printer should have a serial interface and be equipped with IBM character set (850). The settings for the printer are set under **Peripherals** in the **Setup** menu in the terminal. Refer to the printer manual for the correct configuration.

---

**Note!**

A printer with parallel interface can be connected if the IFC PI expansion card is used.

---

## 13.2 Text block printouts

The reports are created as Text blocks with static and dynamic text. The report has a maximum width of 150 characters. In the text block you enter a free text, for example, table header and other static text which will always be printed. To get dynamic values from the process in the report you define dynamic objects, presenting the value for the signal the object is linked to. See the chapter “Text-based presentation and maneuvering”. Via time channels it is possible to decide when the report is to be printed.

## 13.3 Graphic block printout

In the E900 and E910 it is possible to print graphic blocks to a Epson FX-80 compatible printer. It is also possible to use a HP PCL5/PCL5E/PCL6 compatible printer to print graphic blocks. Graphic blocks can only be printed when it is displayed (screen dump). Only black and white graphics are supported.

If the **NHD** command is entered on the command line under **System signals** in the **Setup** menu you will get a printout of graphic blocks without header (containing block name, block number, date and time).

---

**Note!**

Epson FX-80 graphic do not support gray scales.

---

---

**Note!**

The printer needs at least 5 MB internal memory to print graphic blocks.

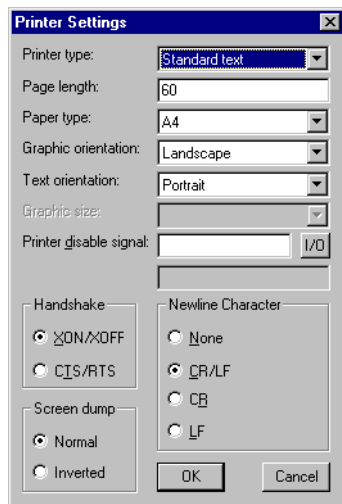
---

## 13.4 Defining the print-out

The **Print signal** parameter in the block header gives the signal which when set activates the print-out. A **Completion signal** is also given here, which is set by the terminal when the print-out is completed. This signal is instead set to zero if **Reset** is marked.

## 13.5 Printer settings

The settings for the printer must be made under **Peripherals** in the **Setup** menu. Under Peripherals you select **Printer** and then press **Edit....** For more detailed information concerning the printer refer to the manual for the printer used. If you select Standard text the printer must support IBM extended ASCII character set. If graphic blocks are printed, the printer must support Epson FX-80 or HP PCL graphics. If you use the IFC PI expansion card it is possible to print on a HP writer.



The image shows a 'Printer Settings' dialog box with the following fields and options:

- Printer type:** Standard text (dropdown)
- Page length:** 60 (text input)
- Paper type:** A4 (dropdown)
- Graphic orientation:** Landscape (dropdown)
- Text orientation:** Portrait (dropdown)
- Graphic size:** (dropdown)
- Printer disable signal:** I/O (text input)
- Handshake:**
  - ☒ XON/XOFF
  - ☐ CTS/RTS
- Screen dump:**
  - ☒ Normal
  - ☐ Inverted
- Newline Character:**
  - ☐ None
  - ☒ CR/LF
  - ☐ CR
  - ☐ LF

Buttons: OK, Cancel

Parameter	Description
<b>Printer type</b>	Select printer, none or installed printer, e.g. HP PCL5.
<b>Page length</b>	The number of lines to be printed out before form feed. Form feed never takes place if the page length is set to 0.
<b>Paper type</b>	Select paper type according to installed printer.
<b>Graphic orientation</b>	States if the graphic printout should be portrait or landscape if a PCL5 compatible printer is used.
<b>Text orientation</b>	States if the report printout should be portrait or landscape if a PCL5 compatible printer is used.
<b>Printer disable signal</b>	State a digital signal which when set ends the printout. The port where the printer is connected will then be free to use for e.g. transparent mode.
<b>New line character</b>	End of line character; CR/LF, CR, LF or none.
<b>Handshake</b>	States whether handshaking between the printer and terminal will take place with XON/XOFF or CTS/RTS.
<b>Screen dump</b>	Possibility to printout the screen, normal or inverted.

## 13.6 Setting a communication port

Settings for the communication port must be made under **Peripherals**. Under Peripherals you select **RS-232** or **RS-422** and then press the **Edit...** button. See the manual for the printer concerned for the correct setting.



Parameter	Description
<b>Baudrate</b>	Select the transfer rate for communication. Must be the same as the external unit.
<b>Parity</b>	Select parity. Must be the same as the external unit.
<b>Databits</b>	Select number of data bits. Must be the same as the external unit.
<b>Stopbits</b>	Select number of stop bits. Must be the same as the external unit.

## 13.7 Control codes to printer

It is possible to type control codes to a printer in textblocks. You type %% followed by a number between 1 and 31, which are control codes to the printer. For example %%12, which is paper feed, for further description of control codes we refer to the manual for the printer used.

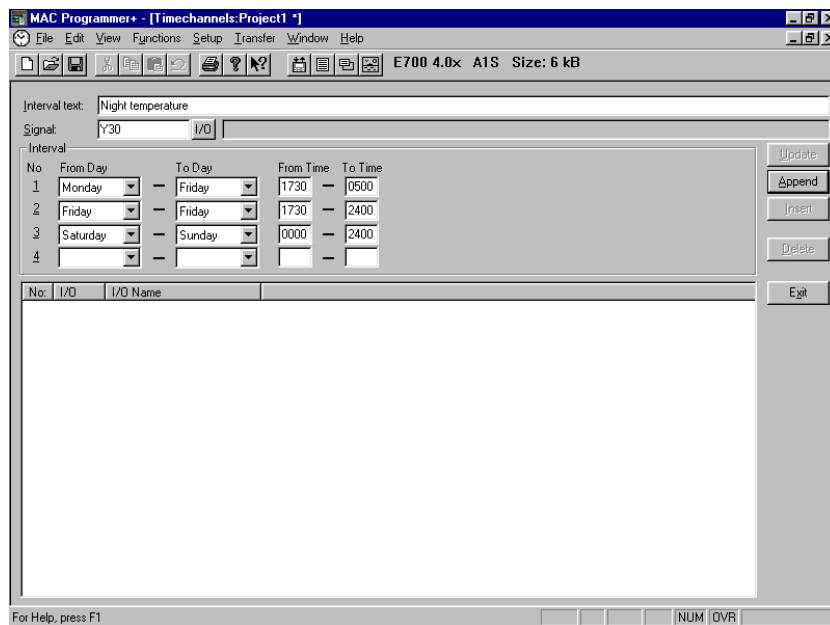
One command must be followed by a blank character. If you have set a transverse (paper) feed this must be at the end of the line. The characters %% must not occur in the text. It is possible to have more than one command/control code on one line.

## 14 Time control

Time control is a function for setting and resetting digital signals relative to the real-time clock. This function is used for controlling events in the process at specific times through the terminal. The time channels replace the time relays and the 7-day clock.

### 14.1 Defining the time channels

The time channels are defined under **Time channels** in the **Functions** menu.



Parameter	Description
<b>Interval name</b>	Enter any text for the time channel.
<b>Signal</b>	A digital signal which is set during the stated time period is stated here.
<b>Interval</b>	The day and time for the interval is stated here. Four intervals can be defined for each time channel.



In the figure on previous page we have defined one time channel. The time channel has the text Night temperature and is linked to the signal Y30. The signal will be set Monday to Friday 17.30 to 05.00, Friday 17.30 to 24.00 and Saturday to Sunday between 00.00 and 24.00. During these periods the temperature in the building will be decreased.

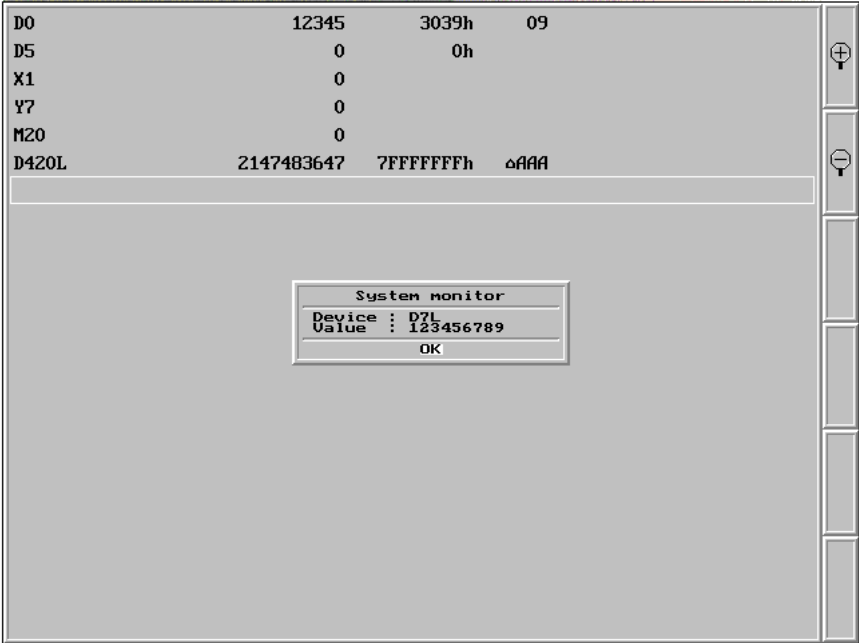
## **14.2 Presentation in run-time mode**

The page with the time channels is shown when system block 991 is activated either through a jump object or a digital signal linked to the block. The time channel's values can be read and/or changed. The option Time channels in the Online settings menu must be set to YES in order to alter the time channel's value in run-time mode.

To read or change values for a time channel you move the cursor to required line and press [ENTER]. Press [DONE] to exit from the time channel's definition. Exit from the time channel menu with [PREV]. The block from which the time channel block was activated is then shown.

# 15 System monitor

The system monitor is a block where the operator can view or change the value for controller signals in run-time. The value is presented as a decimal, hexadecimal and ASCII value.





## 16 Index addressing

Without index addressing, an object will always be tied to the same register. Only the value in that register can be displayed in the object.

With index addressing, however, it is possible during operation to select the register which is to be addressed, and thereby the value to be displayed. This is done by adding the value in the index register to the address of that register which represents the analog signal in the object. This can generally be formulated as follows:

*Displayed value = contents of Register(object address + contents of Index Register)*

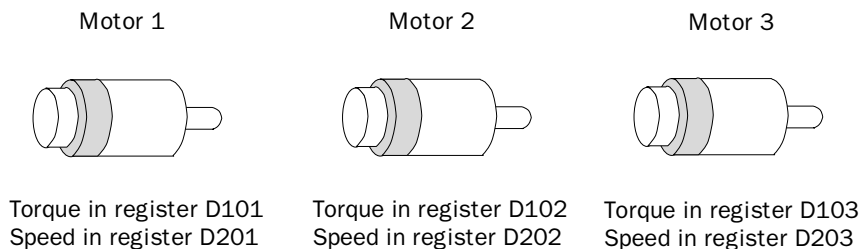
If the contents of the Index Register is 2 and the address of the register specified in the object is 100, the value shown in the object will be retrieved from register 102. If the value in the Index Register changes to 3, the object's value will be retrieved from register 103 instead.

For each project you define which registers are to be index registers. Do this under **Index Registers** in the **Setup** menu. Up to eight different index registers can be used in each project and each index register can be used in more than one object.

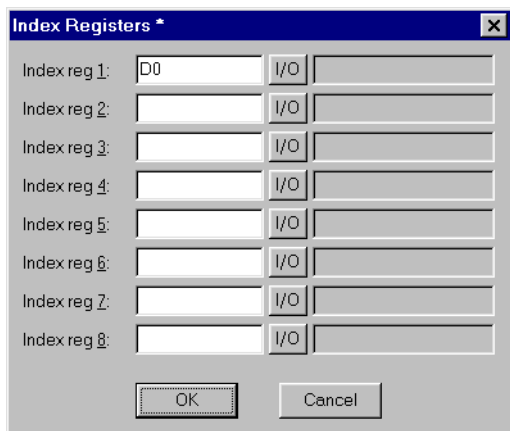
The object used in the project will specify if index addressing is to be used, and the relevant index register. This is done when the object dialog selects I1 to I8 in the entry field to the right of the specified analog signal for the object. In the example below, D100 is the analog signal, and I1 is the relevant index register.



The following example shows three motors which are controlled by a block. The correct torque and speed for the motors are stored in six different registers. In one block, it will be possible to select one of the motors and the correct torque and speed for that particular motor should thus be shown in the block. If another motor is chosen, its correct torque and speed should be selected and displayed. This functionality is possible with index addressing.



Register D0 is defined as Index register 1 at **Index Registers** in the **Setup** menu. The value in this register determines the motor for which the torque and speed are displayed.



If the value in D0 is 1, torque and speed are shown for Motor 1. If the value is 2, torque and speed are shown for Motor 2, and so on. The value in register D0 is controlled by a multiple choice object where the texts, Motor 1, Motor 2 and Motor 3 are specified. These three choices are also made maneuverable.

**Multiple Choice \***

Analog signal: D0 I/O

Text:

0:  ☐ Man.

1: Motor 1 ☒ Man.

2: Motor 2 ☒ Man.

3: Motor 3 ☒ Man.

4:  ☐ Man.

5:  ☐ Man.

6:  ☐ Man.

7:  ☐ Man.

Adjustment

☐ Left

☒ Center

Frame

☐ None

☒ Raised

☐ Inset

BG ☐

Text ☐

Font size: 8x8

Security Level: 0

OK Cancel

Torque and speed are shown in two numerical objects. In the object for torque, D100 is specified as the analog signal, and I1 as the index register.

**Analog Numeric \***

Analog signal: D100 I/O Signed 16-bit

Positions: 5

☐ Maneuverable

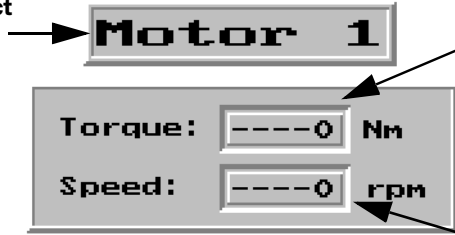
Adjustment

In the object for speed, D200 is specified as the analog signal, and I1 as the index register.

With the maneuverable multiple choice object, it becomes possible during operation to choose Motor 1, 2 or 3. Depending on the choice, a value of 1, 2 or 3 is marked in register D0. This value is added to the addresses for the objects that show torque and speed. These will thus display the values in registers D101, D102 or D103, and D201, D202 or D203 respectively.

**Multiple choice object**

Analog signal D0



**Numeric object**

Analog signal D100  
Index register D0

**Numeric object**

Analog signal D200  
Index register D0

---

**Note!**

Suffixes other than index register can be specified. The index register will not be counted twice if a 32-bit register is used.

---

## 17 Communication

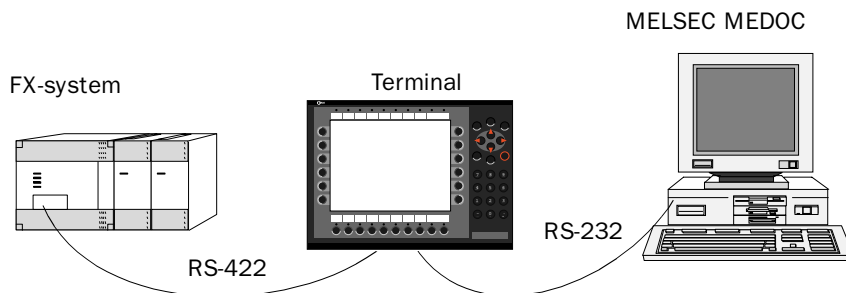
### 17.1 Transparent mode

The programming/printer port on the terminal is used in the transparent mode to connect a unit working in parallel with the PLC system. This unit can be another terminal or a personal computer with, for example, the programming tool for the controller system or a superior operating system.

The transparent mode works together with the MELSEC CPU port and MELSEC A-series communication ports with C24 protocol. For further information see the C24 manual.

#### Connecting a personal computer or other computer system

Personal computers with the MELSEC MEDOC programming tool or other computer system are connected directly to the terminal's programming/printer port.



*Terminal with MELSEC MEDOC connected to the programming/printer port*

#### Settings in the terminal and MELSEC MEDOC

The following settings must be made in the terminal and MELSEC MEDOC for the transparent mode to work.



## The terminal

The printer parameters are set in the **Setup** menu under the menu option **Peripherals**.

The parameters should be 9600 baud, 7 data bits, 1 stop bit and even parity if the terminal is connected to an FX system.

The parameters should be 9600 baud, 8 data bits, 1 stop bit and odd parity if the terminal is connected to a A-CPU or a QnA-CPU.

The parameters should be 19,200 baud, 8 data bits, 1 stop bit and no parity if the terminal is connected to one of the A-series or QnA-series serial communication modules.

## MELSEC MEDOC

The transfer parameters should be changed to 19,200 baud, 8 data bits, 1 stop bit and no parity when transferring via the A-series communication ports. This is done under **Transfer** in **Setup**. The Interface should also be changed from CPU to C24 under Transfer.

The transfer parameters should not be changed in MELSEC MEDOC for other projects. See the Installation manual for cable connections.

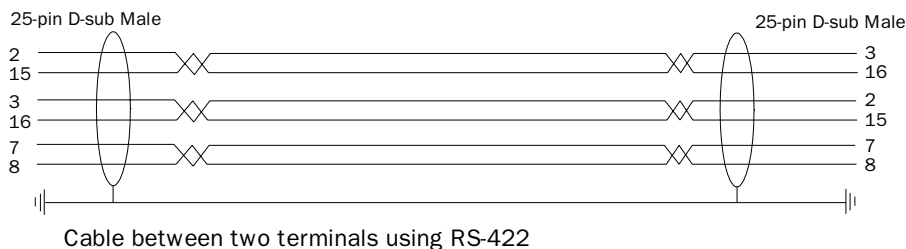
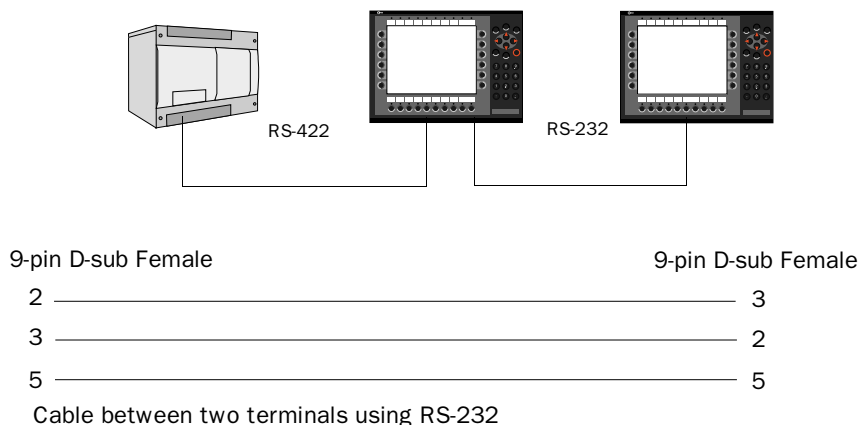
The settings also apply for other PC-based monitoring systems such as SCADA and MELSEC MEDOC *plus*.

## Connecting two terminals in transparent mode

Several terminals can be connected to the same PLC system with the transparent mode function. In this section we describe how to connect two terminals. Several terminals are connected in an analog way.

### Cable connection

When connecting two terminals to a PLC system the first terminal is connected according to the description in the Installation manual. Between the two terminals, a cable is used according to the figures below. The cable is connected between the free port on the first terminal and the corresponding port on the second terminal. If the distance is more than 15 meters using RS-232 communication a signal amplifier must be used.



### Settings in the first terminal

In MAC Programmer+ the communication settings is made in the **Setup** menu under **Peripherals**. Settings for the port connected to the PLC system is stated in the usual way. Settings for the port connected to the other terminal can be freely chosen.

### Settings in the second terminal

In MAC Programmer+ the communication settings are made in the **Setup** menu under **Peripherals**. The PLC system should be placed on the port that is used to connect the second terminal to the first terminal. The settings for this port shall be the same as stated in the first terminal on the port where the second terminal is connected. If any of the FX2N or QnA PLC systems are used the BSD command must be stated on the command line under **System Signals** in the **Setup** menu.

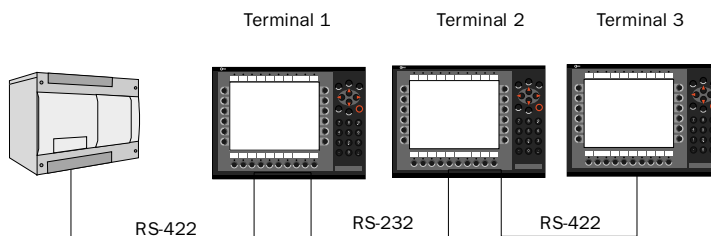
## Transfer speed

The transfer speed can vary between 600 and 19,200 baud. The highest transfer speed between the terminals is recommended to obtain the best results. The speed of communication decreases with the number of terminals connected according to the following table.

Number of terminals	Access time to the PLC			
	Terminal 1	Terminal 2	Terminal 3	Terminal 4
1	100%	-	-	-
2	50%	50%	-	-
3	50%	25%	25%	-
4	50%	25%	12.5%	12.5%

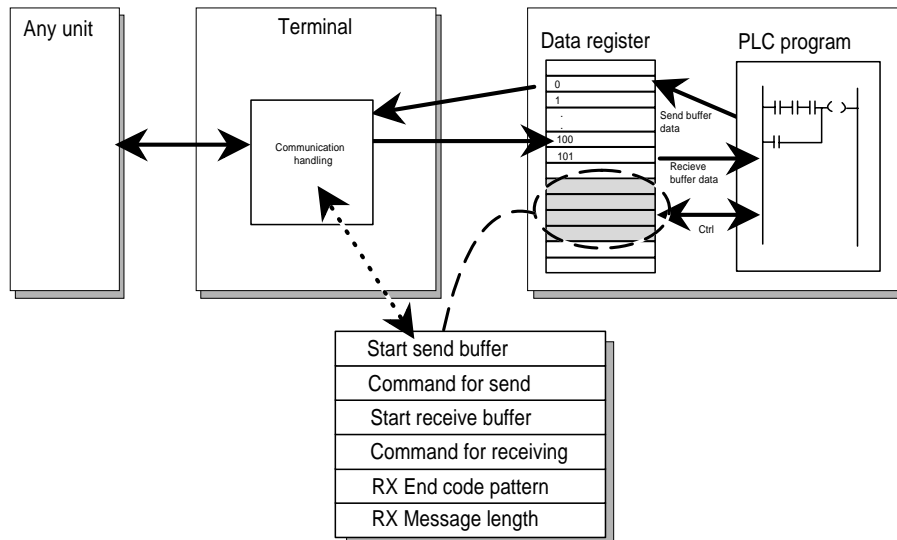
## Connecting three terminals in transparent mode

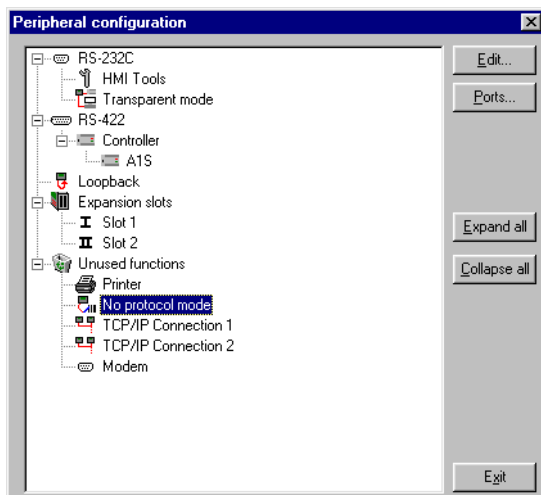
A third terminal can be connected to Terminal 2 in the network using a cable according to the figures above. The settings in the third terminal should be analog to the settings in the second terminal.



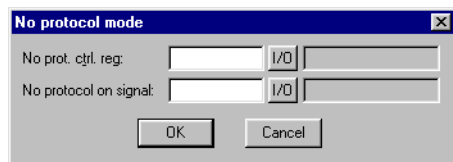
## 17.2 The terminal as communication interface (No Protocol Mode)

The *No protocol mode* is used to connect different PLC systems or connect external units such as bar code readers or weighing scales to the PLC system. The PLC system controls what is to be sent to the communication port. Data coming in to the communication port is written to the PLC register. Communication takes place with the transfer of the PLC register area according to the control block below.





Right click on No protocol mode in Peripheral Configuration under **Peripherals** in the **Setup** menu.

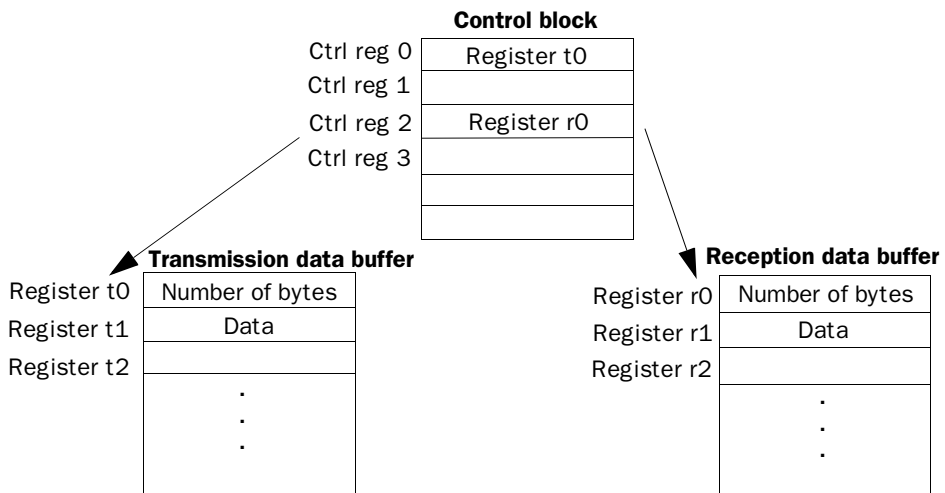


Register	Description
<b>Ctrl reg 0</b>	<i>Start register for the transmission data buffer.</i> The first register in the buffer area contains the total number of bytes to be transmitted. The following registers contains the data to be transmitted. The maximum buffer size is 127 registers = 254 bytes.
<b>Ctrl reg 1</b>	<i>Command register for transmission.</i> Set to 1 by the PLC system when transmission is required. Set to 0 by the terminal when transmission is complete.
<b>Ctrl reg 2</b>	<i>Start register for the reception data buffer.</i> The first register in the buffer area contains the total number of bytes received. The following register contains the data received. The maximum buffer size is 127 registers = 254bytes.

Register	Description
<b>Ctrl reg 3</b>	<i>Command register for reception.</i> <ul style="list-style-type: none"> <li>- Set to 0 by the PLC system when it is ready to receive.</li> <li>- Set to 1 by the terminal when the message is available.</li> <li>- Set to -1 (FFFF) on error message (e.g. too short).</li> <li>- Set to 2 by the PLC system when clearing of the port buffer is required.</li> <li>- Set to 3 by the terminal when clearing of the port buffer has been carried out.</li> </ul> The port buffer is automatically cleared on start up and when switching between transparent mode and no protocol mode, i.e. the register receives the value 3.
<b>Ctrl reg 4</b>	<i>End code (1 or 2 bytes) on the message received.</i>
<b>Ctrl reg 5</b>	<i>Length of the message received. End code is used if 0.</i>

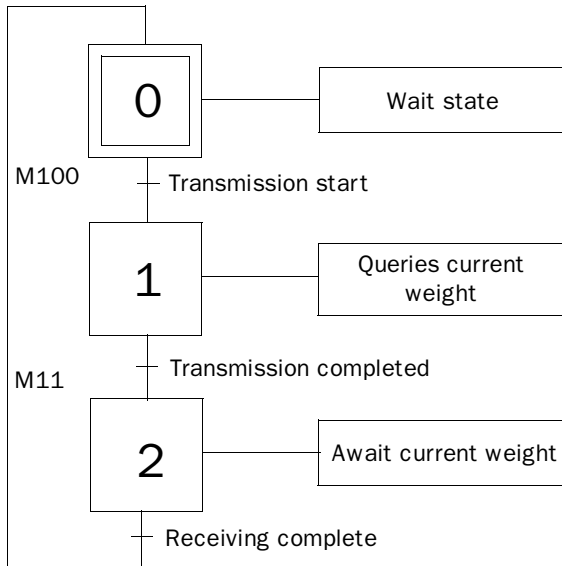
In the **Setup** menu, **Peripherals** you drag the icon No Protocol mode to the communication port you want to use. Right click to define which register will be the first control register in the transfer area. This and the five following registers are used as control registers.

The PLC system can switch between no protocol and transparent/prINTER modes in the run-time mode using a digital signal. State the digital signal in the dialog under No protocol on signal.

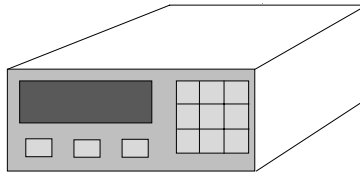


## Example of using the No Protocol Mode

The following example describes the use of the no protocol mode for a weighing system. The diagram below shows a three-step block schedule for communication.



The terminal functions as the master towards the weighing system, that is it continually asks the weighing system for the current weight. The diagram below shows the connections between the PLC system, terminal and weighing system.



The protocol appears as follows

Terminal

| STX | ? | CR | LF |

Weighing system

| STX | Weight in kg | CR | LF |

The PLC program appears as follows:

```
*****
*                               *
*               REQUIREMENT     *
*                               *
*****
```

D0-D5 contains the command registers.  
D10 is the first register in the send buffer.  
D20 is the first register in the receive buffer.

```
*****
*                               *
*               INITIATE        *
*                               *
*****
```

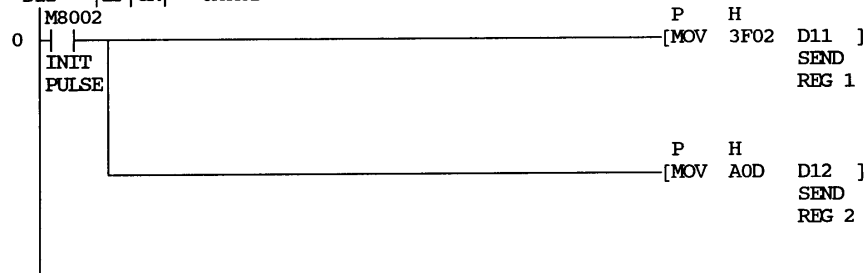
Initiation of question to the Weighing-machine

|STX|?|CR|LF|.

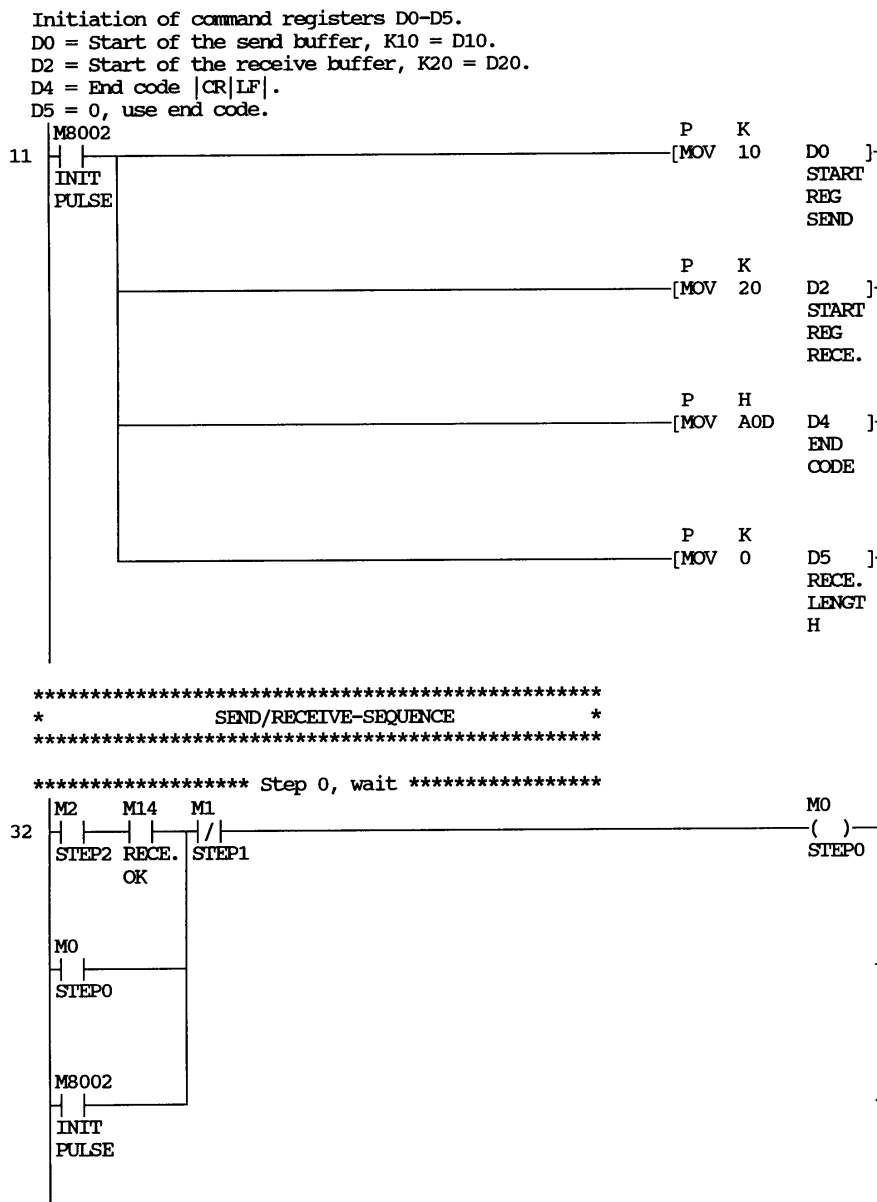
Note the order of the commands in D11 and D12.

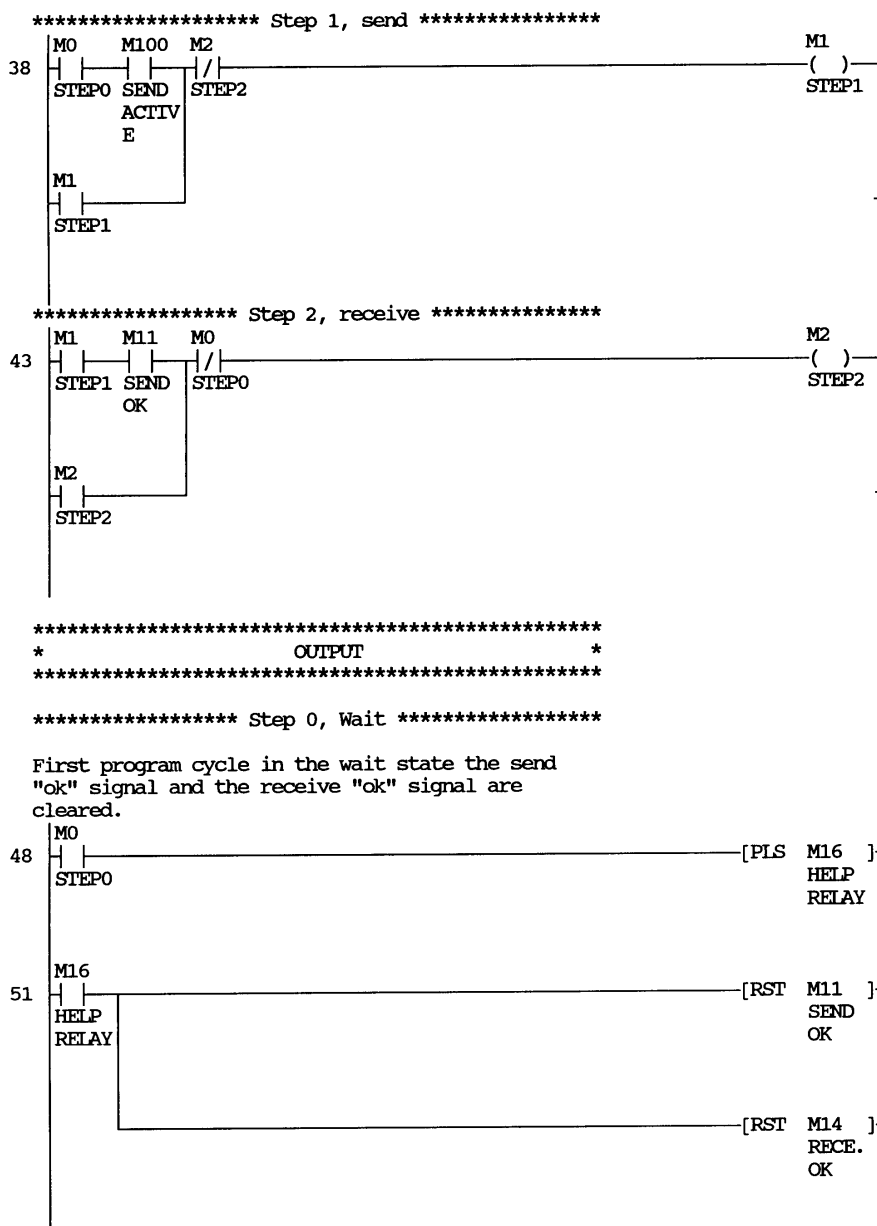
D11 = |?|STX| = H3F02

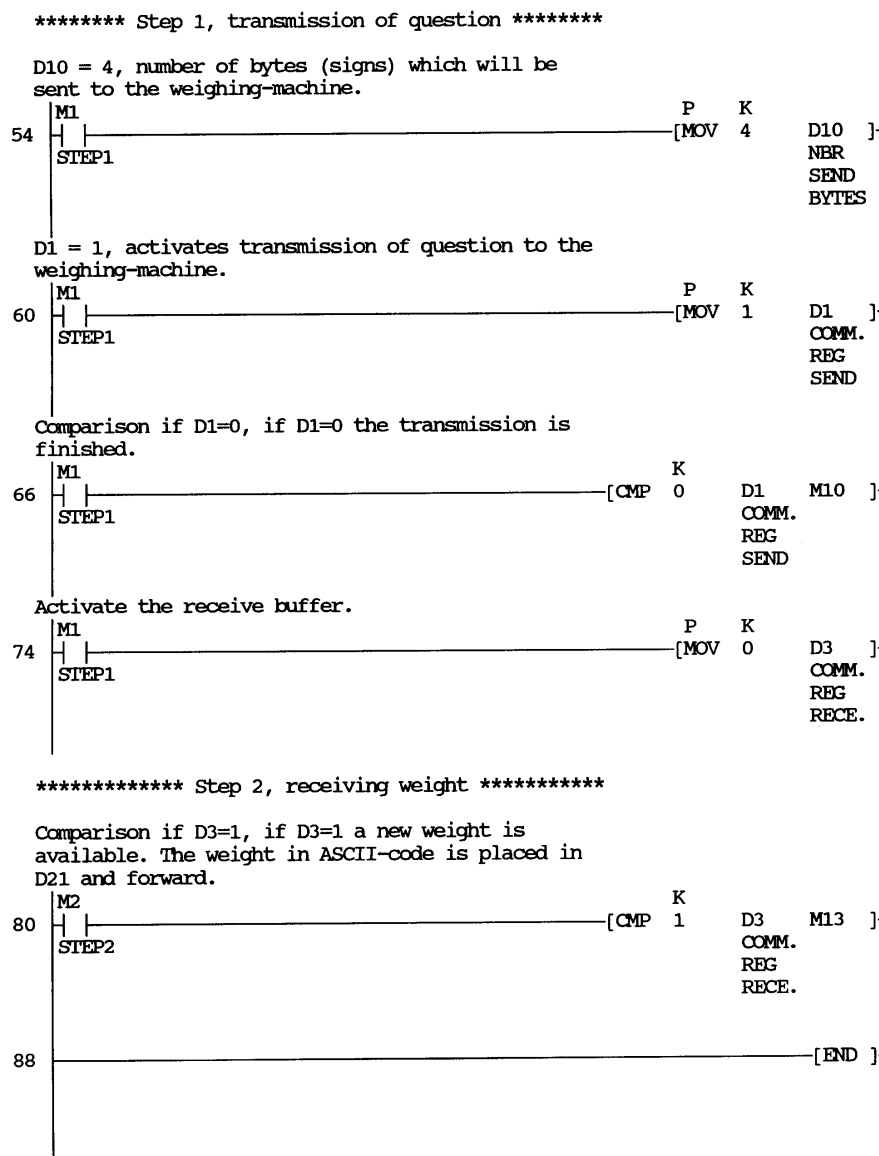
D12 = |LF|CR| = H0A0D





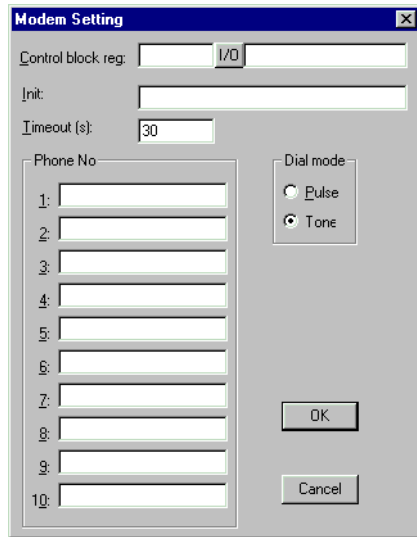






## 17.3 Modem connection

A modem can be used for making a connection to a personal computer. The settings for the connection are made in the **Modem Setting** dialog under **Peripherals** in the **Setup** menu. Select **Modem** and click Edit or click the right mouse button to show the dialog.



### Control block reg:

The communication is established using three control registers in a control block. The first register in the control block is defined at Control block reg in the dialog. The function of the control registers is described in the table below.

Register	Description
Ctrl reg 0	<p>Contains the command describing how the PLC system shall call up and establish connection.</p> <p>0 Wait for command.</p> <p>1-10 Call up with the phone number entered in the field Phone No 1-10. Maximum 40 characters.</p> <p>11 Call up with a phone number stored in the PLC system. This phone number is stored as an ASCII string beginning in the third control register and then some following registers. The string can contain maximum 40 characters, i.e. 20 registers. All registers do not have to be used. The last register that shall be read must contain the ASCII code 0.</p> <p>101-110 An init string is sent to the modem. Enter Hayes modem commands in the fields Phone No 1-10. Command 101 sends the string in the field Phone No 1 etc.</p> <p>111 An init string stored in the PLC system is sent to the modem. Enter Hayes modem commands beginning in the third control register. See Command 11 for details.</p> <p>255 Hang up command.</p>
Ctrl reg 1	<p>The second control register is used as a status register. This register contains the result of the modem command. The status register can contain the following:</p> <p><b>Status codes</b></p> <p>0 Command executed successfully.</p> <p>1 Dial in progress.</p> <p>2 Modem established a connection.</p> <p>3 Modem detected a ring signal.</p> <p><b>Error codes</b></p> <p>101 No connection (when hanging up).</p> <p>102 Modem detected lost carrier.</p> <p>103 Unspecified error from modem.</p> <p>104 Modem gets no dial tone.</p> <p>105 Busy when dialling.</p> <p>106 No answer when dialling.</p> <p>107 No reply from modem.</p> <p>255 Unknown error/status.</p>
Ctrl reg 2	<p>The terminal can call up with a phone number stored in the PLC system. This phone number is stored as an ASCII string beginning in the third control register and then some following registers. The string can contain maximum 40 characters, i.e. 20 registers. All registers do not have to be used. The last register that shall be read must contain the ASCII code 0.</p>

**Init:**

Modem init string.

**Timeout**

Number of seconds before the line is disconnected at inactivity. Default is 30 seconds. Times between 5 and 600 seconds can be stated.

**Dial mode:**

Select between Pulse and Tone.

**Phone No 1-10:**

Complete phone number for call up. Maximum 40 characters.



## 18 Network communication

This chapter describes how the E900 communicate in a network. The network communication takes place via TCP/IP (Transport Control Protocol/Internet Protocol). TCP/IP is a standardized set of protocols developed to allow cooperating computers to share resources across a network. The operator terminals can be connected in a network in many different ways. The connection can either be Ethernet or serial. If Ethernet is used the IFC ETTP or IFC ETCX expansion cards are required in all terminals. IFC ETTP is used to connect the terminals with twisted pair and IFC ETCX is used to connect the terminals with a coaxial cable. If twisted pair is used an Ethernet HUB is required. For further information about TCP/IP please refer to the RFC's published on the Internet. You will find them for example at this address:

<ftp://ftp.sunet.se/pub/internet-documents/rfc>.

Information and description of concept and words used can be found on the web address <http://www.whatis.com>.

The terminal network is a Client/Server network and works as follows. Only clients can access data on the network. Servers only provide access for other clients. A terminal can be both a client and a server to be able to provide access and also access other terminals. Up to 20 different clients can access data from the same server. A client can access data from up to 16 different servers.

Each terminal requires an IP address. IP addresses from 192.168.0.0 to 192.168.254.254 are recommended for internal networks.

With the terminal you can use standardized Internet tools such as browser, mail server and FTP client. You can create a web site in a PC and load it to the terminal. The Web page can contain real-time data from the PLC system or the terminal. Via Internet and a browser you can change values, set signals, acknowledge alarms etc via script.

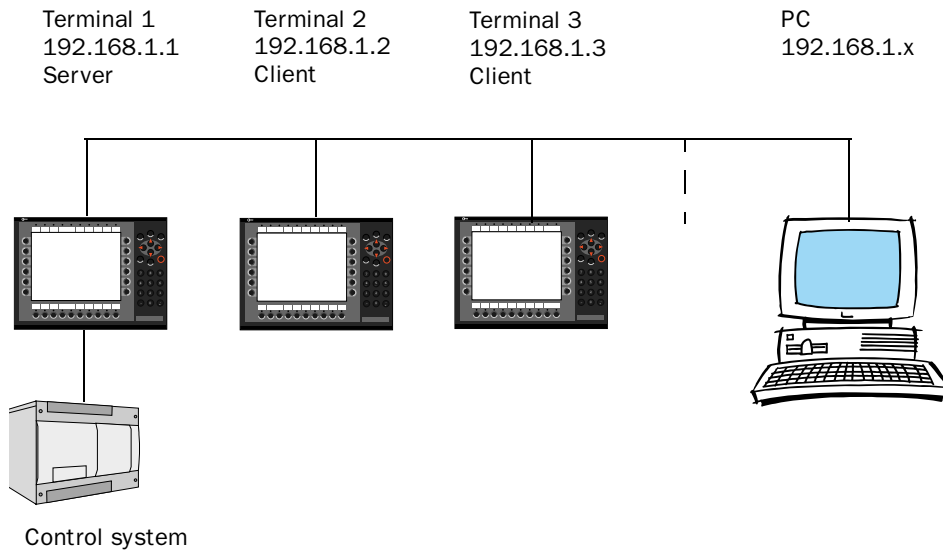
The terminal can also send email for special events such as alarm and status report.



## 18.1 Examples of possible networks

### Example 1

This Ethernet network is designed for one controller together with several terminals.



#### Terminal 1 (192.168.1.1)

Terminal 1 must be a server. Clients can access the controller through the server.

**TCP/IP Network Connection**

Connection name: Terminal 1

Host configuration: Manual

TCP/IP Settings

IP Address: 192.168.1.1

Subnet mask: 255.255.255.0

Gateway: 0.0.0.0

Primary DNS: 0.0.0.0

Secondary DNS: 0.0.0.0

OK Cancel

**Setup BDTP Server service**

Server port: 6002

Max Clients: 8

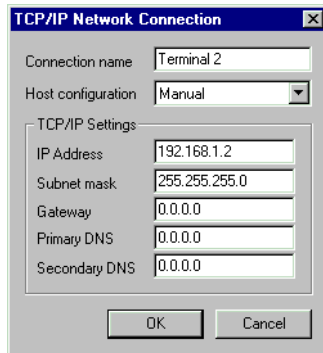
Data register: I/O

☒ Lock server

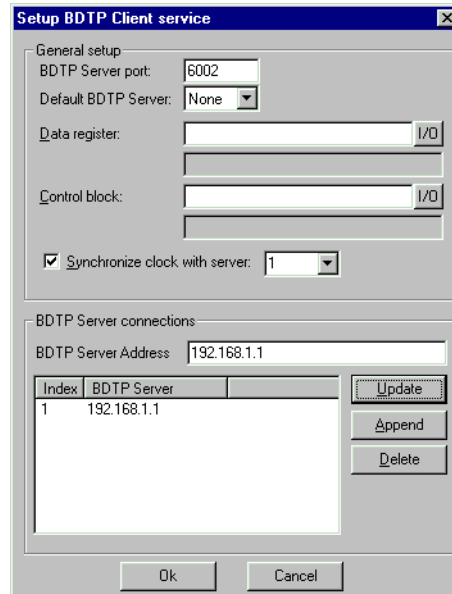
OK Cancel

**Terminal 2 (192.168.1.2)**

Terminal 2 must be a client. A client can access the controller connected to a server.



The TCP/IP Network Connection dialog box is shown. It has a title bar with a close button. The 'Connection name' field contains 'Terminal 2'. The 'Host configuration' dropdown menu is set to 'Manual'. Under the 'TCP/IP Settings' section, the 'IP Address' field contains '192.168.1.2', 'Subnet mask' contains '255.255.255.0', 'Gateway' contains '0.0.0.0', 'Primary DNS' contains '0.0.0.0', and 'Secondary DNS' contains '0.0.0.0'. At the bottom are 'OK' and 'Cancel' buttons.



The Setup BDTP Client service dialog box is shown. It has a title bar with a close button. The 'General setup' section includes: 'BDTP Server port' set to '6002', 'Default BDTP Server' set to 'None', 'Data register' and 'Control block' fields with 'I/O' buttons, and a checked 'Synchronize clock with server' checkbox with a dropdown set to '1'. The 'BDTP Server connections' section has a 'BDTP Server Address' field set to '192.168.1.1' and a table with one entry: Index 1, BDTP Server 192.168.1.1. To the right of the table are 'Update', 'Append', and 'Delete' buttons. At the bottom are 'Ok' and 'Cancel' buttons.

Index	BDTP Server
1	192.168.1.1

### Terminal 3 (192.168.1.3)

Terminal 3 must be a client. A client can access the controller connected to a server.

**TCP/IP Network Connection**

Connection name: Terminal 3

Host configuration: Manual

TCP/IP Settings:

IP Address: 192.168.1.3

Subnet mask: 255.255.255.0

Gateway: 0.0.0.0

Primary DNS: 0.0.0.0

Secondary DNS: 0.0.0.0

OK Cancel

**Setup BDTP Client service**

General setup

BDTP Server port: 6002

Default BDTP Server: None

Data register: I/O

Control block: I/O

☒ Synchronize clock with server: 1

BDTP Server connections

BDTP Server Address

Index	BDTP Server
1	192.168.1.1

Update Append Delete

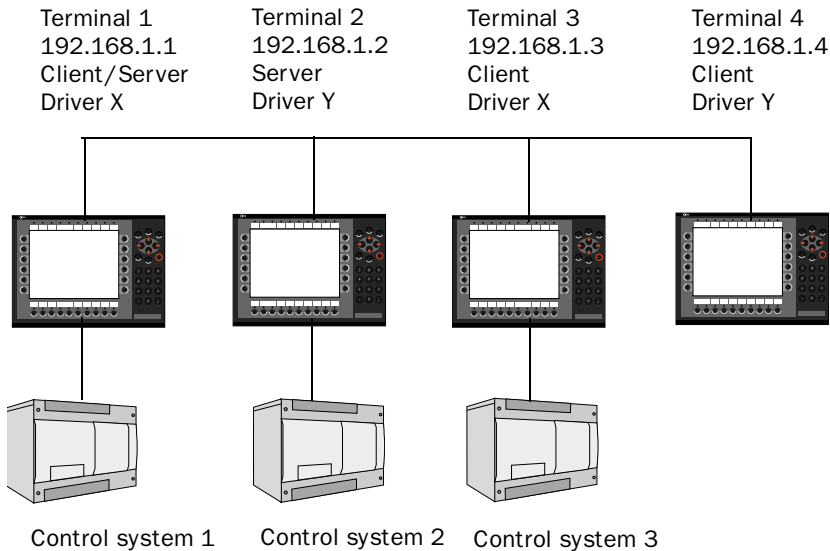
OK Cancel

A PC can be connected for transferring projects to the terminals. The PC can be connected either via an Ethernet card or to one of the serial ports on one of the terminals using CAB5 or CAB6. The Ethernet card is recommended.

## Example 2

This Ethernet network is designed for several controllers and operator terminals in one network. The controllers in this example use different protocol drivers.

The following Client/Server and Driver configuration gives the following access. This example uses two drivers, driver X and driver Y.



### Terminal 1 (192.168.1.1)

Terminal 1 is a client with driver X and can therefore access all the servers with the same driver. It can also access its local controller 1. This results in access to controller 1 only. It cannot access controller 2 because of different drivers and not 3 because the terminal 3 is not a Server.

### Terminal 2 (192.168.1.2)

Terminal 2 is not a Client and cannot access any other stations in the network. It can access its local controller 2. This results in access to controller 2 only

### Terminal 3 (192.168.1.3)

Terminal 3 is a Client with driver X and can therefore access all Servers with the same driver. It can also access its local controller 3. This results in access to controller 1 and 3. Terminal 3 cannot access controller 2 because of different drivers.

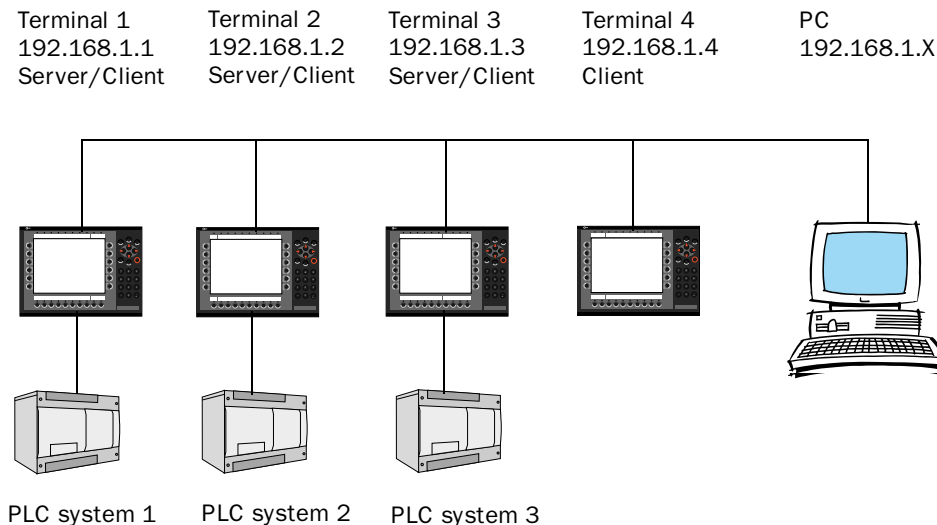
#### Terminal 4 (192.168.1.4)

Terminal 4 is a client with driver Y and can access all servers with the same driver. This results in access to controller 2 only. Terminal 4 cannot access controller 1 because of different drivers and not 3 because it is not a server.

Data between the controllers can be exchanged with a small controller program or using function keys. See “BDTP Client” and “BDTP Server”.

### Example 3

This Ethernet network is a normal configuration where a PC is included. This allows the computer to access any terminal on the network and transfer a project. The PC can be connected either via an Ethernet card or to one of the serial ports on one of the terminals using CAB5 or CAB6. The Ethernet card is recommended. The example assumes that the same driver is used. To change PLC system we refer to the sections “BDTP Client” and “BDTP Server”.



#### Terminal 1 (192.168.1.1)

Terminal 1 is a client and can therefore access all the servers. It can also access its local controller 1. This results in access to controllers 1 and 2.

**Terminal 2 (192.168.1.2)**

Terminal 2 is a client and can therefore access all the servers. It can also access its local controller 2. This results in access to controllers 1 and 2.

**Terminal 3 (192.168.1.3)**

Terminal 3 is a client and can therefore access all the servers. It can also access its local controller 3. This results in access to controllers 1 and 2.

**Terminal 4 (192.168.1.4)**

Terminal 4 is a client and can therefore access all the servers. This results in access to controllers 1 and 2.

Data between the controllers can be exchanged with control registers. See “BDTP Client” and “BDTP Server”.

---

**Note!**

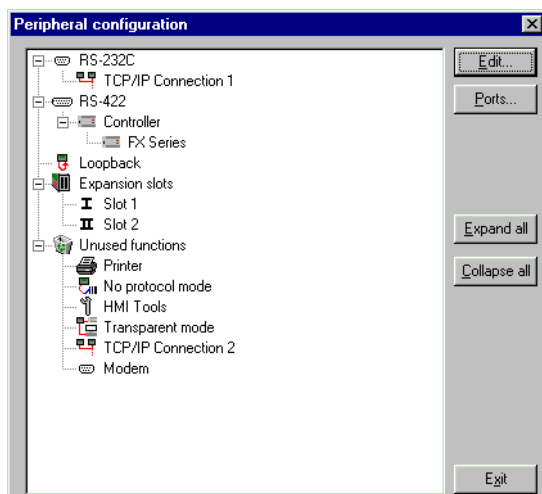
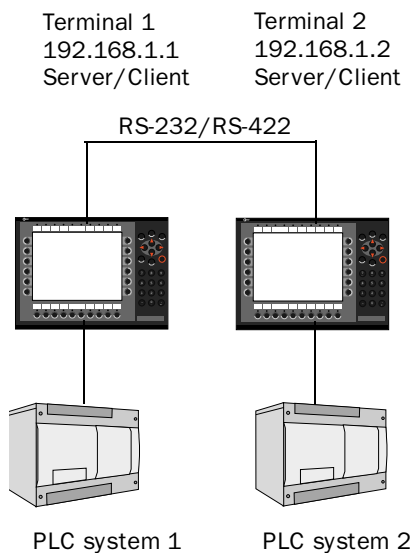
When a project is transferred to a terminal this station will stop communication momentarily. If the terminal is a Server all the clients requesting data from this Server will receive an error message.

---

## Example 4

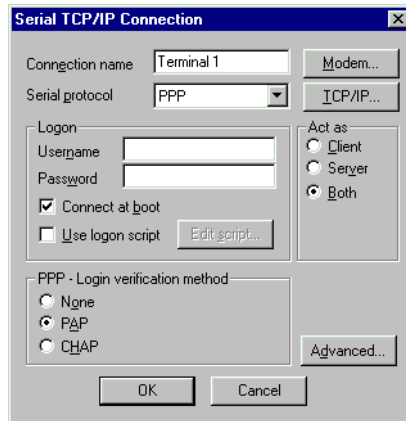
When only two stations are to be used a connection can be established with serial communication. This setup uses the built-in serial port to connect to the second terminal. No Ethernet card is required.

One or two controllers can be used with this "network". Both the controller systems in the network use the same driver and both the terminals have access to both the controller systems 1 and 2.



**Terminal 1 (192.168.1.1)**

Terminal 1 is a client and can therefore access the other server. It can also access its local controller 1. This results in access to controller 1 and controller 2.



**Serial TCP/IP Connection**

Connection name:

Serial protocol:

Logon:

Username:

Password:

☒ Connect at boot

☐ Use logon script

Act as:

☐ Client

☐ Server

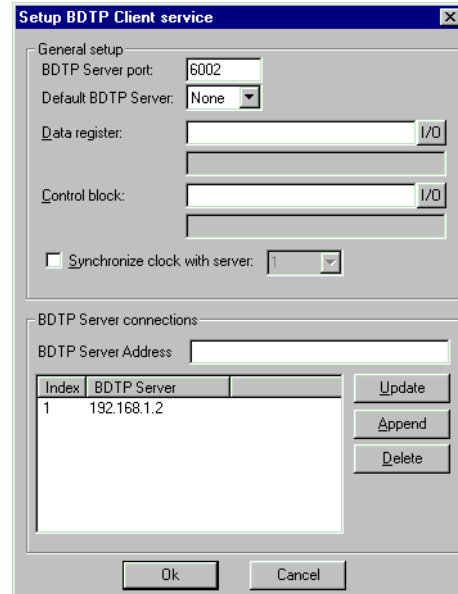
☒ Both

PPP - Login verification method:

☐ None

☒ PAP

☐ CHAP



**Setup BDTP Client service**

General setup:

BDTP Server port:

Default BDTP Server:

Data register:

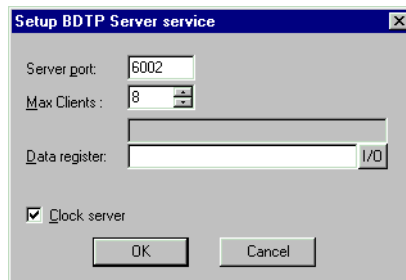
Control block:

☐ Synchronize clock with server:

BDTP Server connections:

BDTP Server Address:

Index	BDTP Server
1	192.168.1.2



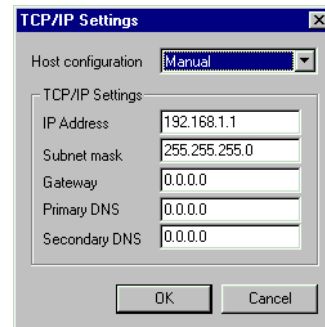
**Setup BDTP Server service**

Server port:

Max Clients:

Data register:

☒ Clock server



**TCP/IP Settings**

Host configuration:

TCP/IP Settings:

IP Address:

Subnet mask:

Gateway:

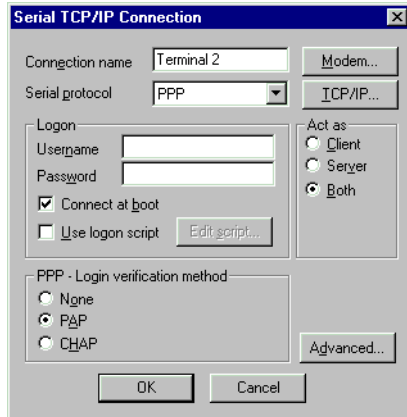
Primary DNS:

Secondary DNS:



**Terminal 2 (192.168.1.2)**

Terminal 2 is a client and can therefore access the other server. It can also access its local controller 2. This results in access to controller 1 and controller 2.



**Serial TCP/IP Connection**

Connection name:

Serial protocol:

Logon:

Username:

Password:

☒ Connect at boot

☐ Use logon script

Act as:

☐ Client

☐ Server

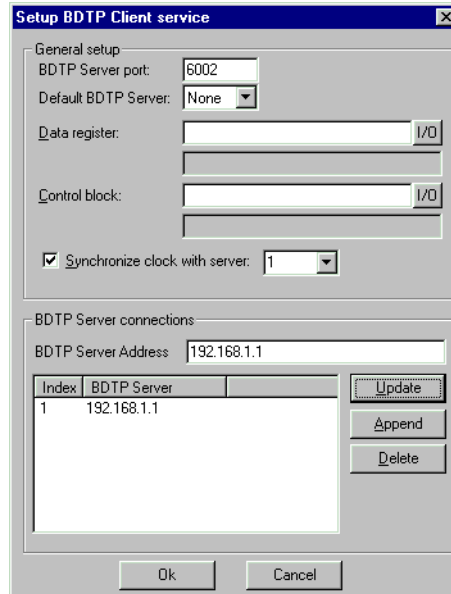
☒ Both

PPP - Login verification method:

☐ None

☒ PAP

☐ CHAP



**Setup BDTP Client service**

General setup

BDTP Server port:

Default BDTP Server:

Data register:

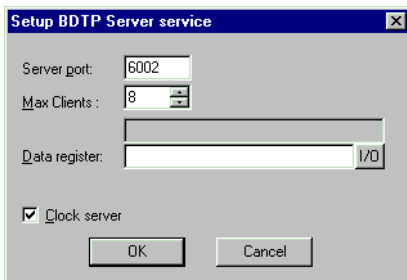
Control block:

☒ Synchronize clock with server:

BDTP Server connections

BDTP Server Address:

Index	BDTP Server
1	192.168.1.1



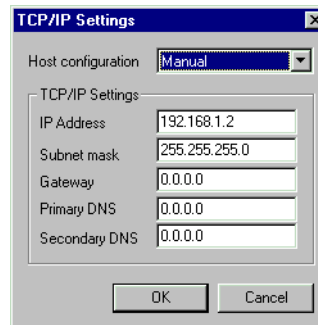
**Setup BDTP Server service**

Server port:

Max Clients:

Data register:

☒ Clock server



**TCP/IP Settings**

Host configuration:

TCP/IP Settings

IP Address:

Subnet mask:

Gateway:

Primary DNS:

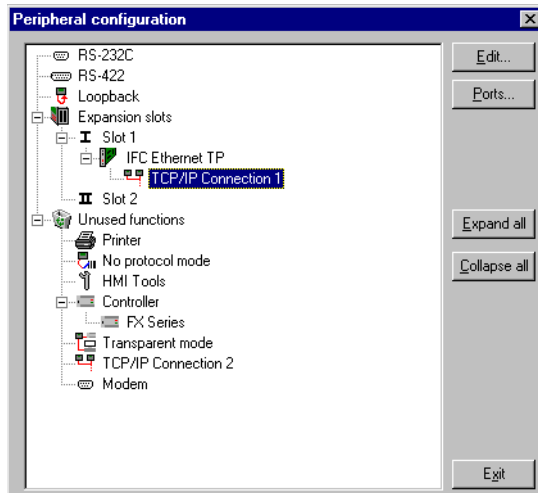
Secondary DNS:

## 18.2 Network communication via ETHERNET

You need one of the IFC ETTP or IFC ETCX expansion cards in the terminals to connect the terminals to a TCP/IP network via ETHERNET. IFC ETTP is used for connection via pair twisted cable and IFC ETCX is used for connection via coaxial cable.

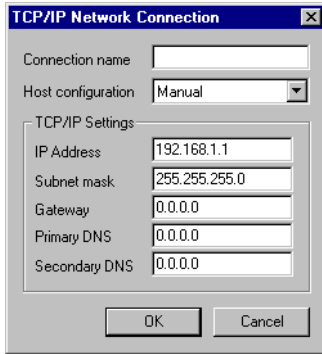
### How to make a connection

Install the IFC ETTP or IFC ETCX card according to the manual for each card. Select **Peripherals** in the **Setup** menu. Mark the selected expansion slot and click the right mouse button. Select network card. Mark **TCP/IP connection** and keep the left mouse button pressed. Then drag the mouse to the selected expansion card and release the mouse button.



## Settings

The settings for the serial connection are made in the dialog shown when the connection is selected and you press **Edit....**



### Connection name

Enter any name for the connection. The parameter does not need to be stated.

### Host configuration

Manual means that the settings of the TCP/IP parameters made in the dialog are used. The other alternatives mean that the terminal provides one or several of the TCP/IP parameters from a network server.

### IP-adress

State the IP address of your node. For information about which IP address to use please refer to your network manager or to the RFC's from the Internet committee. If you are using a local network between the terminals we recommend you use the ip-addresses 192.168.0.0 – 192.168.254.254.

### Subnet mask

State your network identity.

### Gateway

State the network unit in the local network that can verify the identifiers for the other networks on Internet.

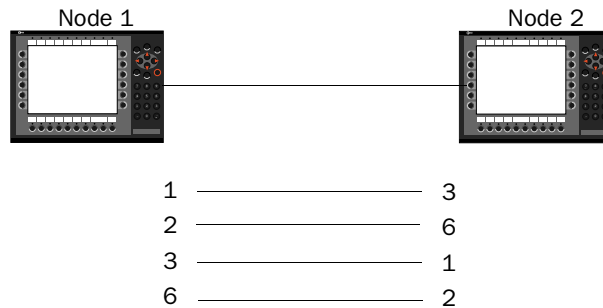
### Primary DNS and Secondary DNS

State the network servers containing information about a part of the DNS data-base.

## ETHERNET connections

The section below shows three examples of ETHERNET connections between two terminals.

### Connection between two terminals via twisted pair (TP)



The cable has RJ45 connectors. The cable is a CAT5 (cross coupled) pair twisted cable, screened or not screened.

#### Note!

If the communication does not work properly, and the LED marked Link on the IFC ETPP card does not light, connections 3 and 6 are probably switched.

### TCP/IP settings in the nodes

Node 1

TCP/IP Network Connection

Connection name:

Host configuration:

TCP/IP Settings

IP Address:

Subnet mask:

Gateway:

Primary DNS:

Secondary DNS:

OK Cancel

Node 2

TCP/IP Network Connection \*

Connection name:

Host configuration:

TCP/IP Settings

IP Address:

Subnet mask:

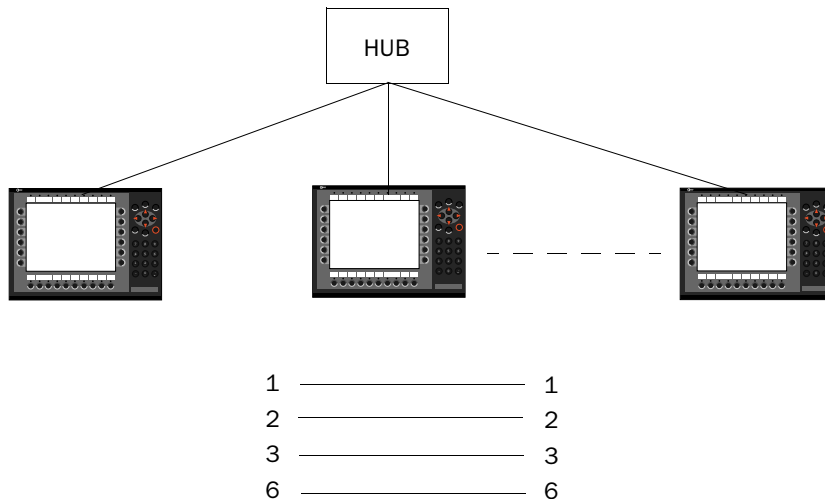
Gateway:

Primary DNS:

Secondary DNS:

OK Cancel

## Connection between more than two terminals via twisted pair (TP)



The maximal length between the terminal and the HUB is 100 m. The number of terminals per HUB is limited by the number of connections on the HUB. The cable has RJ45 connectors. The cable is a CAT5 pair twisted cable, screened or unscreened.

### TCP/IP settings in the nodes

Node 1

Node 1 TCP/IP Network Connection settings:

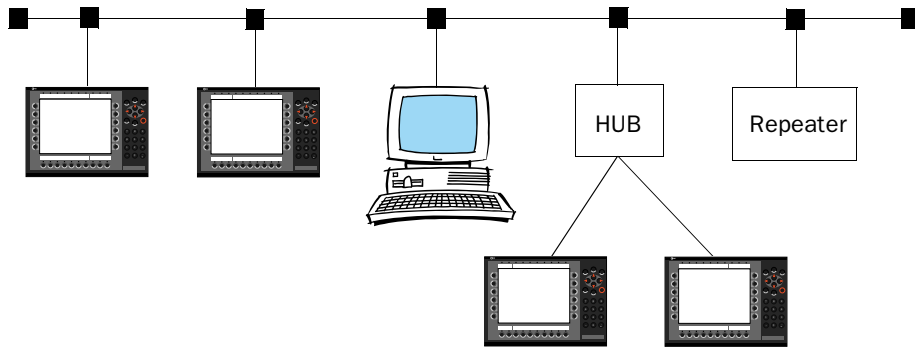
- Connection name: [Empty]
- Host configuration: Manual
- TCP/IP Settings:
  - IP Address: 192.168.1.1
  - Subnet mask: 255.255.255.0
  - Gateway: 0.0.0.0
  - Primary DNS: 0.0.0.0
  - Secondary DNS: 0.0.0.0

Node 2

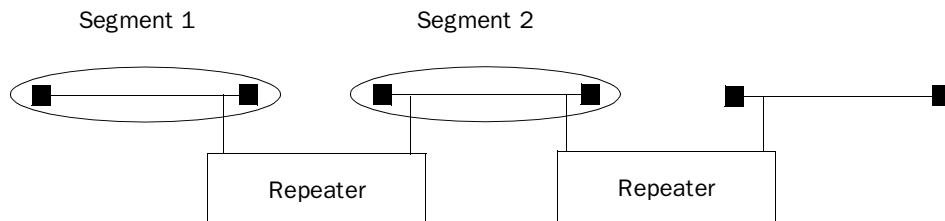
Node 2 TCP/IP Network Connection settings:

- Connection name: [Empty]
- Host configuration: Manual
- TCP/IP Settings:
  - IP Address: 192.168.1.2
  - Subnet mask: 255.255.255.0
  - Gateway: 0.0.0.0
  - Primary DNS: 0.0.0.0
  - Secondary DNS: 0.0.0.0

## Connection via coaxial cable

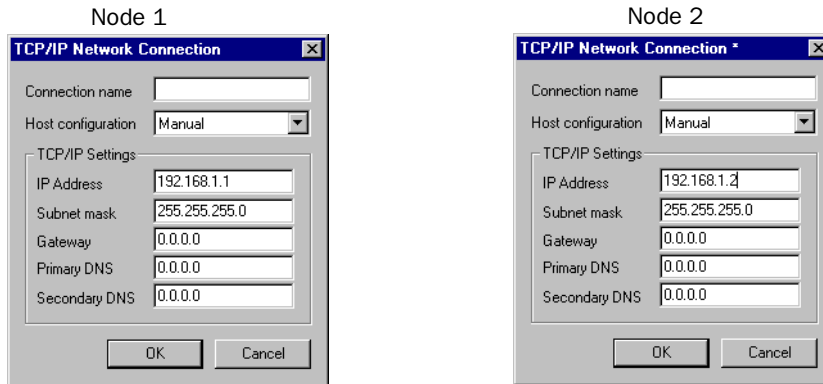


The coaxial cable RG58 is used in the network. There must be bus endings of 50 Ohm. Several coaxial cables can be connected with repeaters to get longer networks. The network will then consist of different segments.



The maximum length of a segment is 185 m and the maximum number of units/segment is 30. Four repeaters can be serially connected, which means that five segments may be serially connected. Only three of the segments can have units connected. This means that there must be at least two empty segments in the network. The RG58 network cables must be grounded on one side. For more information see RFC on the Internet.

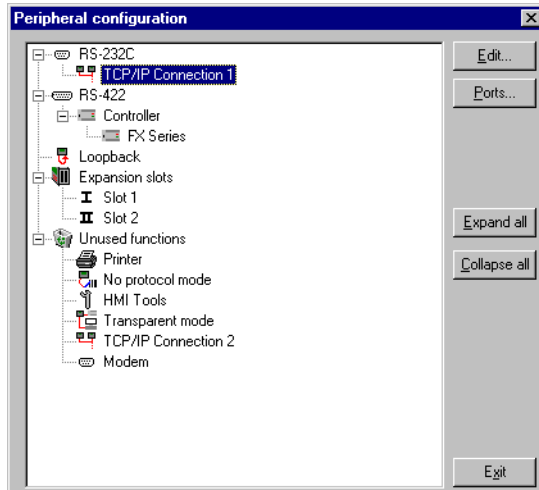
## TCP/IP settings in the nodes



## 18.3 Serial network communication

### How to make a connection

Select **Peripherals** in the **Setup** menu. In the dialog **Peripheral configuration** you mark TCP/IP connection and keep the left mouse button pressed. Then drag the mouse to RS-232 or RS-422 and release the mouse button.

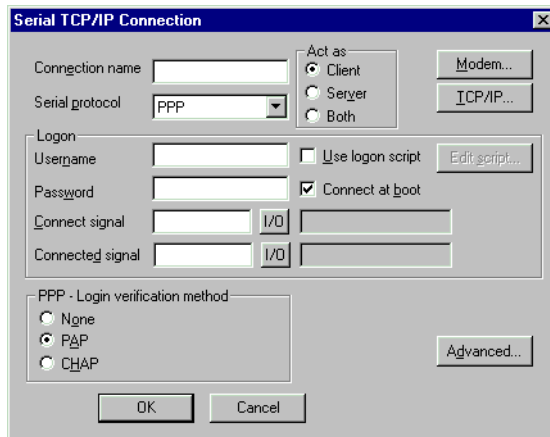


#### Note!

The setting of parity on the port connected to the TCP/IP connection must be NONE.

## Settings

The settings for the serial connection are made in the dialog shown when the connection is selected and you press **Edit...**



### Connection name

Enter freely a name for the connection. The parameter does not need to be stated.

### Serial protocol

For serial communication you can select the protocol PPP.

### User name

State the user name used at login.

### Password

State the password used at login.

### Connect signal

Digital signal which establishes the connection when it is set and disconnects when it is reset.

### Connected signal

Analog register which can have the following status.

- 0        Disconnected (PPP Client)
- 1        Waiting for a connection (PPP Server)
- 2        Connected as a PPP Client
- 3        Connected as a PPP Server
- 7        Connection error



### Use logon script

The function is used to automatize a serial login. The script will look different depending on which server and modem you connect to. The terminal supports the following commands.

- |                     |  |
|---------------------|--|
| 1. WAIT: Text , x   | Wait for the text Text for x seconds. x must not be stated   |
| 2. SEND: Text       | Sends the text Text  |
| 3. LABEL: Label     | Label becomes a reference point in the script.   |
| 4. ONERR: Label     | Jumps Label when an error occurs in previous command.  |
| 5. MESSAGE: Message | Shows a message box with the message Message.  |
| 6. END:             | End the script.  |
| 7. SLEEP: x         | Sleeps in x seconds.   |
| 8. COUNTER: y       | Count up a counter each time the script passes "COUNTER:". If the value y is reached an error occurs. This error is handled by "ONERR:". |

Variables:

- |           |   |
|-----------|---|
| %USERNAME | Compare with the names in accounts.     |
| %PASSWORD | Compare with the passwords in accounts. |

Example 1:

```
WAIT: login:, 10
SEND: KALLE
WAIT: password:,10
SEND: HELLO
```

**Example 2:**

The following script sends the text "CLIENT". If the transmission failed a message box with the text "Send Failed" is displayed. If the transmission succeeded we wait for the text "CLIENTSERVER" for 10 seconds. If this does not appear a message box with the text "Receive Failed" is displayed.

```
SEND: CLIENT
ONERR: Send Failure
WAIT: CLIENTSERVER,10
ONERR: Receive Failure
END:
LABEL: Send Failure
MESSAGE: Send Failed
END:
LABEL: Receive Failure
MESSAGE: Receive Failed
END:
```

**Example 3:**

The following script sends the text "login:" and then waits for the receiver to send a user name. This is checked to see if the name matches a user name in accounts. Then the script sends the text "password:" and wait for the receiver to send a password. This is checked in accounts to see if the password matches the password for the user name sent.

```
SEND: login:
WAIT: %USERNAME
SEND: password:
WAIT:%PASSWORD
```

Normally no script is needed. If you connect to a Windows NT server the script shall be as follows.

**PPP - Login verification method**

Select method to assure the user identity. Does not normally need to be changed. This parameter is only used for PPP connection.

**Act as Client/Server**

For PPP connection you select if the terminal shall act as a PPP client or PPP server. The terminal can both call and be called.

## Advanced

Under **Advanced** you can define further parameters.

### Use VJ compression of IP header

Compression of the IP header. Does not normally need to be changed. This parameter is used only for PPP connection.

### Request/Provide remote address

Requests/provides the IP address for the remote node. Shall be 0.0.0.0 if you want to request the IP address from the remote node. This parameter is only used for PPP connection.

### Request/Provide local address

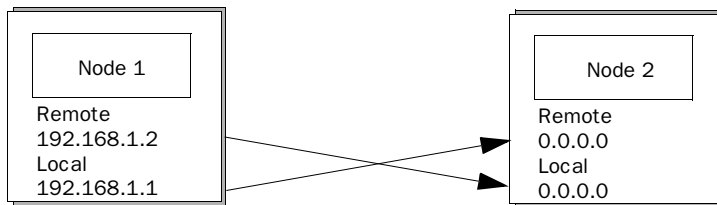
Requests/provides the local IP address. Shall be 0.0.0.0 if you want to request the IP address from the remote node. This parameter is only used for PPP connection.

---

#### Note!

If the remote/local provides addresses changes when the terminal is a server or both server and client the new addresses will be stored. Is the terminal a client the addresses will be set to 0.0.0.0. If you change the terminal to be a server or both a server and a client the addresses stored is used.

---



### Set remote address as gateway

Mark if you want the remote node address to be a gateway, a connection port to another network. Default is OFF. The parameter is only used for PPP connection.

---

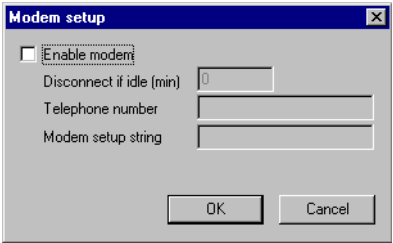
#### Note!

If the parameter “Set remote address as gateway” is not marked and you use a sub network the communication in the network will fail. It also means that email may be transferred from the terminal but you cannot login to the terminal from, for example, a FTP client or a browser.

---

**Modem**

The parameters under **Modem setup** are stated if the connection shall take place via a modem. You also need a special cable which must be according to the following figure.



9-pin female		25-pin male
2	_____	2
3	_____	3
5	_____	7
7	_____	8
8	_____	4

**Enable modem**

Mark if you use modem.

**Disconnect if idle (min)**

Disconnect the connection if inactive for stated number of minutes. 0 means that the connection never will be disconnected.

**Telephone number**

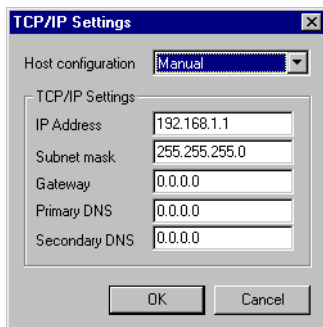
State the telephone number to be called.

**Modem setup string**

A setup string which initializes the modem. For further information please refer to the documentation for the modem.

## TCP/IP

Parameters for the TCP/IP connection.



### IP-adress

State the IP address of your node. For information about which IP address to use, please refer to your network manager or to the RFC's from the Internet committee. If you are using a local network between the terminals we recommend you use the addresses 192.168.1.1 – 192.168.1.254.

### Subnet mask

State your network identity.

### Gateway

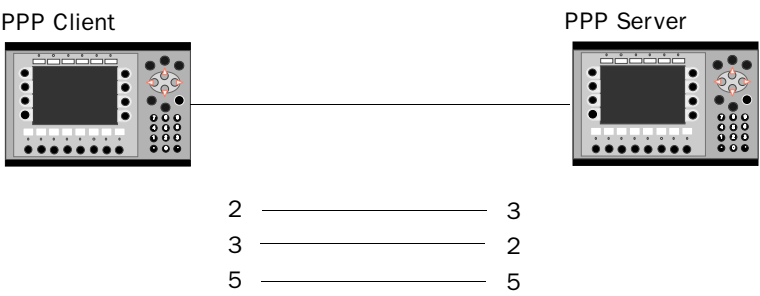
State the network unit in the local network that can verify the identifiers for the other networks on the Internet.

### Primary DNS and Secondary DNS

State the servers containing information for a part of the DNS data- base.

# Serial connection

## A PPP connection between two terminals



**Serial TCP/IP Connection**

Connection name: Client

Serial protocol: PPP

Act as: ☒ Client ☐ Server ☐ Both

Logon: Username: Password: ☐ Use logon script ☒ Connect at boot

Connect signal: I/O Connected signal: I/O

PPP - Login verification method: ☐ Ngne ☒ PAP ☐ CHAP

OK Cancel Advanced...

**Advanced PPP Settings**

☒ Use VJ compression of IP headers

☒ Request/provide remote address Remote address: 0.0.0.0

☒ Request/provide local address Local address: 0.0.0.0

OK Cancel

**Serial TCP/IP Connection**

Connection name: Server

Serial protocol: PPP

Act as: ☐ Client ☒ Server ☐ Both

Logon: Username: Password: ☐ Use logon script ☒ Connect at boot

Connect signal: I/O Connected signal: I/O

PPP - Login verification method: ☐ Ngne ☒ PAP ☐ CHAP

OK Cancel Advanced...

**Advanced PPP Settings**

☒ Use VJ compression of IP headers

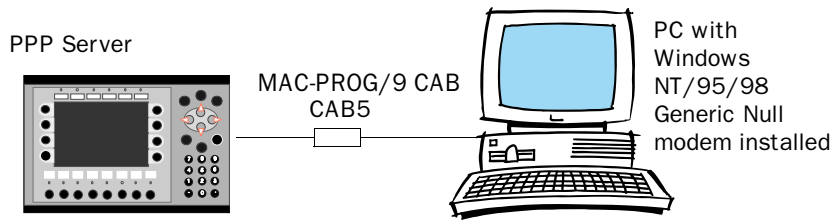
☒ Request/provide remote address Remote address: 192.168.98.2

☒ Request/provide local address Local address: 192.168.98.1

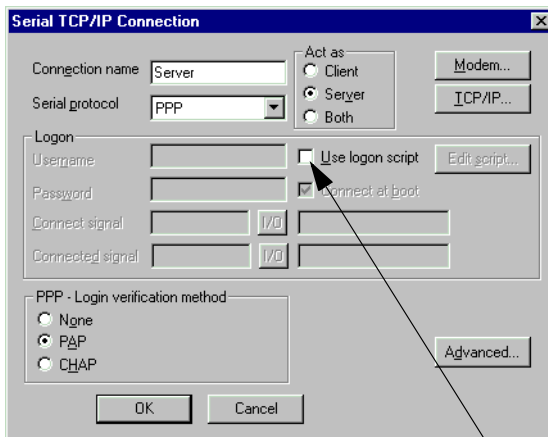
OK Cancel

The settings under TCP/IP, IP Address, Subnet mask and Gateway have no meaning. PPP will change these parameters.

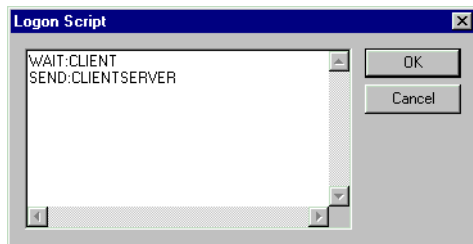
## A PPP connection between a terminal and a personal computer



### Settings in the terminal



The settings under TCP/IP, IP Address, Subnet mask and Gateway have no meaning. PPP will change these parameters. Note that if PAP is selected as Login verification method User name and Password must be stated.



Valid for Windows NT.  
Mark the "Use logon script" box before use logon script.

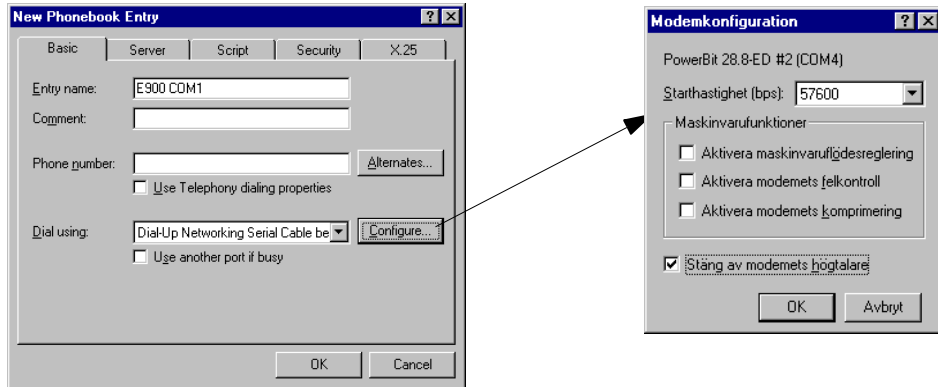
### PC settings

The figures below shows the settings in the different dialogs when creating a serial PPP connection between a PC and a terminal. In this document we assume that the user has installed a Generic Null modem and defined a port for serial cable for Dial-up networking.

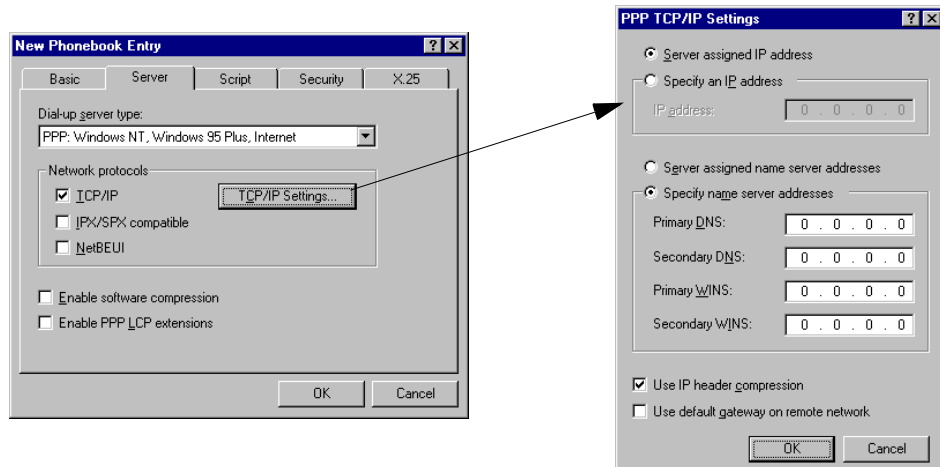
Select the function **New connection** according to the following path:

Start\Programs\Accessories\Dial-up Networking\New connection

The following dialog is shown. Enter the settings according to the following figure.

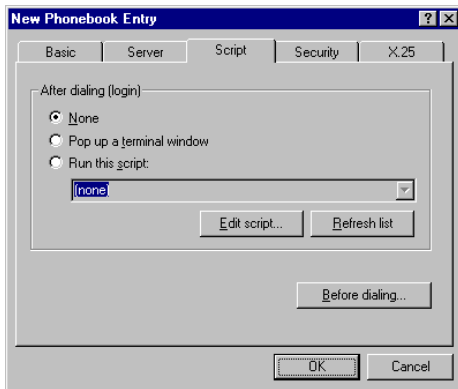


Click on the file tab **Server** and enter the following settings.



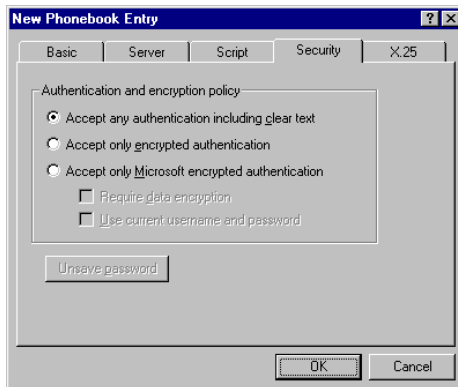


Click on the file tab **Script** and enter the following settings.



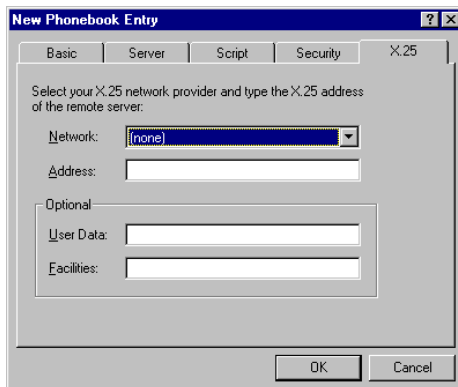
The 'New Phonebook Entry' dialog box is shown with the 'Script' tab selected. The 'After dialing (login)' section has three radio buttons: 'None' (selected), 'Pop up a terminal window', and 'Run this script:'. Below the 'Run this script' option is a dropdown menu showing '(none)'. There are 'Edit script...' and 'Refresh list' buttons. At the bottom are 'OK' and 'Cancel' buttons.

Click on the file tab **Security** and enter the following settings.



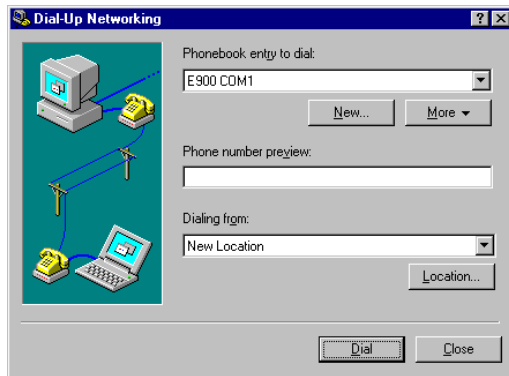
The 'New Phonebook Entry' dialog box is shown with the 'Security' tab selected. The 'Authentication and encryption policy' section has three radio buttons: 'Accept any authentication including clear text' (selected), 'Accept only encrypted authentication', and 'Accept only Microsoft encrypted authentication'. Below these are two checkboxes: 'Require data encryption' and 'Use current username and password'. There is an 'Unsave password' button. At the bottom are 'OK' and 'Cancel' buttons.

Click on the file tab **X 25** and enter the following settings.



The 'New Phonebook Entry' dialog box is shown with the 'X.25' tab selected. The 'Select your X.25 network provider and type the X.25 address of the remote server.' section has a 'Network:' dropdown menu showing '(none)' and an 'Address:' text field. Below this is an 'Optional' section with 'User Data:' and 'Facilities:' text fields. At the bottom are 'OK' and 'Cancel' buttons.

The settings have now been entered. To establish communication you select the Dial-up networking function. Enter the defined post name and select Dial-up.

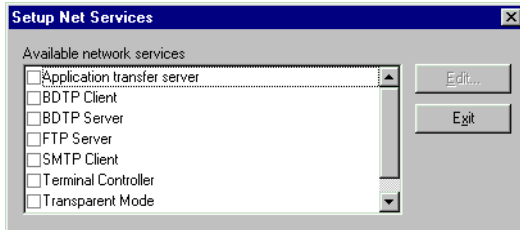


In the following dialog you shall confirm the call-up. Press OK.



## 18.4 Network services

Under **Network/Services** in the **Setup** menu you select the services the terminal shall provide from the network. Select current function and click on **Edit...**



### Application transfer server

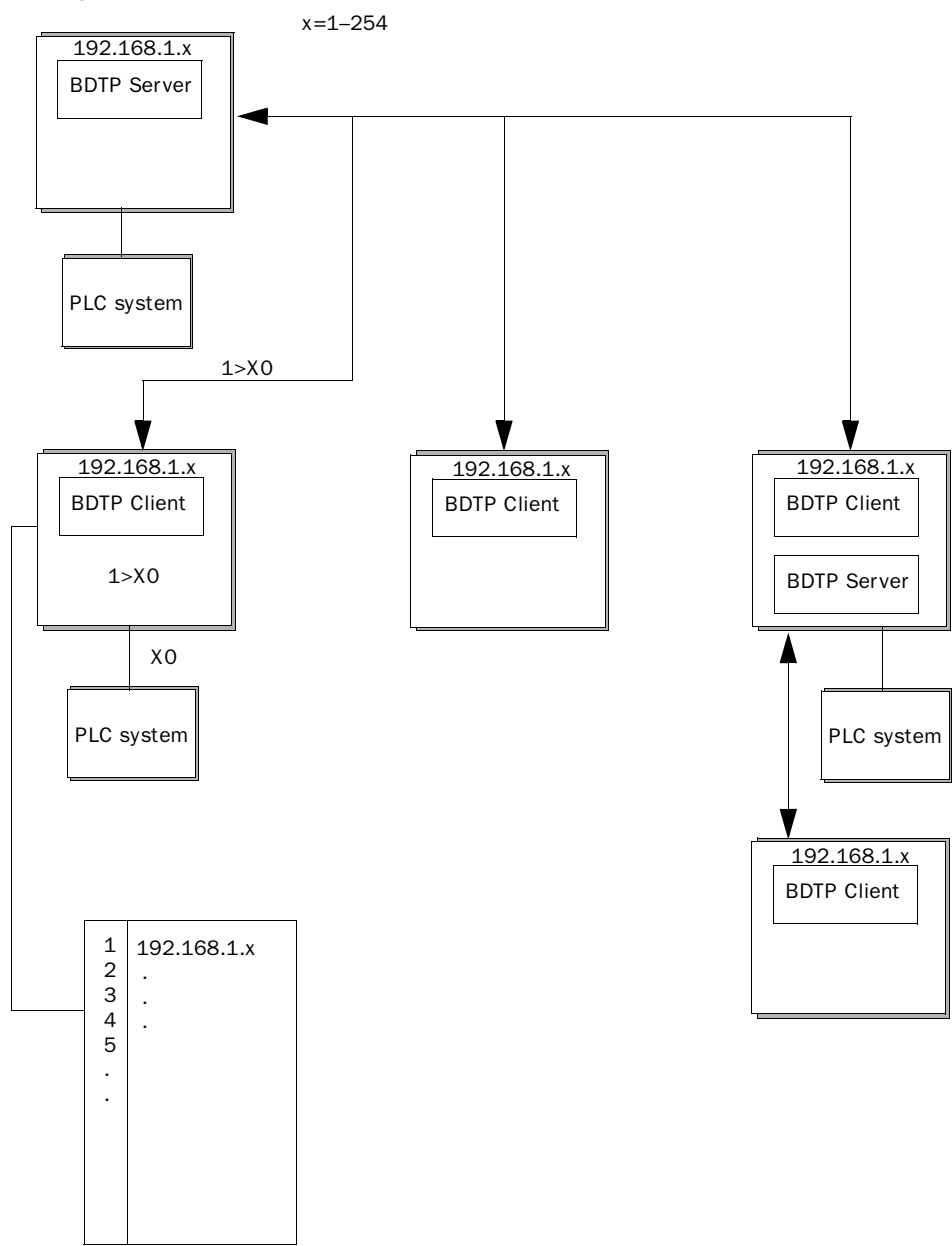
Function for transferring a project via TCP/IP. Press Edit and enter the number of the port that will enable the transfer. Does not normally need to be changed.

### BDTP (Beijer Data Transport Protocol)

BDTP (Beijer Data Transport Protocol) replaces the old MAC network. The BDTP server will listen for I/O request from the BDTP client. The terminal can be client, server or both. The client can receive data from max 16 servers. The IP addresses of the servers are stated in the BDTP client.

Network communication via BDTP is used to connect more than two terminals to a PLC system and simultaneously maintain good performance. An example of use is production lines, where you need one terminal at each workstation.

The figure below shows an example of a network.



## BDTP Client

In the network service BDTP Client you define IP addresses for those BDT servers from which the client can retrieve information. When you press **Edit** the following dialog is shown.

**Setup BDTP Client service**

General setup

BDTP Server port: 1026

Default BDTP Server: None

Data register: [ ] I/O

Control block: [ ] I/O

☒ Synchronize clock with server: 1

BDTP Server connections

BDTP Server Address: [ ]

Index	BDTP Server

Update  
Append  
Delete

Ok Cancel

### BDTP Server port

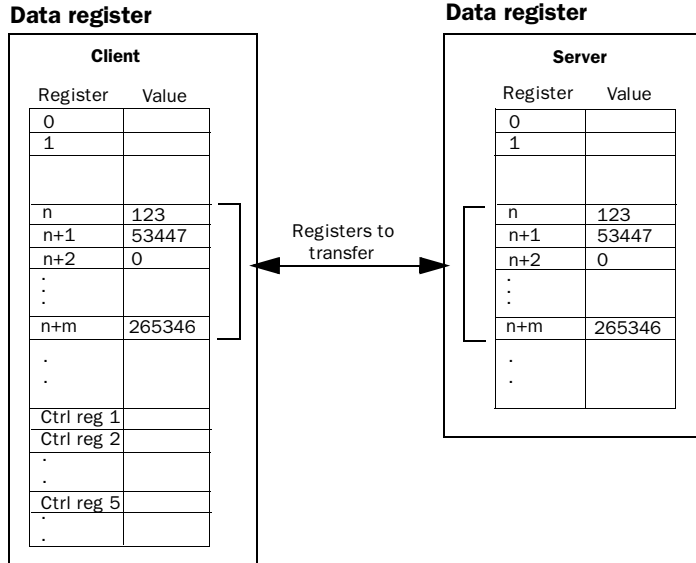
State the communication port to which the BDTP server/network is connected. Does not normally need to be changed.

### Default BDTP Server

Here you can specify a default BDTP server. If you do not specify the address, the client will get the value from the default server.

### Data register

Values in data registers can be transferred between one client and different servers in a network. Under Data register you state the first register in the register block in the client to be transferred to/from the stated server. The type of register must not be the same in the client and the server.



## Control block

Under Control Block you state the first register in the control block in the client. The control block takes five registers as follows.

Register	Content	Description
Ctrl reg 1	Command	Command register which is set in the client. Available commands: 0: No command. 1: Transfer register values from the client to the server stated in control register 3. 2: Transfer register values from the server stated in control register 3 to the client.
Ctrl reg 2	Handshake	Handshake register which is set by the client. 0: Ready for a new command. 1: OK 2: Transfer failed.
Ctrl reg 3	Server index	The number of the server in the network handling the transfer.
Ctrl reg 4	Index register	The value in the index register is added to the address of the register stated under Data register. If zero is stated in the register, the block will start with the address stated under Data register.

Register	Content	Description
Ctrl reg 5	Number of registers	The number of registers the values of which shall be transferred to/from the stated server.

The transfer handling must take place according to the following:

1. The handshake register must be 0. If it is not, make sure that the command register is set to 0.
2. Write the command in the command register.
3. Wait for completion or error code in the handshake register.
4. Set the command register to 0. After this, the terminal will set the handshake register to 0.

### **Synchronize clock with server**

State if the clock in the client shall be synchronized with the selected server. Select server by stating the number of the server in the entry field. If the clock in the client is changed locally, the change is transferred to the server selected.

### **BDTP Server Address**

Under BDTP Server Address you define the addresses to the servers from which the client can retrieve information. The addresses will be indexed in the order they are stated.

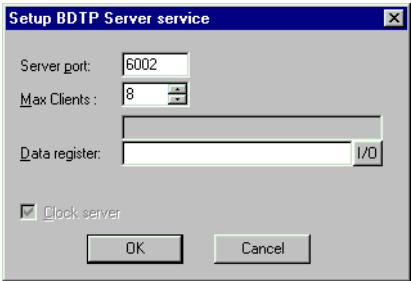
When the object is defined in the project you must specify the source server for the address. in the objects address field specify the following:  
server index>device

If, for example, 2>D15 is entered in the address field the value for the object is loaded from register D15 in server 2.

It is possible to change the server index in a client project with the BDTP Station change function. See "Change BDTP station".

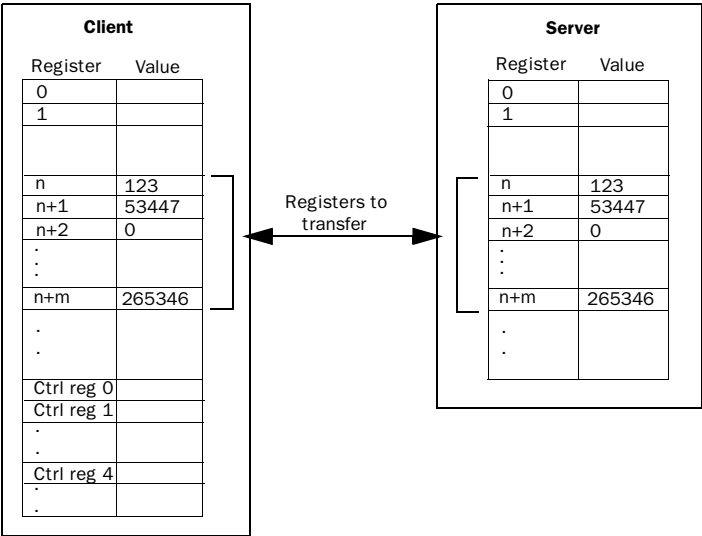
## BDTP Server

Handles the request from clients. Press **Edit** and state port. Does not normally need to be changed.



## Data register

Values in data registers can be transferred between one server and different clients in a network. Under Data register you state the first register in the register block in the server to be transferred to/from the client stated. The type of register must not be the same in the client and the server. For further information about data transfer see the “BDTP Client” section.



### Note!

It is only possible to control data transfer from clients.



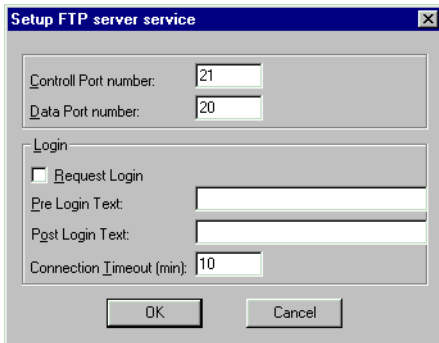
## Clock Server

State if the clients in the network shall synchronize with the clock in current server. See also the “BDTP Client” section. This function is only valid for older terminal versions.

## FTP Server

With this function it is possible to fetch and leave files to and from the terminal. The FTP server in the terminal allows transfer in passive mode (PASV). Passive mode should be used if you **do not** connect the terminal point-to-point. You do not always know what is between the client and the server, for example, router-based fire wall or gateways. By using passive mode you eliminate several errors. Browsers always use this mode. It is possible to use passive mode at point-to-point connection. National characters are not supported in file names.

Select **FTP Server** under **Network services** and click on **Edit...** to enter the settings for the function.



### Control Port number

Default is 21 and should not be changed.

### Data Port number

Default is 20 and should not be changed.

### Request Login

Here you state if the user must login to get access to the FTP server. Users are defined under Accounts. If you select that no user need to login all users has full access to the FTP server.

### Pre Login Text

A text shown before request of user is shown.

**Post Login Text**

A text showing when the user has logged in.

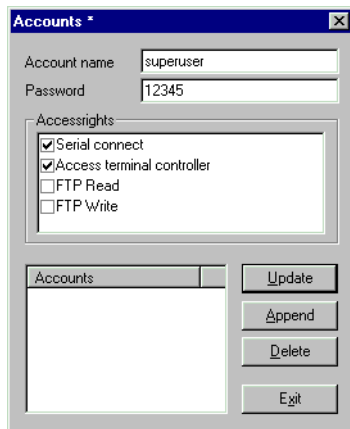
**Connection Timeout**

The time the FTP connection may be inactive before the server disconnects the connection. Default is 10 minutes.

**Accounts**

Under **Accounts** you define user accounts with which you decide if the user will have access to the FTP server.

If no password is defined for the account you can enter any password or leave the entry field empty when logging in.

**FTP Access (Read)**

The user may use the FTP server (read).

**FTP Write**

The user may write to the FTP server. Requires Access.

## SMTP Client

This function makes it possible to send email from the terminal. Trend and recipe files can be sent as attached files to the message. Max 20 messages can be sent at the same time. Select **SMTP Client** under **Network services** and click on **Edit...** Then you enter the following settings.

The screenshot shows the 'Setup SMTP Client service' dialog box. It has a title bar with a close button. The main area contains several input fields: 'Server port' with the value '25', 'Mail Server', 'My Domain Name', and 'My Email Address'. Below these is a dropdown menu for 'Send via Connection' showing 'TCP/IP Connection 1'. A section titled 'Predefined Recipients' includes an 'Email Address' input field and a table with columns 'Index' and 'Email Addresses'. To the right of the table are 'Update', 'Append', and 'Delete' buttons. At the bottom are 'Ok' and 'Cancel' buttons.

### Server port

Connection port 25.

### Mail server

The IP address or the alias name (DNS server) for the SMTP mail server. If you enter an alias name you must enter the IP address to the DNS server in the TCP/IP Network Connection dialog under Network in the Setup menu.

### My Domain Name

The domain name of the terminal or any other domain name used to login on the SMTP server.

### My Email Address

Enter your email address. The name is shown as sender for the receiver. Please enter a real email address so that the mail server can return any error messages.

**Send via Connection**

State the connection used for sending. Note that Connection 1 must be used before Connection 2 can be used.

**Recipients**

A predefined list of max 16 receivers. A receiver address can be max 60 characters long.

**Alarm via email**

Alarms can be printed in the same way as they can be sent as email. The whole alarm list will be sent if you send block 990 (see Block Mail Report).

**Alarms: Project5.mpa**

Alarm text: High level in tank 5

Signal: M5 I/O

Alarm when signal is: ☒ On ☐ Off

Acknowledge notify: I/O ☐ Reset

Remote acknowledge: I/O

Alarm group: DEFAULT

Info block:

Mail to address: bill.wright@yahoo.com

No.	Signal	Alarm Text
1	M5	High level in ta
2	M33	Low temperatu
3	M8	Conveyor empt
4	M82	Conveyor stopp

☒ Ack. required  
☒ History  
☐ To printer

Update  
Append  
Insert  
Delete  
Exit

**Mail to address**

Enter the receiver of the message. Select up to eight receivers from the predefined list in the SMTP Client Service dialog.

**Info block**

If you state an information block and it is a textblock it will be included in the message. For more information see "Alarm handling".

## Report via email

Textblock can be printed in the same way as they can be sent as email.

---

### Note!

Only textblock can be sent. Only the alarm block,990, of the system blocks is sent as email. Trend and recipe files are sent as attached files.

---

### Block name

If you enter a name of a textblock it will be sent as a message/subject.

### Send mail signal:

The message is sent when signal stated is set.

### Mail completion signal:

A digital signal that is activated by the terminal when the message is sent. Normally the signal is set. If you mark the Reset function the signal will be reset when the message is sent.

**Mail to address:**

Enter the email address to the receiver. Select up to eight receivers from the list shown when you click on the ...-button. The list of email addresses is defined in the dialog **SMTP Client Service** under **Network Services** in the **Setup** menu.

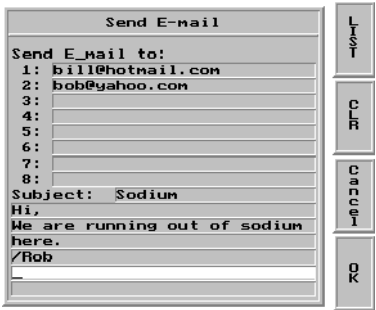
**Attach file:**

Enter a name of a trend or recipe file to be attached with the message. If there are a trend file and a recipe file with the same name the trend file will be attached. The file name may not contain national characters.

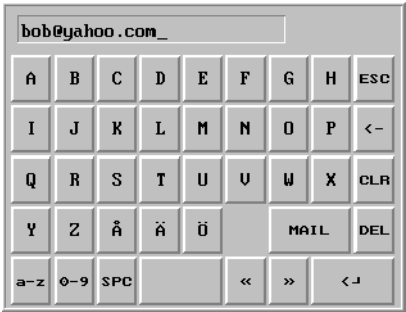
**Email via system block**

By making a jump to the Mail system block, 993, you can write and send a mail during run-time.

E900



E910



**Send E-mail to**

Enter the receiver. You can enter the address or select from the global list shown when you click the LIST button on a terminal with keyboard or on the MAIL button on a terminal with touch screen.

### Subject

Enter the subject for the message. Max 50 characters. The text in the message can be max 10 rows of 50 characters.

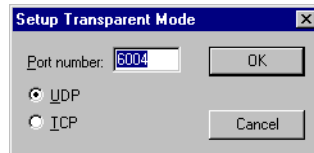
## Terminal Controller

Used for RUN/TRANSFER switching via TCP/IP. Press **Edit** and enter the number of the port to be stated to enable transfer. The port number does not normally need to be changed. Mark Request authentication if user name and password must be stated before transfer..



## Transparent mode

Used for communication in Transparent mode (see also the chapters "Communication" and "Network functions in the terminal") in terminal networks. Press **Edit** and enter the number of the port stated to enable the communication. The port number need not normally be changed. Select protocol, UDP or TCP.



## WWW Server

With this function you can configure the WWW-server in the terminal. A Web server is a program that, using the client/server model and the World Wide Web's Hypertext Transfer Protocol (HTTP), serves the files form Web pages to Web users (whose computers contain HTTP clients that forward their request). See the chapter "Network functions in the terminal".

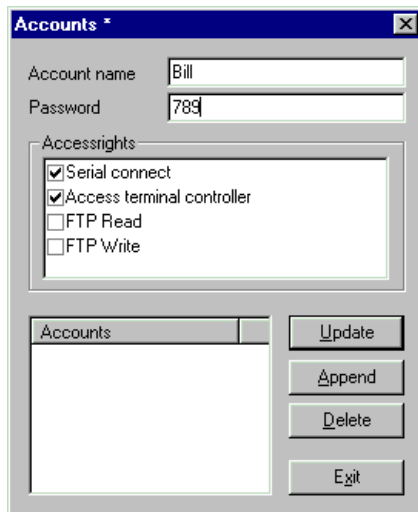


### Port number

Default is 80 and should not be changed.

### Account name

By selecting an account name you protect the HTML pages with a password. Accounts are defined under **Network.. Accounts** in the **Setup** menu.



### Account name

Shows the Account name entered in previous dialog.



**Password**

Enter a password. All HTML pages are protected with this Account name/password.

To protect a single page with another Account name/Password you enter the following in the header of the HTML page.

```
<HTML>
<HEAD>
<META name="superuser" content="12345"*
</HEAD>
```

Here you place the rest of the HTML code.

```
</HTML>
```

\* "superuser" is the user name and "12345" is the password.

---

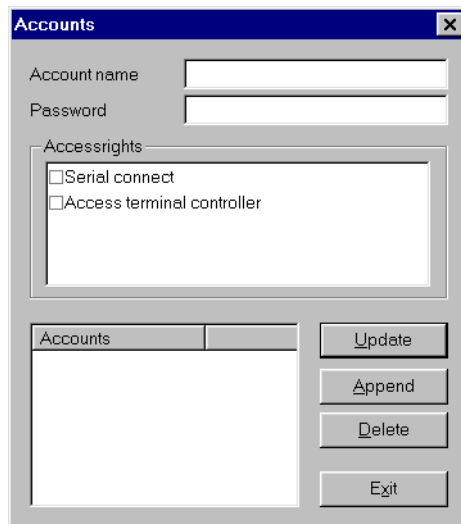
**Note!**

The code above must be found in the header. The Name and Content parameters must correspond to an Account name and a Password.

---

## 18.5 Network accounts

**Accounts** define who may connect to services available at terminals requiring logon. The function is used to create access control. You create a name and password for different users who will have access to different network services. Account name and password may not contain national characters.



According to the figure the account name Connection X has access to all network functions that requires login.



## **19 Network functions in the terminal**

This chapter describes how the FTP server, SMTP client, applet and WWW server functions work in the terminal. For settings in the respective functions we refer to the chapter “Network communication”.

### **19.1 FTP server**

FTP (File Transport Protocol), a standard Internet protocol, is the simplest way to exchange files between computers (terminals) on the Internet. FTP is an application protocol that uses the Internet’s TCP/IP protocols. FTP is commonly used to transfer Web page files from their creator to the computer that acts as their server for everyone on the Internet. It is also commonly used to download programs and other files to your computer from other servers.

Some directories show files with the file length 0. This means that the file contains dynamic data i.e. the size of the file change. This means that the file is not empty even if the length is 0. The terminal does not use a date on the files which means that the time shown is not relevant. The file name may not contain national characters. The FTP server in the terminal can handle three connected clients.

---

**Note!**

All files in the different directories in the ROOT directory charge the project memory. Information about project memory available can be found in the info.text file located in the ROOT directory.

---

### **The ROOT directory**

In the root directory you will find the following directories ALARMS, HTML, RECIPE and TRENDS. Only the libraries that the user has access to are shown. You will also find the file info.txt containing information about the terminal here.

### **The file info.txt**

The file contains information about the terminal according to the following example:

E900

Boot version: 4.01

Firmware version: V5.00

Build number: 124

Driver: FX-SERIES (CPU) V1.12.0

Dynamic memory: 304237 bytes free

Project memory: 184700 bytes free

Nothing can be deleted from the root directory, even if you have write access. If you delete the directories HTML or RECIPE, the content of the directories is deleted but the directories are still there.

### **The ALARMS directory**

The directory is only shown if there are any alarms defined in the terminal and the terminal is in run-time mode. The alarm groups are shown as SKV files with the length 0. This does not mean that there are not any alarms. The files can only be read. To read a file it must be opened with the attribute READ, READWRITE will cause error.

#### **File format**

Each alarm is stored in a row ending with <carriage return><linefeed>.

Syntax

*status;activedate;activetime;inactivedate;inactivetime;ackdate;acktime;alarmtext*

All fields are always there. If the alarm is not acknowledged the fields for date and time are empty.

### **The HTML directory**

The data files handled by the Web server are stored here. You can create subdirectories. The start file must always be named index.htm.

#### **File format**

The file format depends on the type of file. We use standardized file formats such as HTML etc.

## The RECIPE directory

The different recipes are shown as SKV files with the length 0 which does not mean that the recipe is empty. You can both read and write files in this directory. But you may not create a directory.

### File format

Each recipe value are stored in a row ending with <carriage return><line-feed>.

Syntax

*device;value;datatype:length*

The file ends with

END<carriage return><linefeed>

If the data type is array (AR) one value is shown on each line. The first line will be shown as above. The following lines will only contain

*;value*

### Data types for analog devices

Type	Description
tom	Signed 16-bit
+	Unsigned 16-bit
L	Signed 32-bit
L+	Unsigned 32-bit
RB	BCD flyttal
RF	Float with exponent
SB	BCD 16-bit
LB	BCD 32-bit
SH	Hexadecimal 16-bit
LH	Hexadecimal 32-bit
RD	Float
AR	Array of signed 16-bit
ST	String of characters
BI	Bit 0 or 1

## The TRENDS directory

This directory is only available if there are trends defined in the terminal and the terminal is in run-time mode. The trend files are shown as SKV files with the length 0. The files can only be read. To read a file it must be opened with the attribute READ, READWRITE will cause error. Curve 1 in a trend must be used, otherwise the trend is not valid.

### File format

Each sampling is stored in a row ending with <carriage return><line-feed>.

Syntax

*date;time;value1;value2;value3;value4;value5;value6;OFF*

End the file with

END<carriage return><linefeed>

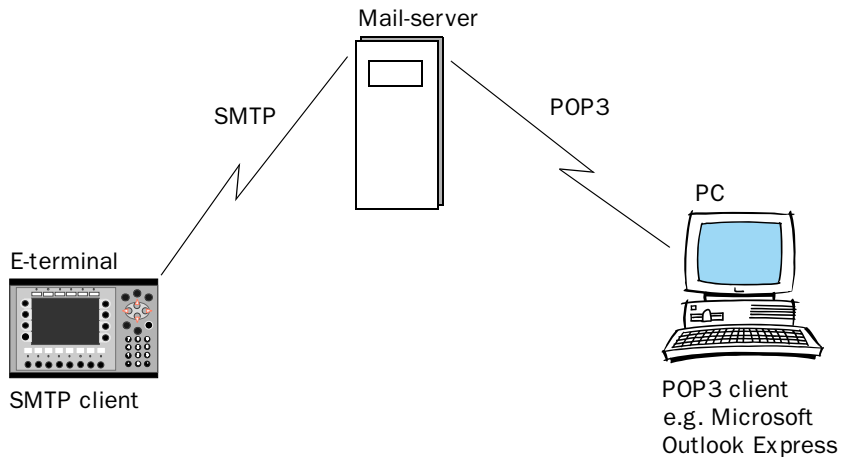
Only used curves in the trend are transferred, i.e. no empty fields.

OFF is in the sample in the following cases and indicates a pause in the sampling.

- When the terminal enters run-time mode. A copy of the latest stored sample is added. The copy has OFF indication. As soon as valid values reach the terminal new values without OFF indication are stored.
- When the signal Trend-enable is used. If the signal is reset a sample is indicated with OFF. When the signal is set a new value is stored without OFF indication.
- When you transfer stored trend values via FTP or HMI tools, a sample will be stored with OFF indication. When the transfer is ready a new value is stored without OFF indication.

## 19.2 SMTP client

SMTP (Simple Mail Transfer Protocol) is a TCP/IP protocol used for sending and receiving email. However, since it is limited in its ability to put messages in a queue at the receiving end, it is usually used with one of two other protocols, POP3 or IMAP, that let the user save messages in a server mailbox and download them periodically from the server. In other words, users typically have a program that uses SMTP for sending email and either POP3 or IMAP for receiving messages that have been received for them at their local server.





## 19.3 Terminal reflection – applet

An applet is a little application program. On the World Wide Web, using Java, the object-oriented programming language, an applet, is a small program that can be sent along with a Web page to a user. Java applets can perform interactive animations, immediate calculations, or other simple tasks without having to send a user request back to the server.

From a PC it is possible to reflect the terminal in a browser, e.g. Microsoft Internet Explorer or Netscape Navigator. This is done by an application program presenting a picture of the terminal on the screen. The picture is updated at continuous time intervals according to stated requests. It is possible to affect by clicking with the mouse on the buttons in the terminal picture or using the keyboard. On a touch screen terminal you click directly on the screen.

The Applet application is a jar-file in the terminal file system, in the HTML directory, and is transferred to the terminal via FTP. The file is unpacked when the Web page it is linked to is activated, see the following example. The communication with the terminal is done with the Terminal control function (port 6001 predefined). The port number can be configured in the source code for the Web page via a parameter in the applet application. It is also possible to change the background color behind the terminal picture on the Web page. You connect to the WWW server in the terminal by entering the IP address in the browser. When the Web page is loaded you will find a terminal symbol with the text Terminal Interaction. The applet application starts when you click on the symbol. Now a login dialog is shown. If the login is defined for the Terminal control function entered, user parameters are verified with the users defined under Accounts. Otherwise you do not need to state anything in this dialog.

### Example

```
<applet
  code=TERMAPPL.class
  archive="e900appl.jar"
  width=117
  height=101>
  <param name=TermCtrlPort value="6001">
  <param name=Background value="DB66FF">
</applet>
```

**Note!**

If the “background value” parameter is used the value field must be filled in with a RGB color code. The value field must not be empty.

## 19.4 Web server

A Web server is a program that, using the client/server model and the World Wide Web’s Hypertext Transfer Protocol (HTTP), serves the files form Web pages to Web users (whose computers contain HTTP clients that forward their requests). Every computer (terminal) on the Internet that contains a Web site must have a Web server program.

### SSI script (server-side include)

A server-side include is a variable value (for example a file) that a server can include in an HTML file before it is sent to the requestor. If you create a Web page, you can insert an attached statement in the HTML file that looks like this:

```
<!--#echo var="LAST_MODIFIED"-->
```

To show values from the terminal on Web pages we support the following SSI script.

Name	Parameters	Description	Example
get_ipaddr.fn	None	Shows the IP address for the Web server. Used in CGI script	<!--#exec cgi="get_ipaddr.fn"-->
get_domainname.fn	None	Shows the Web server domain name	<!--#exec cgi="get_domainname.fn"-->

Name	Parameters	Description	Example
get_date.fn	Date format e.g. MM/DD/YY YY-MM-DD If no format is state the terminal setting is used	Shows the date in the terminal	<!--#exec cgi="/get_date.fn MM/DD/YY"-->
get_time.fn	Time format e.g. HH:MM:SS HH:MM If no format is stated the terminal setting is used	Shows the time in the terminal	<!--#exec cgi="/get_time.fn HH:MM"-->
get_device.fn	X, Y, Z X=device Y=presentation format (see separate table) Z=length	Shows the device value from the PLC-system.	<!--#exec cgi="/get_device.fn D5"--> <!--#exec cgi="/get_device.fn D5LH"--> <!--#exec cgi="/get_device.fn M7"--> <!--#exec cgi="/get_device.fn D9ST,30"--> <!--#exec cgi="/get_device.fn D0AR,10"-->
get_diag.fn	None	Shows the diagnostic page of the terminal	<!--#exec cgi="/get_diag.fn"-->
get_mode.fn	None	Shows the mode of the terminal: RUN/PROG/SETUP/TRANSFER	<!--#exec cgi="/get_mode.fn"-->

## Presentation format for get\_device.fn

Name	Length	Description	Example
None	None	Shows the value as signed 16-bit	<!--#exec cgi=/get_device.fn D1"-->
+	None	Shows the value as unsigned 16-bit	<!--#exec cgi=/get_device.fn D3+"-->
L	None	Shows the value as signed 32-bit	<!--#exec cgi=/get_device.fn D7L"-->
L+	None	Shows the value as unsigned 32-bit	<!--#exec cgi=/get_device.fn D2L+"-->
RB	None	Shows the value as 32-bit float BCD (MELSEC, SIMATIC)	<!--#exec cgi=/get_device.fn D10RB"-->
RF	None	Shows the value as 32-bit IEEE float	<!--#exec cgi=/get_device.fn D8RF"-->
RD	None	Shows the value as 32-bit IEEE float without exponent	<!--#exec cgi=/get_device.fn D1RD"-->
SB	None	Shows the value as 16-bit BCD	<!--#exec cgi=/get_device.fn D3SB"-->
LB	None	Shows the value as 32-bit BCD	<!--#exec cgi=/get_device.fn D7LB"-->
SH	None	Shows the value as 16-bit HEX	<!--#exec cgi=/get_device.fn D2SH"-->
LH	None	Shows the value as 32-bit HEX	<!--#exec cgi=/get_device.fn D1LH"-->
AR	Number of values to be shown	Shows the value as signed 16-bit	<!--#exec cgi=/get_device.fn D5AR,10"-->

Name	Length	Description	Example
ST	Number of characters in the string	Shows a number of registers as string	<!--#exec cgi=/get_device.fn D9ST,30"--> <-->

### Automatic updating

Normally the Web page is not automatically updated. By adding the following, however, you create automatic updating.

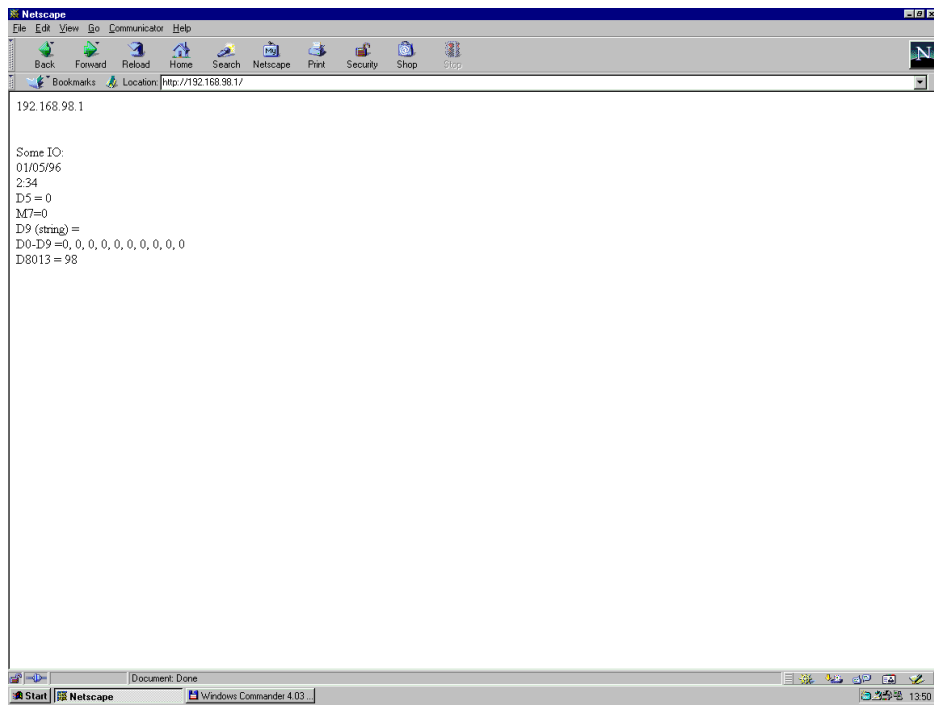
```
<META HTTP-EQUIV="Refresh"CONTENT="5">
```

CONTENT states how often the page will be updated (in seconds).

### Example of a Web page with SSI script

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Refresh"CONTENT="1">
</HEAD>

<!--#exec cgi="/get_ipaddr.fn"--><BR>
<!--#exec cgi="/get_domainname.fn"--><BR>
<BR>
Some IO:<BR>
<!--#exec cgi="/get_date.fn MM/DD/YY"--><BR>
<!--#exec cgi="/get_time.fn HH:MM"--><BR>
D5 = <!--#exec cgi="/get_device.fn D5"--><BR>
M7=<!--#exec cgi="/get_device.fn M7"--><BR>
D9 (string) = <!--#exec cgi="/get_device.fn D9ST,30"--><BR>
D0-D9 =<!--#exec cgi="/get_device.fn D0AR, 10"--><BR>
D8013 = <!--#exec cgi="/get_device.fn D8013"--><BR>
</HTML>
```



## CGI script (common gateway interface)

The common gateway interface (CGI) is a standard way for a Web server to pass a Web user's request to an application program and to receive data back and forward it to the user. When the user requests a Web page (for example, by clicking on a highlighted word or entering a Web site address), the server sends back the requested page. When a user fills out a form on a Web page and sends it in, it usually needs to be processed by an application program. The Web server typically passes the form information to a small application program that processes the data and may send back a confirmation message. This method for passing data back and forth between the server and the application is called the common gateway interface (CGI). It is part of the Web's HTTP protocol.

To change values in the terminal we support the following CGI script:

Name	Parameters	Description	Example
set_date.fn	Date format e.g. MM/DD/YY YY-MM-DD If no format is stated the terminal setting is used	Used together with FORM to set the date in the terminal.	<code>&lt;FORM ACTION="http://&lt;!--#exec cgi="/get_ipaddr.fn"--&gt;/set_date.fn" METHOD="POST"&gt; &lt;INPUT SIZE=10 MAXLENGTH=10 NAME="YY:MM:DD"&gt; &lt;INPUT TYPE="submit" VALUE="Submit"&gt; &lt;/FORM&gt;</code>
set_time.fn	Time format e.g. HH:MM:SS HH:MM If no format is stated the terminal setting is used	Used together with FORM to set the time in the terminal.	<code>&lt;FORM ACTION="http://&lt;!--#exec cgi="/get_ipaddr.fn"--&gt;/set_time.fn" METHOD="POST"&gt; &lt;INPUT SIZE=10 MAXLENGTH=10 NAME="HH:MM:SS"&gt; &lt;INPUT TYPE="submit" VALUE="Submit"&gt; &lt;/FORM&gt;</code>

Name	Parameters	Description	Example
set_device.fn	XY X = device Y = presentation format (see separate table) e.g. D0L+ D5SH	Used together with FORM to set an device in the PLC system.	<FORM ACTION="http://<!--#exec cgi="/get_ipaddr.fn"->/set_device.fn" METHOD="POST"> <INPUT SIZE=10 MAXLENGTH=10 NAME="D0L"> <INPUT TYPE="submit" VALUE="Submit"> </FORM>
set_mode.fn	RUN PROG SETUP TRANSFER	Used together with FORM to set the terminal mode.	<FORM ACTION="http://<!--#exec cgi="/get_ipaddr.fn"->/set_mode.fn" METHOD="POST"> <SELECT NAME="MODE"> <OPTION VALUE="RUN">Run <OPTION VALUE="PROG">Prog <OPTION VALUE="SETUP">Setup <OPTION VALUE="TRANSFER">Transfer </SELECT> <INPUT TYPE="submit" VALUE="Submit"> </FORM>



Name	Parameters	Description	Example
push_key.fn	(see separate table)	Used to simulate pressing a on the terminal.	<pre> &lt;FORM ACTION="http://&lt;!-- #exec cgi="/get_ipaddr.fn"--&gt;/ push_key.fn" METHOD="POST"&gt; &lt;SELECT NAME="F2"&gt; &lt;OPTION VALUE="SET"&gt;Set &lt;OPTION VALUE="RESET"&gt;Reset &lt;OPTION VALUE="TOGGLE"&gt;Toggle &lt;/SELECT&gt; &lt;INPUT TYPE="submit" VALUE="Submit"&gt; &lt;/FORM&gt; &lt;FORM ACTION="http://&lt;!-- #exec cgi="/get_ipaddr.fn"--&gt;/ push_key.fn" METHOD="POST"&gt; &lt;INPUT SIZE=1 MAXLENGTH=1 NAME="Key"&gt; &lt;INPUT TYPE="submit" VALUE="Submit"&gt; &lt;/FORM&gt; </pre>

Presentation format for set\_device.fn

Name	Description
None	Sets the value as signed 16-bit
+	Sets the value as unsigned 16-bit
L	Sets the value as signed 32-bit
L+	Sets the value as unsigned 32-bit
RB	Sets the value as 32-bit float BCD (MELSEC, SIMATIC)
RF	Sets the value as 32-bit IEEE float
RD	Sets the value as 32-bit IEEE float without exponent
SB	Sets the value as 16-bit BCD
LB	Sets the value as 32-bit BCD
SH	Sets the value as 16-bit HEX
LH	Sets the value as 32-bit HEX
ST	Sets a number of registers as a string

## Parameters for push\_key.fn

Parameter	Description	Example
KEY	Can state the values: A-Z 0-9 ACK LIST MAIN PREV BACKSPACE ENTER UP DOWN LEFT RIGHT	<pre> &lt;FORM ACTION="http://&lt;!--#exec cgi="/ get_ipaddr.fn"--&gt;/push_key.fn" METHOD="POST"&gt; Key = &lt;SELECT NAME="Key"&gt; &lt;OPTION VALUE="ENTER"&gt;Enter &lt;OPTION VALUE="A"&gt;A &lt;OPTION VALUE="B"&gt;B &lt;OPTION VALUE="1"&gt;1 &lt;OPTION VALUE="2"&gt;2 &lt;OPTION VALUE="3"&gt;3 &lt;OPTION VALUE="UP"&gt;Up &lt;OPTION VALUE="DOWN"&gt;Down &lt;OPTION VALUE="LEFT"&gt;Left &lt;OPTION VALUE="RIGHT"&gt;Right &lt;OPTION VALUE="PREV"&gt;Prev &lt;/SELECT&gt; &lt;INPUT TYPE="submit" VALUE="Submit"&gt; &lt;P&gt; &lt;/FORM&gt; </pre>
F1-F22	Can state the values: SET RESET TOGGLE	<pre> &lt;FORM ACTION="http://&lt;!--#exec cgi="/ get_ipaddr.fn"--&gt;/push_key.fn" METHOD="POST"&gt; &lt;SELECT NAME="F2"&gt; &lt;OPTION VALUE="SET"&gt;Set &lt;OPTION VALUE="RESET"&gt;Reset &lt;OPTION VALUE="TOGGLE"&gt;Toggle &lt;/SELECT&gt; &lt;INPUT TYPE="submit" VALUE="Submit"&gt; &lt;/FORM&gt; </pre>

### **Example on a Web page with SSI and CGI script:**

```
<HTML>

<FORM ACTION="http://<!--#exec cgi="/get_ipaddr.fn"-->/
set_date.fn" METHOD="POST">

Set date here (YY:MM:DD):

<INPUT SIZE=10
      MAXLENGTH=10
      NAME="YY:MM:DD"
      VALUE="<!--#exec cgi="/get_date.fn"-->">

<INPUT TYPE="submit" VALUE="Submit"> <P>

</FORM>

<FORM ACTION="http://<!--#exec cgi="/get_ipaddr.fn"-->/
set_time.fn" METHOD="POST">

Set time here (HH:MM:SS):

<INPUT SIZE=10
      MAXLENGTH=10
      NAME="HH:MM:SS"
      VALUE="<!--#exec cgi="/get_time.fn"-->">

<INPUT TYPE="submit" VALUE="Submit"> <P>

</FORM>

<FORM ACTION="http://<!--#exec cgi="/get_ipaddr.fn"-->/
set_device.fn" METHOD="POST">

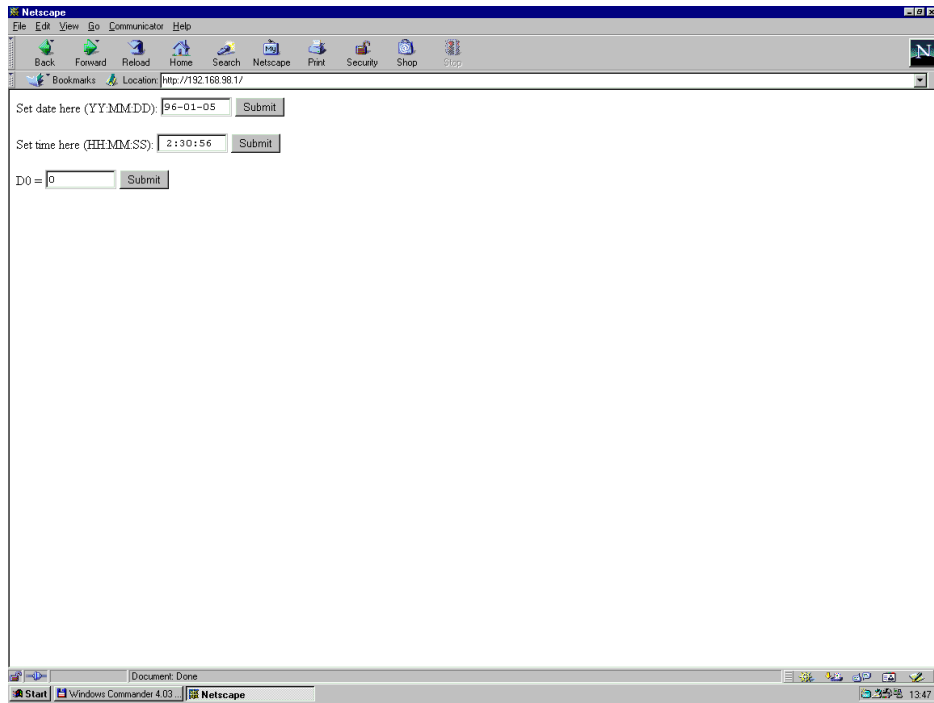
D0 =

<INPUT SIZE=10
      MAXLENGTH=10
      NAME="D0"
      VALUE="<!--#exec cgi="/get_device.fn D0"-->">

<INPUT TYPE="submit" VALUE="Submit">

</FORM>

</HTML>
```



## Save HTML files with FTP

To save HTML files in the terminal you use a standard FTP client software (see the section FTP server).

The files are stored in the HTML directory in the terminal.

The name of the files must be in DOS 8.3 format i.e. the name can contain max 8 characters and the extension must be HTM.

---

### **Note!**

The file INDEX.HTM must be available.

---

## **19.5 Recommendations and limitations for network communication**

To make the communication between terminals and controller systems in a terminal network (BDTP network) quick and efficient, it is important to optimize the transfer of signals in the network. Read the chapter “Efficient communication” and follow the rules described to optimize the network functionality in the terminals. The maximum number of signals that can be transferred in a terminal network are 3000.

### **Example 1**

A terminal network consist of three clients and one server. Each client has access to 1000 signals which means that the server provides the clients with 3000 signals. This is valid even if the address areas for the signals in each client are in accordance with each other. Accordingly it is not possible to transfer more signals in this terminal network.

### **Example 2**

The server collect all addresses the clients are asking for. Therafter the server asks for the status of the controller system that is to be distributed to respective client.

Example

A terminal network (BDTP network) consists of one server and five clients. Each terminal contains 50 alarms with the same address. For the server, this means that 50 addresses must be checked with the controller system. The server shall then distribute 50 alarms to respective client (5x50). This means that the server distribute 250 alarms.

## Transparent mode

To be able to make the Transparent mode function (see also the chapter “Communication”) work in a terminal network with communication via the TCP/IP protocol the following demands must be met.

- The driver and its programming tool must support communication in Transparent mode. For information we refer to the manuals for the driver and controller system and also the chapter “Communication” in this manual.
- A PC program for COM port to TCP/IP redirection must be used in the PC communication with the controller system in Transparent mode via the TCP/IP network.

The programming tools MELSEC MEDOC and MELSEC MEDOC *plus* for PLC systems from Mitsubishi Electric support Transparent mode communication in TCP/IP network with MELSEC PLC systems according to the following table.

Programming tool	PLC system
MELSEC MEDOC	All MELSEC PLC-system except QnA-CPU and QnA-C24
MELSEC MEDOC <i>plus</i>	MELSEC A C24, MELSEC FX-series

## No protocol mode

The No protocol mode function (see the chapter “Communication”), when the terminal is used as communication interface, is not recommended to be used in terminal networks (BDTP networks). A BDTP network is a network where many signals are transferred between the server and clients. When the terminal is used as a communication interface, control registers and control signals are transferred and they affect the communication time and the performance of the network. See also the section Efficient communication in the chapter “General”.

## **Packing of signals**

To make the communication between terminals and controller systems in a terminal network (BDTP network) quick and efficient it is important to optimize the transfer of signals in the network. Read the section Efficient communication in the chapter “General” and follow the rules described to optimize the network functionality in the terminals. This is valid for all stations in a terminal network. If you do not use packing of signals, the time between updates of stations in the network will increase.

## **Alarm handling**

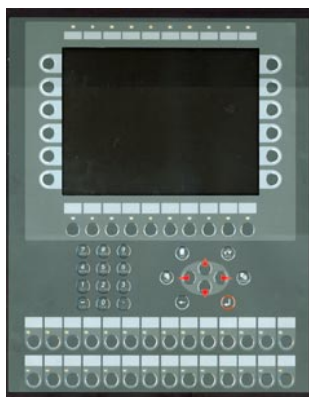
The terminal network is a Client/Server network. Servers provide data, for example, alarm signals to Clients. Many different signals affects the communication time between terminals and controller systems in the network. This means that it is expedient to limit the number of signals affecting the communication. For more information see the section Efficient communication in the chapter “General”. The number of alarm signals in the network cannot exceed what one server can handle in the network. One server can handle between 100 to 300 alarms, depending on application and terminal, consequently the network cannot contain more than 100 to 300 alarms.

## 20 LEDs

The E900 has 20 LEDs built in, ten are located near the function keys and ten are independent. The LEDs are connected to a registers defined under LED in the **Functions** menu. The content of the register determines the color and any blinking of the LED according to the following table.

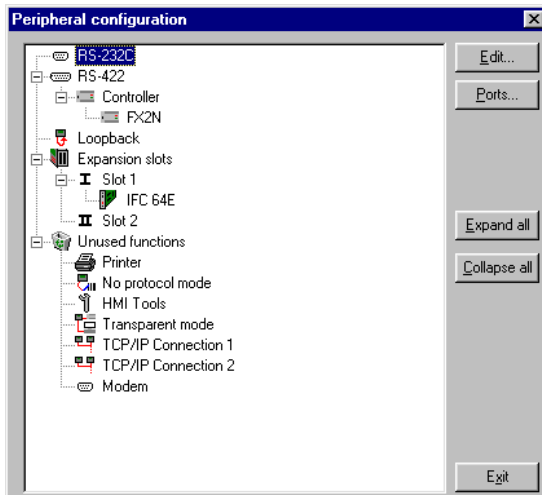
Register value (Hex)	Register value (Dec)	Blinking frequency (Hz)	Color
00	0	-	None
01	1	-	Green
02	2	-	Red
11	17	5	Green
12	18	5	Red
21	33	2.5	Green
22	34	2.5	Red
31	49	1.2	Green
32	50	1.2	Red
41	65	0.6	Green
42	66	0.6	Red

### 20.1 External LEDs in E900 VT





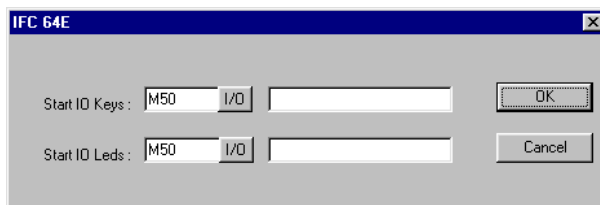
The terminal E900 VT has 28 external LEDs, supported by the built in expansion card IFC 64E. This expansion card is automatically selected at slot 1 when the terminal E900 VT is chosen in project settings in the **File** menu.



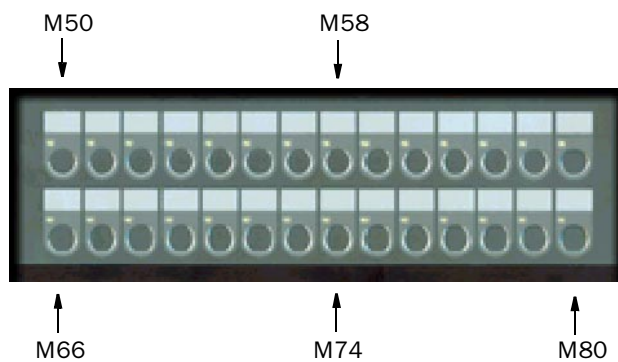
This extra built-in LEDs can only be connected to digital I/O signals, bit devices, and can not get analog values from the terminal.

To type in the desired start I/O for the LEDs, right click on IFC 64E in Peripheral configuration under Peripherals in the Setup menu.

The expansion card IFC 64E handles 32 LEDs, the terminal E900 VT only use 28. There are 32 digital signals reserved consecutive from the start address. The LEDs are grouped together in 4 groups with 7 LEDs in each group, which results in that the last signal in each group of 8 is not used. If the start address is M1 then address M8, M16, M24 and M32 are not used for the extra LEDs, they are yet reserved for the interface (IFC 64E) and should not be used for anything else. See also this following example.



Start I/O: M50  
Last occupied I/O: M81  
I/O for last LED: M80





## 21 Function keys

This section describes how to use the function keys. A function key is linked to a signal (e.g. M100) or a function by writing in its address after each key. The signal linked to a function key is activated according to the function linked to the key when it was defined.

E900 has 22 function keys (F1 to F22) built in.

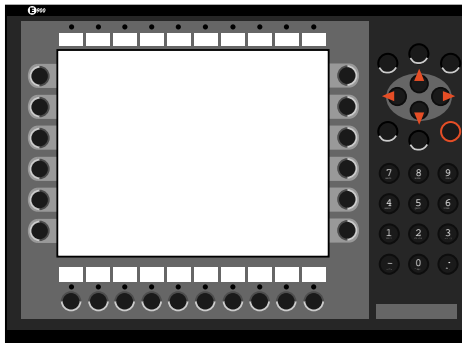
---

### Note!

It is only possible to affect two signals linked to function keys at the same time. If more than two function keys are pressed at the same time it is only the two pressed first that are affected.

---

### E900



### 21.1 Definitions

There are two ways of defining the function keys; *globally* or *locally*. Global function keys are defined and used in the whole application, that is they apply in all the blocks. Local function keys are defined and used in a single block. A global definition is always available in the run-time mode assuming that the block shown on the display does not have a local definition for the function key in question. Local definitions have a higher priority than global ones. Function keys are defined under **Function keys** in the **Functions** menu. Local function keys are defined in the block header for current block under F-keys.

**Function keys:Project1**

☒ I/O:  I/O:

Event:

☐ Set analog object to     
☐ Increment analog object with    or Set digital object  
☐ Decrement analog object with    or Reset digital object  
☐ Set digital object momentarily

☐ Jump to block:

☐ Other function:

Security Level:

Key	Function	Value	Name
F1			
F2			
F3			
F4			
F5			
F6			
F7			
F8			
F9			

Update

Clear

Text strip...

Exit

Function	Description
<b>IO</b>	The signal activated with the function key.
<b>Event</b>	<p>How the key affects the current digital signal.</p> <p><b>Momentary</b> means that the signal is set as long as the key is active.</p> <p><b>Toggle</b> means that the signal is set and reset respectively when the key is activated.</p> <p><b>Set</b> means that the signal is set when the key is activated and remains set.</p> <p><b>Reset</b> means that the signal is reset when the key is activated and remains reset.</p> <p><b>Grouped</b> means that all signals linked to a function key with current group number are reset. The group number is entered under Value. A group can contain max 8 function keys.</p> <p><b>Set Analog</b> means that the analog signal linked to the function key is given the value entered under Value.</p> <p><b>Inc Analog</b> means that the analog signal linked to the function key shall increase the value with the value entered under Value.</p> <p><b>Dec Analog</b> means that the analog signal linked to the function key shall decrease the value with the value entered under Value.</p>
<b>Set analog object to</b>	Sets selected analog object to entered value.
<b>Increment analog object with or set digital object</b>	Increases the value of an maneuverable analog object with entered value or sets a selected digital object.
<b>Decrement analog object with or reset digital object</b>	Decreases the value of an maneuverable analog object with entered value or sets a selected digital object.
<b>Set digital object momentarily</b>	Sets selected digital object as long as the key is pressed.
<b>Jump to block</b>	Jumps to the block with number nnn.
<b>Security level</b>	Function keys can be defined with a security level. The security level means that the operator must log in with a password for the same or a higher security level to be able to use the function key.
<b>Fxx</b>	The function key linked to one of the functions.

## Other functions for Function keys

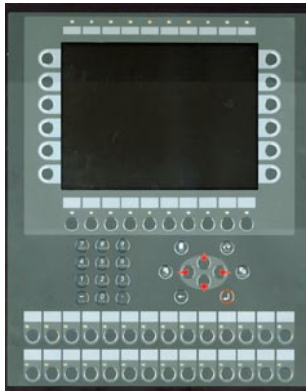
Function	Description
<b>Load recipe</b>	Load recipe from the memory of the terminal.
<b>Save recipe</b>	Save recipe into the memory of the terminal.
<b>Delete recipe</b>	Erase recipe from the memory of the terminal.
<b>Login to specified security level</b>	Logging in.
<b>Logout</b>	Logging out.
<b>Change login password</b>	Change password.
<b>Scroll one page up</b>	Scrolling the pages in text blocks and in the alarm list.
<b>Scroll one page down</b>	Scrolling the pages in text blocks and in the alarm list.
<b>Zoom up text size</b>	Increase the text size in the alarm list.
<b>Zoom down text size</b>	Reduce the text size in the alarm list.
<b>Save recipe on memory card</b>	Save recipe on the memory card defined as backup.
<b>Load recipe from memory card</b>	Load recipe from the memory card defined as backup.
<b>Erase recipe on memory card</b>	Erase recipe from the memory card defined as backup.
<b>Save project on memory card</b>	Save project on the memory card defined as backup.
<b>Load project from memory card</b>	Load project from the memory card defined as backup.
<b>Erase project on memory card</b>	Erase project from the memory card defined as backup.
<b>Acknowledge alarm</b>	Acknowledges alarms in the alarm list.
<b>Show alarm list</b>	Shows the alarm list (block 990).
<b>Jump to info block connected to the alarm</b>	Jump to a block connected to the alarm. See the chapter "Alarm handling".
<b>List alarm groups</b>	Select from which alarm group alarms will be shown in the alarm list.
<b>Return to previous block</b>	Show the previous block, works for nine levels. When returning to block 0 you cannot jump to the previous block using this function. If login in run-time is done on a higher security level than current level this function cannot be used for block jump.

Function	Description
<b>Jump to main block (block0)</b>	Show the start block, block number 0.
<b>Show object info</b>	Shows the min. and max. values for analog objects in the Textblock in the run-time mode.
<b>Enter</b>	Correspond to the Enter key.
<b>Show diagnostics page</b>	Shows the information page. See the chapter “Terminal functions”.
<b>TCPIP Connect</b>	Initiate contact at serial connection.
<b>TCPIP Disconnect</b>	Disconnect serial connection.

## 21.2 Jump to block with the function keys

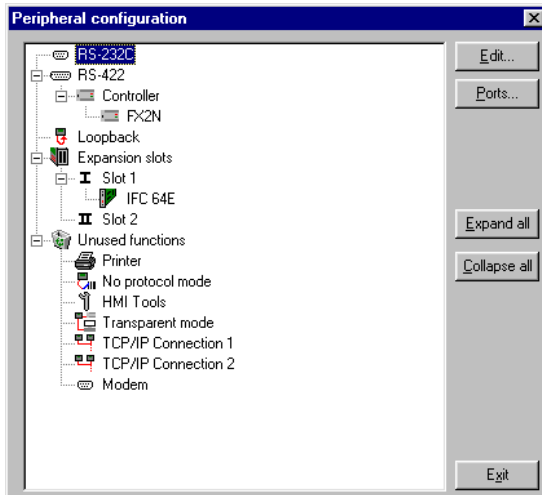
This function makes it possible to use the function keys for jumping to blocks without using the Display signal. When the key is defined, locally or globally, you select Jump to block in the list of options. Changing block via function keys is the easiest way to change block. This method does not use any digital signal in the controller system.

## 21.3 External function keys in E900 VT





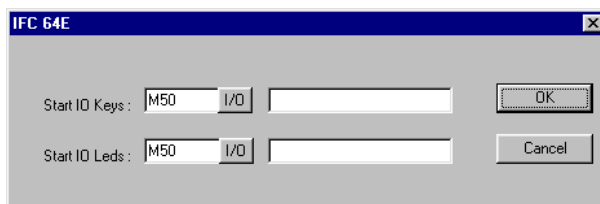
The terminal E900 VT has 28 external function keys, supported by the built-in expansion card IFC 64E. This expansion card is automatically selected at slot 1 when the terminal E900 VT is chosen in Project settings in the File menu.



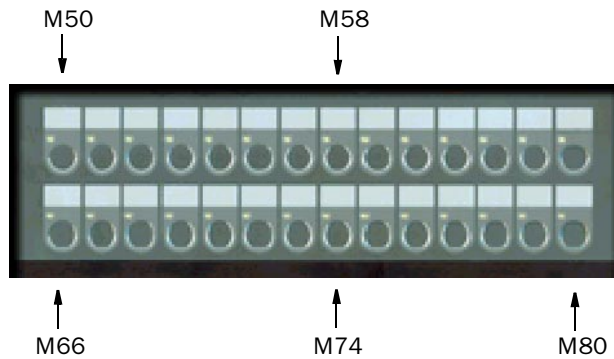
These extra function keys can only be connected to digital I/O signals, bit devices, and can not be used as the other, ordinary function keys.

To type in the desired start I/O for the function keys, right click on IFC 64E in Peripheral Configuration under Peripherals in the Setup menu.

The expansion card IFC 64E handles 32 function keys, the terminal E900 VT only use 28. There are 32 digital signals reserved consecutive from the start address. The function keys are grouped together in 4 groups with 7 keys in each group, which results in that the last signal in each group of 8 is not used. If the start address is M1 then address M8, M6, M24 and M32 are not used for the extra function keys, they are yet reserved for the interface (IFC 64E) and should not be used for anything else. See also this following example:



Start I/O: M50  
Last occupied I/O: M81  
I/O for last LED: M80





## **22 Expansion cards**

In the terminals there is a possibility to connect different expansion cards to increase the communication possibilities.

### **22.1 IFC PBDP**

With the IFC PBDP expansion card the terminal can be connected to and communicate as a slave with a Profibus network. For information about how to set up the card, see the manual for the product. For information about how to configure a Profibus network, see the manual for Profibus network.

### **22.2 IFC MC**

With the IFC MC expansion card you can connect a PCMCIA memory card to the terminal. You can select to use the memory card to expand the project memory, use it as back up or for moving files between terminals. For further information refer to the manual for IFC MC.

### **22.3 IFC 128E**

IFC 128E is an expansion card for connection of external function keys or switches. Up to 128 external function keys/switches can be used. For further information see the manual for the product.

### **22.4 IFC ETTP and IFC ETCX**

The IFC ETTP and IFC ETCX expansion cards are cards for network communication using TCP/IP via a twisted cable or coaxial cable. For further information, see the manual for IFC ETTP and IFC ETCX.

## **22.5 IFC PI**

IFC PI is an expansion card which makes it possible to connect a printer in parallel with the terminal. The RS-232 port on the expansion card cannot be used for external communication. For further information see the manual for IFC PI.

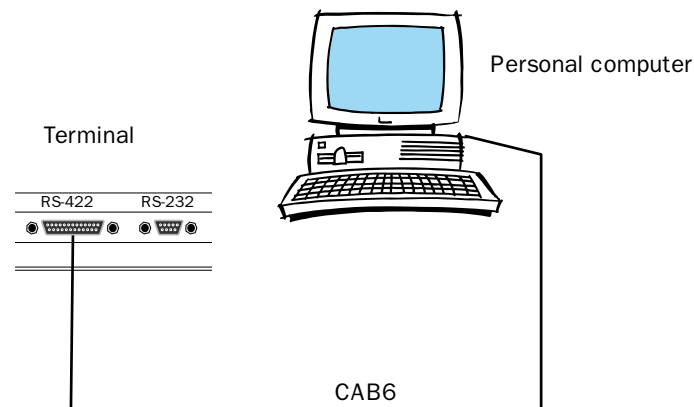
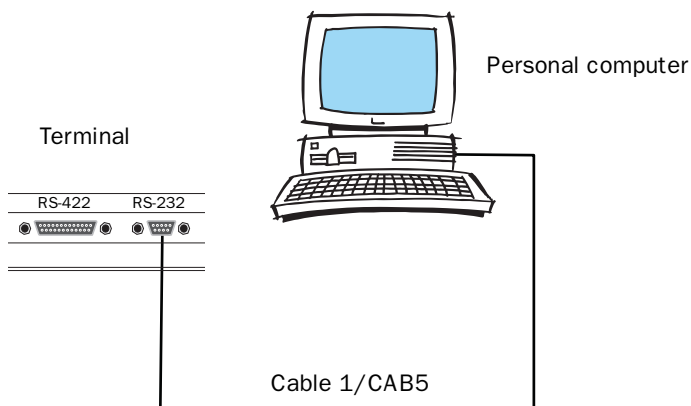
## **22.6 IFC GA**

IFC GA is an expansion card which makes it possible to connect a bus connection module, A7GT-BUS-EUN/ A7GT-BUS2-EUN, to the terminals E900 T, E900 VT and E910 T. IFC GA supports bus connection on PLC type MELSEC AnA, AnS, AnSH, AnU, QnA and QnAS. For further information see the manual for IFC GA.

## 23 Project transfer

The project must be transferred from the PC to the terminal in order to be used in the terminal.

Connect the PC with MAC Programmer+ to the terminal via the cable 1/CAB5 or CAB6 cable, see the table below.



Name	Continentaleurope	GB	Rest of the world
Cable 1	MAC-PC-CAB-R2	MAC-PROG/9-CAB	MAC-PROG/9-CAB

## 23.1 Terminal settings

The transfer parameters are set in the terminal in configuration mode under **Setup, Port Parameters, MP**.

### Note!

The communication settings must be the same in the PC software as in the terminal.

## 23.2 Transfer settings

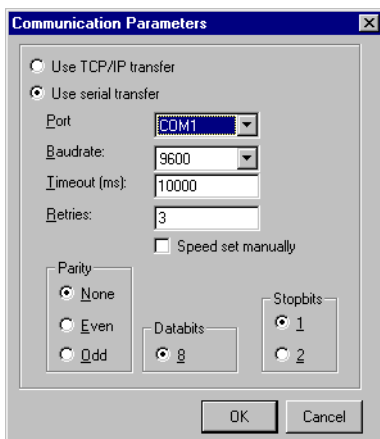
The transfer is controlled from the MAC Programmer+. In the menu Project Transfer you can select what is to be transferred.

Parameter	Description
<b>Percent complete</b>	Shows the status of the transfer during transfer.
<b>Byte count</b>	Shows the number of bytes transferred during transfer.
<b>Time elapsed</b>	Shows elapsed time since one of the function's sent, received or verified was activated.

Parameter	Description
<b>Status</b>	Shows which part of the project that is transferred. For example Setup, block, alarm groups, symbols and function keys.
<b>Info</b>	Display the defined driver that will be sent to the terminal.
<b>Retries</b>	When the transfer fails, the PC software makes a number of retries before the transfer is shut down.
<b>Terminal version</b>	When contact is established with the terminal, current terminal type and program version are displayed here.
<b>Test project on send</b>	Automatically tests the project during transfer.
<b>Automatic terminal RUN/Transfer switching</b>	The terminal is automatically put in transfer mode and will return to the previous mode after transfer.
<b>Check terminal version</b>	Checks the terminal version and gives a warning if the project and the terminal do not match.
<b>Send complete project</b>	Select if you want to transfer the whole project.
<b>Partial send options</b>	<div>None: No blocks are sent to the terminal.</div> <div>All: All blocks are sent to the terminal.</div> <div>From To: State an interval of blocks to be sent to the terminal.</div> <div>Alarms: Select if alarms shall be sent to the terminal.</div> <div>Symbols: Select if symbols shall be sent to the terminal.</div> <div>Time channels: Select if time channels shall be sent to the terminal.</div> <div>LED's: Select if LEDs shall be sent to the terminal.</div> <div>Message library: Select if message library shall be sent to the terminal.</div> <div>Setup: Select if the setup shall be sent to the terminal.</div> <div>Function keys: Select if function keys shall be sent to the terminal.</div> <div>Passwords: Select if passwords shall be sent to the terminal.</div>
<b>Delete</b>	<div>Trend data: Delete all stored trend data in the terminal.</div> <div>Recipe data: Delete all stored recipe data in the terminal.</div>
<b>Download driver</b>	<div>Never: Driver will never be downloaded.</div> <div>Always: Driver will always be downloaded.</div> <div>Automatic: Driver will be downloaded to the terminal when the driver in the terminal and the driver defined in current project are not the same.</div>
<b>Send current time</b>	The clock in the personal computer will be sent to the terminal.
<b>Send</b>	Transfer the project to the terminal according to current settings.



Parameter	Description
<b>Receive</b>	The PC software loads the project in the terminal. Current project in the PC software will be overwritten.
<b>Verify</b>	Compares the project in the PC software with the project in the terminal.
<b>Stop</b>	Interrupt the transfer.
<b>Settings</b>	Select if TCP/IP transfer or Serial transfer. If you select TCP/IP transfer no further settings are necessary. If you select Serial transfer the parameters described in the table below must be stated.



The communication parameters are set under **Comm Settings** in the **Transfer** menu or by choosing **Settings** in the **Project Transfer** dialog.

Parameter	Description
<b>Use TCP/IP transfer</b>	Select if the project shall be transferred via TCP/IP.
<b>Use serial transfer</b>	Select this if you want to use serial transfer of the project.
<b>Port</b>	Select communication port.
<b>Baudrate</b>	Baudrate (bits/second).
<b>Timeout</b>	Number of milliseconds between retries.
<b>Retries</b>	Number of retries before the transfer is shut down.
<b>Parity</b>	Select type of parity control.
<b>Databits</b>	Select number of data bits. Must be 8 bits.

Parameter	Description
<b>Stopbits</b>	Select number of stop bits.
<b>Speed set manually</b>	Used when communication is via a modem. You must set the baudrate in the terminal and PC software manually and they must be the same.

**Note!**

Communication can fail if other programs are running under Windows at the same time as the transfer takes place. Failure is avoided by shutting down the other programs. During block transfer, links to relevant symbols are included. If the symbols are not transferred, problems may occur.

## 23.3 TCP/IP transfer

When transferring via TCP/IP you need to select **Use TCP/IP transfer** in the **Comm Settings** dialog in the **Transfer** menu. When you press **Send** in the **Project transfer dialog** the following dialog is shown.

The screenshot shows a dialog box titled "Sending Project1". It has three input fields at the top: "Host address:" with the value "192.168.98.1", "Terminal control port:" with the value "6001", and "Transfer port:" with the value "6000". Below these is a "Login information" section containing "User ID:" and "Password:" fields, and a checkbox labeled "Save password in project" which is currently unchecked. At the bottom of the dialog are "OK" and "Cancel" buttons.

### Host address

State the IP address of the terminal you want to transfer to.

### Terminal Control port

State the TCP/IP-port number for RUN/TRANSFER switching. Need not normally be changed. Default is 6001.

### Transfer port

State the TCP/IP-port number for transferring. Need not normally be changed. Default is 6000.

**User ID**

State a user name used during RUN/Transfer switching authentication.  
Not needed if the terminal is already in transfer mode.

**Password**

State a password used during RUN/Transfer switching authentication.  
Not needed if the terminal is already in transfer mode.

**Save password in project**

Saves the password and user ID so that you do not need to enter it next time.

## **23.4 Serial transfer**

When serial transfer is used you need to select Use serial transfer in the Comm Settings dialog in the Transfer menu. When you press Send in the Project Transfer dialog the project will be sent to the terminal.

## 23.5 Transfer via modem

Settings when using modem communication between a PC and an operator terminal.

### Modem settings

Make the settings on the modem connected to the operator terminal as follows:

```
AT &F E0 Q1 &D0 &K0 &W
```

Make the settings on the modem connected to the PC as follows:

```
AT &F &D0 &K0 &Q0 &W
```

The table below describes the modem commands.

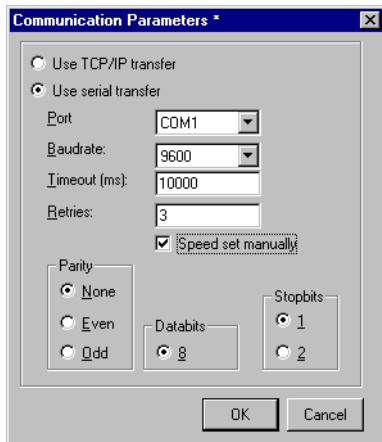
Command	Description
AT	Inform the modem that a command follows. Each command shall be proceeded by AT.
&F	Load the default settings
E0	Echo off
Q1	Command answer off
&D0	The modem ignores the DTR signal
&K0	No flow control
&Q0	Direct asynchronous mode
&W	Save settings

#### Note!

If other programs are running under Windows at the same time as project transfer it may occur communication error. Avoid the error by closing down other programs. Links to symbols in the blocks are transferred with the blocks.

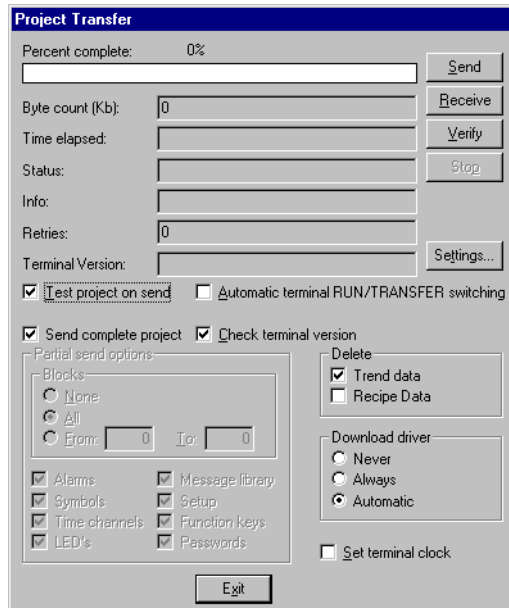
## Communication at selectable speed

1. Configure the modem.
2. Make the communication settings in the **Comm settings** dialog under **Transfer** in MAC Programmer+.



3. Set Baudrate, parity, data bits, stop bits and select Speed set manually.
4. Set the communication parameters in the terminal in the same way as in the MAC Programmer+ (Setup – Port parameters – MP)
5. Set the terminal in Transfer mode.
6. Use HMI Tools Modem program to make the call up.

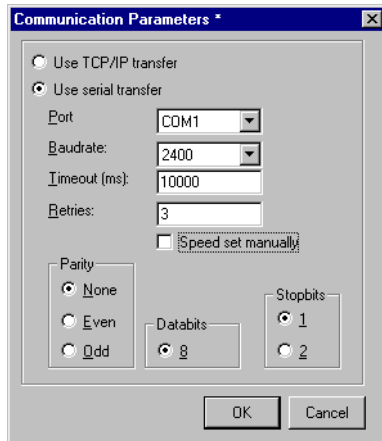
7. Then you select **Transfer** in MAC Programmer+.



**Do not** select Automatic terminal RUN/TRANSFER switching.

## Automatic transfer

1. Configure the modem.
2. Make the communication settings in the **Comm settings** dialog under **Transfer** in MAC Programmer+. The baudrate must be 2400 baud.



3. Use HMI Tools modem program to make the call up.

4. Then you select Transfer in MAC Programmer+.

The screenshot shows the 'Project Transfer' dialog box. It has a title bar 'Project Transfer' and a progress bar labeled 'Percent complete:' showing 0%. Below the progress bar are buttons for 'Send', 'Receive', 'Verify', 'Stop', and 'Settings...'. There are input fields for 'Byte count (Kb):' (0), 'Time elapsed:', 'Status:', 'Info:', 'Retries:' (0), and 'Terminal Version:'. Below these are checkboxes for 'Test project on send' (checked), 'Automatic terminal RUN/TRANSFER switching' (checked), 'Send complete project' (checked), and 'Check terminal version' (checked). There is a section for 'Partial send options' with a 'Blocks' sub-section containing radio buttons for 'None', 'All' (selected), and 'From:' (0) to 'To:' (0). Below this are checkboxes for 'Alarms', 'Symbols', 'Time channels', 'LED's', 'Message library', 'Setup', 'Function keys', and 'Passwords'. To the right of the 'Partial send options' section are checkboxes for 'Delete' (checked), 'Trend data', 'Recipe Data', 'Download driver' (radio buttons for 'Never', 'Always', 'Automatic' with 'Automatic' selected), and 'Set terminal clock' (unchecked). An 'Exit' button is at the bottom center.

---

**Note!**

If Auto terminal RUN/TRANSFER switching is used the baudrate must be set to 2400 baud.

---





## 24 Technical data

### E900 T

Parameter	Data
Front panel	Width 367 x Height 274 x Depth 6.0 mm
Mounting depth	Excl. D-sub: 90 mm Incl. D-sub: 110 mm.
Front panel seal	IP 65
Rear panel seal	IP 20
Keyboard material	Membrane keyboard with polyester snap discs. Overlay film of Autotex 2 F200 with print on reverse side. 1 million operations.
Reverse side material	Yellow-chrome-plated steel plate.
Weight	Excl. D-sub: 3.5 kg
Serial port RS-422	25-pin D-sub contact, chassis mounted female with standard locking screws 4-40 UNC.
Serial port RS-232	9-pin D-sub contact, chassis mounted male with standard locking screws 4-40 UNC.
Expansion slots	2 built-in expansion slots.
Display	TFT display, 640x480 pixel, 256 colors, graphic and text. Back light: lifetime 25000 h at an ambient temperature of +25°C.
Active area of display	B x H, 211.2 x 158.4 mm
Memory	Flash memory, 2 Mb.
Real-time clock	±10 PPM + error because of ambient temperature and supply voltage. Total max error 1 min/month= 12min/year. Minimum life: 10 years of the battery for the real-time clock.
Supply voltage	100-240 VAC
Current consumption at rated voltage	Max: 0.35 A.
Ambient temperature	+0° to +50°C.
Storage temperature	-20° to +60°C.
Relative humidity	Max 85% non-condensing.
EMC tests on the terminal	The terminal conforms with the essential protection requirements in article four of the 89/336/EEC directive. Noise tested according to: EN50081-2 emission and EN50082-2 immunity.

## E900 VT

Parameter	Data
Front panel	Width 335 x Height 430 x Depth 6.0 mm
Mounting depth	Excl. D-sub: 90 mm Incl. D-sub: 110 mm.
Front panel seal	IP 65
Rear panel seal	IP 20
Keyboard material	Membrane keyboard with polyester snap discs. Overlay film of Autotex 2 F200 with print on reverse side. 1 million operations.
Reverse side material	Yellow-chrome-plated steel plate.
Weight	Excl. D-sub: 4.0 kg
Serial port RS-422	25-pin D-sub contact, chassis mounted female with standard locking screws 4-40 UNC.
Serial port RS-232	9-pin D-sub contact, chassis mounted male with standard locking screws 4-40 UNC.
Expansion slots	1 built-in expansion slot.
Display	TFT display, 640x480 pixel, 256 colors, graphic and text. Back light: lifetime 25000 h at an ambient temperature of +25°C.
Active area of display	B x H, 211.2 x 158.4 mm
Memory	Flash memory, 2 Mb.
Real-time clock	±10 PPM + error because of ambient temperature and supply voltage. Total max error 1 min/month= 12min/year. Minimum life: 10 years of the battery for the real-time clock.
Supply voltage	100-240 VAC
Current consumption at rated voltage	Max: 0.35 A.
Ambient temperature	+0° to +50°C.
Storage temperature	-20° to +60°C.
Relative humidity	Max 85% non-condensing.
EMC tests on the terminal	The terminal conforms with the essential protection requirements in article four of the 89/336/EEC directive. Noise tested according to: EN50081-2 emission and EN50082-2 immunity.

**E910 T**

Parameter	Data
Front panel	Width 290 x Height 247x Depth 5.0 mm
Mounting depth	Excl. D-sub: 109 mm Incl. D-sub: 130 mm.
Front panel seal	IP 65
Rear panel seal	IP 20
Keyboard material	Membrane keyboard with polyester snap discs. Overlay film of Autotex 2 F200 with print on reverse side. 1 million operations.
Reverse side material	Yellow-chrome plated steel plate.
Weight	Excl. D-sub: 3.3 kg
Serial port RS-422	25-pin D-sub contact, chassis mounted female with standard locking screws 4-40 UNC.
Serial port RS-232	9-pin D-sub contact, chassis mounted male with standard locking screws 4-40 UNC.
Expansion slots	2 built-in expansion slots.
Display	TFT display, 640x480 pixel, 256 colors, graphic and text. Back light: lifetime 25000 h at an ambient temperature of +25°C. Touch screen
Active area of display	B x H, 211.2 x 158.4 mm
Memory	Flash memory, 2 Mb.
Real-time clock	±10 PPM + error because of ambient temperature and supply voltage. Total max error 1 min/month= 12min/year. Minimum life: 10 years of the battery for the real-time clock.
Supply voltage	100-240 VAC
Current consumption at rated voltage	Max: 0.35 A.
Ambient temperature	+0° to +50°C.
Storage temperature	-20° to +60°C.
Relative humidity	Max 85% non-condensing.
EMC tests on the terminal	The terminal conforms with the essential protection requirements in article four of the 89/336/EEC directive. Noise tested according to: EN50081-2 emission and EN50082-2 immunity.

## 24.1 Membrane keyboard

### Solvent resistance and enviromental data for Autotex 2

Autotex 2 withstands exposure of more than 24 hours duration under DIN 42 115 Part 2 to the following chemicals without visible change:

Ethanol	Formaldehyde 37%-42%	1.1.1. Trichloroethane (Genklene)
Cyclohexanol	Acetaldehyde	Ethylacetate
Diacetone alcohol	Aliphatic hydrocarbons	Diethyl ether
Glycol	Toluene	N-Butyl acetate
Isopropanol	Xylene	Amylacetate
Glycerine	White spirit	Butycellosolve
Methanol	Formic acid <50%	Ether
Triacetin	Acetic acid <50%	Sodium hypchlorite <20% (Bleach)
Dowanol DRM/PM	Phosphoric acid <30%	Hydrogen peroxide <25%
Acetone	Hydrochloric acid <36%	Potassium carbonate
Metyl ethyl ketone	Nitric acid <10%	Washing powders
Dioxan	Trichloroacetic acid <50%	Fabric conditioner
Cyclohexanone	Sulphuric acid <10%	Ferric chloride
MIBK	Cutting oil	Ferrous chloride
Isophorone	Diesel oil	Dibutyl Phthalate
Ammonia <40%	Linseed oil	Diocetyl Phthalate
Caustic soda <40%	Paraffin oil	Sodium carbonate
Potassium hydroxide <30%	Blown castor oil	
Alkalicarbonate	Silicone oil	
Bichromate	Turpentine substitute	

Potassium ferrocyanide/ferricyanide	Universal brake fluid	
Acetonitrile	Decon	
Sodium bisulphate	Aviation fuel	
	Petrol	
	Teepol	
	Water	
	Sea water	

Autotex withstands DIN 42 115 Part 2 exposure of <1 hour duration to glacial acetic acid without visible change.

Autotex is **not** resistant to the following chemicals:

Concentrated mineral acids
Concentrated caustic solution
High pressure steam at over 100°C
Benzyl alcohol
Methylene chloride

Autotex withstands 24 hours exposure to the following reagents at 50°C without visible staining:

Top Job	Grape Juice	Ariel	Ajax
Jet Dry	Milk	Persil	Vim
Gumption	Coffee	Wisk	Domestos
Fantastic		Lenor	Vortex
Formula 409		Downey	Windex

Very slight discoloration was noted under critical viewing conditions with the following materials:

Tomato juice
Tomato ketchup
Lemon juice
Mustard

## Enviromental data

### Minimum use temperature

Tactile embossed Autotex has been tested to 0.5 million switch operations at -40°C (-40°F) without loss of function.

### Maximum use temperature

Low moderate humidity: 85°C

High humidity (>90% RH): 40°C

### Outdoor use

In common with all polyester based films, Autotex is not suitable for use in conditions of long-term exposure to direct sunlight.

# 24.2 Drawings

## RS-232/RS422

RS-232  
E900/910

Pin no	Name	Signal direction MAC → ← XXX
2	TxD	→
3	RxD	←
5	0V	
7	CTS	←
8	RTS	→
9		

D-sub  
9-pin Male

RS-422  
E900/910

Pin no	Name	Signal direction MAC → ← XXX
2	+TxD	→
15	-TxD	
3	+RxD	←
16	-RxD	
4	+RTS	→
17	-RTS	
5	+CTS	←
18	-CTS	
20	**	
21	**	
7,8	0V	
12,13		
24,25		

D-sub  
25-pin Female

\*\*

Pin no 20 connected to pin no 21 internal in E900

**MITSUBISHI ELECTRIC**  
Hiroshima

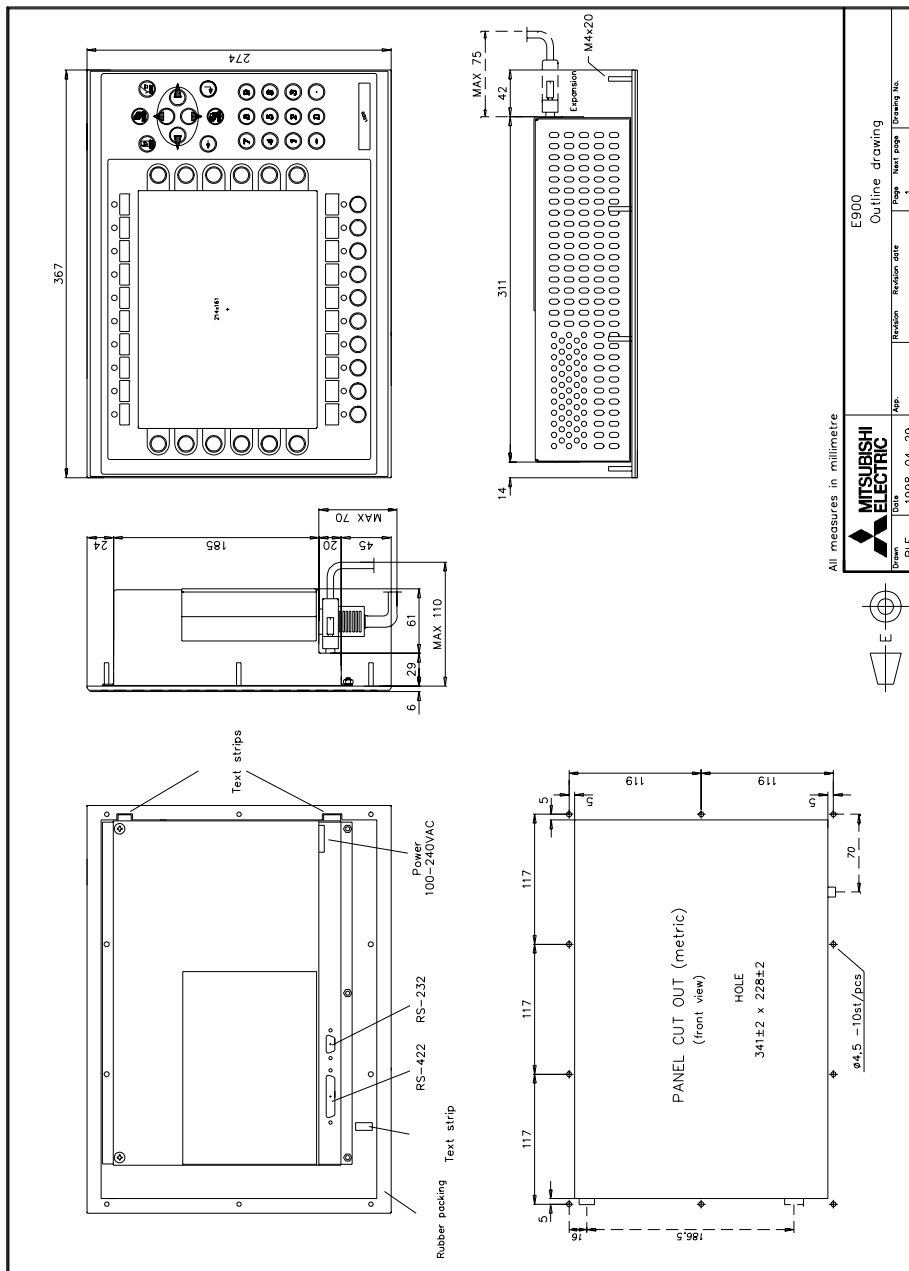
Docum	930907	Rev	980109	Material		Size	1	Form	—	Revision	
										Comment	E900 RS-232/RS-422

A3

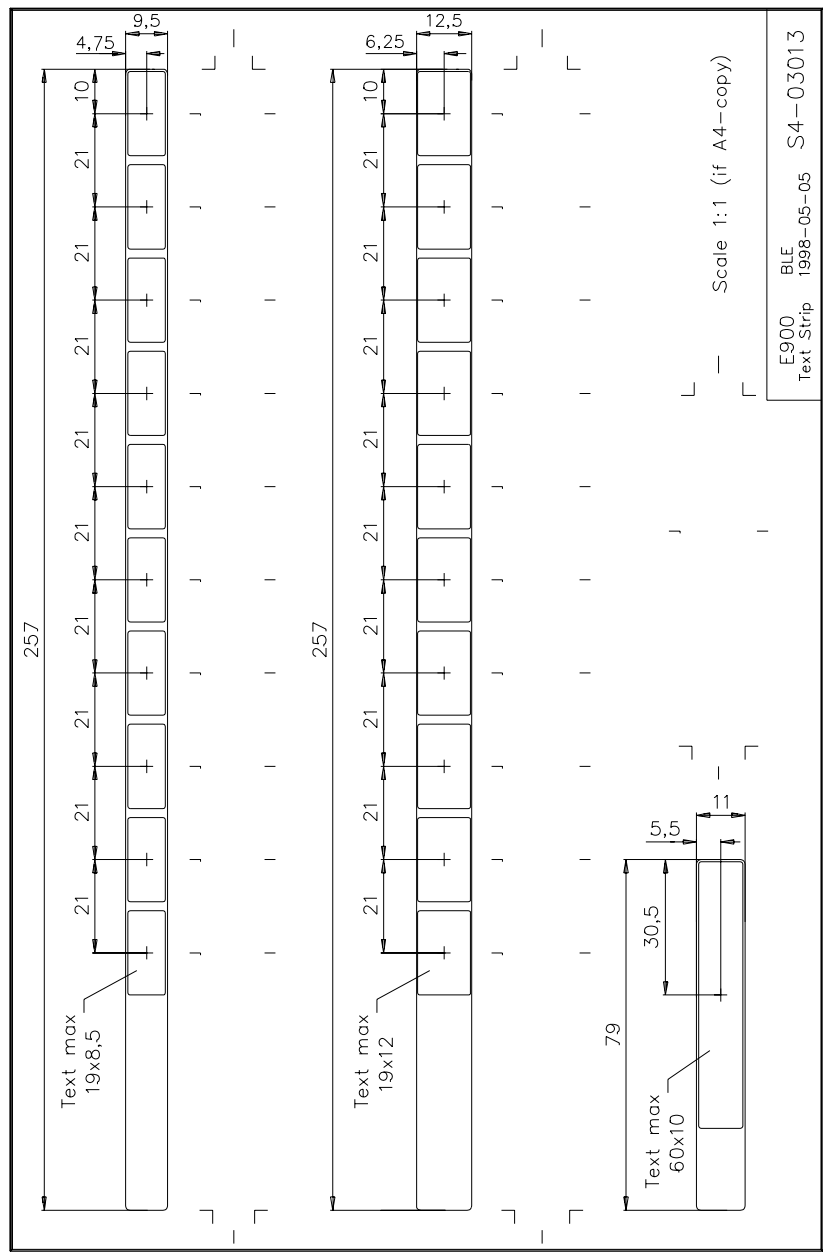
24-7



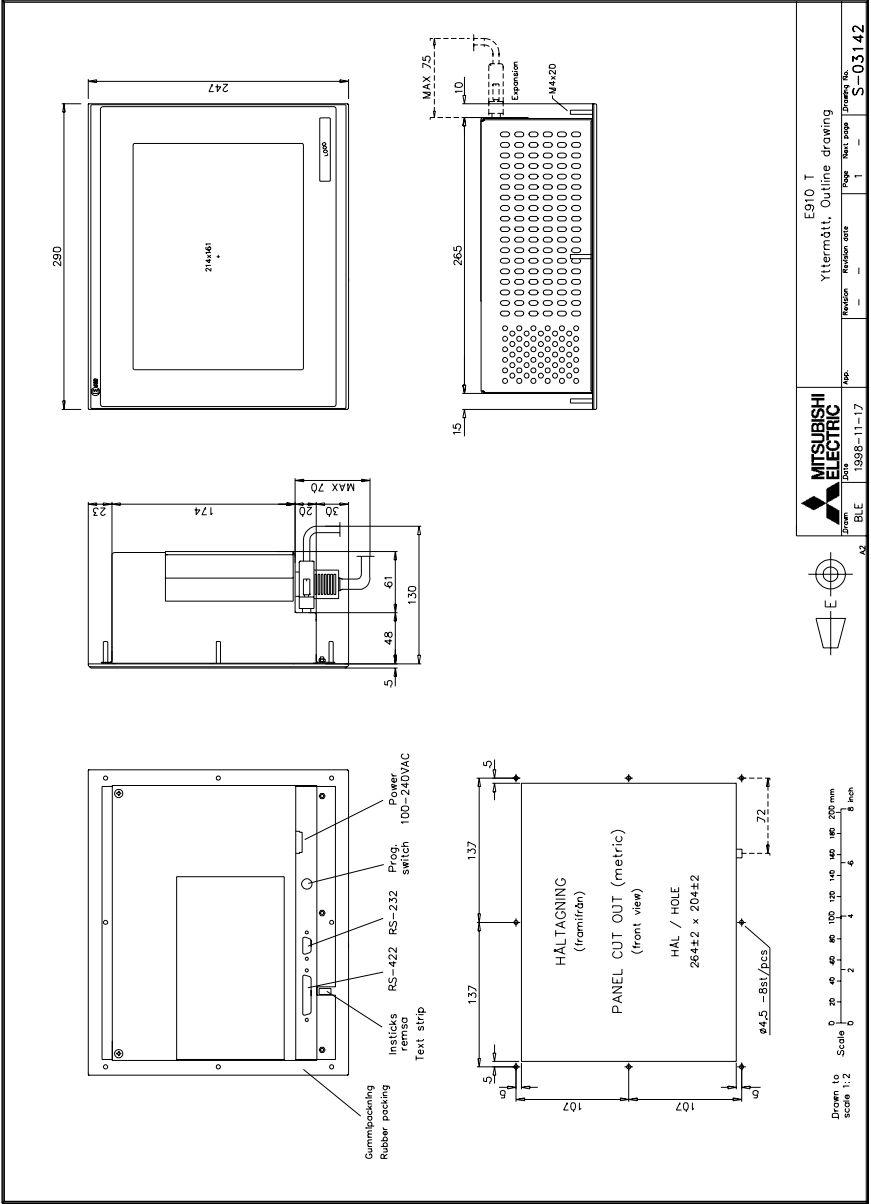
## E900 Outline



E900 Text strip











E910 Outline



## 24.3 Character tables

### Code page 437

The first characters, 0-127, are the same as in code page 850.

Ç 128 Coedil 00C7	É 144 Eacute 00C9	á 160 aacute 00E1	 176 blk14 2591	┌ 192 2514	≡ 208 2568	α 224 alpha 03B1	≡ 240 equiv 2261
ü 129 uuml 00FC	æ 145 aelig 00E6	í 161 iacute 00ED	 177 blk12 2592	┐ 193 2534	⌋ 209 2564	ß 225 szlig 00DF	± 241 plusmn 00B1
é 130 eacute 00E9	Æ 146 AElig 00C6	ó 162 oacute 00F3	 178 blk34 2593	└ 194 252C	⌌ 210 2565	Γ 226 Gamma 0393	≥ 242 ge 2265
â 131 acirc 00E2	ô 147 ocirc 00F4	ú 163 uacute 00FA	179 2502	┐ 195 251C	ℓ 211 2559	π 227 pi 03C0	≤ 243 le 2264
ã 132 auml 00E4	ö 148 ouml 00F6	ñ 164 ntilde 00F1	└ 180 2524	┐ 196 2500	ℓ 212 2558	Σ 228 Sigma 03A3	┌ 244 2320
à 133 agrave 00E0	ò 149 ograve 00F2	Ñ 165 Ntilde 00D1	≡ 181 2561	┐ 197 253C	ℓ 213 2552	σ 229 sigma 03C3	J 245 2321
å 134 aring 00E5	û 150 ucirc 00FB	ª 166 ordf 00AA	≡ 182 2562	ℓ 198 255E	ℓ 214 2553	μ 230 micro 00B5	÷ 246 divide 00F7
ç 135 ccedil 00E7	ù 151 ugrave 00F9	º 167 ordm 00BA	≡ 183 2556	ℓ 199 255F	≡ 215 2568	τ 231 tau 03C4	≈ 247 ap 2248
ê 136 eirc 00EA	ÿ 152 yuml 00FF	¿ 168 iquet 00BF	≡ 184 2555	ℓ 200 255A	≡ 216 256A	Φ 232 PHgr 03A6	° 248 deg 00B0
ë 137 euml 00EB	Ö 153 Ouml 00D6	┐ 169 2310	≡ 185 2563	ℓ 201 2554	┐ 217 2518	Θ 233 Theta 0398	• 249 2219
è 138 egrave 00E8	Ü 154 Uuml 00DC	┐ 170 not 00AC	186 2551	≡ 202 2569	┐ 218 250C	Ω 234 Omega 03A9	· 250 middot 00B7
ï 139 iuml 00EF	ø 155 oslash 00F8	½ 171 frac12 00BD	≡ 187 2557	≡ 203 2566	 219 block 2588	δ 235 delta 03B4	√ 251 radic 221A
î 140 icirc 00EE	£ 156 pound 00A3	¼ 172 frac14 00BC	≡ 188 255D	ℓ 204 2560	 220 lhb1k 2584	∞ 236 infin 221E	ⁿ 252 207F
ì 141 igrave 00EC	¥ 157 yen 00A5	¡ 173 iexcl 00A1	≡ 189 255C	≡ 205 2550	 221 258C	φ 237 phis 03C6	² 253 sup2 00B2
Ä 142 Auml 00C4	Þ 158 20A7	« 174 laquo 00AB	≡ 190 255B	≡ 206 256C	 222 2590	€ 238 epsi 03B5	■ 254 squf 25A0
Å 143 Aring 00C5	ƒ 159 fnof 0192	» 175 raquo 00BB	┐ 191 2510	≡ 207 2567	 223 2580	∩ 239 cap 2229	nbsp 00A0

## Code page 850

Characters 0-127. Note that the characters 0-30 does not work.

	000	► 016		032	0 048	@ 064	P 080	` 096	p 112
		rttrif 25B6	sp 0020		0030	commat 0040		grave 0060	
☺	001	◀ 017	! 033	1 049	A 065	Q 081	a 097	q 113	
	263A	ltrif 25C0	excl 0021		0031		0051	0061	0071
☹	002	↓ 018	" 034	2 050	B 066	R 082	b 098	r 114	
	263B	varr 2195	quot 0022		0032		0052	0062	0072
♥	003	!! 019	# 035	3 051	C 067	S 083	c 099	s 115	
hearts	2665		num 0023		0033		0053	0063	0073
♦	004	¶ 020	\$ 036	4 052	D 068	T 084	d 100	t 116	
diams	2666	para 00B6	dollar 0024		0034		0054	0064	0074
♣	005	§ 021	% 037	5 053	E 069	U 085	e 101	u 117	
clubs	2663	sect 00A7	percent 0025		0035		0055	0065	0075
♠	006	■ 022	& 038	6 054	F 070	V 086	f 102	v 118	
spades	2660		amp 0026		0036		0056	0066	0076
•	007	‡ 023	' 039	7 055	G 071	W 087	g 103	w 119	
bull	2022		apos 0027		0037		0057	0067	0077
■	008	↑ 024	( 040	8 056	H 072	X 088	h 104	x 120	
	2508	uarr 2191	lpar 0028		0038		0058	0068	0078
○	009	↓ 025	) 041	9 057	I 073	Y 089	i 105	y 121	
	25CB	darr 2193	rpar 0029		0039		0059	0069	0079
☉	010	→ 026	* 042	:	058	J 074	Z 090	j 106	z 122
	25D9	rarr 2192	ast 002A	colon 003A			004A	005A	006A
☽	011	← 027	+ 043	;	059	K 075	I 091	k 107	{ 123
male	2642	larr 2190	plus 002B	semi 003B		004B	lsqb 005B	006B	loub 007B
♀	012	└ 028	,	< 060	L 076	\ 092	l 108	124	
female	2640		comma 002C	lt 003C		004C	bsol 005C	006C	verbar 007C
	013		- 045	= 061	M 077	] 093	m 109	}	125
sung	266A	harr 2194	hyphen 002D	equals 003D		004D	rsqb 005D	006D	roub 007D
♪	014	▲ 030	.	> 062	N 078	^ 094	n 110	~ 126	
	266B	utrif 25B2	period 002E	gt 003E		004E	circ 005E	006E	tilde 007E
☼	015	▼ 031	/ 047	? 063	O 079	_ 095	o 111	◻ 127	
	263C	dtrif 25BC	sol 002F	quest 003F		004F	lowbar 005F	006F	2302

# Code page 850

## Characters 128-255

Ç 128 Ccedil 00C7	É 144 Eacute 00C9	á 160 aacute 00E1	█ 176 blk14 2591	Ł 192 2514	ð 208 eth 00F0	Ó 224 Oacute 00D3	- 240 shy 00AD
ü 129 uuml 00FC	æ 145 aelig 00E6	í 161 iacute 00ED	█ 177 blk12 2592	┘ 193 2534	Ð 209 ETH 00D0	ß 225 szlig 00DF	± 241 plusmn 00B1
é 130 eacute 00E9	Æ 146 AElig 00C6	ó 162 oacute 00F3	█ 178 blk34 2593	┘ 194 252C	Ê 210 Ecirc 00CA	Ô 226 Ocirc 00D4	= 242 2017
â 131 acirc 00E2	ô 147 ocirc 00F4	ú 163 uacute 00FA	179 2502	┘ 195 251C	Ë 211 Euml 00CB	Ò 227 Ograve 00D2	¾ 243 frac34 00BE
ã 132 auml 00E4	õ 148 ouml 00F6	ñ 164 ntilde 00F1	┘ 180 2524	— 196 2500	È 212 Egrave 00C8	Ö 228 otilde 00F5	¶ 244 para 00B6
à 133 agrave 00E0	ò 149 ograve 00F2	Ñ 165 Ntilde 00D1	À 181 Aacute 00C1	÷ 197 253C	İ 213 inodot 0131	Û 229 Otilde 00D5	§ 245 sect 00A7
å 134 aring 00E5	û 150 ucirc 00FB	ª 166 ordf 00AA	Â 182 Acirc 00C2	ä 198 atilde 00E3	Í 214 iacute 00CD	µ 230 micro 00B5	÷ 246 divide 00F7
ç 135 ccedil 00E7	ù 151 ugrave 00F9	º 167 ordm 00BA	Á 183 Agrave 00C0	Ä 199 Atilde 00C3	Î 215 icirc 00CE	þ 231 thorn 00FE	¸ 247 cedil 00B8
ê 136 ecirc 00EA	ÿ 152 yuml 00FF	¿ 168 iquest 00BF	© 184 copy 00A9	ℒ 200 255A	Ĭ 216 luml 00CF	Þ 232 THORN 00DE	° 248 deg 00B0
ë 137 euml 00EB	Ï 153 ouml 00D6	® 169 reg 00AE	≡ 185 2563	℔ 201 2554	┘ 217 2518	Ú 233 Uacute 00DA	“ 249 uml 00A8
è 138 egrave 00E8	Û 154 Uuml 00DC	¬ 170 not 00AC	186 2551	≡ 202 2569	┘ 218 250C	Û 234 Ucirc 00DB	· 250 middot 00B7
ï 139 iuml 00EF	ø 155 oslash 00F8	½ 171 frac12 00BD	┘ 187 2557	≡ 203 2566	■ 219 block 2588	Ù 235 Ugrave 00D9	¹ 251 sup1 00B9
î 140 icirc 00EE	£ 156 pound 00A3	¼ 172 frac14 00BC	≡ 188 255D	≡ 204 2560	■ 220 lbrk 2584	Ý 236 yacute 00FD	³ 252 sup3 00B3
ì 141 igrave 00EC	Ø 157 oslash 00D8	¡ 173 iexcl 00A1	¢ 189 cent 00A2	= 205 2550	ı 221 brvbar 00A6	Ý 237 Yacute 00DD	² 253 sup2 00B2
Ā 142 Auml 00C4	× 158 times 00D7	« 174 laquo 00AB	¥ 190 yen 00A5	≡ 206 256C	ì 222 igrave 00CC	˘ 238 macr 00AF	■ 254 sqf 25A0
Å 143 Aring 00C5	ƒ 159 fnof 0192	» 175 raquo 00BB	┘ 191 2510	⌘ 207 curren 00A4	■ 223 uhblk 2580	˘ 239 acute 00B4	255 nbsp 00A0

# Code page 866

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
Ё	ё	Є	є	Ӏ	ӱ	Ӳ	ӳ	•	•	•	√	№	¤	■	nbsp
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

## 25 Appendix

### Error codes in PLC drivers

#### FX, ACPU and C24

Error code	Description
Comm Error	Communication error.
File Reg Err xxxx	An object tries to read a file register not configured in PLC parameters.
Illegal dev [x]	Trying to read an invalid device type (D, X, Y...). Only C24.

#### QCPU driver

Error code	Description
Comm Error xxxx	Communication error, see the table below.
SYSTEM ERROR Comm Error	Communication error during start-up.

Item	Content	Reply status
Common Errors	Sum check error (periphery only)	4000H
	Cannot handle RT	4001H
	Cannot handle SRT	4002H
	Impossible global request command	4003H
	Command not usable during system protect	4004H
	Data range larger than buffer size	4005H
	No password release	4006H
	CID differing from CPU data	4007H
Device setting	Cannot handle device name	1030H
	Device range over	4031H
	Cannot handle device modification	4032H
	No random write possible on the designated Y because it is used by the system	4033H
	Improper device	4034H



Item	Content	Reply status
Protect	Hardware protection active	4050H
	No execution possible because memory protect switch is ON	4051H
Others	Data error	4080H

# Index

## A

Acknowledge notify, 10-3  
 Acknowledge required, 10-4  
 Alarm banner object, 6-5  
 Alarm editor, 5-31  
 Alarm group, 10-2  
 Alarm history, 10-4  
 Alarm info. block, 10-4, 10-8  
 Alarm message, 10-3  
 Alarm remote acknowledge, 10-3  
 Alarm, 10-1  
     Alarm list, 5-46  
     Define alarm group, 10-2  
     Define alarm list, 10-4  
     Define alarm message, 10-3  
     Graphic alarm page, 10-8  
     Maneuvering, 10-6  
 Align, 5-18  
 Alpha/numerical keyboard, 3-6  
 Analog clock object, 6-6  
 Analog fill object, 6-7  
 Analog objects, 6-3  
 Analog signals, 4-8  
 Append, 5-31  
 Arrow keys, 3-7  
 ASCII object, 6-8

## B

Background block, 5-51  
 Backlight, 5-38  
 Bar graph object, 6-9, 7-4  
 Block  
     Create, 5-5  
 Block header, 5-12  
 Block Manager, 5-5  
     Settings, 5-11

Block, 4-6  
     Edit, 5-33  
     Programming, 5-12  
 Built-in function keys, 3-7

## C

Clock  
     Digital, 6-12  
 Clock,  
     Analog, 6-6  
 Color scheme,  
     Change/create, 5-3  
 Colors, 6-1  
 Command line, 5-41  
 Communication error, 3-5  
 Communication, 3-5  
     Transparent mode, 17-1  
 Connect to a printer, 13-2  
 Contrast settings, 3-2  
 Coordinates, 2-2  
 Copy graphics, 5-22  
 Create insertable strip, 5-12  
 Create object, 5-17  
 Create series, 5-21  
 Cross reference, 5-30  
     Presentation, 5-30

## D

Date format, 5-43  
 Date object, 7-5  
 Define block header, 5-7  
 Diagram object, 6-10  
 Digital clock object, 6-12  
 Digital objects, 6-3  
 Digital signals, 4-7  
 Digital symbol object, 6-15  
 Digital text object, 6-16, 7-6

**E**

Editing graphic objects, 3-3  
 Erase memory, 3-2  
 Error message, 3-5  
 Expansion card, 22-1  
 External keyboard, 3-8

**F**

Filled object  
     Digital, 6-13  
 Function keys, 5-31, 21-1  
     Define, 21-1  
     Global, 21-1  
     Jump to block, 21-5  
     Local, 21-1  
 Functions menu, 5-36

**G**

Gain, 6-1, 7-1  
 Graphic alarm page, 10-8  
 Graphic block printout, 13-3  
 Graphic blocks, 4-5  
 Graphic objects,  
     Alarm banner, 6-5  
     Analog clock, 6-6  
     Analog fill, 6-7  
     Analog, 6-3  
     ASCII object, 6-8  
     Bar graph, 6-9  
     Diagram, 6-10  
     Digital clock, 6-12  
     Digital symbol, 6-15  
     Digital text, 6-16  
     Digital, 6-3  
     Filled object, 6-13  
     Jump, 6-17, 7-7  
     Maneuvering, 6-32  
     Message, 6-18  
     Multiple choice, 6-19

Multisymbol, 6-20  
 Numeric table, 6-22  
 Numeric, 6-21  
 Slide, 6-24  
 Speedometer, 6-25  
 Static objects, 6-2  
 TCP/IP command object, 6-26  
 Touch key, 6-28  
 Trend, 6-29  
 VU meter, 6-31

Graphicblock editor, 5-13  
     3D-Controls, 5-17  
     Create object, 5-17  
     Create series, 5-21  
     Cursor, 5-14, 5-15  
     Draft Move & Size, 5-17  
     Group object, 5-20  
     Keys, 5-14  
     Mouse, 5-14  
     Open, 5-13  
     Place object, 5-18  
     Recommendations, 5-17  
     Static graphics, 5-17  
     Toolbox, 5-15  
     Ungroup object, 5-20  
 Grid, 5-16  
 Group object, 5-20

**H**

Help line, 3-4  
 Historical trends,  
     Run-time mode, 8-2

**I**

I/O change, 5-28  
 I/O Cross reference, 5-30  
 IFC MC, 22-1  
 IFC PBDP, 22-1  
 Indexaddressing, 16-1  
 Information page, 3-10  
 Insert, 5-31  
 Insertable strip, 5-12  
 Install driver, 5-2, 5-3  
 Internet, 5-2

**J**

Joystick function 3-11  
 Jump object, 6-17, 7-7  
 Jump to block, 21-5

**K**

Key delay time, 5-45  
 Key repeat, 5-45  
 Key sequences, 3-8  
 Keyboard  
     Alpha/numerical, 3-6

**L**

LEDs, 5-31, 20-1  
     Define, 20-1  
 Levels 4-6  
 Library index register, 5-40

**M**

MAC Programmer+, 2-1, 5-1  
     Edit menu, 5-33  
     File menu, 5-32  
     Functions menu, 5-36  
     Install, 2-1  
     Setup menu, 5-37  
     System requirements, 2-1

Trasfer menu, 5-50  
 View menu, 5-33  
 Window menu, 5-51  
 Maneuver objects,  
     Analog objects, 6-33, 7-11  
     Digital objects, 6-32, 7-11  
 Maneuverable objects, 6-32  
 Maneuvering graphic blocks, 6-32  
 Maneuvering textblocks, 7-11  
 Menu bar, 2-2  
 Menu structure, 4-5  
 Message library, 9-1  
 Message object, 6-18, 7-8  
 Method for programming, 4-1  
 Modem connection, 17-13  
 Modem transfer, 23-7  
 Multiple choice object, 6-19, 7-9  
 Multisymbol object, 6-20

**N**

Name list, 5-34  
 National characters, 3-6  
 Network communication via IFC  
 ET, 18-11  
 Network communication, 18-1  
 No protocol mode, 17-5  
 Numeric objects, 6-21  
 Numeric table object, 6-22  
 Numeric values, 3-7

**O**

Offset, 6-1, 7-1  
 OGC, 6-2, 7-1  
 Operation mode, 3-1  
     PROG, 3-3  
     RUN, 3-5  
     SETUP, 3-2  
     Switch, 3-1  
 Operator's pictures, 4-5

**P**

- Passwords, 5-31, 12-1
  - Change, 12-3
  - Define, 12-1
  - Logging in, 12-3
  - Security levels, 12-1
- PLC system, 5-2
- Popup keyboard, 6-35
  - Control block, 5-39
- PREV, 3-7
- Print alarms, 10-4
- Printer
  - Connection, 13-2
- Printer settings, 5-48, 13-4
- Profibus-DP, 22-1
- Programming mode, 3-3
- Project,
  - Close, 5-32
  - Create, 5-1, 5-32
  - Open, 5-32
  - Programming, 4-1, 5-1
  - Save, 5-32
  - Transfer, 5-50, 23-1

**R**

- Real-time clock,
  - Set, 3-9
- Recipe, 11-1
  - Create, 11-2
  - Erase, 11-4
  - Files, 11-1
  - HMI Tools, 11-4
  - Transfer, 11-2
  - Use recipe in a PC, 11-4
  - Via the PLC-program, 11-5
- Report print-outs, 13-1
  - Define printout, 13-1
  - Define, 13-4
  - Port parameters, 13-6

- Printer settings, 13-4
- Reserved characters, 3-7, 5-24
- RS-232, 5-47
- RS-422, 5-47
- Run-time mode, 3-5

**S**

- Scaling, 6-1, 7-1
- Security level, 3-7, 12-1
  - Define, 12-1
  - Password, 12-1
- Select maneuverable objects, 6-32
- Serial network communication, 18-16
- Serial transfer, 23-6
- Setup mode, 3-2
- Show terminal, 5-11
- Signal format, 4-7, 4-9
- Slide object, 6-24
- Speedometer object, 6-25
- Static graphics, 5-17
- Static objects, 6-2
- Status bar, 2-2
- Switch between modes, 3-1
- Switch between PROG/SETUP, 3-1
- Switch between RUN/PROG, 3-1
- Switch between RUN/SETUP, 3-1
- Symbol handling, 5-33
- Symbol manager,
  - Create, 5-26
  - Delete, 5-27
  - Duplicate, 5-26
  - Edit, 5-26
  - Export, 5-25
  - Import, 5-25
  - More, 5-27
- Symbols,
  - Create, 5-22
- System blocks, 4-6

System program,  
    Down-loading, 2-3  
System signals, 5-38

## T

TCP/IP command object, 6-26  
TCP/IP communication, 18-1  
TCP/IP transfer, 23-5  
Technical data, 24-1  
Terminal as communication inter-  
face, 17-5  
Text objects,  
    Analog object, 7-3  
    Bar graph object, 7-4  
    Date/time object, 7-5  
    Digital text, 7-6  
    Dynamic objects, 7-2  
    Message, 7-8  
    Multiple choice object, 7-9  
    Text object, 7-10  
Textblock editor,  
    Define, 5-24  
    Dynamic object, 5-24  
    Keys, 5-23  
    Mouse, 5-23  
    Open, 5-23  
    Static text, 5-24  
    Toolbox, 5-24  
Textblock printout, 13-2  
Textblocks, 4-5  
    Maneuvering, 7-11  
Tile, 5-19  
Time channels, 5-31, 14-1  
    Define, 14-1  
    Maneuvering, 14-2  
Time control, 14-1  
Time format 5-43  
Time object, 7-5  
Tool box, 5-12  
Touch key object, 6-28

Transfer,  
    Project, 23-1  
Trend object, 6-29  
Trend, 8-1  
    History, 6-34  
    HMI Tools, 8-5  
    Transferring trend data, 8-5  
    Trend files, 8-5  
    Trend object, 8-1

## U

Ungroup object, 5-20  
Update drivers, 5-2, 5-3  
Update, 5-31

## V

VU-meter object, 6-31

## W

Watch-dog, 3-5  
Window menu, 5-51  
Working area, 5-12

