

# **MAC Operator Terminal**

Human-Machine-Interface

User's Manual

**MAC 40+**



## **MAC 40+/MTA-100**

### **Foreword**

This manual presents installation, handling, and technical details of MAC 40+/MTA-100.

MAC 40+/MTA-100 is used with MELSEC PLC-systems.

Beside this manual, the following 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-series, Programming Manual
- AnU-CPU, User's Manual
- AnA-CPU, User's Manual
- AnN-CPU, User's Manual
- AnS-/AnAS, User's Manual
- MAC Programmer/SW-MTA, Manual
- MAC Programmer+/SW-MTA-WIN, 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 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. Only approved spare parts and accessories manufactured according to specifications of Mitsubishi Electric Europe B.V. should be used.

The equipment must not be used in an explosive environment.

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.

<b>Comment on the manual MA-00014 rev A</b>	<b>Page</b>

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**

# Contents

<b>1 Introduction</b> .....	1
<b>2 Installation</b> .....	3
2.1 Contents of package .....	3
2.2 System requirements .....	3
2.3 Connection of the terminal to PLC system .....	4
2.4 Programming from personal computer .....	8
<b>3 Function overview</b> .....	9
<b>4 How to use the terminal</b> .....	11
4.1 The different modes .....	11
4.2 Keyboard and menu selections .....	12
4.3 Character selections .....	15
4.4 Text and system blocks .....	15
4.5 Example: Water tank .....	18
4.6 Static and dynamic information .....	20
4.7 Jump objects .....	21
4.8 Digital objects .....	22
4.9 Analog objects .....	24
4.10 Bar graph objects .....	25
4.11 Multiple choice object .....	26
4.12 Text object .....	27
4.13 Function keys .....	28

<b>5 Programming example</b> .....	29
5.1 Structuring of text blocks according to the application .....	29
5.2 Function keys .....	30
5.3 Changing displayed text block .....	32
5.4 Setting date and time .....	38
5.5 Password .....	38
5.6 Status of machine .....	40
5.7 Program selection .....	41
5.8 Text from PLC .....	43
5.9 Behavior of display .....	44
5.10 How much can be stored in a project .....	45
<b>6 Reference guide</b> .....	47
6.1 Setup mode .....	47
6.2 Programming mode .....	53
6.3 Dynamic data fields .....	57
6.4 Run-time mode .....	65
<b>7 Summary</b> .....	67
7.1 Modes .....	67
7.2 Objects .....	67
7.3 System blocks .....	67
7.4 Keyboard .....	68
7.5 Character settings .....	69
7.6 International character set .....	70
7.7 Keys .....	71
7.8 Basic set-up .....	74
7.9 Data .....	75
<b>Index</b> .....	I

# **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.



# 1 Introduction

This manual describes the operator terminals MAC 40+ and MTA-100. Hereafter these products are referred to as the terminal.

The terminal is included in a family of terminals developed to satisfy the demands made for human-machine communication.

The terminal makes it possible to show text including static words as well as dynamic values. The texts are organized in text blocks. These text blocks can be ordered free of choice by the programmer to form a structure suitable for the application (e.g. menu trees).

Other functions includes maneuvering and function keys.

The programming of the terminal is very easy and made either directly from the built-in keyboard or from a Personal computer equipped with the PC software MAC Programmer/SW-MTA for DOS or Windows.

Read the chapter *Programming examples* before you start programming. In this chapter you will find a lot of tips.



## **2 Installation**

### **2.1 Contents of package**

The terminal package includes:

- The terminal
- Installation instructions

### **2.2 System requirements**

The terminal works with programmable logical controllers type MELSEC F, F1, F2, FX0, FX0N, FX0S, FX, AnN, AnA, A1S, A2C from Mitsubishi Electric. If you have a PLC system type F-20 or F-40, you will need a system memory F-20GC or F-40GC respectively. For these two types of PLC systems maneuvering of PLC data does not work.

## 2.3 Connection of the terminal to PLC system

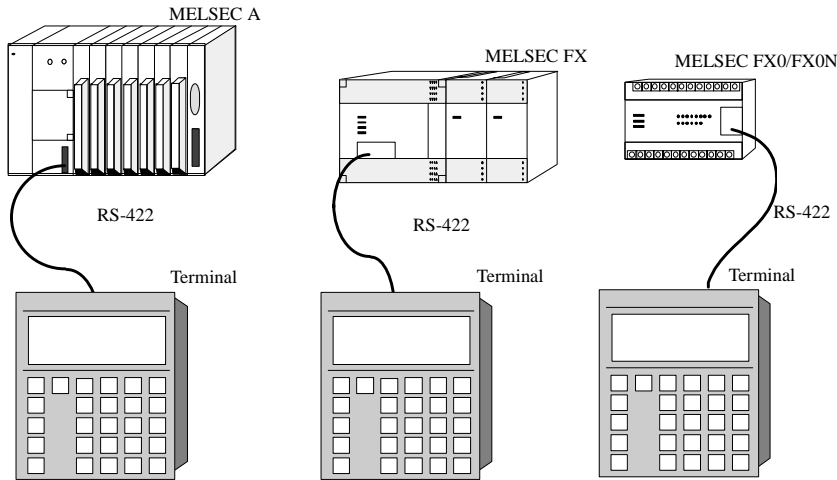
The terminal can be connected to the CPU port, AJ71C24, A1SJ71C24, or the RS-422 port of A2CCPUC24.

### Connection to A or FX CPU port

Use the cable recommended by G & L Beijer Electronics AB to connect to A, FX or FX0 CPU port. The two 25 pin male contacts to the A and FX series are equal. Connect one to the CPU port and the other to the connector on the back of the terminal.

Power for the terminal is supplied from the front port through the cable.

The maximum cable length is three meter. If you need a longer cable you must use a cable in which the power supply voltage is made external (the same cable as you use when the terminal is connected to F1/F2).

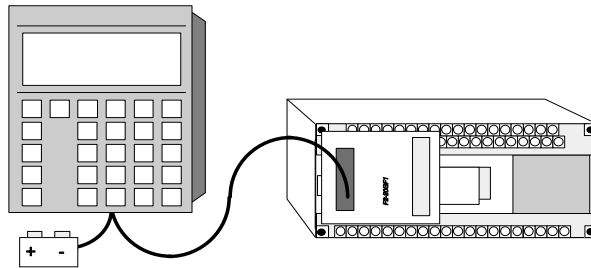


*Connection of the terminal to A, FX, FX0 and FX0N CPU port.*

## Connection to F1/F2

The F1/F2 system must be equipped with a F2-20GF1 interface. The speed should be set to 19200 baud. Power supply voltage +5 VDC is made external.

See the *Installation manual* for cable configuration.

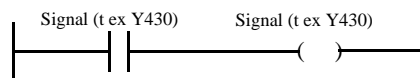


### Connection of the terminal to F1/F2 PLC system.

---

#### Note!

Digital objects in a F/F1/F2 to be controlled from the terminal must exist in the PLC program as an OUT-instruction. At forced on/off, function keys etc, the digital object is only forced for one scan. "The PLC-program takes over", which means that the following circuit is needed in many cases:



## Connection to AJ71C24/AJ71UC

The terminal should be connected to the RS-422 port. Power supply voltage +5 VDC, must be connected from an external power supply.

See the *Installation manual* for cable configuration.

The switches on AJ71C24 should be set like this:

<b>Station number</b>	00
<b>Mode</b>	A
<b>Transmission switches</b>	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.

This means 19200 baud, 8 databits, no parity and 1 stop bit. The parameters in the terminal in the set-up menu under **Port parameters, PLC** must be the same.

## Connection to A1SJ71C24-R4

Power supply voltage +5 VDC is made external.

See the *Installation manual* for cable configuration.

The switches on A1SJ71C24-R4 should be set like this:

<b>Station number</b>	00
<b>Mode</b>	5
<b>Transmission switches</b>	2, 4, 6, 7, 8, 12 ON, remainder OFF

This means 19200 baud, 8 databits, no parity and 1 stop bit. The parameters in the terminal in the set-up menu under **Port parameters, PLC** must be the same.

## Connection to A2CCPUC24

The terminal can be connected to the CPU port or to the RS-422 port.

See the *Installation manual* for cable configuration.

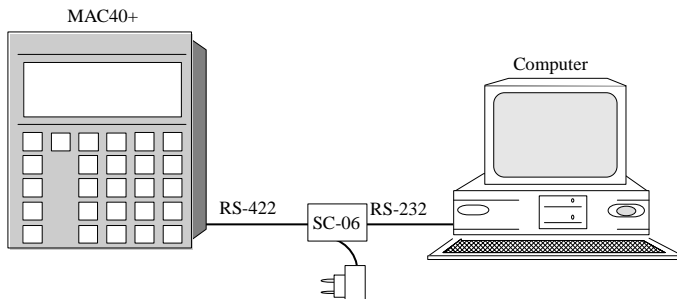
If the terminal is connected to the RS-422 port the switches should be set like this:

<b>Station number</b>	00
<b>Mode</b>	A
<b>Transmission switches</b>	12, 13, 14, 18, 20 ON, remainder OFF

This means 19200 baud, 8 databits, no parity and 1 stop bit. The parameters in the terminal in the setup menu under **Port parameters, PLC** must be the same.

## 2.4 Programming from personal computer

For programming of the terminal it is recommended that the PC software is used. This software will make programming easy and also gives a back-up of the project. To connect a terminal to a PC use the convert unit SC-06N.



*Connection for PC programming with the PC software.*

It is also possible to program directly from the terminal keyboard.



## 3 Function overview

### Basic definitions

#### Text block

Textblock is the basic function of the terminal. They are used to create and structure an application. Text blocks can consist of an unlimited number of lines up to 16 characters long.

#### Static text

Static text is text which is written in text blocks and which cannot be changed during RUN-mode.

#### Dynamic objects

Dynamic objects are those objects which are linked to signals in the PLC system. There are 7 types of dynamic objects: **digital**, **analog**, **jump**, **date/time**, **bar graph**, **multiple choice** and **text object**.

#### Function key

The function keys are connected to a digital signal by typing the signal address for each key. During run-time the digital signal will be on, as long as the key is pressed.

#### LED

The LEDs are connected to a digital signal by typing the signal address for each LED. During run-time the LED will be lit when the digital signal is on.



## 4 How to use the terminal

The terminal is designed according to the object oriented point of view, that is you start from an object and then select the function you want for the object. According to that principle all types of signals are defined in a similar way, which makes it easier for the user.

In this chapter we are going through the basic topics of the terminal and you will get a good introduction in using the terminal. We suggest that you read through the whole chapter while testing the different possibilities described below. For a more detailed description of all functions see the chapter *Reference guide*.

The various picture texts shown in squares in this manual are equal to the text shown on the display. The lines above or below the square represent the current hidden text lines on the display.

We are partly going to work with an example in this chapter. This will make it easier to explain how the different parts of a the terminal application are constructed.

### 4.1 The different modes

The terminal has three different modes:

- Setup mode
- Programming mode
- Run-time mode

In Setup mode you make the basic settings for e. g. Selection of menu language, Character set, Format for presentation of time/date. The settings you make here are valid until a new one is made.

Programming mode is used for building the application. This includes the text blocks, function key assignments and the LED assignments.





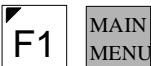



In the third and last mode the application is monitored. You can see how the different objects change and also maneuver some objects.

Press [MAIN MENU] [INFO] to change between programming mode and run-time mode. In programming mode setup mode is selected by pressing any key while the initial picture is displayed.

## 4.2 Keyboard and menu selections

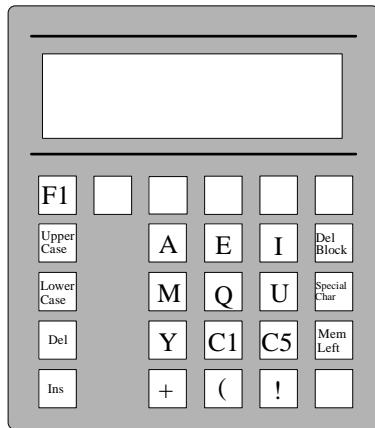
The terminal has a small and very compact keyboard. Each key has several functions which can be used by combining the keys in different ways. To enter values, letters, and other characters use the right side of the keyboard together with the function keys [F1], [F2], [F3], and [F4]. On each numeric key and on the keys [-] and [.] there are four other characters, one in each corner. To type one of these characters use some of the function keys. [F1] for characters in the top left corner, [F2] for characters in the top right corner, [F3] in the bottom left and finally [F4] in the bottom right corner. Press the function key and keep it down while pressing the wanted key.

The codes C1-C8 on key [2] and [3] symbolise character codes with domestic characters. In Swedish language this means that C1 stands for Å, C2 for Ä, and finally C3 for Ö. See the chapter *Summary* for further details. The function key [F1] is also used for selecting which “kind” of character to type. [F1] can be combined with the following keys:

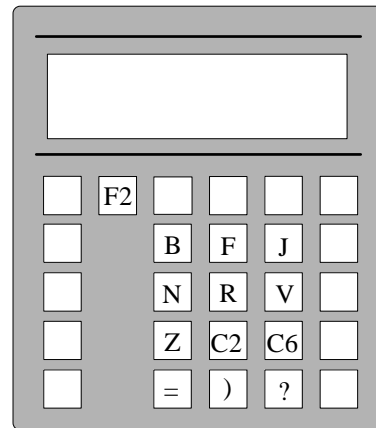
Key	Function
	Uppercase letters (Capital letters)
	Lowercase letters (Small letters)
	Insert of blanks
	Erase character
	Free memory
	Admits special characters
	Erase text block in Block list menu
	Key board layout

To use any of the above combinations first press the function key, just like typing characters, and keep it down while pressing the other key. When selecting uppercase or lowercase letters the selected status remains until a new status is chosen.

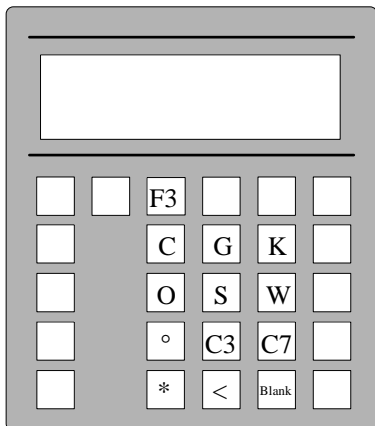
The picture below shows the functions of the different keys when pressing a function key.



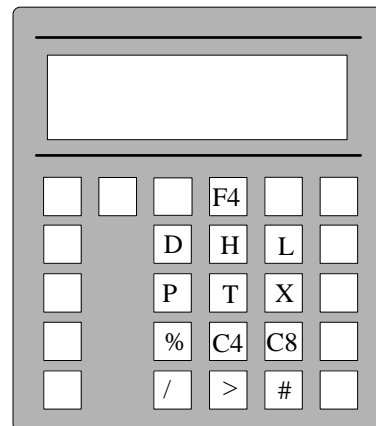
*Keyboard functions when pressing F1.*



*Keyboard functions when pressing F2.*



*Keyboard functions when pressing F3.*



*Keyboard functions when pressing F4.*

As an example we want to type the text "MAC". To do so, press F1 and keep it down while pressing [4]. The M is in the top left corner of this key. You will get the next letter by pressing [F1] again and after that [7]. (The A in top left corner). You also find the letter C on key [7], but this time in bottom left corner, and so you first press [F3] and then [7].

Hereafter we will not explain which one of the function or numeric keys you should use to get a certain character. Only the character is defined.

**[LEAVE]**, (key [F5]), **[NEXT]** ([F6]), **[MAIN MENU]**, **[CLEAR]**, **[INFO]** and **[ENTER]** are used in all modes to selecting from menus, editing text and so on.

The arrow keys on the left part of the keyboard are used for moving the cursor on the display. If you have a menu or a text block containing more lines than the current ones on the display, e. g. more than two, you can scroll the text, (change the current lines) by using [↑] or [↓].

You find the function keys F1-F6 on the top line of the keyboard. They can be connected to a function, selected by the programmer. The information key with the I-symbol is used for supplying information of different objects, e. g. in Run-time mode. To eliminate the risk of mixing it with the letter I, we will type it as [INFO] in the manual.

The entire terminal is built up of a menu system which makes it easy for the user to type and monitor his system. The following general rules are applicable for selections from menus: To select a function you first have to point it out. Use the arrow keys to move the cursor to the line you want. Then press [NEXT]. If there are more alternatives than those on the display, you can scroll the text by using the arrow keys. To leave a menu press [LEAVE].

### 4.3 Character selections

In most languages there are specific domestic characters, for instance Ü, ß in German or Å, Ä, Ö in Swedish. By using the menu selection Character set you can choose the set of characters you want to be equal to the characters C1-C8. After selecting Character set in the Setup menu, the following menu will appear:

```
>Swedish
  German

French
Spanish
Italian
Nor/Dan
```

*Character selection menu with Swedish as current language.*

To change character set, move the cursor to appropriate line and press [LEAVE].

### 4.4 Text and system blocks

The main function in the terminal is named text blocks. These are used for building up the applications. The text blocks can be built up as menus. Between these menus you can define jumps. They can also show state and value of different signals.

A text block contains an unlimited number of lines, each one with 16 positions for characters. The text blocks are created in Programming mode. To make a new text block, follow the instructions below. In **Programming mode** menu select the alternative **Edit**, by moving the cursor to Edit and then press [NEXT].

```
>Edit
  Transfer

Block list
```

*Programming menu with cursor on Edit.*

When selecting Edit, the Edit menu will appear with all the editing alternatives.

```
>Text block
Function keys
```

LED

*Edit menu.*

Now select **Text Blocks**. By pressing [NEXT], the text block menu appears.

```
BLOCK NUMBER : _
Disp sign    :
```

```
Statusword   :
```

*Text block menu.*

Each text block has a number as its identification. The first block, the one that is displayed when starting up the program, is always number **0**. This block is also called "Main menu". The following blocks are numbered from 1 to 999.

Some of the block numbers are reserved for special purposes. These numbers are **998** - LED-indicator overview page and **999** - Function key overview page. See each chapter for further information. These "reserved" blocks are called system blocks as they have special functions in the terminal.

After typing the number of your text block you can also choose **Disp sign:** and **Statusword:**.

**Disp sign:** defines a digital signal which in ON position activates the text block to be current on the display

To type this signal place the cursor on the position after the text. This will happen automatically when pressing [Enter] after typing the block number. When the cursor is in the correct position, type in the signal address. After input press [Enter].

**Statusword:** defines the character of the display in run-time mode. See the reference guide for further information.

Neither **Disp sign:** or **Statusword:** need to be defined.



After making the wanted setups you can start editing the text block. Press [NEXT].

The display will then be cleared and you can start typing the text. The cursor will move on the current line while typing. When the maximum number of characters have been typed, the cursor will automatically move to the beginning of next line. If you want a line shorter than maximum length, press [ENTER]. If the text block will contain more than two lines, these are scrolled during typing.

```
This is an examp  
le of a text blo  
ck with more tha  
n two lines wher  
e you haven't de  
finied the retur  
ns
```

*Example of a text block.*

If you want to make changes in your text block, move the cursor to appropriate position and make the changes by using the functions described under *Keyboard and menu selections* in this chapter.

---

**Note!**

You must specifically define if you want to put a character position into the text. Otherwise you will overwrite the current characters.

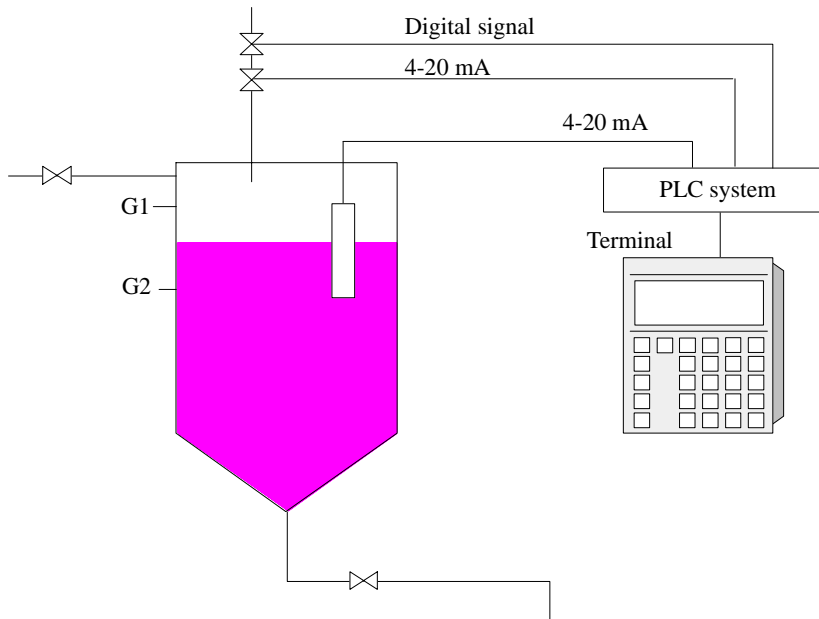
---

## 4.5 Example: Water tank

We will take our example from the paper industry. The part of the process we are illustrating controls a water tank in a closed system. In this tank the PH value is controlled and regulated.

Except the PH-handling also the fluid level of the tank is controlled. This prevents it from getting out of the permitted interval.

In the drawing below there is a model of a water tank with the PH-regulation shown.



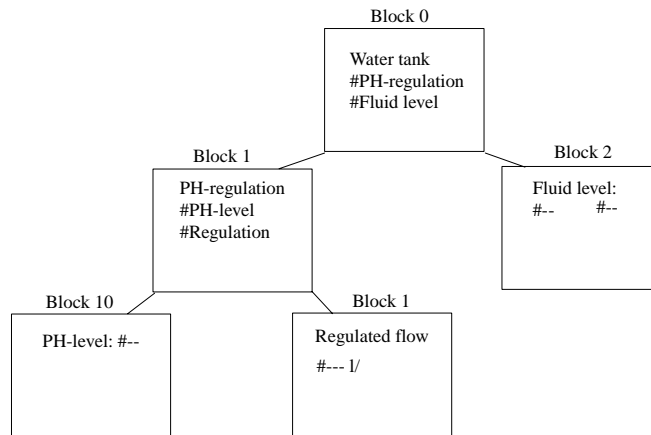
### *The water tank.*

Our application will be built of text blocks, organized in a tree structure. We will divide the example into smaller parts to make it easier to follow the application.

When numbering the text blocks, we recommend that you choose a principle that makes it easy to follow the structure of the application. From the example in the picture below you find that all text blocks subordinated to a certain text block have inherited the first figure of the superior block.

For example, the text blocks emerging from text block one have got the numbers 10 and 11, where the first figure 1 comes from block 1 and the following zero respectively one is an ordinal number. The disadvantage of this procedure is the fact that you can have maximum four levels including block zero.

In our example the application would be built up in the following way.



### *Controlling of water tank built up in a tree structure.*

Text block number zero and number one work as menus in Run-time mode. From here you can jump to text blocks below. From block number zero you can jump to block one and two. From block number one you can jump to block ten and eleven. Block two, ten, and eleven contain objects showing states and values of signals in the PLC system. Text block number two shows the state of the two transmitters, G1 and G2, giving min- and max level for the fluid in the tank. Each one of the text blocks ten and eleven contains an object showing the analog value for the PH-level respectively the flow of the acid fluid.

## 4.6 Static and dynamic information

In the text blocks you work with two types of information:

- Static text
- Dynamic data fields

The static text do not change during execution of the program. They will always appear in the same way. Going on with our example we can figure the following main menu:

```
WATER TANK
# PH-regulation
```

```
# Fluid level
```

*Main menu for controlling the water tank.*

In this menu all text is static, i.e. it will not be changed due to any signal from the PLC system.

The character # defines a dynamic field of one of the following kinds:

- Digital
- Analog
- Jump
- Date/time
- Bar graph
- Multiple choice
- Text

To create a dynamic field, press # (found in the bottom right corner of the key [.]). Now a menu for selection of type of dynamic field (pop-up menu) will occur on the display.

```
|>Digital      |
| Analog      |
|             |
| Jump        |
| Date/Time   |
| Bar graph   |
| Mult.choice |
| Text        |
```

Position the cursor on the appropriate line and press [NEXT]. Then a menu for definition of type of selected dynamic field will appear. Each menu is specific for its type of object.

## 4.7 Jump objects

In the main menu of our example we want to make a jump object before the text PH-regulation. Type a #, move the cursor to Jump, and press [NEXT]. The following menu will then appear:

```
| Cur.block: 0 |
| Jump to :_   |
```

*Jump object menu.*

**Cur.block:** Defines the number of the block you are leaving. At **Jump to:** you enter the block number you want to jump to, which is block 1 in our example. In Run-time mode a text block with jump objects will look like this:

```
Water tank
→-regulation

→Fluid level
```

*Main menu for controlling of water tank in Run-time mode.*

Then on selecting a Jump object in Run-time mode and pressing [ENTER], a jump will be generated to current block and that block will appear on the display.

## 4.8 Digital objects

In our example we find several digital objects. For example the transmitter, G1, stating if the maximum fluid level in the tank has been reached, is a digital type of object.

```
Fluid level:
#--  #--
```

*Text block for controlling the fluid level.*

A digital object is defined by selecting **Digital** from the menu after pressing #. Then the menu for digital objects will occur.

```
| D.Obj  :_  |
| Text 0 :   |
|
| Text 1 :   |
| Maneuver:NO |
```

*The digital object menu.*

First type the signal the object should be connected to: M, X,Y, S,T, C. For Timers or Counters we are referring to the status. Our transmitter is connected to input X0. Type its address and finish by pressing [ENTER].

At **Text 0:** and **Text 1:** you type the texts you want to appear when the signal is ON/OFF.

---

**Note!**

There must be as many free positions on the line of your text block as characters in the longest text.

---

For this transmitter we choose the texts below, e. g. an empty string for input 0 and the text **MA** for input 1. The last alternative in the menu **Maneuver.**, meaning you can change the object's value during operation from the terminal, is not described here. For further information see *Reference guide*.

	D.Obj	:X0	
	Text 0	:	

	Text 1	:MAX	
	Maneuver	:NO	

*The digital object menu.*

During run-time the text corresponding to the current state will be displayed.

## 4.9 Analog objects

The analog objects on the other hand represent analog values. Their menu will have the following appearance:

A.Obj	:_	
Pos	:5	
Dec	:0	
Maneuver	:NO	
Min	:-3276	
Max	:32767	
Off	:0	
Gain	:1	

*The analog object menu.*

The signals which can be addressed from here are type D, T, C, W, R. You will find a complete list for each system in the *Reference guide*. Just like Digital objects you start by typing the address of the signal after the text **A.Obj**:. After that define the number of positions the object will need on the display. This is done after the text **Pos**:. This value should also include decimal points, if any. The value 10.05, for instance, takes five positions, four positions for the figures and one for the decimal point. **Dec**: defines to the system how many decimals should be presented.

**Min**: and **Max**: specify the smallest/largest value possible to input for the signal. Sometimes you may want to scale the input value. Then do as follows: After **Off**: State a figure to be added to each value and after **Gain**: a figure to be multiplied by the input value. The general equation for the displayed value is:

$$DisplayValue = Offset + Gain * SignalAddressValue$$

In Run Time-mode the analog value will be displayed after being calculated by above equation.



As an example take text block number 10. When the actual PH-level is defined, the definition of our analog object will look as follows:

A.Obj	:D0	
Pos	:3	
Dec	:1	
Maneuver	:NO	
Min	:5	
Max	:7	
Off	:0	
Gain	:0.1	

*Analog object showing PH-level.*

In this case we want the value to be displayed with one decimal.

## 4.10 Bar graph objects

Bar graph objects are defined in a way corresponding to the analog objects. They represent the same type of analog values but this time as a bar graph.

The same object as above will look like this in bar graph:

A.Obj	:D0	
Pos	:4	
Dir	:RIGHT	
Min	:5	
Max	:7	
Off	:0	
Gain	:0.1	

*Bar graph showing PH-level.*

The bar will be displayed horizontally. When the PH-value is seven, it will be four positions and when it is five, zero positions.

## 4.11 Multiple choice object

A multiple choice object is connected to three digital signals that together can adopt eight different status. To each status you can connect a text obtaining seven characters.

Assume that we have an elevator and we want to indicate if it is going upwards, downwards or has stopped. Y20 is on when the elevator is going upwards, Y21 when it is going downwards.

The definition of the block will look like this:

D.Obj1 :Y20	
D.Obj2 :Y21	
D.Obj3 :Y21	
Txt000 :STOPPED	
Man000 :NO	
Txt001 :	
Man001 :NO	
Txt010 :DOWN	
Man010 :NO	
Txt011 :	
Man011 :NO	
Txt100 :UP	
Man100 :NO	
Txt101 :	
Man101 :NO	
Txt110 :	
Man110 :NO	
Txt111 :	
Man111 :NO	

*Multiple choice object menu.*

See the *Reference guide* for further information.

## 4.12 Text object

In a textblock it is possible to present texts stored in the data registers of the CPU. The menu looks like this:

T.Obj	:_	
Pos	:8	
Maneuver:NO		

*Text object menu.*

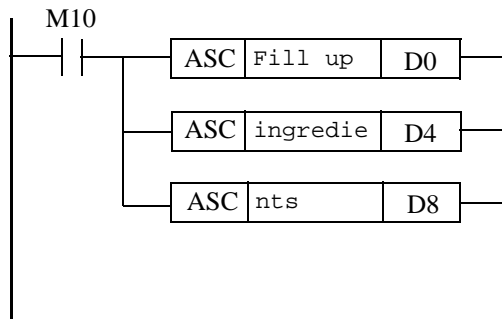
At **T.Obj**: the register in which the first position is stored is defined.

At **Pos**: the number of positions the text will use on the display is defined.

Assume that we have a plant for fruit-syrup making. On the terminal we desire to show which process step that is activated. The process steps can for example be “Fill up ingredients”, “Heating”, “Boiling”, “Hill upp bottles”. The text object will have this configuration:

T.Obj	:D0	
Pos	:24	
Maneuver:NO		

Store the texts in data register by using the ASC instruction. The first text will look like this:



M10 is connected to the process step.

The other process steps are connected to digital signals in the same way, and with the ASC-instruction the correct texts are stored in D0-D11.

The F1/F2 series do not include any instruction compatible with the ASC-instruction in A- and FX-series.

In run-time mode the text stored in D0-D11 will be displayed. The texts in D0-D11 is changed by the PLC-program by activating the signals connected to the process steps.

## 4.13 Function keys

The function keys F1-F6 on the top of the keyboard can be connected to digital signals, which will be ON as long as the function key is pressed. To connect digital signals to the function keys, select **Function keys** from the Edit menu.

```
Text block
>Function keys
```

LED

*Edit menu with cursor on Function keys.*

Function key menu will then appear on the display.

```
F1  : _
F2  :
```

```
F3  :
F4  :
F5  :
F6  :
```

*Function key menu.*

Now the digital signals, controlled by the function keys, are to be defined. After typing one signal, finish the line by pressing [ENTER]. Leave the menu by pressing [LEAVE].

During run-time the digital signals can be forced by pressing the function keys.

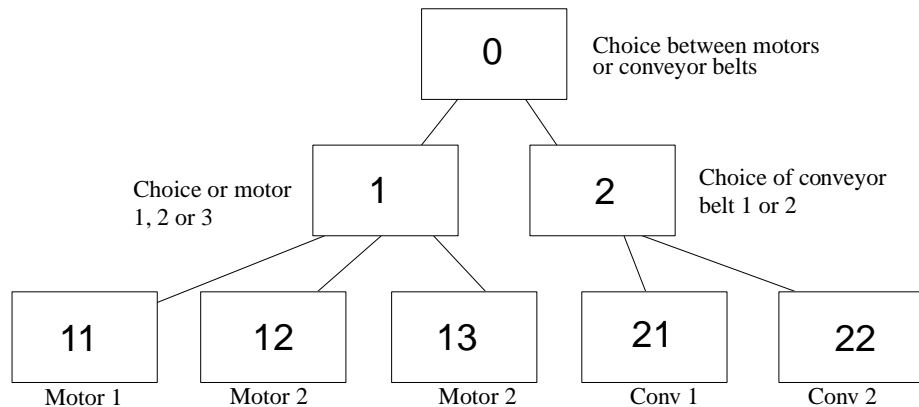
## 5 Programming example

When planning an application there are a number of questions to take into consideration, for instance: Which blocks should you use? How do you combine blocks? How do you use the function keys?

Naturally there are questions of various kinds and we will try to answer some of them.

### 5.1 Structuring of text blocks according to the application

Signals belonging to the same object are normally shown together. Suppose we have an application with 3 motors and 2 conveyor belts. The following menu tree can then be made for the blocks:



Block 12 would for instance look as follows:

MOTOR 2 IS ON	-Digital value
MOMENT :47 NM	-Analog value

---

**Note!**

That all signals belonging to motor 2 (one object) are shown in the same block.

---

## 5.2 Function keys

In the edit menu the function keys are connected to digital signals. The digital signal is ON as long as the function key is pressed.

---

**Note!**

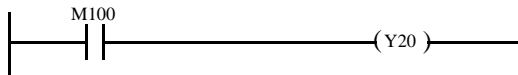
The PLC program has higher priority than forced setting with a function key. If the digital signal exists as an OUT instruction in the PLC-program and the condition before is OFF the signal remains OFF.

---

### The function keys have the same function everywhere

#### MELSEC A and FX

If you want Y20 to be ON as long as a function key is pressed it can for instance be solved with the program below. F1 is connected to M100.



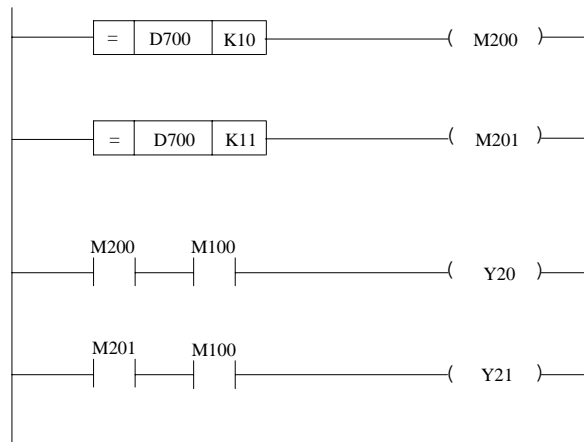
## Different functions of the same function key

You can get the same function key to activate different digital signals depending on which text block you display.

### MELSEC A

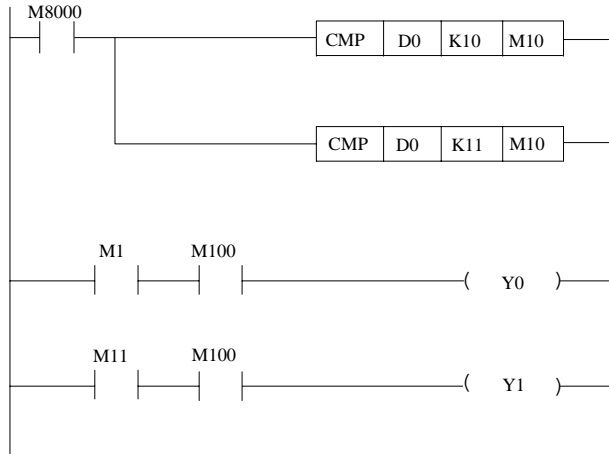
Displaying block 10 we want Y20 to be ON, while pressing F1.

In block 11, pressing F1 we want Y21 to be ON. In system signals we select D700 to store the number of the block which is displayed. We connect F1 to M100.



### MELSEC FX

In text block 10, when pressing F1, we want Y0 to be ON. When pressing F1 in block 11 we want Y1 to be ON. In system signals we select D0 to store the number of the block which is displayed. We connect F1 to M100.



## 5.3 Changing displayed text block

In Run-time mode you can always move to the main menu (block 0) by pressing [MAIN MENU]. The possibilities to move between other blocks in Run-time mode are defined by the programming of the terminal, possibly together with the PLC-program.

---

**Note!**

There can be several possibilities of moving to the same block.

---

### Block changing by jumps

Make a jump object in the block, jumping to another block. This is completely independent of your PLC-program.



## Block changing controlled by PLC program

When selecting a text block you define the digital signal that, when activated, is showing the text block on the display.

We type that text block 5 shall be displayed when X0 is activated.

```
BLOCK NUMBER:5
Disp sign.  :X0
```

Statusword:

*Textblock menu.*

## Block changing by function keys

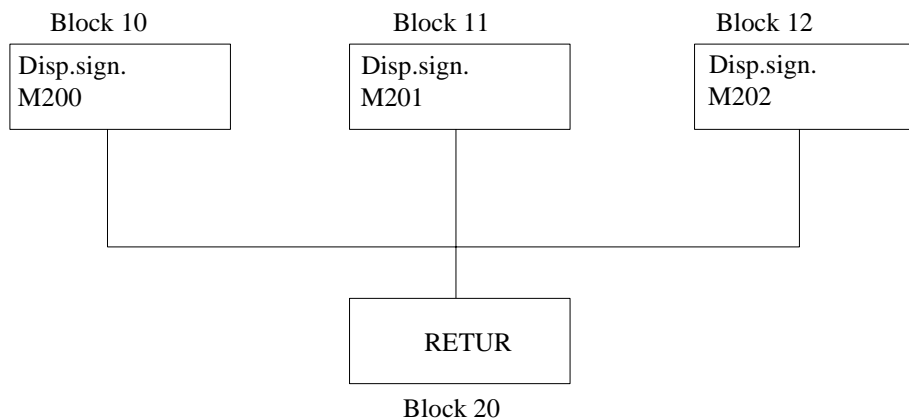
If you want to move to block 10 when pressing F1, independent of the block displayed, follow the instructions below

Connect **F1** to M100.

Define M100 as **Display signal** for block 10.

## Block changing with return to previous block

Displaying in block 20 we want to jump back to the block (10, 11, or 12) where we were before jumping to block 20.



## *Programming example*

We use M200-M202 as Display signal for block 10-12. In block 20 we make a digital object:

D.Obj	:M300
Text 0	:RETURN

Text 1 :RETURN  
Maneuver :YES

*Digital object menu.*

To move to block 20 for instance, we make jump objects in block 10-12.

### MELSEC A

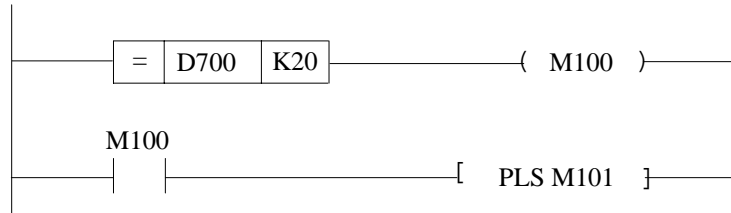
The PLC program looks as follows:

D700 = current block

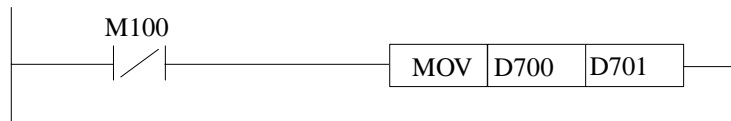
D701 = the block displayed before block 20



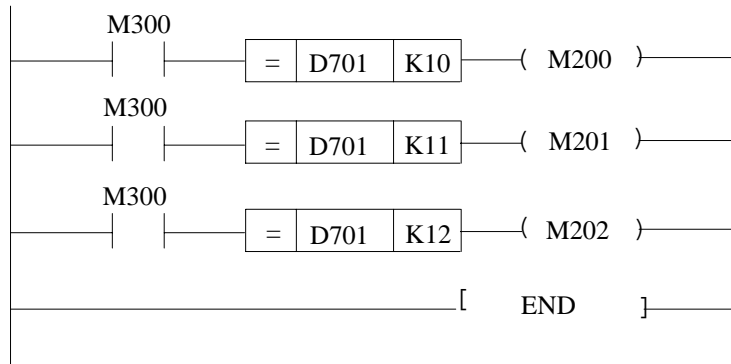
Check if D700 = 20



Check if D700 = 20.



When M300 is activated we compare D701 with K10-K12 to know which block to jump back to.



## Programming example

### MELSEC FX

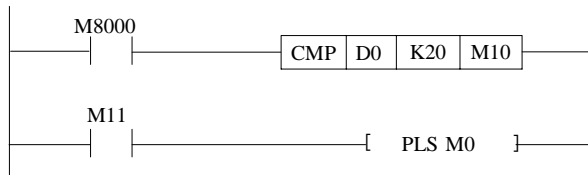
The PLC-program looks as follows:

D0 = current block

D1 = the block displayed before block 20



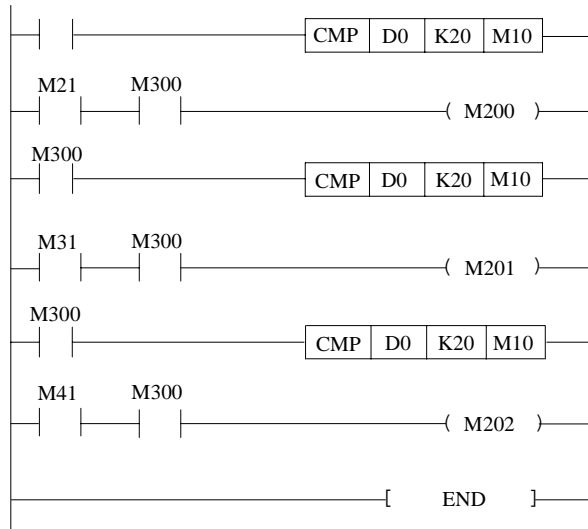
Check if D0=20.



Transfer D0 to D1.



When M300 is activated we compare D1 with K10-K12 to know which block to jump back to.



## Block changing by typing blocknumber

Displaying block 0 we want to type into an analog object which block to change to (block 30, 31 .....39). If we type 30 in the analog object we change to block 30 and so on. Create an analog object in block 0:

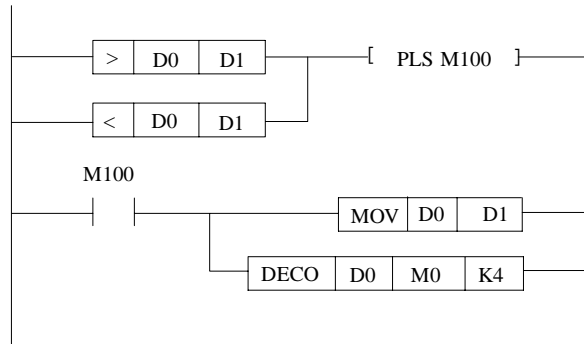
A.Obj	:D0
Pos	:2

Dec :0  
 Man :YES  
 Min :30  
 Max :39  
 Off :30  
 Gain :1

By setting the offset to 30 the actual value in D0 will be 0, 1 ....9.  
 The DECO instruction is used to decode the value in D0.

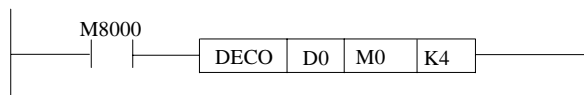
### MELSEC A

Use M0-M9 as display signals for the blocks 30-39.  
 The PLC program looks as follows:



### MELSEC FX

Use M0-M9 as display signals for the blocks 30-39.  
 The PLC program looks as follows:



## 5.4 Setting date and time

If the terminal is connected to a PLC with real-time clock then it is possible to set the clock from the terminal.

To set the right date and time in Run-time mode you have to make a maneuverable date/time object in a block. If you only want to change date or time you must make two objects to avoid problems with the cursor in run-time mode. One with date only and one with time only. If you have date and time in the same object and only want to change the time, then you must start editing the date to reach the positions where the time is displayed.

If you want to show day of the week as well, you make a third object with day of the week only. The day of week can never be changed by the operator, it is calculated by the date.

```
DATE  :#-----  
TIME  :#-----
```

```
DAY   :#-
```

*Text block for date/time object.*

It is possible to have maneuverable date/time objects in several blocks. A change in one block effects then of course also the other blocks.

---

**Note!**

Setting of format for presentation of date respectively time is made in Setup mode.

---

## 5.5 Password

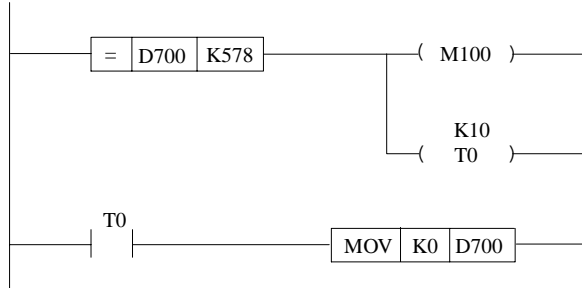
The easiest way to make a password in order to move from one block to another, for instance from block 4 to block 5, is to make a maneuverable object (data register) in block 4. In this analog object you type a value in RUN-mode. If the value is right (this is defined in the PLC-program), a jump is made to block 5.

```
Password: #--
```

*Text block 4.*

### MELSEC A

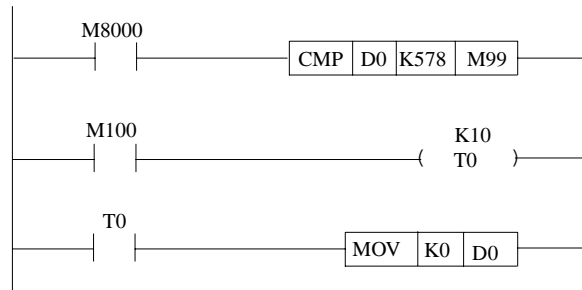
The password is typed into D700. Display signal for block 5 is M100.  
 After input of the right password (578) D700 is cleared. The program looks  
 as follows:



### MELSEC FX

The password is typed into D0. Display signal for block 5 is M100.  
 After input of the right password (578) D0 is cleared.

The program looks as follows:



## 5.6 Status of machine

In this example we have a mixer with functions for emptying, filling, rotation and warming. It is not possible to force warming.

D.Obj1 :M100
D.Obj2 :X101

```

D.Obj 3:Y30
Text 000:Empty
Man 000:YES
Text 001:Filling
Man 001:YES
Text 010:Rotate
Man 010:YES
Text 011:
Man 011:NO
Text 100:
Man 100:NO
Text 101:
Man 101:NO
Text 110:Heating
Man 110:NO
Text 111:
Man 111:NO
    
```

*Multiple choice object menu.*

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



```

--- MIXER ---
Mixer #-----

```

*Text block with multiple choice object.*

By typing YES at **Man 000:**, **Man 001:** and **Man 010:** it is possible to force to these status. A NO at M110 means that you can not force to this status.

## 5.7 Program selection

A multiple choice object can be used for defining different program selections. The example below shows how a machine can be set to turn out white, red, blue, yellow or green boxes.

```

D.Obj  1:M100
D.Obj  2:M101

```

```

D.Obj  3:M102
Text  000:White
Man   000:YES
Text  001:Red
Man   001:YES
Text  010:Blue
Man   010:YES
Text  011:Yellow
Man   011:YES
Text  100:Green
Man   100:YES
Text  101:
Man   101:NO
Text  110:
Man   110:NO
Text  111:
Man   111:NO

```

*Multiple choice object menu.*

Condition	D.Obj 1	D.Obj 2	D.Obj 3
Text 000 White	0	0	0
Text 001 Red	0	0	1
Text 010 Blue	0	1	0
Text 011 Yellow	0	1	1
Text 100 Green	1	0	0
Text 101	1	0	1
Text 110	1	1	0
Text 111	1	1	1

```
Select product  
Program: #-----
```

*Text block with multiple choice object.*

At setup you type that all status must be maneuverable. To change color in run-time mode, go to the multiple choice object with the arrow keys. When pressing [ENTER] a list with all maneuverable conditions is displayed. With the arrow keys you select wanted color and press [ENTER]. Now M100, M101 and M102 are forced into this new status.

## 5.8 Text from PLC

Below is described the process steps in an application for an food production. Text describing the steps is stored in the PLC:s data registers. The text for a step is in D0-D7.

T.Obj	:D0
Pos	:20

Maneuver :NO

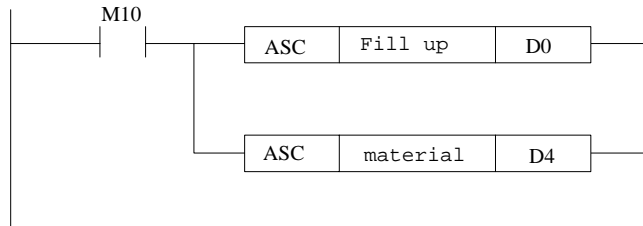
### *Text object menu*

Process step:
#-----

### *Text block with text objects.*

In a sequence built PLC program, each step or working moment is represented by a memory cell. When this is activated the text connected to the process step appears on the display.

The example below shows how to store text in data registers by the ASC-instruction. M10 is connected to the process step.



An instruction equal to the ASC-instruction in A- and FX-series is not available in the F1/F2-series. In the F1/F2-series the decimal values of the letters must be transferred to registers one at a time. Each register contains only one letter.

## 5.9 Behavior of display

Defined status word for each block decides its appearance. This makes it easy to make for example a selection list or an input sheet.

### Selection list

In this example we want to select which pump to check-up. A selection means jumping to the text block in which run-time data for the pump is collected.

```
-- SELECT PUMP -  
#PUMP 1
```

```
#PUMP 2  
#PUMP 3
```

*The text block in programming mode.*

When the text block with selection list is displayed we want the cursor to position the first jump object, not the top left corner in run-time mode.

We therefore type YES at the **1:st obj** in the status word menu.

```
Cursor off :NO  
1:st obj   :YES
```

```
Dis.MAIN   : NO  
MORE-ind   : NO  
Auto Ent   : NO
```

*Status word menu.*

## Input screen

In this example we are going to input length and width for a number of boards. Type length and width in data registers.

```
BOARD DIMENSIONS
B.1 L:#-- W:#--
```

```
B.2 L:#-- W:#--
```

```
B.3 L:#-- W:#--
```

*The text block in programming mode.*

We want the cursor to move automatically to width after having typed length and so on. To get this function we type YES at Auto Data Enter in the status word menu. Then the cursor moves automatically to next maneuverable object after a typing.

```
Cursor off:NO
1:st obj :YES
```

```
Dis.MAIN : NO
```

```
MORE-ind : NO
```

```
Auto Ent : YES
```

*Status word menu.*

## 5.10 How much can be stored in a project

Each project can contain maximum 8 000 bytes. These bytes can be divided freely between different functions.

The list below shows how many bytes every function occupies:

<b>Block</b>	15 bytes
<b>Text block</b>	16 bytes/line
<b>Dynamic object</b>	30 bytes

Function keys do not occupy any bytes from this memory of 8 000 bytes.

When displaying a block in Programming mode, [F1] [INFO] will show how many bytes are left.



## 6 Reference guide

This chapter contains a general description in the functions of the different modes of the terminal. Each function will be presented in the following way:

- Name and purpose of the function.
- Description with detailed information about the different alternatives of the function.
- Parameters with description of the parameters of the dynamic objects.
- Warning of more permanent consequences of a command.

The picture texts shown in squares in this manual are equal to the lines shown on the display. The lines above or below the square represent the currently hidden text lines on the display.

### 6.1 Setup mode

In Setup mode the general parameters for the terminal are set. In programming mode, Setup mode is selected by pressing any key while the initial picture is displayed.

If the terminal is in run mode then switch to programming mode by pressing [MAIN MENU] [INFO].

```
>PLC-selection
System signals
```

```
Language
Character set
Date/Time Form
Port parameters
Erase memory
Terminaloptions
```

*Main menu in Setup mode.*

To make an option from the menu, move the cursor to the appropriate line and press [NEXT].

## PLC-Selection - Selection of PLC system

```
>F1 / F2  
FX
```

A-CPU  
AJ71-C24 (S3)

### *Menu for PLC selection.*

In this menu you select which PLC system the terminal will be connected to. Move the cursor to appropriate PLC system and press [ENTER]. Application in the terminal is not erased when changing PLC system. I/O numbers not allowed in the choosed PLC system will not be erased.

To connect the terminal to AJ71C24, A1SJ71C24-R4 or the RS-422 port on A2CCPUC24, select AJ71-C24 (S3) and set the parameters in Port Parameters in main menu. From now this modules will be mentioned as computer link module.

**System signals** - Set handshaking signal for interaction between the terminal and MELSEC PLC.

```
Disp IO:
```

### *System signal menu.*

At **Disp IO**: type the data register, which in Run-time mode should contain the number of the displayed block. The register is automatically updated after each block change.

---

#### **Note!**

This register is only for reading in the PLC program. You can not change textblock by writing to it.

---



**Language - Select menu language**

```
>English
  United States
```

*Menu for selection of menu language.*

Move the cursor to correct alternative by using the arrow keys to make your selection. Return to Setup menu by pressing [LEAVE], or [ENTER] at the desired language.

**Character set - Select character set**

Select the character set to work with. Eight character codes (C1-C8) are reserved for domestic characters.

```
>Swedish
  German
```

```
  French
  Spanish
  Italian
  Nor/Dan
```

*Character selection menu with Swedish as current language.*

To select a new character set, move the cursor to appropriate alternative and press [ENTER]. For information about the characters, which are connected to each code for the different languages, see *Appendix*.

Notice that this only affects which character C1-C8 corresponds to.

Leave the Character set menu by pressing [LEAVE], or [ENTER] at the desired character set.

## Date/Time format - Set Date/Time format

```
Datefmt:YY-MM-DD  
Timefmt:HH:MM:SS
```

*Display appearance when selecting Date/Time format.*

This function does only make sense if the terminal is connected to a CPU with a real time clock. When changing format of date, move the cursor by using the arrow keys, to the line **Datefmt:** and type the format you want. Use the same method to change format at the line **Timefmt:**.

The following characters can be used for defining format:

D - Date in numeric  
H - Hour in numeric  
M - Minute and months in numeric  
S - Second in numeric  
Y - Year in numeric  
- : . / , [Blank] - Punctuation marks

Valid formats for date are e.g.:

<b>Format</b>	<b>Example</b>
YY-MM-DD	90-08-15
DD.MM.YY	15.08.90
DD/MM-YY	15/08-90

Valid formats for time are e.g.:

<b>Format</b>	<b>Example</b>
HH:MM:SS	15:33:36
HH.MM	15.33

To set the real-time clock you create a maneuverable date/time object in programming mode. Then the operator can set the real-time clock in run mode. You don't have to pay attention to special signals connected to the real-time clock as for example M9025 and M9028 in the A-CPU.

**Port parameters - Port settings**

In port parameters the parameters for PLC and the PC software are set. When selecting port parameters a sub-menu appears in which PLC or MP are selected by pressing [ENTER].

```
>PLC
MP
```

*Sub-menu for port parameters.*

When selecting **PLC** following menu is displayed:

```
Baudrate : 9600
Data bits : 8
```

```
Stobits : 1
Parity : ODD
Station : 0
```

*Menu for setting of PLC parameters.*

The parameters are set when the terminal is connected to a computer link module. The parameters are only changeable when AJ71C24(S3) is selected in the PLC selection menu. To change parameter, move the cursor to appropriate parameter and press [INFO]. The value of next option is displayed. Leave the menu by pressing [LEAVE]. The number at **Station:** must be the same as the station number on the computer link module.

When selecting **MP** following menu is displayed:

```
Baudrate : 9600
Data bits : 8
```

```
Stopbits: 1
Parity : NONE
```

*Menu for setting of MP parameters.*

In this menu the parameters for interaction between the PC software and the terminal are defined. To change parameter, move the cursor to appropriate parameter and press [INFO]. Then the value of next option is displayed. Press [LEAVE] to leave menu.

### **Erase memory** - Erase the whole application memory

By choosing Erase memory the application memory of the terminal is cleared. All text blocks and definitions of function keys, and system signals are deleted. Also Date/Time Format is reset to default values.

When erase is finished the Setup menu will automatically appear on the display.

---

**Note:**

Once you have used the **Erase memory** command, all dates stored in the terminal are lost.

---

### **Terminal options** - Terminal setting

Key Repeat: YES

*Menu for terminal options.*

At **Key Repeat**: you select if a function shall be repeated while pressing the key. Change between YES and NO with [INFO].

## 6.2 Programming mode

In Programming mode the terminal is programmed by using the built-in keyboard, and projects between Personal Computers and the terminal are transferred. To enter Programming mode, press [MAIN MENU] [INFO].

```
>Edit
  Transfer
```

Block list

*Programming mode menu.*

Select from menu by position the cursor on desired line and then press [NEXT].

### Edit

After selecting Edit, choose the type of data to be edited. Move the cursor to correct alternative by using the arrow keys and press [NEXT].

```
>Text block
  Function keys
```

LED

*Edit menu.*

**Text block** - Text block definition

The text block consists of a free amount of lines. Each line has 16 positions for characters. If the block consists of more than two lines it can be scrolled on the display by using the arrow keys. An application contains maximum 990 blocks. Following blocks are reserved as system blocks:

<b>0</b>	Main menu
<b>990 - 997</b>	<i>Reserved, not used</i>
<b>998</b>	LED overview page
<b>999</b>	Function key overview page

In run-time mode Main menu can be addressed directly from the keyboard by the key [MAIN MENU].

You can make jumps to these blocks and also define Display signal and Printer signal in the same way as described under Text block.

Each text block must have a number assigned to it.

After selecting a text block, the following menu will appear on the display:

```
BLOCK NUMBER :  
Disp sign   :
```

```
Statusword  :
```

#### *Definition of text block.*

After the text **BLOCK NUMBER** the appropriate number is typed. If that text block already exists, the defined values, if any, will be displayed automatically. To move between different fields use the arrow keys. Terminate input data with [ENTER] and the cursor will get to the next field.

The text block that should be the main menu of the application, e.g. the one that is displayed when starting up the program, is always number 0. **Disp sign:** defines the signal which activates the text block to be current on the display. To get as quick block changes as possible in run-time mode the display signals should be consecutive. **Status word:** defines the character of a display in run-time mode.

Press [INFO] to enter the menu for status word.

```
Cursor off:NO  
1:st obj   :NO
```

```
Dis MAIN   :NO
```

```
MORE-ind.  :NO
```

```
Auto Ent   :NO
```

#### *Menu for status word.*

To change between YES and NO press [INFO] and leave the menu with the key [LEAVE].

**Curs off:** defines if the cursor shall be displayed in text block in run-time mode.

**1:st obj:** defines if the cursor shall locate the 1st maneuverable object in text block instead of top left corner.

**Dis MAIN:** defines if the [MAIN MENU] key shall be blocked in run-time mode when the text block is displayed.

**MORE-ind:** defines if the + character shall be displayed in bottom right corner and top right corner when there is more current hidden text lines.

**Auto Ent:** defines if the cursor shall move automatically to the next man obj. after [ENTER] is pressed.

When the text block definition is completed then press [NEXT] to edit the textblock. The first time this is done you will get an empty display. To return to previous menu press [LEAVE].

### **Function keys** - Definition of function keys

There are six built-in function keys in the terminal, F1-F6. They are defined in system block 999.

F1: M110
F2: M111

F3: M112

F4: \_

F5:

F6:

### *Menu for definition of function keys.*

The function keys are connected to a digital signal by typing the signal address for each key. During run-time the digital signal will be on, as long as the key is pressed.

## LED - Definition of LEDs

There are six built-in LEDs in the terminal, L1-L6. They are defined in system block 998.

```
L1 :  
L2 :
```

```
L3 :  
L4 :  
L5 :  
L6 :
```

### *Menu for definition of LEDs.*

The LEDs are connected to a digital signal by typing the signal address for each LED. During run-time the LED will be lit when the digital signal is on.

## **Transfer** - Transfer of applications between the terminal and a personal computer

Transfer is used for transferring projects between the terminal and personal computers. This can only be done if the personal computer is equipped with the PC software for programming and documentation of the terminal.

This package is available for IBM PC/XT/AT and PS/2 for DOS or Windows.

## **Block list** - Display all defined text block numbers.

Block list specifies all the blocks used in the application.

```
■USED. BL (4) ■  
0 1 2 10
```

### *Block list menu.*

To erase a block, move the cursor to it and press [F1] [CLEAR]. The block will then be erased.



## 6.3 Dynamic data fields

The dynamic fields can be found everywhere in the text. To create a dynamic field press #. The menu for selection of dynamic field is then displayed.

```
>Digital
  Analog
```

```
Jump
Date/Time
Bar graph
Mult.choice
Text
```

*Menu for selecting type of dynamic field.*

To select type of object, move the cursor to the proper alternative and press [NEXT].

---

### Note!

When the input field of the object is displayed you can copy the latest configuration of respective object by pressing [INFO].

---

### Digital - Digital objects

```
D.Obj   :_
Text 0  :
```

```
Text 1  :
Maneuver : NO
```

*Digital object menu.*

Digital objects can only have two states, 0 or 1.

### Type of signal:

Digital objects can be one of the following signal types:

A series: X, Y, M, B, F, T, C

FX and F series: X, Y, M, S, T, C

For signals type T and type C we are referring to status.

**Parameters:**

**D. Obj:** Signal address.

**Text 0 :** Text displayed when signal is OFF, the text can be seven characters long.

**Text 1 :** Text displayed when signal is ON, the text can be seven characters long.

**Maneuver:** Defines if the object is to be forced from the terminal in run-time mode. Possible value is **YES** or **NO**. **NO** is default. The value is changed by pressing [INFO] with cursor in this position.

**Analog - Analog objects**

A.Obj	: _	
Pos	: 5	
Dec	: 0	
Maneuver	: NO	
Min	: -32768	
Max	: 32767	
Off	: 0	
Gain	: 1	

*Analog object menu.*

Analog objects represent integer numbers or floating point numbers.

**Type of signal:**

Analog objects can be one of the following signal types:

A series: D, W, R, T, C. For T and C actual value is displayed.

FX series: D, R, T, and C. For T and C actual value is displayed.

F1/F2 series: D, T and C.

**Parameters:**

**A. Obj:** Signal address. By providing the analog signal with an affix you can present double registers and/or unsigned values.

---

**Note!**

Values with decimals and/or scaling are stored internally as floating point values and therefore rounded which means that the last digits in a number with more than six significant digits can be inexact. To avoid this use values without decimals which are internally stored as integers.

---

**MELSEC A and MELSEC FX**

Notation	PLC register type	Range
<b>D100</b>	Single	-32768 – +32767
<b>D100+</b>	Single	0 – +65535
<b>D100L</b>	Double	-2147483648 – +2147483647
<b>D100L+</b>	Double	0 – +4294967295

**MELSEC AnA**

Notation	Format type	Range
<b>D100RB</b>	BCD-real number	-9999.9999 – +9999.9999
<b>D100RF</b>	Floating point real number	$\pm 3.4E \pm 28$ Numbers $\geq 1000000$ are displayed with exponent. Decimals definition has no function for these presentation.
<b>D100L</b>	Double	-2147483648 – +2147483647
<b>D100L+</b>	Double	0 – +4294967295

**MELSEC F1/F2**

Notation	PLC register type	Range
<b>D700</b>	Single	0 – 999
<b>D700L</b>	Double	0 – 999 999

For timers and counters either the preset value or the actual value can be presented. Following examples exist:

Notation	Presentation
<b>T50</b>	T50 actual value
<b>T50L</b>	T50 and T51 actual value
<b>T50PRE</b>	T50 preset value (F1/F2 only)
<b>T50LP</b>	T50 and T51 preset value (F1/F2 only)

**Pos:** Specifies the number of positions used by the object. Also characters as minus sign or decimal point must be included, if any. Default is five.

**Dec:** Specifies the number of decimals represented by the object. Default is 0. Notice that the value must be scaled to a decimal value otherwise you will only get zeroes as decimals.

**Maneuver:** Defines if the object shall be changeable from the terminal in run-time mode. Possible values YES and NO. NO is default. Change value by pressing [INFO]. For timers and counters in the A and FX series only the actual value is changeable. In the F1/F2 series only the preset value is changeable.

**Min:** The smallest value that the operator can enter for an object. If the value has more than seven positions scrolling is done horizontally.

**Max:** The largest value that the operator can enter for an object. If the value has more than seven positions scrolling is done horizontally.

**Off:** and **Gain:** are used to scale the signal address value in the PLC to a displayed value according to the equation below:

$$\text{Display value} = \text{Offset} + \text{Gain} * \text{SignalAddressValue}$$

If a value is typed in from the terminal the value is scaled to a signal address value in the PLC according to the equation below:

$$\text{SignalAddressValue} = (\text{Display value} - \text{Offset})/\text{Gain}$$

Scaling does not affect defined max and min values.

**Jumps - Jump objects**

	Cur.block: 12	
	Jump to : _	

***Jump object menu.***

Jump objects make it possible to build up a menu tree in the application.

**Signal type:**

Jump objects do not work with signals from the PLC system.

**Parameters:**

**Jump to:** defines the number of the text block you want to jump to. If a jump to a non-existing block is generated during run-time, system error will occur. **Cur.block:** defines the number of the block you are leaving.

**Date/time - Objects for date/time**

	Weekday : NO	
	Date : NO	
	Time : NO	
	Maneuver : NO	

***Date/time object menu.***

Date/time objects return data from the real time clock in the PLC. You can choose several of the alternatives to be displayed.

**Signal type:** Date/time objects work with the real-time clock in the CPU.

**Parameters:**

**Weekday:** Defines if weekday should be displayed. Possible value YES or NO. NO is default. Change value by pressing [INFO] with cursor in this position. Weekday requires 2 positions.

**Date :** Defines if date should be displayed. Possible value YES or NO. NO is default. Change value by pressing [INFO] with cursor in this position. Date normally requires 8 positions.

**Time :** Defines if time should be displayed. Possible value YES or NO. NO is default. Change value by pressing [INFO] with cursor in this position. Time normally requires 8 positions.

**Maneuver:** Defines if an object should be changeable in run-time mode. Possible value YES or NO. NO is default. Current date/time is changed by typing the new value in run-time mode and pressing [ENTER].

**Bar graph** - Bar graph for analog signal

A.Obj	:	_
Pos	:	16
Dir	:	
Min	:	-32768
Max	:	32767
Off	:	0
Gain	:	1

*Bar graph object menu.*

Bar graph objects represent integral numbers or floating numbers in the shape of a bar graph diagram. The bar graph will be displayed horizontally so that the maximum number of positions are marked when the value has reached the top level.

### Signal type:

Bar graph objects can be analog signals with one of following types:

A series: D, W, R, T, C. For T and C actual value is displayed.

FX series: D, R, T and C. For T and C actual value is displayed.

F1/F2 series: D, T and C. Signal affixes are the same as for analog objects.

### Parameters:

**A. Obj:** Signal address.

**Pos:** Defines the number of positions used for the signal.

Maximum number of positions is 32.

**Dir:** Defines if the bar graph shall be presented to the right or left. Change between Right and Left with the [INFO] key.

**Min:** The smallest value for a signal. If the value has more than seven positions scrolling is made horizontally.

**Max:** The largest value for a signal. If the value has more than seven positions scrolling is made horizontally.

**Off:** and **Gain:** Are used to scale the input value of a signal to a displayed value according to the equation below:

$$\text{Display value} = \text{Offset} + \text{Gain} * \text{SignalAddressValue}$$

Scaling does not affect defined max- and min-values.

**Multiple choice** - Object for several status

D.Obj	1:
D.Obj	2:

D.Obj 3:  
 Txt 000:  
 Man 000:NO  
 Txt 001:  
 Man 001:NO  
 Txt 010:  
 Man 010:NO  
 Txt 011:  
 Man 011:NO  
 Txt 100:  
 Man 100:NO  
 Txt 101:  
 Man 101:NO  
 Txt 110:  
 Man 110:NO  
 Txt 111:  
 Man 111:NO

Multiple Choice Object is an object which can consist of several status. The object is connected to three digital signals that together can adopt eight different status. To each status you can connect a text obtaining seven characters.

**Signal type:**

Digital objects have one of following types:

A series: X, Y, M, B, F, T, C  
 FX and F1/F2 series: X, Y, M, S, T, C

**Parameters:**

**D.Obj 1 - D.Obj 3:** Signal addresses. It is not necessary to type three signals, if two are typed four status can be displayed.

**Text 000-Text 111:** Texts that shall be displayed for respective status in the object. See the table below:

Status	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:** Defines if it shall be possible to force to this status in run-time mode.

**Text - Object for handling of ASCII-strings**

T.Obj	:
Pos	: 8

Maneuver :NO

*Text object menu.*

In a text object it is possible to present texts stored in the CPU's data register. The text shall be in ASCII-format. During run-time mode these texts can be changed in the terminal by open and close an input field with [ENTER].

**Signal type:**

**T. Obj** can have one of following types:

A series: D, W and R

FX and F1/F2 series:D



**Parameters:**

**T. Obj:** Defines the register in which the first position is stored.

**Pos:** Specifies the number of positions the text will use on the display. For the A and FX series each register obtains two characters (two positions) and for the F1/F2 series it obtains one character. Default is eight positions, which is equal to four registers in the A and FX series and eight in the F1/F2 series. The maximum number of positions is 32.

**Maneuver:** Defines if it will be possible to change the object in run-time mode. Change between YES and NO by pressing [INFO].

## 6.4 Run-time mode

In Run-time mode you can show status and control the process from the terminal. To enter this mode press [MAIN MENU] [INFO]. Text block 0 will then automatically be displayed.

The functions of the different objects during run-time will be described below. They do not correspond to a menu selection during operation.

**Text blocks** - monitoring of text blocks.

A text block contains text lines with static words and dynamic data fields.

```
WATER TANK
→PH-regulation
```

```
→Fluid level
  92-06-25  12:33
```

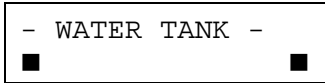
*Example of text block in run-time mode.*

The dynamic fields show current status for the signals to which the objects are connected.

Some dynamic data fields are maneuverable, which means that it is possible to change its status and value during run-time. To change a maneuverable object, use the arrow keys to position the cursor on the object. To change status of digital object, press [ENTER]. Analog and Date/Time objects are maneuvered by typing the new value after having positioned the cursor, and then pressing [ENTER]. When you type the first number in an analog object an input field marked with ■ ■ will appear.

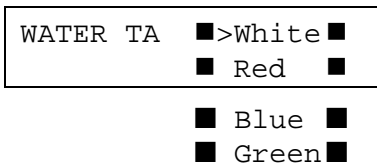
If you want to reverse the new value but have not yet pressed [ENTER] you can leave the field with [↑] or [↓]. The old value will then remain.

To change a text object, position the cursor on the object and press [ENTER]. Then an input field will be displayed. If the text has more than 14 characters scrolling is done in the input field. When typing text the function keys are used as shift keys. When the text has been typed, press [ENTER] and the input field disappears.



*Input field in text object.*

To change a multiple-choice object, position the object with the arrow keys. Press [ENTER] and a selection menu with all maneuverable status is displayed. Move to the appropriate status, press [ENTER] and the three digital signals connected to this status are forced.



*Selection menu in multiple-choice object.*



- Information about the dynamic objects during run-time.

In Run-time mode you can get information about the dynamic objects by positioning the cursor in front of appropriate object and pressing [INFO]. A text line will then appear on the bottom line of the display if the object itself is on the first line, otherwise it will appear on the top line of the display.

To remove the information line, press [INFO] again. For digital objects the texts defined for position OFF/ON will appear and for analog objects the min and max input level.

## 7 Summary

### 7.1 Modes

The terminal has three different modes:

- Programming mode
- Run-time mode
- Setup mode

Change between Programming and Run-time mode by pressing [MAIN MENU] [INFO]. In Programming mode Setup mode is selected by pressing any key while the initial picture is displayed. By pressing [F1] [CLEAR] while power up you will get a total reset of the terminal.

### 7.2 Objects

The following objects can be used in blocks:

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

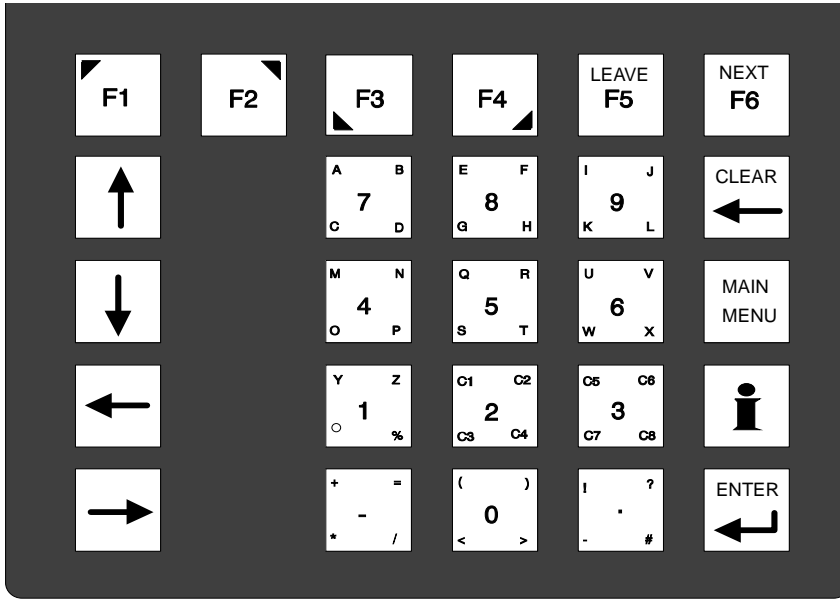
### 7.3 System blocks

Certain blocks are reserved for special purposes:

<b>0</b>	Main menu
<b>990-997</b>	<i>Reserved, not used</i>
<b>998</b>	LED indicators
<b>999</b>	Function keys

## 7.4 Keyboard

The keyboard layout looks like this in programming mode:



## 7.5 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 kind of character setting you have chosen in Setup.

	Swedish		German		French		Spanish		Norwegian/ Danish	
	upp.	low.	upp.	low.	upp.	low.	upp.	low.	upp.	low.
<b>C1</b>	Å	å	Ü	ü	È	È	ñ	ñ	Å	å
<b>C2</b>	Ä	ä	Ä	ä	É	É			Æ	æ
<b>C3</b>	Ö	ö	Ö	ö	Ê	Ê			Ö	ö
<b>C4</b>			β	β					Ø	ø

There are also some special characters available by pressing [F1] [MAIN MENU].

: ' ^ \_ { } £ ε Ω Σ π ÷ ■ α ∞

## 7.6 International character set

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		!	1	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	á						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		»					

## 7.7 Keys

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

### Programming mode



Shift



Shift



Shift



Shift



Leave



Next



Move



Change among alternatives

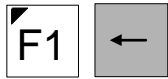


Erase the character in front of the cursor

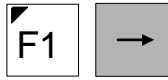


Next line

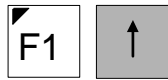
## Summary



Erase the character where the cursor is positioned



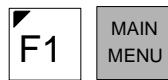
Insert of characters



Upper case (capital) letters



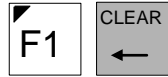
Lower case (small) letters



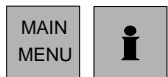
Access to special character table



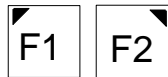
Number of bytes left



Erase block in block menu

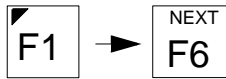


Alters between programming- and run-time mode

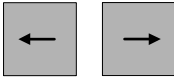


Keyboard layout



**Run-time mode**

Functions keys



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



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 text/values



Alters between programming- and run-time mode

## 7.8 Basic set-up

The terminal is delivered with the following settings:

<b>Language:</b>	Englis
<b>Character setting:</b>	Swedish
<b>Date/Time Format:</b>	Date Fmt YY-MM-DD
	Time Fmt HH:MM:SS

## 7.9 Data

<b>Front panel</b>	Width 5,91 x height 6,69 x 0,12 in (150 x 170 mm).
<b>Mounting depth</b>	Including D-sub connector 1,30 in (33 mm).
<b>Mounting hole</b>	Hole: Dimension $5,20 \pm 0,16$ x $5,98 \pm 0,16$ in ( $132 \pm 4$ mm x $152 \pm 4$ mm). 4 pcs. mounting screws, M4, 5,51 x 6,30 in (140 x 160 mm). Panel thickness $\leq 0,31$ in ( $\leq 8$ mm).
<b>Front panel seal</b>	IP 65
<b>Rear panel seal</b>	IP 20
<b>Front panel material</b>	0,10 in (2,5 mm) black anodised aluminium and membrane key- board, overlay film in polyester with backside printing.
<b>Back side material</b>	Black anodised aluminium.
<b>Weight</b>	Excluding D-sub cable 1,32 lb (0,7 kg).
<b>PLC-system connection</b>	25-position D-sub connector, male with standard screw lock 4- 40 UNC. RS-422 communication.
<b>Cable to PLC-system</b>	6-pair screened D-sub, housing in metallized plastic. Screw lock to PLC-system: M2,5. Screw lock to MTA 100: 4-40 UNC.
<b>Display</b>	LCD-display (Liquid crystal) Supertwist 2 rows x 16 characters, 0,33 in (8 mm) character height. Viewer angle adjustable with potentiometer behind battery cover.
<b>Memory</b>	10 year data retention min. 10 000 write cycles/byte Mounted in 28 pins IC-socked EEPROM.
<b>Supply voltage</b>	From PLC-system or via external power supply 5 VDC $\pm 5\%$ min. 200 mA.
<b>Internal power</b>	+5 V $\pm 5\%$ . Capacitance to chassis: 0,1 $\mu$ F. Resistance to chassis: 1 M $\Omega$ .
<b>Ambient temperature</b>	+32° to +122°F (0° – +55°C).
<b>Storage temperature</b>	-4° to +140°F (-20° – +60°C).
<b>Humidity</b>	90%, no condensation.
<b>EMC test</b>	G & L Beijer Electronics AB confirm that the terminal conforms with the essential protection requirements in article four of the directive 89/336/EEC. Noise tested according to: EN50081-1 emission and EN50082- 2 immunity.



# Index

## A

Analog object, 24  
Analog, 58

## B

Bar graph object, 25  
Bar graph, 62  
Block list, 56

## C

Character selection, 15  
Character set, 49, 69  
Connection to A or FX CPU, 4  
Connection to A1SJ71C24-R4, 6  
Connection to A2CCPUC24, 7  
Connection to AJ71C24, 6  
Connection to F1/F2, 5  
CPU port, 4

## D

Date/time, 50, 61  
Default settings, 74  
Digital object, 22  
Digital, 57  
Dynamic information, 20  
Dynamic object, 57  
    Analog, 58  
    Bar graph, 62  
    Date/time, 61  
    Digital, 57  
    Jump, 61  
    Multiple choice, 63  
    Text, 64

## E

Edit, 53  
    Block list, 56  
    Function keys, 55  
    LED, 56  
    Textblock, 53  
    Transfer, 56  
Erase memory, 52

## F

Function keys, 28, 55

## J

Jump object, 21  
Jump, 61

## K

Keyboard, 12, 68  
Keys, 71

## L

Language, 49  
LED, 56

## M

Menu selection, 12  
Modes, 11, 67  
Multiple choice object, 26  
Multiple choice, 63

## O

Object, 67

## **P**

PLC system, [4](#)  
PLC-selection, [48](#)  
Port parameters, [51](#)  
Programming mode, [53](#)  
Programming, [8](#)

## **R**

Run-time mode, [65](#)

## **S**

Setup mode, [47](#)  
Static information, [20](#)  
System block, [15](#), [67](#)  
System signals, [48](#)

## **T**

Technical data, [75](#)  
Terminal options, [52](#)  
Text object, [27](#)  
Text, [64](#)  
Textblock, [15](#), [53](#)  
Transfer, [56](#)



