Part 8

Host-CT-Interface
for
MCTs with V.24/V.28 connection

Version 0.9

28.07.95

GMD - Forschungszentrum Informationstechnik GmbH

Contents

1. Scope

The aim of this specification is to define the host/CT-interface for those Multifunctional Card-Terminals (MCTs), which are connected to a computer as external devices via a V.24 interface. The description of the interface is limited to the physical layer and the transmission layer. At the application layer, card terminal commands (see CT-BCS) and IC commands occur.
The block transmission protocol T=1as specified in ISO/IEC 7816-3 is used as transmission protocol.

2. Normative references

Deutsche Telekom, GMD, RWTÜV, TeleTrusT Deutschland: 1995
CT-API 1.1 - Application independent CardTerminal Application Programming Interface for IC card applications

TeleTrusT Deutschland: 1995
CT-BCS - Application independent CardTerminal Basic Command Set for IC card applications

ISO/IEC 7816-3: 1989
Identification cards - Integrated circuit(s) cards with contacts
Part 3 - Electronic Signals and transmission protocols
AM 1: Clause 9: Protocol T=1, asynchronous half duplex block transmission protocol
AM 2: Protocol type selection
(The WD October 1994 in which AM 1and AM 2 are integrated is used)

3. Abbreviations

ATR = Answer-to-Reset
BWT = Block Waiting Time
CT = CardTerminal
CT-API =CT Application Programming Interface
CT-BCS = CT Basic Command Set
CTS = Clear to Send
CWT = Character Waiting Time
DAD = Destination Address
EDC = Error Detection Code
HB = Historical Bytes
HTSI = Host Transport Service Interface
ICC = Integrated Circuit(s) Card
IFS = Information Field Size
NAD = Node Address byte
PCB = Protocol Control Byte
R-Host = Remote Host
RTS = Request to Send
SAD = Source Address
Vpp = Programming voltage
XOR = Exclusive Or

4. Transmission parameters

For the communication between host (PC or workstation) and CardTerminal, the following transmission parameters shall be used:

- Speed (baud rate): 9600 baud and optionally additionally higher speeds

- Character frame: 1 start bit, 8 data bits, 1 parity bit (even parity), 1 stop bit, bit b1 is the least significant bit (lsb), bit b8 is the most significant bit (msb); the lsb is always the first to be transmitted

- Information field size card terminal (IFST): 254 byte (i.e. the I/O buffer in the card terminal shall be able to accept at least 258 bytes (254 bytes information field + 4 bytes T=1 frame)

- Maximum waiting time for a transmission block with the response to a prior sent command (Block Waiting Time BWT): 1000 ms

- Maximum interval between 2 characters in a transmission block (Character Waiting Time CWT): 100 ms

- Minimum waiting time between receipt of the last character of a block and transmission of the first character of the response block (Block Guard Time BGT): 2 ms

- Check sum (error detection code EDC): XOR (exclusive Or)

- RTS and CTS wires: RTS and CTS wires are not served by the host software (see annex).

5. Transmission protocol

The transmission protocol to be used is the block transmission protocol T = 1 (see ISO/IEC 7816-3). Fig. 1 shows the structure of a T = 1 block.

Fig. 1: T=1 Block

The following conventions shall be supported:

a) NAD byte

The NAD byte is used to identify the sender and the receiver of a transmission block. The Destination Address (DAD) appears in the most significant half byte, the Source Address (SAD) in theleast significant half byte. When sending a command from the host the following addresses are to be used:

NAD (DAD/SAD)	Meaning
02 05	ICC Command from Host to ICC1 ICC Command fr. R-Host to ICC1
12 15	CT Command from Host to CT CT Command from R-Host to CT
22 25	ICC Command from Host to ICC2 ICC Command fr. R-Host to ICC2
...	...
E2 E5	ICC Command from Host to ICC14 ICC Command fr. R-Host to ICC14

Tab. 1: Addresses when sending a command from Host or R-Host to the CT or an ICC

The support of the DAD addresses '0' (= ICC1) and '1' (= CT) is mandatory, the support of higher addresses is dependent on the number of additional ICC interfaces in a card terminal.

The NAD byte is set by the HTSI module (see CT-API description) whereby the least significant nibble of the DAD byte that is passed in the CT_data function is mapped to the most significant nibble of the NAD byte and the least significant nibble of the SAD byte that is passed in the CT_data function is mapped to the least significant nibble of the NAD byte.

On sending a response the following NAD codes are used by the card terminal, whereby SAD values > '1' only occur from card terminals with more than one ICC interface:

NAD (DAD/SAD)	Meaning
20 50	ICC Response from ICC1 to Host ICC Response fr. ICC1 to R-Host
21 51	CT Response from CT to Host CT Response from CT to R-Host
22 52	ICC Response from ICC2 to Host ICC Response fr. ICC2 to R-Host
...	...
2E 5E	ICC Response from ICC14 to Host ICC Response fr. ICC14 to R-Host

Tab. 2: Addresses on sending a response from the CT or an ICC to Host or R-Host

b) PCB Byte

The protocol control byte (PCB byte) contains information needed to control the transmission. The codes for

- I-Block
- R-Block
- S-Block

are to be set acc. to ISO/IEC 7816-3 (see tab. 3 - 5)

The 3 following tables show the coding of the T=1 blocks

b8	0 (= Indication I-block)
b7	Send sequence number N(S)
b6	Chaining (more data bit M) M = 1 Chained data follow in subsequent block(s) M = 0 Last block of chain
b5-b1	0 (RFU)

Tab. 3: Coding of the I-Block

b8	1
b7	0 (b8,b7 = Indication of R-block)
b6	0 (RFU)
b5	Receive sequence number N(R)
b4-b1	0 = Error free 1 = EDC and/or parity error 2 = Other error(s) Other values RFU

Tab. 4: Coding of the R-Block

b8	1
b7	1 (b8,b7 = Indication of S-block)
b6	0 = Request 1 = Response
b5-b1	0 = RESYNCH (Resynchronisation) 1 = IFS (not used) 2 = ABORT (not used) 3 = WTX (BWT extension) 4 = Vpp error (not used)

	Other values RFU

Tab. 5: Coding of the S-Block

1. Error free transmission

CT commands, ICC commands and the responses hereto are transmitted in the so-called I-block (information block). The send sequence counter is a security characteristic to identify the loss of a transmission block and shall therefore be supported. It has alternately the values 0 and 1, i.e. the first block sent by the host has the code '00' in the PCB byte, the second '40', the third again '00' etc.
The data chaining mechanism (more data bit) is also to be supported, so that application units (e.g. the response to a READ BINARY command) can be longer than the length of a single block. The information is split up into n blocks, whereby (n-1) blocks have a length corresponding to the information field size (254 bytes) and the n-th block contains the remaining bytes (max. 254). As it is necessary to have flow control when sending consecutive information blocks, an I-block with M-bit=1 should be acknowledged with a receive ready block.

2. Transmission with error handling

When a faulty I-block is received, the communication partner should be informed by means of an R-block. Bit b5 of the R-block has here the value of the send sequence number of the block, which should be repeated. If two errors occur consecutively, a resynchronisation should be carried out by the host (see section 4). In other error situations (e.g. faulty R-block or time-out) a resynchronisation should be initiated. Blocks, whose addresses are faulty in the NAD byte, are ignored by the CardTerminal, i.e. no response is sent.

3. Response time extension

If the card terminal receives a command whose execution takes longer than the Block Waiting Time of 1000 ms (that may occur for instance when requesting the ICC), the CardTerminal sends a WTX request (WTX = Waiting Time Extension) which has to be answered with a WTX response from the host. WTX requests/ responses are transmitted in an S-block (supervisory block) whereby the 1 byte long multiplier of the BWT value is given in the INF field. The Waiting Time Extension begins after the last byte of the WTX response has been received. It refers in principle only to the next response block to be transmitted.
A WTX request can also be rejected by the host. In this case as response to a WTX request a RESYNCH request is sent, that has to be answered by a card terminal with RESYNCH response. Details can be found in section 4.

4. Resynchronisation

To initiate a resynchronisation a RESYNCH request can be sent from the PC or workstation. The request must be answered by the card terminal with the RESYNCH response. The RESYNCH request must *always* be sent after the start of an application from the HTSI module in the host to the CardTerminal as a component of the execution of the CT_init function (see CT-API description. In certain error situations (see Section 2) and to cancel a command, if required (see Section 3), the RESYNCH mechanism has to be used. The RESYNCH request/response pair is used to synchronize the transmission protocol automats in the host and in the CardTerminal or to resynchronize them after faulty or interrupted communication. The send sequence counters are also reset to zero. An application command being processed will be cancelled.

c) LEN byte

The LEN byte (binary coding) is used to denote the length of the information field.

d) INF field

In the information field of an I-block the command or the response to the command is transmitted.

e) EDC field

In the EDC field the XOR checksum (1 byte) is transmitted.

Annex (normative)

Plug arrangement for serial communication

To connect the MCT with a host, a nine pin Sub-D socket (female) is used at the MCT side.

For serial communication with the host the following signals are used:

RxD = Received Data
TxD = Transmitted Data
GND = Signal Ground

Tab. 1 shows the pin arrangement on the host side and MCT side using as example for the host a computer of the PC-AT class.

Pin No. PC-AT 9 pin plug	Host	MCT	Pin No. MCT 9 pin socket
2	RxD	TxD	2
3	TxD	RxD	3
5	GND	GND	5

Tab. 1: Pin arrangement

The pin arrangement was selected in such a way that the MCT can be connected with a host by means of a cable which is provided at one end with a socket and at the other end with a plug whereby the pins of the same number are connected without crosswiring.