PC-Programmers Guide

Version 5

V 1.1

Do not distribute!

---

last update: 12-07-03 sr
SPORTident GmbH Arnstadt, Germany

homepage

e-mail

Dear user,

thank you for your interest in SPORTident. This information supports you to write software modules with access to the SPORTident hardware, software modules and databases. We are interested in any feedback to improve this document.

We would also be pleased to hear from the benefits in your projects. We like to publish all the tools made to support our customers best.

SPORTident is subject of continuously ongoing development. Although all system parts are tested out carefully before shipment to customers some limitations in special features can exist. AES is interested to receive error reports as detailed as possible.

For more information about the products please also have a look to our website.

Good luck and have fun using SPORTident!

Revision history

V1.1	12-07-03	sr	card data readout SI-C10/11/SIAC
V1.0	10-04-22	sr	robustness of inductive coupling
V0.3	05-12-29	sr	SI-Card6*, SI-Card8, SI-Card9, pCard handling
	06-03-14	sr	backup memory data format 8 Byte record size
	06-08-30	sr	note to read out tCard
	07-04-19	sr	reference time stamp interpretation
V 0.2	04-11-08	sr	auto send instruction 0xD3 implemented
	04-11-25	sr	warning backup memory access in auto send (chapter 5)

Content

1. Hard- and Software overview, compatibility

Although there are different hardware and software versions all the key system features are downwards compatible with only small limitations. The table features these aspects more in detail.

hardware	sw-version	PC interface software	PC-prog.-guide
BSF3-2, BSF3-3 BSM3-2, BSM3-5 BSx4 BSx6	v4.06 and higher	SI-Manager 9.8 SI-Config 1.2 (time triggered mode not supported)	v4.0x
BSx7 BSx8	v5.29 and higher	SI-Config	v5

2. Communication protocol SPORTident hardware - host

2.1. Instruction sets

The communication protocol used is simple and robust. Important features are the use of start- and stop-signs for all records. The communication is half duplex. Each command string will be responded. In case of no response the host can control the communication process by means of a time out condition.

Two different protocol formats are used. The common base structure is:

STX	command code	parameter/data	ETX

There are a number of characters with importance for control purposes:

Protocol character	**Coding (hex)**	**Meaning**
STX	02	Start of text, first byte to be transmitted
ETX	03	End of text, last byte to be transmitted
ACK	06	Positive handshake return
NAK	15	Negative handshake return
DLE	10	DeLimitEr to be inserted before data characters 00-1F

a) command code < 0x80

The base protocol is format free and uses the fully binary parameter range. Each data character in control characters range (00 - 1F) has to be introduced by DLE. The delimiter sign is used to indicate a following data byte. Reading a string, DLEs have to be removed.

As an exception also the instruction „0xC4“ is part of the base protocol set.

b) command code >= 0x80

An additional set of instructions is implemented in the SPORTident stations series 7 and 8. Most important are a higher data transmission speed and improved reliability in case of inductive coupling. The additional instruction set uses a length byte as first parameter and does not use DLE's. The data transmission security is improved by implementing a 16 bit CRC value into the protocol.

It is recommended to use always the additional instruction set instead of the base protocol instructions.

The additional instruction set is active and applicable when setting the “extended protocol”-flag in SI-Config when programming the SI-Station.

STX	command code	length byte	parameter/data	CRC1, CRC0	ETX

length byte - number of parameter/data bytes following, CRC excluded
CRC1, CRC0 - 16 bit CRC-value
The CRC is computed including the command byte and the length byte. The computation algorithm is given as *.c and *.bas listing in appendix A.

2.2. Protocol start up sequence

To enable a „STX“ detection by the SPORTident station under all circumstances two STX-signs should be transmitted at the beginning of each record. The SPORTident master stations 7 and 8 feature with an auto-wakeup mechanism activated with the first sign received by the SPORTident station on the serial line. To avoid any data losses an additional wakeup-byte „0xFF“ should be sent first. So the complete start-up sequence is:

0xFF	STX	STX	command code	...	...

2.3. Station detect procedure

Writing robust PC-software there is a need to detect stations baud rate, device series and other settings to establish a communication link, use the most advanced instructions possible and read settings to interpret stations data correctly. This can be done in different ways. A sample sequence is given following.

– detect master stations baud rate and device series

Prerequisite is that stations of the series 3, 4, 6 are switched on. Stations baud rate and series can be detected be sending out a protocol instruction with a preselected baud rate. A time out condition must be set.

- set PC baud rate to 38400
- send extended instruction “Set MS-Mode” (0xF0) with switch to direct communication

Set ms-mode	0xF0	master station sets transfer mode	STX, 0xF0, 0x01, M/S, CRC1, CRC0, ETX	STX, 0xF0, 0x03, CN1, CN0, M/S, CRC1, CRC0, 0x03 | NAK

M/S = 0x4D direct communication
(note: M/S = 0x53 transparent communication to slaved station)

In case of no answer the baud rate has to be changed and the instruction has to be repeated. In general it is a “trial-and-error” step.

- positive answer: BSM7/8 at 38400 baud; finish
- in case of no response set PC baud rate to 4800
- repeat instruction 0xF0
- positive answer: BSM7/8 at 4800 baud; finish
- in case of NAK send instruction 0x70

Set ms-mode	0x70	master station sets transfer mode	STX, 0x70, M/S, ETX	STX, 0x70, CN, M/S, ETX | NAK

- positive answer: BSM3/4/6 at 4800 baud; finish
- no answer: error

- read protocol mode (BS7/8)

The protocol mode flag can be read out by using the „Get system value“ (0x83) instruction.

Get system value	0x83	Get protocol configuration	STX, 0x83, 0x02, 0x74 , 0x01, CRC1, CRC0, ETX	STX, 0x83, 0x04, CN1, CN0, 0x74, CPC, CRC1, CRC0, ETX | NAK

CN1, CN0	2 bytes	stations code number 1...999
0x74	1 byte	data address in the system memory
CPC	1 byte	protocol configuration, bit mask value
		xxxxxxx1b extended protocol
		xxxxxx1xb auto send out
		xxxxx1xxb handshake (only valid for card readout)
		xxx1xxxxb access with password only
		1xxxxxxxb read out SI-card after punch (only for punch modes; depends on bit 2: auto send out or handshake)
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN (0x04, 0x14 for the request)

The “extended protocol” bit (CPC, bit 0) defines these properties:

- 8 bytes data record size in the stations backup memory in punch modes (otherwise 6 bytes)
- sub seconds resolution in the sub second value byte
- possibility to handle SI-Card6*, SI-Card8, SI-Card9, pCard in card readout mode

2.4. Interface parameters

baud-rates: 4800 baud, 38400 baud
bits: 8
parity: no
stop bits: 1
hardware handshake: no

The SPORTident stations series 7 and 8 feature with a second higher baud rate of 38400 bit/s.

The BSM7-USB station features 38400 bit/s transmission speed after each wakeup.

Two instructions „0x7E“ or „0xFE“ can be used to adjust the communication speed.

Set baud rate	0x7E	Set baud rate serial port	STX, 0x7E, SPEED, ETX	STX, 0x7E, CN, SPEED, ETX | NAK
Set baud rate	0xFE	Set baud rate serial port	STX, 0xFE, 0x01, SPEED, CRC1, CRC0, ETX	STX, 0xFE, 0x03, CN1, CN0, SPEED, CRC1, CRC0, ETX | NAK

The request comes with the speed of the older baud rate. The Station switches the speed after the response. Following commands have to use the new speed.

SPEED	1 byte	0 → 4800 baud 1 → 38400 baud
CN	1 byte	stations code number 1...254
CN1, CN0	2 bytes	stations code number 1...999
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN

2.5. Robustness of inductive coupling

Access to SPORTident stations BSFx is possible by creating an inductive coupling link between a BSFx station and a BSMx master station. BSFx and BSMx stations should be devices of the same hardware generation (series 7/8 are one and the same generation). Next the BSMx station has to be switched into transparent (slave) mode by using instruction 0xF0 with M/S = 0x53 as parameter. All the following instructions are forwarded to the slaved inductive coupled station.

The slaved station must be in active mode. BSMx and BSFx station need to be synchronized to enable inductive coupled data transmission. Synchronisation process starts with the first instruction transmitted. Both stations are switched into service mode.

Synchronisation and inductive data transmission in general are a sensitive process influenced by a number of factors. Application software should support synchronisation process by implementing a repeat loop for transmitting instructionsfrom the host to a slaved station. The repeat loop has to be used after having got an NAK as feedback value. Also a variable delay time should be added before sending out next instruction inside the repeat loop. Repeat number should be 3...5 and delay time should vary in steps of some 10 ms.

3. Access to the Control Stations backup memory

3.1. Backup memory sizes and memory access

The control station features with an internal non-volatile backup memory.

Depending on the working mode programmed this memory stores complete control card data records or data of the specific punching processes. The table gives an overview about memory sizes and address ranges.

Device	**Memory size**	**Start address**	**End address**	**Capacity Cards** (128 byte package = SI-card 5)	**Capacity punches** (6 byte record size)	**Capacity punches** (8 byte record size)
BSM3-2, BSF3-2	4 k byte	0100h	0FFFh	30	640	
BSF4, BSM4	8 k byte	0100h	1FFFh	62	1322	
BSM3-2, BSF3-2, BSF3-3, BSF4, BSM4, BSF6, BSM6	16 k byte	0100h	3FFFh	126	2688	
BSM3-5, BSM4-5	64 k byte	0100h	FFFFh	510	10752	
BSx7, BSx8	128 k byte	0100h	1FFFFh	1022	21802	16352

Memory access and navigation are done using the „Get backup data“ instruction. The navigation is based on the „memory pointer“ value. The memory pointer directs to the first non used memory location. The memory pointer value has to be read using the „Get system value“ (0x83) instruction.

Get system value	0x83	Get backup memory pointer	STX, 0x83, 0x02, 0x1C , 0x07, CRC1, CRC0, ETX	STX, 0x83, 0x0A, CN1, CN0, 0x1C, 7 bytes data, CRC1, CRC0, ETX | NAK

LEN	1 byte	length byte, 2/10 bytes follow
CN1, CN0	2 bytes	stations code number 1...999
0x1C	1 byte	data string start address in the system memory
data	7 bytes	the 4 byte backup memory address pointer is part of the data string: EP3, EP2, xx, xx, xx, EP1, EP0
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN (0x74, 0x06 for the request)

The backup memory has a size of 128k byte = 0x200000. The lowest address is 0x100 or the backup memory pointer address - 128k. Data stored in the backup memory have to be read out by specifying the backup memory start address and the number of bytes to be read. The maximal record size is 128 bytes. In case of errors in the readout process it is recommended to reduce the record size. The record size NUM should be adapted according to the size of data packages stored in the stations memory:

card data	128 bytes
punch data, 6 byte format	120 bytes or a multiplier of 6
punch data, 8 byte format	128 bytes or a multiplier of 8

Get backup data	0x81	Get data from backup memory	STX, 0x81, 0x04, ADR2 , ADR1, ADR0, NUM, CRC1, CRC0, ETX	STX, 0x81, LEN, CN1, CN0, ADR2, ADR1, ADR0, <NUM data bytes>, CRC1, CRC0, ETX | NAK

LEN	1 byte	length byte, 4/ANZ data byte +5
ADR2, ADR1, ADR0	3 bytes	backup memory start address for the readout process
CN1, CN0	2 bytes	stations code number 1...999
NUM data byte	x bytes	backup memory data string
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN

The backup memory is organized as a ring buffer. The backup memory address pointer increments automatically and the overflow resets the pointer to the start value. First data will be overwritten.

The "Get backup data" instruction does not clear the backup memory. To reset the backup memory address pointer the instruction „Erase bdata“ has to be used.

Erase bdata	0xF5	Reset backup memory address pointer	STX, 0xF5, 0x00, CRC1, CRC0, ETX	STX, 0xF5, 0x02, CN1, CN0, CRC1, CRC0, ETX | NAK

LEN	1 byte	length byte, 0/2
CN1, CN0	2 bytes	stations code number 1...999
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN

3.2. Interpretation of the data read out

a) punch/trigger data

a1 „extended protocol“ flag not set

6 byte record size

SI1, SI0	2 bytes	SI card number
TH	1 byte	time High-Byte / error-code
TL	1 byte	time Low-Byte
TD	1 byte	time day-of-week/half day* 1 Byte b (see “DATE0” below)
SI2	1 byte	card number / card series (SI-card 5)

a2 „extended protocol“ flag set

8 byte record size

SW 5.54-

storage order: S1, S0, TH, TL,TD, SI2, TSS, SI3

SI3, SI2, SI1, SI0	4 bytes	SI card number
TD	1 byte	time day-of-week/half day* 1 Byte b
TH	1 byte	time High-Byte / error-code
TL	1 byte	time Low-Byte
TSS	1 byte	time sub seconds

SW 5.55+

storage order: SI2-SI1-SI0-DATE1-DATE0-TH-TL-MS

SI2, SI1, SI0		3 bytes	SI card number
DATE1, DATE0		2 bytes	date
DATE1	bit 7-2	6 bit year 0-64 part of year	
	bit 1-0	bit 3-2 - part of 4bit-month 1-12	
DATE0	bit 7-6	bit 1-0 - part of 4bit month 1-12	
	bit 5-1	5bit day of month 1-31	
	bit 0	am/pm halfday	
TH, TL		2 bytes	12h binary punching time
MS		1 byte	8bit 1/256 of seconds

>> We reserve the right to modify the 8 byte record size format in a non-compatible way!

b) control card data

Control card data are stored in multiples of 128 bytes.

SI-Card5	1*128 bytes
SI-Card8, SI-Card9, pCard	2*128 bytes
SI-Card6	3*128 bytes[1]
SI-Card6*	8*128 bytes[1]

There are characteristical pattern in the data stored in the very first block of a new card. This enables to identify the start of new card data and to determine the SI-Card type.

SI-Card5	0x00, 0x07 as bytes 30 and 31 (starting with 0)
SI-Card8, SI-Card9, pCard	0xEA, 0xEA, 0xEA, 0xEA as bytes 4...7 (starting with 0)[2]
SI-Card6, SI-Card6*	0xED, 0xED, 0xED, 0xED as bytes 4...7 (starting with 0)[2]

[1]The number of data blocks for the SI-Card depends on the "BN" parameter in the "Get card block" readout command. All data blocks of the SI-card 6 read out are stored in the backup memory. If BN=0x08 in addition the information of the "card readout configuration byte" (stored in the stations system memory) is evaluated. Please have a look at the card readout instruction description for more information.

[2]Within a group the SI-Cards are identified based on their unique SI-Card number family.

4. SPORTident-card data reading

4.1. Overview

The data stored in SPORTident cards are read into a PC software by means of a SPORTident BSM station programmed in "card readout" mode. There are two modes supported.

a) handshake mode
- card detected (auto send)
- read card data
- card removed (auto send)

b) auto send mode
- read card data with auto send

Firmware 5.49 (or higher) is needed to handle SI-Card8, SI-Card9 and pCard. For the SI-Card6* at least firmware 5.51 must be implemented in the BSM station. "Extended protocol"-flag must be set. SI-tCard needs firmware 5.53 (or higher) to be read out.

The card data have to interpreted according to the card data structure. It is described in "SI_cards_data_structure_developer.ods".

If the flag "Extended protocol" is not set SI-cards 8/9/p/t are modeled as SI-cards 6 by the BSM7 station (compatible mode).

4.2. SI-Card5

In handshake mode the SI-BSM stations announces a SI-Card5 detected by sending the 0xE5-instruction.

SI-Card 5 detected	0xE5	auto send		STX, 0xE5, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

CN1, CN0	2 bytes	stations code number 1...999
SI3...SI0	4 bytes	SI-card number
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN

If a SI-Card5 is detected the host initiates card data readout process.

Get SI-Card5	0xB1	read out data	STX, 0xB1, 0x00, CRC1, CRC0, ETX	STX, 0xB1, 0x82, CN1, CN0, 128 byte, CRC1, CRC0, ETX

If a SI-Card5 is removed from the SI-BSM station the 0xE7-instruction is sent out.

SI-Card removed	0xE7	auto send		STX, 0xE7, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

In auto send mode the 0xB1-data string automatically is sent out by the station after having read out a SI-Card5.

4.3. SI-Card8, SI-Card9, pCard, tCard

In handshake mode the SI-BSM stations announces a SI-Card8/9/p detected by sending the 0xE8-instruction.

SI-Card8/9/p detected	0xE8	auto send		STX, 0xE8, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

If a SI-Card8/9/p is detected the host initiates card data readout process. Two data blocks of 128 bytes each are read out in two separate steps. BN addresses the block number

Get SI-Card8/9/p	0xEF	read out data block	STX, 0xEF, 0x01, BN, CRC1, CRC0, ETX	STX, 0xEF, 0x83, CN1, CN0, BN, 128 byte, CRC1, CRC0, ETX

BN 1 byte block number 0/1

If a SI-Card8/9/p is removed from the SI-BSM station the 0xE7-instruction is sent out.

SI-Card removed	0xE7	auto send		STX, 0xE7, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

In auto send mode the 0xEF-data string automatically is sent out by the station after having read out a SI-Card8/9/p. Two data blocks of 128 bytes each are sent distinguished by the block number 0/1.

4.4. SI-Card6, SI-Card6*

In handshake mode the SI-BSM stations announces a SI-Card6/6* detected by sending the 0xE6-instruction.

SI-Card6/6* detected	0xE6	auto send		STX, 0xE6, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

If a SI-Card6/6* is detected the host initiates card data readout process. Data blocks of 128 bytes each are read out. BN addresses the block number

Get SI-Card6/6*	0xE1	read out data block	STX, 0xE1, 0x01, BN, CRC1, CRC0, ETX	STX, 0xE1, 0x83, CN1, CN0, BN, 128 byte, CRC1, CRC0, ETX

BN 1 byte block number 0...8

If BN=8 then all blocks marked in the "CardBlocks"-byte are read out by the station continuously. For compatibility reasons CardBlocks = 0x00 is interpreted like CardBlocks = 0xC1 which addresses blocks 0, 6, 7. CardBlocks = 0xFF reads out all 8 data blocks of the SI-Card6/6*.
CardBlocks = 0xFF is also evaluated by the SI-Station in all punching modes. Using this value each SI-Card6 will be handled as SI-Card6*.

The CardBlocks-byte is stored in SI-stations system memory. It can be read out by using instruction 0x83.

Get system value	0x83	Get configuration	STX, 0x83, 0x02, 0x33 , 0x01, CRC1, CRC0, ETX	STX, 0x83, 0x04, CN1, CN0, 0x33, CardBlocks, CRC1, CRC0, ETX | NAK

If a SI-Card6/6* is removed from the SI-BSM station the 0xE7-instruction is sent out.

SI-Card removed	0xE7	auto send		STX, 0xE7, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

In auto send mode the 0xE1-data string automatically is sent out by the station when reading out a SI-Card6/6*. The number and configuration of data blocks is according to the CardBlocks-byte settings.

4.5. SI-Card10, SI-Card11, SIAC

SI-Card10/11/SIAC are handled by the SI-Station similar to SI-Card8/9. Main difference is the larger memory size. Data packages have a unique size of 128 Bytes each. In difference to SI-C6 the application data are stored starting at block address 4.

In handshake mode the SI-BSM stations announces a SI-Card10/11/SIAC detected by sending the 0xE8-instruction.

SI-Card8/9/p detected	0xE8	auto send		STX, 0xE8, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

If a card is removed from the SI-BSM station the 0xE7-instruction is sent out.

SI-Card removed	0xE7	auto send		STX, 0xE7, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX

If a card is detected the host initiates card data readout process. Data blocks of 128 bytes are read out. BN addresses the block number

Get SI-Card data	0xEF	read out data block	STX, 0xEF, 0x01, BN, CRC1, CRC0, ETX	STX, 0xEF, 0x83, CN1, CN0, BN, 128 byte, CRC1, CRC0, ETX

BN 1 byte block number 0...7

By using parameter BN = 8 the five most relevant data blocks 0, 4...7 are read out at once.

For compatibilty reasons card readout by using instructions 0x61 is still supported. Compatibility is limited to 64 punches only. These data are addressed by using paramter BN = 6 (punches 0...31) and BN = 7 (punches 32...63). Using parameter BN = 8 by default three data blocks 0, 6, 7 are read out.

In auto send mode data blocks are sent out according to the BN = 8 parameter. This means blocks 0, 6, 7 in standard mode and blocks 0, 4, 5, 6, 7 in extended mode.

5. Data record transmission

SPORTident features with a fully bidirectional data exchange between the control card and the control station. Complete data records are stored both in the card and in the stations backup memory.

In addition stations can transmit a data record using the serial communication line. For this purpose the "auto send" flag in the stations settings must be activated. The SPORTident stations 7/8 feature with an extended auto send protocol. As a new part a record number is implemented. The supervision of the record number by the receiver enables to control the completeness of the data records. Because the backup memory location of the data record is used as record number incomplete data can be recovered simply by reading out a part of the stations backup memory. The instruction "Get backup data" (0x81, see chapter 3.1) has to be used. A complete verified data exchange in a closed information loop can be established.

PC-access to the station can delay the handling of a SI-card by the station. So the PC software has to be written in a way that the real time behaviour of the SI-station still meets the requirements. Especially the number of bytes read from the backup memory in one cycle should be as small as possible (8 bytes only).

Remote control of control stations settings is not limited to any instructions. So it is important to be able to control stations real time clock within a session from time to time.

Auto send record:

Transmit record	0xD3	Transmit punch or trigger data in auto send mode		STX, 0xD3, LEN, CN1, CN0, SN3, SN2, SN1, SN0, TD, TH, TL, TSS, MEM2, MEM1, MEM0, CRC1, CRC0, ETX | NAK

LEN	1 byte	length byte, 0Dh = 13 byte	
CN1, CN0	2 bytes	stations code number 1...999	
SN3...SN0	4 bytes	SI-Card number	
TD	1 byte	day-of-week/half day	
		bit5...bit4	4 week counter relative
		bit3...bit1	day of week
			000b Sunday
			001b Monday
			010b Tuesday
			011b Wednesday
			100b Thursday
			101b Friday
			110b Saturday
		bit0	24h counter (0-am, 1-pm)
TH...TL	2 bytes	12h timer, binary	
TSS	1 byte	sub second values 1/256 sec	
MEM2...MEM0	3 bytes	backup memory start address of the data record	
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN	

Instruction 0xD3 is also used to transmit a data record caused by an external trigger signal like a broken light beam. To indicate such a record the fictive SI-Card number SN4...SN0 = 0 is used. In that way the features offered by the older instruction 0x54 are now part of instruction 0xD3.

6. Advanced control stations settings

The SPORTident stations 7/8 use an internal time resolution of 1/256 sec. The flag “extended protocol” defines

- if the sub second time information is stored in the backup memory as part of the 8 byte data record
- if it is used as part of the data record transmitted if “auto send” is active as described in chapter 5

The sub second value is also stored as part of the start and finish punch in the SPORTident cards if the flag “sprint_4ms” is set.

7. BS7, 8 additional instruction syntax

Set Baud rate (0x7E, 0xFE)

see chapter 2.4.

Get backup data (0x81)

see chapter 3.1.

Get system value (0x83)

see chapter 2.3., 3.1.

Erase backup data (0xF5)

see chapter 3.1.

Set/Get time (0xF6, 0xF7)

Set time	0xF6	Set stations time registers	STX, 0xF6, LEN, CD2, CD1, CD0, TD, TH, TL, TSS, CRC1, CRC0, ETX	STX, 0xF6, LEN, CN1, CN0, CD2, CD1, CD0, TD, TH, TL, TSS, CRC1, CRC0, ETX | NAK
Get time	0xF7	Station sends current time	STX, 0xF7, LEN, CRC1, CRC0, ETX	STX, 0xF7, LEN, CN1, CN0, CD2, CD1, CD0, TD, TH, TL, TSS, CRC1, CRC0, ETX | NAK

LEN	1 byte	length byte, 0/9 byte follow	
CN1, CN0	2 bytes	stations code number 1...999	
CRC1, CRC0	2 bytes	16 bit CRC value, computed including command byte and LEN	
CD2...CD0	3 bytes	calendar: yy-mm-dd; binary format	
TD	1 byte	day-of-week/half day	
		bit7, bit6	extendend code number bits b9, b8 (0...1023); 0 default
		bit5...bit4	4 week counter relative
		bit3...bit1	day of week
			000b Sunday
			001b Monday
			010b Tuesday
			011b Wednesday
			100b Thursday
			101b Friday
			110b Saturday
		bit0	24h counter (0-am, 1-pm)
TH...TL	2 bytes	12h timer, binary	
TSS	1 byte	sub second values 1/256 sec	

Appendix A **CRC computation**

```
//*************************************************************************
//** Author: Jürgen Ehms
//** File name: crc529.c
//** Description: Program to generate 16 BIT CRC
//** Return values: 16 BIT CRC
//** Error messages: none
//** Version    last change    description
//** 1.00       07.09.2004
//*************************************************************************

#define POLYNOM 0x8005

unsigned int crc(unsigned int uiCount,unsigned char *pucDat)
{
short int iTmp;
unsigned short int uiTmp,uiTmp1,uiVal;
unsigned char *pucTmpDat;

if (uiCount < 2) return(0);        // response value is "0" for none or one data byte
pucTmpDat = pucDat;
uiTmp1 = *pucTmpDat++;
uiTmp1 = (uiTmp1<<8) + *pucTmpDat++;

if (uiCount == 2) return(uiTmp1);   // response value is CRC for two data bytes
for (iTmp=(int)(uiCount>>1);iTmp>0;iTmp--)
{
if (iTmp>1)
{
 uiVal = *pucTmpDat++;
 uiVal= (uiVal<<8) + *pucTmpDat++;
}
else
{
 if (uiCount&1)              // odd number of data bytes, complete with "0"
 {
  uiVal = *pucTmpDat;
  uiVal= (uiVal<<8);
 }
 else
 {
  uiVal=0; //letzte Werte mit 0
 }
}

for (uiTmp=0;uiTmp<16;uiTmp++)
{
  if (uiTmp1 & 0x8000)
  {
    uiTmp1  <<= 1;
    if (uiVal & 0x8000)uiTmp1++;
    uiTmp1 ^= POLYNOM;
  }
  else
  {
    uiTmp1  <<= 1;
    if (uiVal & 0x8000)uiTmp1++;
  }
  uiVal <<= 1;
 }
}
return(uiTmp1);
}
//*************************************************************************
//** Author: Jürgen Ehms
```

```
//** File name: crc529.bas
//** Description: Program to generate 16 BIT CRC
//** Version    last change    description
//** 1.00       07.09.2004
//************************************************************************
Public Function crc(ByVal s As String) As Long

Const poly = &H8005&
Const BITF = &H8000&

Dim i As Long, j As Long, tmp As Long, val As Long

If s = "" Or Len(s) < 2 Then
  crc = 0
Else

i = 1
tmp = 0
tmp = Asc(Mid(s, i, 1))
tmp = tmp * 256
i = i + 1
tmp = tmp + Asc(Mid(s, i, 1))

If Len(s) = 2 Then
  crc = tmp
Else
  For i = 3 To Len(s) + 2  'Alle Byte+ 2Byte 0 am Ende
    If i < Len(s) Then
      val = Asc(Mid(s, i, 1))
      val = val * 256
      i = i + 1
    val = val + Asc(Mid(s, i, 1))
    Else
     If i = Len(s) Then
      val = Asc(Mid(s, i, 1))
      val = val * 256 'Es war ungerade Anzahl von Bytes
     Else
      val = 0 'letzte Werte mit XOR mit 0
     End If
     i = i + 2 'Abbruch für Schleife
    End If
  'Berechnen des CRC
   For j = 0 To 15
     If tmp And BITF Then
       tmp = (tmp + tmp) And &HFFFF&
       If val And BITF Then
         tmp = (tmp + 1) And &HFFFF&
       End If
       tmp = (tmp Xor poly) And &HFFFF&
     Else
       tmp = (tmp + tmp) And &HFFFF&
       If val And BITF Then
         tmp = (tmp + 1) And &HFFFF&
       End If
     End If
     val = (val + val) And &HFFFF&
    Next
  Next
  crc = tmp
  End If
End If
End Function
```

Appendix B Base instruction set BS3, 4, 6, 7, 8

The instructions are described in “PC programmers guide 4.x”.

Appendix C BS7, 8 only instructions

a) Base instruction set

Command	Code	Action	Command string	Response
Set baud rate	0x7E	Set baud rate on serial port	STX, 0x7E, SPEED, ETX	STX, 0x7E, CN, SPEED, ETX | NAK

b) Additional instruction set

Command	Code	Action	Command string	Response
Get backup data	0x81	Get data from backup memory	STX, 0x81, LEN, ADR2 , ADR1, ADR0, NUM, CRC1, CRC0, ETX	STX, 0x81, LEN, CN1, CN0, ADR2, ADR1, ADR0, <NUM data bytes,> CRC1, CRC0, ETX | NAK
Get system value	0x83	Get system data	STX, 0x83, LEN, ADR , ANZ, CRC1, CRC0, ETX	STX, 0x83, LEN, CN1, CN0, ADR, ANZDAT, CRC1, CRC0, ETX | NAK
Get SI-Card5	0xB1	read out data	STX, 0xB1, 0x00, CRC1, CRC0, ETX	STX, 0xB1, 0x82, CN1, CN0, 128 byte, CRC1, CRC0, ETX
Transmit record	0xD3	Transmit punch or trigger data in auto send mode		STX, 0xD3, 0x0D, CN1, CN0, SN3, SN2, SN1, SN0, TD, TH, TL, TSS, MEM2, MEM1, MEM0, CRC1, CRC0, ETX | NAK
Get SI-Card6/6*	0xE1	read out data block	STX, 0xE1, 0x01, BN, CRC1, CRC0, ETX	STX, 0xE1, 0x83, CN1, CN0, BN, 128 byte, CRC1, CRC0, ETX
SI-Card 5 detected	0xE5	auto send		STX, 0xE5, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX
SI-Card6/6* detected	0xE6	auto send		STX, 0xE6, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX
SI-Card removed	0xE7	auto send		STX, 0xE7, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX
SI-Card8/9/p detected	0xE8	auto send		STX, 0xE8, 0x06, CN1, CN0, SI3, SI2, SI1, SI0, CRC1, CRC0, ETX
Get SI-Card8/9/p	0xEF	read out data	STX, 0xEF, 0x01, BN, CRC1, CRC0, ETX	STX, 0xEF, 0x83, CN1, CN0, BN, 128 byte, CRC1, CRC0, ETX
Set ms-mode	0xF0	master station sets transfer mode	STX, 0xF0, 0x01, M/S, CRC1, CRC0, ETX	STX, 0xF0, 0x03, CN1, CN0, M/S, CRC1, CRC0, 0x03 | NAK
Erase bdata	0xF5	Reset backup memory address pointer	STX, 0xF5, 0x00, CRC1, CRC0, ETX	STX, 0xF5, 0x02 CNH, CNL, CRC1, CRC0, ETX | NAK
Set time	0xF6	Set stations time registers	STX, 0xF6, 0x07, P1...P7, CRC1, CRC0, ETX	STX, 0xF6, 0x09, CN1, CN0, P1...P7, CRC1, CRC0, ETX | NAK
Get time	0xF7	Station sends current time	STX, 0xF7, 0x00, CRC1, CRC0, ETX	STX, 0xF7, 0x09, P1...P7, CRC1, CRC0, ETX | NAK
Set baud rate	0xFE	Set baud rate serial port	STX, 0xFE, 0x01, SPEED, CRC1, CRC0, ETX	STX, 0xFE, 0x03, CN1, CN0, SPEED, CRC1, CRC0, ETX | NAK