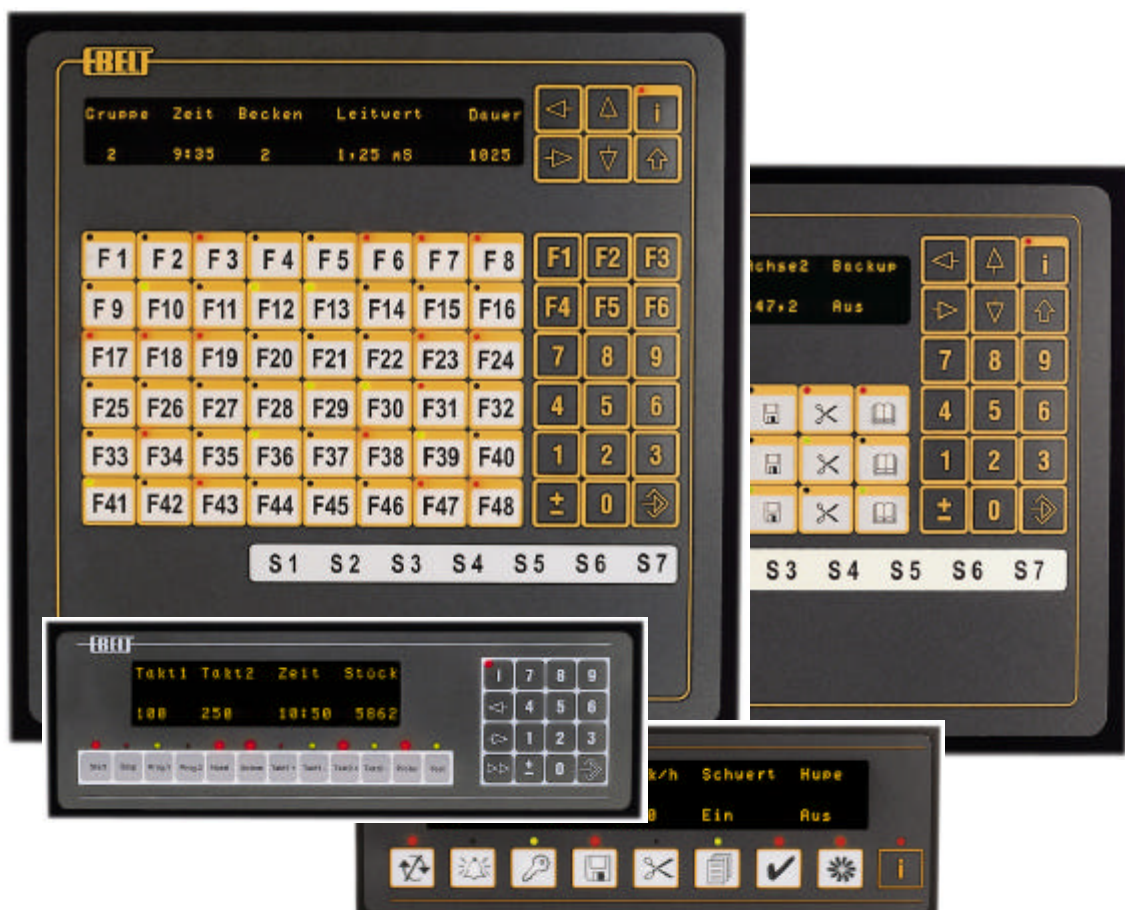


# OPERATING PANELS

# BEST

## PROGRAMMING Manual EBELT OPERATING PANELS

BEST 240-0  
BEST 240-8  
BEST 224-12-Z  
BEST 240-24-Z  
BEST 240-48-Z  
BEST240LCD-8  
BEST 240LCD-12-Z  
BEST 240LCD48-Z



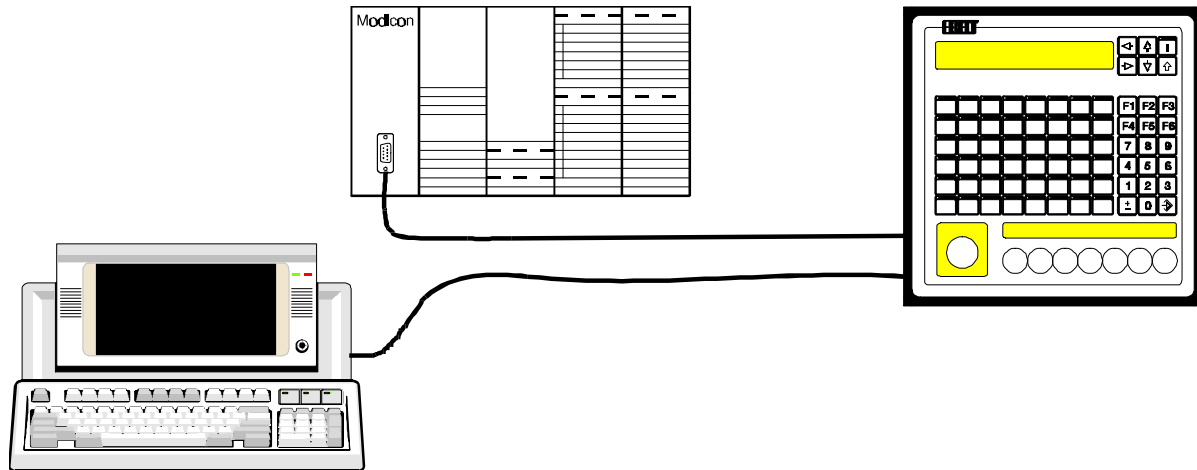
# Programming the Operating Panel

1. Loadable Character Generator
2. Programming the System Settings
  - 2.1 Valid System Settings
3. Structure of a Message
  - 3.1 Variable Definition
  - 3.2 Message Texts with Filler Characters for Variables
  - 3.3 Function Key Reset
  - 3.4 Info Texts
4. Variables
  - 4.1 Indication Variables
    - 4.1.1 Suppression of Leading Zeros
  - 4.2 Input Variables
    - 4.2.1 Indexed Variable Address
  - 4.3 Input Limits
    - 4.3.1 Indexed Input Limits
  - 4.4 Variable Formats
    - 4.4.1 Variable Format UNS
    - 4.4.2 Variable Format INT or KF
    - 4.4.3 Variable Format INTL or MD
    - 4.4.4 Variable Format BCD
    - 4.4.5 Variable Format BCDL or LNG
    - 4.4.6 Variable Format HEX or KH
    - 4.4.7 Variable Format BIT or KM
    - 4.4.8 Variable Format KT
    - 4.4.9 Variable Format ASC
    - 4.4.10 Variable Format TXT
  - 4.5 Scaled Variables
    - 4.5.1 Scaled UNS
    - 4.5.2 Scaled INT
5. Error and System Messages
6. Printer Connection

## Programming the Operating Panel

The message texts and system parameters are programmed on standard programming units under MS-DOS. The "EBELT" software required for this process and the corresponding connecting cables are included in the

installation set. The configuration shown below can be used to program the operating panel and to program the PLC for all PLC connections with an integrated "PG-MUX".



The text memory is divided into 3 areas:

- loadable character generator
- system settings
- message texts with variable definition

Depending on the device type, a Flash-Eprom, EEPROM or a Memory Card is used as text memory. Depending on the device, the memory capacity is between 8 Kbytes (EEPROM) and 512 Kbytes (Flash-EPROM).

### 1. Loadable Character Generator

**In addition to** the internal character generator, the BEST operating panels with LED text display allow loading of a second character set which is independent of this character generator. This allows for mixed representation of various character sets on the display. As the character generator consumes a certain amount of space in the text memory, it always comes with a text memory of at least 32

Kbytes. It can, however, be enabled in the 8 Kbytes version, on special request. Operating panels with LCD text displays provide only 8 freely definable characters with one used by the firmware to represent an acceptable "g". This leaves 7 characters which can be used freely by the user (reference characters are the numbers 1...7). The freely definable character generator is created with the programming software under MS-DOS.

## 2. Programming the System Settings

All operating panel system parameters are set in plain text. This offers the advantage of the parameter settings being documented in an understandable language together with the message texts. The system settings appear in front of the message 0.

The presettings do not have to be typed in letter by letter! Connect the operating panel to your programming unit and select the menu item "Read text display" in the EBELT software.

The default setting (or the currently

programmed setting) is supplied, allowing you to simply make modifications as desired. The following parameters can be set:

Message text selection binary/BCD  
Internal character set CPM/IBM  
Display brightness (for LED displays)  
Scroll and blink times  
Communication flag for keys and LEDs  
Keyboard characteristics (push-button keys or latch-down keys)  
LED characteristics (internal/external/color)  
Printer connection on/o

### 2.1 Valid System Settings

#### Example: BEST224-12-Z

```
*Binär (1)
*Drucker (12)
*IBM Zeichensatz (2)
*Helligkeit: 80% (3)
*Scrollzeit: 3,2s (4)
*Blink-Ein: 0,3s (5)
*Blink-Aus: 0,2s (6)
*Ko-M Tast: 10 LED: 02 (7)
*Ta-Mode: BBBBMMMM (8)
*Ta-Led: IIIIIIEEEE (9)
*Ta-Farbe: RGRGRGR RGRG (10)
*Ta-Blink: EAEAEAEA AEAE (11)
*
BEST 224 ready for operation
```

**\* Binär (1)**  
The binary system is used for message text selection (in contrast to the BCD system). The possible number of message texts is 1000 + 23 system messages.

**\* BCD (1)**  
The BCD (binary coded decimal number) system is used for message text selection. The number of message texts is therefore limited to 400.

**\* Cpm (2)**  
The first internal character set is loaded in accordance with the ASCII-table used with CPM.

**\* IBM (2)**  
The first internal character set is loaded in accordance with the character table used with the IBM PC.

**\* Scrollzeit: 3,2 (4)**  
When a message with more than 2 lines is programmed and to be displayed on a 2-line display, the display will automatically begin to scroll the text one line at a time. This scroll time can be set to a time between 0,1 and 6,3 s.

**\* BLink-Ein: 0,2 (5)**  
The Blink-On duration can be set to a time between 0,1 to 6,3 s.

**\* BLink-Aus: 0,2 (6)**  
The Blink-Off duration can be set to a time between 0,1 to 6,3 s.

**\* Ko-M Tast: 10 LED: 02 (7)**  
Communication flag for F-keys and LEDs  
Exception: S5 is parameterized in the data handling block.

**\* Ta-Mode/LED/Farbe/Blink (8)...(11)**  
The following lines vary with the different operating panels depending on the number of function keys and the assigned LEDs. For operating panels with only 12 function keys, they are defined in one line. For operating panels with more function keys, a new line is used for every 8 function keys.

**\* Ta-Mode: MBBBMMMM MBMB (8)**  
The key mode can be defined separately for every key. M stands for monostable or push-down key, B stands for bistable or latch-down key. The sequence is read from left to right.

**\* Ta-LED: EIIIIIEE IEIE (9)**  
LEDs in the keys can be activated either internally (directly by the key) or externally by the PLC. E stands for externally, I stands for internally. The sequence is read from left to right.

**\* Ta-Farbe: RGRRRGGG RGRG (10)**

The color of the LEDs above the F-keys can be selected: R or 1 stands for red, G or 0 stands for green.

**\* Ta-Blink: EAAAAAEA EA EA (11)**

A blink mask can be defined for the illuminated keys: E or 1 stands for Blink-On, A or 0 stands for Blink-OFF.

The blink bits from the PLC and the blink setting

in the operating panel are linked in accordance with the logic operation "OR".

**\* Drucker (12)**

Most of the BEST-variants allow connection of a logging printer. In this case, the printer (instead of the PC) is connected to COM1 of the operating panel. However, the printer can not be connected while the internal PG-MUX is to be operated.

### 3. Message Structure

In addition to the text to be displayed, the following information is programmed in the message texts:

- variable format
- variable address for variable to be displayed or to be edited

- input limits for input variable
- numbers of function keys that are to be reset on actuating the Enter-key
- info text

The structure of a message can be visualized as consisting of four sections:

**Example:**

@uns 16! @bcd 18 1500 8000 @bcd 20 100 2700	Section 1 Variable definition
Process inspection ACTUAL: ____ SETPOINT: ____ DIM.: __, __ mm	Section 2 Display message image
@7,2	Section 3 F-key reset
System is stopped until all values have Been entered.	Section 4 info text

**Section 1** is used to define the variables. That is, the format, address and input limits of the variables are defined here.

**Section 2** shows the actual message image with the filler characters (\_\_) holding the place for the variables. The number of filler characters determines how many digits are displayed. The input cursor is displayed as a blinking cursor. The variable does not have to be programmed as a blinking variable.

**Section 3** is the separation line to the info text and also contains a list of up to 8 F-keys which are reset by pressing the Enter-key, if they are set. The list applies while the message text is being displayed.

**Section 4** contains the info text which can be called up by the operator with the info key. The LED above the info key is lit if an info text is available.

### 3.1 Variable Definition

Up to 8 variables can be displayed or edited in a message. In this case, variables are numbers (16 bit or 32 bit). The data format is specified in the definition line.

When a message is called up, the variables are

automatically read from the PLC and replace the filler characters holding the place for the variables.

Every variable is assigned to one block of filler characters. This block determines the number of variable digits (displayed).

#### Example:

<pre>@uns 16! @bcd 18 1500 8000 @bcd 20 100 2700</pre>	<p>The variables are defined line by line and displayed in the message text in that order.</p>
<pre>Process inspection ACTUAL: ___ SETPOINT: ___ DIM.: __, __ mm</pre>	<p>In the message text, every group of connected ( ) underlines represents filler characters holding the place for a variable.</p>

### 3.2 Message Texts with Filler Characters for Variables

Blocks of filler characters are separated by any characters except "." and ",".

If the maximum number of 8 variables is used per message text, they can no longer be displayed on a two-line display at once. To allow input, the message image is not automatically scrolled (scroll time), if **input variables** are programmed in a message. The lines are displayed in such a way that the input

cursor is visible. An input variable must therefore be inserted in at least every second line.

If scaled variables are used, one line is required to specify the scaling range. This reduces the maximum number of variables by one each. Specifying an index register for variables or input limits also reduces the possible number of variables by one.

### 3.3 Function Key Reset

The line with the character "@" separates the message text from the assigned info text. Up to 8 function key numbers can be specified in this line. The individual numbers are separated by a space, stop point or comma. If the specified

function keys are programmed as bistable (latch-down) keys and are currently set, they are reset by pressing the Enter-key if the input has been carried out correctly (range check).

### 3.4 Info Texts

In addition to every message and menu text, an info text can be programmed. The red LED above the (i) info key signals that an info text is available for the message currently displayed. Pressing the info key displays the programmed info text in place of the message text. When the info key is released, the message text is

displayed again. The info text display does not interrupt an ongoing input process. This means that after the info key is released, the input cursor is in the previous position. It is not possible to display variables in the info text. Except for the identifier "@", only the numbers of function keys to be reset are allowed in the separation line.

## 4. Variables

The variable definition is placed in front of the message text as already described. In order to distinguish the variable definition from text lines, the character "@" is inserted in front of the variable definition.

General Syntax:

```
@[ttt]□[xxx]□[!]□[min]□[max]
@[ttt] = variable format
```

```
[xxx] = variable address
[!] = identifier for an indication
variable
[min] = lower input limit
[max] = upper input limit
```

In this case, the box (□) stands for a space. When creating text, the underline represents a filler character holding the place for a variable.

### 4.1 Indication Variables

The character "!" signifies an indication variable. Indication variables are continuously read from the PLC. Input is not possible (e.g. temperature display).

Example:

```
@BCD□135!
```

#### 4.1.1 Suppression of Leading Zeros for Indication Variables

Indication variables allow suppression of leading zeros for the following variable formats: UNS, BCD, BCDL

The character "!" is replaced by a "#".

Example:

```
@BCD□135#
```

### 4.2 Input Variables

If limit values are specified after the variable address, this variable is first read once from the PLC and is then be released for an input process. When the Enter-key is pressed, the entered value is checked to ensure that it is within the specified limits. If the value has been

entered correctly, it will be written (back) to the PLC at the specified address.

Example:

```
@UNS□144□12□54670
```

#### 4.2.1 Indexed Variable Address

General Syntax:

```
@[ttt]□i1+k[!]□[min]□[max]
```

```
[i1] = variable address for pointer
[k] = constant
```

The actual variable address comprises the contents of [i] and the constant [k]. It must be

ensured that the sum of [i] and [k] does not exceed the valid range of values for the PLC. The index register is defined in the same message:

```
@idx□xxx!
```

i1 = the first index definition found  
i2 = the second index definition, etc.



## 4.3 Input Limits

The operating panel monitors input variables to ensure they are within an upper and a lower limit. The input limits are programmed directly after the variable address. A corresponding

error message from the text memory (message 1000 or 1001) is displayed if a value either falls below the lower input limit or exceeds the upper limit.

### 4.3.1 Indexed Input Limits

The limits can also be indexed, if the PLC program is also to modify the input limits.

```
@int□15□i1□i2
@idx□16!
@idx□17!
```

In this example, the max-limit is read from register (data word or flag word, depending on the PLC) 17 and the min-limit is read from register 16.

The input limits are read in the same variable format as the input variable (in the example as a signed 16-bit number).

Index registers are defined in the same way as indication variables. The limitation of 8 definitions per message applies here also. This means that the number of possible variables is reduced by one per index register. This applies only to the current message - all other messages may still contain the full 8 variables.

## 4.4 Variable Formats

The representation format of a variable is specified as part of the variable definition. This manual describes all of the formats supported by the BEST operating panels. Because of the

large differences between the various PLC types, each description lists the variable formats actually possible for the PLC connection.

### 4.4.1 Variable Format UNS

The defined data word is interpreted as an integer without a sign in the range of 0 ... 65535. Only the keys 0 and 9 are accepted by

the input editor. The number of filler characters must be between 1 and 5.

### 4.4.2 Variable Format INT or KF

The defined data word is interpreted as a signed integer in the range of -32767 ... +32767. The key [+/-] allows the sign to be reversed. When programming the filler

characters, it must be taken into consideration that one digit is required for the sign. The number of filler characters must be between 2 and 6.

### 4.4.3 Variable Format IL or MD

Represents a signed 32-bit number. The number of filler characters must be between 2 and 11. Two consecutive registers are

automatically read for PLC types which are only provided with 16-bit registers (data words).

#### 4.4.4 Variable Format BCD

The defined data word is interpreted as a 4-digit BCD number in the range of 0 ... 9999. The number of filler characters must be between 1

and 4. If the data word has been predefined with nibbles >>9 by the PLC, they will be represented by the characters A to F.

#### 4.4.5 Variable Format BCDL or LNG

The defined data word is interpreted as an 8-digit BCD number in the range of 0...99999999. The number of filler characters must be between 5 and 8. Two consecutive registers (data words) are automatically read for PLC

types which are only provided with 16-bit registers (data words). If the data word has been predefined with nibbles >>9 by the PLC, they will be represented by the characters A to F.

#### 4.4.6 Variable Format HEX or KH

The defined data word is interpreted as a 4-digit HEX-number in the range of 0 ... FFFF. The number of filler characters must be between 1 and 4. The keys 0 and 9 are accepted by the

input editor. The characters A ... F are produced by simultaneously pressing the key [SFT] and one of the keys 0 ... 5 (corresponds to A .. F).

#### 4.4.7 Variable Format BIT or KM

The defined data word is represented as 16 single bits. The number of filler characters must be between 1 and 16. The keys 0 and 1 are accepted by the input editor.

The bit variable is read from register 10 and represented from bit 7 onwards. The zero is a dummy parameter used to identify the bit variable as an input variable. It should be noted that the bits that are not displayed are still read and written back to the PLC unchanged when the Enter-key is pressed. This can mean that any modifications made by the PLC in the meantime are cancelled.

Example for an input variable:

```
@BIT□10□7□0
```

Example for an indication variable:

```
@BIT□10!□7
```

After the character (!), indication variables require an additional parameter for the start bit.

### 4.4.8 Variable Format KT (Siemens only)

The defined data word is interpreted as a 3-digit BCD number in the range of 0 ... 999. The highest 4 bits contain the time base coding. The key [+/-] allows the selection of a different time base. The time base with the highest resolution will automatically be implemented when storing into the data word. The number of filler characters must be 4.

Representation of the filler characters in the DW:

7.65	Time base 0	0765 (KH)
76.5	Time base 1	1765 (KH)
765	Time base 2	2765 (KH)
7650	Time base 3	3765 (KH)

Whenever an input can be represented using

the next smaller time range, it will automatically be selected: e.g. 0012 - 12.0 (1120). Within the largest range, any 4-digit number entered will automatically be rounded: 1235 - 1240 (3124).

To use the largest time range possible, the input limits must be programmed as follows:

Example:

```
@KT□12□0,01□9990
time setting: ____ s
```

Exactly four digits must be programmed for the input limits. Program either 3 digits and a comma for the 4th digit or 3 digits and a 0 for the 4th digit.

### 4.4.9 Variable Format ASC

Starting with the H-byte of the defined address, ASCII characters will be entered in place of the filler characters. The character set that is specified in the system set-up will be used (IBM, CPM). If the filler characters are assigned to the second (graphics) character set, the characters defined there will be displayed. Only displayable characters are permitted (from 20h onwards). An ASCII variable is always an indication variable and can only be programmed once

within a message. The number of filler characters is limited to the width of one line. It is not permitted to display an ASCII variable and to print an ASCII variable at the same time.

Example:

```
@ASC□12!
Model no. _____
```

### 4.4.10 Variable Format TXT

The variable format TXT is used to display text in a variable field instead of a number. The number of filler characters can range between 1 and 40. As with the other variable formats, the field of the filler characters is limited to the width of one line. The key [+/-] can be used to page through the fixed texts. The message number belonging to the text displayed, is obtained from the contents of the displayed data word and the specified min-limit.

If texts using the 2nd character set are used, the reference characters must be defined in the range of 20h ... 7Fh.

Example:

```
@TXT□12□760□765
selected color: □_____
```

760: red  
761: green  
762: yellow  
763: blue  
764: white  
765: black

The key [+/-] can now be used to page through messages 760 to 765. The selected color is represented in the data word 12 as a number between 0 ... 5.

If the variable format TXT is to be used as indication variable, a min-limit must be specified after the character "!" in order to identify the starting point in the text memory.

Example:

```
@TXT□12!□760  
selected color
```

## 4.5 Scaled Variables

Scaling of variables is necessary whenever there is a discrepancy between the internal range of values and the range of numbers to be displayed.

Example:

A temperature measuring sensor supplies an analog signal of 4..20mA. To obtain optimum resolution, the D/A-converter converts this value to a numerical value of 205...1023. Physically this corresponds to a temperature of 0..100 degrees.

This scaling or standardization of values can be achieved by utilizing the variable formats "SUNS" or "SINT".

The 4 corner points as depicted in the standardization scheme are inserted in the line following the variable definition:

```
@SUNS□25!  
@0□100□205□1023
```

The indication variable is read from register 25. The scaling range is specified next.

When selecting the range for a scaled display, it must be taken into consideration that the number of measuring points cannot be increased and that the ratio of display and input should not exceed 1:31 or 31:1 respectively. In practice, this is not restrictive because inserting a comma allows the range to be extended. The scaling factor is calculated as follows:

$$\text{Factor} = \frac{(\text{display\_max} - \text{display\_min}) * 2048}{(\text{input\_max} - \text{input\_min})}$$

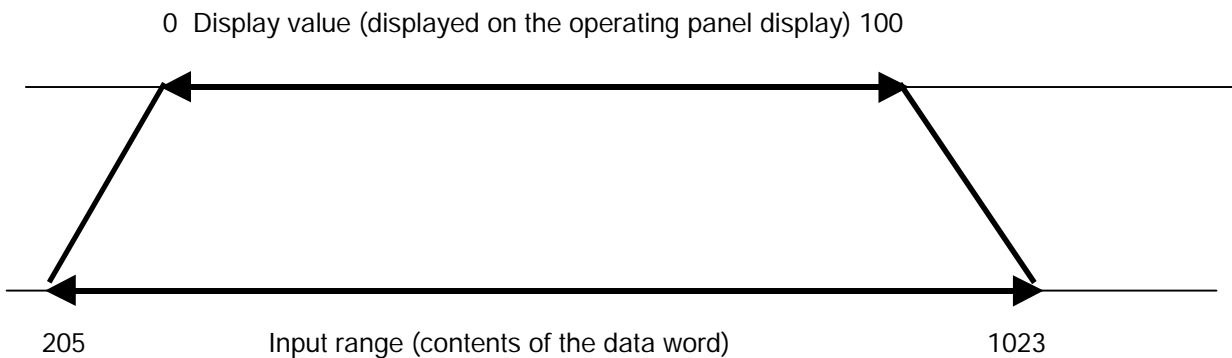
The factor is rounded off to a whole number.

The display value of a scaled variable is calculated as follows:

$$\text{Display} = \frac{(\text{E\_value} - \text{E\_min}) * \text{Factor}}{2048} + \text{B\_beg}$$

The display value is rounded off to a whole number.

### Standardization Scheme:



### 4.5.1 Scaled UNS

Example:

```
@SUNS□25!  
@0□100□205□1023
```

In the case of the variable format SUNS, an unsigned range of numbers is mapped to another unsigned range of numbers. The range of numbers comprises the numbers 0 .. 65535.

### 4.5.2 Scaled INT

Example: Indication variable

```
@SINT□25!  
@-50□+100□-200□1023
```

Signed INT numbers are used. The input range of -200 ... +1023 (=1223 measuring points) is mapped to the range -50 .. +100 (=150 measuring points). The range of numbers comprises the numbers -32767 ... +32768.

Example: Input variable

```
@SINT□25□-20□+100  
@-50□+100□-200□1023
```

The input limits apply to the input range, not to the range of values which appears in the PLC.

## 5. System Messages in the Text Memory

Part of the text memory is used to allow translation of the system and error messages. The messages from message 1000 onwards are system messages. The following are recommended texts and their meaning:

#### 1000

```
Variable _ too small  
Enter at least _____ !  
@  
Press the Enter-key to continue
```

Message 1000 is displayed if a variable is entered that is smaller than the programmed minimum limit. The variable is inserted in place of the first filler character, the limit value is inserted in place of next filler characters. If the BCDL/LNG-format or IL/MD-format (2 data words) is used, at least 11 filler characters are required for the limit value.

#### 1001

```
Variable _ too large  
Enter at maximum _____ !  
@  
Press the Enter-key to continue
```

Message 1001 is displayed if a variable is entered that exceeds the programmed maximum limit. The variable is inserted into the first variable field (1 filler character), the limit value into the next variable field. If the BCDL/LNG-format (2 data words) is used, at least 11 filler characters are required to represent the limit value.

#### 1002

```
Any Messages > 999 are system messages
```

Message 1002 is displayed if the PLC attempts to call up a message >999.

#### 1005

```
PU connection: On=Input Off=0
```

Message 1005 is used to switch the programming unit (PU) connection on or off. This message is only relevant for PLC connections with an integrated **PG-MUX**.

**1006**

Communications cycles \_\_\_\_\_  
\_\_\_\_\_ of which failed

This message is called up by simultaneously pressing the keys SFT (>>) and i. This is where the telegrams between the PLC and BEST are counted and evaluated. The counters are deleted by applying the operating voltage. The message assists in determining the functioning of the data line between the PLC and the BEST. Data line faults do not cause malfunctions but are corrected by repeating the telegram. Each of the two variable fields contains 5 filler characters.

**1007**

Too many @-lines in message: \_\_\_\_\_

More than 8 variable definitions were detected. Four filler characters are required to represent the message number.

**1008**

No text string in message: \_\_\_\_\_

The selected message image does not contain a text string. The variable field contains 4 filler characters.

**1009**

Syntax error in msg: \_\_\_\_\_  
\_\_\_\_\_

A syntax error has been detected in the message specified in the first variable field (4 filler characters). The line containing the error is displayed in the 2nd variable field (24 filler characters).

**1010**

Wrong format M: \_\_\_\_\_  
\_\_\_\_\_

The variable format could not be detected in the message no. specified in the first variable field (4 filler characters). The line containing the error is displayed in the 2nd variable field (24 filler characters).

**1011**

Wrong number of VA-filler characters  
In messages: \_\_\_\_\_

An incorrect number of filler characters holding the place for variables has been detected in the first variable field (4 filler characters). The number cannot be displayed correctly.

**1015**

PLC data handling block  
Without log print-out

(Siemens only)

**1021**

PLC not in "RUN" or connection  
SAS 525 <-> PLC not correct

The PLC is not in the "RUN" mode or the SAS525 interface card is not addressed by the PLC (Siemens only). Display or input operations are therefore not possible.

**1022**

Interface on PLC in use by PU

The programming unit (PU) is accessing the PLC. A correct display is therefore not possible. This message is only required for PLC connections which are provided with an integrated interface switch in place of the PG-MUX.

**1023**

Connection error: \_\_\_\_\_

This message is displayed if the data transmission has not been carried out correctly despite the telegram being repeated. The operating panel continues to try to establish a connection. Possible causes:

- severe line faults or line interruption
- the cycle time is too long.

Four filler characters are required.



### **System Messages for SAS 525 and communication module only**

The following messages are listed for reasons of completeness:

#### **1016**

Incorrect response from ST \_\_\_  
ST no. may be assigned twice

#### **1017**

BUS test mode:

#### **1018**

Wrong function ST\_\_\_

#### **1019**

PLC reports error  
status: \_\_\_

#### **1020**

Read block too large! \_\_\_ DW  
as of DW \_\_\_ max. \_\_\_DW

## **6. Printer Connection - Text Programming for the Printer**

Most PLC connections allow a serial printer to be connected to the COM1 interface of the operating panel. Any messages that are displayed can also be printed, independent of the display process. The line width of the text printout is the same as that used to display the

texts on the operating panel (24 or 40 characters). Since certain control sequences may need to be inserted into the text (bold print, form feed, cut instruction etc.) to allow printer control, we have provided appropriate options.

```
@UNSQ17!  
#1BA12  
#0F  
Demo - printout#0E  
Test#n  
Speed ___ km/h
```

```
Variable definition  
Printer initialization (ESC A 1 2)  
Bold print ON (printer-specific)  
Text + sequence for bold print OFF  
Text + LF suppression  
Text with filler characters in place of variables
```

The text is printed as follows:

```
Demo-printout  
Test speed 78 km/h
```

The character "#" becomes a control character for the printer channel:

```
##      representation of the character "#"
#n      line feed is disabled
#xx     xx= 2-digit Hex-ASCII representation of the binary value
        (e.g. #1B for ESC)
```

If such a message appears in the display, the control sequences are displayed as text. The printer control is independent of the current display. The PLC-specific section contains a description of the handling with the PLC.

The printer must be activated in the system set-up (\*printer).

## 6.1 Printer Character Set

The text memory contents is sent to the printer unchanged. This means that the presetting (\*CPM/\*IBM) applies only to the display. If text variables are printed, the characters between 20h and 9Fh are permitted. The characters

from 20h to 7Fh can be used as reference characters for the second character set. The user-defined characters of the 2<sup>nd</sup> character set are printed in the IBM graphics mode. A suitable printer must therefore be used.

## 6.2 Printer Hardware

Any printers with RS232-interfaces can be used. The baud rate is 9600. The printer must

be capable of temporarily storing 2 characters after the handshake line is deactivated.