

MAC Operator Terminal

Human-Machine-Interface

User's Manual

MAC 50
MAC 90

Manual MAC 50/MTA-250, MAC 90/MTA-G1

Foreword

This manual covers installation, use and technical data for MAC 50/MTA-250 and MAC 90/MTA-G1.

The terminal is used for the MELSEC PLC system.

The following additional manuals are also available:

- FX0/FX0S/FX0N/FX series, Programming manual
- FX0S series, Hardware manual
- FX series, Hardware manual
- FX0/FX0N series, Hardware manual
- A-serien, Programming manual
- AnU-CPU, User's manual
- AnA-CPU, User's manual
- AnN-CPU, User's manual
- AnS/AnAS, User's manual
- MAC Programmer, Manual
- MAC Programmer+, Manual

© Mitsubishi Electric Europe B.V. 1997

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 type of modification made to the equipment.

If you have any comment or recommendations about the manual, please fill out this form to help us improve. Post, fax or e-mail this page, or a copy of it to our address below.

Comment on the manual MA-00019 rev D	Page

Name

Company

Address

City, Zip code

Country

Telephone.....Telefax

E-mail

Mitsubishi Electric Europe B.V. – Factory Automation
Gothaer Str. 8
D-40880 Ratingen
Fax: +49 2102 486 - 0
E-mail: megfamail@meg.mee.com

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 explosive environment.
- Modifications, changes and additions to the product are forbidden.
- Use only spare parts approved by Mitsubishi Electric Europe B.V.
- Read the user instructions carefully before use.
- This equipment should only be operated by qualified personnel.

At installation

- The product is constructed 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 mounted 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 guarantee 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 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.

Contents

1 Introduction	1
2 Function overview	3
2.1 Basic functions	3
2.2 Modes	7
2.3 Keyboard	7
2.4 Menu system	10
3 Installing terminals	13
3.1 Connecting to the PLC system	13
3.2 Settings	22
4 Basics	25
4.1 Method for programming an application	25
4.2 Menu structure	25
4.3 Block	26
4.4 A small example	27
5 Communication	37
5.1 Terminal in transparent mode	37
5.2 Multidrop communication	41
5.3 Network communication	44
5.4 No protocol mode	47
5.5 Transfer trend data and databuffer	55
5.6 Remote control of the terminal via modem	57
6 Reference instructions	59
6.1 Basic editing functions	59
6.2 Signal format	60
6.3 Setup mode	64
6.4 Programming mode	77
6.5 Run-time mode	105

7 Overview	113
7.1 Modes	113
7.2 Keys	117
7.3 Character settings	121
7.4 Technical data	130
Index	I
Drawings	A-1

1 Introduction

This manual describes the operator terminals MAC 50/MTA 250 and (MAC 90/MTA G1. Hereafter these products are referred to as the *terminal*. To separate the products we name MAC 50/MTA 250 *text terminal* and MAC 90/MTA G1 *graphics terminal*. In the manual we refer to the software packages MAC Programmer/SW-MTA for DOS and (MAC Programmer+/SW-MTA-WIN for Windows. Hereafter these products are referred to as the *PC software*.

The operator terminals are included in a family of terminals developed to satisfy the demands made for human-machine communication. The functions in the terminals provide the possibility to display text and graphics (graphics terminal only) with dynamic display, control, alarm display, report printouts, function keys and time channels. Furthermore, the graphics terminal is used for data and recipe storage as well as trend display.

The terminals are programmed directly from the built-in keyboard or from a PC using the PC software. The programmed project is stored in the terminal.

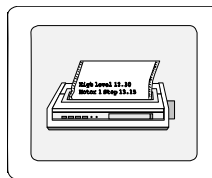
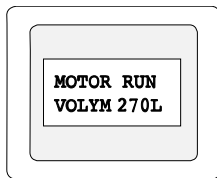
The terminals work mainly in an object orientated way, that is you start with an object and then choose the function you want the object to have. All types of signals are defined according to this principle.

2 Function overview

2.1 Basic functions

Textblock

Textblock is used to build up dialogs and reports. A textblock can consist of an arbitrary number of text lines. The text terminal can display 4 lines à 20 characters and the graphics terminal 16 rows à 30 characters. Each row can be 120 characters long if the textblock is only to be used for print-out on a printer.



Static text

Static text is text written in textblock which is not changed when executing the program.

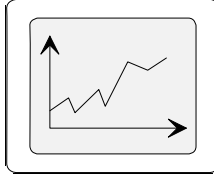
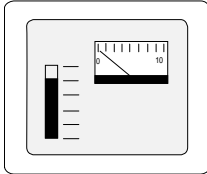
Dynamic objects

Dynamic objects are objects which are linked to the signals in the PLC system. There are seven types of dynamic object: **digital**, **analog**, **jump**, **date**/
time, **bar graph**, **multi-choice object** and **text object**.

Graphblocks

Graphics terminal

Graphblocks are only available in the graphics terminal and are used to create graphic pictures. There are static and dynamic graphics.



Static graphics

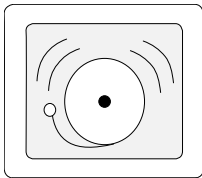
Static graphics form the background graphics such as lines, circles, symbols and texts in various sizes.

Dynamic graphics

Dynamic graphics are graphics which are linked to signals in the PLC system. Examples of dynamic graphics are **bar graphs**, **trend curves**, **analog** and **digital clocks**, **VU meters** etc.

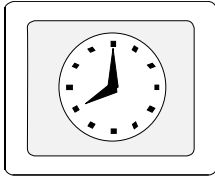
Alarm

In certain cases events can occur in the process which require immediate attention. To meet this requirement there is the possibility in the terminals of linking the signals to an alarm message. The alarm with time stamp is stored in the alarm list in run-time.



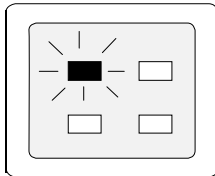
Time channels

The time channels make it possible to set and reset digital signals with the real time clock, the so-called seven-day clock.



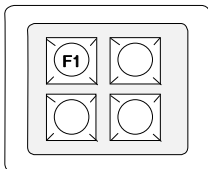
LED

The graphics terminal has 8 LEDs which can assume 4 positions, off, yellow, red and green. The light diodes are controlled by the data register in the PLC system.



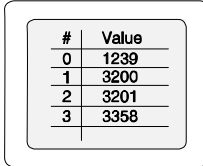
Function keys

The function keys are linked to the digital signals which are set when the function key is pressed and reset when released in run-time mode. Some function keys in the graphics terminal have predefined functions such as [ALARM LIST], [MAIN MENU], [INFO].



Data logging/recipe

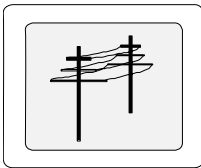
A part of the project memory in the graphics terminal is set aside for logging data or storing menus.



#	Value
0	1239
1	3200
2	3201
3	3358

Communication

The terminals can be used in several communication configurations: *transparent mode*, *multidrop*, *network* and *no protocol mode*.



2.2 Modes

The terminals have three different modes:

- Setup mode
- Programming mode
- Run-time mode

The Setup mode is used for basic settings for such things as choice of PLC system, menu language, character set, format for date and time presentation. Settings apply until changed. The Setup mode is entered as follows: Set the mode switch to the PROG position. You automatically come into the Programming mode. To change to the Setup mode press [LEAVE] twice quickly, that is [F3] (text terminal) or [F8] (graphics terminal).

The Programming mode is the mode where the application is built up. You can create text and graphblocks, define time channels, alarm and function keys. The Programming mode is entered automatically when the mode switch is set to PROG. The terminals can also be programmed with the PC software.

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

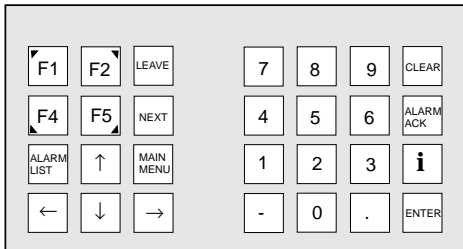
The application is monitored and run in the Run-time mode. You can see how the different objects change and also maneuver certain objects.

The Run-time mode is entered automatically when the mode switch is set to RUN.

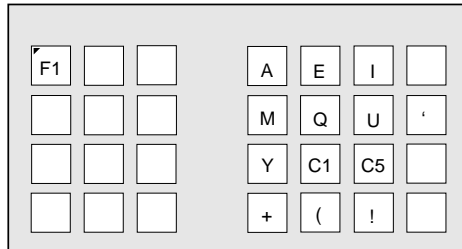
2.3 Keyboard

The terminals have compact keyboards where each key has several functions. There are four further characters, one in each corner of all the number key and the keys [-], [.] and the [ALARM ACK] (text terminal) and [INC] (graphics terminal) keys. To write these in the text terminal use one of the function keys. [F1] for the character in the upper left corner, [F2] for the upper right, [F4] for lower left and [F5] for the lower right. Hold the function key pressed while pressing the required key. In the graphics terminal you use [F10] for the character in the upper left corner, [F11] for the upper right, [F12] for lower left and [F13] for the lower right. Hold the function key pressed while pressing the required key. The codes C1-C8 stand for the character codes specific for your own language.

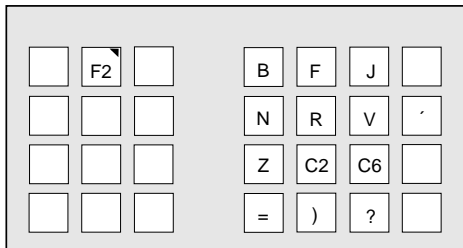
Key functions in the text terminal



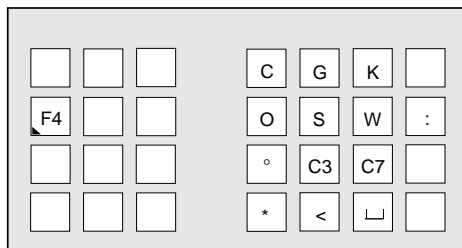
Key functions



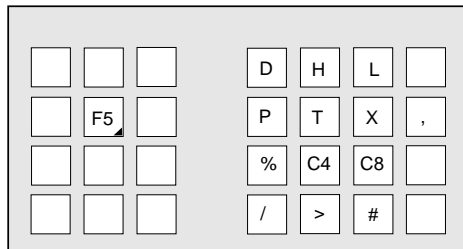
Key functions with [F1] pressed.



Key functions with [F2] pressed.

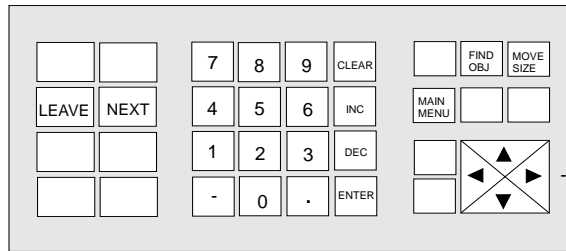


Key functions with [F4] pressed.

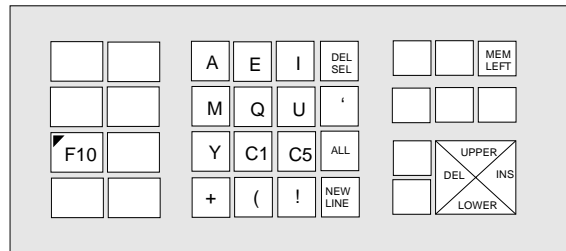


Key functions with [F5] pressed.

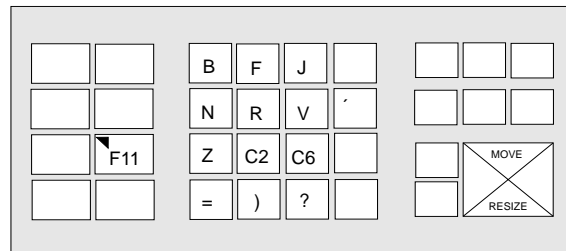
Key functions in the graphics terminal



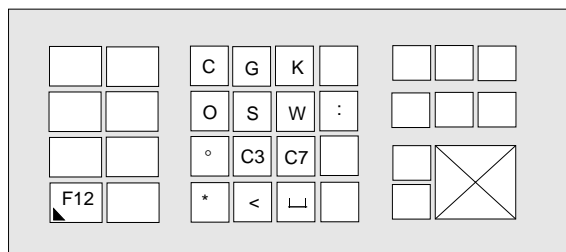
Key functions



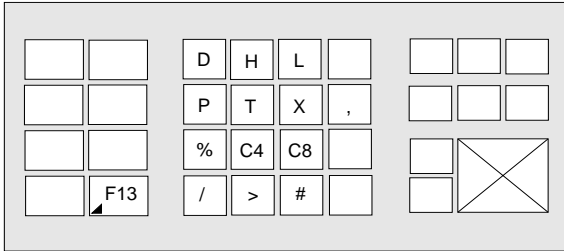
Key functions with [F10] pressed.



Key functions with [F11] pressed.



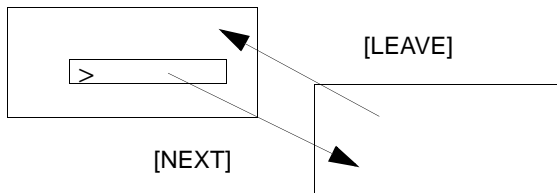
Key functions with [F12] pressed.



Key functions with [F13] pressed.

2.4 Menu system

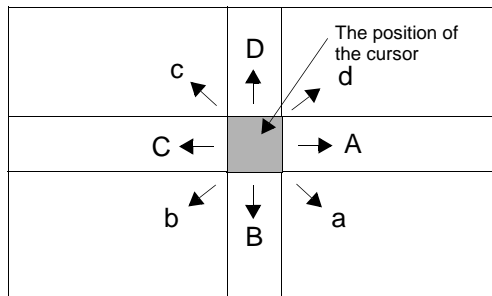
The terminals are designed with a menu system which makes it easy for the user to enter information and monitor his system. There is a general rule when choosing from the menus: Point out the function you require by moving the cursor to the right line with the arrow keys. Then press [NEXT]. If several alternatives exist than there is room for on the display, you can scroll the text with the arrow keys. Press [LEAVE] to exit from a menu.



Choosing maneuverable objects in the run-time mode

Graphics terminal

You can jump between maneuverable objects in the run-time mode with the arrow keys. Choice of object takes place according to the following principle: The cursor's position is assumed to be the center of a cross. Press the right arrow then the first object found in area A in the diagram below is chosen. If the system does not find an object in the narrow field on the right, it searches for it in area a. Arrow down looks for objects in areas B and b, left arrow in areas C and c and arrow up in areas D and d.



3 Installing terminals

3.1 Connecting to the PLC system

Contents of the packet

The terminal packet includes:

- Text terminal or graphics terminal
- Installation instructions

System requirements

The terminals are connected to the PLC system's CPU port or the A series communications ports. The terminals work with programmable control systems of the MELSEC type.

	F	F1	F2	FX0	FX0S	FX0N	FX	AnS	An N	An A	A2C	An U
MAC 50 MTA-250	X	X	X	X	X	X	X	X	X	X	X	X
MAC 90 MTA-G1				X	X	X	X	X	X	X	X	X

If you use a PLC system such as an F-20 or F-40 you need an F-20GC or F-40GC system memory respectively. Changes of PLC signal (maneuvering) do not work with both these types of PLC.

Power requirements

The terminal are powered by a +24 VDC supply. Connections are made on the connection block marked 0 V or +24 V on the back of the terminal. Use the shielded power cable provided. The terminal must be grounded.

Battery check

The terminals have built-in lithium batteries for the project memory, configuration settings and the real-time clock. The battery voltage for these is continually checked. Should the voltage go below the permitted value, 3.21 V (>3.45 V and <2.96 V) BAT.LOW is shown in the display's upper left corner. When this shows the battery should be changed. See the section *Technical data*.

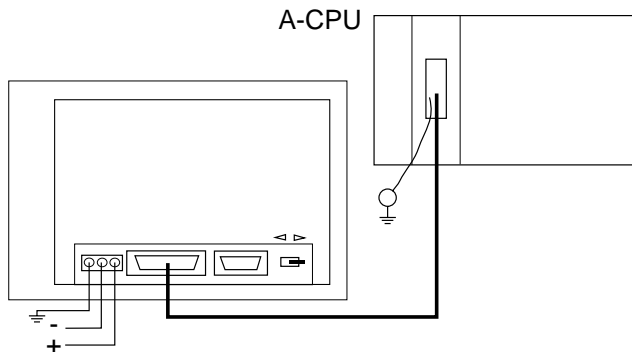
Connecting to the PLC system's CPU port.

Use the cable provided. The 15 pin male contact is connected to the outlet marked RS-422 on the back of the terminal. The 25 pin male contact is connected to the CPU port.

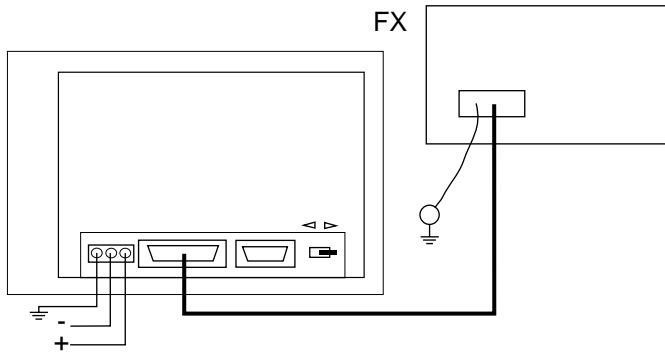
A F2-20GF1 adaptor is required for connection to the F1/F2 series. The F2-20GF1 should be set to a speed of 19200 baud as well as the factory setting.

If the terminal is to be connected to an FX0 system, a MAC-FX0-CAB or MAC-CAB cable together with an FX-20P-CADP must be used.

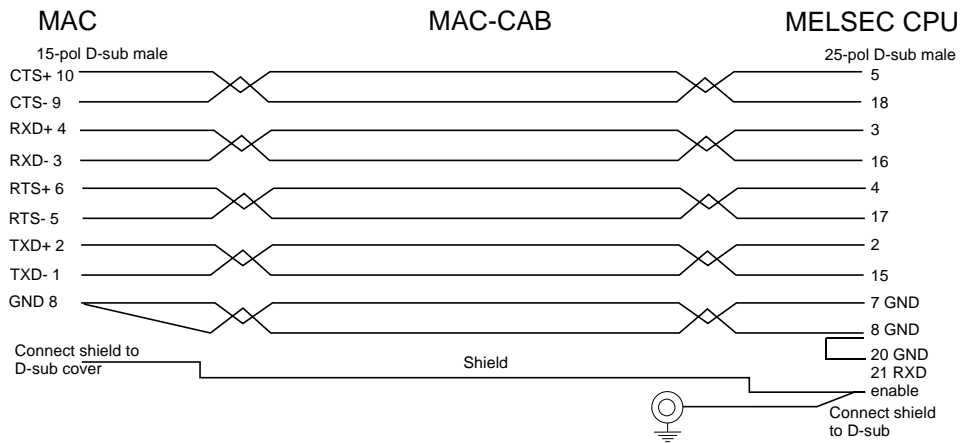
The maximum cable length between a terminal and a PLC system is 15 meters. Should a longer distance between the terminal and the CPU be required use an SC-04.



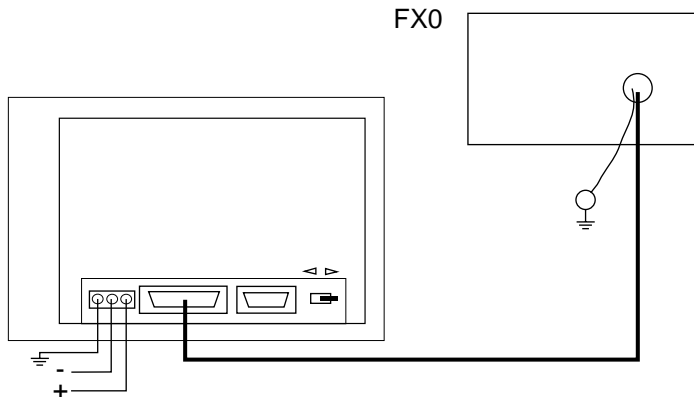
Connection to A-CPU.



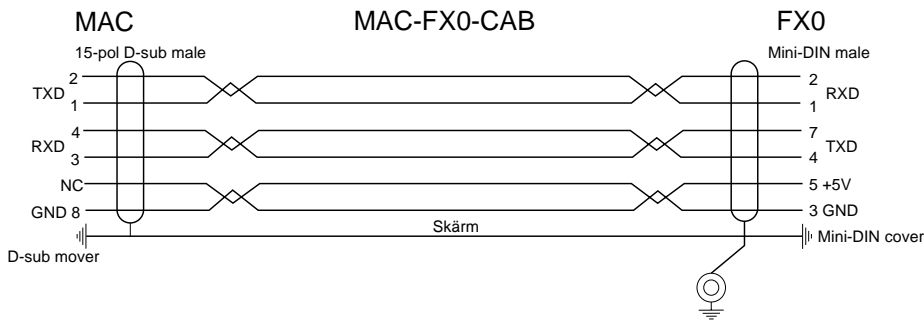
Connection to FX.



Cable configuration for MAC-CAB.



Connection to FX0.

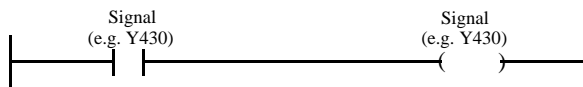


Cable configuration for MAC-FX0-CAB.

Note!

Text terminals connected to F, F1 or F2

Digital objects to be run from the terminal must be included in the PLC program as OUT instructions. Remember that forced settings, function keys etc. only control those digital objects for one program cycle, afterwards the PLC program takes over. The following circuits is therefore useful in many cases:



The functions described in the section *Connecting to MELSEC MEDOC* as well as Communication do not work together with these types of PLC. For these PLC types are only functions corresponding to the text terminal version 2.30 valid.

Connection to an AJ71C24/AJ71UC24

The terminal is connected to the RS-422 port on the AJ71C24.

The switches on the AJ71C24 should be set as follows:

Station number:	00
Mode:	A
Transmission switches:	11, 12, 14, 15, 21 and 22 ON, remainder OFF

Note!

The switches on the AJ71UC24 should be set as above. Furthermore switch 23 should be ON.

The settings result in 19200 baud, 8 data bits, 1 stop bit and no parity.

The parameters in Setup mode under **Port parameters, PLC**, in the terminal should be set in the same way.

Connection to an A1SJ71C24-R4

The terminals are connected to the RS-422 port on the A1SJ71C24-R4.

The switches on the AJS71C24-R4 should be set as follows:

Station number:	00
Mode:	5
Transmission switches:	2, 4, 6, 7, 8 and 12 ON, remainder OFF

The settings result in 19200 baud, 8 data bits, 1 stop bit and no parity.

The parameters in Setup mode under **Port parameters, PLC**, in the terminal should be set in the same way.

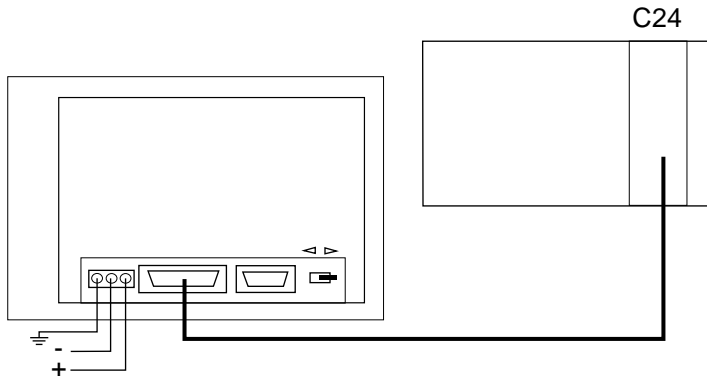
Connection to an A1SJ71C24-R2

The terminals are connected to the RS-232 port on the A1SJ71C24-R2. The PLC system is connected to the RS-232 port or via an RS-232/RS-422 convertor for example an SC04 to the RS-422 port on the terminal. The switches on the A1SJ71C24-R2 should be set as follows:

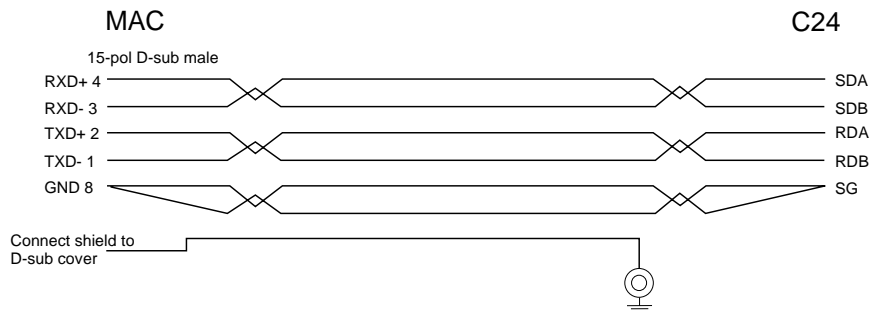
Station number:	00
Mode:	1
Transmission switches:	4, 6, 7, 8 and 12 ON, remainder OFF

The settings result in 19200 baud, 8 data bits, 1 stop bit and no parity.

The parameters in Setup mode under **Port parameters, PLC**, in the terminal should be set in the same way.



Connection to C24.



Cable configuration for the cable between the terminal and AJ71C24.

Connection to an A2CCPUC24

Note!

The terminal can also be connected to the CPU port, see the section *Connecting to the PLC system's CPU port*.

The terminal is connected to the Computer Link port on the A2CCPUC24. The switches on the Computer Link port should be set as follows:

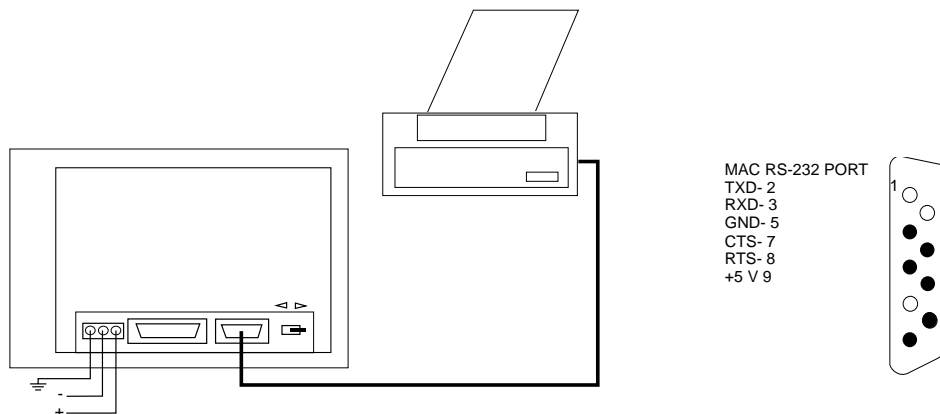
Station number:	00
Mode:	A
Transmission switches:	12, 13, 14, 18, 19 and 20 ON, remainder OFF

The settings result in 19200 baud, 8 data bits, 1 stop bit and no parity.

The parameters in Setup mode under **Port parameters, PLC**, in the terminal should be set in the same way.

Connection to a printer

The printer should have a serial interface and be equipped with IBM character set (850). For graphics printout the printer must be set to emulate EPSON FX-80. The printer is connected to the 9-pin male connector on the back of the terminal. Transmission speed, bit parameters etc are set under **Prt/Transp/No prot** in the **Port parameters** menu. Refer to the printer manual for the correct configuration.



Connection to a printer.

Connection to MELSEC MEDOC

If the terminal is connected to the FX CPU port, the A series CPU port or the A series communications ports (AJ71C24, AJ71UC24, A1SJ71C24-R2, A1SJ71C24-R4 and the Computer Link port on the A2CCPUC24), MELSEC MEDOC can be connected to the terminal's programming/printer port for the so-called transparent mode.

MELSEC MEDOC is used just as if it were directly connected to the PLC system.

For it to function the following settings must be made to the terminal and MELSEC MEDOC.

Terminal settings

The **Prt/Transp/NoProt** parameters are set under the menu choice **Port parameters** in the Setup mode.

The parameters should be RS-232 port, 9600 baud, 7 data bits, 1 stop bit, even parity and station 0 if the terminal is connected to an FX system.

The parameters should be RS-232 port, 9600 baud, 8 data bits, 1 stop bit, odd parity and station 0 if the terminal is connected to one of the A series CPU ports.

The parameters should be RS-232 port, 19200 baud, 8 data bits, 1 stop bit, no parity and station 0 if the terminal is connected to an xxxC24.

MELSEC MEDOC settings

The transmission parameters should not be changed in MELSEC MEDOC for FX projects.

The transmission parameters should not be changed in MELSEC MEDOC if the terminal is connected to one of the A series CPU ports (use the basic settings; 9600 baud, 8 data bits, odd parity and 1 stop bit).

The transmission parameters should be changed to 19200 baud, 8 data bits, 1 stop bit and no parity if the terminal is connected to xxxC24.

This is carried out under Transmit in Setup Interface should be changed from CPU to AJ71 under Transmit.



3.2 Settings

Choice of Setup mode

All the basic settings are made in the Setup mode. Set the switch to **PROG**. You automatically enter the programming mode. To change mode to the Setup mode press [LEAVE] twice quickly.

```
--- SETUP ---  
>PLC choice  
  System signal  
  Index register  
  Language  
  Character set  
  Date/time format  
  Port parameters  
  Erase memory  
  On-line settings  
  Terminal settings  
  Alarm settings  
  Printer settings  
  Contrast settings  
  Data buffer settings  
  Disk settings
```

Contrast settings and data buffer settings are only available in graphics terminal.

Choice of PLC system

You must state in the terminal to which PLC system the terminal is to be connected. The setting is made in the Setup mode.

Move the cursor, `>`, to the PLC choice line and press [NEXT]. The system choice menu is now shown. Move the cursor to the required PLC system and press [ENTER].

```
>FX
  A-CPU
  C24 protocol 1
```

For the text terminal you also have the alternative F1/F2.

Exit from the menu by pressing [LEAVE].

Choice of language and character set

You can choose between different languages for the menus. The choice of language for the menus is carried out in the Setup mode.

Move the cursor, `>`, to the Language line and press [NEXT]. The Language choice menu is now shown. Move to the required language, for example, English and press [ENTER]. The menu language will now be English.

```
British English
German
Swedish
RAM
American English
```

From now on we will assume that English has been chosen as the menu language in diagrams and text.

Press [LEAVE] to exit from the menu.

Many languages have characters specific to that language, ü and ß in German, ñ and í in Spanish, ê and ç in French, å and ä in Swedish, ø and æ in Norwegian and Danish, most with their equivalents in capitals. Using the menu choice Character set, you can set which of the character codes C1-C8 corresponds to which character. The following menu is shown after selecting the Character set in the Setup menu.

```
>Swedish
  English
  French
  Spanish
  Italian
  Nor/Dan
  Katakana
  Cyrillic
```

Katakana and Cyrillic are the Japanese and Russian characters respectively.

The terminals can use Katakana and Cyrillic characters according to the following table.

Terminal	Character set
MAC 50/ML MTA-250/ML	KATAKANA
MAC 50/MV MTA-250/MV	KATAKANA and CYRILLIC
MAC 90/MTA-G1	-

Character editing for setting up Katakana and Cyrillic takes place in the terminal. This means that the PC software can only be used for back-up copies.

4 Basics

This chapter describes how the terminals are constructed and gives a simple example to allow you quickly to be able to start using the terminal.

4.1 Method for programming an application

When the terminal is installed you must carry out the basic settings according to the conditions of the plant. Settings, choice of PLC system for example, are carried out in the Setup mode. You then switch to the programming mode to program the application. Start by planning and structuring the application so you know which block you are going to define, what type of block you are going to use as well as the order they are to be defined.

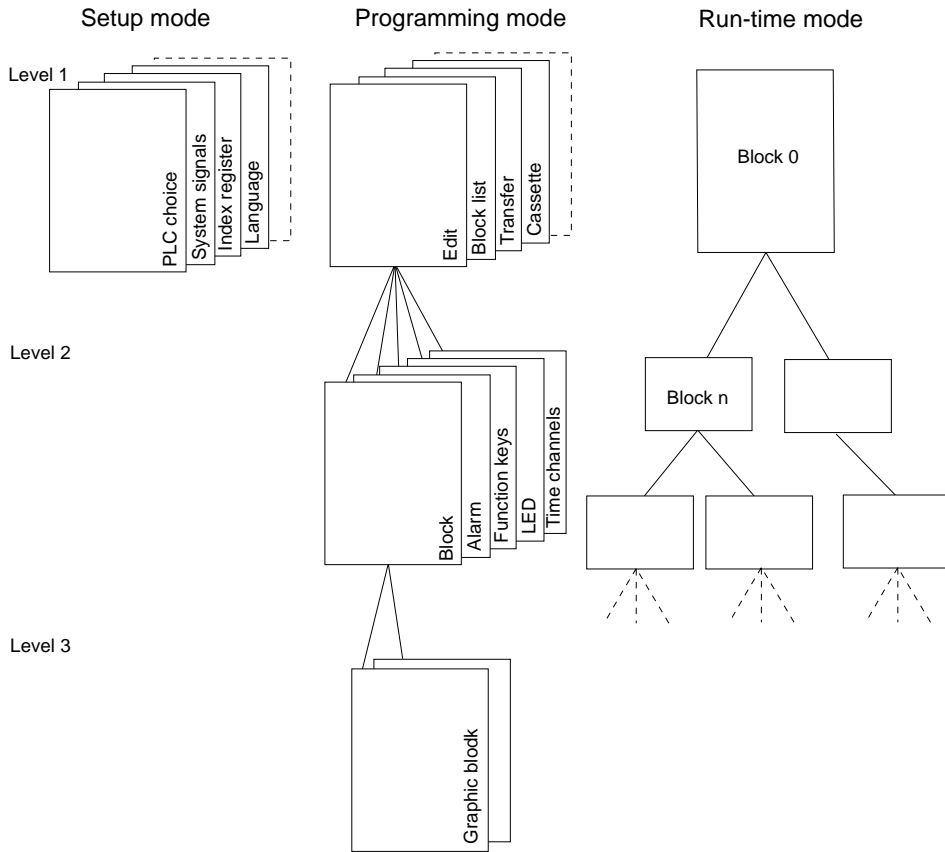
You have the possibility to test the whole or part of the application in the run-time mode before it is used run-time.

4.2 Menu structure

The terminal is divided into three modes (function areas), Setup, programming and run-time. There are a number of levels in each mode depending on the function. Each level consists of a menu where you make a choice or state parameters to proceed to the next level (menu).

The application is built up of **blocks**, Textblocks or Graphblocks (graphics terminal only). The values from the PLC system are shown and altered in the block. The terminal is object oriented which means that a block can contain all the signals linked to the object for controlling and monitoring a pump for example.

Furthermore there are separate functions for alarm, time channels, function keys and LEDs.



4.3 Block

A block header is defined for each block. You state the block number, type of block, status word, password etc in the block header (see Reference description).

Note!

The block type cannot be changed for a defined block.

The alarm, time channels, function keys and LED functions can also be called up as blocks. These are called system blocks.

To get a well structured application the blocks should be created in a hierarchy conforming to an organization natural for the machine operator.

4.4 A small example

To start creating applications quickly, the following section gives a small example. The example we are going to simulate is the monitoring of a water tank and the time control of a fan. The first example is for the text terminal and then for the graphics terminal.

Text terminal

Switch on the terminal and put the mode switch on the back in the **PROG** position. After a short period the programming menu is shown where the **Edit** alternative is already marked:

```
-- PROGRAMMING --
>Edit
  Block list
  Transfer
```

Press [NEXT] to enter the Editing menu:

```
>Textblock
  Alarm
  Time channel
  Function keys
```

Then press [NEXT] to edit your textblock:

```
BLOCK NUMBER: 0
Display signal:
Printer signal:
```

Ensure that the number on the textblock is 0 and press [NEXT] to enter the text editor. An empty screen with the cursor in the top left-hand corner is shown. The first step is to enter static text. To find the letters on the keyboard use the shift keys [F1][F2][F4][F5]. Enter 'LEVEL:'.

The next step is to make a dynamic object to show the level in the water tank. To display the menu for dynamic objects press:

```
#      [F5][.]
```

Go down to **Analog** with the arrow key and press [NEXT]. The list for an analog object is now displayed. We assume that the level in the water tank is stored in the D0 register and therefore enter D0 next to **A.Obj.**

The object must be made maneuverable to be able to simulate the level in the water tank. Go down to the **Maneuver** line and change to Yes by pressing the [INFO] key. The level is assumed to be able to vary between 0 and 100. Enter 0 for **Min** and 100 for **Max**. (To erase values already there press [F1][←].)

Ensure that the following values are entered:

```
A.Obj. :      D0
Position:     5
Decimals:     0
Maneuver:     Yes
Min:          0
Max:          100
Offset:       0
Gain:         1
```

Exit from the list with [LEAVE] and you return to your textblock. You can see how the dynamic object is represented by '#---'.

We shall now make a dynamic object which illustrates the level graphically. Press [ENTER] to come down the beginning of the next line and then press # to display a menu for the dynamic object. Go down with the arrow keys and select **Bar graph** by pressing [NEXT]. Enter D0 next to **A.Obj**, 0 for **Min** and 100 for **Max**.

Ensure that the following values are entered:

```
A.Obj. :      D0
Position:     20
Direction:    Right
Min:          0
Max:          100
Offset:       0
Gain:         1
```

Exit from the list with [LEAVE] and you return to your Textblock.

We shall now show the state of the fan. (How it is time controlled is shown a little later). Press [ENTER] so that the cursor is in the bottom left-hand corner of the screen and enter the text 'FAN:'. Press # and select **Digital** to create a dynamic object showing the fan's state. We assume that the fan is controlled via the M0 memory cell.

Enter M0 next to **D.Object** and then enter the text 'OFF' next to **Text 0** and 'ON' next to **Text 1**.

Ensure that the following values are entered:

```
D.Obj. :      M0
Text 0 :      OFF
Text 1 :      ON
Maneuver :    NO
```

Exit from the list with [LEAVE]. The screen now has the following display:

```
LEVEL: #---
#-----

FAN:  #---
```

We shall now make an alarm object which warns when there is a risk for the water tank overflowing. Go back with [LEAVE] and select **Alarm** in the Editing menu.

```
Textblock
>Alarm
Time channel
Function keys
```

Enter the text in the alarm editor which will be shown in the alarm list when an alarm occurs in run-time. Enter the following:

```
>TANK FULL
```

Then press [NEXT] to define the parameters in the alarm object.

We assume that overflow sensor goes to input X0. Enter X0 next to Signal and do not change the other parameters:

```
Signal:      X0
Printer:     NO
Acknowledge: YES
History:     YES
Ack Sign:
```

We shall now make a time channel for controlling the fan. Go back with [LEAVE] and select **Time channels** in the Editing menu:

```
Textblock
Alarm
>Time channels
Function keys
```

You can enter the text in the time channel editor which identifies the time channel:

```
>FAN
```

Then press [NEXT] to define the parameters in the time channel. We assume that the fan is controlled via the M0 memory cell. Enter M0 next to **Signal**. Then enter the time intervals during which the fan will operate. A maximum of four intervals can be stated but we shall just give one:

```
Signal:M0
1:      Mo-Fr 0800-1700
2:
3:
4:
```

Press [LEAVE] twice to exit from the time channel editor.

Now you can test our example. Set the mode switch on the back to **RUN**. After a short time the Textblock 0 is shown automatically

```
LEVEL: 0
. . . . .

FAN:  ON
```

Press the right arrow to come to the field for Level. Then enter 50 and press [ENTER]. The bar will now be half full.

Switch X0 on to get an alarm. The text **ALARM** is shown in the top right-hand corner. Press [ALARM LIST] to enter the alarm list then [ALARM ACK] to acknowledge the alarm. Press [ALARM LIST] to return to the Textblock you came from.

The status of the fan will show ON Monday to Friday, 08.00 to 17.00 and OFF the rest of the time.

Graphics terminal

Switch on the terminal and put the mode switch on the back in the **PROG** position. After a short period the programming menu is shown where the **Edit** alternative is already marked:

```
>Edit
  Block list
  Transfer
```

Press [NEXT] (key [F9]) to enter the Editing menu:

```
>Block
  Alarm
  Time channel
  Function keys
  LEDs
```

Then press [NEXT] to edit your block:

```
BLOCK NUMBER: 0

Graphics/Text:Graphics
.
.
```

Ensure that the number on the block is 0 and press [NEXT] to enter the graphblock editor. An empty screen with cross hairs in the middle is shown. The first step is to enter text. Move the cross hairs to the upper left-hand corner and press [MAIN MENU]. A menu of available objects is shown on the screen. Go down to **Text** with the arrow key:

```
Line
Rectangle...
Ellipse...
Curve
>Text
  Symbol
.
.
```

Then press [NEXT]. A cursor now appears on the screen. To find the letters on the keyboard use the shift keys [F10][F11][F12][F13]. Enter 'LEVEL:' and press LEAVE (key [F8]) twice.

The next step is to make a dynamic object to show the level in the water tank. Place the cross hairs to the right of the text 'LEVEL:' and press [MAIN MENU] to get the object menu. Select **Analog** and then **Numeric**. The list for an analog numerical object is now displayed. We assume that the level in the water tank is stored in the D0 register and therefore enter D0 next to **A.Signal**. The object must be made maneuverable to be able to simulate the level in the water tank. Go down to the **Maneuver** line and change to YES by pressing the [INFO] key. The level is assumed to be able to vary between 0 and 100. Enter 0 for **Min** and 100 for **Max**. (To erase values already there press [F10][→]).

Ensure that the following values are entered:

```
A.Signal:    D0
Position:    5
Decimals:    0
0-fill:      NO
Adjust:      Left
Maneuver:    Yes
Min:         0
Max:         100
Offset:      0
Gain:        1
Frame:       NO
```

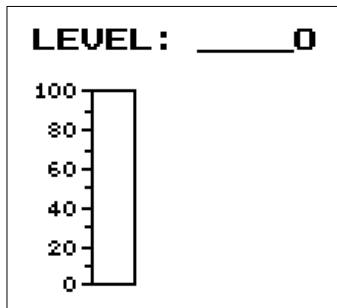
Press [LEAVE] twice to return to your block. You can see how the dynamic object is represented by '___0'.

We shall now make a dynamic object which illustrates the level graphically. Move the cross hairs under the text 'LEVEL:' Press [MAIN MENU] and select **Analog** and then **Bar**. Enter D0 next to **A.Signal**.

Ensure that the following values are entered:

```
A.Signal:      D0
Min:           0
Max:           100
Direction:     UP
Scale:         YES
Box:           YES
Indicator:     NONE
Scale spacing: 20
Scale division:10
Fill pattern:
Offset:        0
Gain:          1
Frame:         NO
```

Press [LEAVE] twice to return to your block. Your screen should now look like this:

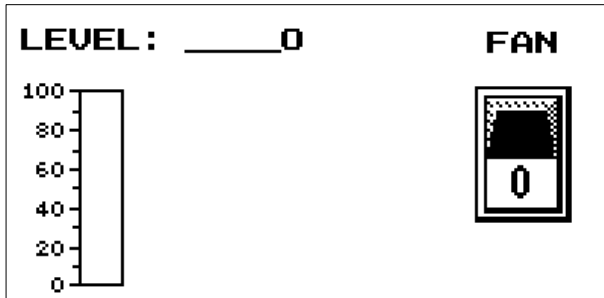


We shall now show the state of the fan (how it is time controlled is shown later). Move the cross hairs to the right of '___0' and enter the text 'FAN' in the same way as the text 'LEVEL' earlier. Then place the cross hairs under the text 'LEVEL' and press [MAIN MENU] select **Digital** and then **Symbol**. We assume that the fan is controlled via the M0 memory cell. Enter M0 next to **D.Signal**. Then go down a line and press [INFO] by **Symbol 0**. Select rocker0 and press [ENTER]. In the same way select rocker1 next to **Symbol 1**.

Ensure that the following values are entered:

```
D.Signal.: M0
Symbol 0: rocker0
Symbol 1: rocker1
Maneuver: NO
```

Press [LEAVE] twice to return to your block. Your screen should now look like this:



We shall now make an alarm object which warns when there is a risk for the water tank overflowing. Go back with [LEAVE] and select **Alarm** in the Editing menu.

```
Block
>Alarm
Time channel
Function keys
LEDs
```

Enter the text in the alarm editor which will be shown in the alarm list when an alarm occurs in run-time. Enter the following:

```
TANK FULL
```

Then press [NEXT] to define the parameters in the alarm object. We assume that the overflow sensor goes to input X0. Enter X0 next to **Signal** and do not touch the other parameters:

```
Signal:      X0
Printer:     NO
Acknowledge: YES
History:     YES
Ack Sign:
```

We shall now make a time channel for controlling the fan. Go back with [LEAVE] and select Time channels in the Editing menu:

```
Textblock
Alarm
>Time channels
  Function keys
  LEDs
```

You can enter the text in the time channel editor which identifies the time channel:

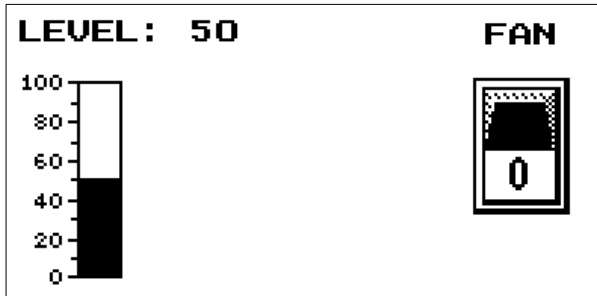
```
>FAN
```

Then press [NEXT] to define the parameters in the time channel. We assume that the fan is controlled via the M0 memory cell. Enter M0 next to **Signal**. Then enter the time intervals during which the fan will operate. A maximum of four intervals can be stated but we shall just give one:

```
Signal:      M0
1:           Mo-Fr 0800-1700
2:
3:
4:
```

Press [LEAVE] twice to exit from the time channel editor.

Now you can test our example. Set the mode switch on the back to RUN. After a short time the block 0 is shown automatically. Enter 50 next to LEVEL and press [ENTER]. The bar will now be half full.



Switch X0 on to get an alarm. The text ALARM is shown in the top right-hand corner. Press [ALARM LIST] to enter the alarm list and then [ALARM ACK] to acknowledge the alarm. Press [ALARM LIST] to return to the block you came from.

The status of the fan will show the 1 button Monday to Friday, 08.00 to 17.00 and the 0 button the rest of the time.

5 Communication

This chapter describes the different communication configurations that the terminal can be used in. The terminal has two serial ports, one RS-232 port and one RS-422 port.

IFC 50 is an RS-232/RS-422-converter which can either be connected to the RS-232 or RS-422 port on the terminal. The converter is attached to the battery cover at the back of the terminal. IFC 50 gives a RS-422/RS-485 interface, which is galvanically isolated from the terminal.

See the IFC 50 manual for correct connection.

5.1 Terminal in transparent mode

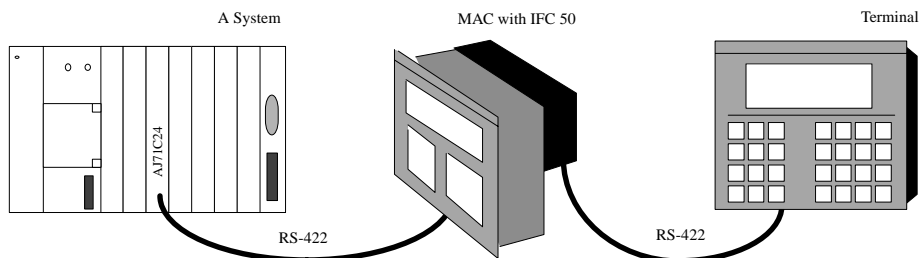
In this mode the programming/printer port on the terminal is used to connect a parallel working unit to the PLC system. This unit can be another terminal or a PC with, for example, the programming tool MELSEC MEDOC or a supervisory operator system.

Transparent mode works together with the FX CPU port, A-CPU port and A series communications ports (AJ71C24, A1SJ71C24-R4, A0J2-C214 and the RS-422 port on A2CCPUC24).

Connecting two terminals

With two terminals connected the IFC 50 is used as a RS-232/RS-422 converter between the terminals.

If the connection between terminals is less than 15 meters long the RS-232 can be used. The other terminal's PLC communication is directed to the RS-232 in the communications set-up for the PLC system.



PLC system with two terminals.

When connecting two terminals to an FX system or an A-CPU port the first terminal is connected using the enclosed standard cable in the normal manner. The second terminal is connected to the IFC50 on the back of the first terminal.

When connecting two terminals to the A-series communication ports the first terminal is connected as described in the Installation chapter. The second terminal is connected to the IFC50 on the back of the first terminal.

Settings for two terminals

For transparent mode to work, the following settings should be made in the terminals.

The first terminal

In Setup mode, set printer parameters at the menu choice

Port parameters. Baud rate can vary between 600-38400 baud for the text terminal and 600-57600 baud for the graphics terminal. Other parameters should be 8 data bits, 1 stop bit and no parity. It is recommended that the highest transmission speed is used between the terminals to achieve the best performance. If a text terminal and a graphics terminal are used together, the highest transmission speed is 19200 baud.

Between the terminals there should always be 8-bit communication.

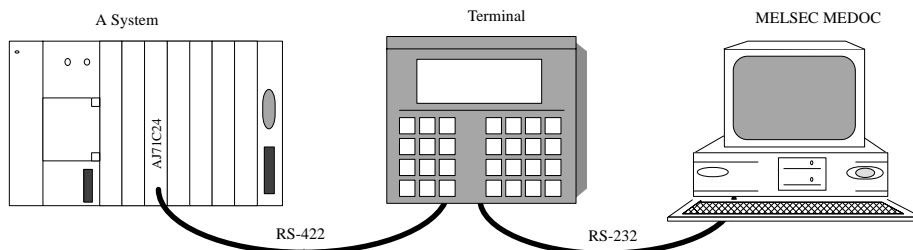
The second terminal

In Setup mode, set PLC parameters at the menu choice **Port parameters**. Baud rate and other parameters should be the same as the printer parameters for the first terminal.

See the chapter *Overview* for the cable configuration.

Connecting PCs or other computer systems

PCs with the MELSEC MEDOC programming tool or other computer systems are connected directly to the terminal's programming/printer port. The terminal is connected as described below.



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

Settings in the terminal and MELSEC MEDOC

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

The terminal

In Setup mode, set printer parameters at the menu choice **Port parameters**.

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

If the terminal is connected to one of the A-series communications modules the parameters should be 19200 baud, 8 data bits, 1 stop bit and no parity.

If the terminal is connected to an A-CPU the parameters should be 9600 baud, 8 data bits, 1 stop bit and odd parity.

MELSEC MEDOC

For transmission via the A-series communication ports (C24) the transmission parameters should be changed to 19200 baud, 8 data bits, 1 stop bit and no parity. This is done at “Transfer” in “Setup”. At “Transfer”, also “Interface” is changed from CPU to AJ71.

For other projects, the transmission parameters should not be changed in MELSEC MEDOC. See the Appendix for cable configuration.

The settings are also valid for other PC based supervision systems e.g. SCADA.

Limitations and advice

The communication throughput drops as the number of connected terminals increases, according to the following table.

	Access time to PLC			
Terminal 1	100%	50%	50%	50%
Terminal 2	-	50%	25%	25%
Terminal 3	-	-	25%	12,5%
Terminal 4	-	-	-	12,5%

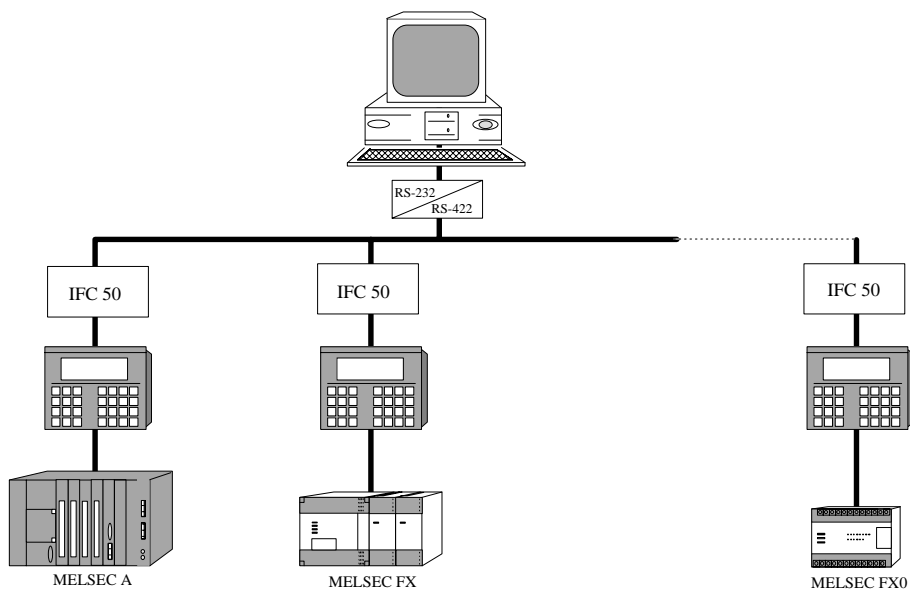
In the terminal connected to the PLC system you should define the BR command on the command line (line at the bottom) in the System signal menu at Setup.

5.2 Multidrop communication

The terminal can be used in multidrop communication. Multidrop communication is used if you want a central program support station for back-up copying or central PLC programming with MELSEC MEDOC, and central supervision and data storing. Multidrop communication also works via modem, for remote supervision and/or remote programming of the PLC system.

Connection example

A multidrop network can appear as follows.



Multidrop communication

Settings in the terminal

Make the settings under **Port parameters**. For multidrop communication you should choose **Prt/Transp/No prot** under port parameters.

The following menu appears:

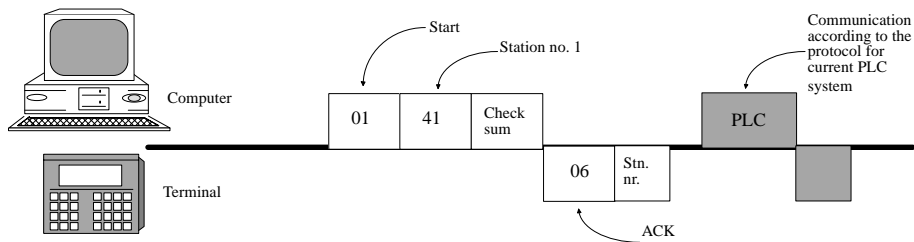
```
Port:      RS232
Baudrate:  9600
Databits:  8
Stopbits:  1
Parity:    NONE
Station:   0
```

Make the settings according to the configuration you will use.

Station number can be 1-32. Station number 0 means no multidrop communication is used.

Connecting from an external unit

The external unit “calls up” the terminal/PLC system according to the following protocol. Subsequent communication between the computer and PLC system runs according to the communication protocol in the PLC system. That is, normal transparent mode is used for communication between Master and PLC system.



Multidrop, setting of the external unit.

The Master must specify the PLC system which communication is directed towards by sending the following message. This message is interpreted by connected terminals. The terminal programmed with the desired station number answers with an ACK message and then opens for transparent mode to the PLC system. It remains open until the next connection sequence occurs.

Start character	01 Hex
Identification character	Station number + 40 Hex (bit six set)
Check sum	Start character + Identification character

The terminal answers as follows:

ACK	06 Hex
Identity	Station number + 40 Hex (bit six set)

Example:

To select station 4, the external unit sends as follows:

Start character	01 Hex
Identification character	44 Hex
Check sum	45 Hex

Note!

If you switch from RUN to PROG on the terminal, communication is broken and a new connection must be made.

Limitations and advice

All terminals must have the same parameter settings except station number, which is always unique for each terminal.

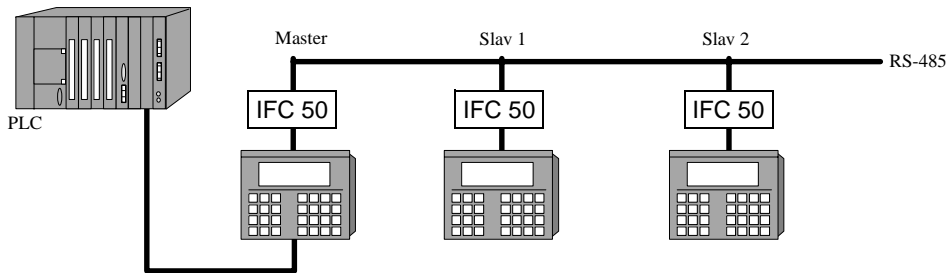
A maximum of 32 terminals can be connected in a multidrop net.

If the Master can manage different protocols it is possible to use different MELSEC-PLC systems in the same net. For example FX-system and A-system.

The PC software diskette contains a PC-program (HOST.EXE) which will link a terminal to a multidrop net.

5.3 Network communication

Network communication is used if you want to connect more than two terminals to a PLC system and simultaneously maintain a good performance, for example for production lines where you want a terminal at each work station. A network can comprise one master and up to seven slaves. The terminal connected to the PLC system must be the Master.



Network communication

Setting the terminals

After start-up, select the PLC system. In the Setup menu select **Port parameters**. Select **Network**.

Under Network you can define the following parameters according to your needs.

Port:	RS422
Baudrate:	38400
Databits:	8
Stop bits:	1
Parity:	NONE
Station:	NONE

It is recommended that the highest transmission speed is used in the network to achieve the best performance. If both text terminal and graphics terminal are used in the same network, the highest transmission speed is 19200 baud.

At station, select MASTER for the terminal connected to the PLC-system, and SLAVE 1 to 7 for the other terminals in the network.

When all terminals are set, create a text- or graphblock in each respective terminal. Switch the terminals to RUN mode. Start with the slaves, and take the master last. Following is the display at switchover to RUN mode. In the example, the master has found slaves 1, 3 and 4. It can vary depending on the application.

```
Network startup
Stn   1  2  3  4  5  6  7
Found X      X  X
Seek
```

Limitations and advice

The network is controlled by events, that is to say data is transmitted when a change occurs in one of the systems in the network.

To achieve the best performance, you should think about minimizing the number of signals and to use the minimum possible number of signals which change the state often.

The total number of alarms, display signals and printer signals cannot exceed 1024.

A sum total of 512 digital objects can be displayed on all terminals at any one time.

A sum total of 180 analog objects can be displayed on all terminals at any one time.

Transparent mode and No protocol mode and data storage are not usable in network.

Use I/O in consecutive sequence for optimum performance. Collect alarm signals in one area, display signals in a second and printer signals in a third.

To show a 32-bit counter in a FX system you must use double data registers.

Graphics terminal only

The application in the master terminal should not hold large symbols.

Clock synchronization

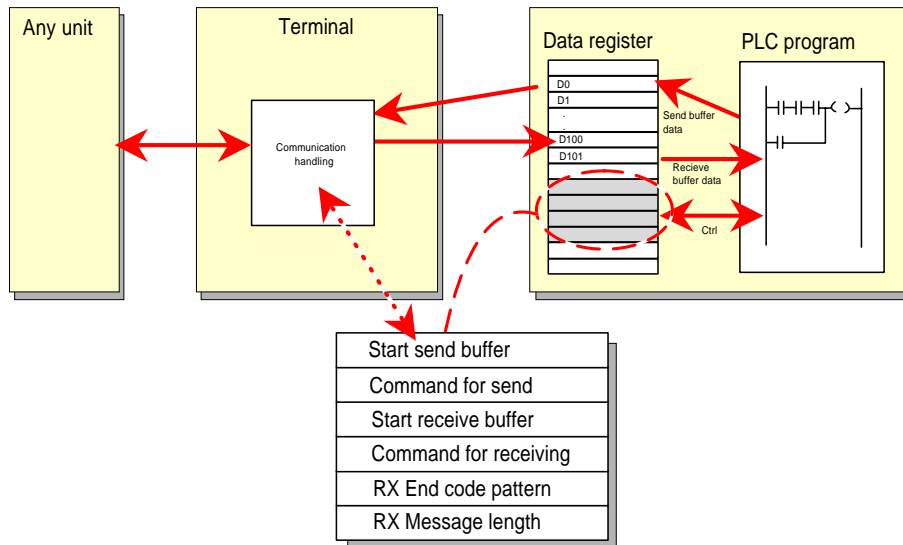
The terminals' clocks synchronize at start-up and thereafter once each minute. Note that the Master's clock takes precedence. All slaves will use their own clocks, while the Master can choose between its own and the clock in the PLC system. Slaves may not pass their clock status to the PLC system.

Note!

Make sure that the grounding is completed according to the installation instructions.

5.4 No protocol mode

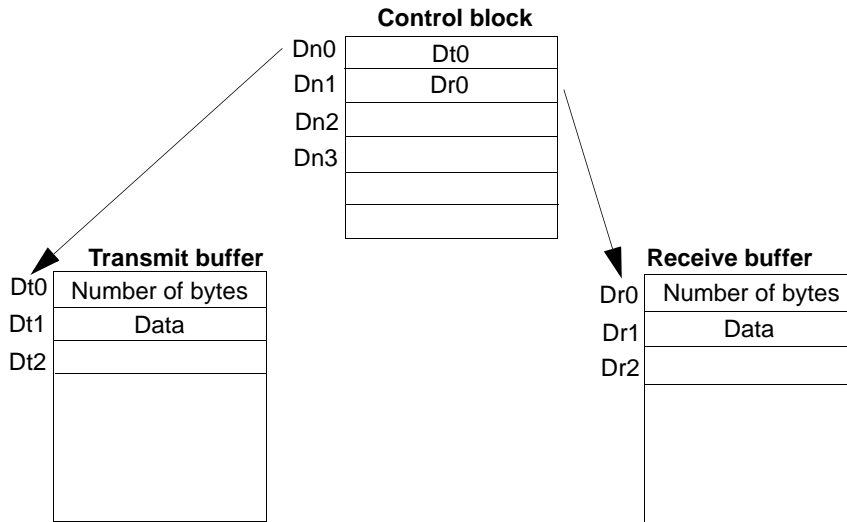
No protocol mode is used to connect different PLC systems, and to connect external units such as bar code readers, weighing scales or phone modems to the PLC system. The PLC system controls what is to be transmitted to the communication port. Data received at the communication port is copied into PLC register. Communication proceeds with transmission of the data register area according to the following control block.



Register	Description
Dn0	Start register for transmit data buffer. First register in buffer contains total bytes which will be transferred. Max buffer is 127 register = 254 bytes. The following registers contain data which will be transferred.
Dn1	Command register for transmit. Changed to 1 by the PLC system when transfer is desired. Changed to 0 by the terminal when transfer is complete.
Dn2	Start register for receive data buffer. First register in buffer contains total received bytes. Max buffer is 127 register = 254 bytes. Following contain received data.
Dn3	Command register for receiving. - Set to 0 by the PLC system when it is ready to receive. - Set to 1 by the terminal when the message is available. - Set to -1 (FFFF) in the event of a faulty message (e.g. too short). - Set to 2 by the PLC system to request clearing. - Set to 3 by the terminal when the port buffer is cleared. Port buffer clears automatically at start-up and at switchover between transparent mode and no protocol mode; i.e. the register gets the value '3'.
Dn4	End code (1 or 2 bytes) on received messages.
Dn5	Length of received message. If 0, end codes is used.

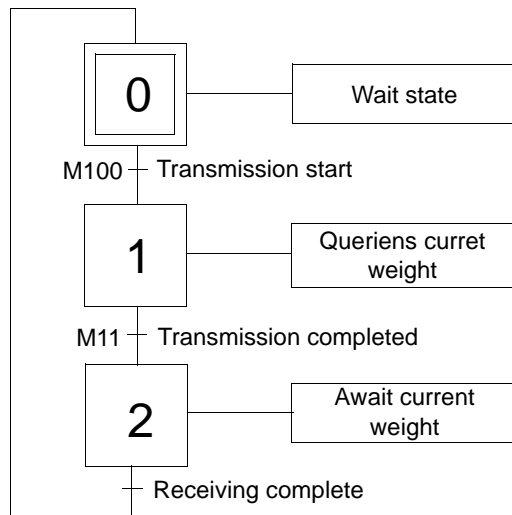
In the Setup menu at System signals you define which register will be the first control register in the transmission area. This and the five following registers are used as control registers. If no register is specified, the terminal runs in transparent mode.

During Run-time mode the PLC system can shift between “no protocol mode” and “transparent/printer mode”. See command NP in System signals.

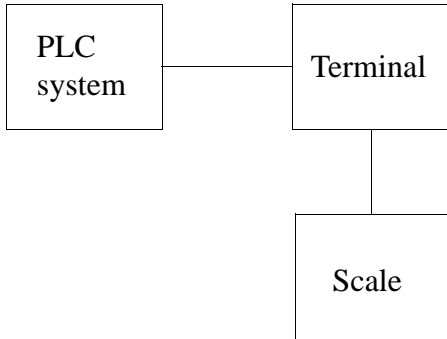


Examples of use of no protocol mode

The following examples describe the use of no protocol mode for a weighing system. The illustration below shows connection and a “three-step block plan” of communication.



The terminal works as a master for the weighing system, i.e. it continually queries the weighing system about the current weight. Connection between the PLC system, terminal and the scale looks like this.



The protocol looks like this.

Terminal

|STX|?|CR|LF|

Weighing system

|STX|Weight in kg|CR|LF|

```
*****
*                               *
*                               *
*****
```

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

```
*****
*                               *
*                               *
*****
```

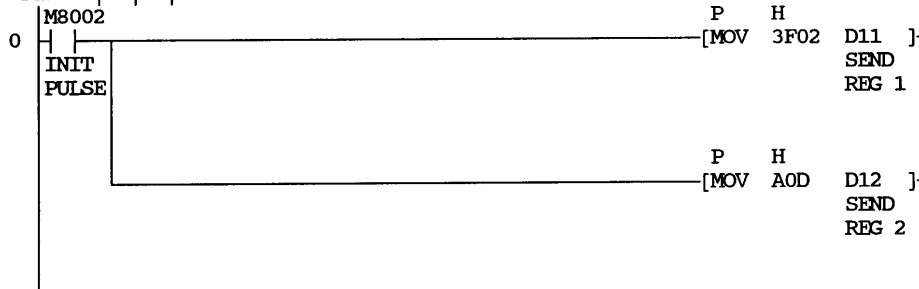
Initiation of question to the Weighing-machine

|STX|?|CR|LF|.

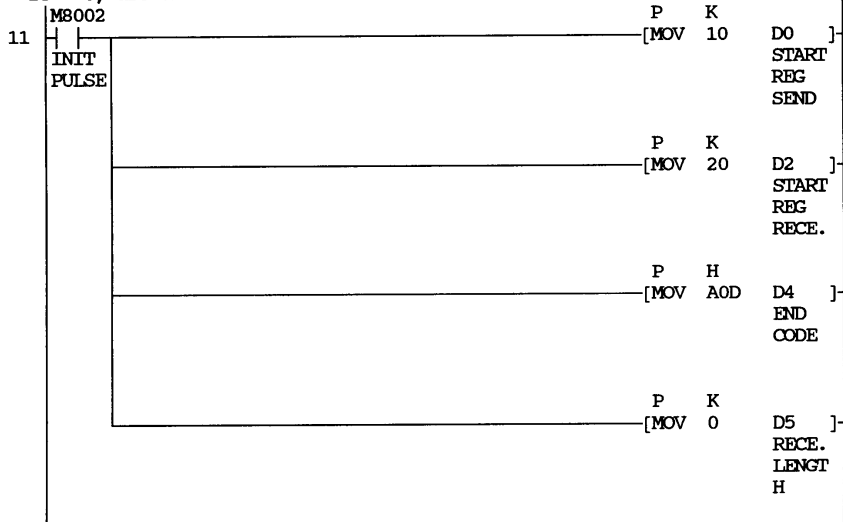
Note the order of the commands in D11 and D12.

D11 = |?|STX| = H3F02

D12 = |LF|CR| = HOA0D

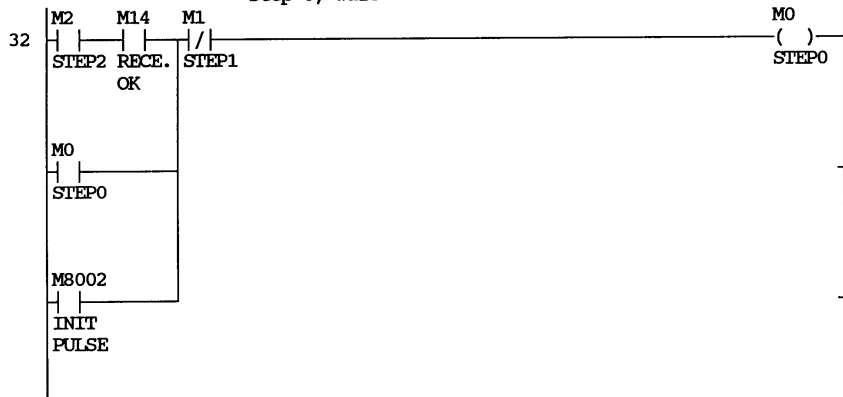


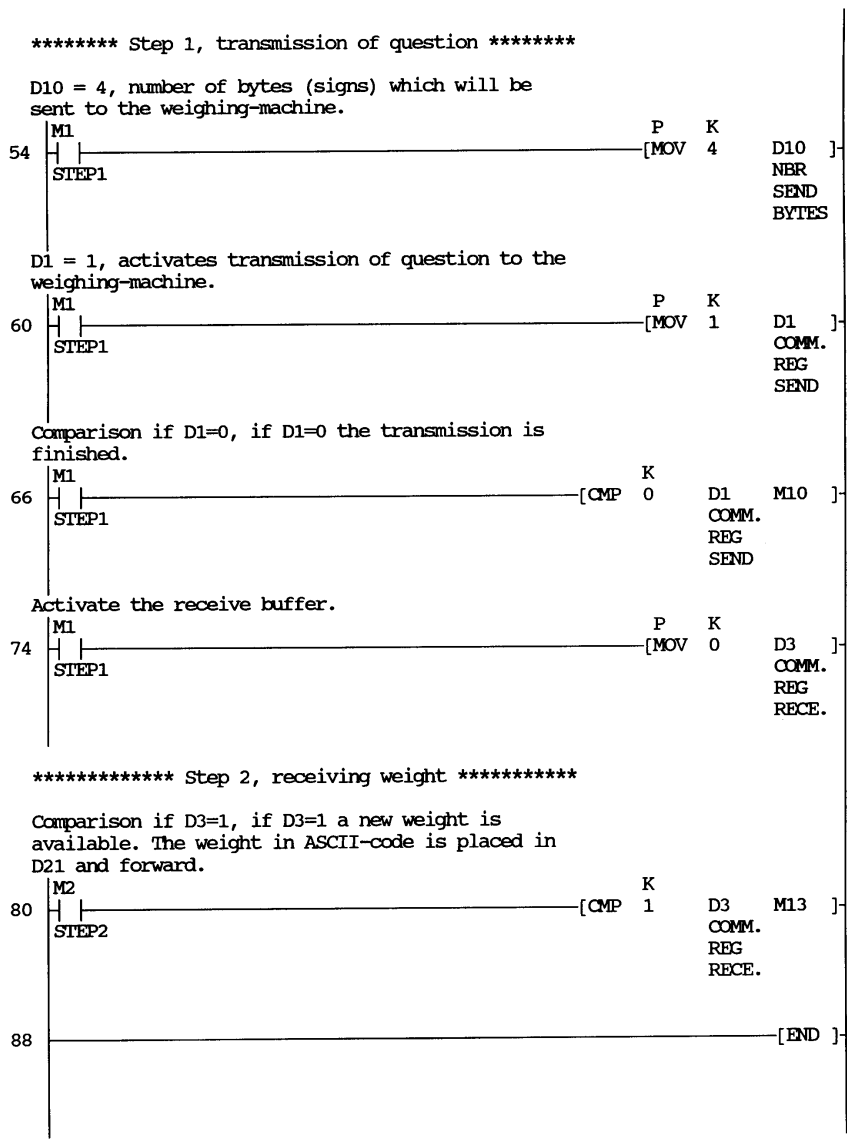
Initiation of command registers D0-D5.
 D0 = Start of the send buffer, K10 = D10.
 D2 = Start of the receive buffer, K20 = D20.
 D4 = End code |CR|LF|.
 D5 = 0, use end code.



 * SEND/RECEIVE-SEQUENCE *

***** Step 0, wait *****





5.5 Transfer trend data and databuffer

Graphics terminal

Trend data and databuffer can be transferred to a PC via a device driver. The device driver MACMTA.SYS is delivered on the PC software diskette. The function is used to transfer trend data and data log values for statistical calculations and other types of presentation and storage.

The terminal and the PC communicate via the serial port. The device drive enables the PC to handle the terminal as if it was a drive e.g. E:. The driver is named automatically.

On this drive the trend data and data log are stored in ASCII files with the extension stated in the Setup, Disk settings. The files can be opened, viewed, copied, printed etc. using the normal DOS commands or by using the File Manager in Windows. Trend files and datalog values can be directly opened from within Microsoft[®] Excel for statistical calculations. You can also change the values and transfer the changed values to the terminal by simply copying or saving the file to the drive.

Install the device driver

Put the file MACMTA.SYS in your CONFIG.SYS file e.g.

```
Device=C:\driverdirectory\MACMTA.SYS 1 19200 128
```

1 is the port, 19200 is the baud rate and 128 is the disk size in bytes. Disk size must be the same as defined disk size in the menu Disk settings. Port can be 1 or 2. The baud rate must be the same as in **Prt/Transp/No prot, Port parameters**. Other settings must be 8 data bits, 1 stop bit and no parity.

Trend files

The names of the trend files are assigned for each trend when defining the trend object. The file is assigned the extension .SKV if nothing else is entered in the Disk settings. The format for each line in the trend file is:

YY-MM-DD;HH:MM:SS;AAAAA;BBBBB

YY-MM-DD	Date format stated in the setup
HH:MM:SS	Time format stated in the setup.
AAAAA	Trend curve 1
BBBBB	Trend curve 2 (if defined).

The oldest value appears on the first line in the file, the most recent on the last line. The format .SKV is directly importable in Microsoft Excel. Within Excel you can use the diagram manager to create statistical diagrams.

Databuffer file

The file is named DATABUFF.SKV. It contains the current content of the databuffer area, starting from index 0 and upwards. The file contains 10 values on each line. The values can be changed in, for example, Microsoft Excel or a text editor and then you can save the file on the drive.

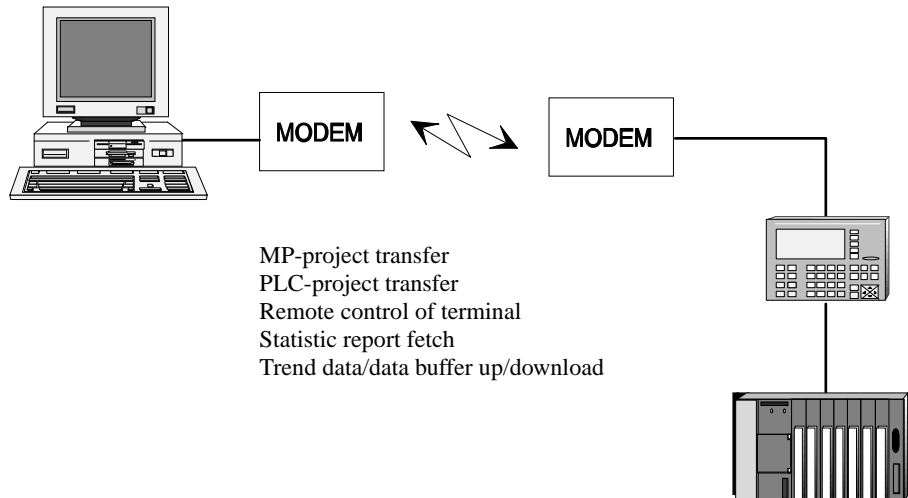
Example

The application consists of two trend objects called 'TEMP' and 'PRESSURE'. A databuffer is also used. The device driver has been loaded as drive letter 'E'. By using the DOS command 'DIR E:' the following will appear on the screen.

```
Volume in drive E is MAC/MTA
Directory of E:
TEMP SKV          1024   95-01-03   14.12
PRESSURE SKV     8065   95-01-03   12.18
DATABUFF SKV     420    95-01-02   13.55
                3 file(s) 9509 bytes
```

5.6 Remote control of the terminal via modem

Graphics terminal



The function remote control is used to transfer project programmed in the PC software to and from unmanned stations. The units communicate via a serial cable or via modem (for longer distance).

By sending a control string to the terminal's transparent port, you can force the terminal to enter Terminal mode. When in Terminal mode you can enter commands using a standard terminal program.

The Mode switch on the terminal must be in PROG position. Changing between PROG and RUN should be performed with the button combination [ALARMACK]+[ALARMLIST].

This function can be used in conjunction with the multidrop mode and the file transfer mode.

Control string

The string contains Start character, Identity character and Check Sum. The string can be sent from a standard terminal program, for example by using a macro.

Start character	01 Hex
Identity character	Station number + 80 Hex (bit seven set)
Check sum	Start character + Identity character

The selected unit will display its help screen when connection is established.

Example

To select station 4 send the control string (hexadecimal): 01 84 85

If multidrop is not enabled, use 00 as station number.

Terminal commands

When connection is established the help screen will be displayed and the terminal is ready to accept commands. The following commands are available:

Command	Description
PROG	Forces the terminal to enter Programming mode.
RUN	Forces the terminal to enter the Run-time mode.
SETUP	Forces the terminal to enter the Setup mode.
VIEW	Shows the content of the terminal's text display. Graphblocks cannot be displayed.
TRANS	Forces the terminal to enter the Transfer mode. To do this, the terminal must be in Programming mode.
MODE	Queries the terminal for current mode, terminal type and version number.
BLOCK n	Force the terminal to display block number n in Run-time mode.
LOGOUT	Causes the terminal to leave Terminal mode.
KEY n	Enter terminal key. Key can be: any ASCII-character, LEAVE, NEXT, ENTER, UP, DOWN, LEFT, RIGHT, ALARM LIST, ALARM ACK, INFO, INC, DEC, MAIN, PGUP, PGDN, CLEAR.
HELP	Shows help screen.

6 Reference instructions

This chapter firstly describes the general editing functions then allowable signal formats etc and finally each function. The description of functions is brief and assumes that the user is well acquainted with the use and principles of the terminal.

6.1 Basic editing functions

There are certain basic rules when programming the terminal. The keyboard is compact and each key has several functions.

Key functions

The arrow keys are used to move the cursor in a menu or dialog.

The [INFO] key is used to switch alternatives for an entry field or to copy previous settings. Furthermore, it is used to show choice lists.

The [ENTER] key is used to confirm settings and proceed to the next line.

The [NEXT] key is used to proceed to the next level. Settings entered are saved.

The [LEAVE] key returns you to the previous level.

The # symbol in the text terminal and the [MAIN MENU] key in the graphics terminal are used to show choice lists for dynamic objects.

Any special functions for the keys are described under the respective function.

Switch between programming mode and run-time mode

When the mode switch is in the PROG position you can enter the run-time mode by pressing [ALARM ACK] + [ALARM LIST]. Return to the programming mode by pressing [ALARM ACK] + [ALARM LIST] again.

Editing graphic objects

Graphics terminal

The arrow keys are used to alter size or move objects in 8 pixel steps when editing graphic objects. Sizing/moving occurs pixelwise by holding [F12] pressed at the same time as pressing one of the arrow keys.

You can go through all the objects stepwise in the editing mode by pressing the [ALARM LIST] key. The marked object is shown with one or several * and parts of its definition are shown in a square on the right of the screen. You can press [ENTER] when an object is marked to edit the object's parameters, [F10][CLEAR] to erase the object or [LEAVE] to exit from the editing mode. Mark an object, press [ENTER] and then [NEXT] if you want to move or alter the size of the object.

You switch between MOVE and SIZE, that is between moving and altering the size of a graphic object with the [INFO] key.

6.2 Signal format

Digital signals

The terminals can cope with digital signals of the following types:

Signal	Description	A series	FX series
X	Input	•	•
Y	Output	•	•
M	Memory cell	•	•
B	Link memory cell (MELSEC NET)	•	
F	Fault memory cell	•	
S	Sequence memory cell		•
T	Timer	•	•
C	Counter	•	•

Analog signals

The terminals can cope with analog signals of the following types:

Signal	Description	A series	FX series
D	Data register	•	•
W	Link register (MELSEC NET)	•	
R	File register	•	
T	Timer	•	•
C	Counter	•	•

Suffix

You can present double registers, decimal numbers and/or just positive values by providing an analog signal with a suffix.

Note!

Decimal values or scaled values are stored internally as floating point numbers with seven digit accuracy. To avoid rounding errors for several figures accuracy you must use numbers without decimals or scaling. These are stored internally as integers.

Code	Format type	Range
Dn	Simple whole number	-32768 – +32767
Dn+	Simple (positive only)	0 – +65535
DnL	Double whole number	-214783648 – +214783647
Dn+L	Double (positive only)	0 – +4294967295
DnRB	BCD floating number	-9999.9999 – 9999.9999
DnRF	Floating number	±3.4E38 Numbers larger than 1,000,000 are shown with exponent. The Decimals parameter has no function for this type of format.
DnIn	Indexing	See section on Index addressing.

In the above table n is the number on the register, for examples n=100 corresponds to D100. If D100L+ is given this means that the D100 and D101 registers are read as positive whole numbers.

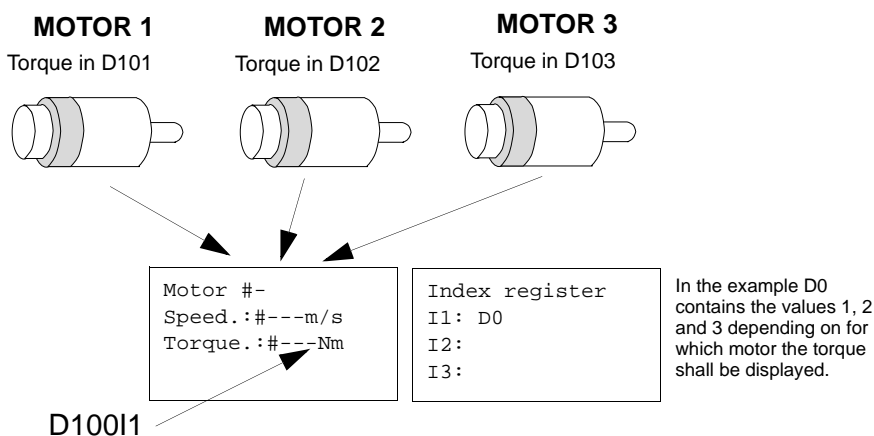
Either the preset value or the current value can be displayed for timers and counters, the following possibilities are available:

Symbol	Display
T50	T50 current value
T50L	T50 and T51 current value
T50PRE	T50 preset value (F1/F2 only)
T50LF	T50 and T51 preset value (F1/F2 only)

Index addressing

Index addressing makes it possible to use the same block; for example, for the control of several motors. In the example below the torques are linked to the D100I1 register. The torque is shown for one of the three motors depending on the value in the index register, in this case the value in D0. The value in D0 decides for which motor the torque is shown. The content of the index register is added to the address and the result forms the address which is shown. Generally it can be said that:

$$\text{ShownValue} = \text{content}(\text{DataRegisterAddress} + \text{content}[\text{Indexdataregister}])$$



The index register is given as a suffix to the address. It is still possible to add other suffixes after the index definition. For example, D100I1L is given for the double register. The index register will not be counted twice if the suffix L is given. I1 to I8 are linked to the data register in the **Setup** menu if the PC software is used or in the Setup mode via the terminal.

Scaling

The Offset and Gain parameters for an object are used to rescale the register value to a displayed value according to the following equation:

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

As the object is altered from the terminal in the run-time mode, the displayed value is rescaled according to the following equation:

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

Scaling does not affect defined max or min values or the number of decimal places.

Reserved characters

The following ASCII codes are reserved for the terminal's internal functions and may **not** be used.

ASCII code:

Hex	Dec	Character	ASCII code
01	01		SOH
02	02		STX
03	03		ETX
04	04		EOT
05	05		ENQ
06	06		ACK
07	07		BEL
15	21	§	
23	35	# (textblock only)	

6.3 Setup mode

This chapter describes the Setup mode functions in alphabetical order. The global parameters for the terminal are set in the Setup mode. The Setup mode is selected by setting the mode switch to PROG and pressing any key when the start-up screen is displayed.

Alarm settings

Description General settings for the alarm list. In the graphics terminal the alarm list are stored in the project memory. Each line occupies eight bytes. Maximum 64 kbytes can be used for the alarm list. Inactive and acknowledged alarms can be deleted from the alarm list with the key [CLEAR].

Note!

In the terminal we differentiate between the *alarm list* and the *alarm definitions*. The alarm list contains alarms that occur during run-time. Under alarm definitions (see Alarm in programming mode) you define the signals which will give an alarm as well as the alarm message for respective signals.

Parameters **Active sign:** States the digital signal which the terminal will set when an active alarm occurs.

Unack sign: States the digital signal which the terminal will set when there is an unacknowledged alarm.

Alm symbol: States when ■ALARM■ will be displayed on the screen.

Choice	■ALARM■ is shown when	
	active	unacknowledge
NO	-	-
UNACK	-	X
ACTIVE	X	-
BOTH	X	X

List size: The number of alarms in the alarm list (both active, not active, acknowledged and unacknowledged). The default setting is 32.

List erase signal: (Graphics terminal) States the digital signal which when set will erase the whole alarm list.

Character set

Description Choice of character set. Eight key codes (C1-C8) have been reserved for national characters.

Character set	MAC 50/ML MTA-250/ML	MAC 50/MV MTA-250/MV	MAC 90 MTA-G1
Swedish	X	X	X
German	X	X	X
French	X	X	X
Spanish	X	X	X
Italian	X	X	X
Nor/Dan	X	X	X
Katakana	X	X	-
Cyrillic	-	X	-

Comments Character editing to set up Katakana and Cyrillic takes place in the terminal. This means that the PC software can only be used for making back-up copies. You can switch between the functions for the alphabetical keys (A-Z, a-z) between ASCII and KATAKANA/CYRILLIC by holding the [F2] key pressed at the same time as pressing [↑] or [↓]. Refer to the Overview to see which character relates to which key in the various languages.

Note!

The different terminal models do not have compatible character sets for Katakana and Cyrillic.

Contrast settings

Graphics terminal

Description Setting the screencontrast

Parameters **Inc:** Makes the display brighter.

Dec: Makes the display darker.

Enter: Closes the function.

Databuffer setting

Graphics terminal

Description Define the memory area for data storage.

Parameter **Ctrl.blk.reg:** Gives the first register in the control block.

Buff.size : Gives the size of the data buffer in the terminal as number of registers. Each register takes up two bytes of the project memory and is backed up by battery.

Comments For further information see section *Data buffer*.

Date/Time format

Description Setting date and time format

Parameters **Date Form:** YY-MM-DD

Y=Year

M=Month

D=Day

Time Form: HH:MM:SS

H=Hour

M=Minutes

S=Seconds

Clock used: The terminal uses the clock built in the terminal. PLC uses the clock in the PLC system.

Clock →PLC: States if the clock in the terminal shall be transferred to the register in the PLC system.

PLC reg: States start address for storing date/time. Only used if Clock → PLC is set to YES.

A series

D9025 is default for the A series.

Data register	8 highest bits	8 lowest bits
D9025	year	month
D9026	day	hour
D9027	minute	second
D9028	null set	day of the week

FX and FX0N series

D8013 is default for the FX and FX0N series.

Data register	Presentation
D8013	second
D8014	minute
D8015	hour
D8016	day
D8017	month
D8018	year
D8019	day of the week

The terminal also functions together with FX-RTC.

F1/F2 series

D700 is default for the F1/F2 series.

Data register	Presentation
D700	second
D701	minute
D702	hour
D703	day
D704	month
D705	year
D706	day of the week

Update int: Updating interval in seconds.

Disk settings

- Description** Settings of the drive when using the device drive MACMTA.SYS.
- Parameters**
- Disk size (kB):** The size of the disk in kBytes. Default is 128 kBytes.
 - File extension:** States the extension of the file. Default is .SKV.
 - Field sep.code:** States a hexadecimal code for field separation. Default is 59 (tab).

Erase Memory

- Description** Erasing the terminal's application memory. That is all blocks and definitions for alarm, time channels, function keys and system signals are erased.
- Parameter**
- [ENTER] Memory is erased.
 - [LEAVE] Return to previous level without erasing the memory.
- Comments** The Setup menu is automatically displayed when erasing is complete.

Note!

All data stored in the terminal is lost on Erase Memory. The language choice, character set, PLC choice and the command lines parameters are not affected. Other parameters are erased or set to the values for default settings. Historical trend data, battery backed alarms and databuffer content are erased.

Index register

- Description** Index addressing of dynamic objects.
- Parameters** **I1, I2, I3:** Data registers which will be defined as internal index registers.
- Comments** For more details see section *Suffix*.

Language

- Description** Choice of menu language; British English, German, Swe-

dish, RAM or American English.

Comments The RAM alternative is used for language choice which is located in the RAM backed-up by battery. This text is downloaded to the terminal from a PC. (Examples of languages can be Italian, Spanish, Dutch etc). Access to the PC software package is required to do this. The default setting is English menu texts.

Online settings

Description Choice of which changes are possible in run-time.

Parameters **Time channels:** Possibility to change which times the signals shall be ON. See section Time channels in Run-time mode.

Function keys: Possibility to change the signal related to a function key. See section *Run-time mode*.

Object: (Textblock only) Possibility to change maneuverable objects' definition in run-time. See section *Run-time mode*.

LED: (graphics terminal only) Possibility to change a signal related to a light diode. See section *Run-time mode*.

PLC selection

- Description** Choose the PLC to which the terminal is to be connected. The project in the terminal is not erased when you change PLC system. I/O which are not permitted in the chosen PLC system remain.
- Parameters** Choose the required PLC system. **FX, A-CPU, C24 protocol 1** and **F1/F2** which only applies to the text terminal.
- Comments** C24 protocol 1 should be selected if the terminal is connected to any of the A series' ports AJ71C24-S(x), A1SJ71C24-R4 or RS-422 ports on the A2CCPUC24. The parameters are set under Port parameters in the main menu in Setup mode.

Port parameters

- Description** Settings for the PLC system, printer, PC software and network communication.
- Parameters** **PLC:** Setting for PLC communication. These are changed automatically when you make a new PLC selection. (The parameters below are for FX).
 Port: RS-422
 Baud rate: 9600
 Data bits: 7
 Stop bits: 1
 Parity: EVEN
 Station: 0. Same as on the communication module or 0 if CPU.
- Prt/Transp/NoProt:** Setting for communication in the transparent mode, no protocol mode and with a printer
 Port: RS-232
 Baud rate: 9600
 Data bits: 8
 Stop bits: 1
 Parity: NONE
 Station: 0

MP: Setting for communication with the PLC software

Port: RS-232

Baud rate: 9600

Data bits: 8

Stop bits: 1

Parity: NONE

Network: Described in the *Communication* chapter

Comments For more information see the chapters *Communication* and *Install terminals*.

Printer setting

Description Setting of page length and characters for line ends when printing block.

Parameters **Newline char:** States line end characters; CR/LF, CR, LF or none.

Page length: Number of lines to be printed before form feed. No form feed occurs if page length is set to 0.

Handshake: States if handshaking between printer and terminal takes place with XON/XOFF or CTS/RTS.

Graphic size: State the size of the printout.

Single = 240x128 pixels

Double = 480x256 pixels

Quadruple = 960x512 pixels

Graphic orient: (Graphics terminal only) States if the printout should be portrait or landscape.

System signals

Description Setting of the handshaking signal between the terminal and the PLC system.

Parameters **Cur.disp. reg:** Data register which includes the number of the block displayed on the screen in run-time mode. The data register is automatically updated by the terminal on block change. This register does not affect choice of Text-block.

New disp. reg: The PLC data register that controls which Textblock is shown on the screen. Take for example D100. If block 34 is to be shown on the screen, the value 34 is entered in the D100 from the PLC program or the terminal.

Prt busy sign: The address of a digital signal which the terminal sets when printing.

Buzzer signal: Digital signal which when set activates the terminal's built-in buzzer.

VF-bright reg: (Text terminal only) Data register which controls the brightness of the terminal's screen. Only applies to VF screens. The functions Display switched off and Screen saver are available for text terminal/ML.

Register content	Brightness
0	Brightness set during terminal setting.
1	25%
2	50%
3	75%
4-FFFD	100%
FFFE	Display switched off.
FFFF	Screen saver.

Screen saver means that the display switches off if no key is pressed for five minutes. It switches on if a key is pressed, a new block is shown or an alarm activates.

Backlight signal: (Graphics terminal only) A digital signal which controls if the display is on or off.

Reverse signal: (Graphics terminal only) A digital signal which controls if the display has black text on a light background (normal) or light text on a black background.

No protocol ctrl: The first control register which is used in the No protocol mode. The No protocol mode is described in the *Communication* chapter.

Cursor ctrl blk: (Graphics terminal) The start register to a control block which write the current cursor position to data registers in the PLC system.

Register	Description
Dn0	Current cursor position X (in pixels), 0-239.
Dn1	Current cursor position Y (in pixels), 0-127.
Dn2	Status register 0 Normal 1 The user tried to move down, but no object below. 2 The user tried to move up, but no object above. 3 The user tried to move to the left, but no object to the left. 4 The user tried to move to the right, but no object to the right.

The status register can be used for example if a recipe form is created, where scrolling possibility is wanted. The PLC program can check when the status register is set to 1, the scrolling recipe data downwards using index register.

Command: The following can be stated in the command line:

RTSn The RS-232 port's RTS is connected to the memory cell, e.g RTSM100.

Tx Time out in x ms.

SW Converts text with Swedish ASCII (7 bits) characters to 8 bits IBM PC extended ASCII .

Rx Maximum number of resends. x=number of attempts.

BR Batch Read - communication with xxxC24, FX and A.

BCx Background characters in bars in textblock are stated as ASCII characters.

EK Permitted external function keys (IFC 128).

NPn Digital signal to switch between No protocol and Transparent/printer mode, e.g NPM100. Used to switch between No protocol mode and Transparent/printer mode in run-time, for example, to call up a computer and transmit a message.

NTx Time out in ms. For messages in No protocol mode.

RC Turns off history on communication alarms.

BWO Turns off the battery low warning.

Terminal options

Description Setting up a terminal

Parameters **Key Beep:** States if a terminal shall beep if a key is pressed. Note that the shift keys do not have a beep sound even if the parameter is set to YES.

Key Repeat: States if a function will be repeated as long as the key is pressed.

VF Bright(%): (Text terminal only) Brightness on the terminal with VF screens is adjustable. The setting is given in percent. Permitted levels are 25, 50, 75 and 100 percent. Setting can also take place via the PLC register, see *System signals*. It is advisable to select 50% brightness for the text terminal to increase working life.

VF speed: (Text terminal only) The flashing rate of the cursor is adjustable. With VF Speed: state the frequency in the range 0-255. Default value is 20.

Display color: (Graphics terminal only) States if the display shall have black text on a light background or light text on a black background. Default setting is NORMAL which gives black text on a light background.

Screensave time: (Graphics terminal only) The time in minutes after the last entry on the screen when the display will switch off. The default setting is 10 which means that the display switches off after 10 minutes. We recommend that you use this function to optimize the lifetime of the display (defined as 10.000 h).

6.4 Programming mode

This chapter describes all functions and menu choices in the programming mode. The terminal is programmed in the programming mode with the built-in keyboard and projects are transferred between personal computers and the terminal. The programming mode is selected automatically when the terminal is started with the mode switch in PROG.

The upper level in the programming mode contains three functions in the graphics terminal, Edit, Block list and Transfer and five functions in the text terminal, Edit, Block list, Transfer, Memory cassette and Documentation.

The following functions are found under Edit: Block, Alarm, Time channels, LED and Function keys.

6.4.1 Block header

A block header must be defined for each block.

Parameters **Block number:** States the block number. The defined values will be automatically filled in if a block with allocated number already exists.

The block which will be the main menu in the application, that is the one to be shown on the screen when the program starts, must always have number 0.

Note!

An application project must always have a main menu.

Graph/Text: (Graphics terminal only) States if the block will be a graphblock or textblock. It is not possible to change type of a defined block.

Display signal: The digital signal which when activated results in the textblock being shown on the screen. Display signals in consecutive order should be used to obtain block changes as rapidly as possible. This field is not filled in if another block change method is used.

Printer signal: The signal which when activated results in the block being printed out. The Display and Printer signals can be the same. Printer signals should be stated in consecutive order to print out as rapidly as possible.

Prt Block Type: (Textblock only) States line length when printing. There are two alternatives 20/30 characters and the number to EOL, carriage return.

Compleat. sign.: (Textblock only) The digital signal which is affected when the printing of textblock is finished. Normally the signal is activated, enter an R after the signal and it is reset, e.g. M100R when the printout is finished.

Status word: States the display's appearance in run-time mode. Press [INFO] to open the status word menu. Status word has no function in the system blocks.

Cursor off: (Textblock only) states if the cursor can be seen in the block in run-time mode.

1st man obj: (Textblock only) states if the cursor shall be on the 1st maneuverable object in the textblock instead of in the upper left corner.

Disable MAIN: states if the [MAIN MENU] key shall be blocked in run-time mode when the block is visible on the screen.

Disable ALARM: states if the [ALARM LIST] key shall be blocked in run-time mode when the block is visible on the screen.

MORE-indicat.: (Textblock only) states if the + character shall be seen at the bottom right and the upper right corners of the screen when there is more of the block above and below respectively than that which can be seen on the screen.

Auto Data Ent.: States if the cursor shall move automatically to the next maneuverable object after an entry. The cursor can only be placed on maneuverable objects at this stage.

Password: A password can be allocated to a block here. An entry field is shown where the operator enters the password when a block defined with a password is to be shown on the screen. The password can be any combination of max six numbers and/or letters.

Function keys: Definition of local (block specific) functions for the function keys F1-F5 for the graphics terminal and F1-F6 for the text terminal. For more information on local and global function keys see the Function key section.

Note!

Local function keys cannot be defined for system blocks (blocks 990-999) in text terminals.

6.4.2 Textblock

The following section describes objects for textblock in alphabetical order.

Analog

Description Object that represent integral or floating numbers.

Parameters **A.Signal:** The signal's address. See *Basic editing functions*.

Positions: The number of positions used by the object. Characters such as minus signs or decimal points must also be counted.

Decimals: The number of decimal places which represent the object.

Maneuver: States if the object can be altered in run-time. Only the current value can be altered for A and FX series timers and counters. Only the preset value can be altered for F1/F2 series timers and counters.

Min val: The minimum value (-32768) you can enter for an object. If the value has more than seven positions scrolling is made horisontally.

Max val: The maximum value (32767) you can enter for an object. If the value has more than seven positions scrolling is made horisontally

Offset: and **Gain:** Used to change scale of the register value. See the *Basic editing functions* section.

Bar graph

Description Object that represent integral or floating point numbers as bar graphs. The bars are shown horizontally so that the maximum number of positions are marked when the value has reached its maximum level.

Parameters **A.Signal:** The signal's address. See *Basic editing functions* section.

Positions: The number of positions which the signal uses. The maximum number of positions is 80 for the text terminal and 30 for the graphics terminal.

Direction: States if the bar shall be presented to the right or the left.

Min val: The minimum value an object can have.

Max val: The maximum value an object can have.

Offset: and **Gain:** Used to change scale of the register value. See the *Signal format* section.

Date/Time

Description Object for setting date and time. Date and time objects take data from the terminal or CPU's real-time clocks.

Parameters **Day of week:** States if the day of the week will be shown. Requires two positions.

Date: States if the date will be shown. Uses the format given under Date/Time format in the Setup.

Time: States if the time will be shown. Uses the format given under Date/Time format in the Setup.

Maneuver: States if the object can be altered in run-time mode. The current date and time are altered by entering the new value and pressing [ENTER] in the run-time mode.

Note!

A maneuverable Date/Time object must be defined to set the clock.

Digital

Description Objects that can have two states, off or on. The maximum length of the text is seven characters.

Parameters **D.Signal:** The signal's address.

Text 0: The text shown when the signal is off.

Text 1: The text shown when the signal is on.

Maneuver: States if the object can be forcibly set in the run-time mode from the terminal.

Jump

Description Object that make it possible to build up a menu tree.

Parameters **Cur.block:** States the number of the block which is exited.

Jump to: States the number of the block to which the jump takes place.

Note!

If a jump is generated during run-time to a block that does not exist the system leaves an error message and restarts after acknowledge.

Comments Jumps to previous textblocks can be made (nine levels back) by linking the RET function to a function key. See the section *Function keys*.

Multiple choice

Description Object that can have several states. The object is linked to three digital signals which together can have eight different states. Text up to seven characters can be linked to each state.

Parameters **D.Obj.1-D.Obj.3:** The signals' addresses. Three signals are not necessary be given. If two are given four different states are shown. Combination 100 means that D.Obj.1 is set, 010 means that D.Obj.2 is set and 001 means that D.Obj.3 is set.

Text 000-Text 111: Those texts which are to be shown for each state of the object are given below.

State	D.Obj 1	D.Obj 2	D.Obj 3
Text 000	0	0	0
Text 001	0	0	1
Text 010	0	1	0
Text 011	0	1	1
Text 100	1	0	0
Text 101	1	0	1
Text 110	1	1	0
Text 111	1	1	1

Man 000-Man 111: States if it is possible to forcibly set to this status in run-time mode.

Text

Description Object for handling ASCII strings. The possibility exists to present texts stored in the CPU's data register in a text object. The texts should be in ASCII format. The texts can be changed in run-time from the terminal by opening and closing the entry field with [ENTER]. While the field is open for writing, [F1], [F2], [F4] and [F5] for the text terminal and [F10], [F11], [F12] and [F13] for the graphics terminal function as shift keys.

Parameters **A.Signal:** States the register where text for the first position is stored.

Positions: The number of positions the text will occupy on the screen. Each register contains two characters (occupies two positions) for A and FX series. For F1/F2 series each register contains one character. The default setting is eight positions which corresponds to four registers in A and FX series and eight registers in F1/F2 series. The maximum number of positions is 80 for the text terminal and 30 for the graphics terminal.

Maneuver: States if the object can be altered in the run-time mode.

Comments The text is converted from Swedish 7 bit ASCII to 8 bit IBM extended ASCII if SW is stated in the command line in the System signals.

6.4.3 Graphblocks

The following section describes objects for graphblocks in alphabetical order and only applies to the graphics terminal.

Arc

Description Static objects used to create background graphics. If you are using the PC software the following parameters can be set:

Start angle: State the start angle for the arc. The value is set between 0 and 360 degrees.

End angle: State the end angle for the arc. The value is set between 0 and 360 degrees. The end angle must be larger than the start angle.

ASCII

Description Object for handling ASCII strings in graphblocks. It is possible in ASCII objects to present text stored in the CPU's data register. The texts must be in ASCII format. While the field is open for writing in run-time, [F10], [F11], [F12] and [F13] function as shift keys.

Parameters **A.Signal:** The register where the text for the first position is stored is stated here.

Position: The number of positions which the text will occupy on the screen. Each register contains two characters (occupies two positions).

Adjust: States if the text shall aligned with left margin or centred.

Maneuver: States if the object can be forcibly set in the run-time mode from the terminal.

Frame: States if there will be a frame around the object or not.

Comment The text can be changed in Run time mode by opening and closing the entry field with the [ENTER] key. When the field is open for entering the keys [F10], [F11], [F12], and [F13] are used as shift keys.

Alarm banner

- Description** Object for displaying one line in the alarm list.
- Parameters**
- Alarm list line no:** The line in the alarm list to get information from (1=first line, 2=second line etc).
 - Positions:** The number of characters to be shown.
 - Show symbol:** States if the symbol should be shown in the string.
 - Show day of week:** States if day of week should be shown in the string.
 - Show date:** States if date should be shown in the string.
 - Show time:** States if time should be shown in the string.
 - Ack possible:** States if it should be possible to acknowledge the alarm.
- Comments** By using several alarm banner objects it is possible to create a complete graphical alarm page, replacing the default text based alarm handler.

Bar

Description Object that represent integral or floating point numbers as bar graphs.

Parameters **A.Signal:** The signal's address. See *Basic editing functions* section.

Min: The minimum value the object can have.

Max: The maximum value the object can have.

Direction: States if the bar shall be presented to the right, left, up or down.

Scale: States if the scale is to be shown or not.

Box: States if a box is to be drawn around the bars.

Indicators: States if the maximum and/or minimum values for the signal will be marked on the axis.

Scale Div: States how the scale is to be divided.

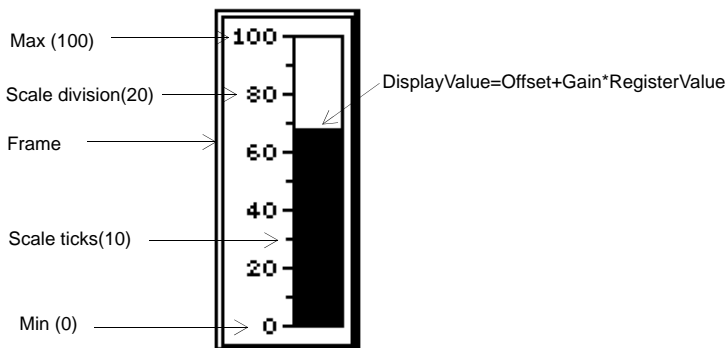
Scale Tics: States the intervals between the scale divisions shown.

Fill pattern: States if the filling will be solid or dashed.

Offset: and **Gain:** Used to change scale of the register value. See Signal format.

Frame: States if a frame shall be drawn around the object.

Comments The indicator is reset when the terminal is switched on. Re-setting can also be done in run-time mode by moving the cursor to the object and pressing [ENTER].



Diagram

Description Object used to draw a diagram based on the content of PLC registers.

Parameters **Reg. pair count:** Number of register pairs.

Y signal: First Y coordinate register.

Y Min: The minimum value a Y coordinate can have.

Y Max: The maximum value a Y coordinate can have.

Y Scale Div: States how the scale is to be divided.

Y Scale Tics: States the intervals between the scale divisions shown.

X signal: First X coordinate register.

X Min: The minimum value a X coordinate can have.

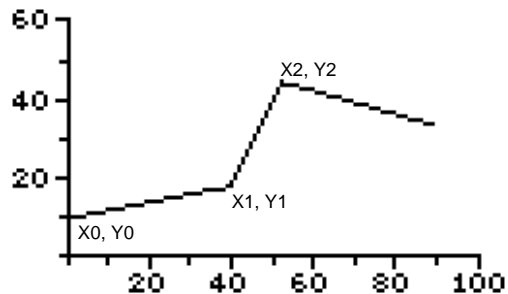
X Max: The maximum value a X coordinate can have.

X Scale Div: States how the scale is to be divided.

X Scale Tics: States the intervals between the scale divisions shown.

Type: States type of diagram, Line or Bar.

Comments It shows two data register banks related to each other. It is a real time function. The object is normally used for not time-based presentation. Time based trending with sample time <1 second can be shown if the PLC system handles the logging of data.



Digital symbol

Description Digital object used to switch between two symbols depending on the state of a digital signal.

Parameters **D.Signal:** The signal's address.

Symbol 0: The symbol shown when the signal is off.

Symbol 1: The symbol shown when the signal is on.

Maneuver: States if the object can be forcibly set in the run-time mode from the terminal.

Digital text

Description Text object used for switching between two texts depending on the state of a digital signal. The maximum length of the text is seven characters.

Parameters **D.Signal:** The signal's address.

Text 0: The text shown when the signal is off.

Text 1: The text shown when the signal is on.

Adjust: States if the entry field shall aligned with left margin or centered.

Maneuver: States if the object can be forcibly set in the run-time mode from the terminal.

Frame: States if a frame will be drawn around the object.

Ellipse

Description Static object used to create background graphics.

Parameters **Transparent:** States if the ellipse will be transparent.

Filled: States if the ellipse will be filled.

Filler

Description Object used to fill in enclosed areas

Parameter **D.Signal:** The signal's address

Pattern: States if the filling will be solid or checkered.

Maneuver: States if the object can be forcibly set in the run-time mode from the terminal.

Note!

Filling very irregular areas in run-time can cause a system error. Filling can, in certaincases, make the screen slow.

Jump

Description Object that make it possible to build up a menu tree in the application.

Parameters **Jump to:** States the number of the block to which the jump shall take place.

Position: The number of positions which the text will occupy.

Text: Optional text for the object. The maximum length of the text is 18 characters.

Adjust: States if the text shall aligned with left margin or centered.

Frame: States if there will be a frame around the object.

Note!

If a jump is generated during run-time to a block that does not exist the system shows an error message with the possibility to restart.

Comments Jumps to previous blocks can be made (nine levels back) by linking the RET function to a function key. See the section *Function keys*.

Lines

Description Static objects used to create background graphics.

Line pattern

Description Function for setting line thickness and if the line will be solid or dashed for the objects line, rectangle, ellipse and arc.

Parameters **Thickness:** States the thickness of the line in number of pixels.

Pattern: States if the line will be solid or dashed.

Comments Dashed lines can only be shown with 1 pixel thickness.

Multiple choice

Description Object that can have several states. The object is linked to a register which can have eight different states. Text of seven characters can be linked to each state.

Parameters **A.Signal:** States the register that determines which text shall be shown.

Text 000-Text 111: Those texts which are to be shown for each state of the object.

Man 000-Man 111: States if it is possible to forcibly set to this status in run-time mode.

Adjst: States if the text shall aligned with left margin or centered.

Frame: States if there will be a frame around the object.

Multisymbol

Description Objects that can show up to eight different symbols depending on the value in a data register.

Parameters **A.Signal:** States the register that determines which symbol will be shown. If the register has the value 1, the symbol 1 is shown and so on.

Move Obj: States the register which contains the X-Y coordinates for the object. The eight lowest bits determine the X coordinate, permissible values are 0-239. The eight highest bits determine the Y coordinate, permissible values are 0-127.

$$\text{RegisterValue} = Y * 256 + X$$

Symbol 0-7: Chooses the symbol to be shown.

Comments The Move Obj. register need only be stated if you wish to move the object around on the screen with the help of the PLC program.

Numeric

Description Object for entering and showing analog values. Used for example to create entry fields.

Parameters **A.Signal:** The signal's address. See *Basic editing functions* section.

Positions: The number of positions with which the entered value will be stated.

Decimals: The number of decimal places with which the entered value should be stated.

Zerofill: States if 0 will printed in the empty positions.

Adjust: States if the entry field will be aligned with left margin or centered.

Maneuver: States if the object can be altered in run-time mode.

Min val: The minimum value which can be stated.

Max val: The maximum value which can be stated.

Offset: and **Gain:** Used to change scale of the register value. See *Signal format*.

Frame: States if there will be a frame around the object or not.

Rectangle

Description Static object used to create background graphics.

Parameters **Transparent:** A transparent rectangle is drawn.

Filled: A filled rectangle is drawn.

S-meter

Description S-meter is a completed object for a speed indicator.

Parameters **A.Signal:** The signal's address. See *Basic editing functions* section.

Min: The minimum value the object can have.

Max: The maximum value the object can have.

Scale Div: States how the scale is to be divided.

Scale Tics: States the intervals between the scale divisions shown.

Angle: States the angle, working area, for the object. 10-360 degrees.

Offset: and **Gain:** Used to change scale of the register value. See *Signal format*.

Frame: States if there will be a frame around the object or not.

Slide

Description Objects which make it possible to increase and decrease the value with the keys [INC] and [DEC].

Parameters **A.Signal:** The signal's address. See *Basic editing functions* section.

Note!

The suffixes RB and RF do not function for slide objects.

Min: The minimum value the object can have.

Max: The maximum value the object can have.

Direction: States in which direction the presentation will take place, upwards or to the right.

Maneuver: States if the object can be altered in the run-time mode.

Scale div: States how the scale is to be divided.

Scale Tics: States the intervals between the scale divisions shown.

Offset: and **Gain:** Used to change scale of the register value. See *Signal format*.

Frame: States if there will be a frame around the object or not.

Symbol (static)

Description Static symbol used to create background graphics.

Symbol maintenance

Description Function for handling symbols.

Parameters **Directory:** Shows a list of symbols in the directory.

Save symbol: Function for saving your own symbol in the directory. Frame the required symbol and press [LEAVE]. Allocate any name and press [LEAVE]. The symbol is now stored in the directory.

Delete symbol: Function for erasing symbols from the directory. Select a symbol you want to erase and press [NEXT].

Text (static)

Description Static text which is written in the graphblocks. Six different sizes are available.

Time

Description There are two types of time object, digital clock and analog clock.

Parameters **Analog clock:**

Seconds: States if the analog clock will have a second hand.

Frame: States if the analog clock shall be shown with a frame.

Digital clock:

Day of week: States if the day of the week is shown.

Date: States if the date is shown.

Time: States if the time is shown.

Maneuver: States if the clock can be set in run-time mode.

Frame: States if a frame is drawn around the object.

Trend

Description Object to present collected values from analog signals.

Parameters **A.Signal 1-2:** States the analog signals for which the object shall collect and present values. Only 16-bit numbers can be used.

Y-min: States the minimum value on the Y axis (-32768).

Y-max: States the maximum value on the Y axis (32767).

Y-Value frm: States the format of the curve.

Sample int: States the time interval between collection of data. Minimum value is 1 second.

Stored samples: States how many values shall be stored.

X-Range: States the time range shown in the trend diagram.

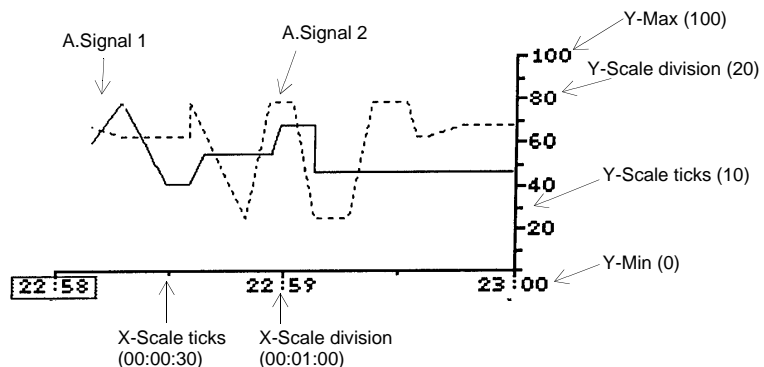
Y-Scale: States which side the Y axis is placed, left, right, both or off.

Y-Scale Div: States how the Y axis is to be divided.

Y-Scale Tics: States the intervals between the scale divisions shown.

X-Scale Div: States how the X axis is to be divided.

X-Scale Tics: States the intervals between the scale divisions shown.



VU meter

Description Object to create a graphic VU meter on the screen.

Parameters **A.Signal:** The analog signal to be presented by the VU meter. You can present double registers, floating numbers and/or just positive values by giving the analog signal a suffix. See *Signal format* section.

Min: States the minimum value on the VU meter scale.

Max: States the maximum value on the VU meter scale.

Scale Div: States how the scale is to be divided.

Scale Tics: States the intervals between the scale divisions shown.

Value form: States the presentation format.

Offset: and **Gain:** Used to change scale of the register value. See the section *Signal format*.

Frame: States if a frame shall be drawn around the object.

6.4.4 Function keys

The function keys are linked to a digital signal or function by entering its address after respective keys. If you press a function key in run-time mode the digital signal linked to the key is set. When the key is released the signal is reset.

There are two kinds of function key definitions; global and local. Global function keys are defined and can be used in the whole application. Local function keys are defined and used in a block. The global definitions are valid in all states during run-time mode, conditional on that the block shown on the display does not have local definitions, in which case these overrides the globals.

Both local and global function keys can be connected to page scrolling and return functions. This is done by defining **PGUP**, **PGDN** and **RET** instead of a signal address. **PGUP** and **PGDN** will scroll one page up or down when you press on the function key. **RET** takes you back to the previous displayed block. This works through nine levels. In the graphics terminal there are predefined keys for these functions.

F100-F227 are used together with the port interface IFC 128. To enable use of these the **EK** command must appear on the command line in system signals.

Text terminal

The text terminal has six function keys built in, F1-F6. If IFC 128 is used a further 128 function keys (F100-F227) can be used.

Graphics terminal

The graphics terminal has thirteen function keys built in (F1-F13) plus a further six (F14-F19) of which five are predefined, [ALARM ACK], [ALARM LIST], [INFO], [MAIN MENU], [RETURN]. If IFC 128 is used a further 128 function keys (F100-F227) can be used.

Block jump with function keys

Graphics terminal

This function makes it possible to use function keys for jump to blocks without having to use the display signal. In the function key editor, local or global, you enter **BLnnn** where nnn is desired block number.

6.4.5 LEDs

Graphics terminal

The graphics terminal has eight built in LEDs, L1-L8. The LEDs can be linked to a data register. The contents in the register controls the color of the light diode as follows.

Register value (Hex)	Register value (Dec)	Flashing frequency (Hz)	Color
00	0	-	None
01	1	-	Green
02	2	-	Red
03	3	-	Yellow
11	17	5	Green
12	18	5	Red
13	19	5	Yellow
21	33	2,5	Green
22	34	2,5	Red
23	35	2,5	Yellow
31	49	1,2	Green
32	50	1,2	Red
33	51	1,2	Yellow
41	65	0,6	Green
42	66	0,6	Red
43	67	0,6	Yellow

6.4.6 Alarm

Alarm editor is shown when alarm is selected. Alarm editor has two cursors, a row cursor, >, and a position cursor, _. The former is moved with the vertical arrow key and the latter with the horizontal.

An alarm definition can consist of a maximum 18 characters for the text terminal and 27 characters for the graphics terminal. The number of alarm definitions is only limited by the terminal's memory, the practical limit is around 500. To keep the alarm delay time as short as possible in the run-time mode, the signals should be in consecutive order, that is M0, M1, M2 etc.

Parameters **Signal:** The digital signal linked to the alarm.

Printer: States if the alarm message shall be printed out directly on changing alarm status.

Acknowledge: States if the alarm shall be acknowledged or not. YES means that the alarm must be acknowledged while NO means that the alarm functions as an "event alarm", that is to say a form of information. The basic setting is YES.

The alarm can have the following status.

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

History: States when the alarm shall be removed from the alarm list. YES means that the alarm remains in the list until the alarm list is full. NO means that the alarm is removed from the list when it is acknowledged and is no longer active. If the Acknowledge parameter is set to NO it is removed from the alarm list as soon as it is no longer active.

Ack Action: Digital signal that is affected when the alarm is acknowledged. Normally the signal is activated. If an R is entered after the signal it is reset. For example, M100R means that M100 is reset when the alarm is acknowledged.

6.4.7 Time channels

Time channels are a function for setting and resetting digital signals relative to the real time clock. The only thing that limits the number of time channels is the terminal's memory.

Parameters **Signal:** State here the digital signal which is set during the given time period. A maximum of four different time periods can be stated per time channel.

The time period is stated as follows:

xx start-stop

xx-yy start-stop

xx yy state the day of the week or the number of days which are of current interest. It can be one to seven days long. The days of the week are represented by one digit or two letters. Either the letters Mo, Tu etc or with numbers according to the table below.

1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

Start states the start time

Stop states the stop time

E.g. Mo-Fr 0900-1800

Note!

To set the clock you need to define a manoeuvrable date/time object.

6.4.8 Block list

The block list gives a count of (directory) the blocks used in the application. To erase a textblock, place the cursor on a block number and press [F1][CLEAR] on the text terminal and [F10][CLEAR] on the graphics terminal.

6.4.9 Documentation

Text terminal

Documentation of applications in the terminal on the printer.

Printing starts by pressing [ENTER]. Documentation contains a list of the textblocks in use with a description of their static and dynamic objects. Blocks are printed in number order. Then follows a list of alarm definitions, time channels and function tangents.

Remember that a printer must be connected to the terminal to be able to select **Document**. How a printer is connected is described in the section *Connecting a printer*.

6.4.10 Memory cassette

An external memory cassette, MMC-4, can be connected to a terminal. The memory cassette is connected to the expansion outlet on the terminal. The memory cassette can store applications in a 128 kbytes Flash memory. The unit can be configured so that the application is loaded automatically in the terminal when starting up.

Read from cassette means that the application is transferred from the memory cassette to the terminal. **Write without boot** means that the application is written on the memory cassette but the automatic loading function is not used. **Write with boot** means that the application is written on the memory cassette and the automatic loading function is used. The **Verify cassette** function compares the application in the terminal with the application in the memory cassette.

Note!

The memory cassette can not be used as an extended memory. It can only be used to store/load back up copies of the application. The application can not be larger than the memory capacity in the terminal, 28 kB for the text terminal and 110 kB for the graphics terminal.

6.4.11 Transfer

Transferring applications between the terminal and personal computers with the PC software for DOS.

The software package for programming and documentation of the terminal is available for IBM PC compatible computers.

6.5 Run-time mode

The status can be shown and the process controlled from the terminal in run-time mode. The mode is selected by setting the mode switch to RUN. The possibility also exists to switch to the run-time mode with the mode switch in PROG by pressing [ALARM ACK][ALARM LIST]. Block 0 is shown automatically on the screen when you go over to run-time mode.

An error message is shown on the screen if a communication error should occur between the terminal and the PLC system. The terminal starts automatically when the communication restarts and the communication error is automatically entered as an event alarm in the alarm list.

Form feed takes place automatically after 60 lines when printing a text-block if nothing else is stated in the Setup mode under printer settings. The printer must be IBM compatible and handshaking takes place with XON/XOFF or CTS/RTS protocol, selected in the Setup mode.

The various functions in run-time are described in this section. These do not correspond to any menu choice in run-time.

6.5.1 Textblocks

Block consist of lines of text with static and dynamic objects. The graphics terminal can also handle graphic in blocks. The dynamic objects show the current status for those signals to which the object is linked.

Certain dynamic data is maneuverable which means that its status and value can be changed in run-time. To change a manoeuvrable object use the arrow keys to move the cursor to the correct position so that the object is marked.

Digital objects

Mark the required object for digital objects and then press [ENTER] to change the status.

Analog objects

Analog and Date/Time objects are maneuvered by marking the right object, entering the new value and then pressing [ENTER].

If you decide not to enter the new value and have still not pressed [ENTER] exit from the field with [↑] or [↓]. The original value then remains.

A text object is changed by moving to the object and pressing [ENTER], and an entry field is shown. The entry field is either the first or the last line depending on where the object is located on the screen. Scrolling of the entry field takes place if the text is longer than the screen width. The function keys act as shift keys when entering text. When the text has been entered press [ENTER] and the entry field disappears.

Press [F1][ALARM LIST] for the text terminal and [F10][ALARM LIST] for the graphics terminal if special characters from IBM's character table are to be used in the entry field. The entry field for ASCII codes is shown, state the code for the required character and press [ENTER]. Not all characters can be handled by the screen in which case ¥ is shown.

A multi-choice object is changed by going with the arrow keys to the object. Press [ENTER] and a choice list with all maneuverable states is shown on the screen. Go to the required state, press [ENTER] and the three digital signals linked to this state are forcibly set.

To obtain information in textblock on these dynamic objects in run-time, place the cursor on the required object and press [INFO]. A line of text is shown at the bottom or the top of the screen depending where the object is located. To remove this line of text press [INFO] again. The texts which are defined for the set or reset states are shown for digital objects and the min. and max. values for analog objects.

There is a further information level in the run-time mode. A display with the object's definition is shown at this level. This level is reached by pressing [INFO] while holding [ALARM ACK] pressed. To return to run-time mode press [INFO]. An object's definition can be changed here if YES has been selected in Online settings.

Information about dynamic objects in run-time can only be had for text-blocks.

6.5.2 Graphblocks

Graphics terminal

Move between the maneuverable objects with the arrow keys. The marked object is shown with a blinking frame.

Digital objects

You change the status of digital objects, text, symbol and filler objects, by pressing [ENTER]. [INC] sets and [DEC] resets the signals linked to the object.

Analog objects

You can set the max./min. indicators to the actual value in bar objects by pressing [ENTER].

Enter the value and press [ENTER] to maneuver numeric objects.

Historic data can be shown in run-time mode with trend curves. Mark the required trend curve and press [ENTER]. A dialog box is now shown where the history, time period and date for the data of interest to be presented can be chosen and shown. Close the dialog box with [NEXT].

Exit from the historic presentation by selecting NO for the choice Show history and press [NEXT].

The slide object is maneuvered with the [INC] and [DEC] keys to go stepwise between the scale divisions.

Press [ENTER] for multiple choice objects and a list of choices for all maneuverable states is shown on the screen. Move to the required state and press [ENTER].

Enter the required text for ASCII objects when the object is marked and finish entering with [ENTER]. Otherwise it functions like a normal text object.

The digital clock can be altered by marking the object, stating the required time and pressing [ENTER].

Jump objects are maneuvered by marking the object and pressing [ENTER].

6.5.3 Data buffer

Graphics terminal

This function means that an area of the battery memory is used as an extra memory bank for the PLC system. The PLC program determines what shall be read/written as well as when/where it shall be done.

The size of the memory bank is configured in the graphics terminal. The available buffer area in the graphics terminal depends on how much project memory is used (max project memory is 110,000 bytes).

The PLC program communicates with the memory bank in the terminal via a data control block (4 registers) in the PLC system. The position in the memory area to read from or write to is stated with an index.

Index Memory bank

0 Value 1

1 Value 2

-m Value m+1

Configuration in the graphics terminal

The address in the graphics terminal is configured to the first data control register in the PLC system (see the next section) as is the size of the memory bank (in number of registers, two bytes per register).

The following parameters are defined in Data buffer setting in the Setup mode.

Parameters **Ctrl.blk.reg:** States the first control block register in the PLC system.

Buff.size: The number of registers.

Example of entering

Ctrl.blk.reg: D10

Buff.size: 1000

Data control block in the PLC system

The data control block consists of 4 registers:

Register	Description
Dn0	Command register.
Dn1	Start index in the memory bank to read from/write to.
Dn2	Number of registers to read/write.
Dn3	Start register in the PLC system to read from/write to. (e.g. if 0 is stated in this register, reading/writing will begin in register D0).

The command register is set in the PLC system and can assume the following values:

- 1** Write to memory bank
- 2** Read from memory bank
- 3** Fill the selected area with the value found in Dn3

The terminal replies in the same register in a similar way to the following:

- 0** Reading/writing completed
- FFFF** Error when reading/writing

6.5.4 Alarm

The alarm is shown by the text ■ALARM ■ in the upper right-hand corner of the screen (can be cancelled in the Setup mode). The alarms are presented in an alarm list with the predefined alarm texts. This contains the latest alarms and is organized so that the latest is at the top. How many alarms the list can contain is determined in the Setup mode.

This list is shown on the screen when jumping to alarm block. The status of the alarm is shown for each alarm.

The alarm can have the following status:

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

To reach the alarm block, either press [ALARM LIST] or define in a block a jump to system 990 or make the PLC system get the list via Display signal for block 990.

To acknowledge an alarm, place the cursor on the line where the alarm is and press [ALARM ACK]. A text with the alarm number as well as the day and time it occurred is shown at the bottom. This display can be switched off by pressing [INFO]. The alarms can be printed out as they happen or change status if a printer is connected. Alarms are then printed with date, time, status and alarm text.

The text “Comm error” is placed in as an event alarm in the alarm list if a communication error occurs between the terminal and the PLC system.

Press [ALARM LIST] to return to the textblock shown before the jump to alarm block.

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

6.5.5 Time channels

The page with time channels is reached by stating the Display signal in system block 991 or by jumping there. Move the cursor to the required line to mark a time channel in order to read it or change its value. Then press [ENTER]. The time channel's value can now be read and/or changed.

The digital signal cannot be changed though in run-time. The choice Time channels in the menu Online settings must be set to YES in the Setup to be able to change the time channel's value in run-time. To leave the current time channels definition, press [LEAVE] or move the cursor to RETURN and press [ENTER]. Exit from the time channel menu with the [MAIN MENU] key after which you return to the block from which you jumped to the time channel block.

The current time channel settings are printed by stating Printer signal for block 991.

6.5.6 Function keys

Graphics terminal

The digital signal linked to a function key is set when the key is pressed. The signal is reset when you release the key. In ASCII object the function keys [F8] - [F13] are used as shift keys. When displaying historic data in trendcurves the keys [F8] - [F13] are not available as function keys.

6.5.7 LEDs

Graphics terminal

Depending on the value in the data register linked to the LED, the LED will light and flash according to the following tables:

Register value (Hex)	Register value (Dec)	Flashing frequency (Hz)	Color
00	0	-	None
01	1	-	Green
02	2	-	Red
03	3	-	Yellow
11	17	5	Green
12	18	5	Red
13	19	5	Yellow
21	33	2,5	Green
22	34	2,5	Red
23	35	2,5	Yellow
31	49	1,2	Green
32	50	1,2	Red
33	51	1,2	Yellow
41	65	0,6	Green
42	66	0,6	Red
43	67	0,6	Yellow

7 Overview

This chapter describes the modes, dynamic object, system blocks, memory capacity, character set, character tables and technical data.

7.1 Modes

The terminal has three different modes:

Programming mode Switch in **prog** position

Run-time mode Switch in **run** position

Setup mode Switch in **prog** position
(press any key at initial display)

Dynamic text objects

The following objects can be used in blocks:

Digital	State (two texts)
Analog	Presentation of value
Bar graph	Proportional filling of analog object
Multiple choice	State (eight texts)
Text	Text string
Jump	Jump to block
Date/Time	Date and/or time

Dynamic graph objects

Only the graphics terminal can handle graphic objects. The following graphic objects can be used in blocks.

ASCII	Object for handling ASCII strings
Filled object	Object for filling in framed areas
Symbol	Digital object for moving between two symbols
Numeric	Object for entering and displaying analog values
VU-meter	Object for graphic VU meter
Bar	Object for presentation of values as a bar graph
Trend	Object for presentation of recorded values
Slide	Object for increase/decrease values via keys
S-meter	Object for a graphic speed indicator

Mutisymbol	Object for displaying eight different symbols
Multiple choice	Object which can assume eight different states
Jump	Object for jumping between blocks in a project
Digital clock	Object for digital clock
Analog clock	Object for analog clock
Alarm banner	Object for displaying one line in the alarm list
Diagram	Object to draw a diagram based on the contents of PLC registers

System blocks

Certain blocks are reserved for special purposes. The system blocks differ between the text terminal and the graphics terminal.

Text terminal

0	Main menu
990	Alarm
991	Time channels
992-998	<i>Reserved, not used</i>
999	Function keys

Graphics terminal

0	Main menu
990	Alarm
991	Time channels
992-996	<i>Reserved, not used</i>
997	Contrast settings of LCD display
998	LED
999	Function keys

How much can be stored in a project

Maximum size for a project is 27,900 bytes for text terminal and 110,000 bytes for graphics terminal. These bytes can be freely assigned to different functions.

The following tables shows how many bytes each function requires:

Object	Text terminal	Graphics terminal
Block	46 bytes	51 bytes
Text in block	20 bytes/row	4 bytes/character
Digital object	25 bytes	28 bytes
Analog object	29 bytes	32 bytes
Jump object	6 bytes	9 bytes
Date/time object	8 bytes	11 bytes
Bar graph object	28 bytes	31 bytes
Multiple choice object	88 bytes	91 bytes
Text object	11 bytes	15 bytes
Time channel	73 bytes	75 bytes
Alarm object	33 bytes	39 bytes

Object	Graphics terminal
ASCII	12 bytes
Digital text	27 bytes
Fyller object	9 bytes
Symbol (digital)	24 bytes
Numeric	30 bytes
VU-meter	39 bytes
Bar graph	43 bytes
Trend	70 + number of defined signals * (number of stored values*6+2) bytes
Slide	40 bytes
S-meter	39 bytes
Multi symbol	75 bytes
Multiple choice	82 bytes

Object	Graphics terminal
Jump	30 bytes
Digital clock	9 bytes
Analog clock	6 bytes
Alarm banner	13 bytes
Diagram	50 bytes
Rectangle	7 bytes
Filled rectangle	7 bytes
Line	7 bytes
Ellipse	7 bytes
Filled ellipse	7 bytes
Arc	11 bytes
Text	35 bytes
Symbol	35 bytes
User defined symbol	$(X*Y)/8+16$ bytes
Data buffer	Buffer size*2+5 bytes
















When displaying a block in programming mode, press [F1] [INFO] for the text terminal and [F10] [INFO] for the graphics terminal to see how many bytes remain available. This value allows you to make a quick estimate of how much space you have used. For example, an application demands the following space in the text terminal.

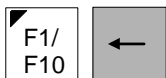
Block	46
One text row	20
Digital object	25
<hr/>	
total	91 bytes

7.2 Keys

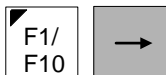
Certain keys have different functions in Programming and Run-time mode. These keys have the following functions:

Programming mode

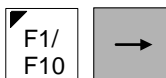
	Shift
	Shift
	Shift
	Shift
 	Leave
 	Next
   	Move
	Change among alternatives Show status word menu
	Erase previous character when entering
	Next line



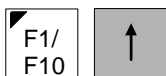
Erase the character where the cursor is positioned. Move all characters after the cursor one step to the left.



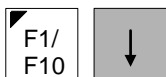
Move all characters after the cursor one step to the right.



Insert of characters



Upper case (capital) letters



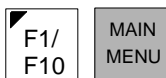
Lower case (small) letters



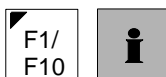
Katakana/Cyrillic (if either character set is chosen).



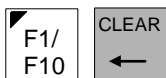
Roman (if Roman character set is chosen). Only valid for the text terminal.



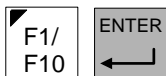
Access to special character table



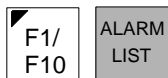
Number of bytes left.



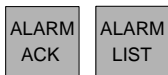
Carriage return (ASCII:0D hex for EOL textblock).



Erase block in block menu.



Possible to type ASCII characters with numerical code.



Alter between programming and run-time mode (when the switch is in PROG position).

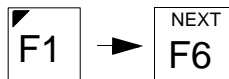


Page up in programming menus.

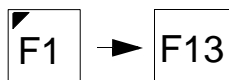


Page down in programming menus.

Run-time mode



Functions keys in the text terminal.



Function keys in the graphics terminal.



Move between objects.



One line up or down.



Show 0- and 1-texts for digital objects.
Show min and max limit for analog objects.



Show text block 0.



Show alarm list.



Show textblock 0 (Text terminal only).



Acknowledge one alarm in alarm list.



Permit jump at jump object.
Finish input analog object.
Force digital object.
Input field text object.
Selection list multiple-choice object.



Erase previous character when entering.



Possible to type ASCII characters with numerical code.



Alters between programming- and run-time mode when the switch is in PROG position.



Set/increase



Reset/decrease



Page up in textblock.



Page down in textblock.



Return to previous textblock.

7.3 Character settings

In the terminal you will find the following characters:

A-Z

a-z

0-9

' ` : , ° % + = - * / () ! ? . # < >

C1-C8 correspond to different characters, depending on the type of character settings chosen in Setup.

Text terminal

	Swedish		German		French		Spanish		Dan/Nor	
	upp.	low.	upp.	low.	upp.	low.	upp.	low.	upp.	low.
C1	Å	å	ü	ü	È	è	ñ	ñ	Å	MV:å ML:Å
C2	Ä	ä	Ä	ä	É	é			Æ	æ
C3	Ö	ö	Ö	ö	Ê	ê			Ö	Ö
C4			ß	ß					Ø	ø

Graphics terminal

	Swedish		German		French		Spanish		Dan/Nor	
	upp.	low.	upp.	low.	upp.	low.	upp.	low.	upp.	low.
C1	Å	å	Û	ü	È	è	Ñ	ñ	Å	å
C2	Ä	ä	Ä	ä	É	é	È	é	Æ	æ
C3	Ö	ö	Ö	ö	Ê	ê	ó	ó	Ö	Ö
C4			ß	ß			á	á	Ø	ø

There are also some special characters available if you press [F1][MAIN MENU] on the text terminal and [F10] [MAIN MENU] on the graphics terminal.

Text terminal with LCD display:

\$ & ; @ [] ^ - { } ß ε Ω ω ÷ ■ ° α ∞ ¥

Text terminal with VF display:

\$ & ; @ [] ^ - { } ß ε Ω ω : ■ ° α ∞ Ø

Graphics terminal with LCD display:

All characters corresponding to the codes 128 - 255 according to the IBM table 850 except 179, 186, 188, 191, 192, 218. Swedish and German character set uses table 437.

Character tables

Text terminal

When you have selected Katakana/Cyrillic character set, the keys' functions change as follows: 40 Hex is added to the key code; i.e. the character is displayed four columns to the right in the character table.

Example:

If you press C on the keyboard, the character * appears on the display if you have a MAC 50MV/MTA-250MV.

Roman and Katakana/Cyrillic characters can be displayed simultaneously. Switch between character sets with the key combination [F2] [↑] or [F2] [↓]. Use ASCII-code entry, via [F1] [ALARM LIST] for those characters which can not be entered with the alphanumeric keys.

Note!

Codes should be specified decimally.

Overview

Decimal Value	♦		32	48	64	80	96	112	128	144	160	176	192	208	224	240
♦	Hex Value		2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0			0	@	P	`	p	Ç	É					α	≡
1	1		!	l	A	Q	α	q	ü			¶			β	
2	2		"	2	B	R	b	r	é	Æ		¶				
3	3		#	3	C	S	c	s							π	
4	4		\$	4	D	T	d	t	ä	ö	ñ		-		Σ	
5	5		%	5	E	U	e	u			Ñ					
6	6		&	6	F	V	f	v	đ						μ	÷
7	7		'	7	G	W	g	w	ç							
8	8		(8	H	X	h	x	ê	ÿ						°
9	9)	9	I	Y	i	y		Ö					θ	•
10	A		*	:	J	Z	j	z	è	Ü					Ω	·
11	B		+	:	K	[k	{						■		√
12	C		,	<	L	\	l								∞	
13	D		-	=	M]	m	}						■		
14	E		.	>	N	^	n	~	Ä		«			■	€	
15	F		/	?	O	_	o	⌘	Å		»					

Character table for MAC 50ML / MTA-250ML.

Decimal Value	♦	32	48	64	80	96	112	128	144	160	176	192	208	224	240
♦	Hex Value	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0		0	@	P	`	p	Ç	É	á				α	≡
1	1	!	1	A	Q	α	q	ü	æ	í	Ⓜ			β	≠
2	2	"	2	B	R	b	r	é	Æ	ó	Ⓜ			Γ	≥
3	3	#	3	C	S	c	s	à	ô	ú				π	≤
4	4	\$	4	D	T	d	t	ä	ö	ñ		-		Σ	f
5	5	%	5	E	U	e	u	à	ò	Ñ				σ	J
6	6	&	6	F	V	f	v	á	û					μ	+
7	7	`	7	G	W	g	w	ç	ù	ø				τ	≈
8	8	(8	H	X	h	x	ê	ÿ	ı				φ	°
9	9)	9	I	Y	i	y	ë	Ö					θ	•
10	A	*	:	J	Z	j	z	è	Ü					Ω	•
11	B	+	;	K	[k	{	ï	ç	½			■	δ	√
12	C	,	<	L	\	l		î	£	¼				∞	n
13	D	-	=	M]	m	}	ï	¥	ı			■	φ	2
14	E	.	>	N	^	n	~	Ä		«			■	€	■
15	F	/	?	O	_	o	⌘	Å	f	»				n	

Character table for MAC 50MV/ MTA-250MV.

		D7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1				
		D6	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1				
		D5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1				
		D4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0				
D3	D2	D1	D0		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	0	0	0			SP	0	0	P	'	P	E	U		一	ウ	三	日	!	
0	0	0	1	1		DC1	!	1	A	0	a	4	F	3		7	チ	4	月	!	
0	0	1	0	2		DC2	"	2	B	R	b	r	Q	W		「	イ	ウ	×	火	■
0	0	1	1	3			#	3	C	S	c	s	*	R		」	ウ	テ	モ	■	
0	1	0	0	4		DC4	\$	4	D	T	d	t	3	0		、	イ	ト	ト	*	!
0	1	0	1	5		DC5	%	5	E	U	e	u	U	0		・	オ	オ	1	金	!
0	1	1	0	6		DC6	&	6	F	V	f	v	R	0		ヲ	カ	ニ	ヨ	土	■
0	1	1	1	7		DC7	'	7	G	W	w	w	0		ヲ	チ	又	ウ	年	■	
1	0	0	0	8	BS	CT0	(8	H	X	h	x	0		ヲ	イ	ワ	本	リ	余	■
1	0	0	1	9	HT	CT1)	9	I	Y	i	y	H		ト	5	ワ	リ	ル	肉	〒
1	0	1	0	A	LF		*	:	J	Z	j	z	0		+	エ	コ	ノ	レ	と	↓
1	0	1	1	B		ESC	+	:	K	C	k	c	0		+	オ	オ	ヒ	ロ	〜	÷
1	1	0	0	C	FF		,	<	L	\	l	\	0		+	オ	ウ	ウ	ウ	■	↑
1	1	0	1	D	CR		-	=	M	I	m	i	0		+	ユ	ズ	ノ	シ	0	+
1	1	1	0	E	CLR		.	>	N	^	n	^	0		+	ウ	ヲ	ヨ	セ	ホ	°
1	1	1	1	F			/	?	0	_	o	_	0		+	キ	ト	ス	ウ	マ	♪

Katakana/Cyrillic character table in the MAC 50MV/MTA-250/MV.

Higher Lower bit bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxx0000	CG RAM (1)		0	A	P	'	F	-	9	3	0	P	
xxx0001	(2)	1	A	a	a	a	7	7	4	a	a		
		2	B	R	B	r	'	Y	X				
xxx0010	(3)	3	C	S	c	a	U	T	T	E	a		
		4	#	3	5	c	a	U	T	T	E	a	
xxx0011	(4)	5	\$	4	D	T	a	t	.	T	T	P	a
		6	%	5	E	U	e	u	.	T	T	a	e
xxx0100	(5)	7	&	6	F	U	f	u	.	T	T	a	e
		8	'	7	G	W	a	w	7	T	T	a	e
xxx0101	(6)	9	(8	H	X	h	x	4	0	*	U	.
		10)	9	I	V	i	v	.	T	T	a	e
xxx0110	(7)	11	*	#	J	Z	j	z	.	T	T	a	e
		12	+	#	K	E	k	.	T	T	a	e	*
xxx0111	(8)	13	,	<	L	#	l	.	T	T	a	e	*
		14	-	=	M	I	m	.	T	T	a	e	.
xxx1000	(9)	15	.	>	N	^	n	.	T	T	a	e	.
		16	/	?	0	_	o	.	T	T	a	e	.
xxx1001	(10)	17											
		18											
xxx1010	(11)	19											
		20											
xxx1011	(12)	21											
		22											
xxx1100	(13)	23											
		24											
xxx1101	(14)	25											
		26											
xxx1110	(15)	27											
		28											
xxx1111	(16)	29											
		30											

Katakana/Cyrillic character table in the MAC 50/ML / MTA-250/ML.

Graphics terminal

Decimal Value	*		32	48	64	80	96	112	128	144	160	176	192	208	224	240
*	Hex Value		2	3	4	5	ó	7	8	9	A	B	C	D	E	F
0	0			0	@	P	`	p	Ç	É	á		Ł	⋈	α	≡
1	1		!	l	A	Q	α	q	ü	æ	í	⋈	⊥	⊞	β	±
2	2		"	2	B	R	b	r	é	Æ	ó	⋈	⊞	⊞	Γ	≥
3	3		#	3	C	S	c	s	á	ó	ú		⊞	⋈	π	≤
4	4		\$	4	D	T	d	t	ä	ö	ñ	⊞	⊞	⋈	Σ	í
5	5		%	5	E	U	e	u	á	ó	Ñ	⋈	⊞	F	σ	J
6	6		&	6	F	V	f	v	á	ú	α	⋈	⋈	⋈	μ	+
7	7		'	7	G	W	g	w	c	ú	o	⋈	⋈	⋈	τ	≈
8	8		(8	H	X	h	x	é	ý	¿	⋈	⋈	⋈	φ	°
9	9)	9	I	Y	i	y	ë	Ö	Γ	⋈	⋈	J	θ	•
10	A		*	:	J	Z	j	z	è	Ü	Γ	⋈	⋈	⋈	Ω	•
11	B		+	:	K	[k	{	ï	¢	½	⋈	⋈	⋈	δ	√
12	C		,	<	L	\	l		î	£	¼	⋈	⋈	⋈	∞	π
13	D		-	=	M		m	}	ï	¥	ı	⋈	=	⋈	φ	²
14	E		.	>	N	^	n	~	Ä	Pl	«	⋈	⋈	⋈	€	■
15	F		/	?	O	-	o	⏏	Å	f	»	⋈	⋈	⋈	∅	

Character table for ASCII-IBM codes (437).

DECIMAL VALUE	→	0	16
↓	HEXA DECIMAL VALUE	0	1
0	0	NUL	DLE
1	1	SOH	DC1
2	2	STX	DC2 XON
3	3	ETX	DC3 XOFF
4	4	EOT	DC4
5	5	ENQ	NAK
6	6	ACK	SYN
7	7	BEL	ETB
8	8	BS	CAN
9	9	HT	EM
10	A	LF	SUB
11	B	VT	ESC
12	C	FF	FS
13	D	CR	GS
14	E	SO	RS
15	F	SI	US

Control codes

7.4 Technical data

	Text terminal	Graphics terminal
Front panel	190 x 175 x 3 mm WxHxD. Black anodized aluminium.	240 x 200 x 3 mm WxHxD. Black anodized aluminium.
Mounting depth	Exkl. D-sub connector 50 mm. Inkl. D-sub connector 100 mm.	Exkl. D-sub connector 54 mm. Inkl. D-sub connector 105 mm.
Panel cut out	Hole: Width 171 x 156 mm, 4 pcs mounting screws, M4, C-C 178 x 163 mm. Max panel thickness 8 mm.	Hole: Width 220 x 180 mm, 8 pcs mounting screws, M4. Max panel thickness 8 mm.
Front panel seal	IP65	
Rear panel seal	IP20	
Front panel/keyboard material	Membrane keyboard. Overlay film in polyester, AUTOTEX 2 F150, with backside printing. Lifetime: 1 miljon operations.	
Back side material	Black anodised aluminium	
Weight	Exkl. D-sub cable 1,1 kg.	Exkl. D-sub cable 1,6 kg.
Serial port RS-422	15-pin D-sub connector, chassis-mounted female with standard lock screws 4-40 UNC, RS-422 communication.	
Serial port RS-232	9-pin D-sub connector, male with standard screw lock 4-40 UNC, RS-232 communication.	
Expansion connector	26-pin Header for standard connector.	
Cable to PLC system	6-pair cable, shielded, shielded D-sub cover of metallized plastic. 25-pin D-sub to PLC system and 15-pin D-sub to RS-422 outlet in the terminal. Locking screws to the PLC system: M2,5. Locking screws to the terminal 4-40 UNC.	
Display	Text terminal /ML: LCD-display (Liquid crystal), Supertwist 4 rows x 20 characters, 8 mm char. hight. Viewing angle adjustable with potentiometer behind battery cover. Lifetime: 50000 timmar. Text terminal /MV: VF-display (vaccum flourescent), 4 rows x 20 characters, 5 mm char. hight. Light intensity software adjustable. Lifetime: 50000-150000 timmar.	LCD-display (Liquid crystal), 240 x 128 pixel, 16 rows x 30 character. Lifetime: for CFL backlight 10000 hours.

	Text terminal	Graphics terminal
Battery	Battery for real-time clock and CMOS-RAM. Lifetime >5 years. Norm. 10 years. Manufacturing year printed on the back of the battery cover. Lithium battery 3,6 V, Capacity: 2 Ah. Size AA (ANSI) or R6 (IEC), 14 x 50 mm.	
Battery change	Warning at low battery voltage is nominal 3,21 V (between 2,96 V and 3,45 V). Change battery with supply connected. If the terminal is not connected to supply, please change within 1 min to avoid loss of information in CMOS-RAM and temporary stoppage of real-time clock.	
Real-time clock	± 10PPM + error because of ambient temperature and supply voltage. Total max error: 1 min/month = 12 min/year.	
Supply voltage	Nominal +24 VDC. Max 30 V, min 20 V. Max ripple: 3 V peak to peak.	
Supply voltage connection	3-pol screw terminal. Max cable area: 2,5 mm ²	
Current consumption without expansion unit	<p>LCD display Nominal approx 80 mA. Higher at 20 V, lower at 30 V. Power consumption: approx: ca 2 VA. Dissipation effect: ca 2 W.</p> <p>VF display Nominal approx. 250 mA. Higher at 20 V, lower at 30 V. Power consumption: approx: 6 VA. Dissipation effect: 6 W.</p>	Nominal ca 400 mA. higher at 20 V, lower at 30 V. Power consumption: approx: 10 VA. Dissipation effect: 10 W.
Internal power	+5 VDC ±5%. Connected to the chassis via 0,1 µF capacitor parallel with 1 Mohms resistance.	
Ambient temperature	<p>MAC 50/MTA-250 ML: 0° to +50°C.</p> <p>MAC 50/MTA-250 MV: 0° to +55°C.</p>	0° to +50°C
Storage temperature	-20° to +55°C	
Relative humidity	Max 90% non-condensing	
EMC-test	G & L Beijer Electronics AB confirm that the terminals conforms with the essential protection requirements in article four or the directive 89/336/EEC Noise tested according to: EN50081-1 emission and 50082-2 immunity.	

Index

A

A series, [13](#)
Alarm banner, [86](#)
Alarm settings, [64](#)
Alarm, [4](#), [101](#)
Analog signals, [61](#)
Analog, [80](#)
Arc, [85](#)
ASCII, [85](#)

B

Bar graph, [81](#), [87](#)
Basic editing functions, [59](#)
Basic functions, [3](#)
Battery check, [14](#)
Block header, [26](#), [77](#)
Block jump, [99](#)
Block list, [103](#)
Block number, [77](#)
Block, [26](#)

C

Character set, [65](#)
Character settings, [121](#)
Character tables, [123](#)
Choice of language and character set, [23](#)
Choice of PLC system, [23](#)
Clock synchtonization, [46](#)

Communication, [6](#), [37](#)
 Multidrop, [41](#)
 Network, [44](#)
 No protocol mode, [47](#)
 Transparent mode, [37](#)
Connecting to the CPU port, [14](#)
Connection to a printer, [19](#)
Connection to an
A1SJ71C24-R2, [18](#)
Connection to an
A1SJ71C24-R4, [17](#)
Connection to an A2CCPU-C24,
A2CCPU-C24, [19](#)
Connection to an AJ71C24/
AJ71UC24, [17](#)
Connection to MELSEC ME-
DOC, [20](#)
Contrast settings, [66](#)

D

Data logging/Recipe, [6](#)
Databuffer files, [56](#)
Databuffer setting, [66](#)
Date/time format, [67](#)
Date/time, [81](#)
Device driver, [55](#)
 Install, [55](#)
Diagram, [88](#)
Digital signals, [60](#)
Digital symbol, [89](#)
Digital text, [90](#)
Digital, [82](#)
Disk settings, [69](#)
Display signal, [77](#)

Documentation, [103](#)
Dynamic graph objects, [113](#)
Dynamic text objects, [113](#)

E

Ellipse, [90](#)
Erase memory, [69](#)

F

F1/F2, [13](#)
Filler, [90](#)
Function keys, [5](#), [99](#)
FX series, [13](#)

G

Gain, [63](#)
Graphblocks, [4](#)

I

Index addressing, [62](#)
Index register, [62](#), [69](#)

J

Jump, [82](#), [91](#)

K

Keyboard, [7](#)
Keys, [117](#)

L

Language, [70](#)
LEDs, [5](#), [100](#)
Line pattern, [91](#)
Lines, [91](#)

M

MACMTA.SYS, [55](#)
MELSEC MEDOC, [39](#)
Memory cassette, [104](#)
Menu structure, [25](#)
Menu system, [10](#)
Method for programming, [25](#)
Modes, [7](#), [113](#)
Multidrop, [41](#)
Multiple choice, [83](#), [92](#)
Multisymbol, [92](#)

N

Network, [44](#)
No protocol mode, [47](#)
Numeric, [93](#)

O

Offset, [63](#)
Online settings, [70](#)

P

Password, [79](#)
PLC selection, [71](#)
PLC system, [13](#)
Port parameters, [71](#)
Power requirements, [13](#)
Printer setting [72](#)
Programming mode, [77](#)

R

Rectangle, [93](#)
 Filled, [93](#)
 Transparent, [93](#)
Remote control, [57](#)
Reserved characters, [63](#)
Run-time mode, [105](#)

S

Scaling, [63](#)
Setup mode, [64](#)
Signal format, [60](#)
Slide, [95](#)
S-meter, [94](#)
Status word, [78](#)
Suffix, [61](#)
Switch between PROG and
RUN, [59](#)
Symbol (static), [95](#)
Symbol maintenance, [96](#)
System block, [114](#)
System signals, [73](#)

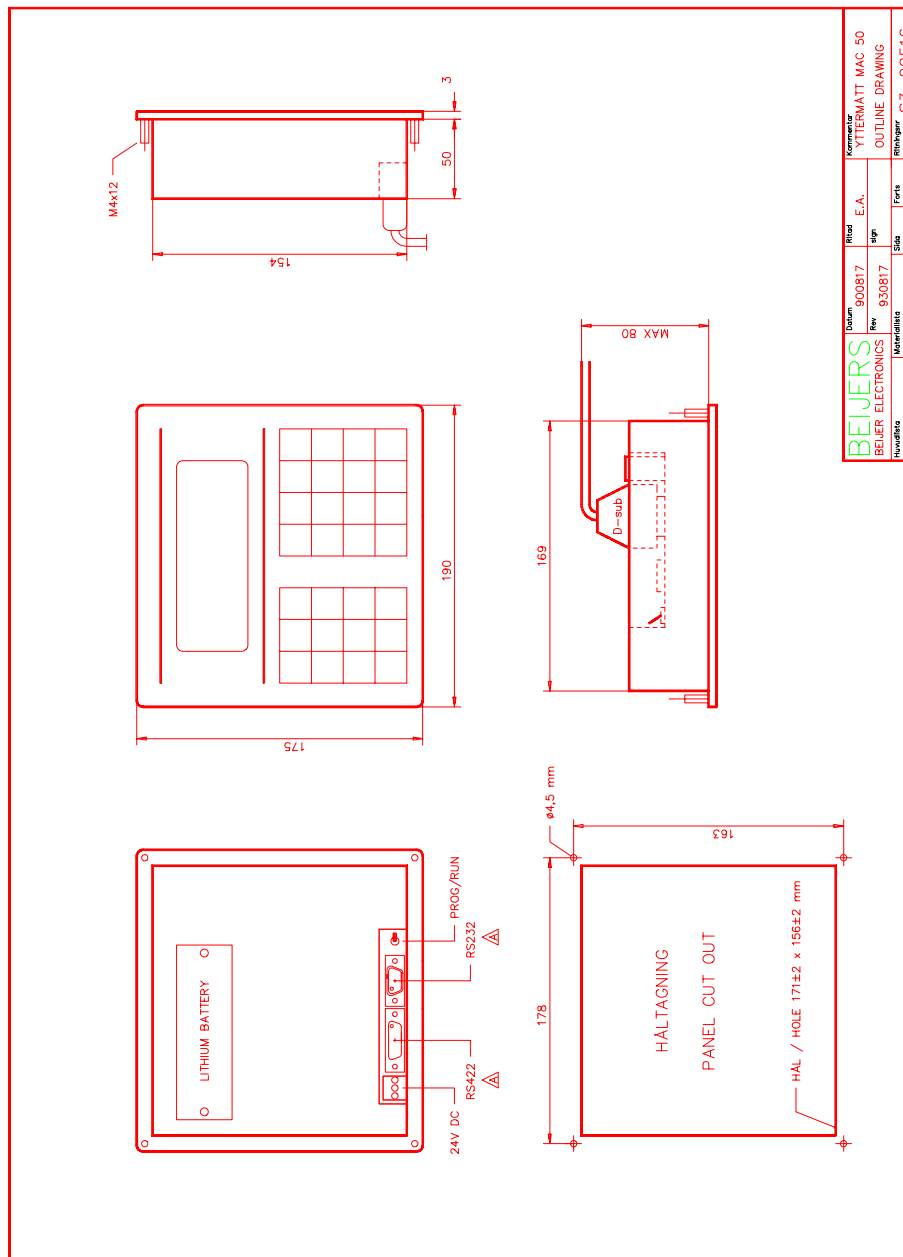
T

Technical data, [130](#)
Terminal commands, [58](#)
Terminal options, [76](#)
Text (static), [96](#)
Text, [84](#)
Textblock, [3](#), [80](#), [105](#)
Time channels, [5](#), [102](#)
Time, [96](#)
Transfer databuffer, [55](#)
Transfer trend data, [55](#)
Transfer, [104](#)
Transparent mode, [37](#)
Trend files, [56](#)
Trend, [97](#)

V

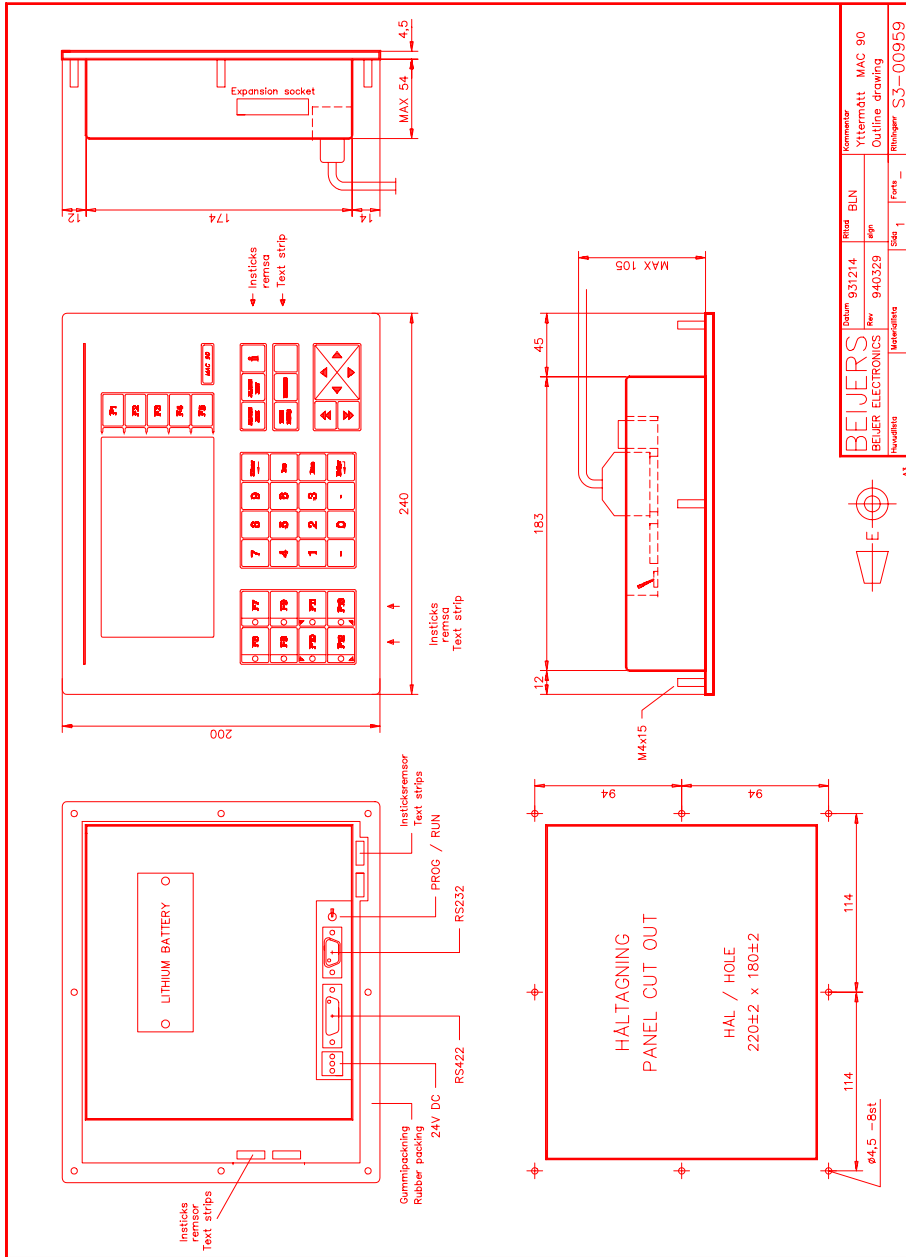
VU meter, [98](#)

Drawings

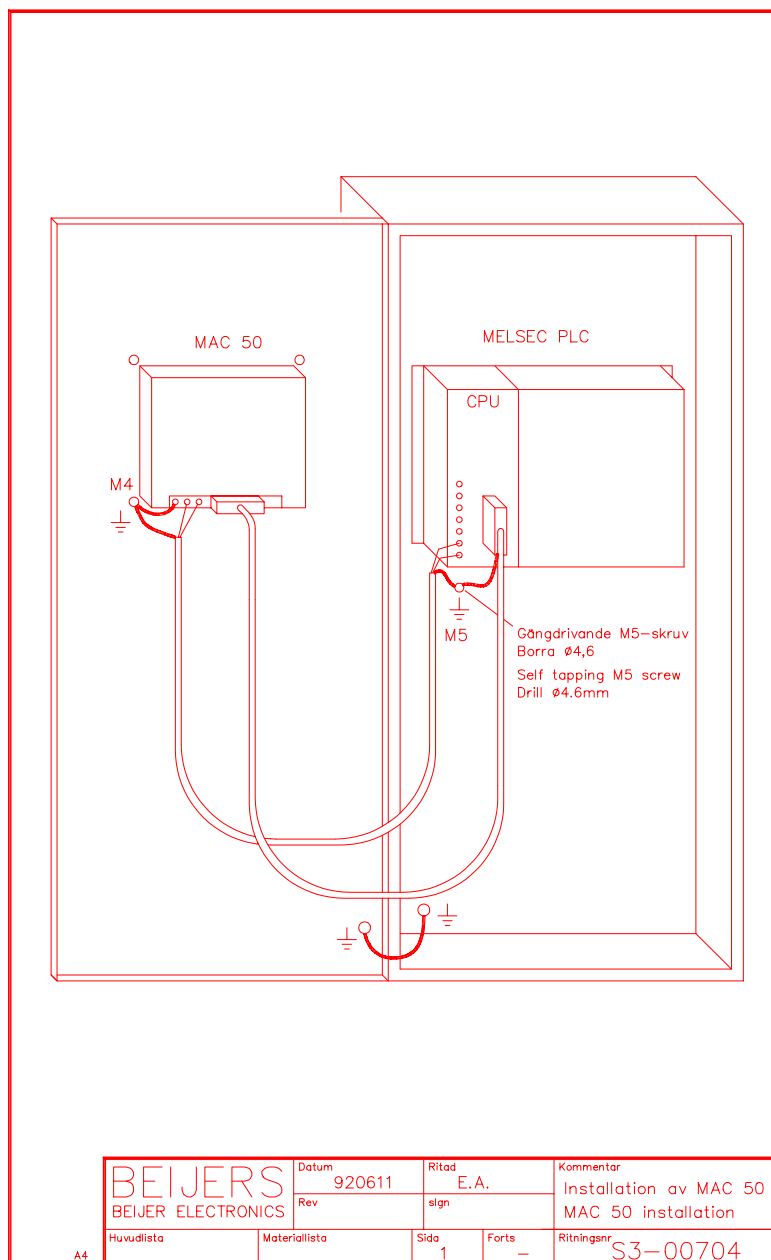


BEIJERS		Date: 900817		Revision: 930817		Scale: 1		Drawing No: S3-00516	
BEIJER ELECTRONICS		E.A.		YTERMATT MAC 50		Form: -		Material: -	
Headoffice		Middelste		Site		Date		Revision	
A3									

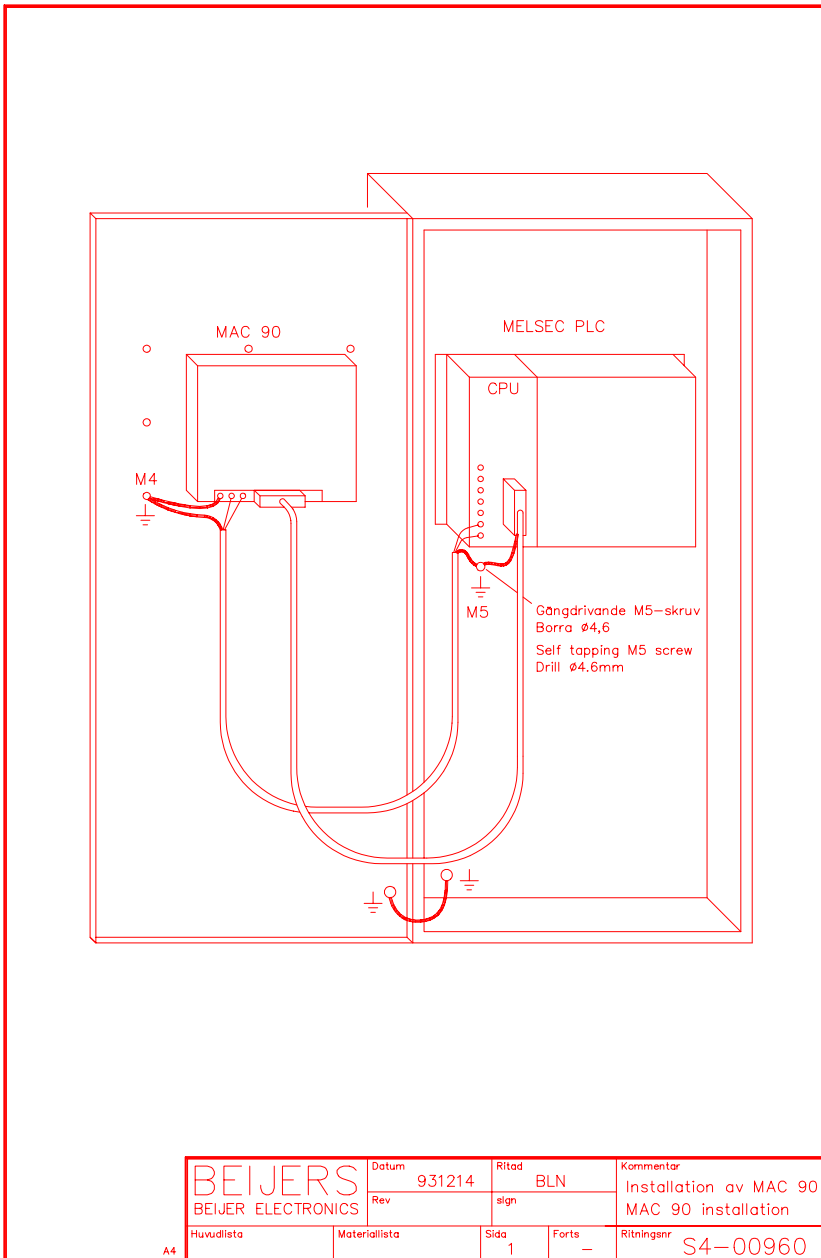
Dimensions for MAC 50/MTA-250



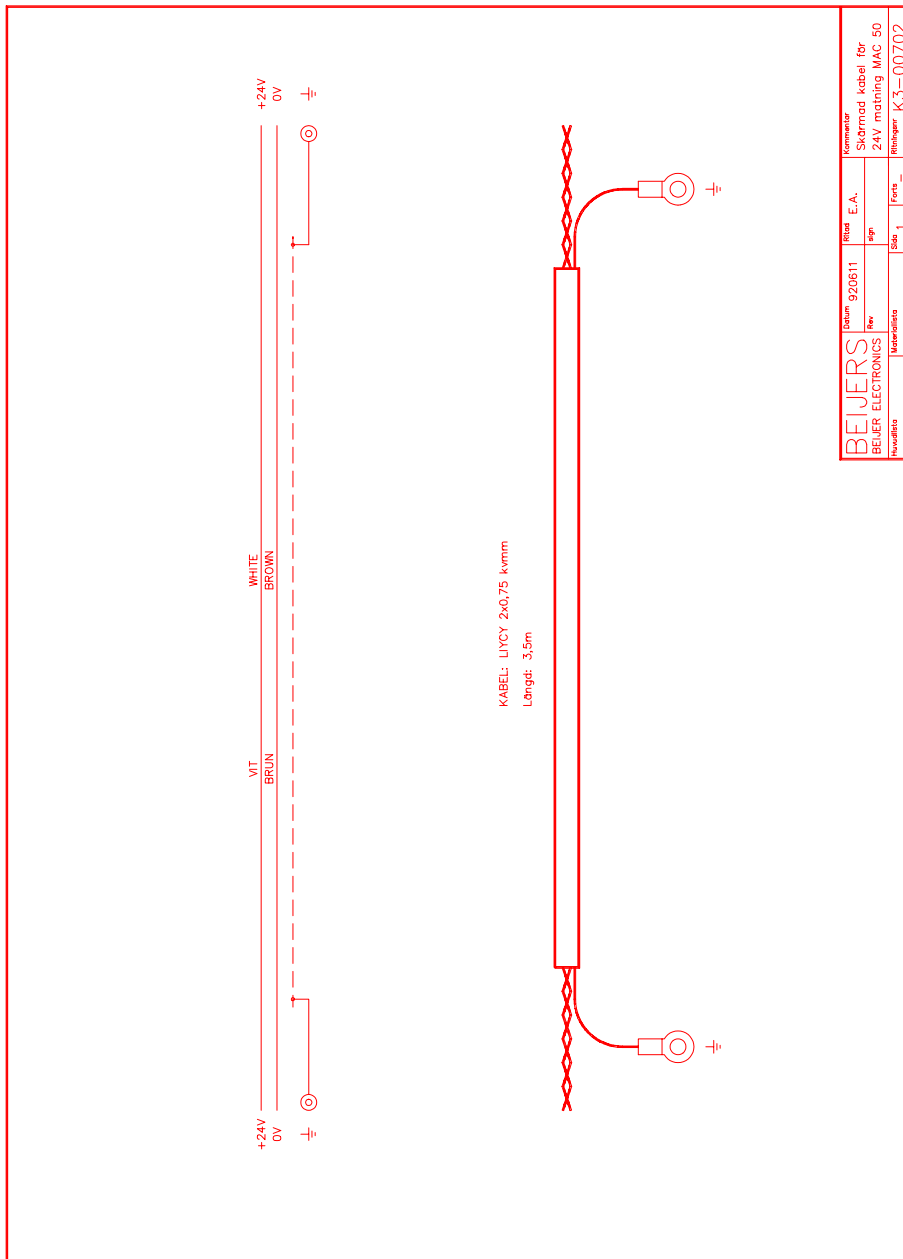
Dimensions for MAC 90/MTA-G1



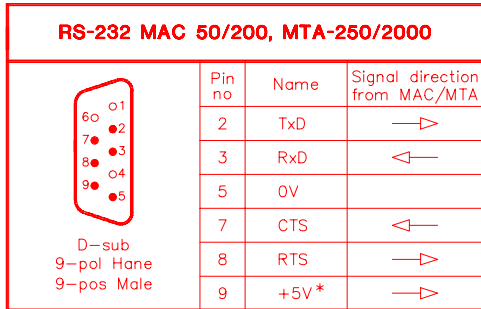
Installation of MAC 50/MTA-250



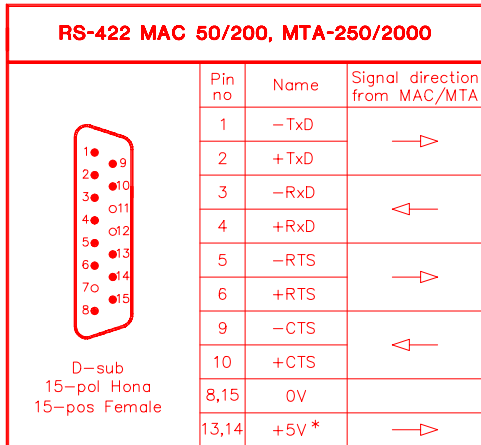
Installation of MAC 90/MTA-G1



Shielded cable for power supply



* MAX 5mA

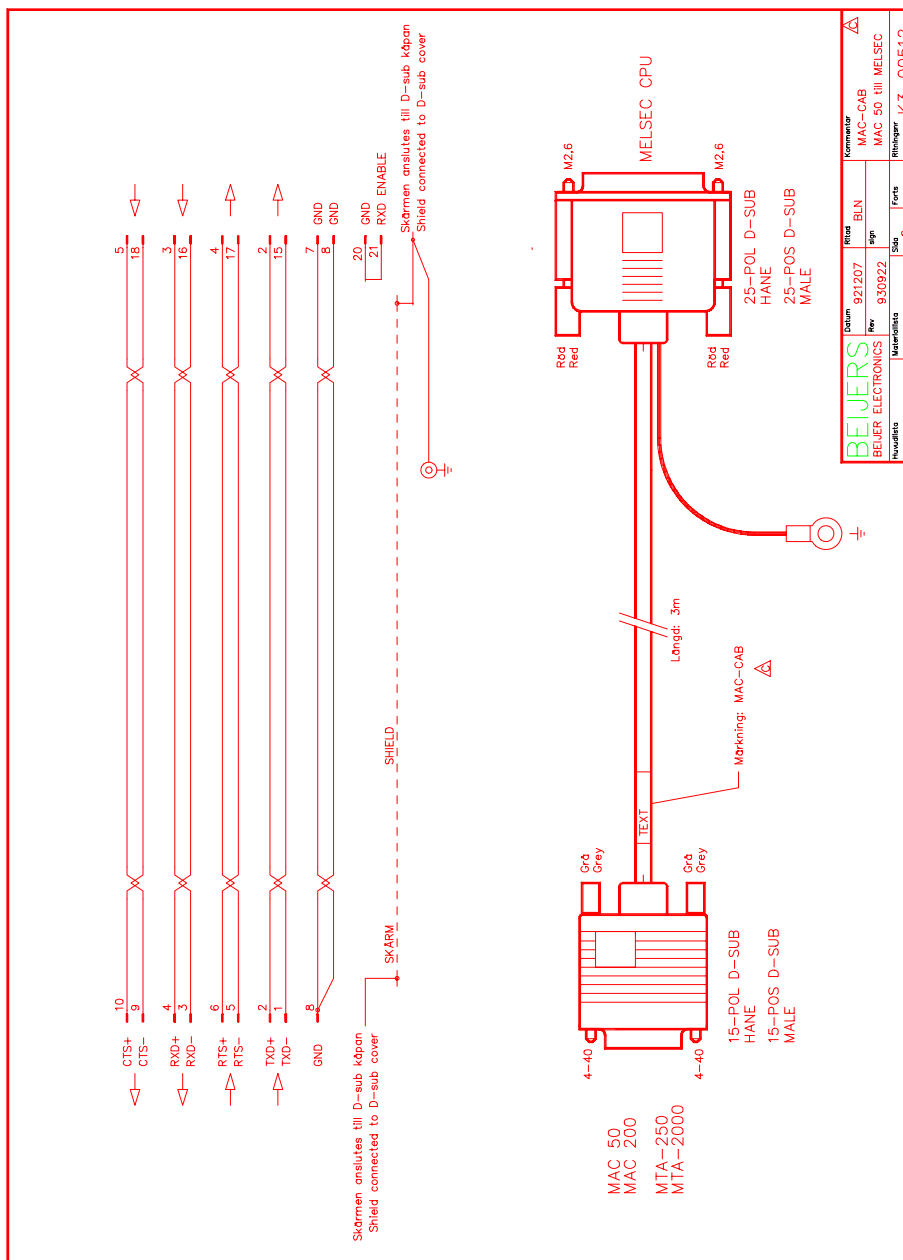


* MAC 50, MTA-250 MAX 20mA
MAC 200, MTA-2000 MAX 400mA

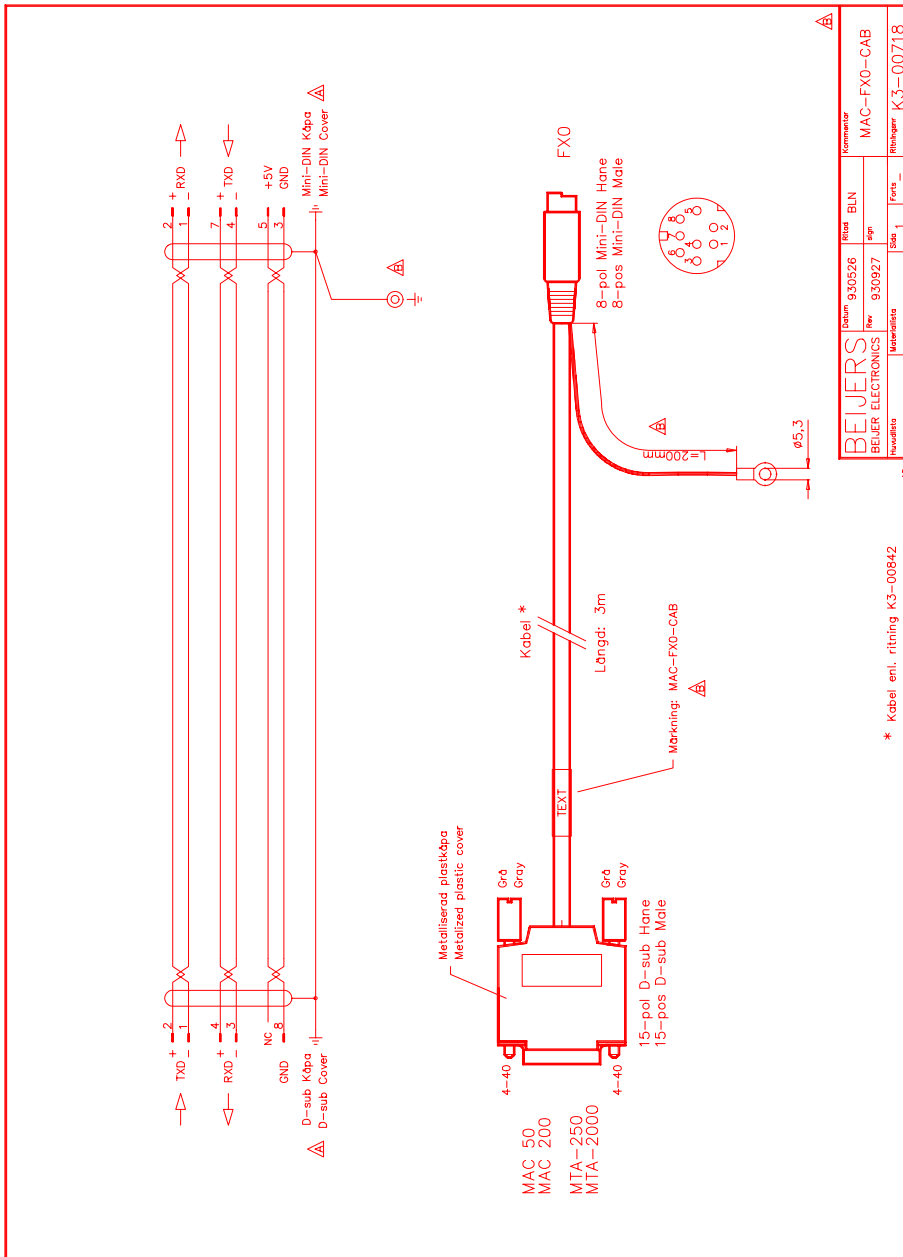
BEIJERS BEIJER ELECTRONICS	Datum	930907	Ritad	BLN	Kommentar MAC 50/200 RS232/RS422 Anslutn.
	Rev		sign		
Huvudlista	Materialista	Sida	1	Forts	-
				Ritningsnr	S4-00724

A4

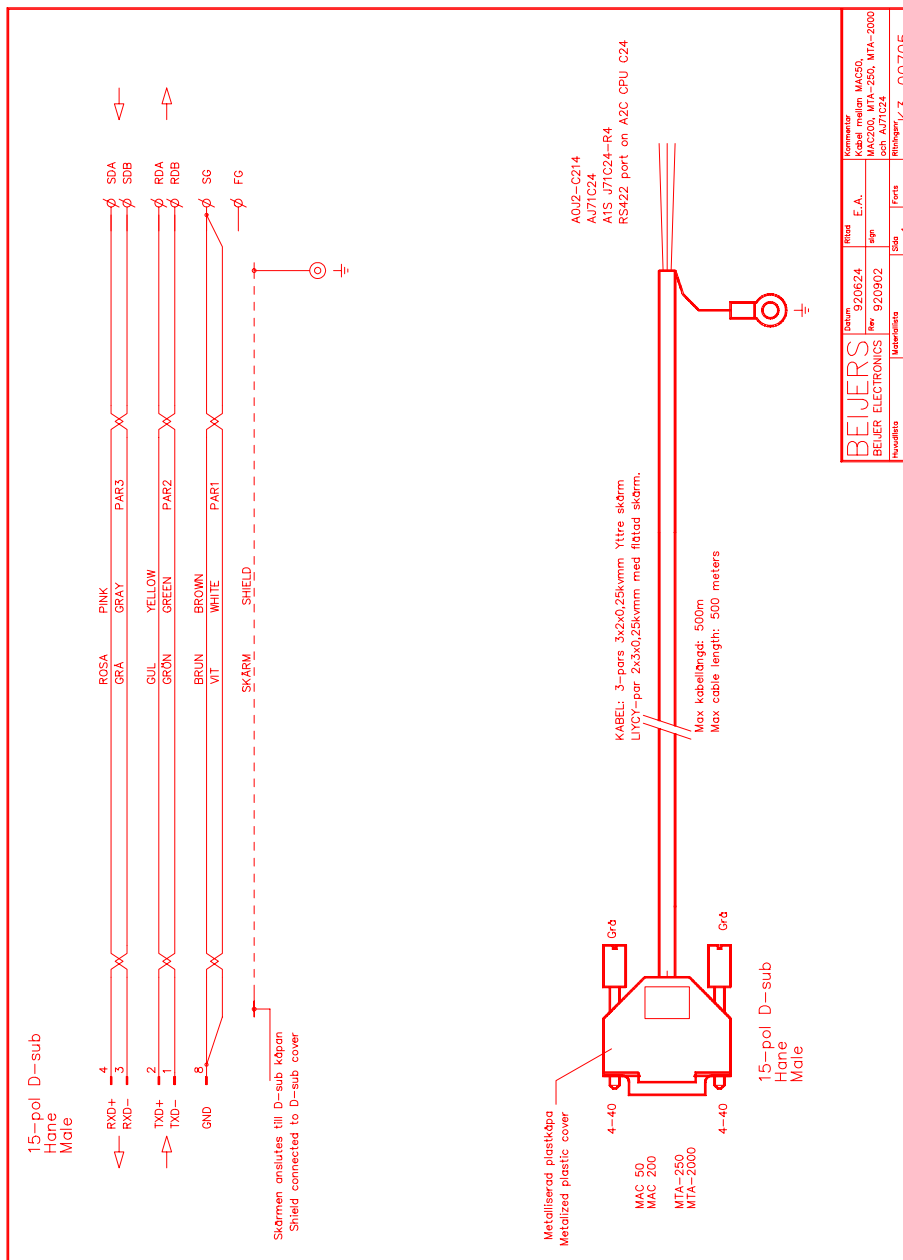
RS-232/RS-485 connection



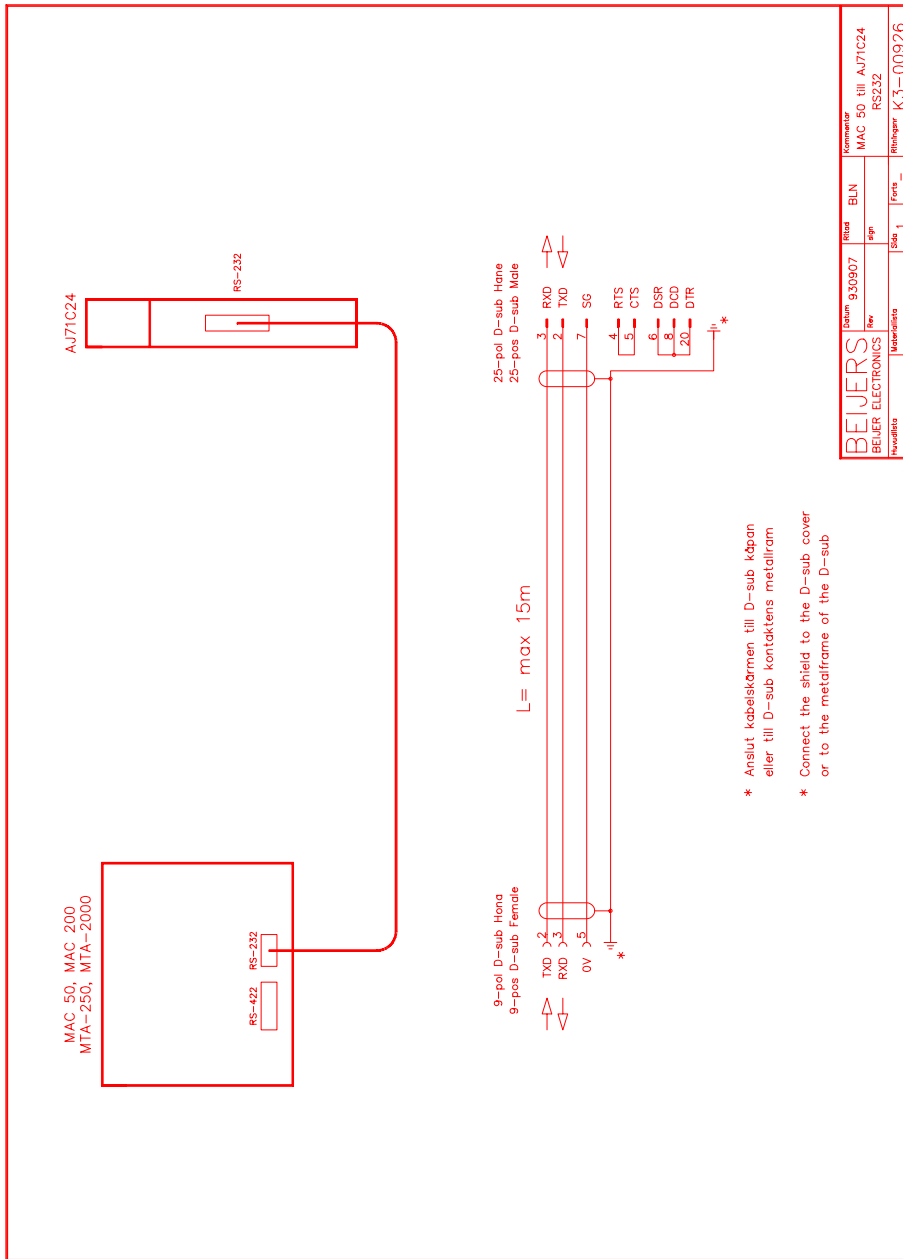
Connection between a terminal and MELSEC CPU



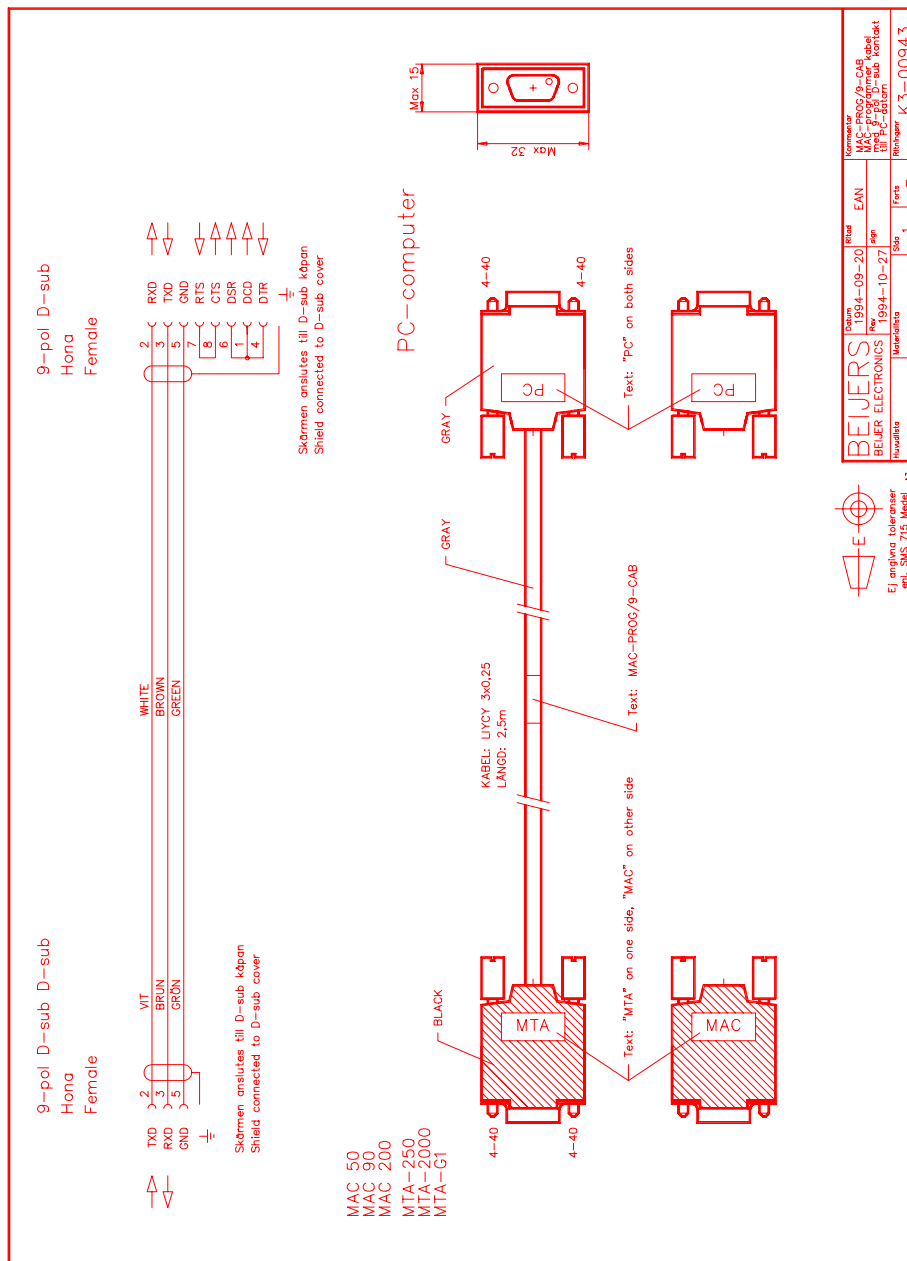
Connection between a terminal and MELSEC FX0



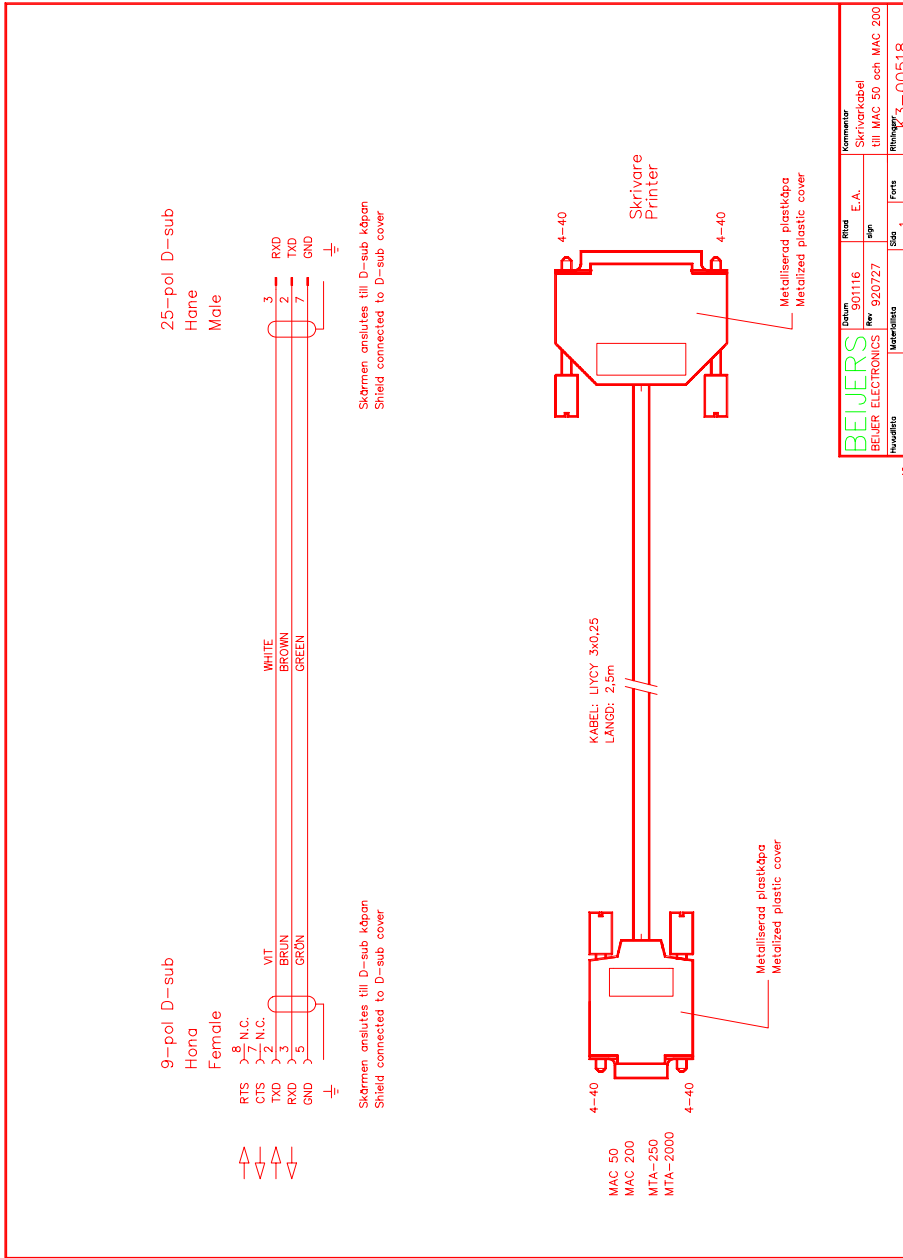
Connection between a terminal and AJ71C24



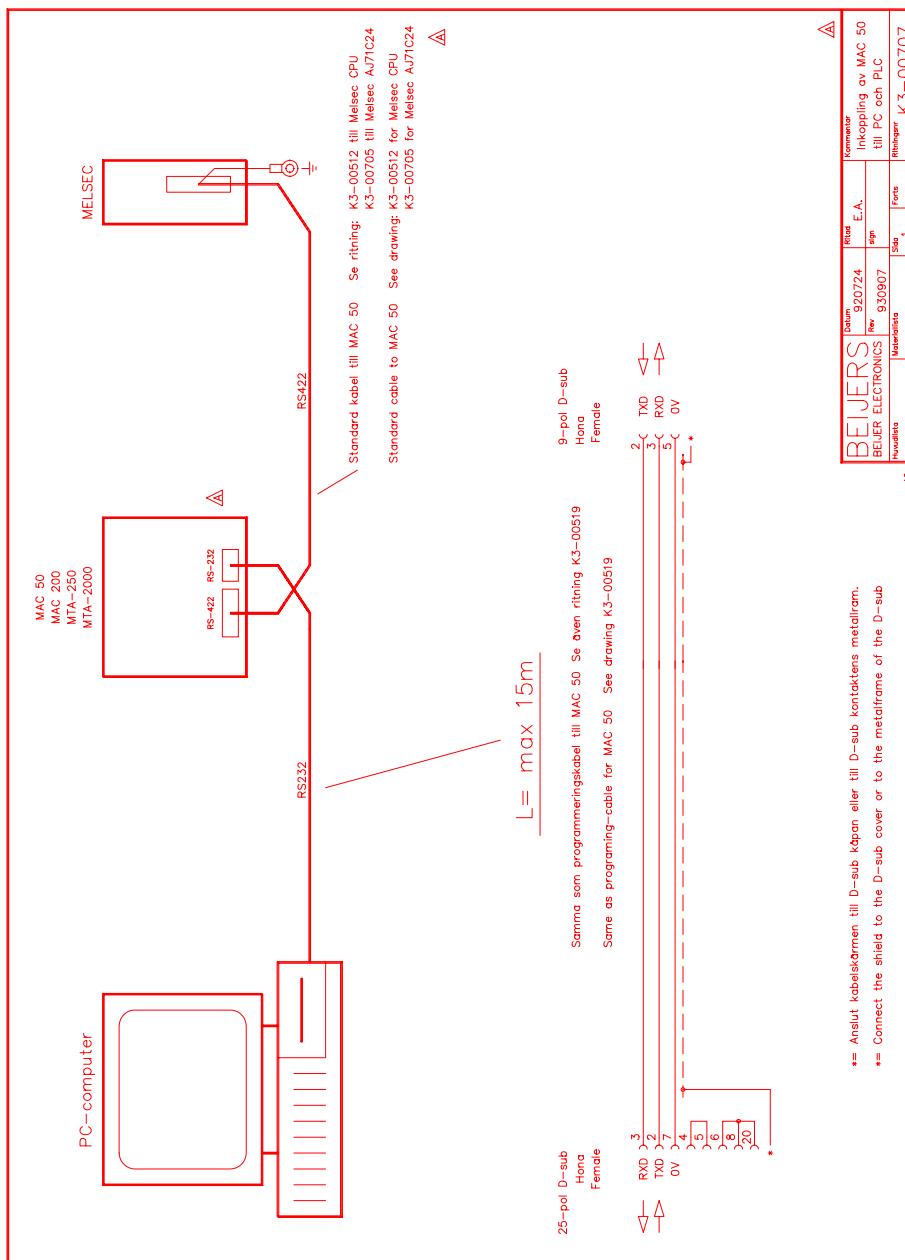
RS232 connection between a terminal and AJ71C24



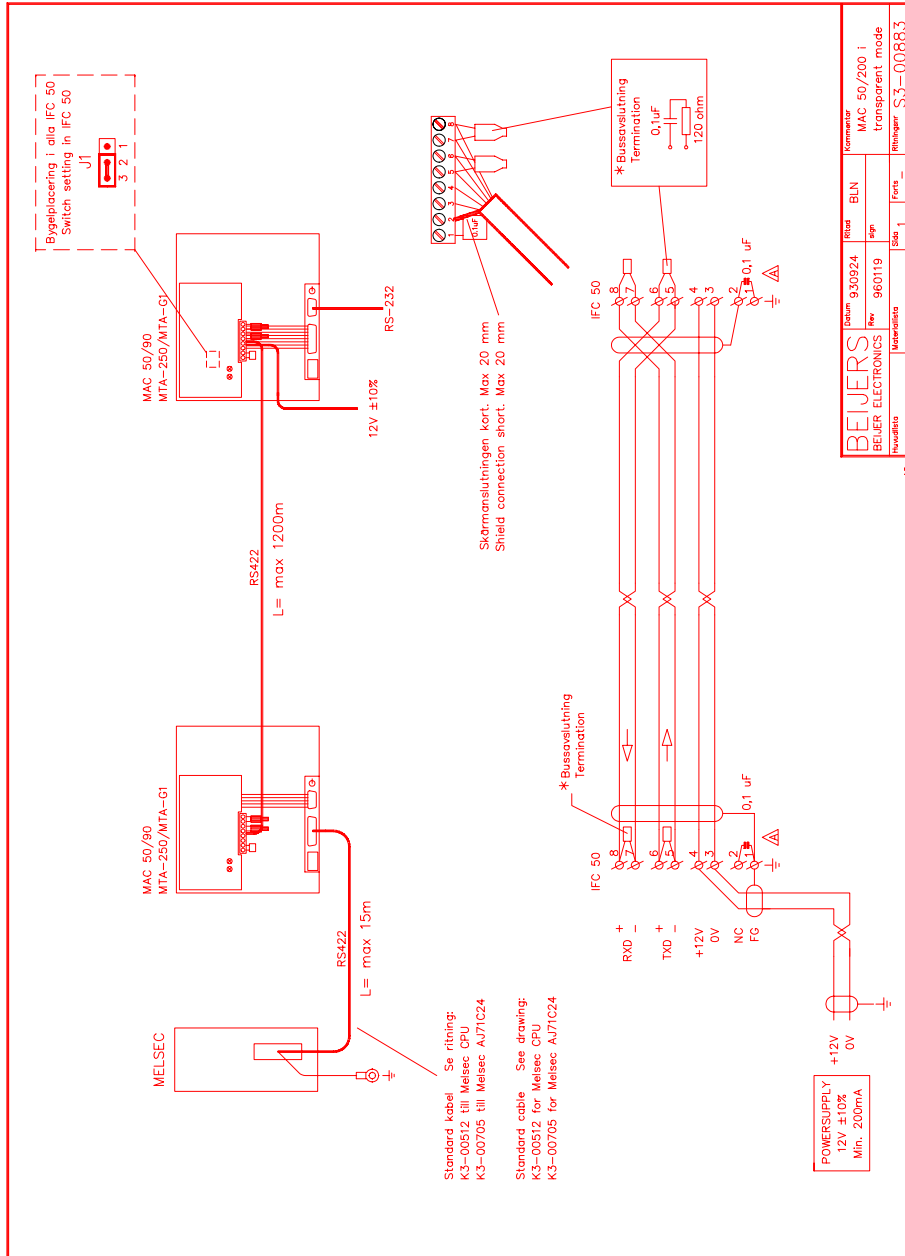
Connection between a terminal and a PC



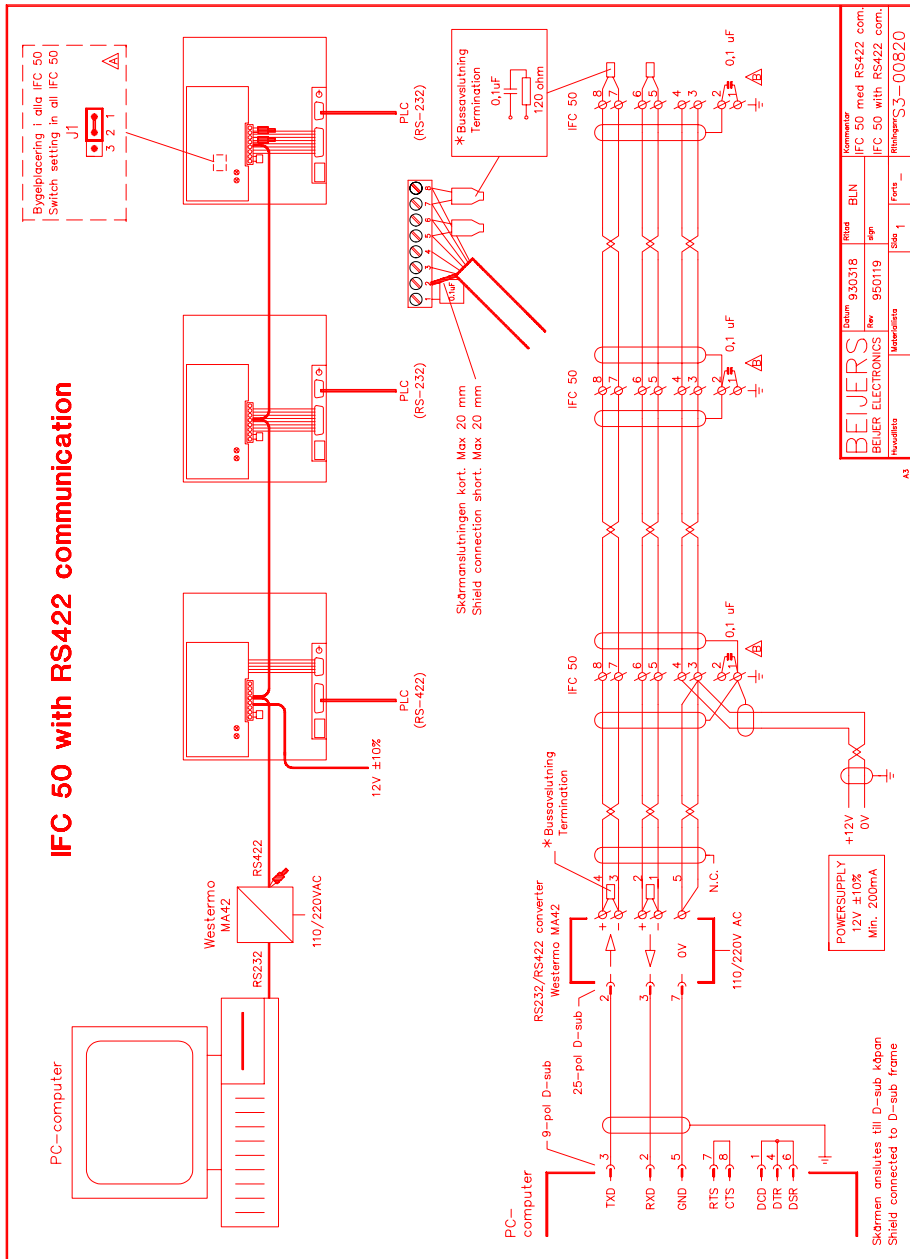
Connection between a terminal and a printer



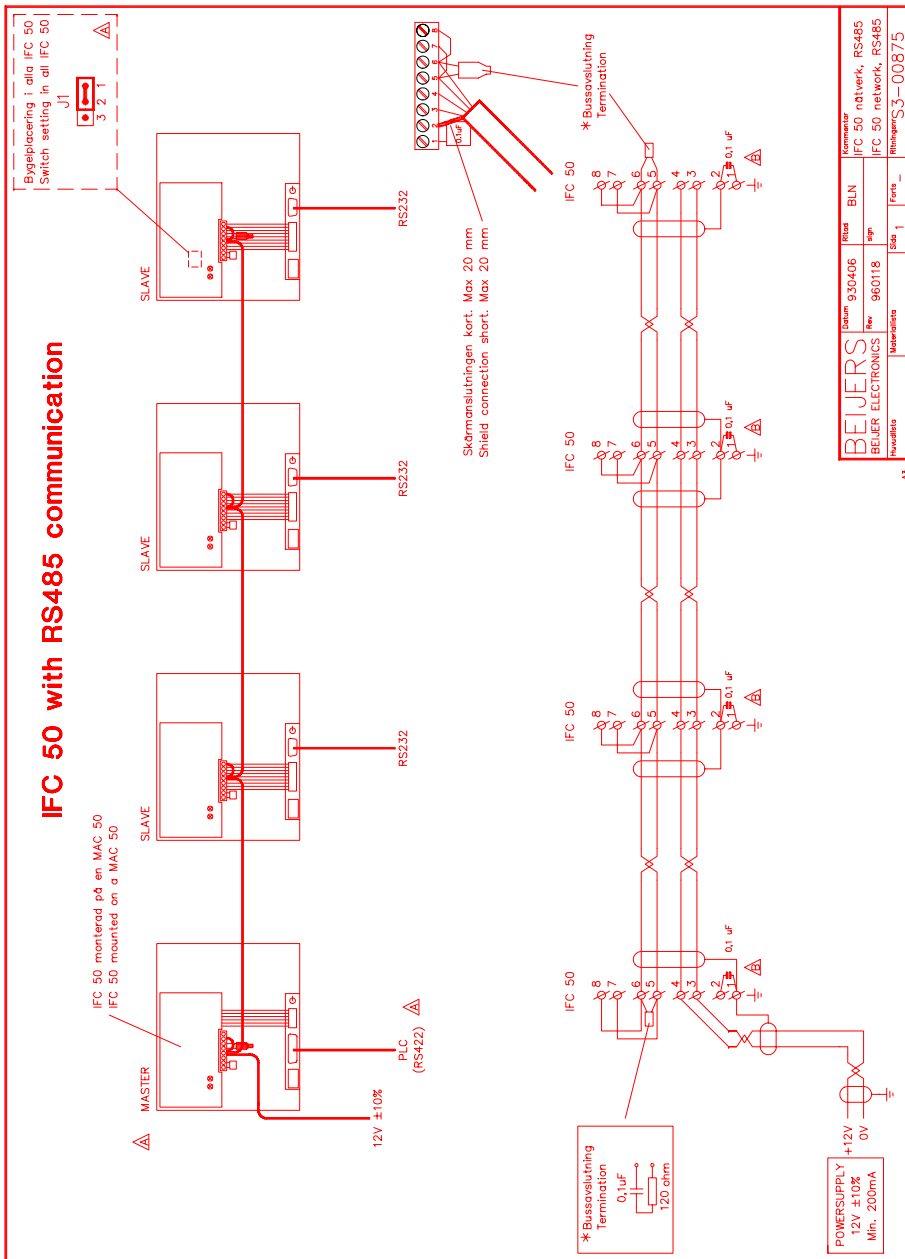
Connecting a terminal to a PC and a PLC system



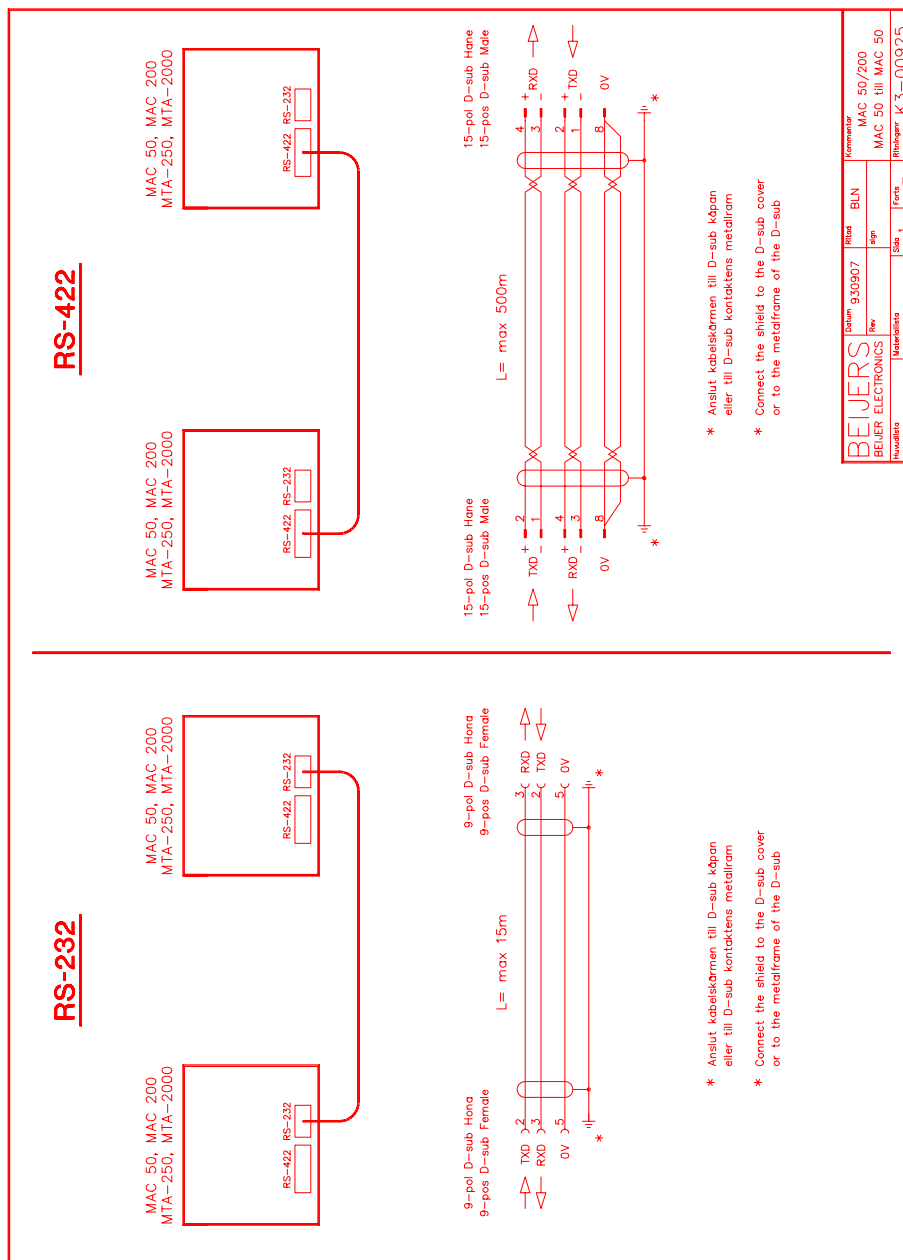
Connection between terminals in transparent mode



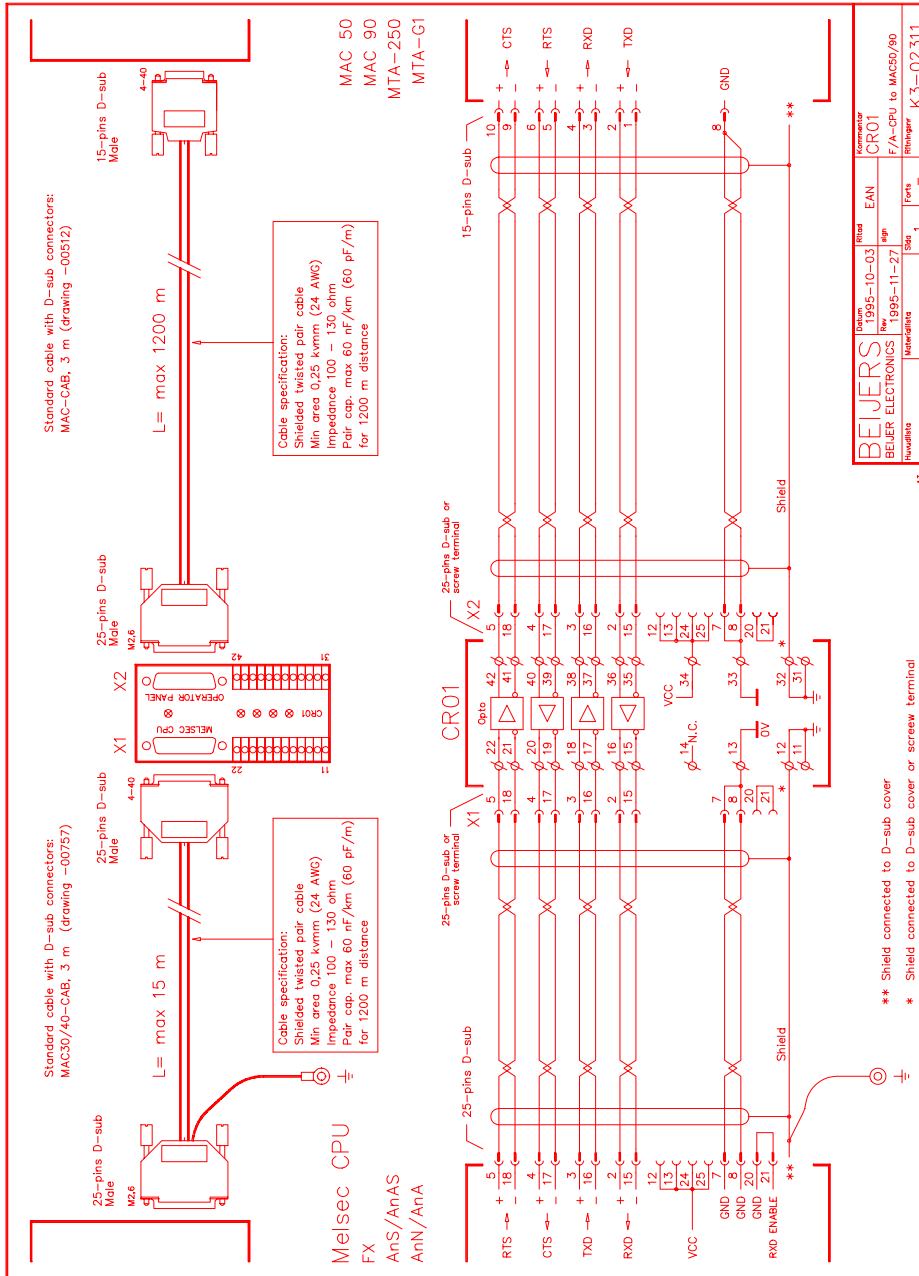
Connecting IFC 50 with RS-232 communication



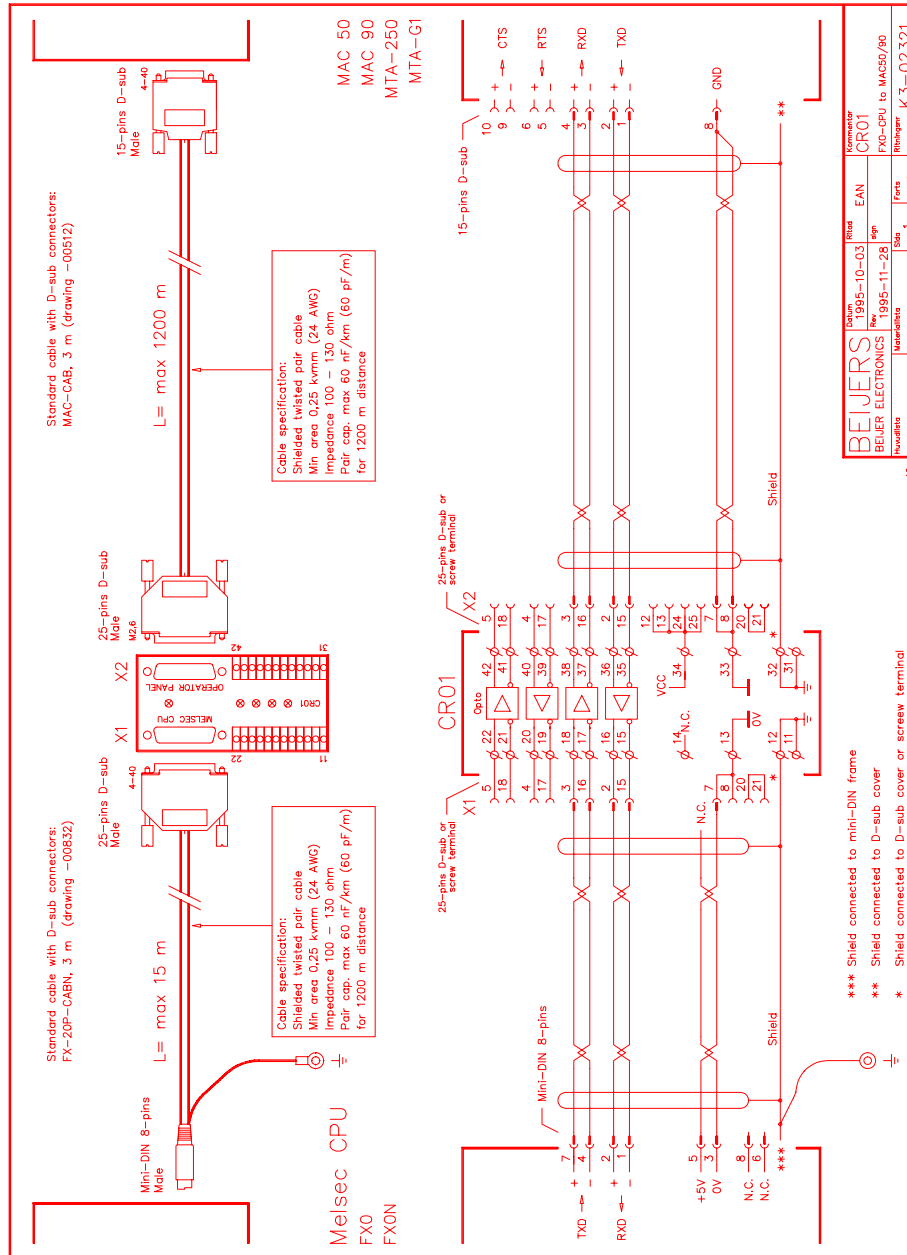
Connecting IFC 50 with RS-485 communication



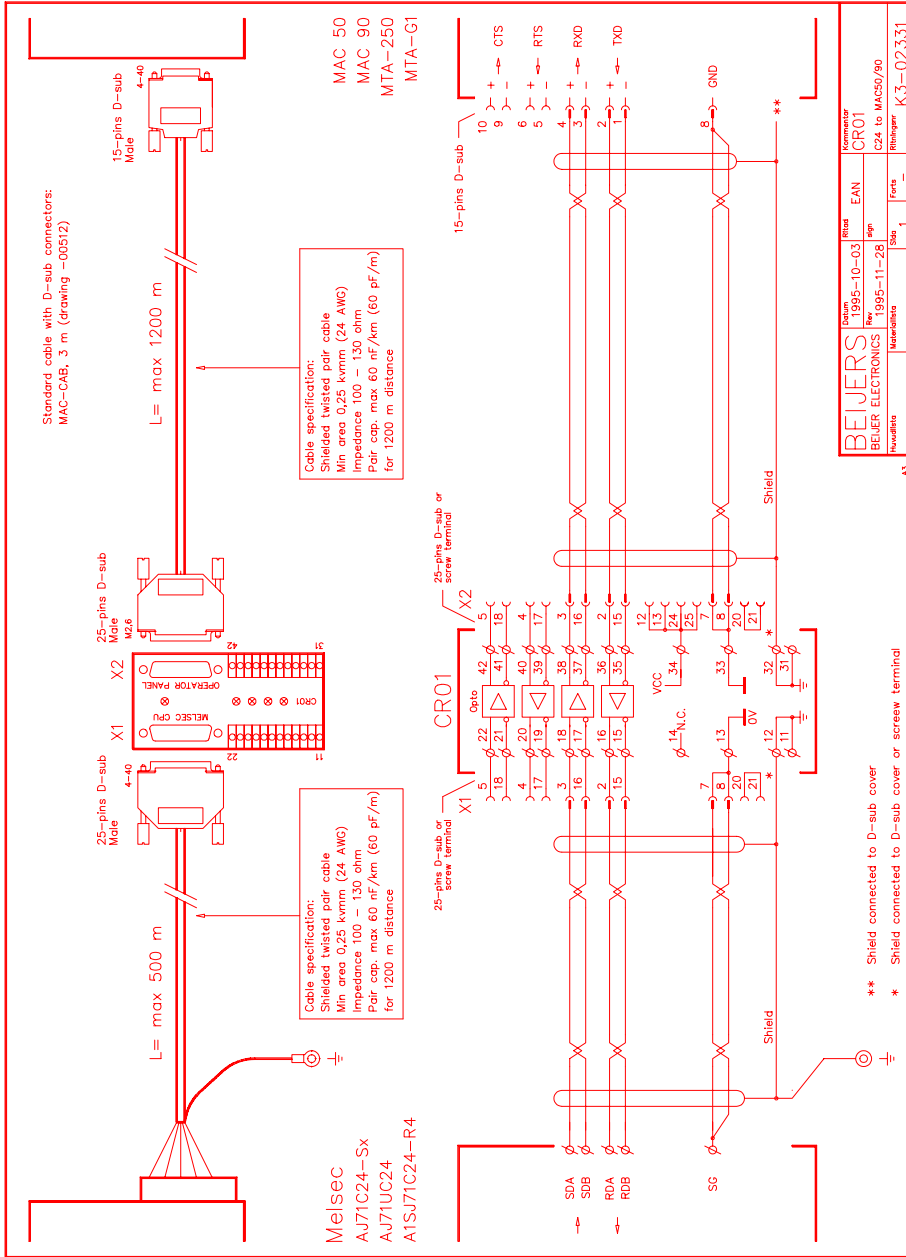
RS-232 or RS-485 connection between terminals



Connecting CR01 to A/F-CPU



Connecting CR01 to FX0-CPU



Connection between a terminal and C24 via CR01

