Specification STARCOIN Cash Register Interface

Version 0.9.1/Revision 30.10.98

Author O. Pannke - 3FE5

Status PRELIMINARY/CONFIDENTIAL

Version 0.9.1/Revision 30.10.98

Giesecke & Devrient GmbH
Prinzregentenstr. 159
Postfach 80 07 29
81607 München

Giesecke & Devrient GmbH
Prinzregentenstr. 159
Postfach 80 07 29
81607 München
Germany

Content

1 Introduction

This specification describes the interface between the ICT800/810 (ICT) or and a cash register. Cash register means an intelligent cash register system developed by other manufacturers.

Which commands are realised in the software version of a terminal is recognisable at the software string. This software string can be read from the cash register (see 0 3.3 'Software Version).

The functions transfer and payment transaction can be started from the cash register.

The functions are realised via suitable commands which are specified in the following.

1.1 Release Information

Version 0.04 08.06.1998 first preliminary version

Version 0.9.1 19.08.1998 second preliminary version

changes to version 0.04:

2.3:	example of a CRC calculation
2.4.5:	detailed course of a command
2.5:	example for a communication
3.1	more than one Logon possible
3.3	format of 'software-string' specified
3.4.1	6 byte amount value, BCD coded
	2 byte <info> length
3.4.1.1	new, optional information for a payment transaction
3.4.2	2 byte <info> length
3.4.3.1	new bitmap 19h
4.3	Progress commands all 3 seconds
	2 byte <info> length
4.4.1	bitmap 03h: s no longer defined
	bitmap 29h no longer defined
4.4.2	bitmap 29h no longer defined
	bitmap 3Ch 31 byte subfile header
4.4.3	bitmap 29h no longer defined
4.4.4	bitmap 29h and 3Ch no longer defined
5	2 byte error codes

2 Communications Interface

The interface of the terminal is specified in detail in order to allow the terminal to be connected to cash register made by various manufacturers.

In accordance with the OSI standardisation by the ISO, the interface is specified in several layers. In this simple application, three layers are sufficient. These are:

- format of character transmission = ISO layer 1
- transport protocol = ISO layer 2
- application protocol = ISO layer 7

2.1 Character Format

The data communication is asynchronous, with 8 data bits, no parity bit and two stop bits. The transmission speed is 9600 baud.

2.2 Transport Protocol

A simple communications protocol is used in accordance with BSC, in the transparent variant. Correspondingly, the application message to be transmitted are embedded in a frame comprising the control character combinations DLE STX and DLE ETX.

To safeguard the transparency of the code, all characters of the message which happen to be equivalent to the DLE bit pattern are transmitted twice. Further control characters from the character set of the BSC procedures which are used for positive or negative acknowledgement are the characters ACK, respectively NAK.

Every message is secured by means of a CRC checksum for which the CCITT polynomial ($x^{16}+x^{12}+x^5+1$) is used. The low byte of the checksum (CRC_LO) is transmitted after the terminating sequence DLE ETX of a message. Afterwards, the high byte (CRC_HI) is transmitted.

The checksum includes all characters which are transmitted after the initial DLE STX, including the closing sequence. **Please note that the DLE characters which are inserted for code transparency are not taken into consideration, in particular the DLE from the closing sequence.**

The format of a transport layer message is therefore:

DLE STX <application message> DLE ETX CRC_LO CRC_HI

It is acknowledged with ACK by the receiving side if a correct checksum was received, otherwise with NAK. **In case of a negative acknowledgement the transmitter repeats the message up to two times.** After that, the transmitter and the receiver report an error status to the higher protocol layer(s).

On the receiving side, the transport protocol works with a two-stage time control. The receiver waits for a message and reports a message time-out to the higher protocol layer if it does not receive starting sequence DLE STX within a certain time span T_1. In case there is a pause greater than T_2 during reception of a message, the receiver sends a negative acknowledgement and waits for the message to be repeated, during which T_1 is monitored again. It is the job of the higher protocol layer to take action to restore the communication in case of an error.

The time-out times are fixed at T_1 = 5 s and T_2 = 0.2 s.

Apart from the transport protocol for application messages specified above, the ICT sometimes transmits the control character ENQ to the cash register. The cash register must respond with ACK and so reports its presence and readiness. After a cash register is recognised and a logon is carried out no more ENQ will be sent.

The control characters are defined as follows:

Control Characters	ASCII - Code
DLE	10h
STX	02h
ETX	03h
ACK	06h
NAK	15h
ENQ	05h

2.3 C Program for Calculating the CRC

16-bit CRC with CCITT polynomial X**16 + X**12 + X**5 + 1.

```
/*=================== Program description =====================
;  unsigned crc16_ctitt ( unsigned char data,unsigned *bcc )
;  The CRC modulo is charged to the fixed CCITT polynomial
;
;                    X**16 + X**12 + X**5 + 1
;
;
;
;                                                  (data) ─────┐
;                                                              │
;                                                              ↓
;   ┌→ │ │ │ │ │ │  →(+)→ │ │ │ │ │ │ │ │  →(+)→ │ │ │ │ │→(+)
;   │                  ↑                         ↑              │
;   └──────────────────┴─────────────────────────┴──────────────┘
;
;
================== End of program description ===============*/

unsigned short crc16_ccitt( unsigned char data, void *bcc ) {

   register unsigned short a, d = *( unsigned short* ) bcc;

   d ^= ( data & 0xff );                                    /* 1 */
   a  = ( d ^ ( d << 4 ) )  & 0xff;                         /* 2 */
   d  = ( d >> 8 ) ^ ( a << 8 ) ^ ( a << 3 ) ^ ( a >> 4 );   /* 3 */
   d &= 0xffff;                        /* only for portability */

   return   *( unsigned short* ) bcc = d;
}
```

Example of a CRC-Calculation:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	CRC_LO	CRC_HI
80h	00h	00h	03h		F5h	1Fh
06h	C0h	00h	03h		9Bh	73h
06h	1Eh	01h	64h	03h	09h	3Ch
06h	C0h	01h	05h	03h	BEh	16h

2.4 Application Protocol

Basis of the application protocol used for communication between ICT and the cash register is the application protocol described in the „Interface Description EC-Terminal ZVT 700 & Cash Register„. However this protocol is adapted and expanded to satisfy the requirements of STARCOIN.

This protocol definition differentiates between commands and responses. For every command sent, there must be a response from the respective unit. All actions are started by commands from the cash register. The ICT can only send commands in reaction to the cash register commands. After a command from the cash register usually the ICT will

send a response back to the cash register followed by a command containing the required data. **It is not permitted, however, for a command to be sent while another command is being processed.** This will be explained in more detail in the description of the commands.

The base element of this protocol is the **APDU** (Application Protocol Data Unit), consisting of a **control field** (CFLD), a **data length field** (DLNG) and a **data block**. Accordingly, an APDU is structured as follows:

APDU			
CFLD		DLNG	Data block
1 byte	1 byte	1 byte	Instruction data

The individual fields of the APDU are described in more detail below.

NOTICE: All values in the APDU are in hexadecimal notation!

2.4.1 Control Field of the Application Protocol

The control field of an **APDU** contains two sub-fields, each of one byte length, which are used in different ways depending on whether a command or a response to a command is involved.

The **CLASS** and **INSTR** sub-fields are used in the case of a command. These together specify a certain application command. **CLASS** specifies an instruction class whereas **INSTR** specifies an instruction within a class. The MSB of **CLASS** is always zero; this is how a command can be unambiguously identified.

C O N T R O L F I E L D	
CLASS	INSTR
0	

Control field of a command

In a response message, the sub-fields are called **CCRC** (Command Class Return Code) and **APRC** (Application Return Code). The MSB of the **CCRC** is always set to one. **CCRC** shows the (global) success of a preceding command, whereas **APRC** offers the possibility of an application-oriented status message.

CONTROL FIELD	
CCRC	APRC
1	

Control field of a response

In this application, the **APRC** field is used for the coded transfer of error conditions. The table of exception conditions given in the GZS specification of the POS terminal (version 3.0 of 24.06.88, pg. 4-33 ff.) is taken, where relevant, as a reference in this case.

At this point, it should be pointed out that there are two equivalent possibilities for signaling a positive result of a command. On the one hand, a **CCRC** with the value **80H** represents a global indication of successful completion; on the other hand, however, it is also possible to refer to a value of the **APRC** using a **CCRC** of **84H**. In this case, an **APRC** of **00H** in turn indicates positive completion.

Both described values of the **CCRC** are furthermore the only values used in the implementation of the **ICT**.

2.4.2 Length Field of the Application Protocol

The length field DLNG contains the length ll for the following data block. Only the protocol variant with a 1-byte length field is supported in the implementation of the terminal; numbers between 0 and 254 are permitted in this length field. The length of 255 may not be used because, in the original protocol, it marks an extension of the length field to two bytes. This extension of the protocol, however, is presently not implemented.

2.4.3 Data Block of the Application Message

The data block contains the application-dependent information to be transmitted. For some commands the data field is subdivided in two one byte parameter fields (PAR1 and PAR2) and the real data field of variable length (see 3.6 Payment Transaction). The length results from the preceding length field of the message. The structure of the data block is specified in the descriptions of the individual application messages.

2.4.4 Instruction Classes and Coding

The instruction classes for the general case are defined in the table below. Not all of these, however, are relevant for the application protocol to be specified here. They are used hereafter with the mnemonic terms listed here.

Name	Meaning	Code
DIAG	Diagnostic class	01
CNTR	Control class	02
STAT	Status class	03
WRTE	Write class	04
READ	Read Class	05
EXEC	Run class	06
AUTO	Authentication class	07
ADMI	Administration class	08

The semantic meaning of the instruction classes can be deduced from the term. The assignment of specific application functions to instruction classes shall not however be discussed further here. As far as it is practical, it is modelled on the stipulations of the GZS specification.

2.4.5 Coding of the Response Messages

The response messages are generally coded in two variants. If the **CCRC** field contains the value **80H**, this generally means a positive completion of the basic command. In this case, the **APRC** field is not relevant and is generally set to the value 00H. The data field can then contain further command-specific data.

Receiver ——— Transmitter			
CCRC	APRC	DLNG	Data
80	xx	*ll*	**.....**

When the completion of the command is negative, a response with a **CCRC of 84H** is sent instead; this means that the following **APRC** is valid. In this case, an error ID is set in the **APRC**. To a certain extent, error ID **00H** represents a special case. It signals a successful completion, therefore this case can be coded in two different ways! This response message can also contain a command-specific and/or error-specific data field.

Receiver ——— Transmitter			
CCRC	APRC	DLNG	Data
84	Error ID	*ll*	**.....**

2.4.6 Sequence of a Command

The basic sequence of a command, which always has to be started from the cash register,

is:

Cash register	———	ICT	Command from the cash register
Cash register	———	ICT	Acknowledgment from ICT to correct receipt of the command

Depending on the command from the cash register, the ICT can now send commands which are confirmed by an acknowledgment from the cash register:

Cash register	———	ICT	Command from ICT
Cash register	———	ICT	Acknowledgment from the cash register to correct receipt of the command

Finally, the ICT sends the closing or the cancel command which the cash register confirms again.

Cash register	———	ICT	Closing command from the ICT
Cash register	———	ICT	Acknowledgment from the cash register to correct receipt

The initiative is now transferred to the cash register again (basic status).

2.5 Example for a communication

This is an example of a communication between ICT and a cash register. The cash registers transmitted („t„) and received („r„) bytes are all in hexadecimal notation.

Cash registers logon at ICT with password „123456„.	t10 t02 t06 t00 t0C t12 t34 t56 t00 t0C t98 t12 t24 tAA t22 t10 t05 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
ICT indicates a successful completion of the command	r10 r02 r06 r0F r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
Cash register checks terminal software version	t10 t02 t05 t04 t00 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
ICT transmits software version	r10 r02 r06 r0F r03 r01 r01 r01 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register starts payment transaction with amount 0,01 and	t10 t02 t06 tA0 t0A t01 t01 t04 t00 t00 t00 t00 t00 t01 t19 tE1 t10 t03 txx txx r06

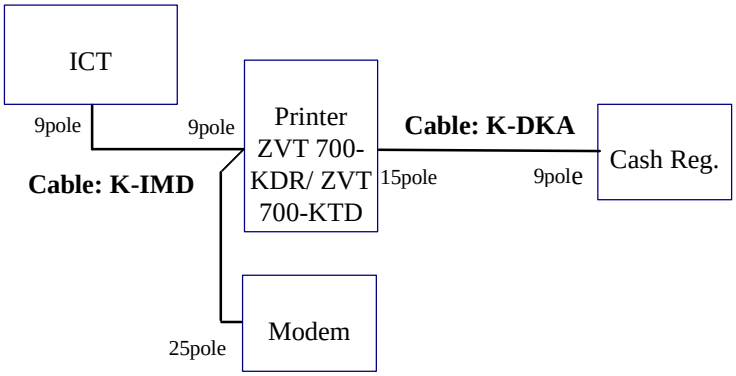
proposes an electronic cheque	
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
positive termination of payment transaction command with info „wait for card„	r10 r02 r06 r0F r02 r0A r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register asks for terminal status	t10 t02 t06 tA0 t02 t01 t02 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal transmits status „wait for card„	r10 r02 r06 r0F r02 r0A r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register asks for terminal status	t10 t02 t06 tA0 t02 t01 t02 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal transmits status „wait for card„	r10 r02 r06 r0F r02 r0A r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register asks for terminal status	t10 t02 t06 tA0 t02 t01 t02 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal transmits status „card inserted„	r10 r02 r06 r0F r02 r64 r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register starts transaction	t10 t02 t06 tA0 t02 t01 t03 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal concludes transaction with error message „D1 01„ „correction transaction required„, with the transaction data and with the information that it was a purse transaction	r10 r02 r06 r1E r35 rD1 r01 rA7 r45 r0A r00 r01 rFF r00 r00 r00 r02 r28 r02 r55 r03 r01 r00 r00 r9C r51 r00 r00 r00 r01 r00 rB3 r00 r00 r00 r13 r00 r00 r00 r07 r00 r00 r96 r20 r50 r2D r90 r00 r00 r00 r00 r00 r14 r82 r24 r88 r19 rE0 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register initiates correction transaction from purse	t10 t02 t06 tA0 t06 t01 t04 t10 t03 txx txx r06
ICT acknowledges the correct	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06

receipt of the command	
positive termination with info „wait for card„	r10 r02 r06 r0F r02 r0A r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register asks for terminal status	t10 t02 t06 tA0 t02 t01 t02 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal transmits status „card inserted„	r10 r02 r06 r0F r02 r64 r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register starts transaction	t10 t02 t06 tA0 t02 t01 t03 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal concludes transaction	r10 r02 r06 r0F r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register starts transfer to cash register with password 00 00 00	t10 t02 t06 t52 t04 t00 t00 t00 t01 t10 t03 txx txx r06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal transmits startup dialog to cash register	r10 r02 r04 rFE r07 r03 r81 r01 r55 r3C r00 r01 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits subfile header to cash register	r10 r02 r04 rFE r26 r03 r83 r00 r01 r3C r41 r05 r01 r00 r07 r00 r00 r00 r02 r09 r10 r00 r03 r00 r00 r00 r19 r28 r02 r55 r00 r98 r11 r14 r16 r03 r01 r01 r02 r00 r0F r00 r04 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits transaction record to cash register	r10 r02 r04 rFE r38 r03 r80 r00 r02 r3C rF0 rF4 rF8 r45 r0A r00 r01 rFF r00 r00 r00 r02 r28 r02 r55 r03 r01 r00 r00 r9C r51 r00 r00 r00 r01 r00 rB3 r00 r00 r00 r13 r00 r00 r00 r07 r00 r00 r96 r20 r50 r2D r90 r00 r00 r00 r00 r00 r14 r82 r24 r88 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits correction record to cash register	r10 r02 r04 rFE r3C r03 r80 r00 r03 r3C rF0 rF5 rF2 r4B r00 r00 r01 rFF r00 r00 r00 r02 r28 r02 r55 r03 r01 r00 r00 r9C r51 r00 r00 r00 r01 r00 rB4 r00 r00 r00 r13 r00 r00 r00 07 r00 r01 r96 r20 r90 r00 r90 r00 r14 r82 r24 r88 rC6 rA0 rFE r5D r22 r0B r73 rCB r10 r03 rxx rxx t06

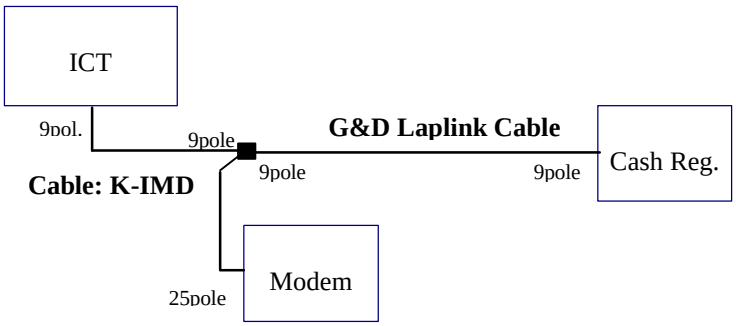
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits sum set record to cash register	r10 r02 r04 rFE r21 r03 r80 r00 r04 r3C rF0 rF2 rF5 r4D r00 r07 r00 r00 r00 r08 r00 r01 r00 r00 r00 r11 r02 r01 r00 r05 r00 r00 r00 r03 r99 r91 rDC rB1 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits monitoring record to cash register	r10 r02 r04 rFE r4D r03 r80 r00 r05 r3C rF0 rF6 rF9 r53 r00 r07 r00 r00 r00 r02 r09 r10 r00 r03 r00 r01 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r03 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r01 r00 r00 r00 r00 r00 r01 r00 r03 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits end record to cash register	r10 r02 r04 rFE r0B r03 r80 r00 r06 r3C rF0 rF0 rF3 r5A r00 r04 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal transmits end dialog to cash register	r10 r02 r04 rFE r04 r03 r82 r00 r06 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
terminal concludes transfer	r10 r02 r06 r0F r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06
cash register logs off	r10 r02 r06 r02 r00 r10 r03 rxx rxx t06
ICT acknowledges the correct receipt of the command	r10 r02 r80 r00 r00 r10 r03 rxx rxx t06
terminal confirms logoff	r10 r02 r06 r0F r00 r10 r03 rxx rxx t06
Cash register confirms the correct receipt of the command	t10 t02 t80 t00 t00 t10 t03 txx txx r06

2.6 Connecting cash register and ICT

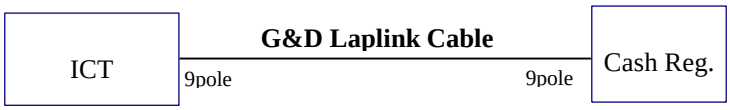
2.6.1 With printer ZVT 700-KDR/ZVT 700-KTD and Modem



2.6.2 With Modem



2.6.3 Direct connection



2.6.4 Article Numbers of the Cables

Name	Article No.	Length	Pin definition	Info
G&D Laplink Cable	184 336 000	2 m	9 pole female SUB-D plug ⇔ 9 pole fem. SUB-D plug	ICT ⇔ Cash Register
K-IMD	184 510 001	2 + 1 m	9 pole female SUB-D plug ⇔ 9 pole male SUB-D plug ⇔ 25 pole male SUB-D plug	ICT ⇔ Printer KDR/KTD ⇔ Modem
K-DKA	184 511 001	1 m	15 pol. SUB-D female ⇔ 9 pol. SUB-D female	KDR/KTD ⇔ Cash Register

3 Commands to the ICT

The following section describes the application commands which the ICT can carry out. These commands are sent from the cash register to the ICT. The ICT carries out the command and then sends an appropriate command to the cash register.

The abbreviations listed below are used to mark the data fields in the individual commands:

AC	Response code
PW	Password
NPW	New password
(X)	Length of the field in bytes

The length of the data fields in bytes is often indicated in brackets. In stand-alone mode, the passwords (PW, NPW) correspond to the retailer password (01) which occupies the highest position in the hierarchy. An initial value is assigned when the unit is manufactured and is issued at delivery.

Bitmap positions are defined for most of the data fields used, as specified in the chapter dealing with the telecommunication interface. These data fields are used with the notation 'BMP <data>' in the internal interface, where BMP is the bitmap position in binary representation and the <data> part contains the user data as well as the prefixed length information when the fields have differing lengths (see the Appendix for the structure of the length information). The meaning of some BMPs is slightly modified in the internal interface. These are described below:

Symbol	EBCDIC-Code	Length	example
LLVAR	FxFy	00 = xy = 99	32 = „F3 F2„
LLLVAR	FxFyFz	000 = xy = 999	142 = „F1 F4 F2„

3.1 'Logon' command

The ICT accepts the rest of the commands after it has received this command. This command must be called up at least once after switching on, in order to allow the unit to be used. Without a logon the terminal answers all other commands with an abort command. The ICT accepts commands from the cash register only if the cash register is selected in the ICT menu (see user manual ICT 800 for further information).

Cash register ——————————→ ICT			
CLA	INS	DLNG	Data
EXEC	00h	ll	PW(2) <config>

Cash register ←—————————— ICT		
CCRC	APRC	DLNG
80	xx	00

The configuration byte <config> is reserved for future use (rfu).

After this command the cash register waits for a response from the ICT to indicate correct receipt of the command. Then it waits for the ICT to give either a positive termination command without data or an abort command.

If the Logon succeeded a cash register symbol is displayed together with the STARCOIN symbol.

3.2 ‘Logoff’ command

After this command the ICT is logged off from the cash register. Only the Logon command is accepted after this command.

Cash register ——————————→ ICT			
CLA	INS	DLNG	Data
EXEC	02h	00h	

Cash register ←—————————— ICT		
CCRC	APRC	DLNG
80	xx	00

After this command the cash register awaits for a response from the ICT for correct reception of the command, then either a positive termination command without data or an abort command.

3.3 ‘Software Version’ command

With this command the cash register checks the software version of the ICT

Cash register ——————————→ ICT			
CLA	INS	DLNG	Data
READ	04h	00h	

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

The ICT concludes this command with a Positive Termination command including the software version or an Abort command.

Cash register ←———————— ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	03h	<3 Byte SW - Version>

Cash register ————————→ ICT		
CCRC	APRC	DLNG
80	xx	00

The structure of the software version is:

first byte	second byte		third byte
	first nibble	second nibble	
manufacturer ID	hardware version	software version	

3.4 Set Date/Time

The „Set Date/Time„ function can be used for setting the internal real-time clock of the ICT.

Cash register ————————→ ICT			
CLA	INS	DLNG	Data
EXEC	91h	0Bh	PW(2) AA yy mm dd 0C hh mm ss

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

In the positive case, the ICT terminates this function with the „Positive Termination„ command in which no data are transferred.

In the negative case, it transfers the „Abort„ command.

3.5 Change Password

The cash register password set on the ICT can be changed with the „Change Password„ command.

Cash register ——————→ ICT			
CLA	INS	DLNG	Data
EXEC	95h	06h	PW(2) NPW(2)

Cash register ←—————— ICT		
CCRC	APRC	DLNG
80	xx	00

The passwords (PW and NPW) are BCD-coded. The ICT terminates this function with the „Positive Termination„ command in which no data are transferred.

3.6 Payment Transaction

In an extension of the original protocol some commands use the first two bytes of the data field as parameter bytes (see 0 2.4.3 Data Block of the Application Message).

CLA	INS	DLNG	DATA		
			PAR1	PAR2	DATA

The commands of the payment transaction use this extension. This commands have all the same CLASS (EXEC) and the same INSTRUCTION (0xA0). They differ from each other only by the parameter bytes.

3.6.1 'Transaction Amount' command

With this command the cash register initialises a payment transaction at the terminal. It must already transmit the transaction amount in the data field. The amount is transmitted as 6 byte BCD coded in minor units of the currancy with leading zeros. The prefix 04h indicates the ISO bitmap position.

Cash register ——————→ ICT					
CLA	INS	DLNG	PAR1	PAR2	Data
EXEC	A0h	09h	01h	01h	04h Amount(6) [<optional information>]

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

After this command the cash register awaits a Positive Termination command with a two byte data field or an Abort command.

Cash register ←———————— ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	02h	<info>

Cash register ————————→ ICT		
CCRC	APRC	DLNG
80	xx	00

The data field of the Positive Termination command uses the same <info> bytes.

The following values are currently defined for <info>:

<info>	Bedeutung
07h00h	wrong card inserted
08h00h	unknown card inserted
09h00h	inserted card exceeded
0Ah00h	waiting for card
0Ch00h	card inserted wrongly or unknown card
0Eh00h	ICT proceeds transaction
64h00h	card inserted
65h00h	ICT displays balance
66h00h	ICT performs transfer
67h00h	ICT is dialling
68h00h	ICT transmits data
6Dh00h	ICT checks new terminal card

3.6.1.1 <optional information>

The data field of the optional information starts with a characteristic ISO bitmap position and the data with the described format.

CCh[1]	<application>	LLVAR (2Bytes). The cash register determines all allowed applications for the payment transaction. For each allowed

[1] The bitmap position 'CCh' for the <application> and the definied application-IDs for the applications are preliminary and possibly will be changed

		application a corresponding application-ID is transmitted. **If no 'CCh' is transmitted all applications are allowed.**
19h	<proposed application>	1 Byte length. Proposes **one** application for the payment transaction by using the same applicationIDs than <application>. If possible the terminal also proposes this application for the payment transaction but the customer is allowed to select another application.

The possible applications for a payment tranaction are:

application-ID	application
E0h	purse
E1h	electronic cheque
E2h	offline POS
E4h	online POS

3.6.2 'Status Request' command

After the cash register has sent the Transaction Amount command or a Correction Transaction command and received a Positive Termination command from the terminal, it periodically sends the Status Request command until the ICT is ready to start the transaction.

Cash register ——————→ ICT					
CLA	INS	DLNG	PAR1	PAR2	Data
EXEC	A0h	02h	01h	02h	

Cash register ←—————— ICT		
CCRC	APRC	DLNG
80	xx	00

After this command the cash register awaits a Positive Termination command with a two byte data field or an Abort command. The data field of the Positive Termination command uses <info> bytes.

Cash register ←———————— ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	02h	<info>

Cash register ————————→ ICT		
CCRC	APRC	DLNG
80	xx	00

The following values are currently defined for <info>:

<info>	Bedeutung
07h00h	wrong card inserted
08h00h	unknown card inserted
09h00h	inserted card exceeded
0Ah00h	waiting for card
0Ch00h	card inserted wrongly or unknown card
0Eh00h	ICT proceeds transaction
64h00h	card inserted
65h00h	ICT displays balance
66h00h	ICT performs transfer
67h00h	ICT is dialling
68h00h	ICT transmits data
6Dh00h	ICT checks new terminal card

3.6.3 'Start Transaction' command

If the cash register receives a Status Information command with the status message „card inserted„ it can start the payment transaction by sending the Start Transaction command. After this command the payment transaction may not be aborted any more.

Cash register ————————→ ICT					
CLA	INS	DLNG	PAR1	PAR2	Data
EXEC	A0h	02h	01h	03h	

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

After receiving this command the ICT will send a response to the cash register. When the payment transaction has finished and the ICT concludes the Start Transaction command with a Positive Termination command including the transaction subfile in the data field. If a Correction Transaction is performed the ICT can conclude the command with a Positive Termination command without data. **In this case the payment transaction was**

not successfully finished. The ICT can also abort the transaction, e.g. because the customer card has been removed. In this case the data field of the Abort command can contain optional transaction data (see 0 4.2 'Abort' command). If the Error-ID of the Abort command is 'D1h' '01h' („Correction transaction required„) the cash register has to perform a correction transaction. **If no Correction Transaction will be performed a loss of money is possible**.

Cash register ←———————— ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	ll	<transaction data>

Cash register ————————→ ICT		
CCRC	APRC	DLNG
80	xx	00

3.6.3.1 <transaction data>

The data field of the transaction data starts with a characteristic ISO bitmap position and the transaction data with the described format.

A7h	<single/correction transaction>	LLVAR (2Bytes) the structure is described in the „Requirements Specifications STARCOIN„ chapter „Single Transaction Record„ and chapter „Correction Transaction Record„
19h	<application>	1Byte length. The terminal transmits the application-ID of the application used for the payment transaction.

3.6.4 'Correction transaction' command

This command should prevent a loss of money if a payment transaction wasn't finished successfully.

After this command the cash register has to proceed in the same way it proceeds after the Transaction Amount command.

Cash register ————————→ ICT					
CLA	INS	DLNG	PAR1	PAR2	Data
EXEC	A0h	02h	01h	04h	

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

In the case of a correction transaction the whole payment transaction was successfully finished only if the Positive Termination command after the Start Transaction command includes transaction data. If the correction transaction was not necessary, the terminal concludes this command also with a Positive Termination but without transaction data. In this case the payment transaction was not succesfully finished. No money were paid!!

3.7 'Abort' command

This command aborts a transaction. Currently it can only be used to abort a payment (correction) transaction as long as no Start Transaction command is being sent.

Cash register ————————→ ICT			
CLA	INS	DLNG	Data
EXEC	B0h	00h	

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

The ICT concludes this command with a Positive Termination command without data or an Abort command.

3.8 'Transfer' command

This command causes the terminal to perform a transfer and depending on the <config>byte to transmit the transfer records to the cash register, to a transfer card or online to the C&A system. The current version supports the transmitting to the cash register only.

Cash register ————————→ ICT			
CLA	INS	DLNG	Data
EXEC	52h	04h	PW(2) <config>

Cash register ←———————— ICT		
CCRC	APRC	DLNG
80	xx	00

config byte	Discription
0000 0001	transfer records to cash register
0000 0010	rfu (transfer to transfer card)
0000 0100	rfu (on-line transfer to C&A system)
0000 1000	rfu
0001 0000	rfu
0010 0000	rfu
0100 0000	rfu
1000 0000	rfu

Depending on the config byte the ICT answers this command with Transfer Data commands until all transfer data has been and concludes the Start Transaction command with a Positive Termination without data.

4 Commands from the ICT

4.1 'Positive Termination' command

This positive termination message is used by the terminal to conclude processing of a job received from the cash register. If need be, data for the current transaction are transmitted to the cash register at the same time. This involves either data for the current transaction or the transaction sums which are calculated from the internal transaction file of the terminal.

Cash register ← ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	ll	<transaction data>

Cash register → ICT		
CCRC	APRC	DLNG
80	xx	00

The possible variants of the <transaction data> are described in more detail with the individual commands from the cash register.

4.2 'Abort' command

With this command, the ICT concludes processing of a job which was received from the cash register and which could not be completed because an error occurred. The condition which led to the abort is coded in the transmitted error ID (see 0 5 Error Codes). After that, the cash register can perform an appropriate error routine. In case of an aborted payment transaction after the Start Transaction command was transmitted, the data field of the Abort command optionally contains transaction data.

Cash register ← ICT			
CLA	INS	DLNG	Data
EXEC	1Eh	02h	<error-ID>[<transaction data>]

Cash register → ICT		
CCRC	APRC	DLNG
80	xx	00

The <transaction data> is described in more detail with the Start Transaction commands from the cash register.

4.3 'Transfer Data' command

With this command the ICT sends the transfer data to the cash register.

This command doesn't conclude a transaction.

Cash register ←——————— ICT			
CLA	INS	DLNG	Data
WRTE	FEh	ll	<dialog data>

Cash register ———————→ ICT		
CCRC	APRC	DLNG
80	xx	00

After the Transfer command the ICT transmits transfer subfiles which are still untransferred or which must be retransferred. The communication starts with a „Startup dialog„. Afterwards the ICT sends a „subfile header„ and one or more „Records„ for each transfer subfile. At the end the ICT transmits a „end dialog„.

4.3.1 Startup dialog

03h	<tag>	N(6) 3 byte BCD, 81sLLL LLL: maximum length of dialog data field (3Ch), BCD s: rfu
3Ch	<special dialog data>	LLLVAR: nnnn 2 byte BCD, number of subfiles

4.3.2 Subfile header

03h	<tag>	N(6) 3 byte BCD 83nnnn nnnn: BCD coded sequential block number, starting with 0001
3Ch	<special dialog data>	31 byte subfile header 2 byte BCD number of records of this subfile

4.3.3 Records

03h	<tag>	N(6) 3 byte BCD 80nnnn nnnn: BCD coded sequential block number, starting with 0001
3Ch	<special dialog data>	LLLVAR: This field contains the records of a transfer subfile (single transaction record, correction record, update confirmation, monitoring data record, pool sum record, end record). See „Requirements Specifications STARCOIN„ Version 1.1/02.06.1998, chapter 3.3.2.1 „Transfer Subfile„

4.3.4 End Dialog

03h	<tag>	N(6) 3 byte BCD, 82nnnn nnnn: number of the last transmitted block, BCD coded

5 Error Codes

Basically, the error codes of the ISO 8583 standard and the GZS specification are used wherever it makes sense to do so. Because of the difference in implementation, however, only a subset of the codes provided by the GZS is used.

The following error codes are used as data field of an abort message. They are always binary coded.

Error code decimal	Error code hexadecimal		Meaning
	1 byte	2 byte	
4-00	04	00	Card not valid
13-00	0D	00	Available amount exceeded
21-00	15	00	Original authorisation not found
25-00	19	00	No transaction amounts
30-00	1E	00	Request format incorrect
31-00	1F	00	Card issuer not authorised
33-00	21	00	Card expired
34-00	22	00	Suspected manipulation
54-00	36	00	Card expired
55-00	37	00	Code number incorrect
56-00	38	00	Card invalid
57-00	39	00	Incorrect card for cancellation
58-00	3A	00	Terminal not initialised
61-00	3D	00	Withdrawal limit exceeded
62-00	3E	00	Card frozen
63-00	3F	00	Breach of security
65-00	41	00	Maximum number of payments reached
68-00	44	00	No answer from system
75-00	48	00	PIN counter exceeded
76-00	49	00	Key index incorrect
81-00	51	00	Initialisation error, repetition needed
82-00	52	00	Network operator’s terminal blocked
83-00	53	00	Change of terminal not permitted
86-00	56	00	Master records missing
87-00	57	00	Unknown terminal
89-00	59	00	CRC sum incorrect
97-00	61	00	incorrect MAC

98-00	62	00	Date/time incorrect
99-00	63	00	PAC incorrect
100-00	64	00	Read error
101-00	65	00	Cannot process card data
102-00	66	00	Processing error
105-00	69	00	Password incorrect
106-00	6A	00	Transaction memory full
107-00	6B	00	Function deactivated
108-00	6C	00	Abort by time-out or ABORT key
109-00	6D	00	Read error or incorrect card position / chipcard error
111-00	6F	00	Data format incorrect
112-00	70	00	ICT processing a power-off cancellation
113-00	71	00	Cashcard credit not sufficient
114-00	72	00	Chip fault, not readable
119-00	77	00	Transaction file closure impossible, try again later
125-00	7D	00	Communication error with coprocessor
154-00	9A	00	Protocol error
155-00	9B	00	Disruption of communication
156-00	9C	00	Please wait
200-00	C8	00	Card removed, transaction aborted
201-00	C9	00	Amount to high
202-00	CA	00	Balance to low
203-00	CB	00	Terminal card not yet active
204-00	CC	00	Terminal card expired
205-00	CD	00	Customer card not yet active
207-00	CF	00	Card error
209-00	D1	00	System error
209-01	D1	01	Correction transaction required
210-00	D2	00	System error, call card provider
211-00	D3	00	No transfer data
214-00	D6	00	Terminal card error
215-00	D7	00	No response from terminal card
217-00	D9	00	Payment not possible

6 Table of the Commands

Commands from the cash register to the terminal				
CLA	**INS**	**P1**	**P2**	**Description**
EXEC	00			Logon of the cash register at the terminal
EXEC	02			Logoff of the cash register at the terminal
READ	04			Checks the terminal software version
EXEC	07	01	01	Initialisation of a payment transaction
EXEC	07	01	02	Request terminal status
EXEC	07	01	03	Start the payment transaction
EXEC	07	01	04	Initialisation of a correction transaction
EXEC	52			Starts a transfer at the terminal
EXEC	91			Set date and time of the terminal
EXEC	95			Change password of terminal
EXEC	B0			Abort

Commands from the terminal to the cash register				
CLA	**INS**	**P1**	**P2**	**Description**
EXEC	0F			Positive termination of a command
EXEC	1E			Negative termination of a command
WRTE	FE			Transfer data to cash register