ECR-Interface
ZVT-Protocol

Manufacturer-independent procotol between
payment terminals and electronic cash-register systems/vending machines

Transport-Protocol
Application-Protocol

Distributor: www.terminalhersteller.de

Disclaimer

The following information is based on the current state of knowledge and is provided without guarantee. Modifications and errors excepted.

Revision	04
Date	13th July 2016
Status	released

Revision

Revision	Date	Reason for Revision	Modified by
01	5.3.2004	Separation of document into two documents Transport-Protocol/Application-Protocol and Commands/Bitmaps/Error-Messages Changes Chap.3.2.6 Time-out T4: - Communication Dial-up over ECR also triggers T4	K. Höflich
02	15.3.2007	Additions to interfaces Ethernet, USB and Transport-protocol TCP/IP	
02	10.07.07	English translation	S. Atherton
03	16.04.2008	Change copyright	T. Lilienthal
04	13.07.2016	Reference to trace emaples removed	D. Ferlings

1 Definitions

All numerical values, unless otherwise specified, are **hexadecimal**.

1.1 Terms and Abbreviations

Term	Definition
APDU	Application Protocol Data Unit (= a complete request or response)
BMP	bitmap, pre-defined data field
CC	Currency Code, 09 78 = Euro
ECR	Electronic Cash Register. System that transmits the amount to the payment terminal. May also be a vending-machine.
PS	Personalisation System (= Host for OPT-Action)
PT	Payment Terminal (table-top or integrated unit)
RC	Return-Code
TID	Terminal-ID, 8 character numerical
TKS	Terminal Configuration Server (= Server responsible for software-updates etc.)
xx	any value/undefined/dependent on the data
ZVT	Zahlungsverkehrterminal (= Point-Of-Sale Terminal)
<Feld>	A parameter shown in angled-brackets is a place-holder.The place-holder is explained in the following text
[<Feld>]	A parameter shown in square-brackets is optional

1.2 Special Characters in Transfer-Protocol

Character	Transmited Value	Definition
DLE	10	data line escape
STX	02	start of text
ETX	03	end of text
ACK	06	acknowledged
NAK	15	not acknowledged
CR	0D	carriage return
LF	0A	line feed

2 Hardware-Interfaces

The following interfaces are possible for the interface between ECR and PT:

1. RS232
2. Ethernet
3. USB

2.1 RS232

The data-transfer is carried-out according to RS232C with the following parameters:

- 9600 (decimal) Baud, optional 115200 (decimal) Baud, asynchronous
- no handshake
- 8 data-bits
- no parity-bit
- 2 Stop-bits (transmit/receive; some terminals also accept data-packets with 1 stop-bit to increase compatibility, this is however not mandatory)

As transport-protocol, the serial protocol is used.
Definition of the hardware connections is not a part of this specification, and is to be determined between ECR-manufacturer and terminal-manufacturer.

2.2 Ethernet

The transmission is carried-out according to IEEE 802.3. All conventional media-types, i.e. 10Base-T or 100Base-T can be used.

TCP/IP is used as transport-protocol with the following provisions:

- Default port: 20007; the port must be changeable via configuration in the ECR and terminal.
- The connection is opened and closed by the ECR. All messages from the terminal (i.e. Intermediate-Status, Status-Information etc.) are sent within the same connection.
- Special case: for service-functions the terminal can send autonomously to the ECR without needing to have master-right, when agreed with the ECR-manufacturer. For this the terminal has to set up its own socket. The default port for this is 20008; the port must be changeable via configuration in the ECR and terminal.
  Caution: Service-functions are understood to be individual actions such as SW-Update, and not the messages typically sent during a transaction, such as (Intermediate-) Status-Information.

Definition of the hardware connections is not a part of this specification, and is to be determined between ECR-manufacturer and terminal-manufacturer.

2.3 USB

For a USB-connection the ECR idt defined as Host and the PT as Slave/Client.
The terminal must provide its own power-supply.

Depending on the ECR/PT the modes Low-Speed (1,5 Mbit/s), Full-Speed (12 Mbit/s) or High-Speed (480 Mbit/s) can be used.

Definition of the hardware connections is not a part of this specification, and is to be determined between ECR-manufacturer and terminal-manufacturer.

Depending on the ECR/PT, either serial transport-protocol or TCP/IP can be used for USB.

3 Serial Transport-Protocol

3.1 Block-Structure

The information to be transferred (hereafter referred to as APDU) is always packed in the following block-structure:

DLE	STX	APDU	DLE	ETX	CRC Low-byte	CRC High-byte

This type of block-structure is referred to in the following description as message.

Note:
If the APDU contains a “10” (=DLE), a second DLE will be added. The second DLE is however not included in the CRC-calculation or length parameter.

3.2 Response

The receiver immediately sends a positive or negative acknowledgement to each received message (before processing the information contained in the APDU). The acknowledgement only confirms that the message was received error-free or not.

Error-free means: the message is formally correct constructed and the CRC is also correct.
The acknowledgement is not an APDU, therefore is not packed as a message.
There is no acknowledgement to an acknowledgement itself.

3.2.1 Positive Acknowledgement

The receiver sends an ACK

3.2.2 Negative Acknowledgement

The receiver sends a NAK

3.3 CRC-Calculation

The CRC-checksum is calculated using the CRC-XModem with polynomial $x^{16}+ x^{15}+ x^{10}+ x^{3}$.
All characters from the APDU plus ETX are used in the checksum calculation (see chapter Block-Structure).

The following are **not** used in the CRC-calculation:
- the DLE from the start-sequence
- STX
- the DLE from the end-sequence (ETX **is** used for the calculation)
- any DLEs inserted for code-transparency

3.4 Time-outs and the Response to Time-outs

3.4.1 T1 (wait-time between two bytes)

Within a message the wait-time between two bytes may not reach 200msec. If this time-out is reached, the receiver sends a NAK.

3.4.2 T2 (wait-time between blocks and acknowledgement with ACK/NAK)

The receiver must immediately acknowledge a received message to transmitter with ACK or NAK. The wait-time between reception of the message and transmission of the acknowledgement must not reach 5s. If this time-out is reached, the message is repeated by the transmitter.

3.5 Error-Handling

If the receiver responds to a message with NAK, or if time-outs T1 or T2 apply, the transmitter repeats the message up to two times. Following this, if still no valid message has been received (NAK, Time-out T1 or Time-out T2), then both communications-partners report a transmission-error to the application level.

4 Transport-Protocol TCP/IP

The following is generally valid for TCP/IP:
1. The TCP/IP-layer makes the receiver-data available transparently.
2. The TCP/IP-layer makes the receiver-data available in the correct order.
3. The receiver must be able deal with arbitrarily separated data. It is not guaranteed that the complete message will be received in a single packet (timing, paket-size).

4.1 Block-Structure

The information to be transmitted, hereafter described as APDU or message, is always sent directly (i.e. without DLE, STX, DLE, ETX, CRC):

		APDU				

The construction of the APDU is described in the following chapter.

5 Application-Protocol

5.1 General

As a rule the basic state is with ECR as master and PT as slave. That means the PT cannot send unsolicited commands to the ECR. Using a certain command sequence, the PT is given the “master-rights”, where necessary, by the ECR. After completion of the command-sequence the ECR the „master-rights“ are given back to the ECR.

Exception:
For a TCP/IP the PT can make an unsolicited connection to the ECR for service-functions. In this case the PT is (only for the service-functions) the master.
It must be checked here if the ECR supports such separate socket connections!

5.2 APDUs and Application Response

5.2.1 Principle construction of the APDU

APDU		
Control-field	Length-field	Data-block
2 byte	1 byte/3 byte	xx

The APDU is information to be transmitted (see also Transport-Protocol).
The Control-field contains the commands (e.g. Authorisation, Closing, etc.) or the response to the coommands.
The Length-field contains the length of the following data.
In the Data-block extra parameters can be sent with the command. Not all commands require parameters.

5.2.2 Command Control-field

Control-field	
CLASS	INSTR

5.2.3 Response Control-field

Basically after each command a response is sent (not to be confused with ACK and NAK from the transport-layer):

Control-field	
CCRC	APRC

Possible responses:

- CCRC = 80 ➔ Positive completion, APRC is always 00, data-field may contain data, depending on the command.
- CCRC = 84 ➔ Negative completion, APRC contains the error-ID, data-field may contain data.
  Exception: The combination 84-00 can likewise indicate a positive completion.
  The possible error-IDs are listed in a separate chapter. See chapter Error-Messages.

5.2.4 Length-field

The length-field contains the length of the directly adjacent data-block. The control-field and the length-field themselves are not included in the length-calculation.
If the APDU contains a “10” (=DLE), then for a **serial-connection** a further DLE is added directly after the "10", but is not included in the length-calculation. For a **TCP/IP-connection** no extra DLE is attached.

The lenth-field itself has a length of one byte. The following data-block can therefore have a length of 0 to max. 254 bytes:

APDU		
Control-field	Length-field	Data-block
2 byte	1 byte	data with a length of 0 to 254 byte

Should more than 254 bytes be transferred in a data-block (e.g. for a software-update) then length-field contains ‚FF‘, meaning the following 2 bytes are defined as an extended length-field:

APDU				
Control-field	Length-field	Extended Length-field		Data-block
2 byte	1 byte	2 byte		data with a length of 0 to 65535 byte
		Lo-byte	Hi-byte	

In this case only the extended length-field contains the length of the following data-block, the length-field is simply an indicator, that an extended length-field folllows.
The extended length-field itself is likewise not included in the length calculation. Therefore the data-block can have a maximum length of 65535 bytes.

In all subsequent chapters only the simple APDU version without extended length-field will be shown – even though, where necessary, it may be used with an extended length-field.

5.2.5 Time-out T3 (time between a command and the response 80-00/84-XX)

The response 80-00 or 84-xx to a command must be received within time-out T3 (default: 5 seconds), otherwise the application-protocol layer reports a time-out error to the application, which then begins a second attempt or aborts. If the ECR sends a time-out together with a command (e.g. for text-displays with numerical input), then this time-out is taken as T3.

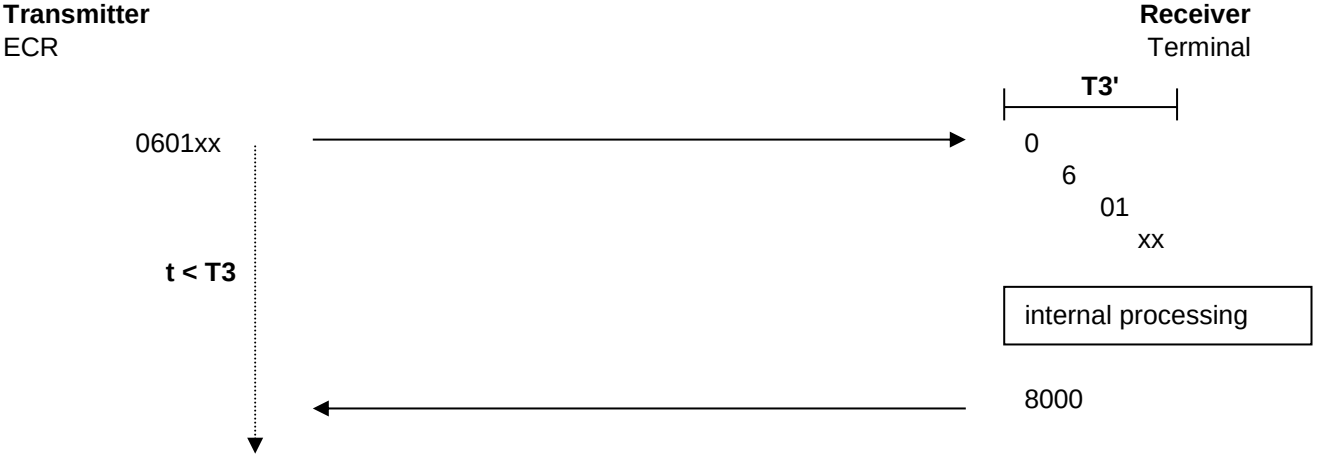
Special-case for TCP/IP-connections:
Depending on the type and layout of the network the data may be more or less delayed during transport. Therefore the time-out T3 should be adjustable.

Time-out case a)
If the transmitter does not receive a complete response to its request within time-out T3, then the connection is judged to aborted. In this case the transmitter should close the socket. The initiator of the process (typically the ECR; for service-actions it may also be the terminal) can subsequently re-open the socket and start the complete process anew or it can send an error-message to the application.

Time-out case b)
If the receiver does not receive the data in a single packet, and the individual packets of the message are received with such delays that time-out T3 exceeded is, then the receiver should close the socket. The initiator of the process (typically the ECR; for service-actions it may also be the terminal) can subsequently re-open the socket and start the complete process anew or it can send an error-message to the application.

If the socket is closed prematurely, then the whole process is judged to be aborted.

Examples:
1) Best case



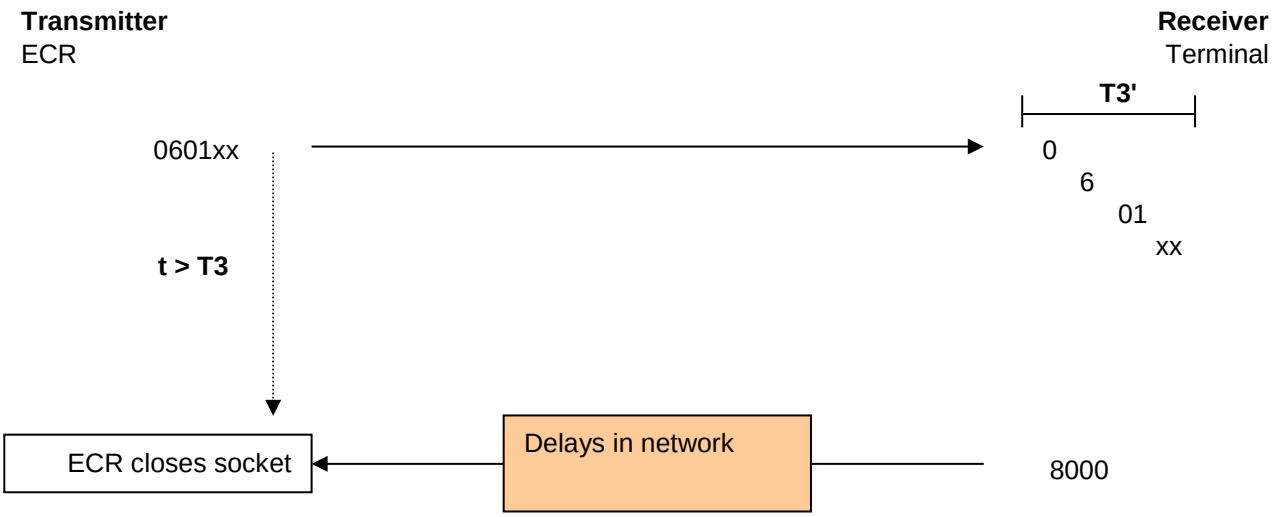
t = elapsed time until response; T3 = time-out T3; T3' = time for the reception of the message.

2) time-out case a: receiver responds too late

t = elapsed time until response; T3 = time-out T3; T3' = time for the reception of the message.

The ECR can re-open the socket and send command 0601 anew or send an error-message to the application.

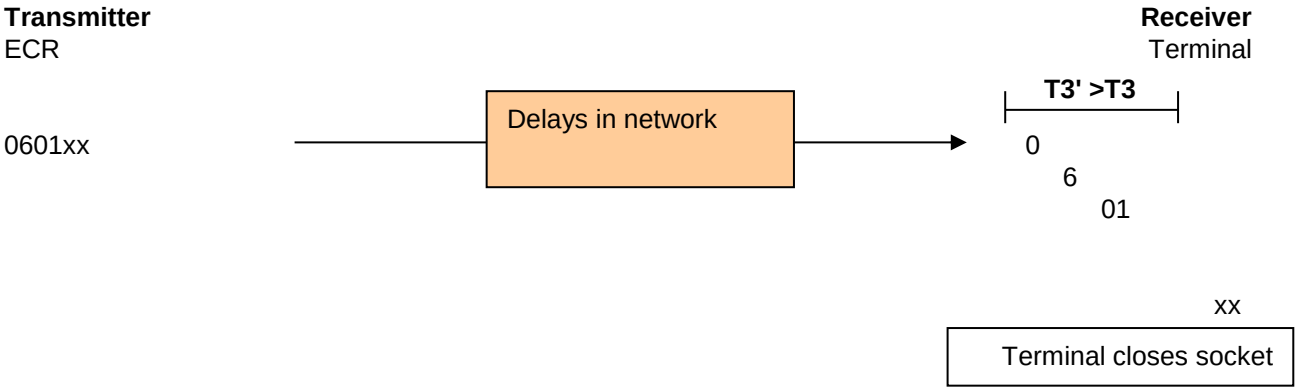
3) time-out case b: response delayed --> response arrives too late



t = elapsed time until response; T3 = time-out T3; T3' = time for the reception of the message.

The ECR can re-open the socket and send command 0601 anew or send an error-message to the application.

4) time-out case c: receiver receives packets with too long a delay



The ECR can re-open the socket and send command 0601 anew or send an error-message to the application.

5.2.6 Time-out T4 (time between 80-00 from PT and completion-command)

For commands which comprise a request from the ECR and a response (80 00) from the PT, after which they are terminated with a completion, time-out T4 between PT response and completion is 180s.

The time-out is reset via commands **Intermediate Status-Information** and **Status-Information**. The command may be repeated as necessary.

If a time-out occurs, the application-protocol layer sends a time-out error to the application, which can then either re-attempt the process or abort it.

For dial-up via the ECR time-out T4 is also triggered anew.

The time-out T4 should be adjustable.

5.2.7 Parameters and Bitmaps

Some commands transfer multiple parameters in the data-block. Sometimes fixed parameters are used (with consideration to position and presence in the data-block). Frequently, and especially for optional parameters, bitmaps are used to identify the parameter. Pre-fixing the parameter is a bitmap-number (as parameter-name) and following that are the parameter-contents. Which method is used for which commands is evident from the command definition.

Example for parameter without bitmaps:

Registration:

ECR → PT			
APDU			
Control-field		Length-field	Data-block
CLASS	INSTR		
06	00	xx	<password><config-byte>[<CC>]

Example for parameter with bitmaps:

Completion:

PT → ECR			
APDU			
Control-field		Length-field	Data-block
CLASS	INSTR		
06	0F	xx	19<status-byte>29<TID>[49<CC>]

‚19' is the identifier for the parameter <status-byte>, ‚29' is the identifier for the parameter <TID> and ‚49' is the identifier for the optional (recognised by the angled brackets) parameter <WKZ>.

In this specification the word „field" is used synonymously with "parameter".

5.2.8 Data-fields with variable length

Certain data-fields have a variable field-contents length. These fields are identified as LLVAR or LLLVAR.

5.2.8.1 LLVAR

The field always begins with „FxFy“ whereby xy positions follow.
e.g.
F1 F2 01 23 45 67 89 01 23 45 67 89 01 23 (F1 F2 states that the following 12 bytes belong to this field)
F0 F3 01 23 45 (F0 F3 means 3 following bytes)

5.2.8.2 LLLVAR

The field always begins with „FxFyFz“ whereby xyz positions follow.
e.g.
F0 F1 F2 01 23 45 67 89 01 23 45 67 89 01 23 (F0 F1 F2 states that the following 12 bytes belong to this field)
F0 F0 F3 01 23 45 (F0 F0 F3 means 3 following bytes)

6 Further questions

For further questions regarding the ZVT implementation with your terminal, please contact your vendor or the terminal manufacturer of the terminal.

7 References

The current version of this documention is available under www.terminalhersteller.de

PA00P015 ECR Interface ZVT-Protocol – Commands, Bitmaps, Error-Messages
PA00P017 Implications of TA7.0 / DC POS2.4 on the ECR-Interface Protocol

8 Change-Control

The change-control for this documentation is assigned to EL-ME AG. The current versions are announced on www.terminalhersteller.de.