SIEMENS
pocket reader

pocket reader V1: Interface pocket reader ⇔ PC
Functional Specification

REVISION 5b

English Version
Translated by Philip Pemberton [philpem@btinternet.com]

Distribution:

Huber	EZE TNA 2	Lager	EWW NP
König	PSE PRO SRT	Smarda	EWW NP
Szelestey	PSE PRO SRT		
Tschirk	PSE PRO SRT		

Status	Signature / Date	Signature / Date
☐ valid	Dr. Doppler	Huber
☒ provisional, objections to:	Author	Counter signature / Review note

Verfasser		Document		
Name:	:Dr. Doppler	Doc.Nr.:	Z3981404.43.01.05be	Siemens AG Österreich
Department:	:EWW NP	Date :	02-01-28	**EWW NP**
Telephone:	:35767	File :	pr1_i05b_eng.doc	Electronics Plant Vienna
Fax:	56444	Page :	1/19	New Products

CONTENTS

0 General

With the pocket reader text can be entered and stored off-line. The entered text can be transferred over a serial interface to any computer - PC, Mac, Newton, Psion (series 3 or 5), Handheld PC (WindowsCE) or Palm (III or V) - for further processing.

Note:
Italically written sections of this document are relevant only for test purposes. In the product version of the pocket reader they are not supported.

1 Outline

1.1 Hardware: Interface cable, Adapter

The serial interface pocket reader corresponds electrically to the **RS-232** specification. It is formed from 3 lines (RxD, TxD and Gnd).

With the pocket reader a connection cable is provided for the link to an industry standard PC according to the following description. For all other computer platforms appropriate adapters (not contained in the package as shipped) are needed.

1.1.1 PC connection cable

The PC connection cable has a 9-pin Sub-D socket for the link to the PC as well as a 2.5mm jack plug for the link to the pocket reader.

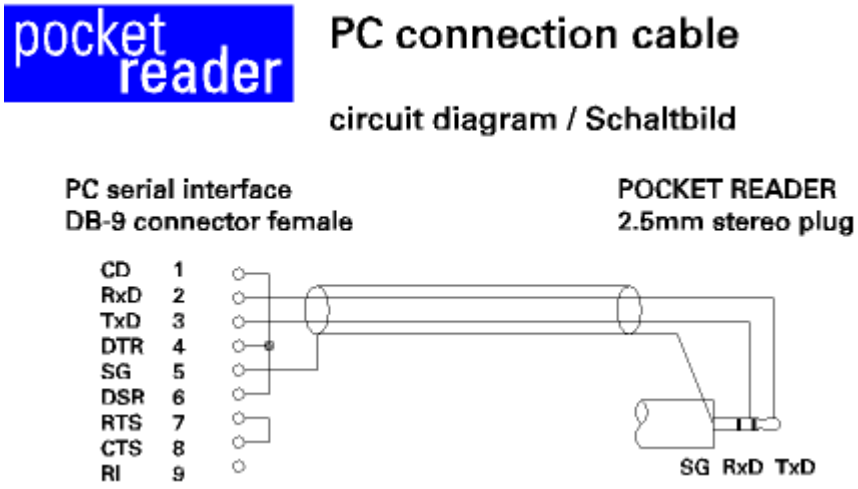

Figure 1: PC cable

1.1.2 Mac connection

The pocket reader can be connected to a serial port (modem, printer) on an Apple Macintosh via a cable illustrated in the following figure:

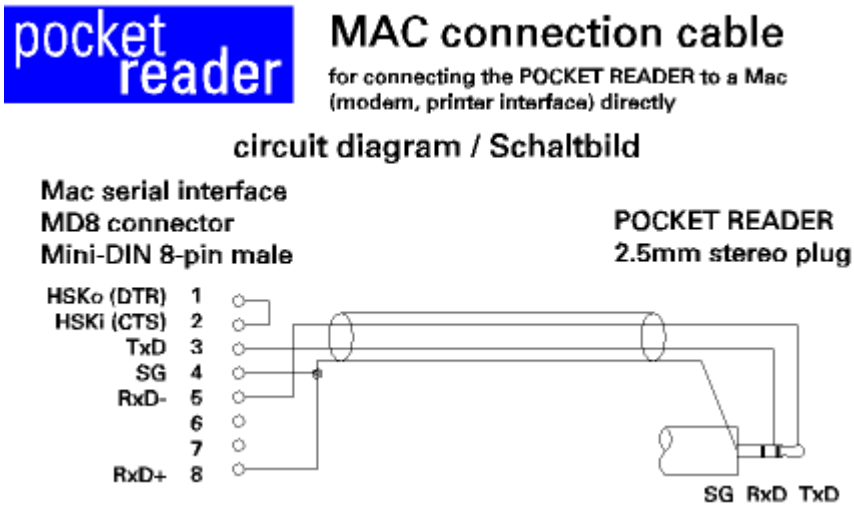

Figure 2: Mac cable

The pocket reader can, with the supplied PC cable (see above) and an adapter, be connected to a serial port (modem, printer) of an Apple Macintosh via the adapter illustrated in the following figure:

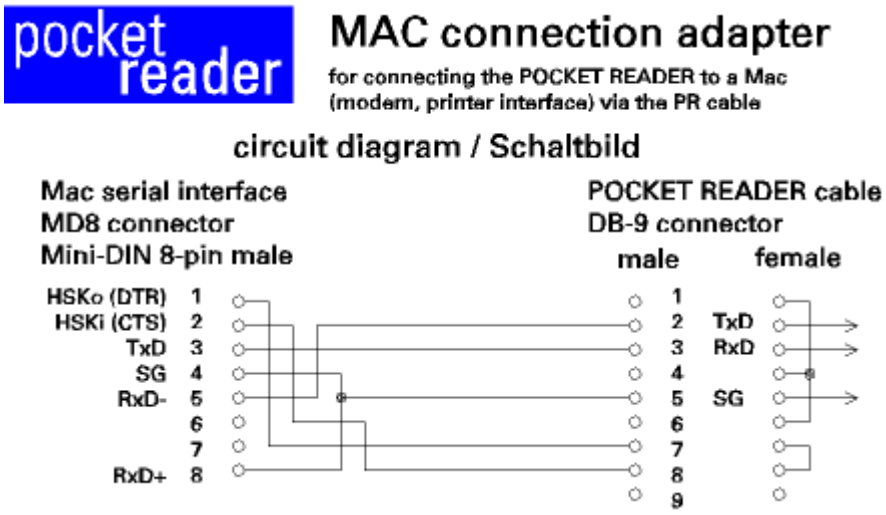

Figure 3: Mac adapter

Note:
For Macs, which have no serial, but only USB ports, one needs a **USB-to-serial converter**. Pay attention that the converter must support low baud rates (down to 300Bd) and supply (emulate) a standard interface, e.g. a modem or a printer interface.

1.1.3 PDA Connection

To connect the pocket reader with any PDA, the PC cable supplied with the respective device and the pocket reader cable can be connected with an adapter illustrated in the following figure:

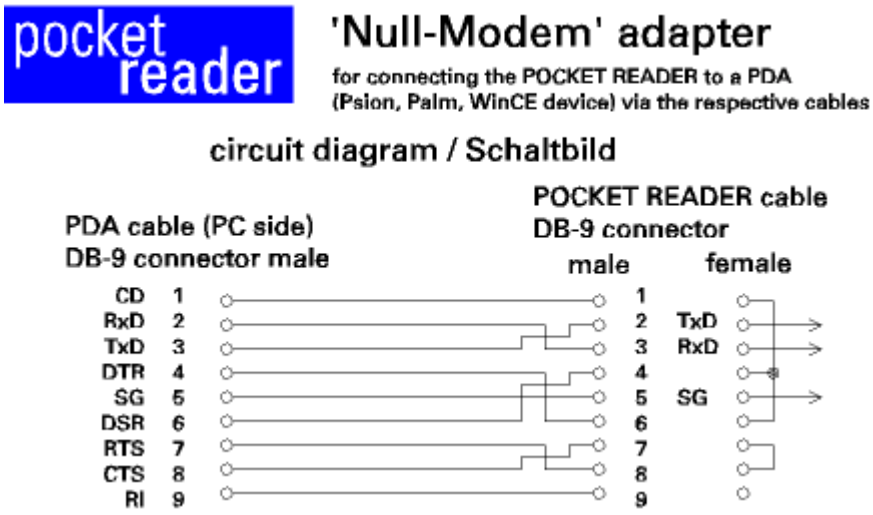

Figure 4: 'Null Modem' Adapter

1.1.4 PSION Connection

The pocket reader can be connected to the PSION Series 3c, 3mx, 5, 5mx, 5mxPro, 7 and Siena serial port with a cable illustrated in the following figure:

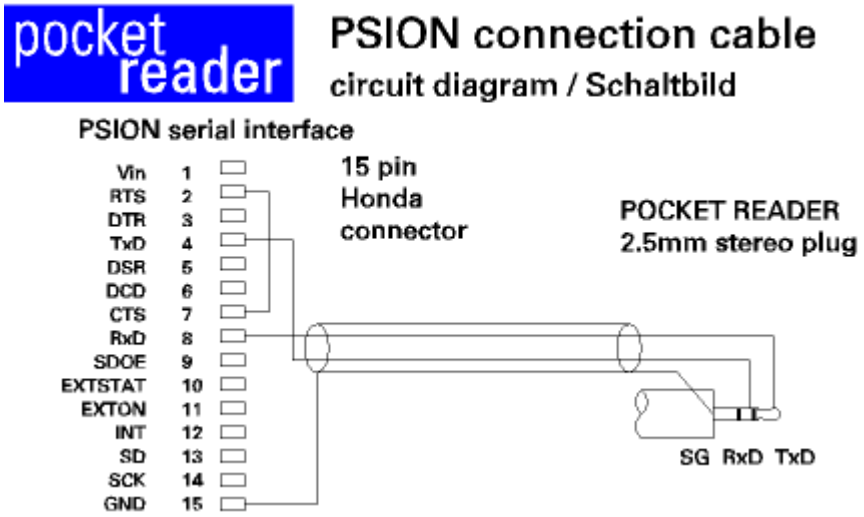

Figure 5: PSION Cable

1.1.5 Palm Connection

The pocket reader can be connected to a 3Com Palm III or Palm V serial port with the cable illustrated in the following figure:

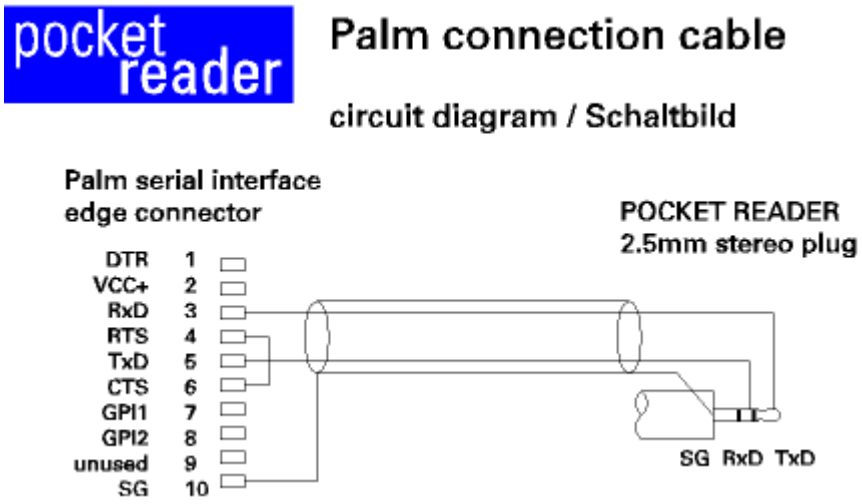

Figure 6: Palm cable

Note:
The Palm III and Palm V use different connectors. Thus, for the two series, two cable versions are necessary!

1.2 PC Software

In the following text, "PC" is used as synonym for all supported computer and operating system platforms. With the pocket reader, SW is available for the following systems:

Computers	Operating System
PCs	Windows 95, 98, Me, NT4.0, 2000, XP
PCs	Windows 3.1
PCs	Linux
Macintosh	PowerPC and 68K Processors, operating system versions 7.x, 8,1 and 8,5
Newton	MessagePad 2100, NewtonOS 2.0 and 2.1
Psion	Series 3c, 3mx, 5, 5mx, 5mxPro, and Siena, EPOC
Handheld PCs	Windows CE 2.0 and higher
3Com Palm	III and V, OS 3.0 and higher

Table 1: Supported computer and operating system platforms

With the PC-SW you can:
- Download and display text stored in the pocket reader to the PC,
- Erase the Flash EPROM on the pocket reader and
- Configure the pocket reader.

The text can processed further, stored and/or be transferred to other programs (word processing, database etc.)

1.3 Message flows

Information flow - PC ⇔ pocket reader

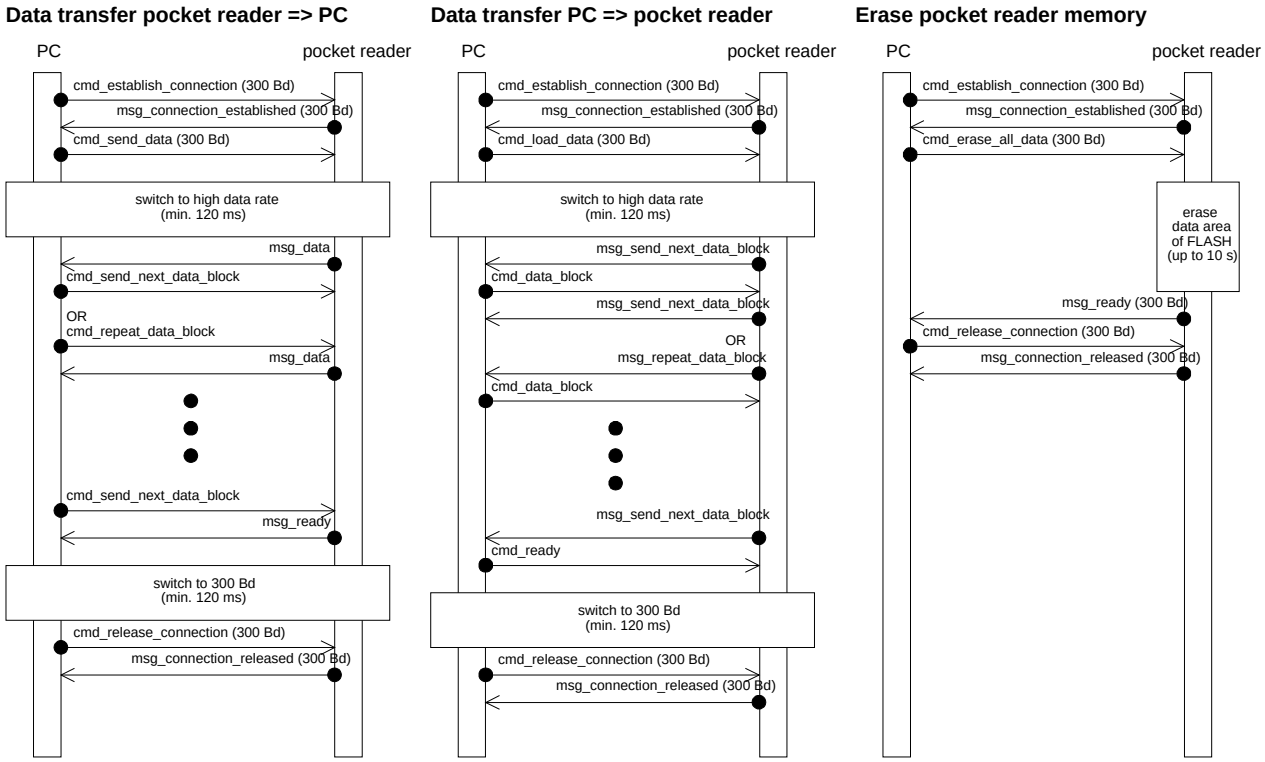

Figure 7: Information flow - PC ⇔ pocket reader

1.4 Transfer protocol

Communication between the PC and the pocket reader is performed by means of a simple unsecured protocol on the serial interface at 300 Baud with 8 data bits, even parity and 1 stop bit. For the duration of the actual data communication (text*, compiler/translator word list, Flash sector, ROM sector or memory dump*) the Baud rate is increased to between 1200 and 115200 Baud (bi-directional).

Baud rate	Opcode
300	0
1200	1
2400	2
4800	3
9600	4
19200	5
38400	6
57600	7
115200	8

Table 2: Baud rates

When changing the Baud rate you must wait at least 120ms after the complete transmission/reception of the last byte at the original rate, before the transmission of the next byte with the modified Baud rate can begin.

1.5 Handling of Transfer Errors

Data communication (in general with high Baud rates) is checksummed (XOR Checksum).

To handle transfer errors (checksum errors, or the PC detected a timeout), at least two attempts should be made to repeat the message. Thereafter, if no correct transfer results, then the connection is aborted using 'cmd_release_connection', and the Baud rate is reset to 300Bd afterwards, if applicable (the pocket reader should answer with 'msg_connection_released' at 300 Baud). If the PC does not receive a response within one second, then 'cmd_release_connection' is sent again at 300 baud. If the PC receives no further response, then this is indicated to the user (in a window).

In the pocket reader-FW no Timeout monitoring is implemented. In the case of repeated transfer errors it might be necessary to turn the pocket reader off and on manually.

2 Data Format

2.1 Configuration Data

For the format of the configuration data see the transfer commands 'cmd_configure_new' and 'msg_configuration'.

2.2 Text

The text read with the pocket reader is stored 'scanwise' and is transferred in the same way. A scan contains a maximum of 104 (*127*) characters, where in each case 2 bytes are stored.

The 'line break' inserted with the <Return> key <↵> is stored as one scan of the 'LF' (0x0a) character.

2.3 Characters

In the pocket reader the read characters are stored in 2 bytes (and likewise transferred to the PC): 1 byte character code and 1 byte additional information. In the table below all characters are listed, which the pocket reader can detect (according to the configured text language) (characters on a grey background can not be detected by the version 1 pocket reader).

Depending upon the platform of the communication SW it may be necessary to convert certain character codes to platform-specific values. This applies in particular to (French) œ, Œ and the EURO-Symbol €.

Note:
The PC-SW should convert "œ" and "Œ" into the letter combinations "oe" and "OE".
The pocket reader V1 does not recognise the € symbol. Future FW versions of the pocket reader will convert € into the letter combination "EUR" automatically.

2.3.1 Character codes

The pocket reader supports the following characters (characters on a grey background are not detected by the pocket reader V1):

Char.	Code	Char.	Code	Text language					
(upper)	(hex)	(lower)	(hex)	De	En	Fr	Sp	It	All
A	41	a	61						✓
B	42	b	62						✓
C	43	c	63						✓
D	44	d	64						✓
E	45	e	65						✓
F	46	f	66						✓
G	47	g	67						✓
H	48	h	68						✓
I	49	i	69						✓
J	4a	j	6a						✓
K	4b	k	6b						✓
L	4c	l	6c						✓
M	4d	m	6d						✓
N	4e	n	6e						✓
O	4f	o	6f						✓
P	50	p	70						✓
Q	51	q	71						✓
R	52	r	72						✓
S	53	s	73						✓
T	54	t	74						✓
U	55	u	75						✓
V	56	v	76						✓
W	57	w	77						✓
X	58	x	78						✓
Y	59	y	79						✓
Z	5a	z	7a						✓
À	c0	à	e0			✓		✓	
Á	c1	á	e1				✓		
Â	c2	â	e2			✓			
Ä	c4	ä	e4	✓					
È	c8	è	e8			✓		✓	
É	c9	é	e9			✓	✓	✓	
Ê	ca	ê	ea			✓			
Ë	cb	ë	eb			✓			
Ì	cc	ì	ec					✓	
Í	cd	í	ed				✓		
Î	ce	î	ee			✓			
Ï	cf	ï	ef			✓			
Ò	d2	ò	f2					✓	
Ó	d3	ó	f3				✓	✓	
Ô	d4	ô	f4			✓			
Ö	d6	ö	f6	✓					
Ù	d9	ù	f9			✓		✓	
Ú	da	ú	fa				✓		
Û	db	û	fb			✓			
Ü	dc	ü	fc	✓			✓		
		ß	df	✓					
Œ	01	œ	00			✓			
Ç	c7	ç	e7			✓			

Char.	Code	Char.	Code	Text language					
(upper)	(hex)	(lower)	(hex)	De	En	Fr	Sp	It	All
Ñ	d1	ñ	f1				✓		
<bl>	20								✓
<↵>	0a								✓
!	21								✓
"	22								✓
#	23								✓
$	24								✓
%	25								✓
&	26								✓
'	27								✓
(	28								✓
)	29								✓
*	2a								✓
+	2b								✓
,	2c								✓
-	2d								✓
.	2e								✓
/	2f								✓
0	30								✓
1	31								✓
2	32								✓
3	33								✓
4	34								✓
5	35								✓
6	36								✓
7	37								✓
8	38								✓
9	39								✓
:	3a								✓
;	3b								✓
<	3c								✓
=	3d								✓
>	3e								✓
?	3f								✓
@	40								✓
^	5e								✓
_	5f								✓
¡	a1						✓		
£	a3								✓
«	ab								✓
»	bb								✓
¥	a5								✓
§	a7								✓
¿	bf						✓		
÷	f7								✓
EUR (€)	02								✓

Table 3: Character codes

2.3.2 Character info

For every character the following additional information (in a byte) is stored and transferred:

Bit	Meaning	Remarks
0 (LSB), 1	Size	0..small, 1..medium, 2..large, 3..huge
2	*Italic*	*0..normal, 1..italic; not V1*
3	*Bold*	*0..normal, 1..bold; not V1*
4	*Serif*	*0..sans serif, 1..serif; not V1*
5	*Proportional*	*0..proportional, 1..monospaced; not V1*
6, 7	Quality	0..poor, 1..medium, 2..good, 3..excellent

Table 4: Additional info. for read characters

Classification of the size information:

Value	Char. height (Pixel)	Char. height (mm)	Char. size (Point)	Char. size (mm)
0	< 35	< 2,22	up to 8 pt	up to 2,82
1	35..42	2,22 .. 2,67	9..10 pt	3,18 .. 3,53
2	43..54	2,73 .. 3,43	11..13 pt	3,88 .. 4,59
3	> 54	> 3,43	14 pt and more	4,94 and more

Table 5: Character sizes

2.4 Flash, ROM

Data in the Flash EPROM or in the ROM of the pocket reader can be written sector-wise with 'cmd_load_data' or be read with 'cmd_send_data'. The Flash EPROM is 512 kB in size, the ROM in the V1 is 2 MB or in the V2 8 MB. The sectors of the Flash EPROM are 64 kB in size, those of the ROM are 128 kB; 512 or 1024 data blocks are to be transmitted accordingly.

3 Protocol

3.1 Command outline

Command (PC ⇨ pocket reader)	opcode (hex)	Message (pocket reader ⇨ PC)	opcode (hex)
cmd_establish_connection	00	msg_connection_established	80
cmd_release_connection	01	msg_connection_released	81
cmd_send_configuration	02	msg_configuration	82
cmd_configure_new	03	msg_ready	83
cmd_send_data	04	msg_data_block / msg_ready	84 / 83
cmd_send_next_data_block	05	msg_data_block / msg_ready	84 / 83
cmd_repeat_data_block	06	msg_data_block	84
cmd_erase_all_data	07	msg_ready	83
cmd_load_data	08	msg_send_next_data_block	85
cmd_data_block	09	msg_send_next_data_block / msg_repeat_data_block	85 / 86
cmd_ready	0A		

Table 6: Commands and messages - outline

The PC program frames each data communication or each functional command (i.e. to erase the memory) with the commands 'cmd_establish_connection' and 'cmd_release_connection'. The pocket reader responds to the data communication only if it is in Online Mode, thus beforehand the command cmd_establish_connection must be sent.

3.2 Parameter language

For different configuration adjustments the 'LANGUAGE' parameter is necessary. These are:

Parameter	Designation	Remarks
menu_language	Control language	Language for displaying menu texts and status messages
text_language	Text language	Language of the scanned text (relevantly for the language specific character set, e.g. umlauts in 'German', as well as for a possible context check by means of dictionary)
target_language	*Target language*	*Target language for translation*

Table 7: Languages - outline

Value	Designation	Remarks
00H	deutsch	
01H	english	
02H	français	
03H	español	
04H	italiano	
FFH	*auto*	*Recognition of the text language by the FW*

Table 8: Languages - values

3.3 Commands from PC ⇨ pocket reader

3.3.1 cmd_establish_connection

Byte #	Name	Value	Remarks
0	Opcode	00H	Open connection

The PC program sends this command before requesing data communication, e.g. text download (cmd_send_data), to read or modify the configuration, or delete the pocket reader memory. It serves among other things also to check whether the pocket reader is connected to the PC and switched on. The pocket reader answers with msg_connection_established and is now in Online Mode.

3.3.2 cmd_release_connection

Byte #	Name	Value	Remarks
0	opcode	01H	Close connection

The PC program sends this command after data communication has finished. The pocket reader answers with msg_connection_released and is again in the Offline Mode.

In general, this command is sent at the standard protocol Baud rate of 300 Baud. As an exception to this rule it may be sent at the higher Baud during data communication, in order to abort e.g. the text transfer. The pocket reader always answers at 300 Baud.

3.3.3 cmd_send_configuration

Byte #	Name	Value	Remarks
0	opcode	02H	The pocket reader transmits current configuration data.

With this command the PC program determines the current configuration of the pocket reader, which can be queried and modified by the user. This command is transmitted with the selection of the menu option " pocket reader configuration... ". Response: see 'msg_configuration'.

3.3.4 cmd_configure_new

Byte #	Name	Value	Remarks
0	opcode	03H	New configuration data is transmitted to the pocket reader
1	*mode*	*00H..scanner 01H..translator*	*is always transferred, but is not relevant in the V1.*
2	menu_language	00H..04H	Menu language (⇨ Tables 7 and 8).
3	text_language	00H..04H	Text language (⇨ Tables 7 and 8).
4	*target_language*	*00H..04H*	*Target language for translation (⇨ Tables 7 and 8); is always transferred, but is not relevant in the V1.*

With this command a new configuration is transmitted to the pocket reader. This command is sent when closing the configuration window of the PC program.

3.3.5 cmd_send_data

Byte #	Name	Value	Remarks
0	opcode	04H	Request the pocket reader to transmit data to the PC
1	data_rate	0..8	Baud rate for data communication (⇨ Table 2)
2	data_type	0..text *1..transl_hist* *2..flash_sect* *3..rom_sect* *4..data_memory* *more tbd.*	*Flash and ROM sector can be requested only in the test operation*
3	*sector_number*	*00H..07H Flash* *00H..3FH ROM*	*Transfer only if data_type is '3' or '4' (Flash or ROM sector)*

With this command the PC program requests the transfer of the stored data in the pocket reader (text, *translator history, Flash or ROM sector, memory dump*).

The transfer takes place block-by-block (a scan-block with max. 127 characters per scan, whereby for each character 2 bytes will be transferred: a 1 byte character code and 1 byte of additional info. - see Tables 4 and 5, *an entry in the translator history or in each case 128 bytes of a Flash or a ROM sector or from the internal data memory of the signal processor*). The first scan will be sent by the pocket reader as response to the command by means of msg_data_block (sent with the agreed upon - higher - Baud rate), all further blocks will be sent to the PC after cmd_send_next_data_block command. If a block is transmitted incorrectly (checksum errors) or the PC detects a timeout, then a repetition is requested using cmd_repeat_data_block. As soon as all the data is transferred, the pocket reader acknowledges the request for a further block with the response msg_ready.

The Baud rate for the transfer of the data (pocket reader ⇨ PC) is set to 'data_rate' after this command. After the last protocol element, i.e. the transmission of msg_ready as an acknowledgement - no more data is to be transferred -, all further transfer is handled again at 300 Baud. When changing the Baud rate after the complete transmitting/receiving of the last byte at the original Baud rate, you must wait at least 120 ms, until with the transmission of the next byte with the modified Baud rate can begin.

If nothing is stored in the pocket reader Flash EPROM, then the command is acknowledged with msg_ready. The user receives a message in a window in this case: 'No data is stored in the pocket reader'.

The data communication is secured with a checksum (XOR Checksum), whereby the Checksum is formed only over the net data (i.e. the protocol elements are not included in the checksum) - see also 'msg_data_block'.

3.3.6 cmd_send_next_data_block

Byte #	Name	Value	Remarks
0	opcode	05H	Request the pocket reader to transmit the next data block

See 'cmd_send_data'.

3.3.7 cmd_repeat_data_block

Byte #	Name	Value	Remarks
0	opcode	06H	Request the pocket reader to repeat the data block transmitted last

See 'cmd_send_data'.

3.3.8 cmd_erase_all_data

Byte #	Name	Value	Remarks
0	opcode	07H	Request the pocket reader to delete the part of the Flash EPROMs used for scanned text *and translations*

With this command the PC program requests the deletion of the Flash EPROM of the pocket reader, the pocket reader acknowledges with msg_ready.

If a part of the contents of the Flash EPROMs is to remain intact, then the PC must control this. It is possible e.g.:

- *All scanned text and the translation history are downloaded and buffered, then*
- *The Flash EPROM is deleted and afterwards*
- *Text and (max. 20) translation history entries are loaded into the pocket reader again.*

3.3.9 cmd_load_data - not supported in the release version V1

Byte #	*Name*	*Value*	*Remarks*
0	*Opcode*	*08H*	*Request the pocket reader to get data from the PC*
1	*data_rate*	*0..8*	*Baud rate for data communication (⇨ Table 2)*
2	*data_type*	*0..text 1..transl_hist 2..flash_sect 3..rom_sect more tbd.*	
3	*Sector_number*	*00H..07H Flash 00H..3FH ROM*	*Transfers only if data_type is '3' or '4' (Flash or ROM sector)*

With this command the PC program loads data (text, compiler History, Flash sector or ROM sector) into the pocket reader.

The pocket reader acknowledges, already at the higher Baud rate, by means of msg_send_next_data_block. The transfer takes place block-by-block (a text block, an entry in the translator history or in each case 128 bytes of a Flash or a ROM sector). The first block will be sent by the PC as response to the acknowledgement by means of cmd_data_block. All further blocks will be sent to the pocket reader after successive msg_send_next_data_block requests. If a block is transmitted incorrectly (checksum errors), then a repetition is requested with msg_repeat_data_block. As soon as the entire data is transferred, the PC acknowledges the request for further blocks with cmd_ready.

The Baud rate for the transfer of the data (pocket reader ⇨ PC) is set to data_rate after sending this command. After the last protocol item, i.e. the transmission of the 'cmd_ready' as an acknowledgement - no

more data is to be transferred -, all further transfer is handled again at 300 Baud. When changing the Baud rate, the PC must wait 120ms after the complete transmitting/receiving of the last byte with the original rate, before the transmitting of the next byte at the modified Baud rate can start.

The data communication is secured with a checksum (XOR Checksum), whereby the Checksum is formed only over the net data.

3.3.10 cmd_data_block

See also 'cmd_load_data'

3.3.10.1 Text

Byte #	*Name*	*Value*	*Remarks*
0	*opcode*	*09H*	*PC transmits data block (text) to the pocket reader*
1	*text_len (n)*	*1..100*	*Number of characters which can be transmitted*
2..n+1	*text_data*	*0..255*	*Data of the scanned text (in each case 1 byte character code and 1 byte addition info. - see Tables 4 and 5)*
n+2	*XOR-checksum*	*0-255*	*Checksum over all data (text_data)*

Transfer of data to the pocket reader: corresponds to scanned text

3.3.10.2 Translations

Byte #	*Name*	*Value*	*Remarks*
0	*opcode*	*09H*	*PC transmits data block (translations) to the pocket reader*
1-3	*transl_data*	*0-255*	*Translation info.*
4	*XOR-checksum*	*0-255*	*Checksum over translation info. (transl_data)*

Transfer of data to the pocket reader: Translation History entry

3.3.10.3 Flash or ROM sector

Byte #	*Name*	*Value*	*Remarks*
0	*Opcode*	*09H*	*PC data block (sector section) transmits to pocket reader*
1-128	*flash_data*	*0-255*	*Data*
129	*XOR-checksum*	*0-255*	*Checksum over data (flash_data)*

Transfer of data to the pocket reader: Flash or ROM sector

3.3.11 cmd_ready

Byte #	*Name*	*Value*	*Remarks*
0	*opcode*	*0AH*	*PC indicates to the pocket reader that all the data has been transferred*

See 'cmd_load_data'.

3.4 Messages from pocket reader ⇨ PC

3.4.1 msg_connection_established

Byte #	Name	Value	Remarks
0	opcode	80H	Acknowledgement when opening the connection

Response to 'cmd_establish_connection'.

3.4.2 msg_connection_released

Byte #	Name	Value	Remarks
0	opcode	81H	Acknowledgement when closing the connection (always transferred in low Baud rate)

Response to 'cmd_release_connection'. This message is always transferred with the standard protocol Baud rate (300 Baud). After transmitting this message the pocket reader executes a soft RESET.

3.4.3 msg_configuration

Byte #	Name	Value	Remarks
0	Opcode	82H	Configuration data
1	*Mode*	*00H..scanner* *01H..translator*	*Not relevant in the V1*
2	menu_language	00H..04H	Menu language (⇨ Tables 7 and 8)
3	text_language	00H..04H	Text language (⇨ Tables 7 and 8)
4	*target_language*	*00H..04H*	*Language for translations (⇨ Tables 7 and 8); not relevant in the V1*
5	free_memory	00H..64H	Decimal 0..100 Free storage space (representation in %)
6	batt_state	00H..03H	Battery voltage (0.. too low / 3.. full)
7	fw_vers_high	00H..FFH	pocket reader FW Version (upper Byte)
8	fw_vers_low	00H..FFH	pocket reader FW Version (lower Byte); should be displayed in two digits ('1.01') *if most significant bit is set (80H), then that means that the pocket reader can translate ("executive" version); this bit is not to be interpreted as part of the version number and is not to be displayed by the PC-SW. i.e. the 'lower' part of the version number cannot be larger than 127*

Response to 'cmd_send_configuration'.

3.4.4 msg_ready

Byte #	Name	Value	Remarks
0	Opcode	83H	

Response to 'cmd_configure_new', 'cmd_send_data', 'cmd_send_next_data_block' or 'cmd_erase_all_data'.

3.4.5 msg_data_block

See also 'cmd_send_data'

3.4.5.1 Text

Byte #	Name	Value	Remarks
0	Opcode	84H	pocket reader data block (scanned text) is transmitted to the PC
1	text_len (n)	1..127	Number of characters that can be transmitted
2..2*n+1	text_data	0..255	Data of the scanned text (in each case 1 byte Character code and 1 byte addition info. - see Tables 4 and 5)
2*n+2	XOR-checksum	0-255	Checksum over data (text_data)

Transfer of data to the PC: Text

3.4.5.2 Translations

Byte #	*Name*	*Value*	*Remarks*
0	*Opcode*	*84H*	*pocket reader data block (translations) is transmitted to the PC*
1-3	*Transl_data*	*0-255*	*Translation info.*
4	*XOR-checksum*	*0-255*	*Checksum of Translation info. (transl_data)*

Transfer of data to the PC: Translation History entry

3.4.5.3 Flash or ROM sector, memory dump

Byte #	*Name*	*Value*	*Remarks*
0	*Opcode*	*84H*	*Pocket reader data block (sector section) is transmitted to the PC*
1-128	*flash_data*	*0-255*	*Data*
129	*XOR-checksum*	*0-255*	*Checksum over data (flash_data)*

Transfer of data to the PC: Flash or ROM sector, memory dump

3.4.6 msg_send_next_data_block

Byte #	*Name*	*Value*	*Remarks*
0	*Opcode*	*85H*	*Pocket reader* requests PC to transmit the next data block

Transfer of data to the pocket reader (see 'cmd_load_data')

3.4.7 msg_repeat_data_block

Byte #	*Name*	*Value*	*Remarks*
0	*opcode*	*86H*	*Pocket reader requests the PC to repeat the last transmitted data block*

Transfer of data to the pocket reader (see 'cmd_load_data')

4 APPENDIX

4.1 Index of Tables

4.2 Index of Figures

4.3 Glossary

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
AOQL	Accepted Outgoing Quality Level
ATS	Österreichische Schilling
CE	Communie Européenne
CE	Consumer Electronic
CPU	central processing unit
CRC	Cyclic redundancy check
DOA	Dead on Arrival
DOS	Disk Operating System
DSP	Digital Signal Processor
EMC	electromagnetic compatibility
EMV	Elektromagnetische Verträglichkeit
EPROM	erasable/programmable read only memory
ESD	electrostatic discharge
EUR	Euro (1 Euro = 13,7603 ATS)
EWW	Elektronikwerk Wien
EZE	Entwicklungszentrum für Elektronik
FAA	Federal Aviation Administration
FCC	Federal Communications Commission
FW	Firmware
HTML	Hypertext Mark-up Language
HTTP	Hypertext transfer protocol
HW	Hardware
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
ISBN	international standard book number
ISO	International Standards Organization
JAA	Joint Aviation Authorities
KB	Kilobytes
MB	Megabytes

OCR	Optical Character Recognition (optische Zeichenerkennung)
PC	Personal Computer
PDA	Personal Digital Assistant
PR	pocket reader
PR	Public Relations
RAM	random access memory
ROM	read only memory
RTF	Rich Text Format
SW	Software
TATS	Tausend Österreichische Schilling
TCP/IP	Transmission Control Protocol/Internet Protocol
UL	Underwriters Laboratories
URL	Uniform Resource Locator
WWW	World Wide Web

4.4 Literature

Doppler: pocket reader User Interface (pr1_ui05.doc)