KeTop T40

Handheld Terminal

KEBA Standard Protocol

User’s Manual V 1.1

KEBA®

Automation by innovation.

Notes on This Manual

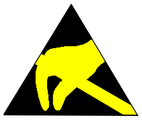
At various points in this manual you will see notes and precautionary warnings regarding possible hazards. The meaning of the symbols used is explained below.

⚠ DANGER
● **DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.**

⚠ WARNING
● **WARNING indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.**

⚠ CAUTION
● **CAUTION indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.**

CAUTION
● **CAUTION used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property injury.**



This symbol reminds you of the possible consequences of touching electrostatically sensitive components.

Note

Notes on use of equipment and useful practical tips are identified by the “Notice” symbol. Notices do not contain any information that draws attention to potentially dangerous or harmful functions.

Document: version 1.1 / material no.: 61869
Filename: t40_stdprot_en.doc, last saving on: 10. 12. 2004

KEBA AG, Postfach 111, Gewerbepark Urfahr, A-4041 Linz
Tel.: ++43 / 732 / 70 90-0, Fax: ++43 / 732 / 73 09 10, E-Mail: keba@keba.com, www.keba.com

KEBA GmbH, Ulmer Straße 123, D-73037 Göppingen
Tel.: ++49 / 7161 / 97 41-0*, Fax: ++49 / 7161 / 97 41-40

KEBA Corp., 100 West Big Beaver Road, Troy, MI 48084
Tel. ++1 / 248 / 526 - 0561, Fax: ++1 / 248 / 526 - 0562, E-Mail: schr@us.keba.com

History

Modification from / to	Date	Modified pages	Description	Author
V1.0	July 02		First edition	sam
V1.0/V1.1	12-2004	all	New Layout	sam

Contents

1 Introduction

This document is a **supplement** to the user's manual "KeTop T40 Handheld Terminal – General Information" and exclusively refers to the KEBA standard protocol.

The following chapters describe the communication between the KeTop T40 handheld terminal and a PLC using the KEBA standard protocol.

The data exchange between the HT and the PLC takes place via commands that are explained in this manual on the basis of examples.

This manual does not describe in detail the configuration, the basic functions and the key labelling of the KeTop. For this information, please refer to the manual "KeTop T40 Handheld Terminal – General Information".

2 Connection

The KEBA standard protocol must be operated via a serial interface of the handheld terminal.

For details about the connection of the interfaces, refer to the corresponding chapter in the manual "KeTop T40 Handheld Terminal – General Information".

3 Configuration of KeTop T40

Configuration Software

For setting the device configuration and creating the texts, KEBA supplies a configuration software that is executable under Windows.

The configuration is described in detail in the manual "KeTop T40 Handheld Terminal – General Information" (e.g. functions for editing the keypad layout and for loading the program) and in general also applies to the KEBA standard protocol coupling.
Therefore this chapter only describes the specific details of the KEBA standard protocol.

Configuration

Selection of protocol

After selecting the "**serial KEBA standard protocol**" the following protocol parameters must be set:

Baudrate	Selection of requested transmission rate: 9600, 19200, 38400, 57600, 115200 Baud.
Parity	The setting of parity is EVEN and cannot be changed.
Data bit	The number of data bits is 8 and cannot be changed.
Stop bit	The number of stop bits is 1 and cannot be changed.

Numbering of variables

For the serial data transmission of the KeTop, the following **restriction** must be taken into account in the text editing mode:

User variables may be defined only from variable number 100 on. The range from 1 to 99 is reserved for the system variables (see also chapter "KeTop System Variables").

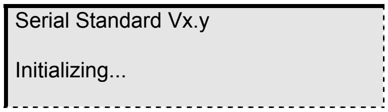
EXCEPTION: string variables (numbering from 0 to 255)

4 Power-Up Phase

The handheld terminal carries out a self-test after turning on. For details about the test steps, refer to the manual "KeTop T40 Handheld Terminal – General Information".

- ▶ The first part of the test is identical for each KeTop T40 coupling and thus described in the User's Manual "KeTop T40 Handheld Terminal – General Information".
- ▶ Then the configuration data are loaded.

The following message is displayed:

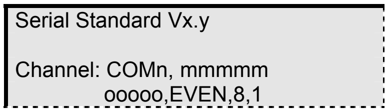


```
Serial Standard Vx.y

Initializing...
```

In case of an error, the handheld terminal remains in this condition.

- ▶ After the configuration data have been loaded successfully, the following mask appears on the display:



```
Serial Standard Vx.y

Channel: COMn, mmmmm
         ooooo,EVEN,8,1
```

x	program version
n	number of interface port
m	type of interface (RS232 or RSxx2 for RS232/RS422)
o	baud rate (9600, 19200, 38400, 57600, 115200 Baud)

This mask remains on the display until the PLC activates a text display on the handheld terminal.

5 KEBA Standard Protocol

The HT is used in industrial applications, especially for operating machines. Errors in the HT or during data transmission may endanger men and machine. Therefore higher demands are made on the data transmission compared to a standard ASCII terminal, for example.

Features

- Transmission errors can be recognized reliably through a horizontal and vertical parity check.
- The HT acknowledges whether a message has been received correctly or not. In case of negative acknowledgement, the PLC must repeat the message.
- The acknowledgement enables the PLC to control the time. If the response expected from the HT does not occur, the HT, the connection cable or the interface of the PLC are probably defective.

Structure of Messages

Each message is started with the control character STX. Then the command and the data are transmitted. The control character ETX terminates the data part. By means of the BCC checksum sent with the message, the receiver checks if the transmission has been completed successfully.

Depending on the command, we distinguish between messages with fixed length and messages with variable length:

Messages with fixed length

- The data part may also contain the characters STX (02H) and ETX (03H).

Messages with variable length

- The data part only contains printable characters (ASCII code ≥ 20H), i.e. the beginning and the end of a message are clearly defined by the control characters STX, ETX and BCC.

Control Characters

Control character	HEX	Description
EOT	04	The HT sends this control character after the power-on self-test or after a reset command.
STX	02	Start of data transmission
ETX	03	End of data transmission
ACK	06	The HT has correctly understood the message it received (correct checksum).
NAK	15	HT has not correctly understood the message it received (wrong checksum). The send station must repeat the message.

Checksum Calculation (BCC)

The checksum is calculated in accordance with DIN 66219. It always consists of one byte. The following example describes the checksum calculation of the message for activating the buzzer:

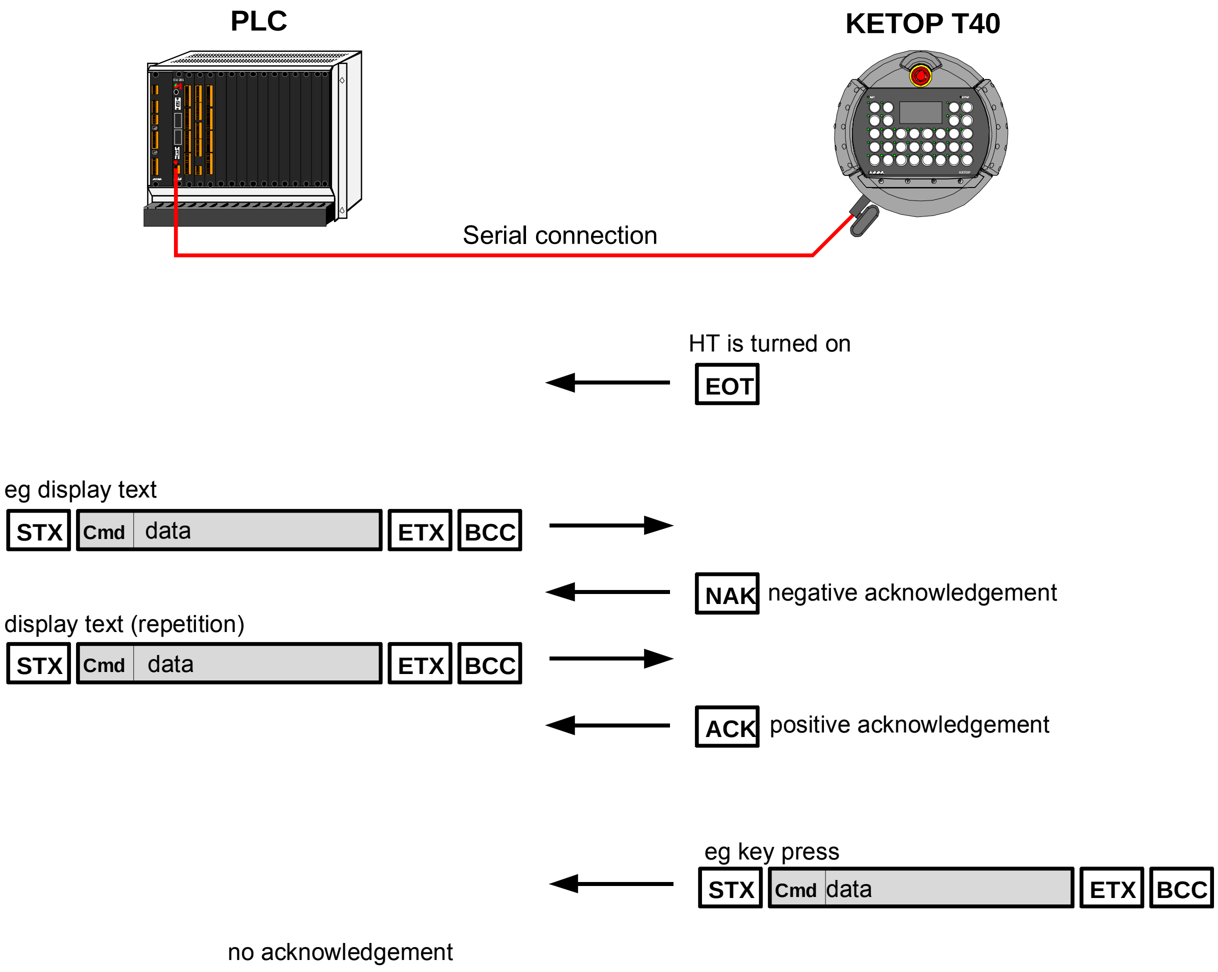
E.g.: PLC (control) sends to HT (handheld terminal):

Buzzer activation	STX	<command>	<duration>	ETX	BCC
	02	*07*	*35*	*03*	*31*

Character No.	Character	HEX	BINARY	Keys on VT100 terminal
1	STX	*02*	00000010	CNTL B
2	Command	*07*	00000111	CNTL G
3	Duration	*35*	00110101	5
4	ETX	*03*	00000011	CNTL C
5	BCC	*31*	00110001	1

The BCC checksum is calculated by the XOR logic of all characters beginning after STX and ending before the BCC. In this example the character 5 is calculated by the XOR logic of the characters 2, 3 and 4.

Protocol Sequence



Cmd ... Command

Example of communication between PLC and HT

Overview of Messages

	PLC ---> HT		HT ---> PLC	
Function	Cmd	Data	Cmd	Data
Keyclick on	‘e’	-		
Keyclick off	‘f’	-		
Config.: signalize pressing of key	‘a’	-		
Config.: signalize releasing of key	‘b’	-		
Event “key pressed"			‘(‘	[key code]
Event “key released"			‘)’	[key code]
1-line text call	‘p‘	[line] (text number)		
4-line text call	‘q’	(text No. 0) ... (text No. 3)		
8-line text call	‘Q’	(text No. 0) ... (text No. 7)		
16-line text call	‘P’	(text No. 0) ... (text No. 15)		
Clear display	‘g’	-		
Position cursor	‘C’	[line] [column]		
Cursor attribute	‘E’	[attribute]		
Text attribute (8-/16-line LCD)	‘H’	[attribute]		
Text from cursor position	‘B’	"text..."		
Text line 0	‘0’	"text..."		
Text line 1	‘1’	"text..."		
Text line 2	‘2’	"text..."		
Text line 3	‘3’	"text..."		
Text line 4	‘4’	"text..."		
Text line 5	‘5’	"text..."		
Text line 6	‘6’	"text..."		
Text line 7	‘7’	"text..."		
Text xy (ASCII)	‘A’	"line" "column" "text..."		
Text xy (binary)	‘T’	[line] [column] "text..."		
Set timeout	‘t’	[duration]		
Buzzer activation	BEL	"duration"		
Reset (warm start)	‘d’	-		
Status configuration	‘S’	[status sending time] [reserve (5)] [status content]		
Status			‘J’	<..configured contents...>
Request key status	‘h’	-	-	[G1][G2][G3][G4]
Request code of all keys pressed	‘c’	-	-	n * [key code] (0≤n≤5)
Red LED map	‘L’	[G1][G2][G3][G4][G5]		
Green LED map	‘M’	[G1][G2][G3][G4][G5]		
Red flashing LED map	‘F’	[G1][G2][G3][G4][G5]		
Green flashing LED map	‘G’	[G1][G2][G3][G4][G5]		
LED on red	‘+’	[LED code]		
LED on green	‘*’	[LED code]		
LED flashing red	‘%’	[LED code]		
LED flashing green	‘&’	[LED code]		
LED off	‘-’	[LED code]		
Graphic functions:				
Pixel	‘Z’‘P‘	(x) (y)		
Line	‘Z’‘L‘	(x0) (y0) (x1) (y1)		
Rectangle	‘Z’‘R‘	(x0) (y0) (x1) (y1) [fill]		
Circle	‘Z’‘C‘	(x) (y) (r) [fill]		
Ellipse	‘Z’‘E‘	(x) (y) (rx) (ry) [fill]		
Text	‘Z’‘T‘	(x) (y) [fontsize] [font_attr] „text....“		
Line type	‘Z’‘l‘	(type)		
Clear window	‘Z’‘c‘	(x0) (y0) (x1) (y1)		
Set colors	‘Z’‘o‘	[fg] [bg]		
Display of bitmaps	‘Z’‘B‘	(bitmap No) (x) (y)		

	PLC <---> HT		HT <---> PLC	
Function	Cmd	Data	Cmd	Data
Request 1-byte variable	‘u’	(varNo)	‘U’	(varNo) [Wert]
Request 2-byte variable	‘v’	(varNo)	‘V’	(varNo) (Wert)
Request 4-byte variable	‘w’	(varNo)	‘W’	(varNo) {Wert}
Request floating point variables	‘x’	(varNo)	‘X’	(varNo) {Wert}
Request string variables	‘y’	[varNo] [length]	‘Y’	[varNo] [length] "value"
Send 1-byte variable	‘U’	(varNo) [value]		
Send 2-byte variable	‘V’	(varNo) (value)		
Send 4-byte variable	‘W’	(varNo) {value}		
Send floating point variable	‘X’	(varNo) {value}		
Send string variable	‘Y’	[varNo] [length] "value"		

Cmd	**Command**
" "	value in ASCII coding
[ ]	8-bit value in binary coding
( )	16-bit value in binary coding
{ }	32-bit value in binary coding

All values are transmitted in the Motorola format (high byte before low byte).

6 KeTop System Variables

System variables offer additional functions to the user. If the system variables are sent with the corresponding index they can activate/deactivate certain functions in the handheld terminal (e.g. automatic sending of key code).

The system variables range from 1 to 99 and must not be programmed by the user.

At the system variables, we distinguish between 1-byte, 2-byte and 4-byte variables.

Var No.	Designation	Data type	Access		Default value	Description
			HT	PLC		
1-byte variables						
1 (01H)	Reserved	-	-	-	-	-
2 (02H)	Reserved	-	-	-	-	-
3 (03H)	Reserved	-	-	-	-	-
4 (04H)	Reserved	-	-	-	-	-
5 (05H)	Reserved	-	-	-	-	-
6 (06H)	BEEP	BOOL		R/W	FALSE	Buzzer activation (permanent tone)
7 (07H)	KEY_CLICK_PRESSED	BOOL		R/W	FALSE	Beep when pressing a key
8 (08H)	SEND_PRESSED_KEY_NR	BOOL		R/W	FALSE	The HT sends the key number when a key is pressed.
9 (09H)	SEND_RELEASED_KEY_NR	BOOL		R/W	FALSE	The HT sends the key number when a key is released.
10 (0AH)	Reserved	-	-	-	-	-
11 (0BH)	Reserved	-	-	-	-	-
12 (0CH)	Reserved	-	-	-	-	-
13 (0DH)	SEND_KBD_MAP	BOOL		R/W	FALSE	Will not be evaluated.
14 (0EH)	Reserved	-	-	-	-	-
15 (0FH)	Reserved	-	-	-	-	-
16 (10H)	Reserved	-	-	-	-	-
17 (11H)	SYSTEM_RESET	UINT8		W	-	System reset at bitmap 55H
18 (12H)	FETCH_TEXT_VAR	BOOL		W	FALSE	Upon text calls, the HT requests once all output variables contained in the text.
19 (13H)	SEND_PRESSED_KEY_CODE	BOOL		R/W	FALSE	Will not be evaluated.
20 (14H)	ORDER_BOX	UINT8		W	-	Will not be evaluated.
21 (15H)	KEY_CLICK_RELEASED	BOOL		R/W	FALSE	Beep when releasing a key

Var No.	Designation	Data type	Access		Default value	Description
			HT	PLC		
22 (16H)	EVENT_VAR_NOT_ON_SCREEN	BOOL		R/W	FALSE	Event message when a variable is written, for which no output field is provided on the display => EVENT_CODE
23 (17H)	DISABLE_MENU	BOOL		R/W	FALSE	Enabling(0) or disabling(1) the main menu call (pressing the 1st and 4th key)
24 (18H)	DISABLE_EDITOR	BOOL		R/W	FALSE	Enabling(0) or disabling(1) the input of variables on the HT
25 (19H)	Reserved	-	-	-	-	-
26 (1AH)	Reserved	-	-	-	-	-
27 (1BH)	Reserved	-	-	-	-	-
28 (1CH)	EVENT_MENU	BOOL		R/W	FALSE	Event message upon menu access and exit => EVENT_CODE
29 (1DH)	KBD_LOCKED	BOOL		R/W	FALSE	Enabling (0) or disabling (1) of the keypad on the HT
30 (1EH)	NEXT_FIELD_AFTER_ENTER	BOOL		R/W	TRUE	After Enter is pressed, the cursor changes to the next input field.
31 (1FH)	SEND_RELEASED_KEY_CODE	BOOL		R/W	FALSE	Will not be evaluated.
32 (20H)	AUTO_KEY_REPEAT	BOOL		R/W	FALSE	Automatic key repetition when key is pressed.
2-byte variables						
1 (01H)	Reserved	-	-	-	-	-
2 (02H)	FETCH_TEXT_VAR_CYCLE_TIME	UINT16		R/W	0	Cycle time in ms at which the HT requests all output variables displayed.
3 (03H)	BEEP_TIME	UINT16		R/W	0	Buzzer activation (duration of beep in ms)
4 (04H)	APPLICATION_SW_VERSION	UINT16		R	-	Version of HT application software Vx.y: High Byte -> x, Low Byte -> y
5 (05H)	DEVICE_TYPE	UINT16		R	-	40 ... KeTop T40
6 (06H)	Reserved	-	-	-	-	-
7 (07H)	Reserved	-	-	-	-	-
8 (08H)	AUTO_KEY_DELAY_TIME	UINT16		R/W	250	Repetition rate in ms.
4-byte variables						
1 (01H)	ERROR_CODE	UINT32	W	R		Error message HT -> PLC (not used at present)
2 (02H)	EVENT_CODE	UNIT32	W	R		Event message HT -> PLC: Bits 24 - 31: event number Bits 0 - 23: event-dependent information

Access

R	Read
W	Write

Values ranges of data types used

BOOL	Boolean number (0 = FALSE, 1 = TRUE)
UINT8	integral number without sign (0...255)
UINT16	integral number without sign (0...65535)
UINT32	integral number without sign (0...4294967295)

7 KeTop Functions

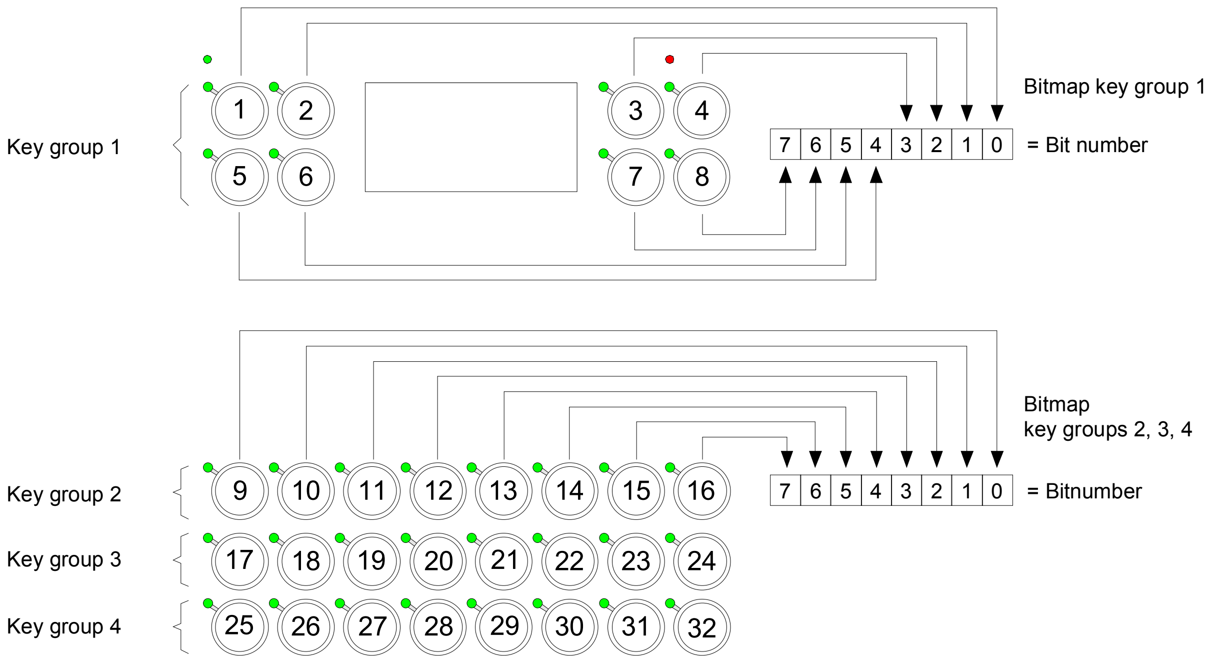
In this chapter, the commands are first represented in ASCII, and in the second line in HEX in *italics*:

Function	in ASCII:	STX	<command><data>	ETX	BCC
Transmission direction (->) (PLC ... control) (HT... handheld terminal)	*in HEX:*	*02*	*...*	*03*	*...*

Keypad

The following drawing shows the physical grouping of the HT keys and the assignment of the particular keys to the bits of the parameter "key group n". This grouping and assignment is identical for the LEDs of the keypad.

Physical grouping



Key groups of KeTop T40 keypad

Key Status

Request key status	STX	'h'	ETX	BCC
PLC -> HT	*02*	*68*	*03*	*6B*

This command requests from the KeTop T40 the bit-coded map of all keys pressed at the moment.

Response	STX	<bitmap key group 1> <bitmap key group 2> <bitmap key group 3> <bitmap key group 4>	ETX	BCC
HT -> PLC	*02*	*...*	*03*	*...*

This response contains the bit-coded map of the KeTop T40 keypad.

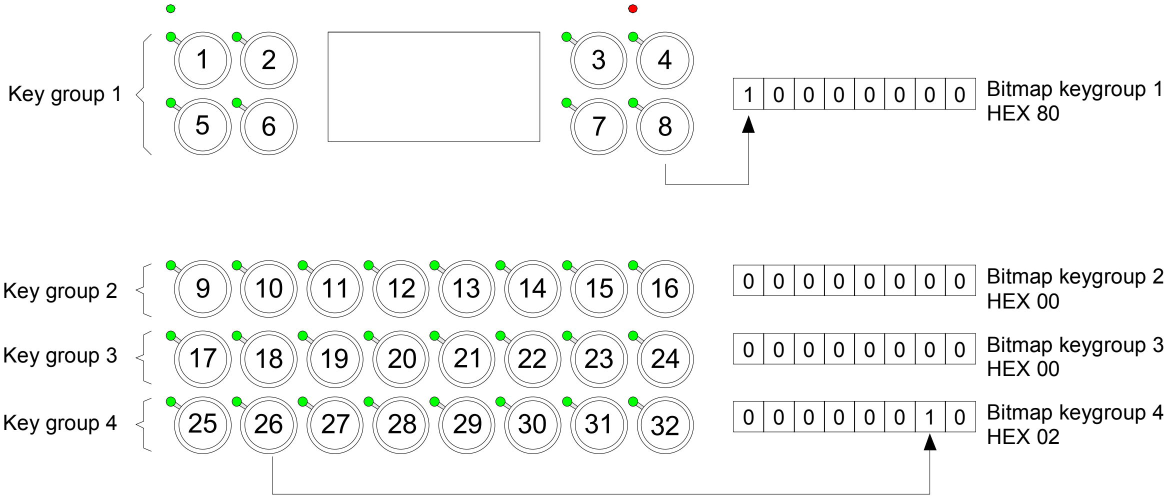
Example

PLC -> HT:

STX	'h'	ETX	BCC

HT -> PLC:

STX	<bitmap key group 1> to <bitmap key group 5>	ETX	BCC
02	*80 00 00 02 00*	*03*	*81*



Request codes of all keys pressed	STX	‘c’	ETX	BCC
PLC -> HT	*02*	*63*	*03*	*60*

This command requests from the KeTop T40 the key codes of all keys pressed at the moment.

Response HT:	STX	{< key code >}	ETX	BCC
HT -> PLC	*02*	*...*	*03*	*...*

The data part of this response contains in ascending order the codes of all keys pressed at the moment.

Example

When the keys ‘0’, ‘M’ and ‘^’ are pressed simultaneously, the KeTop T40 sends the following response:

HT -> PLC:

STX	‘0’ ‘M’ ‘^’	ETX	BCC
02	*30 4D 5E*	*03*	*20*

If no key is pressed no response will be sent.

Keyclick on **Keyclick off**	STX	'e' 'f'	ETX	BCC
PLC -> HT	*02*		*03*	

After receiving the ‘e’ command, the HT acknowledges every pressed key with an audible signal. The ‘f’ command deactivates the keyclick again.

‘e’	activate keyclick
‘f’	deactivate keyclick

Configuration

Signalize pressing of key **Signalize releasing of key**	STX	$\begin{bmatrix}'a'\\'b'\end{bmatrix}$	ETX	BCC
PLC -> HT	*02*	$\begin{bmatrix}61\\62\end{bmatrix}$	*03*	$\begin{bmatrix}62\\61\end{bmatrix}$

After receiving one of these commands the HT sends the key code each time a key is pressed and/or released.

‘a’ send key code each time a key is pressed
‘b’ send key code each time a key is released

After receiving the ‘a’ command, the HT will send the event message „key pressed“ each time a key is pressed.
After receiving the ‘b’ command, the HT will send the event message „key released“ each time a key is released.
Thus the handheld terminal becomes the active station in both cases and sends the key code without corresponding request from the PLC.

Event

Key pressed **Key released**	STX	$\begin{bmatrix}'('\\')'\end{bmatrix}$	<key code>	ETX	BCC
HT -> PLC	*02*	$\begin{bmatrix}28\\29\end{bmatrix}$	...	*03*	...

‘(‘ The key code of the pressed key is sent to the PLC (after initialization by the ‘a’ command).
‘)’ The key code of the released key is sent to the PLC (after initialization by the ‘b’ command).

Example

PLC -> HT:

STX	'b'	ETX	BCC
02	*62*	*03*	*61*

Assumption:
On the KeTop T40, the 2nd key from the left in the second row (character 5) is pressed. Each time this key is released, the KeTop sends the following message to the PLC:

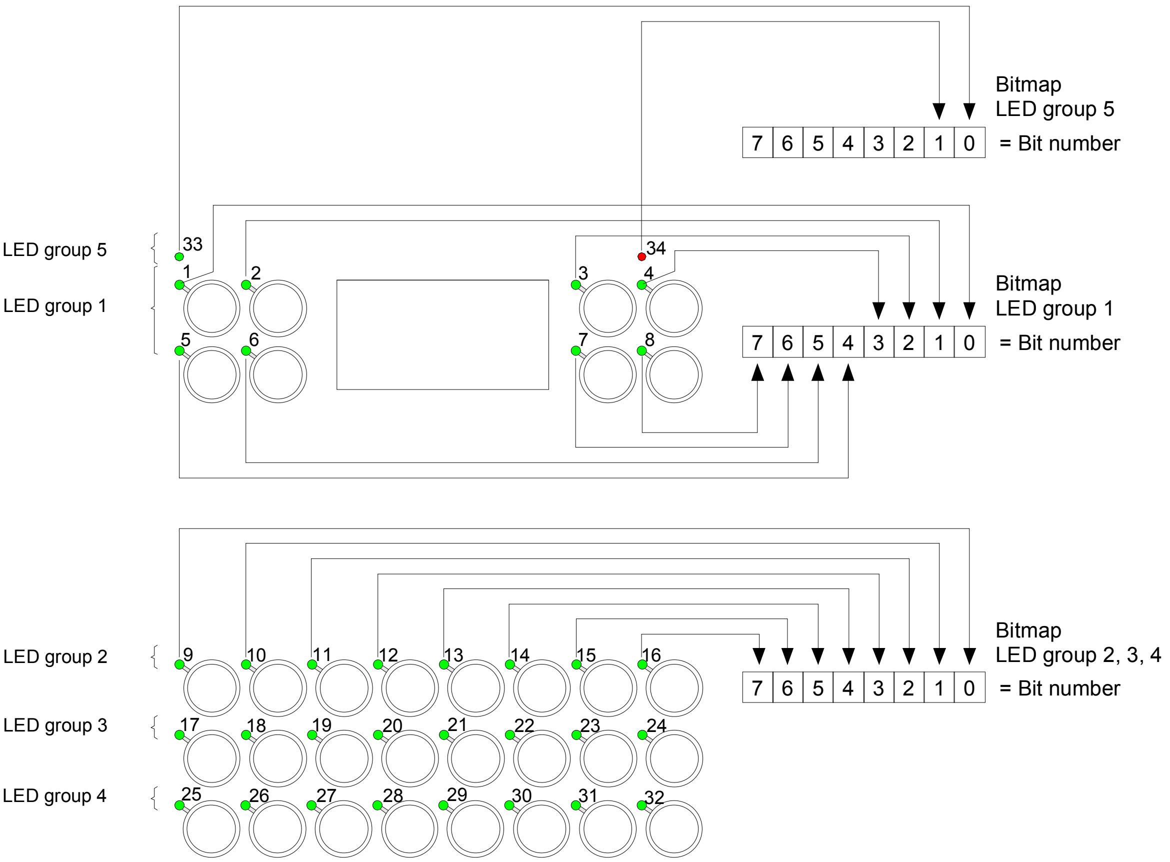
HT -> PLC:

STX	')'	5	ETX	BCC
02	*29*	*35*	*03*	*1F*

LEDs

The following drawing shows the physical grouping of the LEDs on the KeTop T40 and the assignment of the individual LEDs to the bits of the parameter „LED group x“.

Physical grouping



LED groups on KeTop T40 keypad

Notice
The LEDs of the keys „Shift Lock", „Ctrl Lock" and „Alt Lock" are activated internally by the HT and cannot be switched by the PLC.

LED bitmap **LED flashmap**	STX	'L' 'M' 'F' 'G'	<bitmap LED group 1> <bitmap LED group 2> <bitmap LED group 3> <bitmap LED group 4> <bitmap LED group 5>	ETX	BCC
PLC -> HT	*02*	*4C* *4D* *46* *47*	...	*03*	...

These commands switch on/off the red LED (No. 34) and all green LEDs on the KeTop T40 according to the parameters <bitmap LED group 1> to <bitmap LED group 5> or switch these LEDs to a flashing mode (the flashing frequency is 2 Hz).

‘L’ or ‘M’	switch on all LEDs set to 1
‘F’ or ‘G’	switch all LEDs set to 1 to the flashing mode

Important

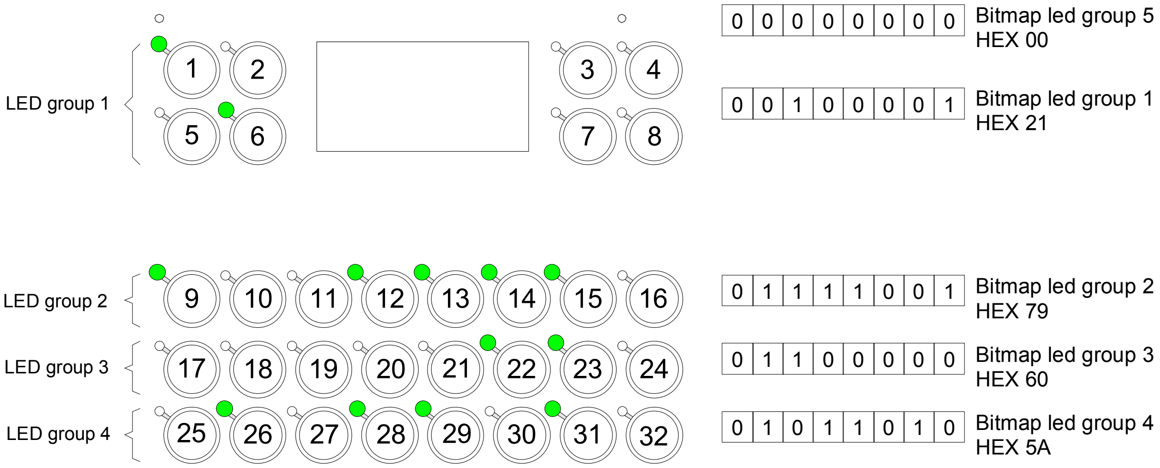
- All 5 groups must be specified even if one group does not have any LED activation.

The commands 'F' and 'G' for switching the LEDs to the flashing mode have the priority over the commands 'L' and 'M'.

Example

PLC -> HT:

STX	'M'	<bitmap LED group 1> to <bitmap LED group 5>	ETX	BCC
02	*4D*	*21 79 60 5A 00*	*03*	*2C*



LED on / off / flashing	STX	'+' '*' '_' '%' '&'	<LED No.>	ETX	BCC
PLC -> HT	*02*	*2B* *2A* *2D* *25* *26*	...	*03*	...

According to the LED No., the LEDs can be switched on, off or to the flashing mode:

'+' or '*'	switch on LED
'-'	switch off LED
'%' or '&'	let LED flash, flashing frequency 2 Hz

Example

PLC -> HT:

STX	'*'	"	ETX	BCC
02	*2A*	*22*	*03*	*5A*

On the KeTop T40, the ASCII character " (decimal 34) switches the LED with the number 34.

Texts

Using the programming software the texts are created on a PC and loaded into the handheld terminal. The PLC can call texts line by line by sending the text number and the line number to the HT where these numbers will be displayed.

Furthermore the handheld terminal offers the possibility to display and alter variables directly on the device. In the programming software, input and output fields for the variables can be defined in the text lines.

The KeTop T40 provides memory for variables and texts of 192kBytes.

Memory needed

Text	24 bytes
Variable	6 bytes

If 5000 texts are created, for example, there will remain space for approx. 12500 variables.

Text Variables

The numbers of text variables range from 100 - 65535 (for system variables, the range from 0 to 99 is reserved). We distinguish between the following variables:

- Input variables (HT -> PLC)
- Output variables (PLC -> HT)
- Input/output variables (PLC <-> HT)

Text call	STX	‘p’ ‘q’ ‘Q’ ‘P’	<line> <text number> <textNo-H line 0> <textNo-L line 0> to <textNo-H line 3> <textNo-L line 3 > <textNo-H line 0> <textNo-L line 0> to <textNo-H line 7> <textNo-L line 7> <textNo-H line 0> <textNo-L line 0> to <textNo-H line 15> <textNo-L line 15>	ETX	BCC
PLC -> HT	*02*	*70* *71* *51* *50*	... ...	*03*	...

The PLC calls the texts from the internal text memory of the HT. If the number FFFFH is specified as text number the corresponding display line will not be overwritten.

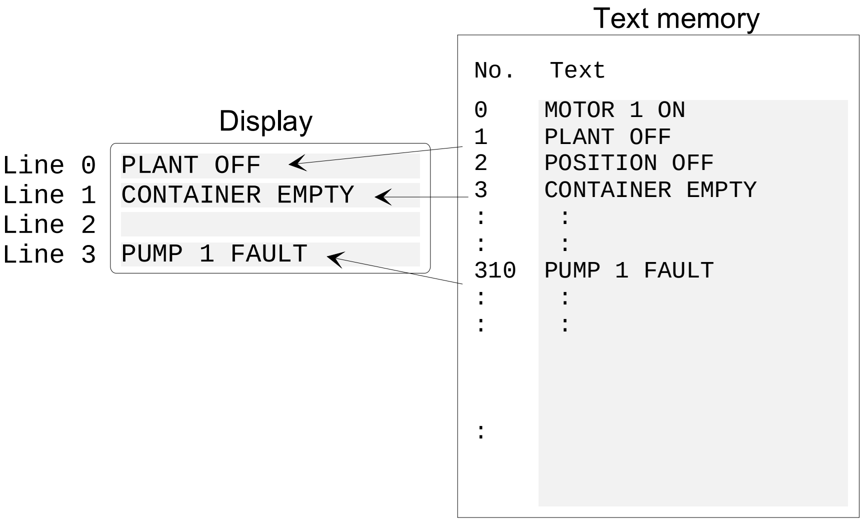
The commands 'q' and 'Q' must always contain the text numbers for all 4, 8 or 16 lines.
If the text call contains more lines than are available on the display only the first lines of the text call will be displayed.

‘p’	1-line text call
‘q’	4-line text call
‘Q’	8-line text call
‘P’	16-line text call
TextNo-H	text number, high byte
TextNo-L	text number, low byte

Example

PLC -> HT:

STX	‘q’	<TextNo-H line 0 > to <TextNo-L line 3>	ETX	BCC
02	*71*	*00 01 00 03 FF FF 01 36*	*03*	*47*



Text Monitor

With the exception of the ‘g’ command („Clear display“), all text monitor functions work independently of the text functions.

Notice
If a text mask with input or output fields is activated on the display before a text monitor function is called, the display will be overwritten. The stored cursor positions for the input and output fields are preserved. => Risk of inconsistency of display.

Clear display	STX	‘g’	ETX	BCC
PLC -> HT	*02*	*67*	*03*	*64*

Deletes all characters displayed at the moment.

Position cursor	STX	‘C’	<line> <column>	ETX	BCC
PLC -> HT	*02*	*43*	*...*	*03*	*...*

Range: 8-line LC display	(0/0) to (7/19)

Cursor attribute	STX	‘E’	<attribute>	ETX	BCC
PLC -> HT	*02*	*43*	*...*	*03*	*...*

Attribute:	0H	cursor not visible
	FFH	cursor visible

Text attribute	STX	‘H’	<attribute>	ETX	BCC
PLC -> HT	*02*	*48*	*...*	*03*	*...*

The text attribute specifies how texts are represented on the display. The following settings are allowed:

Attribute:	01H	normal display
	02H	inverse display
	05H	normal flashing display
	06H	inverse flashing display

Text from cursor position	STX	‘B’	<text>	ETX	BCC
PLC -> HT	*02*	*42*	*...*	*03*	*...*

This command displays the transmitted text from the current cusor position. The remaining part of the display remains unchanged. The characters for which there is no space in the corresponding line will be displayed in the next line.

Text line 0 to 7	STX	'0' '1' '2' '3' '4' '5' '6' '7'	<text>	ETX	BCC
PLC -> HT	*02*		*...*	*03*	*...*

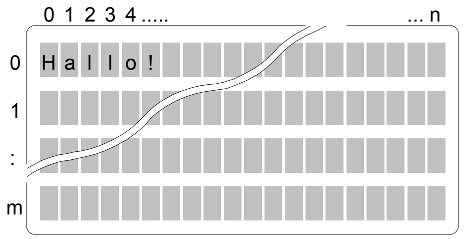
These commands are used for the output of a text in the specified line. If the text does not need the space of the entire line the remaining characters must be filled with blanks (20H).

Example

The text “Hallo !“ should be displayed in line 0.

PLC -> HT:

STX	0	H a l l o !	ETX	BCC
02	*71*	*48 61 6C 6C 6F 21 20 20 20 20 20 20 20 20 20 20 20 20 20 20*	*03*	*54*



Text xy (ASCII/binary)	STX	‘A’ ‘T’	<30H+line> <30H+column> <line> <column>	ETX	BCC

PLC -> HT	*02*	*41* *51*	... ...	*03*	...

This command positions a text on the display at a certain line and column. The remaining part of the display remains unchanged. The characters for which there is no space in the corresponding line will be displayed in the next line.

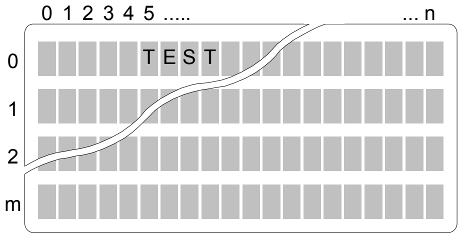
Example

PLC -> HT:

STX	'A'	<30H+line 0> <30H+column 5> T E S T	ETX	BCC
02	*41*	*30 35 54 45 53 54*	*03*	*51*

or

STX	'T'	<line 0> <column 5> T E S T	ETX	BCC
02	*54*	*00 05 54 45 53 54*	*03*	*46*



Variables

The handheld terminal distinguishes between system and user variables.

System Variables

The system variables influence the behavior of the HT (also refer to chapter "KeTop System Variables").

Variable numbers:	0 - 99	for 1/2/4-byte variables

Apart from few exceptions, the system variables can be read and written by the PLC.

User Variables

User variables can be defined as input or output variables within texts.

Variable numbers:	100 - 65535	for 1/2/4-byte variables
	0 – 255	for string variables

User variables are not stored in the HT and, thus, cannot be read out by the PLC.

Input variables

The input variables are the user variables the user can enter on the handheld terminal. After pressing the Enter key, these variables will be transmitted from the HT to the PLC.

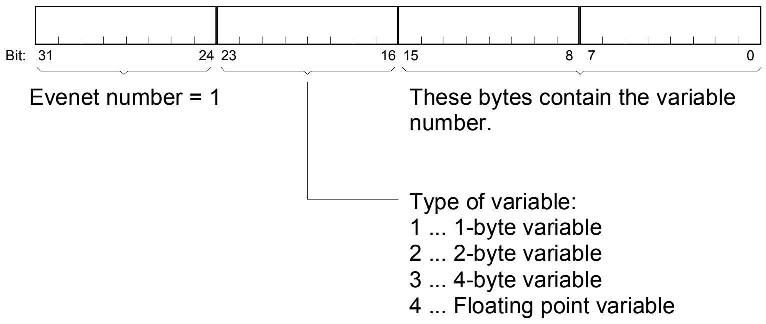
Exception:
If the status message (see chapter "Special Functions") with the content "event variable" has been activated, the entered variable will only be sent with the next status message (polling with FIFO principle) after pressing Enter.

Output variables

The output variables are the user variables the PLC can send to the handheld terminal where they are displayed.
The HT will only be able to display the variables if an output field for this variable number has been programmed for at least one of the texts just activated.

If, for the sent variable, no output field is available on the display the HT can inform the PLC about this fact. To activate the transmission of this information, the 1-byte system variable No. 22 (16H) „EVENT_VAR_NOT_ON_SCREEN“ must be set to TRUE.

For each variable which cannot be displayed, the HT sends the 4-byte system variable No. 2 „EVENT_CODE“ to the PLC („W“ command).



Sending Variables

Send 1/2/4-byte variables	STX	'U' 'V' 'W'	<varNo-H><varNo-L><value>	ETX	BCC
PLC -> HT HT -> PLC	*02*		*...*	*03*	*...*

These commands are used to write integer variables with different lengths:

‘U’	send 1-byte variable (data type UINT8)
‘V’	send 2-byte variable (data type UINT8 and SINT16)
‘W’	send 4-byte variable (data type UINT32 and SINT32)

VarNo-H	variable number, high byte
VarNo-L	variable number, low byte
Wert	1, 2 or 4 bytes (Motorola format)

Application: system and user variables.

Example

By sending the variable KEY_CLICK_PRESSED (var.No. 07H), a beep should be activated when a key is pressed.

PLC -> HT:

STX	'U'	00 07 01	ETX	BCC
02	*55*	*00 07 01*	*03*	*50*

Send floating point variable	STX	'X'	<varNo-H><varNo-L><value>	ETX	BCC
PLC -> HT HT -> PLC	*02*	*58*	*...*	*03*	*...*

This commmand is used to write floating point variables with a length of 4 bytes (data type FLOAT32).
The floating point variables are coded in accordance with the IEEE standard.
Application: user variables.

Send string variables	STX	'Y'	<varNo><length><value>	ETX	BCC
PLC -> HT HT -> PLC	*02*	*59*	*...*	*03*	*...*

This command is used to write string variables with a variable length (data type STRING).
Application: user variables.

Requesting Variables

Request 1/2/4-byte variables	STX	'u' 'v' 'w'	<varNo-H><varNo-L>	ETX	BCC
PLC -> HT HT -> PLC	*02*	*75* *76* *77*	*...*	*03*	*...*

These commands are used to request integer variables with different lengths:

‘u’	request 1-byte variable (data type UINT8 and SINT8)
‘v’	request 2-byte variable (data type UINT16 and SINT16)
‘w’	request 4-byte variable (data type UINT32 and SINT32)

varNo-H	variable number, high byte
varNo-L	variable number, low byte
Value	1-, 2- or 4-byte (Motorola format)

Application: system variables (PLC -> HT) and user variables (HT -> PLC).

Response:	The message „send 1/2/4-byte variables“ (command 'U'/'V'/'W') is sent to the receiver as response.
HT -> PLC or PLC -> HT	

Request floating point variables	STX	'x'	<varNo-H><varNo-L>	ETX	BCC
PLC -> HT HT -> PLC	*02*	*78*	...	*03*	...

This command is used to request floating point variables.

varNo-H	variable number, high byte
varNo-L	variable number, low byte

Application: user variables

Response PLC: PLC -> HT	As response, the PLC must send the message „send floating point variable“ (command 'X').

Request string variables	STX	'y'	<varNo><length>	ETX	BCC
HT -> PLC	*02*	*79*	...	*03*	...

This command is used to request string variables.
Application: user variables

Response PLC: PLC -> HT	As response, the PLC must send the message „send string variable“ (command 'Y').

Buzzer Activation

Buzzer activation	STX	BEL	<30H+duration>	ETX	BCC
PLC -> HT	*02*	*07*	...	*03*	...

This command activates the signal buzzer of the HT. The length of the audible signal can be set by means of the duration:

duration of beep = 2^<duration> x 10ms

The duration value ranges between 0 and 7. Therefore the duration of the beep can be set to a minimum value of 10ms (2^0 x 10ms) up to a maximum value of 1.28s (2^7 x 10ms).

System Reset

System reset (warm start)	STX	'd'	ETX	BCC
PLC -> HT	*02*	*64*	*03*	...

After receiving this command, the handheld terminal performs a warm start (without self-test). The display will be cleared, all LEDs switched off, all settings activated on the HT reset and the handwheel value set to 0. Following that, the HT sends the control character EOT to the PLC.

Communication Timeout

Set timeout	STX	't'	<duration>	ETX	BCC
PLC -> HT	*02*	*74*	...	*03*	...

This command enables the user of the handheld terminal to continuously control the communication with the PLC.

If the PLC does not send any message to the handheld terminal for a time longer than the set timeout, the message "Communication Timeout" appears on the display (see chapter "Error Messages"). To suppress the display of this message, 0 must be sent as <duration> 0.

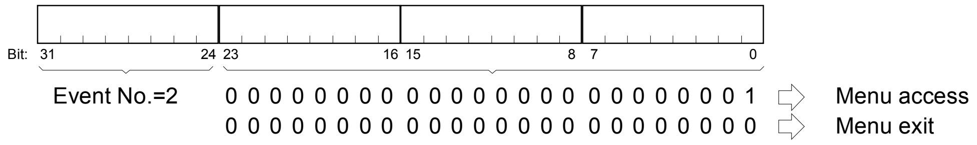
Timeout = <duration> x 100ms

duration = 0	set timeout is deactivated
duration = 20	factory setting (20x100ms=2s)

Menu Access/Exit

If the 1-byte system variable 28 (EVENT_MENU) has been set, the HT will send, upon access to and exit from the menu, an event message to the PLC by means of the 4-byte system variable (EVENT_CODE).

Structure of event variable:



Event No.=2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 Menu access
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Menu exit

Special Functions

Status Configuration

Status configuration	STX	'S'	<sending time> <5 bytes reserve> <content>	ETX	BCC
PLC -> HT	*02*	*53*	*... ... ...*	*03*	*...*

This command is used to define the content of the status messages ('J' command) and to define when the HT should send the status message.

Sending time (when should the HT send the status message)

The sending time is defined by one of the following bits:

Bits 0=1	response to a received message (only possibility at present)
Bits 1-7	reserve
=> sending time = 01H	

Bit	**Content**	**Length in status message**
Bit 0	Key status	4 bytes
Bit 1	Reserved	2 bytes
Bit 2	Reserved	1 byte
Bit 3	Reserved	1 byte
Bit 4	Reserved	1 byte
Bit 5	Reserved	2 bytes
Bit 6	Reserved	2 bytes
Bit 7	Event variable	n bytes

If only the sending time is defined, but no content, no status message will be sent.

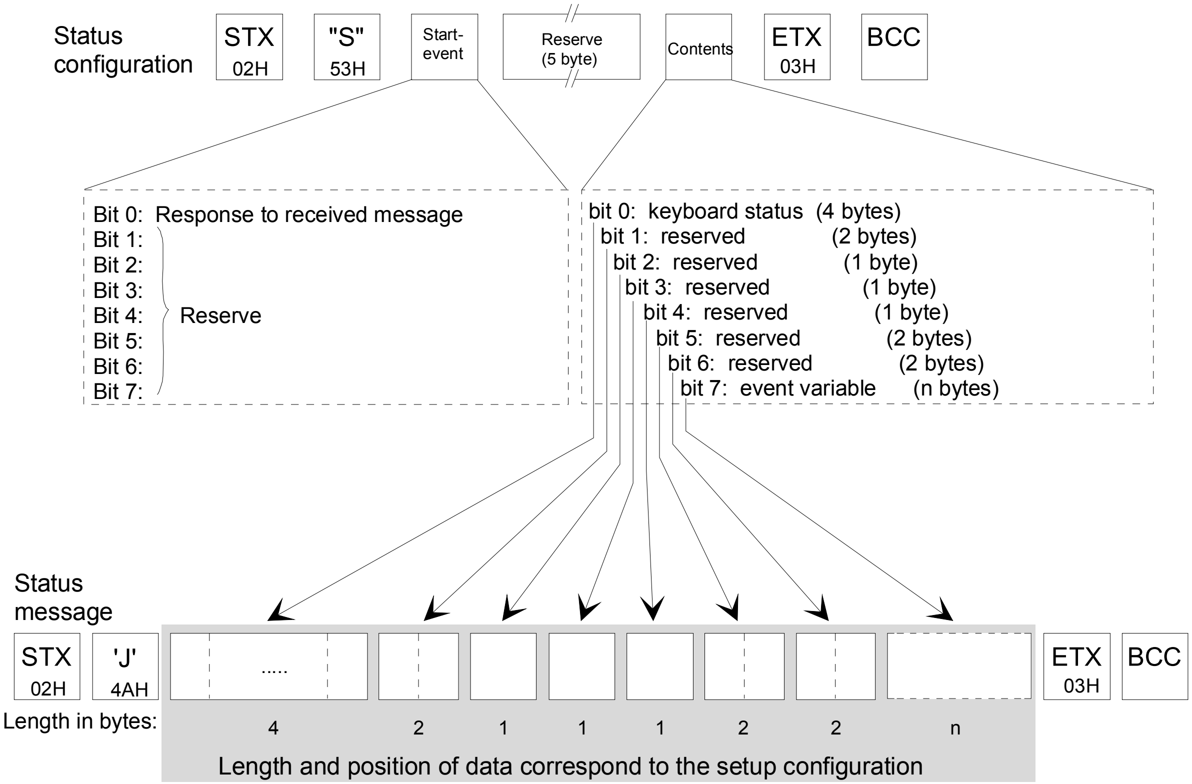
No status message will be sent either if only the event variable has been defined as content, and if no variable is to be transmitted at the sending time.

Status

Status	STX	'S'	<according to configured content>	ETX	BCC
HT -> PLC	*02*	*4A*	*... ... ...*	*03*	*...*

The content and the sending time of the status message (‘J’ command) depend on the settings of the message „Status configuration“ (‘S’ command) received before.

Structure



If the bit 0 "response to received message" has been activated with the message "Status configuration", the behavior of the HT will depend on the type of the message received:

- When receiving PLC messages which are not followed by any response data (e.g. "red LED on"), the HT only returns the status message:

Response HT:	STX	‘J’	<status data>	ETX	BCC
HT -> PLC	*02*	*4A*	....	*03*	...

Exception:
After the reset command 'd', no status message will be sent.

- In case of PLC messages which are followed by response data (e.g. "request key status") the status message will be sent directly after the response data:

ID of response ↓

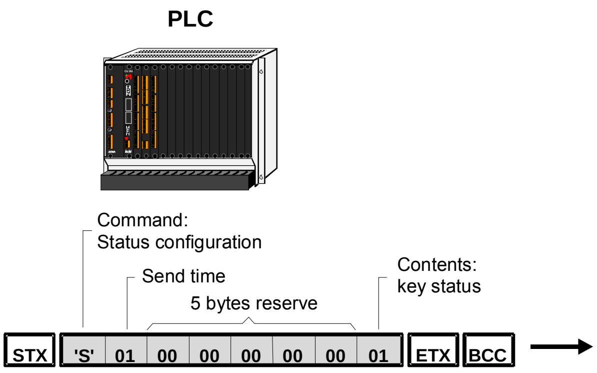
ID of status information ↓

Response HT:	STX	‘..’	<data of response>	‘J’	<status data>	ETX	BCC
HT -> PLC	*02*	*..*	*....*	*4A*		*03*	*...*

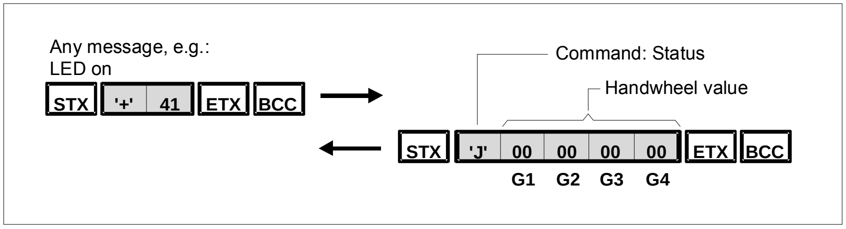
Example: Send status configuration and status

With the status message (command 'J'), the KeTop T40 should send the current key status to the PLC. For that purpose, the „bit 0: key status“ must be set with the message „Status configuration“.

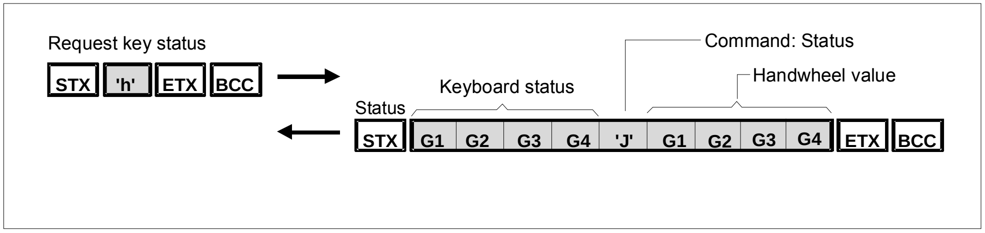
Each time the handheld terminal receives a message from the PLC, the status message too should be sent to the PLC (sending time=01H).



See *notice A)*



See *notice B)*



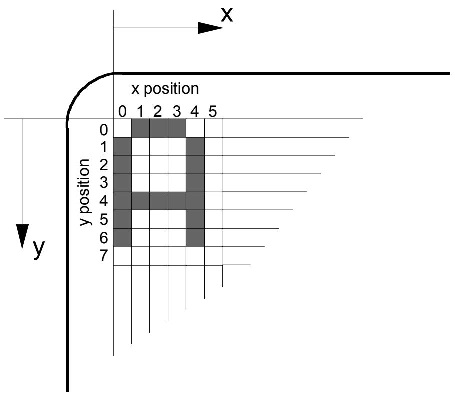
Notice

A) If the HT receives a message that is not directly followed by a response, the HT will only return the status message to the PLC.

B) If the HT receives a message that is directly followed by a response, the HT will return the response including the status information to the PLC.

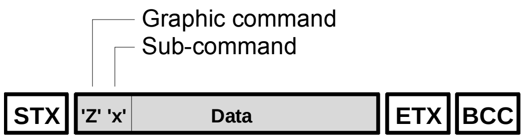
Graphic Functions

The graphic functions are activated from the control. The functions enable displaying a text and drawing graphical elements (line, circle, rectangle and ellipsis) at any position of the display (pixel-oriented).



Letter A in standard size at x/y position 0/0 of HT display

The messages consist of a graphic command and a sub-command. The messages have the following structure:



Structure of message for graphic functions

Graphical Elements

The elements circle, rectangle and ellipse can be represented as filled areas. To delete a graphical element, draw the same element of the same size, at the same position and in inverse color.

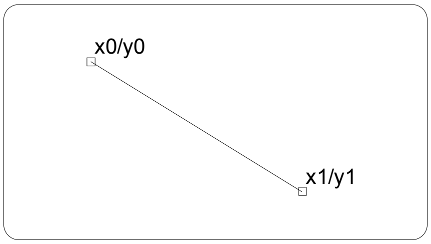
If you position graphical elements over existing texts or bitmaps, these texts and bitmaps will be overwritten.

Pixel	STX	'Z' 'P'	<x-H> <x-L> <y-H> <y-L>	ETX	BCC
PLC -> HT	*02*	*5A 50*	... ...	*03*	...

This command enables representing a single pixel at the specified x/y position of the HT display.

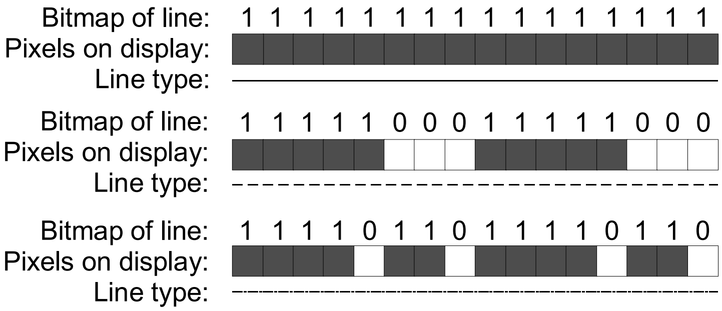
Line	STX	'Z' 'L'	<x0-H> <x0-L> <y0-H> <y0-L> <x1-H> <x1-L> <y1-H> <y1-L>	ETX	BCC
PLC -> HT	*02*	*5A 4C*	... ...	*03*	...

This command enables representing a line on the HT display according to the specified x/y coordinates.



Set line type	STX	'Z' 'l'	(line type)	ETX	BCC
PLC -> HT	*02*	*5A 6C*	*... ... ...*	*03*	*...*

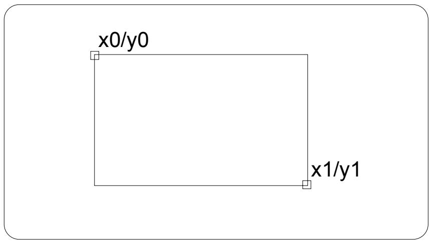
This command is used to set a line type on which all other graphical elements are based. This setting remains valid until a new line type is defined.


Bitmap of line: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Pixels on display:
Line type:
Bitmap of line: 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0
Pixels on display:
Line type:
Bitmap of line: 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0
Pixels on display:
Line type:

Notice
When defining a line type take care to arrange several identical bitmaps one after the other. If you choose an unsuitable bitmap the line might get irregular.

Rectangle	STX	'Z' 'R'	<x0-H> <x0-L> <y0-H> <y0-L> <x1-H> <x1-L> <y1-H> <y1-L> [fill]	ETX	BCC
PLC -> HT	*02*	*5A 52*	*... ... ...*	*03*	*...*

This command enables representing a rectangle on the HT display according to the specified x/y coordinates.

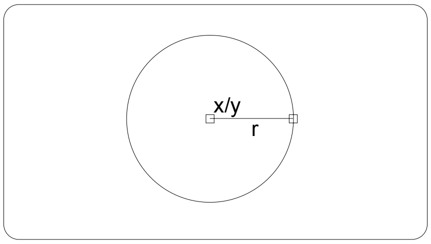


fill

0H	graphic not filled
1H	graphic filled

Circle	STX	'Z' 'C'	<x-H> <x-L> <y-H> <y-L> <r-H> <r-L> [fill]	ETX	BCC
PLC -> HT	*02*	*5A 43*	*... ... ...*	*03*	*...*

This command enables representing a circle with the radius r on the HT display at the specified x/y position.

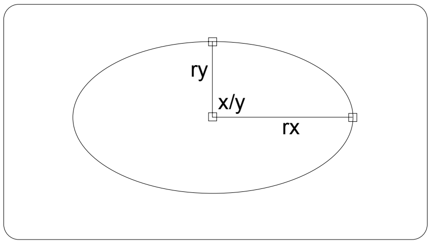


fill

0H graphic not filled
1H graphic filled

Ellipse	STX	'Z' 'E'	<x-H> <x-L> <y-H> <y-L> <rx-H> <rx-L> <rx-H> <rx-L> [fill]	ETX	BCC
PLC -> HT	*02*	*5A 45*	*... ... ...*	*03*	*...*

This command enables representing an ellipse with the radiuses rx and ry on the HT display at the specified x/y position.

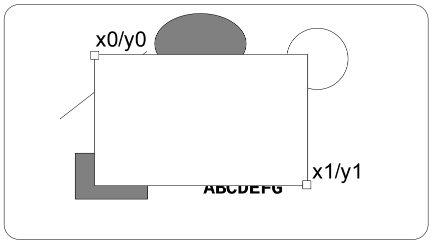


fill

0H graphic not filled
1H graphic filled

Clear window	STX	'Z' 'c'	<x0-H> <x0-L> <y0-H> <y0-L> <x1-H> <x1-L> <y1-H> <y1-L> [fill]	ETX	BCC
PLC -> HT	*02*	*5A 63*	*... ... ...*	*03*	*...*

This command enables clearing a rectangular area on the HT display according to the specified x/y coordinates.



This command corresponds to the drawing of a filled rectangle with the set background color.

Set colors	STX	'Z' 'o'	[fg] [bg]	ETX	BCC
PLC -> HT	*02*	*5A 6F*	*... ...*	*03*	*...*

This command enables setting the foreground and background color (at present only black or white) for all following graphical elements and graphical texts. The color remains active until another color is set.

fg (foreground)

0H	white
FFH	black

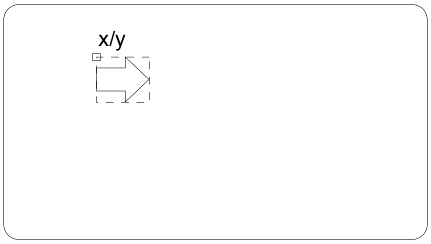
bg (background)

0H	white
FFH	black

Display of bitmaps	STX	‘Z’ ‘B’	<bitmap No.-H>< bitmap No.-L> <x> <y>	ETX	BCC
PLC -> HT	*02*	*5A 42*	*.. .. .. ..*	*03*	*...*

This command enables representing the content of a Windows bitmap file, stored in the bitmap memory of the handheld terminal, on the HT display at the specified x/y position.

The bitmap size is limited by the display size of the KeTop T40 (128x64 pixels).



During configuration, identification numbers are assigned to the bitmaps.

Graphical Text

These texts must be prepared in the PLC and can then be sent to the handheld terminal with the command ‘Z“T‘.

The text can be placed freely on the display. The text characters may be represented in double height or double width. As standard the characters are displayed in the format 5 x 7 pixels (incl. space 6 x 8 pixels). So characters with a double height have a format of 5 x 14 pixels (incl. space 6 x 16 pixels) and characters with a double width 10 x 7 pixels (incl. space 12 x 8 pixels).

Displaying a character means that the area for the character is cleared and, following that, the character is written into the empty field. An existing graphic or bitmap representation will be deleted at this position.

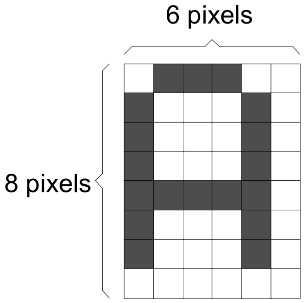
Notice

A) Programmed texts stored in the HT can only be displayed in lines or columns. These texts cannot be displayed at any pixel position or with a larger size.

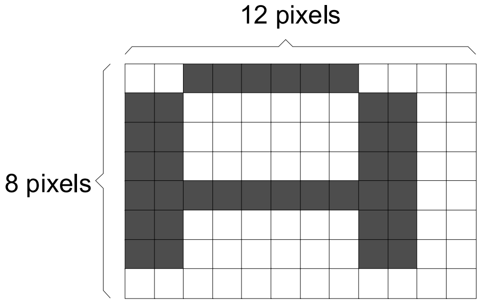
B) For graphical texts that are directly sent from the PLC to the handheld terminal display, no editor function is available. That means the HT will not correctly interpret symbols (___, ###) for the input and output of variables.

Text	STX	'Z' 'o'	<x-H><x-L> <y-H><y-L> [font_size] [font_attr] “Text..:“	ETX	BCC
PLC -> HT	*02*	*5A 54*	... ...	*03*	...

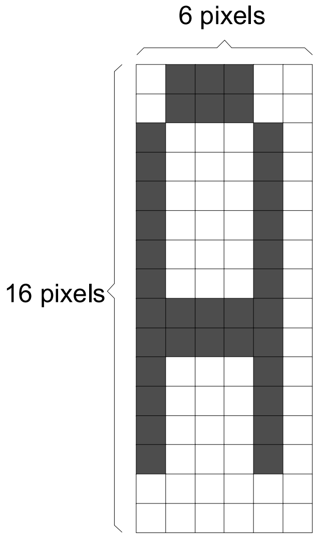
This command enables representing a maximum of 40 text characters at the specified x/y position of the HT display (provided the characters are not displayed with a double width and begin at the pixel column 0).
The text is displayed up to the end of the line (no line folding and no continuation in the next line).



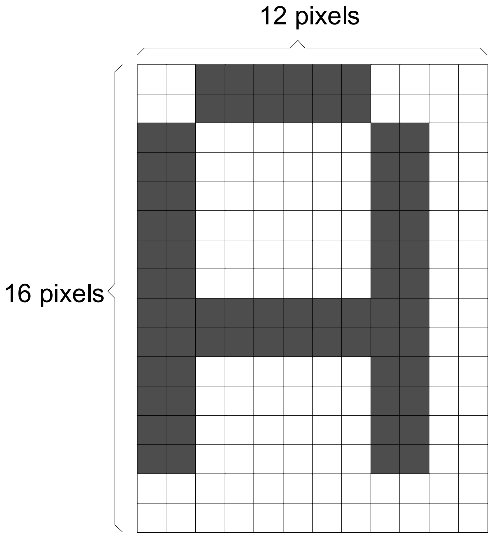
font_size = 0
(Character: 5x7 pixels)



font_size = 2
(Character: 10x7 pixels)



font_size = 1
(Character: 5x14 pixels)



font_size = 3
(Character: 10x14 pixels)

font_size This parameter defines the size of the text characters on the HT display. The following settings are possible:

0 ... normal
1 ... double height
2 ... double width
3 ... double height, double width

font_attr This parameter contains the font attribute and specifies if the transmitted text should be displayed normally, in inverse characters, normally flashing or inverse flashing.

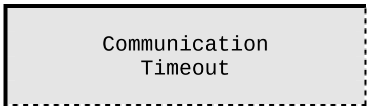
0 ... normal
1 ... inverse
2 ... normal flashing
3 ... inverse flashing

8 Error Messages

In case of an error caused by the user, the HT triggers a short warning tone. Then an error message will be displayed. In this condition, inputs nor outputs are no longer possible. The device remains inactive until the error is eliminated. Then the error message automatically disappears from the display and the data displayed before the error will appear again.

Possible errors:

Communication timeout



```
Communication
   Timeout
```

If this error appears immediately after power-on:

- check the set protocol parameters of the PLC and the connecting cable.

If this error appears during operation:

- check the mechanical connection to the PLC and if the PLC still responds to the HT.