ORGA
The Smart Card Integrator

WANTED

MCT 5040 BCS – LAN

Card Terminal Server

10 Mbit/s Ethernet Interface
Embedded TCP/IP
Card Terminal Session Layer (CTSL)
Card Terminal Access Manager (CTAM)
Card Terminal Basic Command Set (CT-BCS)
Multi Channel Mode
Single Channel Mode
Login with User Name and Password
DES Encryption
Configuration over Network
1 MB SRAM, 8 MB FLASH
Other Technical Data such as MCT 5000 BCS

CT-API Library with integrated CTSL for Windows and Linux

Compatible with existing Programs
RS232 Port Mapping
Configuration File for TCP/IP Parameters

Card Terminal Server

10 Mbit/s Ethernet Interface
IEEE 802.3 compatible Ethernet Controller, RJ45 connection (operation via RS232 still possible).

Embedded TCP/IP
Embedded TCP/IP protocol stack in accordance with IP/v4 und RFC.

Card Terminal Session Layer (CTSL)
Protocol for safe network communication with client (PC, HOST).
Login with user name and password.

Card Terminal Access Manager (CTAM)
Resource management (CARD TERMINAL, ICC1, ..., ICC6). Filters all `CT_data` commands according to the "Host to Card" connection and passes them on to the CT-BCS module.

Card Terminal Basic Command Set (CT-BCS)
Command interpreter for the execution of CT and ICC commands.

Multi Channel Mode
A chip card can be assigned to exactly one client process. This creates a safe "Host to Card" connection with a reduced CT-command record. Depending on the equipment version (number of card slots), up to six „Host to Card“ connections can be realized.

Single Channel Mode
A client has the sole unrestricted access to the terminal and the chip cards. No restriction to the CT command record (Admin Mode). This mode corresponds to the conventional serial connection (RS232/USB) "Host to Terminal".

Login / Logout
The login is executed implicitly with `CT_init.` User name, password and slot number are sent to the terminal where they are checked. If the indicated slot number is not busy, it will be assigned to the client and will no longer be available for other clients. A logout is carried out implicitly with `CT_close,` thereby releasing the respective slot.

DES Encryption
It is possible to set a DES encrypted data transfer.

Configuration over Network
The parameters for the network connection can be set with Tool mctconfig:

```
username      password      tcp_port      ip_addr       sub_mask
adminname     admin_password
```

CT-API Library with integrated CTSL for Windows and Linux

Compatible with existing Programs
Thanks to the new CT-API library, all existing applications can use the network terminal in a „Single Channel Mode“ without modification of the source code. The required TCP/IP parameters will be written into a configuration file. By means of the RS232 Port Mapping, the connection is being redirected to TCP.

Configuration File for TCP/IP Parameters on Linux Host

```
pn=PORT_TCP1          # CT-API port
PORT_COM1=PORT_TCP1   # RS232 port mapping
ip_addr=192.168.1.1
tcp_port=9732
username=tux
password=mctlan
encryption=1          # with DES encryption
```

Multifunctional Card Terminal with CT-BCS and Ethernet-Interface (CT-BCS-LAN)

Manual for Linux Host

Version 0.5
1.02.2005
ORGA Kartensysteme

1. Quick Start Up

a) Connect the MCT 5140 LAN using a patch cable (RJ45/TP) via switch or with a cross-over cable directly.

b) Install the CT-API library libCTORGT1.so.1.2.0 (deinstall the old library beforehand, see readme) The new library is suitable for RS232 as well as for Ethernet. The selection of the interface is effected automatically to the same interface (RS232 or Ethernet) that has been used to carry out a CT-init() first thing after switching on the terminal. The other interface will be blocked.

c) Create file /usr/local/etc/ctorg.conf having the following contents:
(sample file ctorg.conf is included)

```
# Configuration file for the CT-API Library libCTORGT1.so.1.2.x
# when using MCT LAN Terminal
#
# The main use of this configuration file is to support old applications which are
# linked to an older library which does not support TCP communication. Therefore
# you can call CT_init(ctn, PORT_COM1) with PORT_COM1 for using the LAN-MCT by
# mapping the PORT_COM1 to a PORT_TCPx in this configuration file.
# Nevertheless you can use this file for general use but you must know that the
# parameters (i.e. password) are in plain text.
# For secure parameters use the functions CT_set_tcp_parms() and CT_get_tcp_parms()
# to do your own parameter management (see ctapi.h).
#
# The place of this file is under /usr/local/etc/ctorg.conf.
#
# Discription
#
#   # This is a comment
#
# Please write the line without blanks (pn=PORT_TCP1 and not pn = PORT_TCP1)
# A data section always begins with the pn=PORT_TCPx line.
#
# The line encrypted= can be set to encrypted=1 or encrypted=0 or you can leave it.
# When using encrypted=1 the Host-Terminal communication will be always encrypted
# with DES, which means that you have secure but slower data transfer than without
# encryption (encryption=0). The password is part of the encrytion key.
#
# If you do not set the encryption here in this file you can set it as parameter in
# CT_init(ctn, pn | ENCRYPTED) separatly for each card slot (see manual).
# If the encryption flag is set in the configuration file (this file), then
# ENCRYPTED has no effect.
#
# If you need support do not hesitate to write a mail to cardterminals@orga.com
#
pn=TCP_PORT1
PORT_COM1=PORT_TCP1     # RS232 port mapping for using with old applications
ip_addr=192.168.1.1     # IP Addr of your MCT
tcp_port=9700           # Port of your MCT
username=orga
password=bcslan
encrypted=1
```

```
### COM2 to TCP
pn=TCP_PORT2
PORT_COM2=PORT_TCP2     #  RS232 port mapping for used applications
ip_addr=192.168.1.1     # IP Addr of your MCT
tcp_port=9700           # Port of your MCT
username=orga
password=bcslan
encrypted=1

### COM3 to TCP
pn=TCP_PORT3
PORT_COM3=PORT_TCP3     # RS232 port mapping for used applications
ip_addr=192.168.1.1     # IP Addr of your MCT
tcp_port=9700           # Port of your MCT
username=orga
password=bcslan
encrypted=1

### COM4 to TCP
pn=TCP_PORT4
PORT_COM4=PORT_TCP4     # RS232 port mapping for used applications
ip_addr=192.168.1.1     # IP Addr of your MCT
tcp_port=9700           # Port of your MCT
username=orga
password=bcslan
encrypted=1

pn=TCP_PORT5
ip_addr=192.168.1.1     # IP Addr of your MCT
tcp_port=9700           # Port of your MCT
username=orga
password=bcslan
encrypted=1

### default admin port
pn=PORT_TCP30
ip_addr=192.168.1.1
tcp_port=9700
username=admin
password=1234
encrypted=1

### default user port
pn=PORT_TCP31
ip_addr=192.168.1.1
tcp_port=9700
username=orga
password=bcslan
encrypted=1
```

d) Configure the terminal using mctconfig

The MCT is configured with "factory default" parameter

username:	orga
password:	bcslan
tcp_port:	9700
ip_addr:	192.168.1.1
sub_mask:	255.255.255.0
gateway:	0.0.0.0 gateway not required
adminname:	admin
admin password:	1234

All factory defaults can be restored in the MCT menu:
Main Menu > Service > Communication > LAN > Set Default.

In order to set parameters for your own network, please connect the terminal to a PC with 192.168.1.x address, or enter an IP-Address and Subnet-Mask of your network (LAN) in the Card Terminal Menu, install the CT-API library and call mctconfig with the respective options. Before that, please execute a ping 192.168.1.1 to check if the terminal is accessible.

You will find information for working with „mctconfig“ in the sample script "conf-sh". mctconfig –H gives you an overview of all options.

In order to read and set parameters, you must be logged in as administrator. For doing so, you need to make the correct entries in the file /usr/local/etc/ctorg.conf .

The attached sample file ctorg.conf contains all necessary parameters in pn=PORT_TCP30.

e) Start and test the application.

2. CT-API with Card Terminal Session Layer

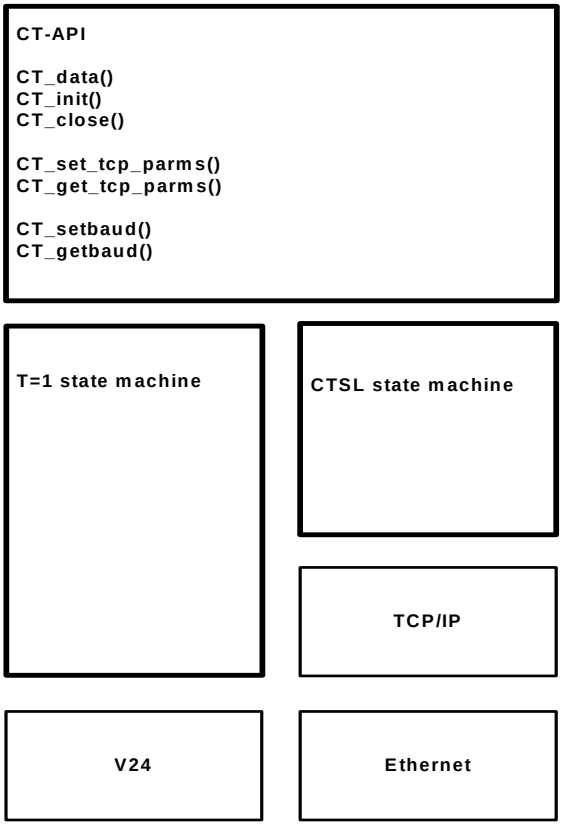
The Card Terminal Session Layer (CTSL) works according to OSI in layer 5 upside of TCP and is integrated in the CT-API library. For the application, you may use the conventional CT-API Interface having the following functions:

```
CT_init(ctn, pn)
CT_close(ctn)
CT_data(ctn, dad, sad, lenc, command, lenr, response)
```

and an additional interface for the configuration of the network connection:

```
int CT_set_tcp_parms(unsigned short pn, struct tcp_parms * tcpio);
int CT_get_tcp_parms(unsigned short pn, struct tcp_parms * tcpio);

struct tcp_parms {
      unsigned short pn;       /* ct-api port number */
      unsigned char ip_addr_dot[21]; /* ip addr in dot notation (string) */
      unsigned short tcp_port; /* tcp port number of card terminal */
      unsigned char username[21];
      unsigned char password[21];
      unsigned short encrypted;
      };
```



CT-API Protocol Driver for Linux Host

The constants for pn have been extended in ctapi.h for the TCP connections:

```
PORT_TCP1
PORT_TCP2
.
.
.
PORT_TCP64
```

In addition to the port you also indicate the chip card you want to communicate with. You will have exclusive access rights for this card until the call for CT_close().
The Card Terminal Access Manager (CTAM) manages the access rights inside the card terminal.

```
SCARD1    # Connection to the chip card in slot 1
SCARD2    # Connection to the chip card in slot 2
SCARD3    # Connection to the chip card in slot 3
SCARD4    # Connection to the chip card in slot 4
SCARD5    # Connection to the chip card in slot 5 if present MCT 5060
SCARD6    # Connection to the chip card in slot 6 if present MCT 5060
```

Furthermore, there are 3 optional flags as described

```
CARDTERMINAL  # Single Channel Mode, access to all, exclusively
CTADMIN       # Superuser Mode for the configuration of the card terminal
ENCRYPTED     # connection is encrypted
```

Examples for the call of CT_init():

```
1. CT_init(ctn, PORT_TCP2 | SCARD1 | SCARD2);
```

A login into the terminal is effected, requiring user name and password, reserving slot 1 and slot 2 for cards 1 and 2 respectively. If the slots are already busy, you will receive an error message.

```
2. CT_init(ctn, PORT_TCP2);
```

In order to assure compatibility to existing implementations, the complete terminal will be reserved if only PORT_TCPx (x= 1 – 64) is indicated (Single Channel Mode). This means, only one host / process has complete access to the terminal, like with a serial point to point connection. This call is equivalent to `CT_init(ctn, PORT_TCP1 | CARDTERMINAL);`

```
3. CT_init(ctn, PORT_TCP30 | CTADMIN | ENCRYPTED)
```

Terminal access as superuser (CTADMIN) with encryption (ENCRYPTED) in order to configure / administrate the terminal. The superuser can log in at any time and has unrestricted access to the terminal (all CT-BCS commands, access to every chip card, also in Multi Channel Mode). Due to the power a superuser has, this password should be kept safe.

```
4. CT_init(ctn, PORT_COM1)
```

Normally, this is to initialize the terminal via the serial RS232 connection. However, if the configuration file /usr/local/etc/ctorg.conf contains the line

PORT_COM1=PORT_TCP1,

the connection is then redirected (mapped) to PORT_TCP1 and the login will be carried out via TCP. This function has been designed to ensure the compatibility to existing applications and is only possible in connection with the configuration file ctorg.conf.

3. Possibilities for setting the configuration parameters

3.1 The TCP parameters are read and set via the functions CT_get_tcp_parms() and CT_set_tcp_parms().

Example:

```
 struct tcp_parms  tcpio;

CT_get_tcp_parms(PORT_TCP1, &tcpio);
  tcpio.pn = PORT_TCP1;
  strcpy((char *)tcpio.ip_addr_dot, "172.20.5.206");
  tcpio.tcp_port = 9732;
  strcpy((char *)tcpio.username, "tux");
  strcpy((char *)tcpio.password, "mct5040");
  encrypted = 0;
CT_set_tcp_parms(PORT_TCP1, &tcpio);
```

3.2 In order to assure compatibility to existing implementations, the TCP parameters are read and set automatically from the file /usr/local/etc/ctorg.conf if same is existing, as soon as the CT-API library is being loaded. As an example, the file may then contain the following entries:

```
# Any data record starts with the port number pn
#
# sample parameters for a MCT 5040 BCS-LAN in an Office LAN
#
pn=PORT_TCP1
PORT_COM1=PORT_TCP1   # optional: mapping RS232 to TCP
ip_addr=172.20.5.206
tcp_port=9732
username=tux
password=mct5040
encrypted=0

pn=PORT_TCP2
#PORT_COM2=PORT_TCP2   # optional: mapping RS232 to TCP
ip_addr=172.20.5.206
tcp_port=9732
username=tux
password=mct5040
encrypted=0

#default factory parameter for superuser
#
pn=PORT_TCP30
ip_addr=192.168.1.1
tcp_port=9700
username=admin
password=1234
encrypted=0
```

4. Restrictions of the CT-BCS in Multi Channel Mode

If the LAN terminal is operated in **Single User Mode** (CARDTERMINAL), it reacts like a terminal in serial operation. There are no restrictions for the CT commands of the CT-BCS.
If operated in **Multi User Mode** (SCARD1, SCARD2, SCARD3, SCARD4, SCARD5, SCARD6) there are restrictions for the CT commands of the CT-BCS.
These restrictions are shown in the following hard copies of a test program.

4.1 Menu for choice of port pn and of the slot/s.

```
 Current CTN: 1     PORT is: (PORT_COM1 | [CARDTERMINAL])

 CTN     PORT      Channel
( 1) PORT_COM1
( 2) PORT_COM2
( 3) PORT_COM3
( 4) PORT_COM4

( 5) PORT_TCP5 | SCARD1 | ENCRYPTED
( 6) PORT_TCP6 | SCARD2
( 7) PORT_TCP7 | SCARD3
( 8) PORT_TCP8 | SCARD4
( 9) PORT_TCP9 | SCARD5
(10) PORT_TCP10 | SCARD6
(11) PORT_TCP11 | CARDTERMINAL | ENCRYPTED

(12) PORT_TCP12 | SCARD1 | SCARD2
(13) PORT_TCP13 | SCARD3 | SCARD4
(14) PORT_TCP14 | SCARD1 | SCARD2 | SCARD3 | SCARD4
(15) PORT_TCP15 | SCARD2 | SCARD3 | SCARD4
(16) PORT_TCP16 | CTADMIN | ENCRYPTED

( 0) Back

choose the CTN :
Wahl: 5

...
```

4.2 The option CARDTERMINAL has been chosen which means operation in Single User Mode without restrictions.

CT_init(ctn, PORT_TCP11 | CARDTERMINAL | ENCRYPTED)

```
Channel to [ CARDTERMINAL ]
Allowed Commands for CARDTERMINAL Channel (All Commands)
Choose a Command for Testing Access Permission

 ( 1) RESET CT (CT)              ( 2) RESET CT ICC1              ( 3) RESET CT ICC2
 ( 4) RESET CT ICC3              ( 5) RESET CT ICC4              ( 6) RESET CT ICC5
 ( 7) RESET CT ICC6              ( 8) REQUEST ICC1               ( 9) REQUEST ICC2
 (10) REQUEST ICC3               (11) REQUEST ICC4               (12) REQUEST ICC5
 (13) REQUEST ICC6               (14) GET STATUS CT (CM-DO)      (15) GET STATUS CT (ICC DOs)
 (16) GET STATUS CT (FU DO)      (17) GET STATUS ICC1            (18) GET STATUS ICC2
 (19) GET STATUS ICC3            (20) GET STATUS ICC4            (21) GET STATUS ICC5
 (22) GET STATUS ICC6            (23) EJECT ICC1                 (24) EJECT ICC2
 (25) EJECT ICC3                 (26) EJECT ICC4                 (27) EJECT ICC5
 (28) EJECT ICC6                 (29) INPUT                      (30) OUTPUT
 (31) PERFORM VERIFICATION ICC1  (32) PERFORM VERIFICATION ICC2  (33) PERFORM VERIFICATION ICC3
 (34) PERFORM VERIFICATION ICC4  (35) PERFORM VERIFICATION ICC5  (36) PERFORM VERIFICATION ICC6
 (37) MODIFY VERIFICATION ICC1   (38) MODIFY VERIFICATION ICC2   (39) MODIFY VERIFICATION ICC3
 (40) MODIFY VERIFICATION ICC4   (41) MODIFY VERIFICATION ICC5   (42) MODIFY VERIFICATION ICC6
 (43) B1 RESET CT                (44) B1 RESET ICC1              (45) B1 RESET ICC2
 (46) B1 DEACTIVATE ICC1         (47) B1 DEACTIVATE ICC2         (48) ACS COLD RESET CT
 (49) ACS COLD RESET CT ICC1     (50) ACS COLD RESET CT ICC2     (51) ACS COLD RESET CT ICC3
 (52) ACS COLD RESET CT ICC4     (53) ACS COLD RESET CT ICC5     (54) ACS COLD RESET CT ICC6
 (55) ACS WARM RESET CT          (56) ACS WARM RESET CT ICC1     (57) ACS WARM RESET CT ICC2
 (58) ACS WARM RESET CT ICC3     (59) ACS WARM RESET CT ICC4     (60) ACS WARM RESET CT ICC5
 (61) ACS WARM RESET CT ICC6     (62) ACS FREEZE ON              (63) ACS FREEZE OFF

 ( 0) Back          ==>Wahl:
```

4.3 In the following, single slots have been chosen (Multi User Mode) with access restriction on the CT-BCS

4.3.1 CT_init(ctn, PORT_TCP5 | SCARD1 | ENCRYPTED)

```
 Channel to [ ICC1 ]
 Allowed Commands for SCARD1 Channel (ICC1)
 Choose a Command for Testing Access Permission

 ( 1) RESET CT ICC1              ( 2) REQUEST ICC1               ( 3) GET STATUS CT (CM-DO)
 ( 4) GET STATUS CT (ICC DOs)    ( 5) GET STATUS CT (FU DO)      ( 6) GET STATUS ICC1
 ( 7) EJECT ICC1                 ( 8) PERFORM VERIFICATION ICC1  ( 9) MODIFY VERIFICATION ICC1
 (10) B1 RESET ICC1              (11) B1 DEACTIVATE ICC1         (12) ACS COLD RESET CT ICC1
```

```
(13) ACS WARM RESET CT ICC1
( 0) Back           ==>Wahl:
```

4.3.2 CT_init(ctn, PORT_TCP5 | SCARD2)

```
Channel to [ ICC2 ]
Allowed Commands for SCARD2 Channel (ICC2)
Choose a Command for Testing Access Permission

( 1) RESET CT ICC2              ( 2) REQUEST ICC2               ( 3) GET STATUS CT (CM-DO)
( 4) GET STATUS CT (ICC DOs)    ( 5) GET STATUS CT (FU DO)      ( 6) GET STATUS ICC2
( 7) EJECT ICC2                 ( 8) PERFORM VERIFICATION ICC2  ( 9) MODIFY VERIFICATION ICC2
(10) B1 RESET ICC2              (11) B1 DEACTIVATE ICC2         (12) ACS COLD RESET CT ICC2
(13) ACS WARM RESET CT ICC2
( 0) Back           ==>Wahl:
```

4.3.3 CT_init(ctn, PORT_TCP5 | SCARD3)

```
Channel to [ ICC3 ]
Allowed Commands for SCARD3 Channel (ICC3)
Choose a Command for Testing Access Permission

( 1) RESET CT ICC3              ( 2) REQUEST ICC3               ( 3) GET STATUS CT (CM-DO)
( 4) GET STATUS CT (ICC DOs)    ( 5) GET STATUS CT (FU DO)      ( 6) GET STATUS ICC3
( 7) EJECT ICC3                 ( 8) PERFORM VERIFICATION ICC3  ( 9) MODIFY VERIFICATION ICC3
(10) ACS COLD RESET CT ICC3     (11) ACS WARM RESET CT ICC3
( 0) Back           ==>Wahl:
```

4.3.4 CT_init(ctn, PORT_TCP5 | SCARD4)

```
Channel to [ ICC4 ]
Allowed Commands for SCARD4 Channel (ICC4)
Choose a Command for Testing Access Permission

( 1) RESET CT ICC4              ( 2) REQUEST ICC4               ( 3) GET STATUS CT (CM-DO)
( 4) GET STATUS CT (ICC DOs)    ( 5) GET STATUS CT (FU DO)      ( 6) GET STATUS ICC4
( 7) EJECT ICC4                 ( 8) PERFORM VERIFICATION ICC4  ( 9) MODIFY VERIFICATION ICC4
(10) ACS COLD RESET CT ICC4     (11) ACS WARM RESET CT ICC4
( 0) Back           ==>Wahl:

...
```

4.3.5 CT_init(ctn, PORT_TCP5 | SCARD5)

```
Channel to [ ICC5 ]
Allowed Commands for SCARD5 Channel (ICC5)
Choose a Command for Testing Access Permission

( 1) RESET CT ICC5              ( 2) REQUEST ICC5               ( 3) GET STATUS CT (CM-DO)
( 4) GET STATUS CT (ICC DOs)    ( 5) GET STATUS CT (FU DO)      ( 6) GET STATUS ICC5
( 7) EJECT ICC5                 ( 8) PERFORM VERIFICATION ICC5  ( 9) MODIFY VERIFICATION ICC5
(10) ACS COLD RESET CT ICC5     (11) ACS WARM RESET CT ICC5
( 0) Back           ==>Wahl:
```

4.3.6 CT_init(ctn, PORT_TCP5 | SCARD6)

```
Channel to [ ICC6 ]
Allowed Commands for SCARD6 Channel (ICC6)
Choose a Command for Testing Access Permission

( 1) RESET CT ICC6              ( 2) REQUEST ICC6               ( 3) GET STATUS CT (CM-DO)
( 4) GET STATUS CT (ICC DOs)    ( 5) GET STATUS CT (FU DO)      ( 6) GET STATUS ICC6
( 7) EJECT ICC6                 ( 8) PERFORM VERIFICATION ICC6  ( 9) MODIFY VERIFICATION ICC6
(10) ACS COLD RESET CT ICC6     (11) ACS WARM RESET CT ICC6
( 0) Back           ==>Wahl:
```