Specification STARCOIN Cash Register Interface

Version 0.0.4/Revision 08.06.98

Author hrd - 3FE3
Status PRELIMINARY/CONFIDENTIAL

Version 0.0.4/Revision 08.06.98

R&D

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 (also called „retailer unit“). 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 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.

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;
}
```

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.

In this protocol only commands are defined and all actions are started by commands from the cash register. The ICT can only react on cash register commands. **The cash register is always master and the ICT is always slave.** After a command from the cash register usually the ICT will send one command containing the required data back to the cash register. In some cases when the processing of a command takes more than five seconds additional status messages being sent periodically to the cash register until the command is being processed. **It is not permitted, however, for a command to be sent while another instruction 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:

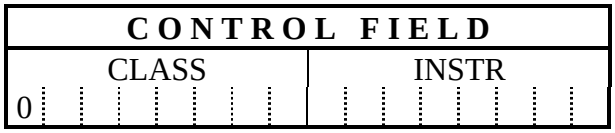
APDU			
CFLD		DLNG	Data block
1 byte	1 byte	1 byte	Instruction data

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**, the **CLASS and INSTR sub-field.**. 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 because the original protocol differs with this bit between commands and responses.



Control field of a command

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 3.4 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 Course of a command

The course of a command, which is always started from the cash register is:

cash reg ⟶ ICT command from the cash register
[cash reg ⟵ ICT optional status messages from ICT]
cash reg ⟵ ICT command from ICT containing the response data

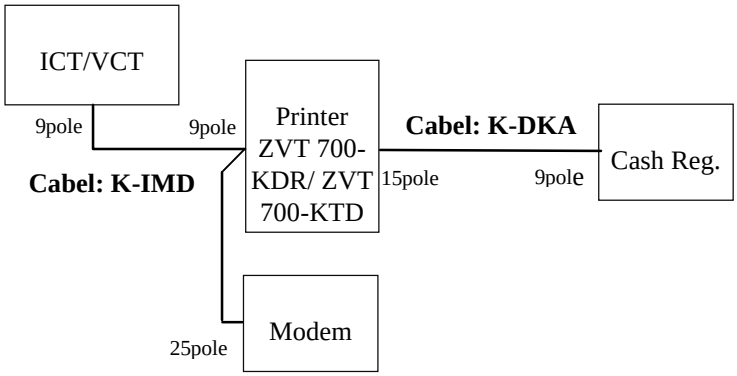
2.5 Example for a communication

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.

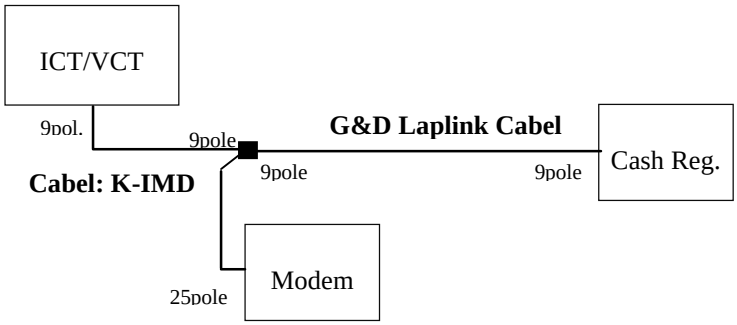
Cash registers logon at ICT with password „123456“.	`t10 t02 t06 t00 t0C t12 t34 t56 t00 t0C t98 t12 t24 tAA t22 t10 t05 t10 t03 txx txx r06`
ICT acknowledges the correct receipt of the command	`r10 r02 r06 r0F r00 r10 r03 rxx rxx t06`

2.6 Connecting cash register and ICT

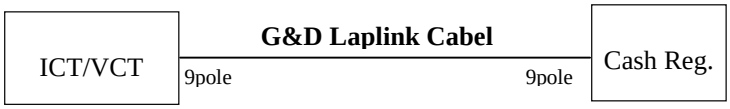
2.6.1 With printer ZVT 700-KDR/ZVT 700-KTD and Modem



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. 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(3) <config> [AA yy mm dd] [0C hh mm ss]

The configuration byte <config> is reserved for future use (rfu). The date, starting with tag 'AA' and the time starting with tag '0C' are optional and may be on various positions behind the password and the configuration byte.

After this command the cash register awaits 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	

After this command the cash register awaits 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	

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	<SW - ICT>

3.4 Payment Transaction

In an extension of the original protocol some commands use the first two bytes of the data field as parameter bytes (see 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.

3.4.1 '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 a 4 byte unsigned long 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	07h	01h	01h	04h Amount(4)

After this command the cash register awaits a Positive Termination command with a one byte data field or an Abort command. The data field of the Positive Termination command uses the same <info> byte than the Progress command (see 4.3 'Progress).

Cash register ←—————————— ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	01h	<info>

3.4.2 '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	

After this command the cash register awaits a Positive Termination command with a one byte data field or an Abort command. The data field of the Positive Termination command uses the same <info> byte than the Progress command (see 4.3 'Progress).

Cash register ←—————————— ICT			
CLA	INS	DLNG	Data
EXEC	0Fh	01h	<info>

3.4.3 '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	

After receiving this command the ICT will send Progress commands periodically every five seconds until 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 4.2 'Abort' command). If the Error-ID of the Abort command is DCh („Correction transaction required") the cash register has to perform a correction transaction. **If no Correction Transaction is performed a loss of money is possible**.

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

3.4.3.1 <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“ Version 1.1/02.06.1998, chapter 3.3.2.1.3 „Single Transaction Record“ and chapter 3.3.2.1.4 „Correction Transaction Record“

3.4.4 ‘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	

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.

3.5 ‘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	

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

3.6 '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(3) <config>

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 transmitted or with Progress commands until the transfer has been finished 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>

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 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	01h	<error-ID>[<transaction data>]

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

4.3 'Progress' command

During the proceeding of a transaction the ICT transmits information about the progress of this transaction periodically all 5 seconds. If a command requires less than 5 seconds until it is finished, this command isn't send.

This command doesn't conclude a transaction.

Cash register ←——————— ICT			
CLA	INS	DLNG	Data
WRTE	FFh	01h	<info>

The following values are currently defined for <info>:

<info> in		Bedeutung
dezimal	hex	
07	07h	wrong card inserted
08	08h	unknown card inserted
09	09h	inserted card exceeded
10	0Ah	waiting for card
12	0Ch	card inserted wrongly or unknown card
14	0Eh	ICT proceeds transaction
100	64h	card inserted
101	65h	ICT displays balance
102	66h	ICT performs transfer
103	67h	ICT is dialling
104	68h	ICT transmits data
109	6Dh	ICT checks new terminal card

4.4 '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>

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".

4.4.1 Startup dialog

03h	<tag>	N(6) 3 byte BCD, 81sLLL LLL: maximum length of dialog data field (3Ch), BCD s: bit 0: splitting; transfer data is splitted to several messages bit 1-7; rfu
29h	<terminal ID>	4 Byte BCD
3Ch	<special dialog data>	LLLVAR: nnnn 2 byte BCD, number of subfiles

4.4.2 Subfile header

03h	<tag>	N(6) 3 byte BCD 83nnnn nnnn: BCD coded sequential block number, starting with 0001
29h	<terminal ID>	4 Byte BCD
3Ch	<special dialog data>	36 byte subfile header 2 byte BCD number of records of this subfile

4.4.3 Records

03h	<tag>	N(6) 3 byte BCD 80nnnn nnnn: BCD coded sequential block number, starting with 0001
29h	<terminal ID>	4 Byte BCD
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“

4.4.4 End Dialog

03h	<tag>	N(6) 3 byte BCD, 82nnnn nnnn: number of the last transmitted block, BCD coded
29h	<terminal ID>	4 Byte BCD
3Ch	<special dialog data>	LLLVAR: nnnn 2 byte BCD, number of subfiles

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	Where did it occur	Meaning
4	AS/FEP	Card not valid
13	AS/FEP	Available amount exceeded
21	AS/FEP	Original authorisation not found
25	Internal	No transaction amounts
30	AS/FEP	Request format incorrect
31	AS/FEP	Card issuer not authorised
33	AS/FEP	Card expired
34	AS/FEP	Suspected manipulation
54	AS/FEP	Card expired
55	AS/FEP	Code number incorrect
56	AS/FEP	Card invalid
57	AS/FEP	Incorrect card for cancellation
58	FEP	Terminal not initialised
61	AS/FEP	Withdrawal limit exceeded
62	AS/FEP	Card frozen
63	AS/FEP	Breach of security
65	AS/FEP	Maximum number of payments reached
68	Int./FEP	No answer from system
75	AS/FEP	PIN counter exceeded
76	AS/FEP	Key index incorrect
81	FEP	Initialisation error, repetition needed
82	FEP	Network operator's terminal blocked
83	FEP	Change of terminal not permitted
86	FEP	Master records missing
87	FEP	Unknown terminal
89	FEP	CRC sum incorrect

97	AS/FEP	incorrect MAC
98	AS/FEP	Date/time incorrect
99	AS/FEP	PAC incorrect
100	Internal	Read error
101	Internal	Cannot process card data
102	Internal	Processing error
105	Internal	Password incorrect
106	Internal	Transaction memory full
107	Internal	Function deactivated
108	Internal	Abort by time-out or ABORT key
109	Internal	Read error or incorrect card position / chipcard error
111	Internal	Data format incorrect
112	Internal	ICT processing a power-off cancellation
113	Internal	Cashcard credit not sufficient
114	Internal	Chip fault, not readable
119	Internal	Transaction file closure impossible, try again later
125	Internal	Communication error with coprocessor
154	Internal	Protocol error
155	Internal	Disruption of communication
156	Internal	Please wait
200	ICT	Card removed, transaction aborted
201	ICT	Amount to high
202	ICT	Balance to low
203	ICT	Terminal card not yet active
204	ICT	Terminal card expired
205	ICT	Customer card not yet active
207	ICT	Card error
209	ICT	System error
210	ICT	System error, call card provider
211	ICT	No transfer data
214	ICT	Terminal card error
215	ICT	No response from terminal card
217	ICT	Payment not possible
220	ICT	Correction transaction required

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	B0			Abort
EXEC	52			Starts a transfer at the terminal

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
WRTE	FF			Status information about the current transaction