

Installation Manual

P/N 0-230049-02



MicroBar 9730 Decoder

 **ntermec**

A **UNOVA** Company

Regulatory Statements



Intermec hereby declares that the MicroBar 9730 has been tested and found compliant with the below listed standards as required by the EMC Directive 89/336/EEC as amended by 92/31/EEC and by the Low Voltage Directive 73/23/EEC as amended by 93/68/EEC:

EN55022 (1992) EN50082-1 (1998) EN60950 (1993)

USA: This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. It generates, uses and can radiate radio frequency energy. If not installed and used in accordance with the instructions, it may cause interference to radio communications. If this equipment causes interference, the user will be required to correct the interference at the user's own expense.

This equipment complies with the UL 1950 standard.

Canada: This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations. Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

This equipment complies with the UL 1950 standard.
Cet équipement est conforme à la norme UL 1950.

Australia-New Zealand: This equipment has been tested and found to conform to the Australian EMC framework concerning Class B digital devices, prescribed by the Australian and New-Zealand standard AS/NZS 3548.



N309

Mexico: Este equipo cumple con la certificación NOM.
This equipment complies with the NOM certification.

The information contained in this document is for informational purposes only and is subject to change without notice. No part of this document may be copied or reproduced in any manner without the prior written permission of Intermec Technologies Corporation.

© Intermec Technologies Corporation All Rights Reserved

The word Intermec, the Intermec logo, MicroBar and EasySet are either trademarks or registered trademarks of Intermec.

Throughout this document, trademarked names may be used. Rather than put a trademark (™ or ®) symbol in every occurrence of a trademarked name, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement.

North America / Asia Pacific / Latin America

6001 36th Avenue West
PO Box 4280
Everett WA 98203-9280
Tel : +1 425 348 2600
Fax : +1 425 348 2833
U.S. service and technical support, tel: 1.800.755.5505
Canadian service and technical support, tel: 1.800.688.7043

Europe / Middle East / Africa

Sovereign House
Vastern Road
Reading RG1 8BT
England
Tel : +44 118 987 9420
Fax : +44 118 987 9416

Internet :

<http://www.intermec.com>

E-mail :

info@intermec.com

Support :

<http://datacapture.intermec.com>

Contents

1. Using EasySet	9
online setup with RS-232 cable 0-234032-00	10
2. Reset all parameters	11
3. Terminal / cash register used in your application	13
connections.....	14
keyboard wedge – 2-cable connection.....	14
keyboard wedge – Y-cable connection (DEC VT 220/320/420)....	15
keyboard wedge – Y-cable connection (HP 700/xx, Wyse 60/65/xx)	16
RS-232 – direct connection	17
RS-232 – dual RS connection	18
predefined terminal selections	19
keyboard wedge	19
RS-232	24
custom interface number	25
4. Output transmission settings.....	27
keyboard wedge.....	28
special keys interpretation (Code 39)	28
control character substitution	31
keyboard remapping	32
Alt mode transmission	33
inter-character delay (output to host)	33
inter-message delay (output to host)	34
end-of-transmission keyboard character status.....	35
RS-232	37
baud rate.....	37
data bits	38
parity.....	38

stop bits	39
hardware/software protocols timeout	39
ENQ	40
ACK	41
NAK	42
Typical ENQ / ACK / NAK scenarios	43
XON / XOFF software protocol	44
RTS / CTS hardware protocol	44
LRC (longitudinal redundancy check)	45
inter-character delay (output to host)	47
inter-message delay (output to host)	48
message length information	50
preamble	52
keyboard wedge	52
RS-232	53
input port identifier	55
position	55
port 1 (wedge Y cable / RS-232) / port 3 (wedge 2 cables)	56
port 4	56
port 5	57
port 6	57
symbology identifier	58
AIM	58
custom	59
data editing (EasySet 3.0 and later)	64
data reformatting (EasySet 2.01, 2.02, 2.03)	66
Overview	66
"Define input" page	68
"Define output" page	70
postamble	75
keyboard wedge	75

RS-232	77
5. Symbolologies	79
Ames Code	80
Codabar.....	81
start/stop	81
CLSI library system	82
check digit (AIM recommendation)	82
concatenation.....	83
Code 11.....	85
check digits	85
Code 39.....	86
format	86
start/stop	90
check digit.....	91
multiread	92
Code 93.....	94
multiread	94
Code 128 / EAN 128	96
EAN 128 identifier.....	96
CIP 128 French pharmaceutical codes	97
FNC1 separator character (EAN 128 norms)	97
ISBT 128 concatenation.....	98
Interleaved 2 of 5	99
check digit.....	99
barcode length	100
Matrix 2 of 5.....	102
MSI Code.....	103
check digit.....	103
Plessey Code.....	104
check digit.....	104
Standard 2 of 5	105

format	105
check digit mod 10	105
barcode length	106
Telepen	108
format	108
UPC / EAN	109
add-on digits	110
check digit.....	111
UPC number system	112
re-encoding UPC-A, UPC-E, EAN-8.....	113
6. Input devices	115
laser scanners / CCD scanners in laser emulation.....	116
activation	116
laser trigger management.....	118
consecutive same read data validation.....	121
magstripe readers	122
track selection.....	122
transmission sequence	122
track separator character	123
start/end sentinels	123
LRC check digit.....	123
scales / weighing machines	124
scale solicitation	124
RS-232 input devices on port 4.....	126
baud rate	126
data bits.....	127
parity	127
stop bits	128
ENQ / ACK / NAK.....	128
Typical ENQ / ACK / NAK scenarios	130
XON / XOFF software protocol.....	131

RTS/CTS hardware protocol	131
transmission to the host system	132
special keys interpretation (output to keyboard wedge)	134
UDI / CICS	139
RS-232 input devices on port 5.....	140
baud rate.....	140
data bits	141
parity.....	141
stop bits	142
ENQ / ACK / NAK.....	142
XON / XOFF software protocol.....	144
RTS/CTS hardware protocol	144
transmission to the host system	145
special keys interpretation (output to keyboard wedge)	147
UDI / CICS	152
7. Operating settings	153
beeps / green indicator LED	154
volume	154
note (tone frequency).....	154
good read beeps.....	155
good read LED	156
external beep command	157
8. Configuration modes and utilities	159
temporary configuration mode.....	162
configuration password mode	163
A. ASCII character codes	165
B. Additional preamble / postamble characters—Keyboard wedge.....	181
C. Number codes	187
D. Test codes.....	189

Contents

- E. Technical data.....191
 - Input/output ports191
 - Overview.....191
 - Input device priority management192
 - Output to host system.....192
 - I/O port functions.....193
 - I/O port pinouts194
- Index197

1. Using EasySet

- offline setup
 1. select your Intermec product in EasySet ("Select reader" icon over the product image)
 2. double-click the setup commands in EasySet to send them to the setup sheet
 3. print out the setup sheet and read the configuration codes with a barcode reader
 - if you use a ScanPlus or MaxiScan 2200 barcode reader, print out and read the "ScanPlus / MaxiScan transparent configuration mode" code (see section 8, "Configuration modes and utilities") before you set up the MicroBar (if not, you will set up the ScanPlus / MaxiScan and not the MicroBar !)
- online setup :
 - use RS-232 cable 0-234032-00 for online setup (follow the procedure described in this section)

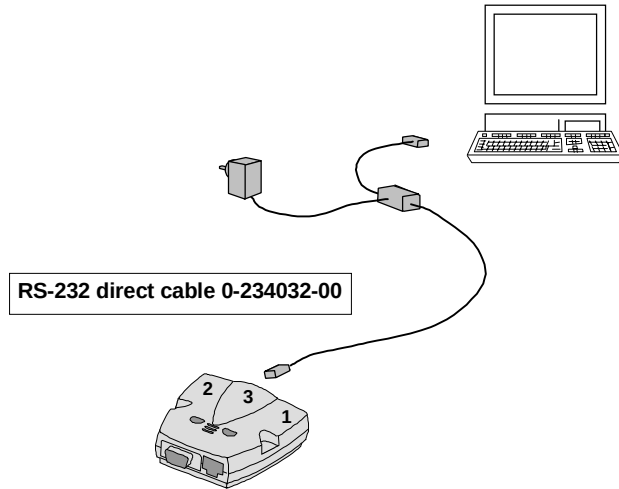
start EasySet online setup



\41\4A\0F\29\60

- only necessary for setup with RS-232 cable 0-234032-00
- print out and read this code to activate online setup with EasySet (double-click the command to send it to the setup sheet)
- follow the procedure for online setup with RS-232 cable 0-234032-00

online setup with RS-232 cable 0-234032-00



- online setup with RS-232 cable 0-234032-00 :
 1. connect the RS-232 cable to port 3 of the MicroBar if it is not already connected
 2. connect the RS-232 cable to the serial port of your PC
 3. connect the power supply to the RS-232 cable and provide electrical power
 4. select your Intermec product in EasySet ("Select reader" icon over the product image)
 5. read the "start EasySet online setup" code (send it to the setup sheet and print it out, or read it from the Getting Started Guide or Installation Manual)
 6. open the "Configure communication PC/Reader" dialog box in the "Reader" menu
 7. select the COM serial port your product is connected to (19200 bauds, 8 data bits, no parity)
 8. select the "Send to reader" checkbox under the commands window
 9. double-click on the setup commands to send them directly to your product (they are also sent to the setup sheet)
 10. connect your normal product cable to your product (if applicable) when you have finished online setup

2. Reset all parameters

- global reset of all parameter settings - useful for a first-time setup or for a fresh start with a new application
- default settings are indicated by (*)



- resets all configuration parameters to their default values and cancels the terminal / cash register selection
- after a global reset, you must select the terminal / cash register used in your application (>> next section) and customize your setup parameters if required

2. Reset all parameters

3. Terminal / cash register used in your application

- terminal selection is necessary to enable transmission to your terminal / cash register !!!
- look for your system configuration in the terminal selector (full list of all supported terminals) or use the predefined terminal selections if applicable
- the number of green LED flashes at power-up indicates the cable / interface configuration for your product
- green LED always on = no terminal / cash register selected

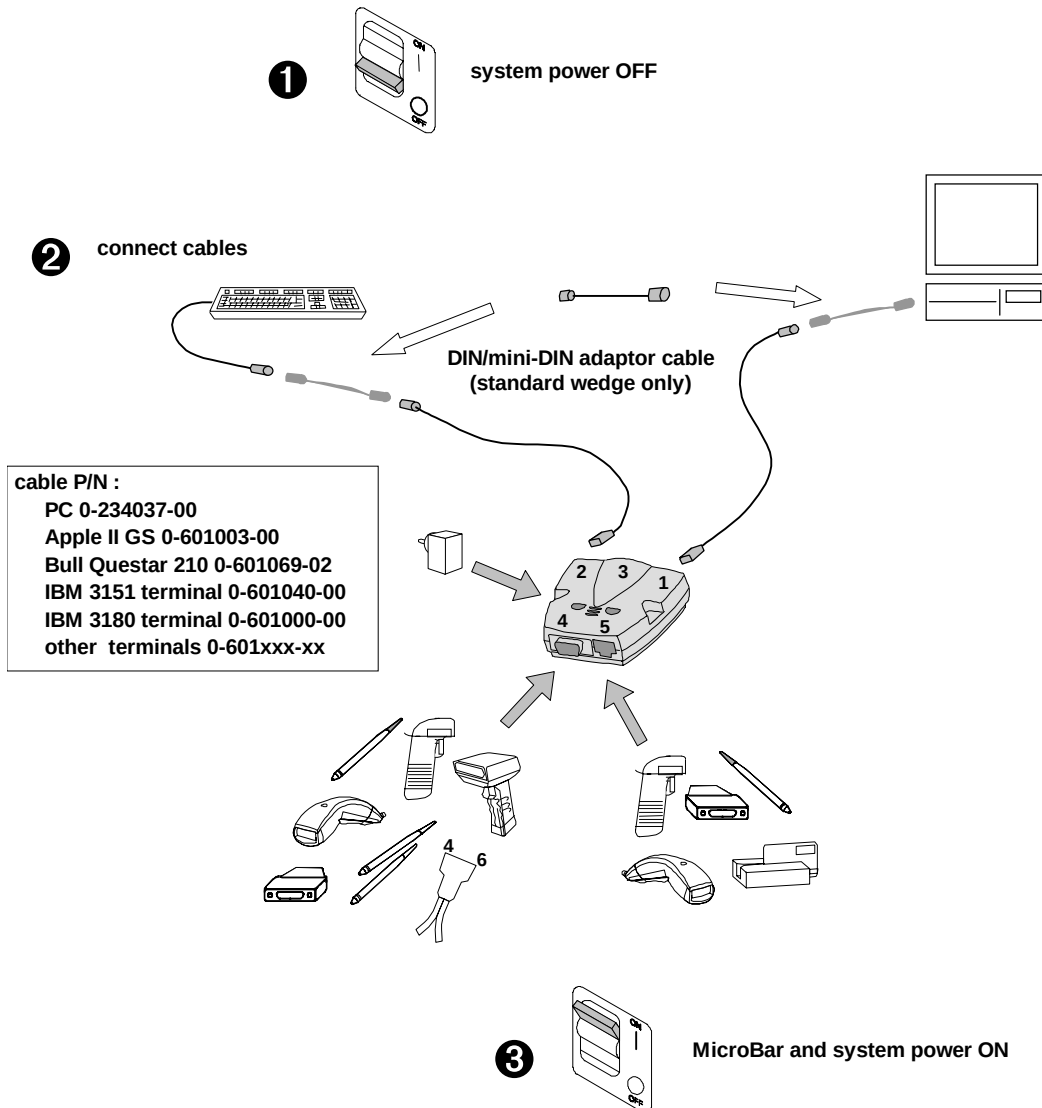
terminal selector (all terminals):

this option only available
with EasySet

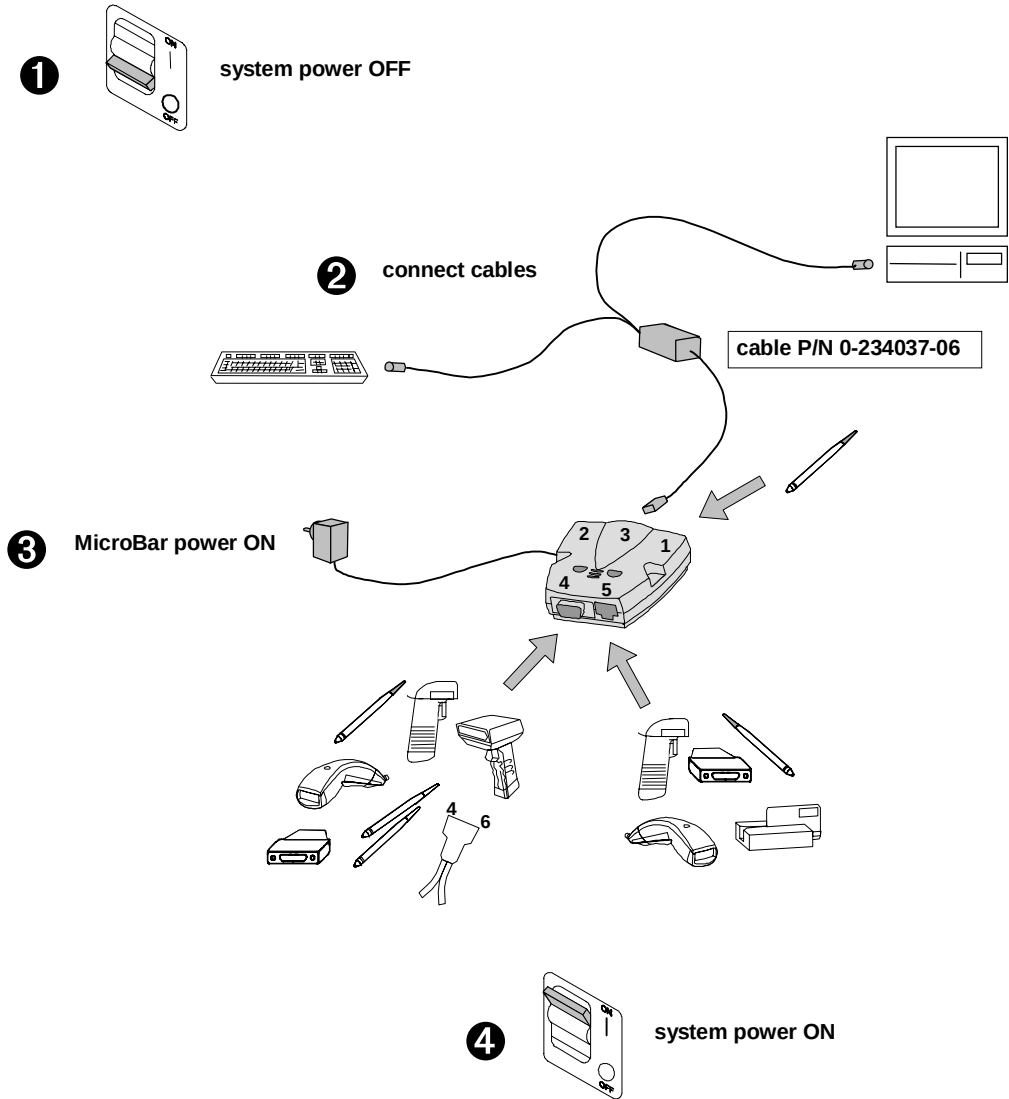
- database of all supported terminals (look for your system configuration in the list)

connections

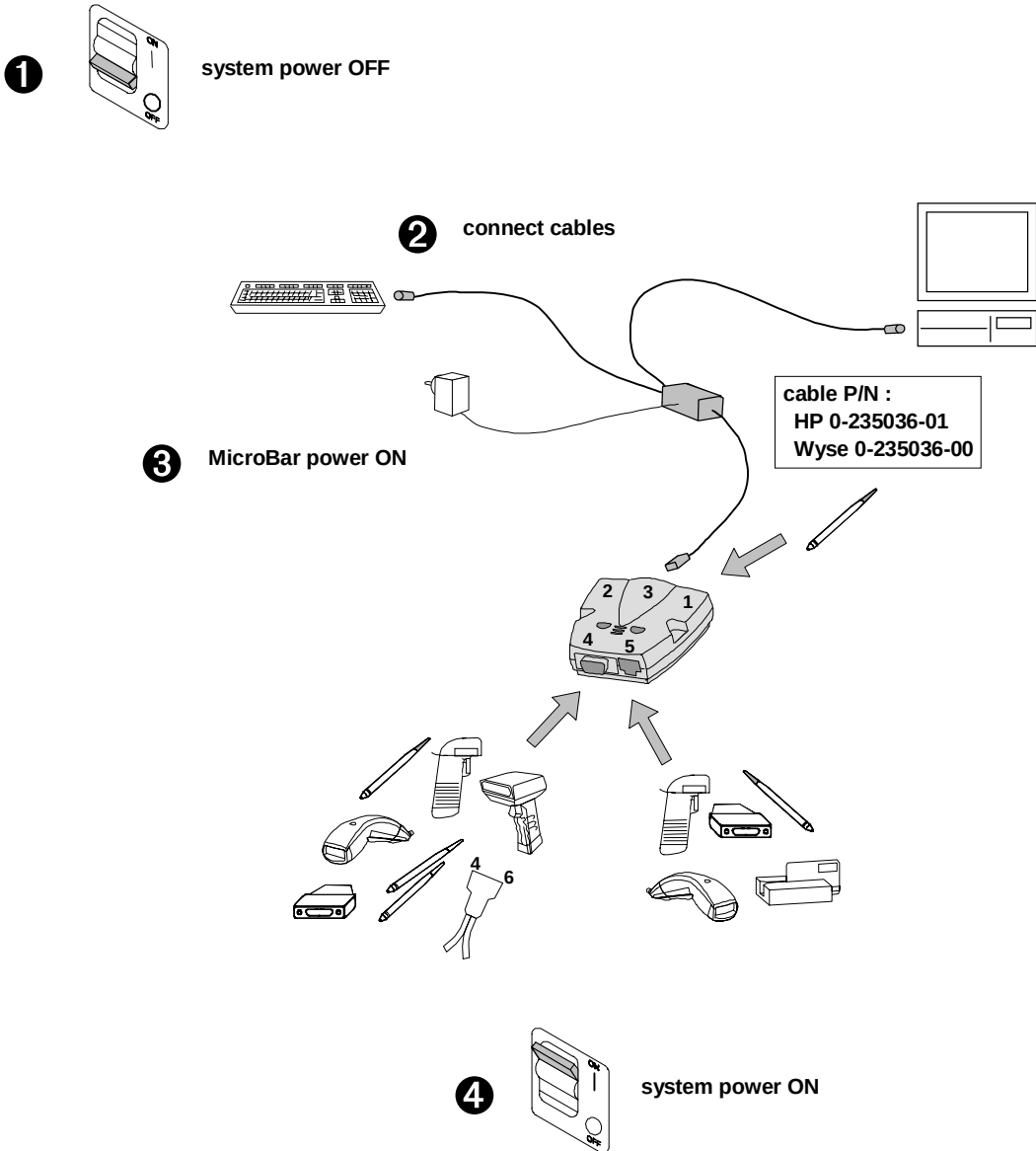
keyboard wedge – 2-cable connection



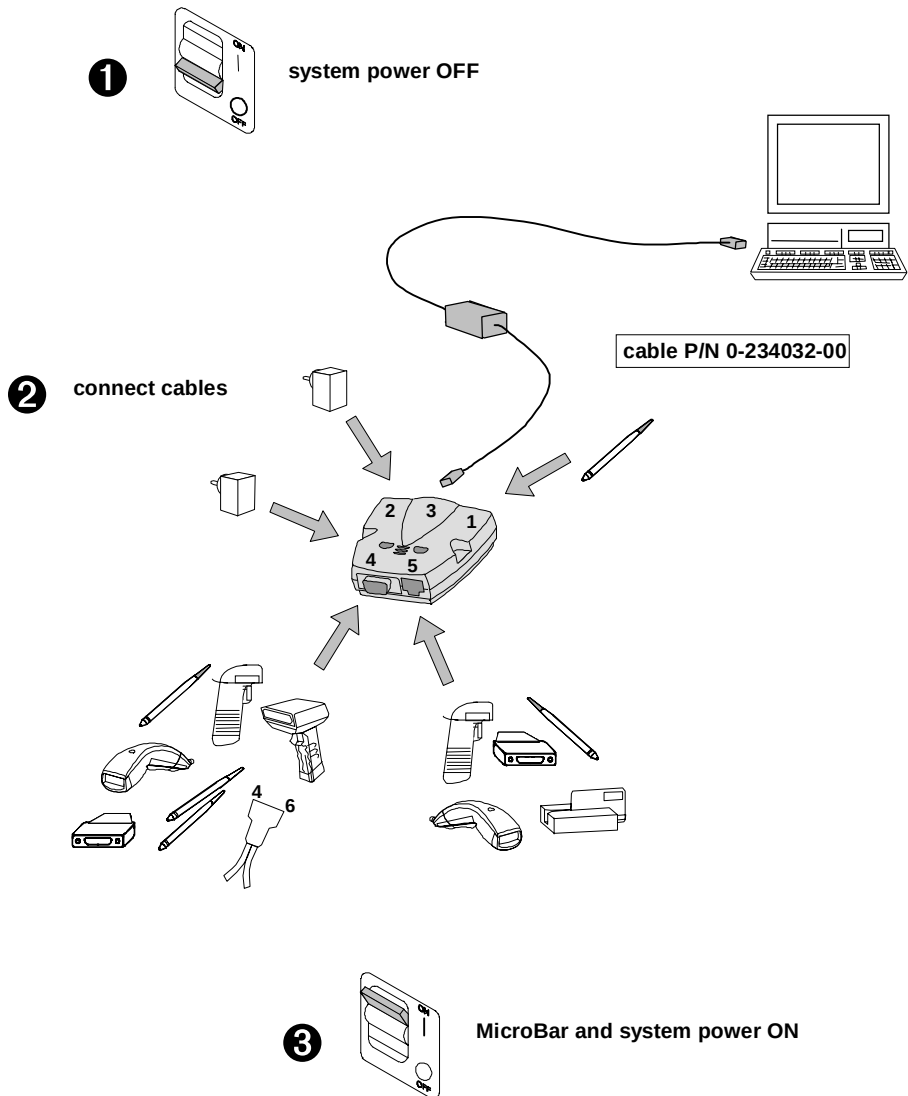
keyboard wedge – Y-cable connection (DEC VT 220/320/420)



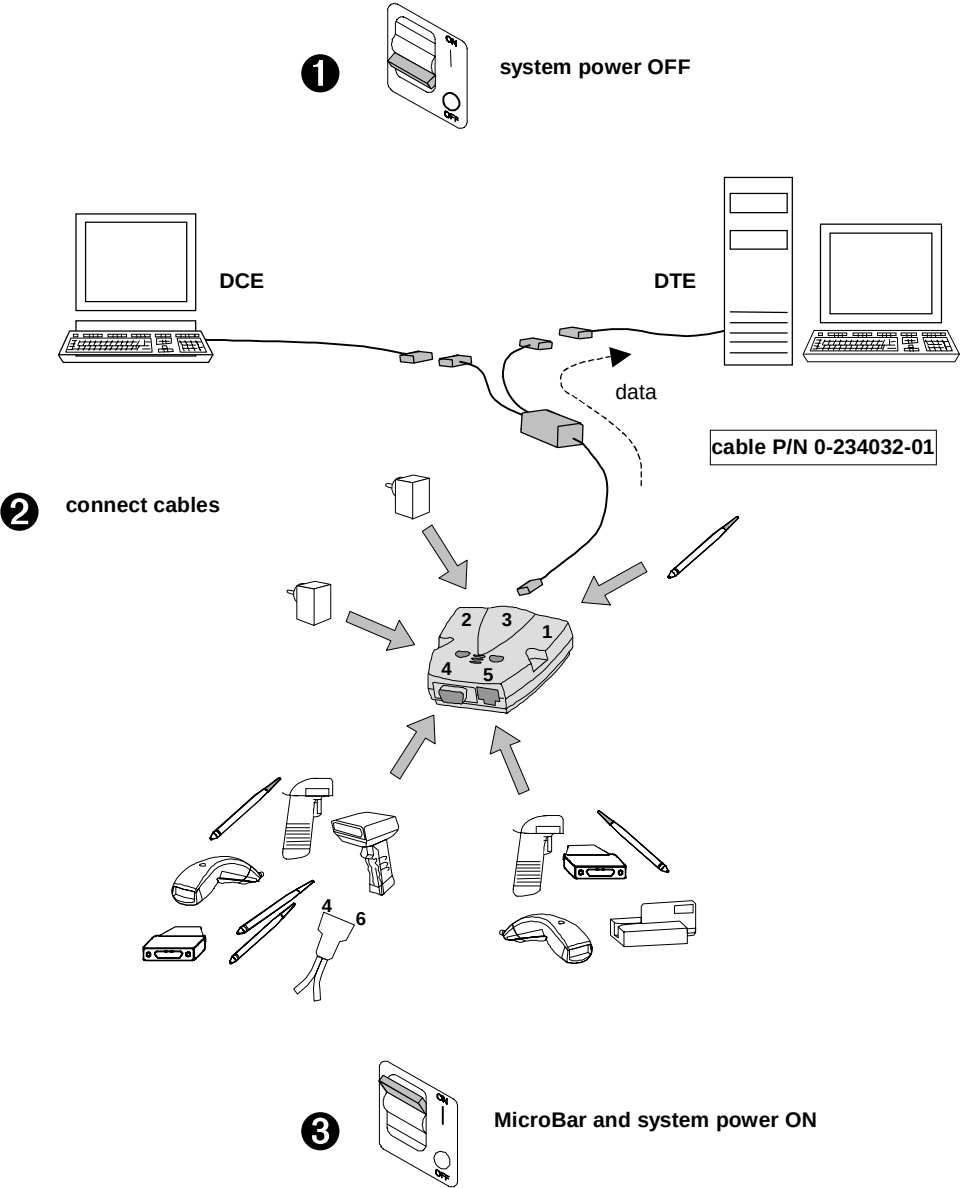
keyboard wedge – Y-cable connection (HP 700/xx, Wyse 60/65/xx)



RS-232 – direct connection



RS-232 – dual RS connection



predefined terminal selections

- look in the terminal selector if you cannot find your system configuration in this section

keyboard wedge

- "keyboard wedge" = connection between a keyboard and the host system (data is transmitted in keyboard emulation mode)
- switch off the host system before you connect the cables and optional power supply !
- connections :
 - wedge applications including PC AT type : connect keyboard to port 2, host computer to port 1
 - wedge applications including Wyse, DEC 220 / 320 / 420 : connect Y cable to port 3
- 7 green LED flashes at power-up
- IMPORTANT: always unplug the optional power supply (if applicable) if you switch off the PC !!!

standard PC

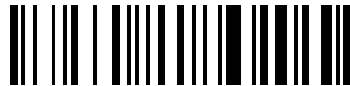
- look in the terminal selector if you cannot find your system configuration in this section

QWERTY English



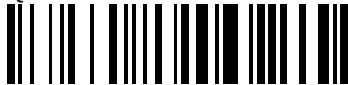
\41\4A\03\08\60

AZERTY French



\41\4A\03\09\60

QWERTZ German



\41\4A\03\0C\60

3. Terminal / cash register used in your application predefined terminal selections

QWERTY Italian



QWERTY Danish



QWERTZ Swiss / French



QWERTY Swedish / Finnish



QWERTY Norwegian



QWERTY Spanish



IBM 31xx, 32xx, 34xx

- look in the terminal selector if you cannot find your system configuration in this section

QWERTY English



AZERTY French



AZERTY international



\41\4A\03\28\60

AZERTY caps QWERTY



\41\4A\03\29\60

QWERTZ German



\41\4A\03\2A\60

QWERTY data entry



\41\4A\24\06\60

AZERTY data entry



\41\4A\24\09\60

QWERTZ numeric keypad



\41\4A\24\0A\60

DEC VT 220, 320, 420

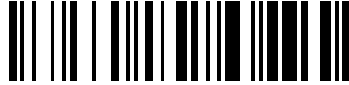
- look in the terminal selector if you cannot find your system configuration in this section

QWERTY PC type



\41\4A\06\1A\60

AZERTY PC type



\41\4A\06\1B\60

QWERTZ PC type



QWERTY PC type Swedish / Finnish



DEC VT / PC 510

- look in the terminal selector if you cannot find your system configuration in this section

AZERTY PC type



QWERTY PC type Swedish / Finnish



AZERTY VT type French



QWERTY VT type Swedish / Finnish



QWERTZ PC type Swiss / French



Apple / Macintosh

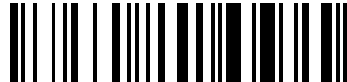
- look in the terminal selector if you cannot find your system configuration in this section

QWERTY English



\41\4A\03\1C\60

AZERTY French



\41\4A\03\1D\60

QWERTZ German



\41\4A\03\20\60

Hewlett Packard 700/92

- look in the terminal selector if you cannot find your system configuration in this section

QWERTY English



\41\4A\04\04\60

AZERTY French



\41\4A\04\05\60

QWERTZ German



\41\4A\04\08\60

Wyse 60, 65, 99GT, 120

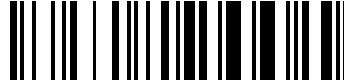
- look in the terminal selector if you cannot find your system configuration in this section

QWERTY 102 keys PC/AT fast



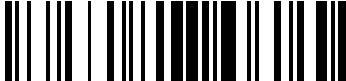
\41\4A\04\2C\60

AZERTY 102 keys PC/AT fast



\41\4A\04\2D\60

QWERTZ 102 keys PC/AT fast



\41\4A\04\30\60

RS-232

- switch off the host system before you connect the cables and optional power supply !
- connect host system cable to port 3
- 2 green LED flashes at power-up

standard RS-232 C (9600, 7, E, 2)



\41\4A\01\24\60

- predefined terminal selection also available with the terminal selector
- PC serial port compatible
- customize individual RS-232 settings to suit your system after reading this parameter

custom interface number

compose: [range: 100-30000]



\41\4A\60

enter a number string and scan End Selection
(Appendix C)

for special host system configurations

null interface



\41\4A\00\60

- cancels the current interface settings (useful for test purposes if you want to read bar codes but do not want to transmit the data)
- green LED always on = no terminal / cash register selected
- reselect your terminal / cash register and customize your data transmission settings (if required) to return to normal operation

4. Output transmission settings

- interface-specific parameters for communication with the host system
- modify output transmission settings to optimize performance

keyboard wedge

- default values (*) are for IBM PC AT and compatible - QWERTY English

special keys interpretation (Code 39)

- certain dual-character combinations in Code 39 bar codes can be interpreted and transmitted as special keyboard keys (Code 39 is not full ASCII and does not support direct encoding of special keyboard keys such as <Enter> and <Tab>)
 - only for keyboard wedge applications with Code 39 bar codes !!! (both the MicroBar and your barcode reader must be enabled to read Code 39 codes with the correct barcode length settings)

emulated key	characters	emulated key	characters
DEL	.A	PF1	0A
ENTER	.B	PF2	0B
RETURN	.C	PF3	0C
SEND	.D	PF4	0D
FIELD +	.E	PF5	0E
FIELD EXIT	.F	PF6	0F
HOME	.G	PF7	0G
END	.H	PF8	0H
TAB	.I	PF9	0I
ALT	.J	PF10	0J
BACK TAB	.K	PF11	0K
BACKSPACE	.L	PF12	0L
right arrow	.M	PF13	0M
left arrow	.N	PF14	0N
up arrow	.O	PF15	0O
down arrow	.P	PF16	0P
CLEAR	.Q	PF17	0Q
FIELD -	.R	PF18	0R
DUP	.S	PF19	0S
ESC	.T	PF20	0T
LINE FEED	.U	PF21	0U

emulated key	characters	emulated key	characters
RESET	.V	PF22	0V
CTRL	.W	PF23	0W
SPECIAL	.X	PF24	0X



special dual-character combinations not interpreted (original code string transmitted without interpretation)

Example (. l dual-character combination = <Tab>)

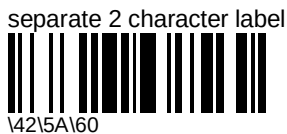
original Code 39 code:	interpreted and transmitted as:
msg . lxx	msg . lxx
. l	. l
x - . l	x - . l
msg - . lxx	msg - . lxx



special dual-character combinations always interpreted as special keys

Example (. l dual-character combination = <Tab>)

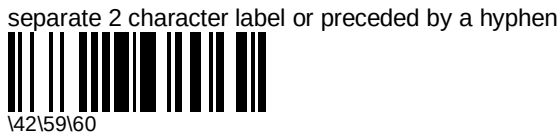
original Code 39 code:	interpreted and transmitted as:
msg . lxx	msg<Tab>xx
. l	<Tab>
x - . l	x <Tab>
msg - . lxx	msg<Tab>xx



special dual-character combinations interpreted if
only 2 characters in the original code

Example (. I dual-character combination = <Tab>)

original Code 39 code:	interpreted and transmitted as:
msg . lxx	msg . lxx
. I	<Tab>
x - . I	x - . I
msg - . lxx	msg - . lxx



special dual-character combinations interpreted if
only 2 characters in the original code or if there is
a hyphen in front of the 2 characters

Example (. I dual-character combination = <Tab>)

original Code 39 code:	interpreted and transmitted as:
msg . lxx	msg . lxx
. I	<Tab>
x - . I	x <Tab>
msg - . lxx	msg<Tab>xx

control character substitution

- systematically replaces a specified character by a control function sequence

not active (*)



transmits characters as they appear in the original message (<ctrl> + <character> function not activated)

active on Form Feed character



- all Form Feed characters replaced by control function sequence

compose activation character and activate: [range: 1]



enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)

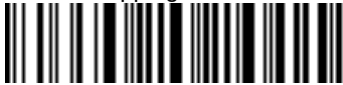
specifies the character used to activate a control function sequence (<specified_character> + <next_character> will be replaced by <ctrl_down> + <next_character> + <ctrl_up>)

Example : if you compose <FF> (Form Feed) as the activation character, the input string 'A <FF> B C' will be transmitted as 'A <ctrl B> C'

keyboard remapping

- define the characters you want to 'remap' before they are transmitted ('input character' transmitted as 'output character')
- keyboard remapping is applied after data editing !!! (if applicable)
- maximum number of character pairs in MicroBar = 178

reset remapping in MicroBar



\4C\46\3E\00\3E\00\60

removes all remap settings in the MicroBar
(recommended before you define new
remapping)

define:

this option only available
with EasySet

- define the character pairs you want to remap
- each new character pair definition is added to the list already in the MicroBar
- if you remap the same input character more than once, the MicroBar uses the latest remap definition (reset the remapping before you define a new set of character pairs)
- maximum number of character pairs in the MicroBar = 178
- maximum length for a single setup command = 255 datastring characters or approximately 100 character pairs (send successive batches if you need more character pairs !)
- keyboard remapping is applied after data editing !!! (if applicable)

Alt mode transmission

- emulates PC AT keyboard [<Alt> + decimal_sequence] function (for bar codes containing ASCII characters not on your keyboard)



- use only if necessary as transmission is slower (all characters are transmitted as <Alt> sequences)

Example

"A { B" is transmitted as:

<Alt> + <6> + <5> <Alt> + <1> + <2> + <3> <Alt> + <6> + <6>

inter-character delay (output to host)

- avoids dropping characters if transmitting decoded data too fast for the host system





enter a number string and scan End Selection
(Appendix C)

inter-message delay (output to host)

- gives the host system enough time to process each message received, especially when the host must act on end-of-message control codes such as <Enter>
- "rolling buffer" operation :
 1. a new inter-message delay cycle is activated each time the MicroBar receives a message
 2. if the MicroBar receives a new message before the end of the inter-message delay, it stores the previous message in a "rolling buffer" memory and the cycle starts again
 3. when no more messages are received before the end of an inter-message delay cycle, all the messages stored in the buffer are transmitted to the host system (First-In-First-Out, each message separated by the inter-message delay)

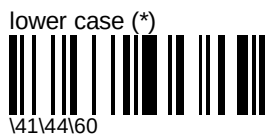




enter a number string and scan End Selection
(Appendix C)

end-of-transmission keyboard character status

- sets keyboard to lower case or upper case at end of transmission
- Code 39 is transmitted in upper case
- Code 128 is transmitted in lower case / upper case (full ASCII)



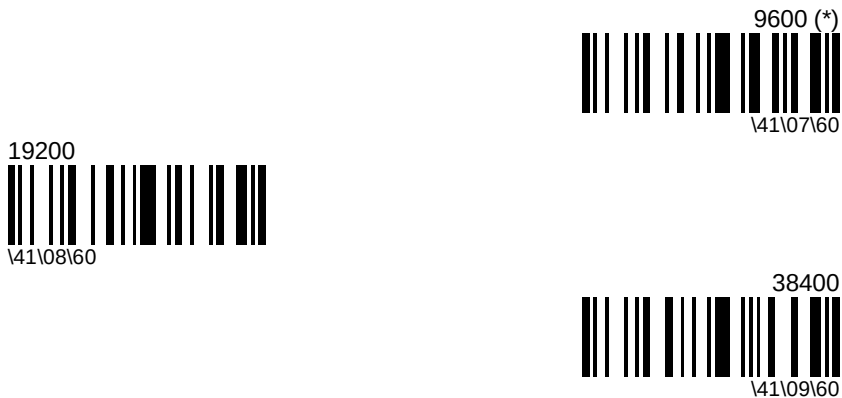


RS-232

- default values (*) are for standard RS-232 C (9600, 7, E, 2)

baud rate

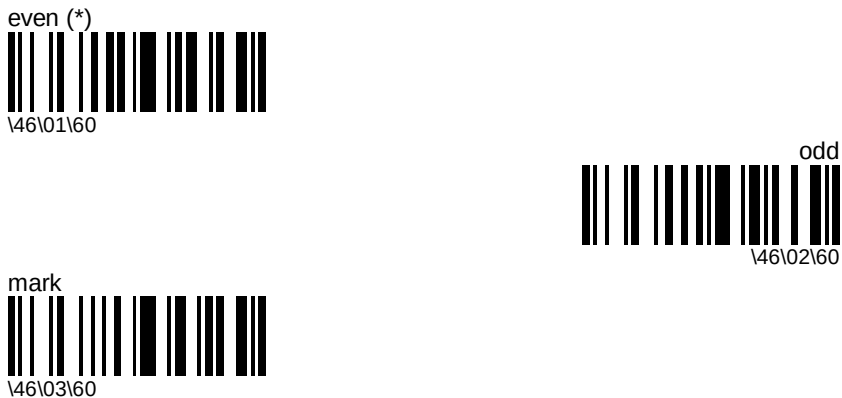




data bits



parity





stop bits



hardware/software protocols timeout

- the same timeout applies to all RS-232 protocols supported (ENQ/ACK/NAK, XON/XOFF, RTS/CTS)



compose (ms): [range: 0-2500 ms]



enter a number string and scan End Selection
(Appendix C)

ENQ

- activating ENQ activates the ENQ protocol
- ENQ not used:
 - barcode data is transmitted without waiting for a request from the host system
- ENQ used:
 - data is transmitted when ENQ character received from host system before end of hardware/software protocols time-out
 - "beep after transmission" if good read ("beep before transmission" also possible - see section 7, "Operating settings")
 - 6 error beeps and no transmission if no ENQ received before end of timeout
- ENQ/ACK/NAK exclusive with XON/XOFF and RTS/CTS

not used (*)



used (ENQ 05h)



compose:



enter an ASCII character and scan End Selection
(Appendix A)

ACK

- activating ACK or NAK activates the ACK/NAK protocol
- before reading a new bar code after transmission, the product waits until the end of the hardware/software protocols time-out
- "beep after transmission" if good read ("beep before transmission" also possible - see section 7, "Operating settings")
- 6 error beeps if no ACK (positive acknowledge) received from the host system before end of timeout
- ENQ/ACK/NAK exclusive with XON/XOFF and RTS/CTS

not used (*)



\48\3E\00\60

used (ACK 06h)



\48\3E\06\60

compose:



\48\60

enter an ASCII character and scan End Selection
(Appendix A)

NAK

- activating ACK or NAK activates the ACK/NAK protocol
- a NAK (negative acknowledge) from the host system indicates an unsuccessful transmission attempt
- "beep after transmission" if good read ("beep before transmission" also possible - see section 7, "Operating settings")
- 6 error beeps after 3 unsuccessful transmission attempts (3 NAKs) or if no ACK (positive acknowledge) received from the host system before end of timeout
- ENQ/ACK/NAK exclusive with XON/XOFF and RTS/CTS

not used (*)



compose:

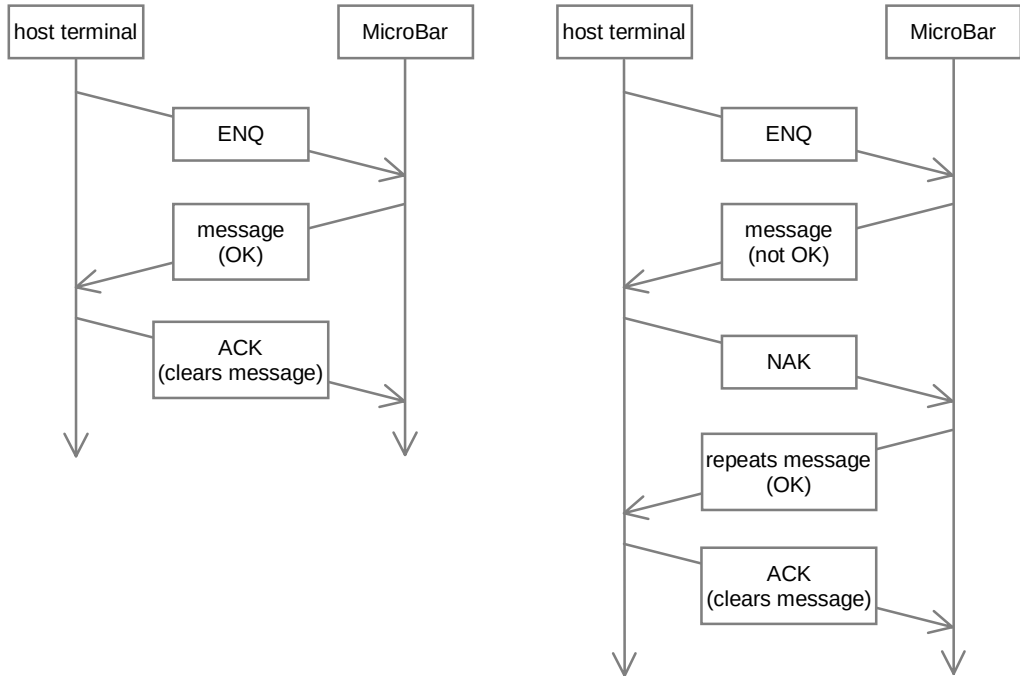


used (NAK 15h)



enter an ASCII character and scan End Selection
(Appendix A)

Typical ENQ / ACK / NAK scenarios



XON / XOFF software protocol

- the host system controls the flow of data from the reader
- XOFF (ASCII character DC3) from the host at the end of the hardware/software protocols time-out interrupts transmission
- XON (ASCII character DC1) restarts the data flow
- "beep after transmission" if good read ("beep before transmission" also possible - see section 7, "Operating settings")
- XON/XOFF exclusive with ENQ/ACK/NAK and RTS/CTS



RTS / CTS hardware protocol

- MicroBar acts as a standard terminal and manages the RTS/CTS lines when configured
- CTS is tested before transmission of each character - data is only transmitted when CTS is activated (high)
- if CTS not used by the host system, MicroBar keeps the level high
- RTS is activated (high) before data is transmitted
- RTS is deactivated (low) when message transmission is completed after the specified timeout
- "beep after transmission" if good read ("beep before transmission" also possible - see section 7 "Operating settings")
- 6 error beeps if no response from the host system before end of timeout
- RTS/CTS exclusive with ENQ/ACK/NAK and XON/XOFF





RTS/CTS not active : MicroBar maintains the low level on the RTS pin

RTS/CTS active : MicroBar applies the normal RTS/CTS protocol



- RTS/CTS not active : MicroBar maintains the high level on the RTS pin
- RTS/CTS active : MicroBar reverses the levels (but not the data) of the RTS/CTS protocol (high becomes low and low becomes high)

LRC (longitudinal redundancy check)

- checksum which your application can use to validate the data transmitted to the host system
 1. XOR is calculated for the hex values of the first two ASCII characters received
 2. XOR is calculated for the next character against the result of the previous XOR calculation (repeated for all characters in the message)
 3. two ASCII characters corresponding to the two digits of the final hex value are added to the end of the transmitted message

4. Output transmission settings RS-232

Example (data received = C O D E - 3 9 <CR> <LF>)

ASCII	hex	binary
C	43h	0100 0011
O	4Fh	0100 1111
...	XOR :	0000 1100
D	44h	0100 0100
...	XOR :	0100 1000
E	45h	0100 0101
...	XOR :	0000 1101
-	2Dh	0010 1101
...	XOR :	0010 0000
3	33h	0011 0011
...	XOR :	0001 0011
9	39h	0011 1001
...	XOR :	0010 1010
<CR>	0Dh	0000 1101
...	XOR :	0010 0111
<LF>	0Ah	0000 1010
...	XOR :	0010 1101

- hex value for 0010 1101 = 2Dh
 - transmitted message = C O D E - 3 9 <CR> <LF> 2 D



inter-character delay (output to host)

- avoids dropping characters if transmitting decoded data too fast for the host system

none (*)



20 ms



40 ms



compose (ms): [range: 1-999 ms]



10 ms



30 ms



50 ms



enter a number string and scan End Selection
(Appendix C)

inter-message delay (output to host)

- gives the host system enough time to process each message received, especially when the host must act on end-of-message control codes such as <Enter>
- "rolling buffer" operation :
 1. a new inter-message delay cycle is activated each time the MicroBar receives a message
 2. if the MicroBar receives a new message before the end of the inter-message delay, it stores the previous message in a "rolling buffer" memory and the cycle starts again
 3. when no more messages are received before the end of an inter-message delay cycle, all the messages stored in the buffer are transmitted to the host system (First-In-First-Out, each message separated by the inter-message delay)

none (*)



100 ms



500 ms



1000 ms (1 sec)



2000 ms (2 sec)



5000 ms (5 sec)



compose (ms): [range: 1-6000 ms]



enter a number string and scan End Selection
(Appendix C)

message length information

[message_length] [preamble] [input_id] [symbology_id] <data> [input_id] [postamble]
[LRC]

- message length = all characters transmitted :
[preamble] + [input_id] + [symbology_id] + <data> + [input_id] + [postamble]
+ [LRC]

not transmitted (*)



\4A\4B\00\60

transmitted as 2 ASCII decimal digits



\4A\4B\0F\60

- available for 99 characters max

transmitted as 2 ASCII hexadecimal digits



\4A\4B\07\60

available for 255 characters max

transmitted as 1 ASCII decimal digit



\4A\4B\0B\60

- available for 9 characters max

transmitted as 1 ASCII hexadecimal digit



\4A\4B\03\60

available for 15 characters max

transmitted as 8-bit value



- available for 255 characters max

transmitted as 16-bit value



available for 65535 (64 K) characters max (most significant byte transmitted first)

preamble

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						

keyboard wedge

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						



- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- maximum = 10 characters
- wedge PC : for characters with ASCII values from 01h to 1Fh (SOH to US), the value transmitted corresponds to <Ctrl> + <letter> (SOH = <Ctrl A>, STX = <Ctrl B>, . . .)

character	transmitted as
SOH	<Ctrl A>
STX	<Ctrl B>
ETX	<Ctrl C>
EOT	<Ctrl D>
ENQ	<Ctrl E>
ACK	<Ctrl F>
BELL	<Ctrl G>
BS	<Ctrl H>

character	transmitted as
HT	<Ctrl I>
LF	<Ctrl J>
VT	<Ctrl K>
FF	<Ctrl L>
CR	<Ctrl M>
SO	<Ctrl N>
SI	<Ctrl O>
DLE	<Ctrl P>
DC1	<Ctrl Q>
DC2	<Ctrl R>
DC3	<Ctrl S>
DC4	<Ctrl T>
NACK	<Ctrl U>
SYN	<Ctrl V>
ETB	<Ctrl W>
CAN	<Ctrl X>
EM	<Ctrl Y>
SUB	<Ctrl Z>
ESC	<Ctrl [>
FS	<Ctrl \ >
GS	<Ctrl] >
RS	<Ctrl ^ >
US	<Ctrl _ >

RS-232

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						



4. Output transmission settings preamble



- enter an ASCII string and scan End Selection (Appendix A)
- maximum = 10 characters

preamble / postamble

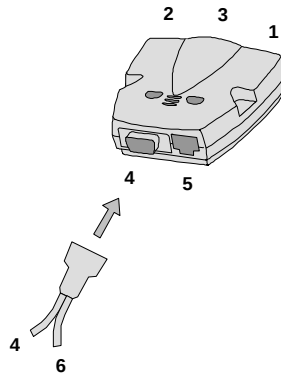
[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						



input port identifier

```
[message_length] [preamble] [input_id] [symbology_id] <data> [input_id] [postamble]
[LRC]
```

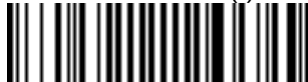
- the MicroBar can send an input identification string before or after the barcode data to identify the input port (and input device !) used



position

```
[message_length] [preamble] [input_id] [symbology_id] <data> [input_id] [postamble]
[LRC]
```

before barcode data (*)



\4A\46\00\00\01\60

after barcode data



\4A\46\00\00\02\60

port 1 (wedge Y cable / RS-232) / port 3 (wedge 2 cables)

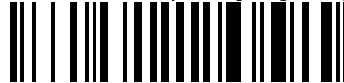
- 1 pen input device
- input port depends on host interface / cable configuration

none (*)



\4A\46\00\01\00\60

compose: [range: 15]



\4A\46\01\01\60

- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- maximum = 15 characters

port 4

- true RS-232 C input device, undecoded laser gun, CCD scanner in laser emulation, 1 pen, 2 pens with dual pen adaptor
- dual pen adaptor provides an extra port (port 6) for port 4

none (*)



\4A\46\00\04\00\60

compose: [range: 15]



\4A\46\01\04\60

- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- maximum = 15 characters

port 5

- RS-232 TTL input device, undecoded laser gun, CCD scanner in laser emulation, 1 pen, undecoded magstripe reader

none (*)



\4A\46\00\05\00\60

compose: [range: 15]



\4A\46\01\05\60

- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- maximum = 15 characters

port 6

- 1 pen in extra port provided by a dual pen adaptor on port 4

none (*)



\4A\46\00\06\00\60

compose: [range: 15]



\4A\46\01\06\60

- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- maximum = 15 characters

symbology identifier

[message_length]
[LRC]

[preamble]

[input_id]

[symbology_id]

<data>

[input_id]

[postamble]

AIM

[message_length]
[LRC]

[preamble]

[input_id]

[symbology_id]

<data>

[input_id]

[postamble]

- optional 3-character symbology identifiers standardized by the AIM Committee
Example: "] A 0 " identifies standard Code 39 without check digit
- refer to the official AIM documentation on symbology identifiers for full information on the different processing options supported

symbology	]	<symbology_id>	<processing_option>
Ames Code	]	X	0
Codabar	]	F	0, 2, 4
Code 11	]	H	0, 1, 2
Code 39	]	A	0, 1, 2, 4
Code 93	]	G	0
Code 128/EAN 128	]	C	0, 1
Interleaved 2 of 5	]	I	0, 1, 2
Matrix 2 of 5	]	X	0
Standard 2 of 5	]	S	0, 1, 2
MSI Code	]	M	0, 1
Plessey Code	]	P	0
Telepen	]	B	0, 1
UPC/EAN (1)	]	E	0, 3, 4
UPC/EAN (2)	]	X	0

(1) UPC/EAN "standard" lengths = 8, 13, 15 (add-on 2), 18 (add-on 5) characters
(2) UPC/EAN other lengths (no check digit, . . .)



- activates AIM symbology identifier transmission for all symbologies

custom

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						



- activates custom symbology identifier transmission for all symbologies

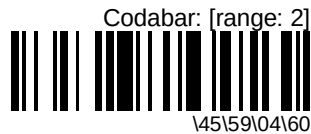
compose

- symbology identifier = 1 or 2 characters



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)

default = *

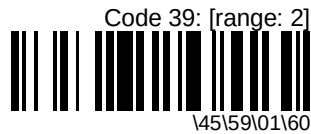


- enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
- default = D



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)

default = *



- enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
- default = *

Code 93: [range: 2]



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)

default = D

Code 128 / EAN 128: [range: 2]



- enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
- default = D

EAN-8: [range: 2]



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)

default = FF

EAN-13: [range: 2]



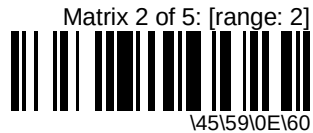
- enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
- default = F

Interleaved 2 of 5: [range: 2]



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)

default = I



- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- default = D



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
default = D



- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- default = D



enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
default = D



- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- default = *

UPC-A: [range: 2]



\45\59\08\60

enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)

default = A

UPC-E: [range: 2]



\45\59\09\60

- enter an ASCII / Special Wedge Characters string
and scan End Selection (Appendix A and B)
- default = E

data editing (EasySet 3.0 and later)

- the MicroBar can edit the data it receives before it transmits it to the host system
- define up to 7 input scenarios to filter out the data you want to edit
- the order in which you define the scenarios is important (the MicroBar compares incoming data with each scenario in turn and edits the data for the first matching scenario it finds)
- data editing is applied before keyboard remapping !!! (if applicable)
- if the data editing and scale solicitation functions have an identical input scenario, data editing has priority (scale solicitation will not occur !!!)

reset scenarios in MicroBar



resets all 7 scenarios and editing in the MicroBar
(recommended before you define new scenarios
and editing)

define:

this option only available
with EasySet

1. define the input data you want to intercept for editing :
 - select any combination of input port, input type (symbolology, RS-232...), message length, input mask
 2. define the editing you want to apply to this input :
 - select a scenario
 - define the editing for the selected scenario
- input mask :
- you can use up to 24 consecutive characters (or more if you start with a string of '?' wildcards to indicate the start position of the actual filter)
 - '*', '?', '<num>', '<not num>' are wildcards

- use a backslash '\' in front of '*', '?' and '\' if you want to include them as actual characters in your filter ('*' = '*', '\\?' = '?', '\\\' = '\')
- '<num>' and '<not num>' are available in a right-click popup menu in the "Input mask" field
- make sure that the input scenarios you define actually correspond to incoming data conditions !!!
 - correct input port (no value = 'any port')
 - correct input type (no value = 'any input type')
 - correct message length (no value = 'any length')
 - correct input mask (no value = 'any characters')
- data editing is applied before keyboard remapping !!! (if applicable)
- if the data editing and scale solicitation functions have an identical input scenario, data editing has priority (scale solicitation will not occur !!!)

data reformatting (EasySet 2.01, 2.02, 2.03)

- reformats the data received before transmission to the host system
- input data is selected for reformatting by user-defined input scenarios
- if the MicroBar finds a matching scenario, it applies the reformatting for that scenario
- scenarios can be saved in a file



resets all 5 scenarios and reformatting in the MicroBar

define reformatting

this option only available
with EasySet

1. define the input data for reformatting
 - select any combination of input port, input type (symbology, RS-232...), message length or input mask
 - input mask : use up to three consecutive constants and wildcards * and ? (a wildcard preceded by a backslash is interpreted as a constant : * = *, \? = ?, \\ = \)
2. define the output reformatting you want to apply to this input
 - select a scenario
 - define the reformatting for the selected scenario

Overview

The MicroBar receives data from a wide range of input devices and transmits this data to the host system. The data can be transmitted exactly as it is received, or it can be "reformatted" (modified to suit the requirements of the host application) by the MicroBar before it is transmitted.

There are 2 "pages" in the reformatting tool : "Define input" and "Define output".

In the "Define input" page, you can define which input data you want the MicroBar to reformat (up to 5 input scenarios for each port).

In the "Define output" page, you can select 5 different input scenarios and define the output reformatting you want the MicroBar to apply for each scenario.

When the MicroBar receives an input string, it looks at each of the 5 scenarios in turn (starting with scenario 1) to see if there is a matching scenario.

As soon as the MicroBar finds a matching scenario, it applies the reformatting defined for that scenario and sends the reformatted data string to the host system.

General functions

Open opens a file containing a saved set of scenarios and displays the scenarios in the "Reformat data" window

Save saves the current set of scenarios to a file (the associated reformatting is not saved)

Reset all resets all the scenarios and reformatting in the "Reformat data" window (does not affect scenarios and reformatting in the MicroBar)



Define input displays the "Define input" page where you define the different input scenarios you want the MicroBar to detect for reformatting

Define output displays the "Define output" page where you choose one of the input scenarios and define the reformatting you want the MicroBar to apply to the matching input string

OK sends the currently selected scenario and associated reformatting command to the setup sheet / MicroBar (replaces the corresponding scenario in the MicroBar)

Cancel closes the "Reformat data" window

"Define input" page

1 - Define input

choose a **Port** button to tell the MicroBar to reformat input data received on that port

Example: select "Port 4" if that's where the data to be reformatted is coming in

choose an **Input type** if you only want reformatting for a given symbology, RS-232 input or magstripe input

Example: select "Code 128" if you only want to reformat Code 128 data

if you leave the "Input type" field empty, the MicroBar will reformat any type of data received on the selected port

define a **Length** if you only want the MicroBar to reformat input data of a certain length

Example: enter "10" if you only want to reformat messages if they are 10 characters long

if you leave the "Length" field empty, the MicroBar will reformat input messages of any length received on the selected port

enter an **Input mask** if the input messages you want to reformat can be identified by specific characters in a specific position

Example: enter "+H" if you only want to reformat messages which begin with the characters "+H"

if you leave the "Input mask" field empty, the MicroBar will reformat input messages with any characters received on the selected port

Input mask restrictions

You can only enter 1, 2 or 3 consecutive "search" characters in a mask.

The total number of characters in the mask cannot be longer than the fixed length selected in the "Length" field (if applicable).

Using wildcards in the input mask

You can use character substitution wildcards like the ones used in file searches etc. to indicate where the search string is located in the message:

- * replaces any number of characters (any characters)
- ? replaces 1 character (any character)

It is not necessary to add wildcard characters after the string of 1, 2 or 3 character constants.

If you want to look for the characters * ? \ in the input message, you must put an additional backslash in front of these characters to distinguish them from wildcard characters.

Examples	Input mask	input string	reformatting ?
	???gg	abcggdefg	yes
		aggdefg	no
	*gg	abcggdefg	yes
		aggdefg	yes
	*gg	*ggdefg	yes
		aggdefg	no

"Define output" page

2 - Choose a scenario

5 list boxes contain a list of all the input scenarios defined in the "Define input" page

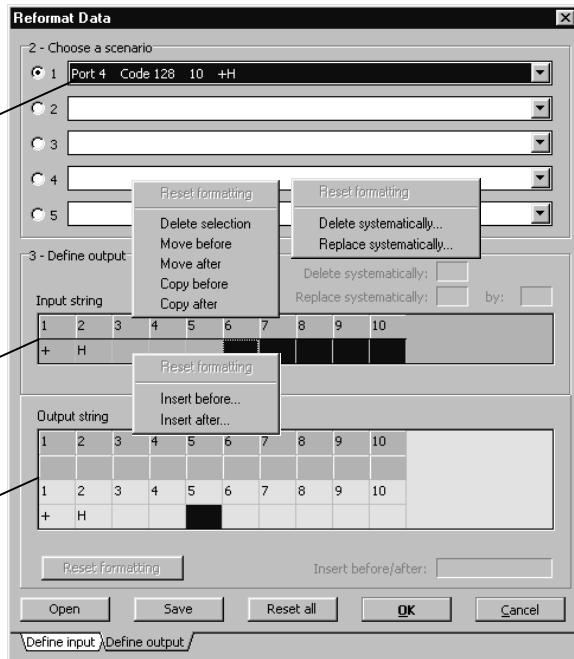
choose a scenario corresponding to the input data you want to reformat

Example: select " Port 4 Code 128 10 +H "

3 - Define output

Input string (blue) shows the original format of the data received including message length and input mask characters if applicable

Output string shows changes made to the input string (blue) and the resulting output string (yellow) after reformatting



1. select the input scenario you want the MicroBar to detect
2. define the reformatting you want to apply for that scenario according to what you want to do :
 - select one or more characters in the **Input string** and choose a menu option
 - click in the **Delete / Replace systematically** area and choose a menu option
 - select a character location in the **Output string** and choose a menu option
3. click **OK** to send the <input scenario + reformatting> to the MicroBar

grid length

Before reformatting, grid length is limited to 200 characters if no length is specified in the "Length" field for the selected scenario.

The maximum grid length is 2000 characters (maximum input string length).

Reformatting restrictions

You can only perform one reformatting action at a time on the chosen input scenario.

You can only send one input-scenario-with-reformatting combination at a time to the MicroBar.

The MicroBar stores 5 input-scenario-with-reformatting combinations.

If you send scenario 1 and associated reformatting, you will replace the current scenario 1 stored in the MicroBar, if you send scenario 2, you will replace scenario 2, etc.

Unexpected results

As more than one scenario in the MicroBar can match the input string received, this can give unexpected reformatting results.

Example

- you define reformatting for scenario 4
- both scenario 2 and scenario 4 match the input string
- the MicroBar applies the reformatting defined for scenario 2 (the first matching scenario)

To limit the risk of this happening, you can :

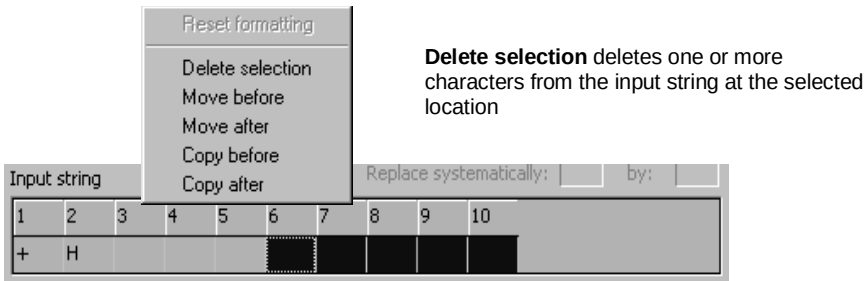
- use the **data reformatting - reset** command in EasySet to clear the 5 scenarios in the MicroBar
- always start with scenario 1, then scenario 2, etc. when you define new reformatting

Reformatting menu options

The different reformatting functions are provided in 3 popup menus depending on where you click in the "Define output" page.

Select the **Reset formatting** pushbutton or menu option to reset formatting for the selected scenario.

Input string popup menu



Move before / after moves one or more characters in the input string to a new location in the output string

in the output string, click the location where you want to move the characters – they will be moved before or after this location as requested

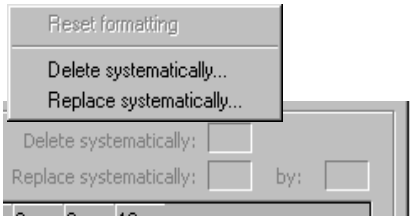
Copy before / after copies one or more characters in the input string to another location in the output string

in the output string, click the location where you want to copy the characters – they will be copied before or after this location as requested

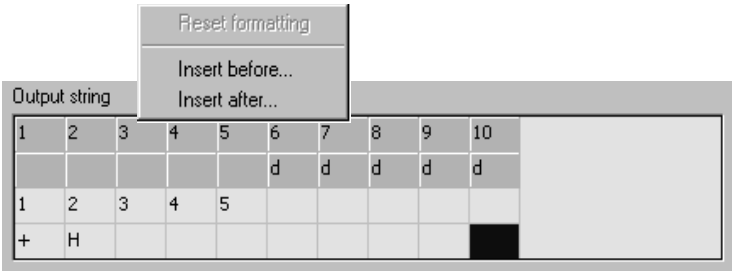
Delete / Replace systematically popup menu

Delete systematically deletes all occurrences of the specified character from the input string regardless of their location

Replace systematically replaces all occurrences of a given character in the input string by the specified replacement character



Output string popup menu



Insert before / after systematically inserts one or more characters in the output string

in the output string, click the location where you want to insert the new characters – they will be inserted before or after this location as requested

Input string grid

input string index :

input string with input mask characters if applicable :

Input string										Replace systematically
1	2	3	4	5	6	7	8	9	10	
+	H									

Output string grid

input string index:

reformatting action indicators :

output string index :

output string with specified characters if applicable :

Output string									
1	2	3	4	5	6	7	8	9	10
					d	d	d	d	d
1	2	3	4	5					
+	H								

reformatting action indicators

Reformatting actions are indicated in the second row of the Output string grid by reformatting action indicator characters.

reformatting action indicator	action	popup menu
d	Delete selection	Input string
m	Move before / after	Input string
c	Copy before / after	Input string
(d)	Delete systematically	Delete / Replace systematically
(r)	Replace systematically	Delete / Replace systematically
i	Insert before / after	Output string

postamble

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						

keyboard wedge

none



\45\54\3E\00\60

Enter (*)



\45\54\04\02\01\60

Carriage Return



\45\54\04\02\02\60

Tab



\45\54\04\02\08\60

Field Advance



\45\54\04\02\04\60

Field Exit



\45\54\04\02\05\60

Down Arrow



\45\54\04\02\0F\60

compose:



\45\54\60

- enter an ASCII / Special Wedge Characters string and scan End Selection (Appendix A and B)
- maximum = 10 characters
- wedge PC : for characters with ASCII values from 01h to 1Fh (SOH to US), the value transmitted corresponds to <Ctrl> + <letter> (SOH = <Ctrl A>, STX = <Ctrl B>, . . .)

character	transmitted as
SOH	<Ctrl A>
STX	<Ctrl B>
ETX	<Ctrl C>
EOT	<Ctrl D>
ENQ	<Ctrl E>
ACK	<Ctrl F>
BELL	<Ctrl G>
BS	<Ctrl H>
HT	<Ctrl I>
LF	<Ctrl J>
VT	<Ctrl K>
FF	<Ctrl L>
CR	<Ctrl M>
SO	<Ctrl N>
SI	<Ctrl O>
DLE	<Ctrl P>
DC1	<Ctrl Q>
DC2	<Ctrl R>
DC3	<Ctrl S>
DC4	<Ctrl T>

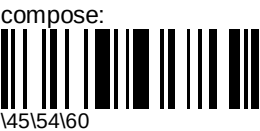
character	transmitted as
NACK	<Ctrl U>
SYN	<Ctrl V>
ETB	<Ctrl W>
CAN	<Ctrl X>
EM	<Ctrl Y>
SUB	<Ctrl Z>
ESC	<Ctrl [>
FS	<Ctrl \ >
GS	<Ctrl] >
RS	<Ctrl ^ >
US	<Ctrl _ >

RS-232

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						



4. Output transmission settings postamble



enter an ASCII string and scan End Selection
(Appendix A)
maximum = 10 characters

preamble / postamble

[message_length]	[preamble]	[input_id]	[symbology_id]	<data>	[input_id]	[postamble]
[LRC]						



5. Symbologies

- symbology = bar code type or family (e.g. Code 39, UPC, EAN)
- the following barcode readers can be connected to the MicroBar :
 - pen barcode readers
 - laser / CCD barcode readers in wand / laser emulation
- activate the symbologies you need and modify the settings for your symbologies if required
- to optimize performance, only activate symbologies you need !!! (deactivate the Code 39 and UPC/EAN default symbologies if you don't need them)
- if your input device is a ScanPlus in wand / laser emulation :
 - activate the same symbologies and code lengths for the ScanPlus and the MicroBar
 - disconnect the ScanPlus to make sure that "ScanPlus transparent configuration mode" is disabled before you configure the ScanPlus



- deactivates all the symbologies activated
- use the "not active" options to deactivate individual symbologies
- does not reset individual parameter settings for each symbology (when you reactivate a symbology, you recover the parameter settings stored in memory for that symbology when it was disabled - use reset factory defaults to reset all the symbology parameters to their factory default settings)

Ames Code

- numerical symbology



Codabar

- numerical symbology

not active (*)



active



start/stop

not transmitted (*)



a, b, c, d



A, B, C, D



a, b, c, d/

t, n, *, e



DC1, DC2, DC3, DC4



CLSI library system

- spaces inserted after characters 1, 5, 10 in the 14-character label (used in the USA by libraries using the CLSI system)
Example: " 39990000192148 " is transmitted as " 3 9990 00019 2148 "
- start/stop can be transmitted or not transmitted as required



- inserts spaces

check digit (AIM recommendation)

- AIM has a recommended check character for Codabar
- each Codabar data character (including Start/Stop) has a value assigned to it:
0 = 0 1 = 1 2 = 2 3 = 3 4 = 4 5 = 5 6 = 6 7 = 7 8 = 8 9 = 9 - = 10 \$
= 11 : = 12 / = 13 , = 14 + = 15 A = 16 B = 17 C = 18 D = 19
- the values are added and the check is calculated: check = [(next multiple of 16) - (sum of assigned AIM values)]

Example

data characters:	A 0 1 2 3 4 B
AIM values = 16 + 0 + 1 + 2 + 3 + 4 + 17:	43
next multiple of 16:	48
check = 48 - 43:	5
final message:	A 0 1 2 3 4 5 B

not used (*)



\46\56\60

checked and transmitted



\46\54\60

checked but not transmitted



\46\55\60

concatenation

- 2 adjacent Codabar codes scanned in a single sweep can be transmitted as a single message
- if the scanner is configured to transmit start/stop characters, MicroBar will only transmit first start and last stop (intermediate start/stop characters not transmitted)

not active



\43\56\60

- no concatenation

active

no start/stop restrictions (*)



\43\57\60

concatenates adjacent codes

intermediate start/stop characters not transmitted



- intermediate start/stop characters not transmitted
- only concatenates adjacent codes if stop character of first code = start character of second code :

barcode_1 (left)	barcode_2 (right)
<start = C>....(<stop = A>)	(<start = A>)....<stop = D>



intermediate start/stop characters not transmitted
ABC : only concatenates adjacent codes if stop character of first code = D and start character of second code = D :

barcode_1 (left)	barcode_2 (right)
<start = A / B / C / D>....(<stop = D>)	(<start = D>)....<stop = A / B / C / D>

transmit



Code 11

- numerical symbology

not active (*)



active



check digits

1 digit (*)



2 digits



checked and transmitted (*)



checked but not transmitted

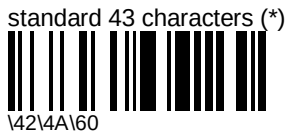


Code 39

- alphanumeric symbology
- letter case not defined - transmitted in upper case
- default format = Standard 43 Characters



format



- extra characters encoded using 1 of 4 control characters (\$, %, /, +) combined with a letter :

Code 39	ASCII	Hex	Dec
%U	NUL	00	0
\$A	SOH	01	1
\$B	STX	02	2
\$C	ETX	03	3
\$D	EOT	04	4
\$E	ENQ	05	5
\$F	ACK	06	6
\$G	BEL	07	7
\$H	BS	08	8
\$I	HT	09	9
\$J	LF	0A	10
\$K	VT	0B	11
\$L	FF	0C	12
\$M	CR	0D	13
\$N	SO	0E	14
\$O	SI	0F	15
\$P	DLE	10	16
\$Q	DC1	11	17
\$R	DC2	12	18
\$S	DC3	13	19
\$T	DC4	14	20
\$U	NAK	15	21
\$V	SYN	16	22
\$W	ETB	17	23
\$X	CAN	18	24
\$Y	EM	19	25
\$Z	SUB	1A	26
%A	ESC	1B	27
%B	FS	1C	28
%C	GS	1D	29
%D	RS	1E	30
%E	US	1F	31
Space	SP	20	32
/A	!	21	33

Code 39	ASCII	Hex	Dec
/B	"	22	34
/C	#	23	35
\$	\$	24	36
%	%	25	37
/F	&	26	38
/G	'	27	39
/H	(	28	40
/I	)	29	41
/J	*	2A	42
+	+	2B	43
/L	,	2C	44
/M	-	2D	45
/N	.	2E	46
/	/	2F	47
0	0	30	48
1	1	31	49
2	2	32	50
3	3	33	51
4	4	34	52
5	5	35	53
6	6	36	54
7	7	37	55
8	8	38	56
9	9	39	57
/Z	:	3A	58
%F	;	3B	59
%G	<	3C	60
%H	=	3D	61
%I	>	3E	62
%J	?	3F	63
%V	@	40	64
A	A	41	65
B	B	42	66
C	C	43	67

Code 39	ASCII	Hex	Dec
D	D	44	68
E	E	45	69
F	F	46	70
G	G	47	71
H	H	48	72
I	I	49	73
J	J	4A	74
K	K	4B	75
L	L	4C	76
M	M	4D	77
N	N	4E	78
O	O	4F	79
P	P	50	80
Q	Q	51	81
R	R	52	82
S	S	45	83
T	T	54	84
U	U	55	85
V	V	56	86
W	W	57	87
X	X	58	88
Y	Y	59	89
Z	Z	5A	90
%K	[	5B	91
%L	\	5C	92
%M	]	5D	93
%N	^	5E	94
%O	_	5F	95
%W	`	60	96
+A	a	61	97
+B	b	62	98
+C	c	63	99
+D	d	64	100
+E	e	65	101

Code 39	ASCII	Hex	Dec
+F	f	66	102
+G	g	67	103
+H	h	68	104
+I	i	69	105
+J	j	6A	106
+K	k	6B	107
+L	l	6C	108
+M	m	6D	109
+N	n	6E	110
+O	o	6F	111
+P	p	70	112
+Q	q	71	113
+R	r	72	114
+S	s	73	115
+T	t	74	116
+U	u	75	117
+V	v	76	118
+W	w	77	119
+X	x	78	120
+Y	y	79	121
+Z	z	7A	122
%P	{	7B	123
%Q		7C	124
%R	}	7D	125
%S	~	7E	126
%T	DEL	7F	127

start/stop





accepted characters



check digit



modulo 43

- provides extra validation of data



French CIP

- French pharmaceutical industry
- only used for codes with 7 characters

checked and transmitted



checked but not transmitted



Italian CPI

- Italian pharmaceutical industry
- transmitted as standard Code 39 if checksum not validated

checked and transmitted



checked but not transmitted



multiread

- multiread stores a number of bar codes in the MicroBar memory before transmitting them as a single message :
 1. MicroBar receives a "multiread" string (Code 39 bar code beginning with multiread indicator character) and stores it in memory
 2. each new multiread string is added to the multicode message in the MicroBar memory

3. MicroBar transmits the full multicode message to the terminal or host when it receives a string which does not begin with the multiread indicator character or if the memory is full (maximum 255 characters)

- the multiread indicator character is not transmitted with the message
- default multiread indicator character = <space>

not active (*)



\42\56\3E\00\60

active (default multiread character = space)



\42\56\00\60

compose multiread character:



\42\56\60

enter an ASCII character and scan End Selection
(Appendix A)

Code 93

- alphanumeric full ASCII symbology - letter case defined

not active (*)



active



multiread

- multiread stores a number of bar codes in the MicroBar memory before transmitting them as a single message :
 1. MicroBar receives a "multiread" string (Code 39 bar code beginning with multiread indicator character) and stores it in memory
 2. each new multiread string is added to the multicode message in the MicroBar memory
 3. MicroBar transmits the full multicode message to the terminal or host when it receives a string which does not begin with the multiread indicator character or if the memory is full (maximum 255 characters)
- the multiread indicator character is not transmitted with the message
- default multiread indicator character = <space>

not active (*)



active (default multiread character = space)



compose multiread character:



enter an ASCII character and scan End Selection
(Appendix A)

Code 128 / EAN 128

- alphanumeric full ASCII symbology - letter case defined
- "EAN 128" = Code 128 with the FNC1 character in the first position
- EAN 128 is auto-discriminating with Code 128 (recognition of the FNC1 start character used)

not active (*)



\41\5B\60

active



\41\5A\60

EAN 128 identifier

- the]C1 AIM identifier for EAN 128 is automatically added by default in front of EAN 128 bar codes

include]C1 identifier (*)



\47\5A\01\60

remove]C1 identifier



\47\5A\00\60

CIP 128 French pharmaceutical codes

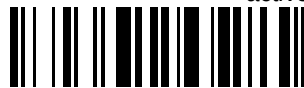
- embedded CIP 39 data
- fixed length 14 characters
- Code 128 character set C

not active (*)



\47\5D\00\60

active



\47\5D\01\60

FNC1 separator character (EAN 128 norms)

- default = GS function character (ASCII 29)
- used as separator when multiple identifiers and their fields are concatenated

GS (ASCII 29) (*)



\46\52\3E\1D\60

compose:



\46\52\60

- enter an ASCII character and scan End Selection (Appendix A)
- useful for keyboard wedge interfaces where the GS character can not be transmitted

ISBT 128 concatenation

not active (*)



transmit 2 concatenated codes



- only transmits if 2 codes are read before end of timeout or trigger release
- does not transmit if only 1 code

transmit 1 code / 2 concatenated codes



transmits 1 code or 2 concatenated codes if a second code is read during the next 5 scans

Interleaved 2 of 5

- numerical symbology



check digit

- especially recommended for variable length Interleaved 2 of 5 and if "consecutive same read data validation" (data decoding security parameters) is not activated



mod 10



French CIP HR

- French pharmaceutical industry
- only used for codes with 7 characters

checked and transmitted



\43\41\60

checked but not transmitted



\43\42\60

barcode length

= <barcode data> + [check digit]

- minimum length possible = 2 characters
- default = 1 fixed length 14 characters !!!
- if the codes in your application have fixed lengths, use one of the "fixed lengths" options !!!
- Interleaved 2 of 5 always encodes an even number of characters
- for codes with an odd number of characters, you can add a last character printed as 5 narrow bars (not transmitted)

1 fixed length = 14 characters (*)



\43\44\0E\60

compose 1 or 2 or 3 fixed lengths:



- enter a number string and scan End Selection for each length and scan End Selection twice to finish (Appendix C)
- defining fixed lengths provides the best performance and security if the codes in your application have fixed lengths
- minimum length possible = 2 characters
- all 3 lengths = "0" resets default length (14 characters)

get fixed lengths from codes: [range: 1-3]



enter a number string and scan End Selection (Appendix C)

only for codes longer than 4 characters !!! (use "compose 1 or 2 or 3 fixed lengths" option for short codes)

defining fixed lengths provides the best performance and security if the codes in your application have fixed lengths

1. enter the number of fixed lengths you want to define (1, 2 or 3)
2. read 1, 2 or 3 Interleaved 2 of 5 bar codes with the lengths used in your application

the MicroBar will only read codes with the lengths you have defined

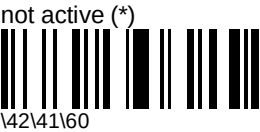
any length



- cancels any fixed length settings and accepts any valid length (2 or more characters)

Matrix 2 of 5

- numerical symbology



MSI Code

- numerical symbology

not active (*)



active



check digit

mod 10

checked and transmitted (*)



checked but not transmitted



double mod 10

checked and transmitted



checked but not transmitted



Plessey Code

- numerical symbology



check digit



Standard 2 of 5

- numerical symbology
- default format = Identicon (6 start/stop bars)
- also referred to as "Straight 2 of 5" and "Industrial 2 of 5"

not active (*)



\41\51\60

active



\41\50\60

format

Identicon (6 start/stop bars) (*)



\43\47\60

Computer Identics (4 start/stop bars)



\43\46\60

check digit mod 10

not used (*)



\43\4A\60

checked and transmitted



\43\48\60

checked but not transmitted



barcode length

= <barcode data> + [check digit]

- minimum length possible = 3 characters
- default = 1 fixed length 6 characters !!!
- if the codes in your application have fixed lengths, use one of the "fixed lengths" options !!!

1 fixed length = 6 characters (*)



compose 1 or 2 or 3 fixed lengths:



- enter a number string and scan End Selection for each length and scan End Selection twice to finish (Appendix C)
- defining fixed lengths provides the best performance and security if the codes in your application have fixed lengths
- minimum length possible = 3 characters
- all 3 lengths = "0" resets default length (6 characters)

get fixed lengths from codes: [range: 1-3]



enter a number string and scan End Selection
(Appendix C)

only for codes longer than 4 characters !!! (use
"compose 1 or 2 or 3 fixed lengths" option for
short codes)

defining fixed lengths provides the best
performance and security if the codes in your
application have fixed lengths

1. enter the number of fixed lengths you want
to define (1, 2 or 3)
2. read 1, 2 or 3 Standard 2 of 5 bar codes
with the lengths used in your application

the MicroBar will only read codes with the lengths
you have defined



- cancels any fixed length settings and accepts any
valid length (3 or more characters)

Telepen

- alphanumeric full ASCII symbology - letter case defined
- default format = ASCII

not active (*)



\42\43\60

active



\42\42\60

format

ASCII (*)



\47\4C\60

numeric



\47\4B\60

UPC / EAN

- numerical symbology
- deactivate EAN-13 for unregular UPC-E with number system equal to 1 (usually the first printed character)

all active (*)



reactivates all deactivated UPC/EAN families

not active



UPC-A deactivated



UPC-E deactivated



EAN-8 deactivated



EAN-13 deactivated



add-on digits

not required but transmitted if read (*)



required and transmitted



add-on 2

not active (*)



active



add-on 5

not active (*)



active



check digit

<leading character>	<number system>	<data>	<check digit>
---------------------	-----------------	--------	---------------

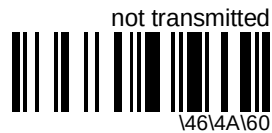
UPC-A



UPC-E



EAN-8



EAN-13

transmitted (*)



not transmitted



UPC number system

<leading character>	<number system>	<data>	<check digit>
---------------------	-----------------	--------	---------------

UPC-A

- | |
|---|
| <ul style="list-style-type: none">- regular UPC-A has a transmitted number system equal to 0- to transmit the additional leading character (country code), select the "UPC-A transmitted as EAN-13" option |
|---|

transmitted (*)



not transmitted



UPC-E

- deactivate EAN-13 for unregular UPC-E with number system equal to 1 (usually the first printed character)

transmitted (*)



not transmitted

**re-encoding UPC-A, UPC-E, EAN-8**

<leading character> <number system> <data> <check digit>

- converts decoded data to other code formats
- transmission only takes into account the parameters available for the target bar code format
- regular UPC-A has a transmitted number system equal to 0
- to transmit the additional leading character (country code), select the "UPC-A transmitted as EAN-13" option

UPC-A transmitted as EAN-13 (*)



UPC-A transmitted as UPC-A



UPC-E transmitted as UPC-E (*)



EAN-8 transmitted as EAN 8 (*)



UPC-E transmitted as UPC-A

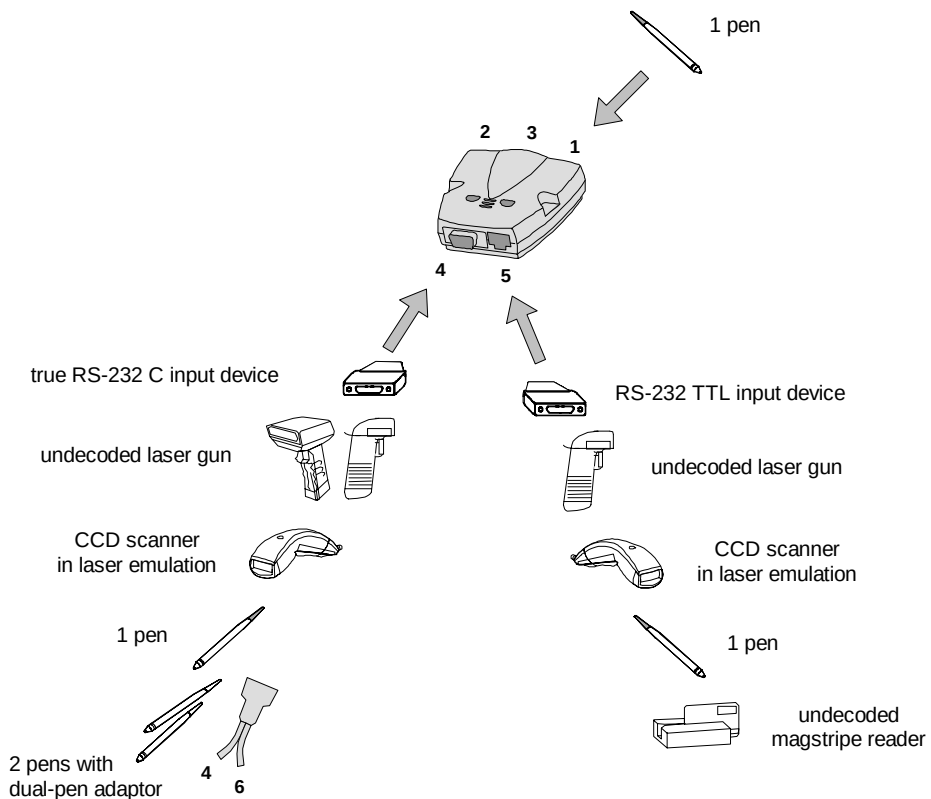


EAN-8 transmitted as EAN-13



6. Input devices

- the following input devices can be connected to the MicroBar
 - pen barcode readers
 - laser / CCD barcode readers in wand / laser emulation
 - undecoded magstripe readers
 - RS-232 input devices (electronic scales, barcode readers, magstripe readers, . . .)



laser scanners / CCD scanners in laser emulation

- laser scanners or CCD scanners in laser emulation connected to ports 4 and 5 of the MicroBar

activation

- IMPORTANT !
- Do not read the 'port 4' and 'port 5' activation codes in this section if the scanner you are using is already reading correctly as this could deactivate the MicroBar for your scanner !!!
- if this happens, use another barcode reader (a pen for example) to read each of the port activation codes in this section until you reactivate the MicroBar for your scanner

port 4

Intermec Sabre 1550 X01 (Intermec interface) (*)



\4B\5D\00\04\60

Sabre 1550 X01 (Intermec interface) with
squeeze-to- release cable connector (default
scanner for port 4)

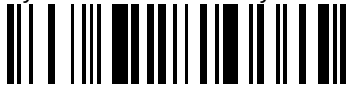
Intermec Sabre 1550 X05 (standard interface)



\4B\5D\02\04\60

- Sabre 1550 X05 (standard interface) with
squeeze-to- release cable connector

Symbol LS 3000 LR only



\4B\5D\01\04\60

port 5

Intermec Sabre 1550 X05 (standard interface) (*)



\4B\5D\02\05\60

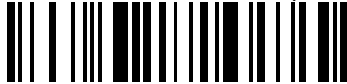
Sabre 1550 X05 (standard interface) with 10-pin modular plug (default scanner for port 5)

Symbol LS 3000 LR only



\4B\5D\01\05\60

Intermec Sabre 1550 X01 (Intermec interface)

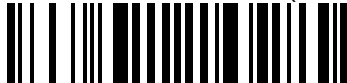


\4B\5D\00\05\60

Sabre 1550 X01 (Intermec interface) with 10-pin modular plug

ports 4 and 5

Intermec Sabre 1550 X05 (standard interface)



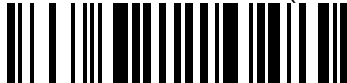
\4B\5D\02\00\60

Symbol LS 3000 LR only



\4B\5D\01\00\60

Intermec Sabre 1550 X01 (Intermec interface)

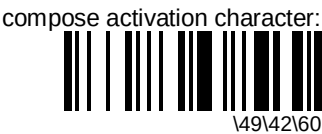


\4B\5D\00\00\60

laser trigger management

RS-232 trigger

- some laser scanners can be activated by a user-defined RS-232 character sent by the host system (software activation)



- enter an ASCII character and scan End Selection (Appendix A)

CTS trigger

- some laser scanners can be activated according to the level of the CTS signal received (voltage level activation)



laser timeout before deactivation

- MicroBar can control the timeout period before deactivation (laser scanners normally remain active until the trigger is released or they reach the end of their timeout)
- valid for all laser trigger configurations



the laser scanner remains active all the time

LASER SAFETY: CAUTION - LASER LIGHT
CAN DAMAGE YOUR EYES. DO NOT STARE
INTO BEAM.



"code not read" message

- informs the host system if the barcode data has not been successfully decoded when the trigger is released or the timeout period is reached (particularly useful to indicate unread codes when scanners are used in autostand mode for example)

not active (*)



compose non-read ASCII character:



- enter an ASCII character and scan End Selection (Appendix A)
- character used to indicate a non-read

compose repetition number: [range: 1-20]



enter a number string and scan End Selection (Appendix C)

number of non-read characters sent to the host system (max 20)

consecutive same read data validation

- ensures correct transmission of data for difficult reading conditions and varying levels of barcode quality (poorly printed labels, variable lengths and no check digit, "fragile" symbologies)
- data is only transmitted after repeated reads give the same result
- fixed length / minimum length settings for the symbologies in your application also increase the reliability of the barcode data transmitted (see "barcode length" for your symbologies)
- increasing the security level reduces the reading speed !!!

single read before transmission (*)



\45\4B\01\60

compose number of consecutive same reads before transmission: [range: 1-10]



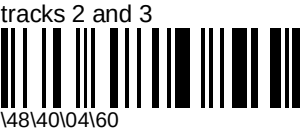
\45\4B\60

- enter a number string and scan End Selection (Appendix C)

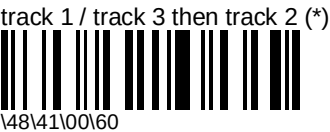
magstripe readers

track selection

- MicroBar can read 1 track or 2 tracks from a possible total of 1, 2, or 3 tracks



transmission sequence



track separator character

- default = # character (ASCII 35)



compose separator character:



- enter an ASCII character and scan End Selection (Appendix A)

start/end sentinels



LRC check digit



scales / weighing machines

scale solicitation

- sends a solicitation message to an input device on port 4 or port 5 (typically scales or a weighing machine) when a matching input scenario is encountered (the scales will then send data for transmission to the host system)
- if the data editing and scale solicitation functions have an identical input scenario, data editing has priority (scale solicitation will not occur !!!)
- use adaptor cable 0-234032-02 to connect an RS-232 device to port 4



deactivates scale solicitation in the MicroBar

define:

this option only available
with EasySet

1. define the input scenario you want to use to trigger scale solicitation :
 - select any combination of input port, input type (symbology, RS-232...), message length, input mask
 2. define the the solicit message you want to send to the scale and specify the port the scale is connected to
- make sure that the input scenario you define actually corresponds to incoming data conditions !!!
 - correct input port (no value = 'any port')
 - correct input type (no value = 'any input type')
 - correct message length (no value = 'any length')
 - correct input mask (no value = 'any characters')

- if the data editing and scale solicitation functions have an identical input scenario, data editing has priority (scale solicitation will not occur !!!)

RS-232 input devices on port 4

- for data received from RS-232 C input devices (electronic scales, barcode readers, . . .) connected to Port 4 of the MicroBar
- use adaptor cable 0-234032-02 to connect an RS-232 device to port 4

enable



enable communication with RS-232 C input devices on port 4

disable (*)



baud rate

300



600



1200



2400

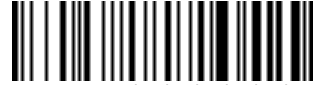


4800



\4A\46\03\04\04\60

9600 (*)



\4A\46\03\04\05\60

19200



\4A\46\03\04\06\60

data bits

seven (*)



\4A\46\04\04\00\60

eight



\4A\46\04\04\01\60

parity

even (*)



\4A\46\05\04\01\60

odd



\4A\46\05\04\02\60

mark



\4A\46\05\04\03\60

space



\4A\46\05\04\04\60

none



\4A\46\05\04\00\60

stop bits

one (*)



\4A\46\06\04\01\60

two



\4A\46\06\04\02\60

ENQ / ACK / NAK

- the RS-232 input device can synchronize communication with the MicroBar :
 1. the RS-232 input device sends an ENQ character (05h) to the MicroBar
 2. the MicroBar answers with an ACK character (06h), even if ENQ / ACK / NAK has not been activated in the MicroBar
 3. the RS-232 input device sends a message to the MicroBar (the message is terminated with the selected end-of-message character)
 4. the MicroBar sends an ACK character to the RS-232 input device at the end of transmission if the message is understood, or a NAK character (15h) if the message is not understood

- there is no hardware/software protocols timeout for RS-232 input devices

ENQ

not active (*)



active (ENQ 05h)



ACK

not active



active (ACK 06h) (*)



NAK

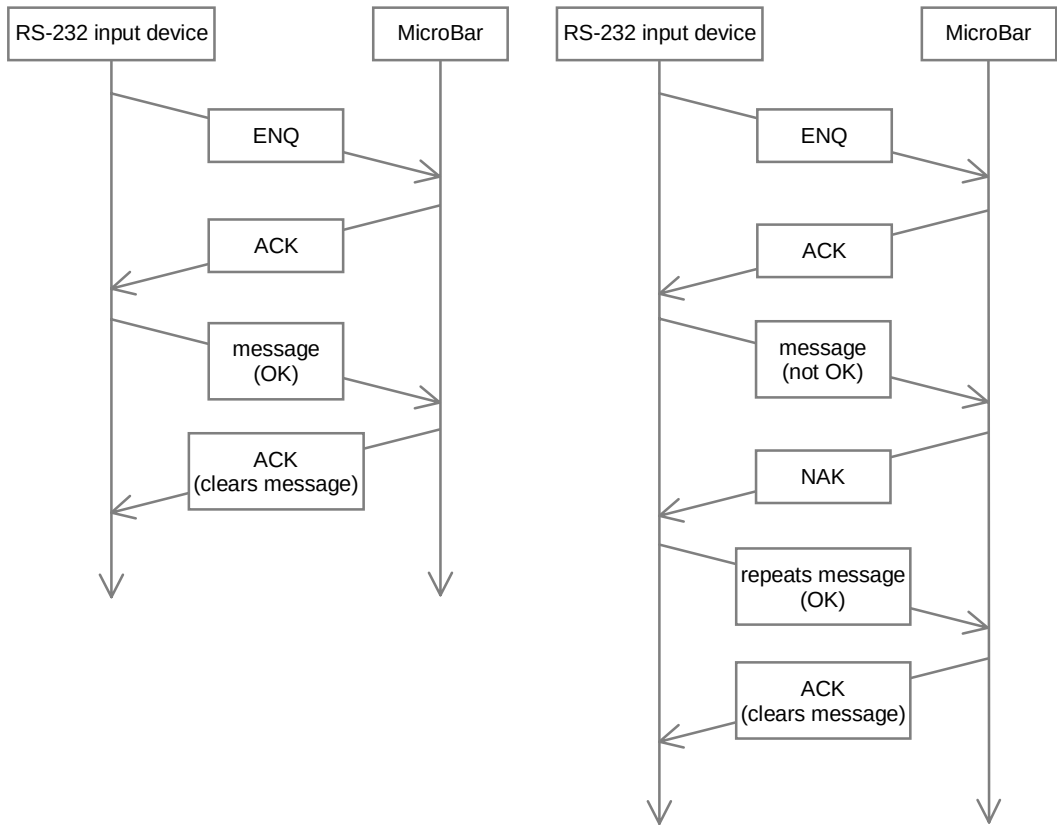
not active



active (NAK 15h) (*)



Typical ENQ / ACK / NAK scenarios



XON / XOFF software protocol

- XON / XOFF allows the MicroBar to control the flow of data from the RS-232 input device
- the MicroBar transmits XOFF (ASCII character DC3) to interrupt the reception of data
- the MicroBar transmits XON (ASCII character DC1) to restart the data flow
- activating XON / XOFF deactivates the RTS / CTS and ENQ / ACK / NAK protocols

not active (*)



active



RTS/CTS hardware protocol

- when RTS / CTS is activated, the MicroBar waits for an RTS from the RS-232 input device and sends a CTS when it is ready to receive data

not active (*)



active



transmission to the host system

transmit on end-of-message character

- the RS-232 input device transmits data message by message
- by default, the MicroBar transfers the data to an input buffer (250 characters maximum) and sends the data to the host system when it receives an end-of-message character

active (*)



\4A\46\10F\04\00\60

transmit on Carriage Return (0Dh) (*)



\4A\46\10\04\3E\0D\60

transmit on Carriage Return + Line Feed



\4A\46\10\04\04\02\0D\60

compose end-of-message character:



\4A\46\10\04\60

- enter an ASCII character and scan End Selection (Appendix A)

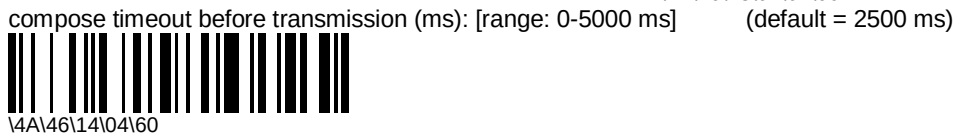
transmit character by character

- the MicroBar transmits each character to the host system as it is received (the characters do not go through the input buffer)
- activate the RTS/CTS protocol with the input device (recommended)



transmit on timeout

- the MicroBar transmits the buffered message to the host system after the specified time (if the RS-232 input device does not send an end-of-message character)
- only applies if transmit on end-of-message character is active



enter a number string and scan End Selection
(Appendix C)

ASCII control code transmission filter

- the RS-232 input device may send control characters that you do not want the MicroBar to transmit to the host system (preambles and end-of-message characters for example)
- the ASCII control code transmission filter removes any ASCII control characters 00h-20h received from the RS-232 input device before the data is transmitted to the host system

transmit all ASCII characters (*)



do not transmit ASCII characters 00h-20h



special keys interpretation (output to keyboard wedge)

- generation of special keyboard functions not available with RS-232 :
 - single characters with ASCII values greater than 7Fh interpreted as special keyboard functions (81h = <Enter>, 88h = <Tab>, etc.)
 - dual-character combinations interpreted as special keyboard functions (" . B " = <Enter>, " . I " = <Tab>, etc.)
- the commands are mutually exclusive
- only for keyboard wedge applications !

not active (*)



original message string transmitted without
single-character / dual-character special keys
interpretation

Example (single character 88h = <Tab>)

original message:	transmitted as:
88	88
msg88xx	msg88xx

Example (". l" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg . lxx
. l	. l
x - . l	x - . l
msg - . lxx	msg - . lxx

single character ASCII value > 7Fh

- interpreted as special keyboard function if encountered in the original message
- the RS-232 input device must have " data bits = 8 " if you want to use this option !



Example (single character 88h = <Tab>)

original message:	transmitted as:
88	<Tab>
msg88xx	msg<Tab>xx

single character ASCII value > 7Fh

emulated key	character > 7Fh	emulated key	character > 7Fh
DEL	80h	PF1	98h
ENTER	81h	PF2	99h
RETURN	82h	PF3	9Ah
SEND	83h	PF4	9Bh
FIELD +	84h	PF5	9Ch
FIELD EXIT	85h	PF6	9Dh
HOME	86h	PF7	9Eh
END	87h	PF8	9Fh
TAB	88h	PF9	A0h
ALT	89h	PF10	A1h
BACKTAB	8Ah	PF11	A2h
BACKSPACE	8Bh	PF12	A3h
right arrow	8Ch	PF13	A4h
left arrow	8Dh	PF14	A5h
up arrow	8Eh	PF15	A6h
down arrow	8Fh	PF16	A7h
CLEAR	90h	PF17	A8h
FIELD -	91h	PF18	A9h
DUP	92h	PF19	AAh
ESC	93h	PF20	ABh
LINE FEED	94h	PF21	ACH
RESET	95h	PF22	ADh
CTRL	96h	PF23	A Eh
PF24	AFh		

dual-character combinations

- interpreted as special keyboard functions (" . B " = <Enter>, " . I " = <Tab>, etc.) if encountered in the original message

emulated key	characters	emulated key	characters
DEL	.A	PF1	0A
ENTER	.B	PF2	0B
RETURN	.C	PF3	0C
SEND	.D	PF4	0D
FIELD +	.E	PF5	0E
FIELD EXIT	.F	PF6	0F
HOME	.G	PF7	0G
END	.H	PF8	0H
TAB	.I	PF9	0I
ALT	.J	PF10	0J
BACK TAB	.K	PF11	0K
BACKSPACE	.L	PF12	0L
right arrow	.M	PF13	0M
left arrow	.N	PF14	0N
up arrow	.O	PF15	0O
down arrow	.P	PF16	0P
CLEAR	.Q	PF17	0Q
FIELD -	.R	PF18	0R
DUP	.S	PF19	0S
ESC	.T	PF20	0T
LINE FEED	.U	PF21	0U
RESET	.V	PF22	0V
CTRL	.W	PF23	0W
PF24	0X		



dual-character values in original message string
always interpreted as special keys

Example (". l" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg<Tab>xx
. l	<Tab>



special dual-character combinations interpreted if
only 2 characters in the original message

Example (". l" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg . lxx
. l	<Tab>



special dual-character combinations interpreted if
there is a hyphen in front of the 2 characters

Example (". l" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg . lxx
x - . l	x <Tab>
msg - . lxx	ms<Tab>xx

UDI / CICS

- special keyboard wedge applications
- the MicroBar converts ASCII character FF (0Ch) to wedge character <Enter> and ASCII character DC1 (11h) to wedge character <Tab>

not active (*)



\4A\46\17\04\00\60

active



\4A\46\17\04\01\60

RS-232 input devices on port 5

- for data received from RS-232 TTL input devices (electronic scales, barcode readers, magstripe readers, . . .) connected to Port 5 of the MicroBar

enable



\4A\46\02\05\01\60

enable communication with RS-232 TTL input devices on port 5

disable (*)



\4A\46\02\05\00\60

baud rate

300



\4A\46\03\05\00\60

600



\4A\46\03\05\01\60

1200



\4A\46\03\05\02\60

2400



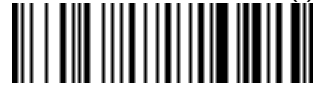
\4A\46\03\05\03\60

4800



\4A\46\03\05\04\60

9600 (*)



\4A\46\03\05\05\60

19200



\4A\46\03\05\06\60

data bits

seven (*)



\4A\46\04\05\00\60

eight



\4A\46\04\05\01\60

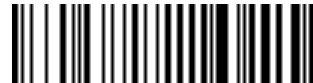
parity

even (*)



\4A\46\05\05\01\60

odd



\4A\46\05\05\02\60

mark



\4A\46\05\05\03\60

space



\4A\46\05\05\04\60

none



\4A\46\05\05\00\60

stop bits

one (*)



\4A\46\06\05\01\60

two



\4A\46\06\05\02\60

ENQ / ACK / NAK

- the RS-232 input device can synchronize communication with the MicroBar :
 1. the RS-232 input device sends an ENQ character (05h) to the MicroBar
 2. the MicroBar answers with an ACK character (06h), even if ENQ / ACK / NAK has not been activated in the MicroBar
 3. the RS-232 input device sends a message to the MicroBar (the message is terminated with the selected end-of-message character)
 4. the MicroBar sends an ACK character to the RS-232 input device at the end of transmission if the message is understood, or a NAK character (15h) if the message is not understood

- there is no hardware/software protocols timeout for RS-232 input devices

ENQ

not active (*)



active (ENQ 05h)



ACK

not active



active (ACK 06h) (*)



NAK

not active



active (NAK 15h) (*)



XON / XOFF software protocol

- XON / XOFF allows the MicroBar to control the flow of data from the RS-232 input device
- the MicroBar transmits XOFF (ASCII character DC3) to interrupt the reception of data
- the MicroBar transmits XON (ASCII character DC1) to restart the data flow
- activating XON / XOFF deactivates the RTS / CTS and ENQ / ACK / NAK protocols

not active (*)



active



RTS/CTS hardware protocol

- when RTS / CTS is activated, the MicroBar waits for an RTS from the RS-232 input device and sends a CTS when it is ready to receive data

not active (*)



active



transmission to the host system

transmit on end-of-message character

- the RS-232 input device transmits data message by message
- by default, the MicroBar transfers the data to an input buffer (250 characters maximum) and sends the data to the host system when it receives an end-of-message character

active (*)



\4A\46\10F\05\00\60

transmit on Carriage Return (0Dh) (*)



\4A\46\10\05\3E\0D\60

transmit on Carriage Return + Line Feed



\4A\46\10\05\04\02\0D\60

compose end-of-message character:



\4A\46\10\05\60

- enter an ASCII character and scan End Selection (Appendix A)

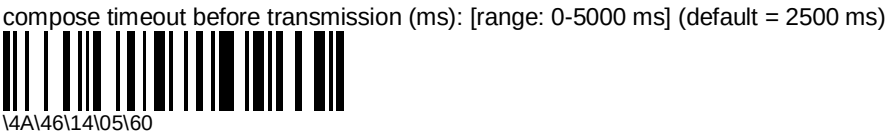
transmit character by character

- the MicroBar transmits each character to the host system as it is received (the characters do not go through the input buffer)
 - activate the RTS/CTS protocol with the input device (recommended)



transmit on timeout

- the MicroBar transmits the buffered message to the host system after the specified time (if the RS-232 input device does not send an end-of-message character)
 - only applies if transmit on end-of-message character is active



enter a number string and scan End Selection
(Appendix C)

ASCII control code transmission filter

- the RS-232 input device may send control characters that you do not want the MicroBar to transmit to the host system (preambles and end-of-message characters for example)
- the ASCII control code transmission filter removes any ASCII control characters 00h-20h received from the RS-232 input device before the data is transmitted to the host system

transmit all ASCII characters (*)



do not transmit ASCII characters 00h-20h



special keys interpretation (output to keyboard wedge)

- generation of special keyboard functions not available with RS-232 :
 - single characters with ASCII values greater than 7Fh interpreted as special keyboard functions (81h = <Enter>, 88h = <Tab>, etc.)
 - dual-character combinations interpreted as special keyboard functions (" . B " = <Enter>, " . I " = <Tab>, etc.)
- the commands are mutually exclusive
- only for keyboard wedge applications !

not active (*)



original message string transmitted without
single-character / dual-character special keys
interpretation

6. Input devices RS-232 input devices on port 5

Example (single character 88h = <Tab>)

original message:	transmitted as:
88	88
msg88xx	msg88xx

Example (". l" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg . lxx
. l	. l
x - . l	x - . l
msg - . lxx	msg - . lxx

single character with ASCII value > 7Fh

- interpreted as special keyboard function if encountered in the original message
 - the RS-232 input device must have " data bits = 8 " if you want to use this option !
 - see the Installation Manual on the cd-rom for the table of ASCII values > 7Fh



Example (single character 88h = <Tab>)

original message:	transmitted as:
88	<Tab>
msg88xx	msg<Tab>xx

single character with ASCII value > 7Fh

emulated key	character > 7Fh	emulated key	character > 7Fh
DEL	80h	PF1	98h
ENTER	81h	PF2	99h
RETURN	82h	PF3	9Ah
SEND	83h	PF4	9Bh
FIELD +	84h	PF5	9Ch
FIELD EXIT	85h	PF6	9Dh
HOME	86h	PF7	9Eh
END	87h	PF8	9Fh
TAB	88h	PF9	A0h
ALT	89h	PF10	A1h
BACKTAB	8Ah	PF11	A2h
BACKSPACE	8Bh	PF12	A3h
right arrow	8Ch	PF13	A4h
left arrow	8Dh	PF14	A5h
up arrow	8Eh	PF15	A6h
down arrow	8Fh	PF16	A7h
CLEAR	90h	PF17	A8h
FIELD -	91h	PF18	A9h
DUP	92h	PF19	AAh
ESC	93h	PF20	ABh
LINE FEED	94h	PF21	ACH
RESET	95h	PF22	ADh
CTRL	96h	PF23	Aeh
PF24	AFh		

dual-character combinations

- interpreted as special keyboard functions (" . B " = <Enter>, " . I " = <Tab>, etc.) if encountered in the original message

emulated key	characters	emulated key	characters
DEL	.A	PF1	0A
ENTER	.B	PF2	0B
RETURN	.C	PF3	0C
SEND	.D	PF4	0D
FIELD +	.E	PF5	0E
FIELD EXIT	.F	PF6	0F
HOME	.G	PF7	0G
END	.H	PF8	0H
TAB	.I	PF9	0I
ALT	.J	PF10	0J
BACK TAB	.K	PF11	0K
BACKSPACE	.L	PF12	0L
right arrow	.M	PF13	0M
left arrow	.N	PF14	0N
up arrow	.O	PF15	0O
down arrow	.P	PF16	0P
CLEAR	.Q	PF17	0Q
FIELD -	.R	PF18	0R
DUP	.S	PF19	0S
ESC	.T	PF20	0T
LINE FEED	.U	PF21	0U
RESET	.V	PF22	0V
CTRL	.W	PF23	0W
PF24	0X		

in all messages



\4A\46\16\05\01\60

dual-character values in original message string
always interpreted as special keys

Example (". I" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg<Tab>xx
. I	<Tab>

in 2-character messages



\4A\46\16\05\03\60

special dual-character combinations interpreted if
only 2 characters in the original message

Example (". I" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg . lxx
. I	<Tab>

preceded by a hyphen (-)



\4A\46\16\05\02\60

special dual-character combinations interpreted if
there is a hyphen in front of the 2 characters

Example (". I" dual-character combination = <Tab>)

original message:	transmitted as:
msg . lxx	msg . lxx
x - . I	x <Tab>
msg - . lxx	ms<Tab>xx

UDI / CICS

- special keyboard wedge applications
- the MicroBar converts ASCII character FF (0Ch) to wedge character <Enter> and ASCII character DC1 (11h) to wedge character <Tab>

not active (*)



\4A\46\17\05\00\60

active



\4A\46\17\05\01\60

7. Operating settings

- settings that affect the way your product operates (beeps / green indicator LED)

beeps / green indicator LED

- power-up beeps
 - 2 beeps = successful power-up
 - 3 long beeps + 5 short beeps = flash EEPROM integrity error (contact your Intermec representative !)

volume



note (tone frequency)



compose frequency (Hz): [range: 100-4000 Hz]



- enter a number string and scan End Selection (Appendix C)

good read beeps

number

- "normal" bar codes: 1 beep (default) = good read
- configuration codes: 2 beeps = good read, 6 beeps = setup error there may be a slight delay between beeps (< 1 second) for configuration codes (time required to update the flash EEPROM)



duration

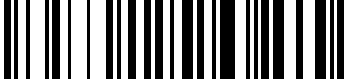


200 ms



\45\49\03\08\60

80 ms (*)



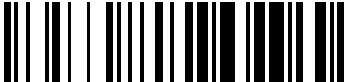
\45\49\01\10\60

compose (ms): [range: 0-999 ms]



\45\49\60

300 ms



\45\49\04\12C\60

enter a number string and scan End Selection
(Appendix C)

timing

before transmission (*)



\45\52\60

after transmission



\45\51\60

good read LED

on (*)

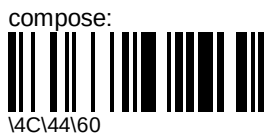


\45\4F\01\10\60



external beep command

- only for RS-232 host applications
- useful to indicate successful or unsuccessful processing by your application
- when it receives the specified beep command from the RS-232 host system, the MicroBar 9730 will :
 - emit a beep (internal MicroBar beeper)
 - send a square signal (identical in frequency and duration to the MicroBar beep) to pin 3 of port 4 to control an external beeper



enter an ASCII character and scan End Selection
(Appendix A)

8. Configuration modes and utilities

get version info (EPROM software / Smart cable software)



version beeps / display on terminal screen :

- EPROM software version = "V" < N - N N > "C2" [L]
- Smart cable version (if applicable) = < N L N N > "SC" [L] (the first "L" character is represented by green LED flashes)

beeps only for EPROM software version and Smart cable software (if applicable)

Examples

EPROM standard software version :

- screen display : * MicroBar 9730 * V02-01C2_ *
- EPROM version beeps :

beeps	< 3 long >	< 1 long >	< 2 long >
interpretation:	N + 1	N + 1	N + 1
EPROM version:	2-	0	1

EPROM special software version ("B" for example) :

- screen display : * MicroBar 9730 * V02-01C2B *
- EPROM version beeps :

beeps:	< 3 long >	< 1 long >	< 2 long >	< 2 short >
interpretation:	N + 1	N + 1	N + 1	L
EPROM version:	2-	0	1	B

Smart cable standard software version (if applicable) :

- screen display : * MicroBar 9730 * V02-01C2B * 1A03SC_ *
- EPROM software version beeps as above, then series of low beeps / flashes for Smart cable standard software version :

8. Configuration modes and utilities

low beeps / flashes :	< 2 long >	< 1 flash >	< 1 long >	< 4 short >
interpretation:	N + 1	L	N + 1	N + 1
cable s/w version:	1	A	0	3

Smart cable special software version ("B" for example) :

- screen display : * MicroBar 9730 * V02-01C2B * 1A03SCB *
- EPROM / Smart cable beeps / flashes as above + extra short beeps at the end to indicate special version (2 beeps for "B")

flash memory upgrade with RS-232 cable 0-234032-00



- necessary for online upgrade of the MicroBar flash memory if you use RS-232 cable 0-234032-00 !
- do not send this command directly from EasySet to the MicroBar - you must read this command as a configuration code (send to the setup sheet and print out) !!!
- see section 1 "Using EasySet" for details on how to connect up for flash memory upgrade with RS-232 cable 0-234032-00

ScanPlus / MaxiScan transparent configuration mode



allows you to use a ScanPlus or MaxiScan 2200 barcode reader to set up your MicroBar product (setup commands are transmitted to the MicroBar but do not affect the ScanPlus / MaxiScan 2200)

1. activate the same symbologies and code lengths for the ScanPlus / MaxiScan 2200 and the MicroBar (disconnect the ScanPlus / MaxiScan 2200 to make sure that "ScanPlus / MaxiScan transparent configuration mode" is disabled before you configure those products !)
2. use your ScanPlus / MaxiScan 2200 to read the transparent mode configuration code from the setup sheet
3. use your ScanPlus / MaxiScan 2200 to set up the MicroBar
4. switch off the ScanPlus / MaxiScan 2200 to deactivate transparent configuration mode at the end of MicroBar setup

display data string mode



- displays data string and checksum values on a terminal screen when you read configuration bar codes (the EasySet "View - Data String" command shows the data string but does not show the checksum)
- hexadecimal values separated by backslashes or other characters (depending on your terminal emulation setup)
- remains active until you switch off the product

temporary configuration mode

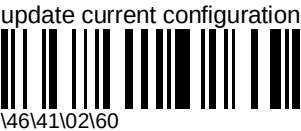
- test new setup without losing current setup



subsequent configuration actions are applied temporarily but will only be permanently saved if "update current configuration" is selected



- ignores any temporary configuration actions and quits temporary configuration mode



permanently saves any temporary configuration actions and quits temporary configuration mode

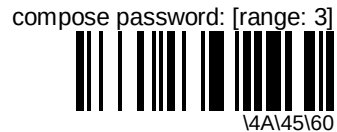
configuration password mode

- protects the MicroBar against unauthorized configuration
- password active : you must read the password in a Code 39 bar code before you set up, otherwise your setup will not be saved (make sure that the MicroBar and your barcode reader are enabled to read Code 39 !!!)



to deactivate the password:

1. read the Code 39 password code
2. enter "not active"



- enter an ASCII string and scan End Selection (Appendix A)
- password format = * char1 char2 char3 \$ *
- enter the values for "char1" "char2" "char3"

A. ASCII character codes

NUL (00h)



STX (02h)



EOT (04h)



ACK (06h)



SOH (01h)



ETX (03h)



ENQ (05h)



BEL (07h)



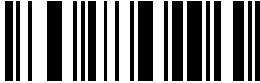
end selection



A. ASCII character codes

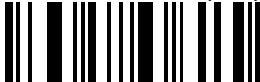
BS (08h)

\3E\08\60

HT or TAB (09h)

\3E\09\60

LF (0Ah)

\3E\0A\60

VT (0Bh)

\3E\0B\60

FF (0Ch)

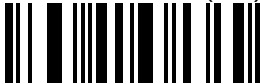
\3E\0C\60

CR (0Dh)

\3E\0D\60

SO (0Eh)

\3E\0E\60

SI (0Fh)

\3E\0F\60

end selection

\64\60

DLE (10h)

 \3E\10\60

DC1 (11h)

 \3E\11\60

DC2 (12h)

 \3E\12\60

DC3 (13h)

 \3E\13\60

DC4 (14h)

 \3E\14\60

NAK (15h)

 \3E\15\60

SYN (16h)

 \3E\16\60

ETB (17h)

 \3E\17\60

end selection

 \64\60

A. ASCII character codes

CAN (18h)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\18\\60

EM (19h)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\19\\60

SUB (1Ah)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\1A\\60

ESC (1Bh)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\1B\\60

FS (1Ch)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\1C\\60

GS (1Dh)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\1D\\60

RS (1Eh)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\1E\\60

US (1Fh)
A barcode consisting of vertical black bars of varying widths on a white background.
\\3E\\1F\\60

end selection
A barcode consisting of vertical black bars of varying widths on a white background.
\\64\\60

SP (20h)



\00\60

! (21h)



\01\60

" (22h)



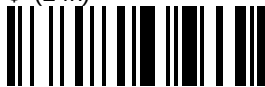
\02\60

(23h)



\03\60

\$ (24h)



\04\04\60

% (25h)



\05\60

& (26h)



\06\60

' (27h)



\07\60

end selection



\64\60

A. ASCII character codes

((28h)

\08\60

) (29h)

\09\60

* (2Ah)

\0A\60

+ (2Bh)

\0B\60

, (2Ch)

\0C\60

- (2Dh)

\0D\60

. (2Eh)

\0E\60

/ (2Fh)

\0F\0F\60

end selection

\64\60

0 (30h)



\10\60

1 (31h)



\11\60

2 (32h)



\12\60

3 (33h)



\13\60

4 (34h)



\14\60

5 (35h)



\15\60

6 (36h)



\16\60

7 (37h)



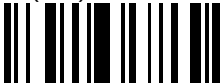
\17\60

end selection



\64\60

A. ASCII character codes

8 (38h)

\18\60

9 (39h)

\19\60

: (3Ah)

\1A\60

; (3Bh)

\1B\60

< (3Ch)

\1C\60

= (3Dh)

\1D\60

> (3Eh)

\1E\60

? (3Fh)

\1F\60

end selection

\64\60

@ (40h)

 \20\60

A (41h)

 \21\60

B (42h)

 \22\60

C (43h)

 \23\60

D (44h)

 \24\60

E (45h)

 \25\60

F (46h)

 \26\60

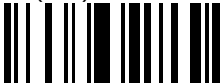
G (47h)

 \27\60

end selection

 \64\60

A. ASCII character codes

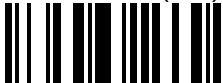
H (48h)

\28\60

I (49h)

\29\60

J (4Ah)

\2A\60

K (4Bh)

\2B\60

L (4Ch)

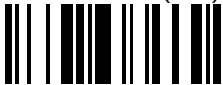
\2C\60

M (4Dh)

\2D\60

N (4Eh)

\2E\60

O (4Fh)

\2F\60

end selection

\64\60

P (50h)



\30\60

Q (51h)



\31\60

R (52h)



\32\60

S (53h)



\33\60

T (54h)



\34\60

U (55h)



\35\60

V (56h)



\36\60

W (57h)



\37\60

end selection



\64\60

A. ASCII character codes

X (58h)

\38\60

Y (59h)

\39\60

Z (5Ah)

\3A\60

[(5Bh)

\3B\60

\ (5Ch)

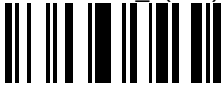
\3C\60

] (5Dh)

\3D\60

^ (5Eh)

\3E\3E\60

~ (5Fh)

\3F\60

end selection

\64\60

` (60h)

this character not
available

b (62h)



d (64h)



f (66h)



a (61h)



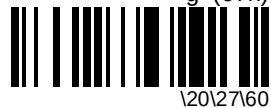
c (63h)



e (65h)



g (67h)



end selection



A. ASCII character codes

h (68h)

\20\28\60

i (69h)

\20\29\60

j (6Ah)

\20\2A\60

k (6Bh)

\20\2B\60

l (6Ch)

\20\2C\60

m (6Dh)

\20\2D\60

n (6Eh)

\20\2E\60

o (6Fh)

\20\2F\60

end selection

\64\60

p (70h)



\20\30\60

q (71h)



\20\31\60

r (72h)



\20\32\60

s (73h)



\20\33\60

t (74h)



\20\34\60

u (75h)



\20\35\60

v (76h)



\20\36\60

w (77h)



\20\37\60

end selection



\64\60

A. ASCII character codes

x (78h)

\20\38\60

y (79h)

\20\39\60

z (7Ah)

\20\3A\60

{ (7Bh)

\20\3B\60

| (7Ch)

\20\3C\60

} (7Dh)

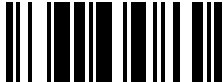
\20\3D\60

~ (7Eh)

\20\3E\60

DEL (7Fh)

\20\3F\60

end selection

\64\60

B. Additional preamble / postamble characters—Keyboard wedge

PF 1



\04\02\18\60

PF 2



\04\02\19\60

PF 3



\04\02\1A\60

PF 4



\04\02\1B\60

PF 5



\04\02\1C\60

PF 6



\04\02\1D\60

PF 7



\04\02\1E\60

end selection



\64\60

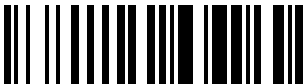
B. Additional preamble / postamble characters—Keyboard wedge

PF 8



\04\02\1F\60

PF 9



\04\02\20\60

PF 10



\04\02\21\60

PF 11



\04\02\22\60

PF 12



\04\02\23\60

PF 13



\04\02\24\60

PF 14



\04\02\25\60

PF 15

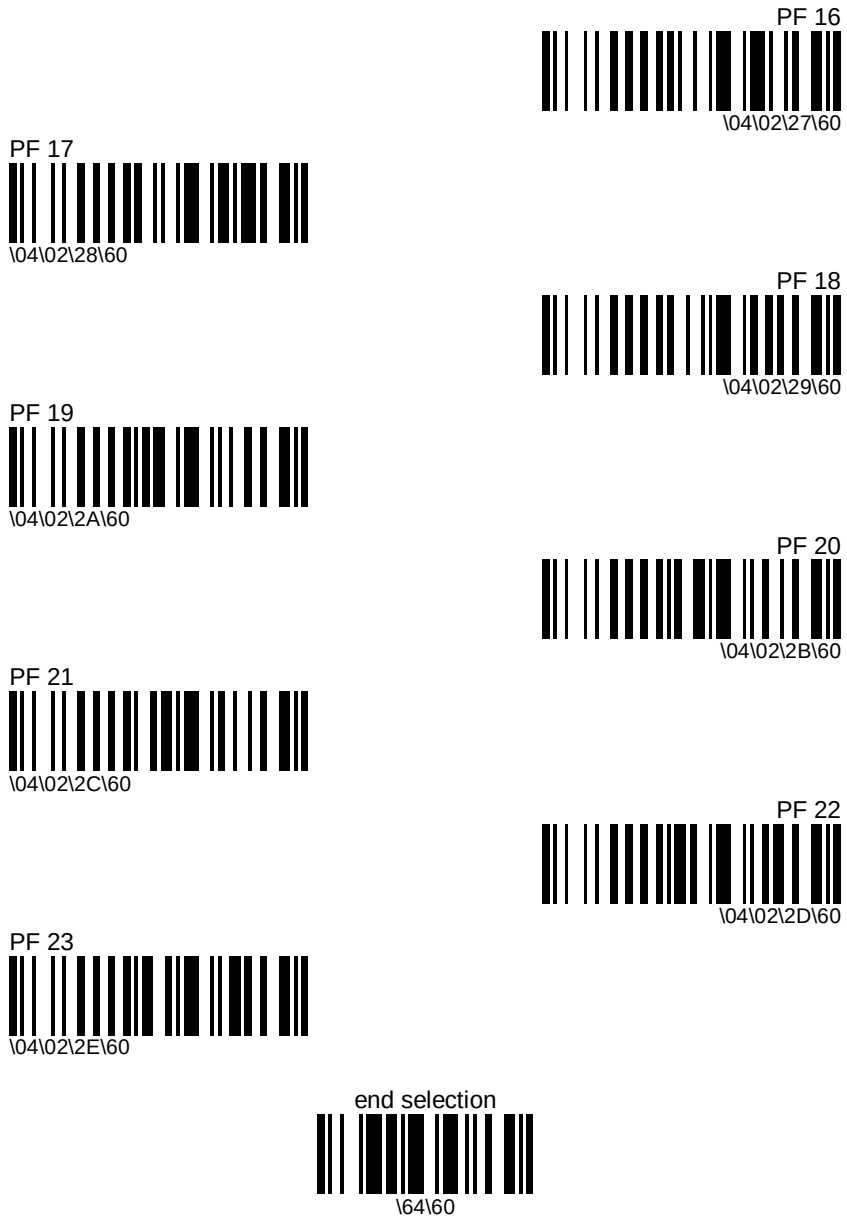


\04\02\26\60

end selection



\64\60



B. Additional preamble / postamble characters—Keyboard wedge

DEL

\04\02\00\60

RETURN

\04\02\02\60

FIELD +

\04\02\04\60

HOME

\04\02\06\60

end selection

\64\60

PF 24

\04\02\2F\60

ENTER

\04\02\01\60

SEND

\04\02\03\60

FIELD EXIT

\04\02\05\60

TAB = Ctrl i



\04\02\08\60

BACK TAB



\04\02\0A\60

ARROW RIGHT



\04\02\0C\60

ARROW UP



\04\02\0E\60

END



\04\02\07\60

ALT



\04\02\09\60

BACK SPACE



\04\02\0B\60

ARROW LEFT



\04\02\0D\60

end selection



\64\60

B. Additional preamble / postamble characters—Keyboard wedge

CLEAR



\04\02\10\60

DUP



\04\02\12\60

LINE FEED



\04\02\14\60

ARROW DOWN



\04\02\0F\60

FIELD -



\04\02\11\60

ESC



\04\02\13\60

RESET



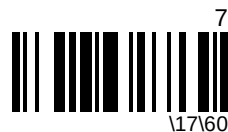
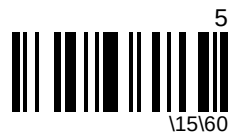
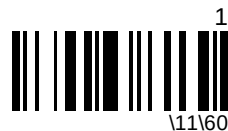
\04\02\15\60

end selection



\64\60

C. Number codes





D. Test codes

Codabar



123456

Code 39



CODE-39

Code 93



CODE-93

Code 128



CODE-128

EAN 128



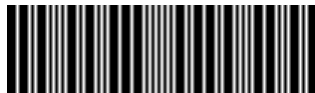
]C1EAN 128

Interleaved 2 of 5



12345678901234

Standard 2 of 5



123456

Matrix 2 of 5



012345

D. Test codes

MSI Code



12345666

Plessey Code



80001495050

EAN-8



12345670

EAN-13



1234567890128

UPC-A



0 01234 50000 7

UPC-E



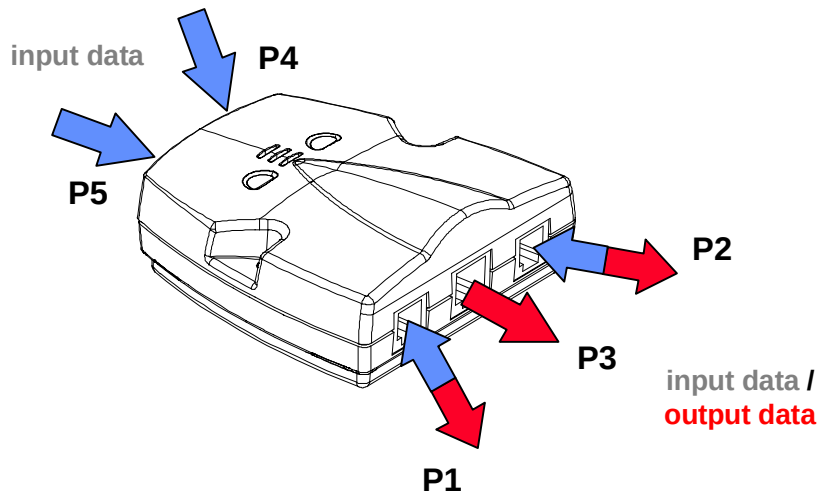
0 012345 7

E. Technical data

Input/output ports

Overview

- port 4 and port 5 are input ports
- port 5 is Intermec input device compatible



- ports 1, 2 and 3 are output ports
- ports 1, and 2 can also be used as input ports when free (depending on output interface)
- all ports are TTL level (port 4 also accepts RS-232 C level)
- ports 1 and 2 accept cables used with the MicroBar NX and cable management modules (CMM)
- all wedge connections can be made through port 3 with ScanPlus 1800 compatible cables
- DEC VT 510, 520 can be connected as standard wedge
- some Wyse terminals are PC-compatible and are considered as standard PC
- laser and wand emulation outputs are not available as a standard options

Input device priority management

- input devices are classified in two priority groups :
 - high priority : RS-232 inputs on ports 4, 5
 - low priority : wand, laser, magstripe inputs on ports 1, 3, 4, 5
- priority depends on the type of input device (RS-232, wand, laser, magstripe), not on the port

simultaneous event priority levels		priority management
high	high	• 1st event processed first (top priority) • other event processed or lost depending on speed and protocol
high	low	• high-priority event processed correctly • low-priority event will be lost
low	low	• 1st event processed first (top priority) • second event will be lost

Output to host system

port	output interfaces	cable	comments
1 & 2	wedge (standard)	single passive cable set	Cxx compatible (601xxx-xx)
3	wedge Wyse / HP	Y smart	ScanPlus 1800 ST compatible
3	wedge DEC VT 220,320,420 / RS based	Y active	ScanPlus 1800 ST compatible
3	dual RS-232C	Y active	ScanPlus 1800 ST compatible
3	RS-232 C	single active	ScanPlus 1800 ST compatible
3	RS-232 TTL	single passive	ScanPlus 1800 ST compatible
3	cash registers (OCIA, IBM)	single smart	ScanPlus 1800 ST compatible

I/O port functions

To avoid power supply conflicts, do not connect more than one input power supply at a time (the 5V power supply pins on all ports are interconnected).

port	direction	connector	devices	comments
1	output	6-pin female modular plug, unscreened	to system for standard wedge (except Wyse & DEC 220/320/420)	passive cable - UBI 601xxx compatible
	input		5V power supply	old UBI power supply cable end style (available if not already used for standard wedge transmission)
	input		external analog/digital pen	UBI wand compatible (available if not already used for standard wedge transmission)
2	output	6-pin female modular plug (no key), unscreened	to keyboard for standard wedge (except Wyse & DEC 220/320/420)	passive cable - UBI 601xxx compatible
	input		5V power supply	old UBI power supply cable end style (available if not already used for standard wedge transmission)
3	output	10-pin female modular plug, unscreened	host system output cable single passive type for RS-232 TTL single active type for RS-232 C single smart type for all cash registers & USB (when available) Y active type for dual RS-232 C Y smart type for special wedges (Wyse, DEC, ...)	ScanPlus 1800 fully compatible cable type (without boot but adapted overmold)
	input		1 analog/digital pen	specific wand pin-out (available only if standard wedge transmission on ports 1 and 2)
4	input	9-pin subD female, unscreened	undecoded laser gun CCD scanner in wand emulation	PSC/Symbol compatible

E. Technical data Input/output ports

port	direction	connector	devices	comments
	input		analog/digital pen	WA/DL compatible
			2 analog/digital pens	using the UBI dual pen adaptor cable option
	input		RS-232 C input	electronic scales or any other RS-232 C input device
5	input	10-pin modular female (LH key), unscreened	Intermec wands undecoded laser gun CCD scanner in wand emulation analog/digital pen undecoded magnetic stripe reader dual tracks RS-232 TTL input	Intermec input device compatible including left hand key connectors

I/O port pinouts

Port 1

pin	pen	wedge system	power supply
1	Data	Clock	
2		Data	
3			
4			
5	VDC	VDC	Power +5 VDC (I/O)
6	GND	GND	Power GND

Port 2

pin	wedge keyboard	power supply
1	Clock	
2		
3		
4		
5	VDC	Power +5 VDC (I/O)
6	GND	Power GND

Port 3

pin	pen	RS-232 / dual RS-232	wedge RS (DEC, ...)	standard wedge	smart cables
1	+5V output	+5V (I/O)	+5V (I/O)	+5V (I/O)	+5V (I/O)
2					
3				keyboard active	reset
4		TXD (MicroBar 9730 output)	TXD (MicroBar 9730 output)	data	TXD (MicroBar 9730 output)
5		RTS (MicroBar 9730 output)		clock	RTS (MicroBar 9730 output)
6	data	RXD (MicroBar 9730 input)	RXD (MicroBar 9730 input)		RXD (MicroBar 9730 input)
7		CTS (MicroBar 9730 input)		cable detection (=p8)	CTS (MicroBar 9730 input)
8		cable detection (=+5V)	cable detection (=GND)	cable detection (=p9)	cable detection (=GND)
9		cable detection (=GND)	cable detection (=+5V)	cable detection (=p7)	cable detection (=GND)
10	GND	GND	GND	GND	GND

Port 4

pin	RS-232 C	laser	pen port 4	2nd pen with port 4 adaptor
1	GND	SOS	Data	do not connect
2	GND	Data	do not connect	Data
3	TXD (output)	ACK/beep	do not connect	do not connect
4	CTS (input)	CTRL	do not connect	do not connect
5	GND	trigger	do not connect	do not connect
6	RTS (output)	enable	do not connect	do not connect
7	GND	GND	GND	GND
8	RXD (input)	do not connect	do not connect	do not connect
9	VDC (I/O)	VDC (I/O)	VDC (I/O)	VDC (I/O)

Port 5

pin	laser	pen	RS-232 TTL	magstripe
1	do not connect	do not connect	GND	csi
2	decode	do not connect	GND	data iso2
3	SOS	do not connect	TXD (output)	clk iso2
4	data	data	RXD (input)	clk iso1
5	do not connect	do not connect	GND	tbd
6	beep	do not connect	RTS (output)	data iso1
7	enable (active low)	do not connect	do not connect	do not connect
8	trigger	do not connect	CTS (input)	do not connect
9	GND	GND	GND	GND
10	+5V (I/O)	+5V (I/O)	+5V (I/O)	+5V (I/O)

Index

A

ACK 39-44, 52, 76, 87, 128-131, 142-144, 165, 195
 add-on 58, 110
 AIM 58-59, 82, 96
 Alt mode 33
 Apple / Macintosh 23

B

bar 25, 28, 33, 41, 79, 92, 94, 96, 101, 107, 113, 155, 161, 163
 barcode length 28, 100, 106, 121
 baud rate 37, 126, 140
 beep 40-42, 44, 155, 157, 195-196

C

Carriage Return 75, 77, 132, 145
 cash register 11, 13, 25, 192-193
 check digit 58, 82, 85, 91, 99-100, 103-106, 111-113, 121, 123
 CIP 92, 97, 100
 CLSI 82
 Codabar 58, 60, 81-83, 189
 Code 128 35, 58, 61, 96-97, 189
 Code 39 28-30, 35, 58, 60, 79, 86-87, 92, 94, 163, 189
 Code 93 58, 61, 94, 189
 Computer Identics 105
 control character 31, 86, 134, 147
 CTS 39-42, 44-45, 118, 131, 133, 144, 146, 195-196

D

data bit 10, 38, 127, 135, 141, 148
 data editing 32, 64-65, 124-125
 data string 67, 161
 data validation 99, 121
 DEC VT 15, 21-22, 191-192
 DEC VT / PC 22
 decoding 99
 default 11, 28, 37, 60-63, 79, 86, 93-94, 96-97, 100-101, 105-106, 108, 116-117, 123, 132-133, 145-146, 155
 delay 33-34, 47-48, 155
 disable 79, 126, 140
 display 159-160, 161
 Down Arrow 76
 duration 155, 157

E

EAN 58, 61, 79, 96-97, 109, 111-114, 189-190
 EAN 128 58, 61, 96-97, 189
 EAN-13 61, 109, 112-114, 190
 EAN-8 61, 109, 111, 113-114, 190
 editing 32, 64-65, 124-125
 ENQ 39-44, 52, 76, 87, 128-131, 142-144, 165
 Enter 28, 34, 48, 75, 134, 137, 139, 147, 150, 152
 EPROM 159-160

F

Field Advance 75
 Field Exit 75
 fixed length 68, 97, 100-101, 106-107, 121
 flash memory 160
 FNC1 96-97

Index

French CIP 92, 100
frequency 154-155, 157
full ASCII 28, 35, 86, 94, 96, 108

H

hardware 39-41, 44, 129, 131, 143-144
Hewlett Packard 23
HR 100

I

IBM 20, 28, 192
IBM PC AT 28
Identicon 105
input devices 66, 115, 126, 129, 140, 143, 192
input port 55-56, 64-66, 124, 191
input port identifier 55
inter-character 33, 47
interface number 25
Interleaved 2 of 5 58, 61, 99-101, 189
inter-message 34, 48
Italian CPI 92

K

keyboard 14-16, 19, 28, 32-33, 35, 52, 64-65, 75, 97, 134-135, 137, 139, 147-148, 150, 152, 193-194, 195
keyboard remapping 32, 64-65
keyboard wedge 14-16, 19, 28, 52, 75, 97, 134, 139, 147, 152

L

laser scanner 116, 118-119
library system 82
Line Feed 77, 132, 145

lower case 35
LRC 45, 50, 52-55, 58-59, 75, 77-78, 123

M

magstripe 57, 115, 122, 140, 192, 196
Matrix 2 of 5 58, 62, 102, 189
message length 50, 64-66, 124
minimum length 100-101, 106, 121
mod 10 99, 103, 105
mode 9, 19, 33, 79, 120, 161-162, 163
modulo 43 91
MSI Code 58, 62, 103, 190

N

NAK 39-44, 87, 128-131, 142-144, 167
number system 109, 111-112, 113

P

parity 10, 38, 127, 141
pharmaceutical 92, 97, 100
Plessey Code 58, 62, 104, 190
postamble 50, 52-55, 58-59, 75, 77-78, 181
preamble 50, 52-55, 58-59, 75, 77-78, 181
predefined 13, 19, 24
protocol 40-42, 44-45, 131, 133, 144, 146, 192

R

re-encoding 113
remapping 32, 64-65
reset 11, 32, 64, 66, 71-72, 79, 195
restore 162
RS-232 9-10, 17-18, 24, 37, 39, 53, 56-57, 64, 66, 77, 115, 118, 124, 126, 128-129,

131-135, 140, 142-148, 157, 160, 191-195, 196
RTS 39-42, 44-45, 131, 133, 144, 146, 195-196

S

scale solicitation 64-65, 124-125
scales 115, 124, 126, 140, 194
security 99, 101, 106-107, 121
separator character 97, 123
Smart cable 159-160
software 39-41, 44, 118, 129, 131, 143-144, 159-160
solicitation 64-65, 124-125
space 39, 93-94, 128, 142
special keys interpretation 28, 134, 147
Standard 2 of 5 58, 62, 105, 107, 189
start/stop 81-84, 90, 105
stop bit 39, 128, 142
STX / ETX 54, 78
symbolologies 59, 79, 121, 161
symbology 50, 52-55, 58-60, 64, 66, 75, 77-81, 85-86, 94, 96, 99, 102-105, 108-109, 124
symbology identifier 58-59, 60

T

Tab 28-30, 75, 134-135, 137-139, 147-148, 150-151, 152
terminal 11, 13, 19-25, 44, 93-94, 159, 161
timeout 39-42, 44, 98, 119-120, 129, 133, 143, 146
timing 156
transmission 13, 25, 27, 33, 35, 40-42, 44, 59, 66, 113, 121-122, 124, 128, 132-134, 142, 145-147, 156, 193

U

UPC 58, 63, 79, 109, 111-114, 190
UPC number system 112
UPC-A 63, 109, 111-114, 190
UPC-E 63, 109, 111, 113-114, 190
update 155, 162
upper case 35-36, 86

V

version 159-160
volume 154

W

wand 79, 115, 191-193, 194
Wyse 16, 19, 24, 191-192, 193

X

XON / XOFF 44, 131, 144

