

**CT-API FÜR**

**CARD  
STAR** | medic 



## Bedienungshinweise

14. Februar 2013  
Version 2.70  
Copyright: Celectronic



**celectronic**  
[www.celectronic.de](http://www.celectronic.de)

## Inhaltsverzeichnis zur Bedienungsanleitung des *CARD STAR /medic2*

<b>1.</b>	<b>CARD STAR /medic2</b>	<b>1</b>
1.1	Einsatzumgebungen	1
1.2	Terminalbetrieb	1
1.3	Einstellung der Terminalparameter	2
1.4	Kartensteckfunktion	2
1.5	Schnittstellenparameter	3
1.6	Protokoll-Timing	3
1.7	<i>CARD STAR /memo2</i>	3
<b>2.</b>	<b>Ansteuerung vom Host</b>	<b>4</b>
2.1	Geräteschnittstelle	4
2.2	CT-API	5
<b>3.</b>	<b>CT-API für WINDOWS</b>	<b>7</b>
3.1	Allgemeines und Fehlerkodierungen	7
3.2	CT-API für die serielle Schnittstelle (COM)	8
3.3	CT-API für USB	9
3.4	CT-API mit COM/USB-Adaption	11
3.5	CT-API mit „well known identifier“	11
<b>4.</b>	<b>KVK und eGK</b>	<b>12</b>
4.1	Spezielle Hinweise für den Ablauf	12
4.2	Lesen der Krankenversichertenkarte (KVK) für <i>stationär</i>	14
4.3	Lesen der Krankenversichertenkarte (KVK) für <i>mobil</i>	14
4.4	Lesen der elektronischen Gesundheitskarte (eGK) für <i>stationär</i>	15
4.5	Lesen der elektronischen Gesundheitskarte (eGK) für <i>mobil</i>	17
4.6	Interpretation der eingelesenen Daten der elektronischen Gesundheitskarte	18

Dieser Dokumentation liegen folgende Versionen der Produkte zugrunde:

CT-API: 2.6.0  
(ct\_api\_com.dll , ct\_api\_usb.dll)

*CARD STAR /medic2*: M1.20G, M1.20F oder M1.50G

*CARD STAR /memo2*: P1.11H oder P1.11I

# 1. **CARD STAR /medic2**

## 1.1 **Einsatzumgebungen**

*CARD STAR /medic2* ist ein Chipkartenlesegerät (Terminal), das für mehrere Einsatzfälle konzipiert ist.

*CARD STAR /medic2* ist gemäß der Spezifikation "MKT – **M**ultifunktionales **K**arten**T**erminal" in der Version 1.0 konzipiert (Spezifikation der TeleTrusT Deutschland e.V. vom 15.04.1999). Es arbeitet dann in einer Applikation, die im PC abläuft und sich des MKT-Standards bedient.

*CARD STAR /medic2* ist ein Versichertenkarten-Lesegerät, wie von der **K**assenärztlichen **B**undes**V**ereinigung (KBV) definiert.

*CARD STAR /medic2* arbeitet auch bereits mit der **e**lektronischen **G**esundheits**K**arte (eGK) als BCS-Terminal (gemäß Spezifikation der Gematik mit dem gleichen Befehlssatz wie ein MKT).

Für den späteren Einsatz besitzt es bereits die Schaltungskomponenten für den Betrieb an einem Computer-Netzwerk (Ethernet). In diesem Fall wird es nach der Spezifikation SICCT (**S**ecure **I**nteroperable **C**hip **C**ard **T**erminal) arbeiten, als sogenanntes "SICCT" oder "eHealth-Terminal".

*CARD STAR /memo2* ist ein mobiles Kartenlesegerät als Ergänzung zum *CARD STAR /medic2*. Es kann im mobilen Betrieb Kartendaten speichern und kann diese später an einen PC ausgeben, wenn es im *CARD STAR /medic2* eingesteckt ist. Vergleiche Abschnitt 1.7.

## 1.2 **Terminalbetrieb**

*CARD STAR /medic2* arbeitet grundsätzlich im Remotebetrieb über eine der drei Schnittstellen und wird logisch gesteuert von einem Primärsystem, das auf dem Host läuft (Verwaltungsprogramm für Praxis, Klinik, Apotheke usw.) .

Das Terminal hat nur sehr eingeschränkte lokale Bedienmöglichkeiten:

- Einstellung der Geräteparameter. Einige dieser Parameter sind auch für die Ansteuerung über eine Schnittstelle von Bedeutung.
- Eine Erkennung und entsprechende Anzeige ausgewiesener Kartentypen (ab Version M1.01)
- Anzeige- und Druck der Daten von Versichertenkarten (KV-Karten) und der elektronischen Gesundheitskarte (ab Version M1.10)
- Vergleichbar Anzeigefunktion für das mobile Terminal *CARD STAR /memo2*, sowie hierfür eine Selektions- und Löschfunktion

### 1.3 Einstellung der Terminalparameter

Folgende Parameter sind für die Ansteuerung über eine Schnittstelle maßgeblich:

Liste der Einstellparameter			
Parameter	Gruppe	Abschnitt in der Anleitung M1.2x	Abschnitt in der Anleitung M1.5x
PC-Datenformat	KVK	5.6	5.6
Kartensteckfunktion	KVK	5.7	5.7
Schnittstelle	Host/PC	7.2	8.2
Baudrate, seriell	Host/PC	7.3	8.3
Protokoll-Timing	Host/PC	7.4	8.4
Werkseinstellung	Werkseinst.	7.6	8.7
CARD STAR /memo2	siehe Kapitel 5 der Anleitung zum CARD STAR /memo2		

Die Parameter sind in der Bedienungsanleitung des *CARD STAR* /medic2 (Version M1.2x oder M1.5x) beschrieben. Der jeweilige Abschnitt dort ist in der Tabelle ausgewiesen.

Die Menüpunkte der Gruppen „Host/PC“ und „Werkseinstellung“ sind Administratorfunktionen, die unter der Tastenkombination **F 9** erreichbar sind. Bei der BCS-Version M1.5x ist hierzu die Administrator-PIN einzugeben.

Die Punkte der Gruppe KVK (bzw. alle Einstellungen für ältere Versionen, also bis M1.1x) sind in **F 0** enthalten.

### 1.4 Kartensteckfunktion

Wenn die Kartensteckfunktion aktiviert ist, so wird ohne Befehl vom Host versucht, die Karte zu lesen und zu interpretieren.

Nur die KV-Karte und die eGK können jedoch verarbeitet werden. Alle anderen Kartentypen werden wie folgt behandelt (Terminalversion ab M1.01 bzw. M1.10):

Sowohl Speicher- als auch Prozessorkarten werden geprüft. Falls die Merkmale einer bekannten Karte erkannt werden, erfolgt eine entsprechende Anzeige der Kartenapplikation, zum Beispiel: "GeldKarte 29,20 EUR"

Kann die Karte nicht zugeordnet werden, so wird "kein gültiger Kartentyp" oder "keine gültige KV-Karte" angezeigt.

Fehlerhafte oder unlesbare Karten erzeugen weiterhin eine Fehleranzeige.

Nur die KV-Karte und die eGK führen zur Anzeige oder zum Druck des Karteninhaltes.

## 1.5 Schnittstellenparameter

Die Schnittstelle muss entsprechend der gewünschten Verbindung auf "seriell" oder "USB" gestellt werden. Bei der Einstellung "seriell" ist zusätzlich die Baudrate zu konfigurieren.

Die LAN-Schnittstelle ist zur Zeit noch nicht in Betrieb, sondern für den zukünftigen Einsatz im SICCT-Terminal vorgesehen.

## 1.6 Protokoll-Timing

Das *CARD STAR* /medic2 ist zum einen ein KVK-Lesegerät, ebenso wie das *CARD STAR* /medic1 der alten Generation. Andererseits ist es ein „Multifunktionales Kartenterminal“, wie es auch für die elektronische Gesundheitskarte eingesetzt wird.

Nach beiden zugrunde liegenden Spezifikationen ist die Funktionalität „Waiting Time Extension (WTX)“ definiert und somit obligatorisch. Entsprechend der Validierung durch den TÜViT und Listung durch die KBV ist das die zugelassene Einstellung.

Celectronic hat beim Test diverser Software-Produkte festgestellt, dass einige Programme WTX nicht unterstützen. Offensichtlich ist in diesen Systemen die Funktionalität WTX gar nicht implementiert. Daraufhin ist diese Funktionalität in unserem Gerät abschaltbar gestaltet worden (Einstellung hierzu: "nicht aktiv").

## 1.7 *CARD STAR* /memo2

*CARD STAR* /memo2 ist ein mobiles Chipkartenlesegerät (Terminal) mit Speicherfunktion für die KV-Karte und die eGK.

*CARD STAR* /medic2 dient dabei als Dockingstation, da das *CARD STAR* /memo2 keine externen Standardschnittstellen besitzt. Sobald ein *CARD STAR* /memo2 in der Aufnahme steckt, schaltet das *CARD STAR* /medic2 automatisch auf Dockingbetrieb und arbeitet nicht mehr als stationäres Kartenterminal. Die gesamte Bedienung zum Auslesen gespeicherter Datensätze erfolgt über die Dockingstation, ebenso zum Anzeigen und Ausdrucken der Daten.

Die Bedienung ist detailliert in der Bedienungsanleitung des *CARD STAR* /memo2 beschrieben, besonders wird auf den Abschnitt 4.7 "Datenübertragung zum Host" hingewiesen, sowie auf den Einstellparameter "Selektion in F6" (Abschnitt 5.4).

**Es wird im Folgenden vorausgesetzt, dass die Geräte so vorbereitet sind, dass das Auslesen ferngesteuert möglich ist:**

- **Geräte sind physikalisch und logisch verbunden (gepairt).**
- **Mindestens ein Datensatz ist im *CARD STAR* /memo2 gespeichert.**
- **Funktion ist F6 angewählt** (ggf. automatisch).
- **Selektion ist erfolgt** (ggf. implizit).
- **Somit erscheint der Text in der Anzeige: "...remote".**

### Hinweis zur Anleitung des *CARD STAR* /memo2, nur Ausgabe 2008:

Nur durch ein Einlegeblatt in der) wird auf folgende, zusätzliche Möglichkeit der Terminaleinstellung hingewiesen:

Einstellung „**Selektion in F6**“ → „**nie + F6 Start**“:

Allein durch Einsetzen des *CARD STAR* /memo2 ist bereits nach den beiden Meldungen „Version ...“ und „Datensätze = ...“ die Datenübertragung möglich, das spart also die Anwahl von F6 und die Selektion.

## 2. Ansteuerung vom Host

Die Befehlsschnittstelle für ein **Multifunktionales KartenTerminal** ist in der MKT-Spezifikation festgelegt.

Prinzipiell gibt es auf zwei Ebenen nutzbare Schnittstellen:

1. Geräteschnittstelle
2. CT-API (Card Terminal Application Programming Interface)

### 2.1 Geräteschnittstelle

Es werden alle Schichten der Schnittstelle definiert. Im PC wird entsprechend die physikalische Schnittstelle programmiert. Dabei kann auf geeignete im Betriebssystem vorhandene Treiber zurückgegriffen werden.

Bei dieser Realisierung spricht man auch von "Steckerkompatibilität". PCs und Terminals ließen sich beliebig kombinieren, ohne dass in einem der beiden Geräte Anpassungen nötig sind. Bei ausgewiesenen Optionen sind Einstellungen nötig, wie zum Beispiel beim Datenformat.

**Im Fall des MKT trifft diese Situation ausschließlich auf die serielle Schnittstelle zu !**

Die von der Kassenärztlichen Bundesvereinigung spezifizierte serielle Schnittstelle ist in das MKT-Konzept übernommen worden. Hieraus resultiert auch die Beschränkung auf 9.600 Baud.

## 2.2 CT-API

In diesem Konzept liefert der Terminalhersteller Software, die von dem Programm des Applikationsherstellers genutzt wird.

Dieses Verfahren hat mehrere Vorteile:

- Proprietäre Anpassungen auf der Geräteschnittstelle werden ermöglicht. Dazu zählen auch Optimierungen, wie zum Beispiel die Baudratenumstellung bei der seriellen Schnittstelle.
- Die Programmierung durch den Applikationsentwickler wird einfacher, da die hardwarenahen Steuerungen durch die Software des Terminalherstellers bereits umgesetzt sind.
- Im Allgemeinen bleibt die Applikation unverändert, wenn auf eine andere physikalische Schnittstelle übergegangen wird, Beispiel: Verwendung der USB-Schnittstelle.

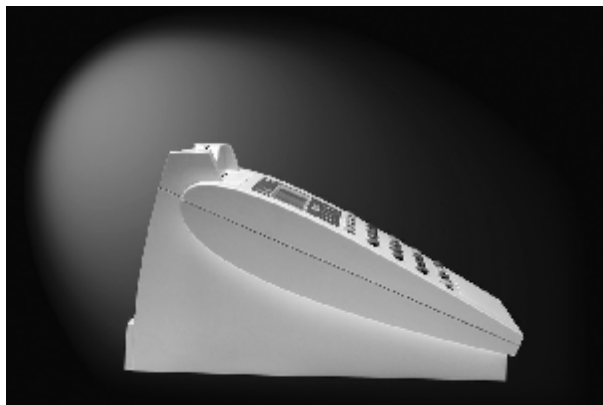
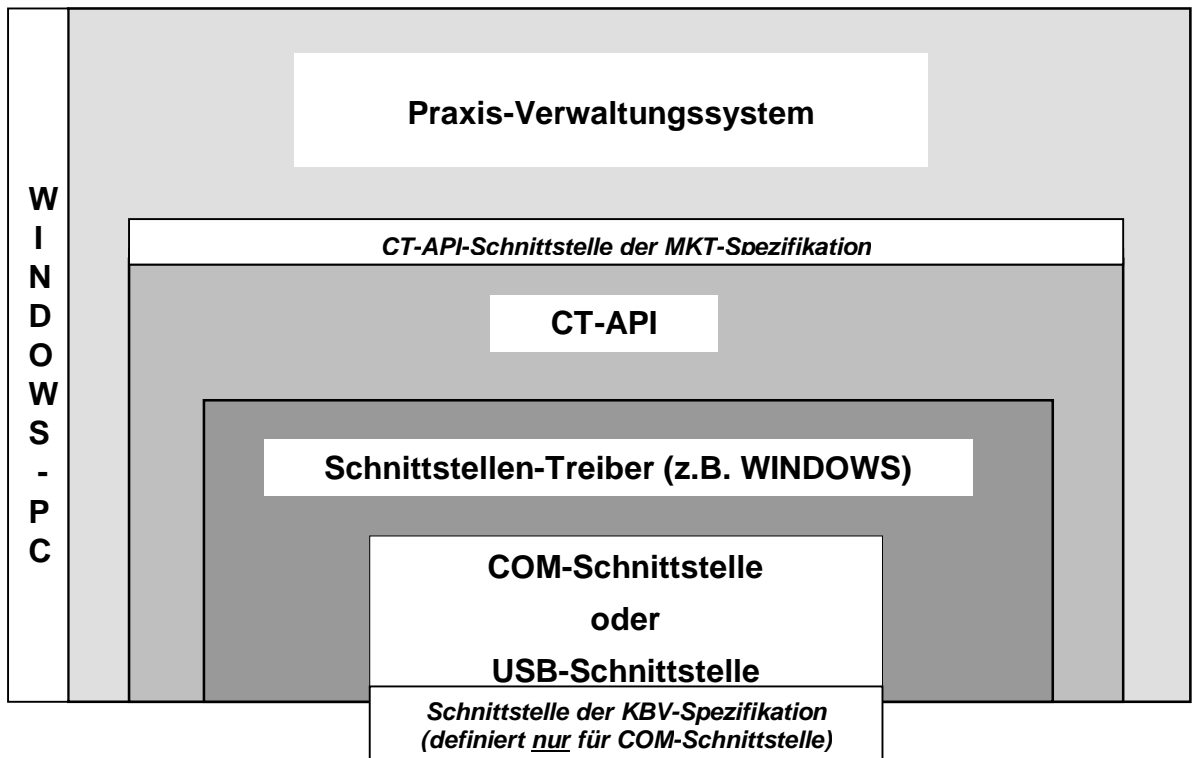
Dieser Ansatz hat auch Nachteile:

- Es muss eine geeignete Schnittstelle definiert sein.
- Der Terminalhersteller leistet den Aufwand, das Softwaremodul zu entwickeln.
- Das Softwaremodul ist von den Gegebenheiten des Zielsystems abhängig, insbesondere vom Betriebssystem.
- Die Steckerkompatibilität besteht nicht mehr. Bei Austausch des Terminals ist ggf. auch die spezifische Software auszutauschen.

In der MKT-Spezifikation (siehe Abschnitt 1.1) wird eine CT-API (**C**ard **T**erminal **A**pplication **P**rogrammer Interface) definiert.

Celectronic hat diese Software für WINDOWS Betriebssysteme umgesetzt und stellt jeweils eine CT-API für die serielle Schnittstelle und für USB zur Verfügung.

Weitere Ausführungen sind für Linux und Mac OS verfügbar. Diese sind jedoch nicht Inhalt dieser Beschreibung.





## 3. CT-API für WINDOWS

### 3.1 Allgemeines und Fehlerkodierungen

Zwei CT-API werden zur Verfügung gestellt:

1. ct\_api\_com für die serielle Schnittstelle
2. ct\_api\_usb für USB

#### Ethernet

Eine CT-API für Ethernet ist nicht vorgesehen, auch wenn das *CARD STAR /medic2* mit einer LAN-Schnittstelle und entsprechender Ansteuerung ausgestattet ist. Ein MKT ist nicht netzwerkfähig.

Vielmehr wird die Erweiterung in Form der Konvertierung zu einem SICCT (Secure Interoperable Chip Card Terminal) stattfinden. Das Netz basiert dann wiederum auf einer komplett definierten Geräteschnittstelle, siehe Abschnitt 2.1.

#### Aufrufe

Die Aufrufdefinitionen der CT-API sind in der MKT-Spezifikation, Teil 3 dargestellt. Diese Spezifikation wird im Folgenden als bekannt vorausgesetzt.

#### Adressierung

Es ist wichtig, dass beim Kommando zu einer Befehlsausführung ( ct\_data ) die Adressen korrekt angegeben werden: Source ist immer der Host, die Adresse also immer '02'. Destination ist entweder das Kartenterminal CT (Kodierung '01') oder ein Kartenslot, somit die Karte:

Kodierung '00' für Slot1, das ist der vordere oder einzige Slot des *CARD STAR /medic2* (Kartenformat: ID-0)

Kodierung '02' für Slot2, das ist der hintere Slot des *CARD STAR /medic2* (Kartenformat: ID-0)

Kodierungen '03' und '04' für die Plug-In-Slots (Kartenformat ID-000)

#### Fehlerkodierungen

Die Hinweise auf die Fehlerkodierungen sind in der Spezifikation sehr knapp. In der CT-API von Celectronic gelten folgende erweiterte Interpretationen:

Wert	Name	Fehlerbedeutung
-1	ERR_INVALID	<ul style="list-style-type: none"> <li>• Aufrufparameter einer Funktion inhaltlich falsch (z.B. Schnittstellenummer V24=0)</li> <li>• Aufruf von CT_close ohne vorheriges CT_init</li> <li>• Mehrmaliges CT_init ohne CT_close</li> <li>• Logische Terminal-Nummer eines Kommandos entspricht nicht der Nummer von CT_init</li> </ul>
-8	ERR_CT	<ul style="list-style-type: none"> <li>• WTX abgelaufen</li> <li>• Anzahl der möglichen Befehlswiederholungen abgelaufen</li> </ul>
-10	ERR_TRANS	<ul style="list-style-type: none"> <li>• Timeout- Fehler beim Senden/Empfang eines Datenblockes</li> <li>• Angeschlossenes Terminal ist nicht betriebsbereit</li> <li>• andere Fehler beim Schreiben/Lesen</li> <li>• Kanal schon durch andere Applikation/DLL belegt</li> </ul>
-11	ERR_MEMORY	Puffer der Applikation zu klein für die empfangene Datenmenge
-127	ERR_HOST	<i>nicht verwendet:</i> kein Funktionsabbruch möglich
-128	ERR_HTSI	Protokollfehler: <ul style="list-style-type: none"> <li>• Checksummenfehler</li> <li>• PCB der Antwort falsch/unpassend</li> <li>• Adressen der Antwort unpassend</li> <li>• andere Protokollfehler</li> </ul>

Hinweis: Wird der Wert (signed character) abweichend interpretiert, so ergeben sich andere Angaben, z.B. 'FF' für -1 oder '80' statt -128.

### 3.2 CT-API für die serielle Schnittstelle (COM)

Die ct\_api\_com benutzt diverse Programmteile des Betriebssystems, wie zum Beispiel den universellen COM-Treiber von WINDOWS.

In der Standardvariante ist ct\_api\_com für WINDOWS 2000, WINDOWS XP und VISTA geeignet. Für ältere WINDOWS-Versionen ist diese Ausführung ungeeignet.

#### WINDOWS 98

Für WINDOWS98 gibt es eine gesonderte Ausführung der ct\_api\_com. Auf die Lieferung dieser Variante wird in Zukunft verzichtet.

#### WINDOWS 95, WINDOWS ME und ältere Versionen

Celectronic bietet hierfür keine spezifischen Ausführungen der CT-API an und hat auch die existierenden CT-APIs nicht mit diesen Betriebssystemen getestet. Entsprechend können diese Kombinationen nicht empfohlen werden. Ein entsprechender Support von Celectronic ist ausgeschlossen!

### Einstellung der Baudrate

Die Baudrate für die Kommunikation mit dem Terminal entspricht anfangs dem Default-Wert der CT-API, im Standardfall also 9.600 Baud. Es gibt einen proprietären Befehl zur Umstellung der Baudrate, dieser wird zusätzlich zur MKT-Spezifikation angeboten.

Funktion zur Umstellung der Baudrate:

*CT\_param*

mit Aufruf:

*char CT\_param(long param1, short \*version)*

Eingabeparameter *param1*: Baudrate (Werte: 9600,19200,38400,57600,115200)

Ausgabeparameter: *version* Version der DLL (dezimal), z.B.: 2410 für 2.4.1.0

Rückgabe: 0 okay

ERR\_INVALID = -1 Eingabe-Wert ist für Baudrate nicht erlaubt

### **3.3 CT-API für USB**

Die USB-Schnittstelle bedarf eines eigenen, spezifischen Treibers *celmkt*, der ebenfalls zur Verfügung steht (Celectronic-Web-Seite), aktuell ist die Version 2.51.

In der neuesten Version wird der Treiber mit einem Installer geliefert. Dieser sorgt insbesondere dafür, dass keine Mischung verschiedener Treiberversionen registriert werden können. Die Installation hat sich damit vereinfacht, es ist lediglich das Programm „setup USB251.exe“ aufzurufen. Das Terminal sollte dabei angeschlossen sein (USB-Betrieb). Es wird im Lauf der Installation das Entfernen und erneute Anschließen des Gerätes verlangt. Die Installation erfordert Administratorrechte.

Es steht ein zweiter USB-Treiber zur Verfügung der eine serielle Schnittstelle simuliert, siehe hierzu Abschnitt 3.4.

Für eine konventionelle Installation gilt Folgendes (nicht empfohlen):

Der Treiber besteht aus zwei Dateien für die aktuellen Betriebssysteme WINDOWS 2000, WINDOWS XP, VISTA und WINDOWS7 (außerdem separate Variante für 64-Bit-Systeme):

*celmkt.inf*

*celmkt.sys*

Sobald ein *CARD STAR/medic2* betriebsbereit über die USB-Schnittstelle mit dem Host verbunden wird, fordert das Betriebssystem den Treiber an. Über die manuelle Installation kann dann der Treiber geladen werden. Hierzu ist das Verzeichnis anzugeben, in dem der Treiber gespeichert worden ist. In jedem Fall sind Administratorrechte erforderlich.

Die Ausführungen für WINDOWS98 (*celmkt98.inf* und *celmkt98.sys*) werden in Zukunft nicht mehr ausgeliefert!

Bei VISTA und WINDOWS7 wird beim Installieren eine Warnung ausgegeben, dass der Treiber keine Signatur enthält. Diese Warnung ist zu bestätigen.

### Terminal-USB-Identität

Jedes Terminal besitzt aus Sicht des PC-Betriebssystems eine Eindeutigkeit durch die Seriennummer des Gerätes.

Wird also das Terminal gewechselt, so ist wieder eine neue Treiberinstallation erforderlich.

### Verwaltung von mehreren Terminals

Vorausgesetzt sei an dieser Stelle, dass für alle über USB angeschlossenen Terminals auch die Treiber installiert worden sind (siehe voriger Absatz).

Der Parameter ‚Port-Nummer‘ *pn* der Funktion <CT\_init> wird bei USB generell ignoriert (automatische Zuordnung durch Enumeration).

Nach dem Laden der CT-API und Aufruf von <CT\_init> mit der logischen Terminalnummer *ctn* wird das erste (willkürlich) vom Betriebssystem gefundene und betriebsbereite CARD STAR Terminal enumeriert (registriert) und diesem Wert von *ctn* fest zugeordnet. Eine benutzerkontrollierte Zuordnung bei mehreren angeschlossenen Terminals ist nicht möglich. Hierzu wäre eine Erweiterung der Schnittstelle der CT-API erforderlich, z.B. mit Übergabe einer Seriennummer.

Hinweis: Ein einfacher Weg zur kontrollierten Zuordnung ist, nacheinander jeweils nur ein neues (noch nicht registriertes) Terminal anzuschließen, so dass bei <CT\_init> auch nur dieses gefunden werden kann. Die Zuordnung muss vom aufrufenden System oder dem Benutzer verwaltet werden.

Die beschriebene feste Zuordnung hat folgende Konsequenzen:

- Wiederholtes <CT\_init> mit der gleichen Terminalnummer *ctn* führt zum Fehler ERR\_TRANS.
- Weitere Terminals können mit jeweils einer anderen Nummer *ctn* zur simultanen Nutzung administriert werden
- Nach <CT\_close> und erneutem <CT\_init> mit der gleichen Terminalnummer *ctn* wird auch dasselbe Terminal wieder angesprochen (sofern noch verfügbar).
- Andererseits kann ein bereits registriertes Terminal nicht mit einer anderen Terminalnummer *ctn* erreicht werden.

Wird die CT-API beendet und wieder neu geladen, so sind alle früheren Registrierungen hinfällig.

Vor jedem Senden eines Kommandos, prüft die DLL, ob das USB-Terminal noch enumeriert ist und registriert es dann ggf. neu.

### Hinweis:

Die CT-API ist mit Zusatzfunktionen ausgestattet, die es erlauben, den geschilderten Automatismus zu umgehen und manuell Zuordnungen zu kontrollieren. Die Darstellung dieser Methode sprengt jedoch den Rahmen dieser Beschreibung, bitte ggf. gesondert anfragen.

### 3.4 CT-API mit COM/USB-Adaption

Wird der USB-Treiber *celmktcom* verwendet (aktuell Version 1.82), so ist die CT-API für die serielle Schnittstelle (COMx) zu benutzen (*ct\_api\_com*), obwohl das Terminal an der USB-Schnittstelle angeschlossen ist.

Dieser Treiber wird mit dem gleichen Installationsmechanismus geladen, der Aufruf lautet „setup virtCOM182.exe“. Über den Gerätetreiber muss die Port-Nummer ermittelt werden:

Die Portnummer x der Schnittstelle (COMx) wird für die Einstellung des Primärsystems benötigt. Sie wurde bei der Installation des Treibers vom Betriebssystem automatisch zugewiesen und kann aus den Parametern der Systemsteuerung (Gerätetreiber: Anschlüsse COM und LPT) ermittelt werden. Die Baudrateneinstellung spielt keine Rolle.

Hinweis: Es wird dringend davor gewarnt, den Standard-USB-Treiber gegen den virtCOM-Treiber – oder vice versa - mit einer konventionellen Installation auszutauschen. Über das Installationstool ist der Wechsel dagegen kein Problem.

#### Wechsel des USB-Treibers (*celmkt* gegen *celmktcom* oder umgekehrt)

Die empfohlene Vorgehensweise ist das Deinstallieren des aktuellen Treibers und Neuinstallation. Nicht empfohlen ist das Aktualisieren des Treibers. In jedem Fall sind Administratorrechte erforderlich.

#### Anwendungsbeispiel

Das Primärsystem verlangt eine serielle Schnittstelle, der PC (Notebook) bietet jedoch nur USB-Schnittstellen.

### 3.5 CT-API mit „well known identifier“

Der Anhang von Teil 3 der MKT-Spezifikation sieht die Variante mit einem „well known identifier“ als Option vor, um mehrere CT-APIs unterschiedlicher Hersteller zeitgleich nutzen zu können.

Celetronic stellt auch diese Variante zur Verfügung (nicht für W98 und 64-Bit-Systeme und nicht mit COM/USB-Adaption). Die Versionierung für dieses Derivat ist identisch mit der regulären Version der CT-API, zur Zeit 2.6.0.

Gemäß der MKT-Spezifikation gelten folgende Konventionen:

Dateiname für serielle Schnittstelle:     **ctCEL20C.dll**

Dateiname für USB-Schnittstelle:       **ctCEL20U.dll**

Funktionsaufrufe:                         **ctDECEL20\_...**  
   **(ctDECEL20\_init, ctDECEL20\_data, ctDECEL20\_close)**

## 4. KVK und eGK

Der Fokus der MKT-Anwendungen liegt derzeit beim Auslesen der Daten aus der Karte des Patienten. Hierbei unterscheidet sich die auf Speicherkarten (KVK) basierende Technik grundsätzlich von der sicherheitsbetonten neuen Generation der elektronischen Gesundheitskarte (eGK). Beide Kartentypen ermöglichen jedoch eine lokale Anzeige- und Druckfunktion im Terminal (ab Terminalversion M1.10).

Die Kommandosequenzen der beiden Techniken sind verschiedenartig.

Die Kommandos an die eGK werden inhaltlich ungeprüft an die Karte weitergeleitet und die Antwort ebenso transparent an den Host zurückgeliefert. Das Entpacken und Interpretieren der Daten steht in der Verantwortung des Primärsystems. Dagegen werden die Versichertendaten der KVK im Terminal geprüft.

Entsprechend den Spezifikationen sind beim stationären Terminal *CARD STAR /medic2* und beim Mobilgerät *CARD STAR /memo2* die Methoden zum Auslesen der Daten unterschiedlich. Das ergibt sich aus der unterschiedlichen Technik: Leser von Kartendaten beim Stationärgerät - Lesen eines gespeicherten Datensatzes beim Mobilterminal.

Insbesondere soll darauf aufmerksam gemacht werden, dass sich auch die Lesetechnik für die eGK unterscheidet:

- *CARD STAR /medic2*:  
Es werden blockweise nur jeweils 256 Byte aus der Karte ausgelesen, wobei ab dem zweiten Block ein entsprechender Offset anzugeben ist (Kommunikation zwischen Primärsystem und Karte). Es hängt von den Fähigkeiten der Karte ab, ob das Verfahren für das Mobilgerät (s.u.) auch funktionieren würde.
- *CARD STAR /memo2*:  
Über einen Befehl mit "*extended length*" werden alle gespeicherten Daten eines Datensatzes in einem einzigen großen Block ausgelesen (Kommunikation zwischen Primärsystem und Mobilterminal). Das Offset-Verfahren - wie für die stationären Geräte favorisiert - ist nicht verwendbar.

### 4.1 Spezielle Hinweise für den Ablauf

#### Grundsätzlich

Jedes Terminal- oder Karten-Kommando wird in den Befehl *ct\_data* eingebettet. Für Start und Ende der Kommunikation sind die Befehle *ct\_init* und *ct\_close* zu verwenden.

Die Zieladresse ***dad*** muss der Funktion *ct\_data* als Parameter übergeben werden (zusammen mit der Quelladresse '02'). Auf die korrekte Angabe der Zieladresse ist zu achten, da sie nicht für alle Befehle identisch ist: Es wird entweder das Terminal oder eine Karte angesprochen.

Es bestehen Unterschiede bei den Kommandos und Antworten für stationäre und Mobilgeräte. Gegebenfalls wird im Folgenden wie folgt unterschieden:

- "stationär" zutreffend für das CARD STAR /medic2
- "mobil" zutreffend für das CARD STAR /memo2, also wenn das Mobilteil in das CARD STAR /medic2 eingesetzt ist und die Funktion F6 aufgerufen ist

Fehlerfrei bearbeitete Befehle werden in der Regel mit dem Fehlercode ,90 00' beantwortet. Diese Antwort ist in den folgenden Tabellen nicht mehr ausdrücklich aufgeführt.

### RESET CT

In jedem Fall ist es empfehlenswert, vor der eigentlichen Kommandosequenz ein RESET CT an das Terminal zu senden, allein schon, um den Terminaltyp automatisch zu erkennen:

	Kommando	dad	APDU	Bemerkung
	Reset CT	01	20 11 00 00 00	Antwort <i>stationär</i> : '90 00' Antwort <i>mobil</i> : '95 00'

### REQUEST ICC

#### **stationär**

Das Einsetzen der Karte wird mit dem Kommando REQUEST ICC umgesetzt. Das Kommando sieht einen Zeitparameter vor. Entsprechend lange wartet das Terminal auf die Karte, in Folge dessen auch der Host auf die Antwort. In dieser Wartezeit erfolgt eine Eingabeaufforderung auf der Terminalanzeige, optional mit einem Text, den der Host vorgegeben hat.

	Kommando	dad	APDU	Bemerkung
	Request ICC1	01	20 12 01 00 01 0a	10 Sekunden Wartezeit

#### Methode zum Pollen der Kartenpräsenz:

Eine Besonderheit ist es, diesen Parameter auf Null zu setzen. Dann wird die Eingabeaufforderung unterdrückt und die Antwort kommt sofort. Mit dieser Methode kann der Host den Kartenslot permanent abtesten (pollen), zum Beispiel im Zyklus von einer Sekunde. Eine Textausgabe auf der Terminalanzeige muss dann allerdings nach jedem Pollen mit einem Befehl OUTPUT durchgeführt werden, da die Anzeige durch den Befehl REQUEST wieder gelöscht wird.

	Kommando	dad	APDU	Bemerkung
	Request ICC1	01	20 12 01 00	ohne Wartezeit (1. Variante)
	Request ICC1	01	20 12 01 00 01 00	ohne Wartezeit (2. Variante)

#### **mobil**

Das Kommando bezieht sich auf den Datenspeicher im CARD STAR /memo2. Sinngemäß kommt eine Fehlerantwort ('62 00' = „keine Karte eingesetzt“), wenn kein Datensatz existiert, also der Speicher leer ist.

Die Angabe der Wartezeit hat für das Mobilgerät nur formalen Charakter, eine Antwort ist generell sofort verfügbar. Nur die beiden gelisteten Varianten sind korrekt und werden akzeptiert.

	Kommando	dad	APDU	Bemerkung
	Request ICC1	01	20 12 01 00	1. Variante
	Request ICC1	01	20 12 01 00 01 01	2. Variante

Hinweis: Die Kommandosequenz '20 12 01 00 01 00' ist in der Spezifikation der Gematik z.T. noch gelistet, aber aktuell nicht mehr zulässig!

#### 4.2 Lesen der Krankenversichertenkarte (KVK) für *stationär*

Hinweis: Die Antwort '90 00' beim Befehl REQUEST ICC bezeichnet eine Synchronkarte, so dass dann eine KVK angenommen werden sollte.

Bereits beim Einschalten der Karte (Kommando REQUEST ICC) erkennt das CARD STAR /medic2, dass es sich um eine KVK handelt und aktiviert intern einen entsprechenden Status. In Folge prüft das Terminal die Konsistenz der Versichertendaten entsprechend der Spezifikation der KBV. Ebenso wird das Schreiben (Kommando UPDATE) in diesem Status als fehlerhaftes Kommando abgewiesen.

Die folgenden Terminal- bzw. Karten-Kommandos führen zur Ausgabe der Versichertendaten an den Host (es wird der vordere Kartenschlitz adressiert):

	Kommando	dad	APDU	Bemerkung
1	Request ICC1	01	20 12 01 00	ohne Wartezeit (alternativ: mit Wartezeit)
2	Select File (KVK)	00	00 a4 04 00 06 d2 76 00 00 01 01	
3	Read Binary	00	00 b0 00 00 00	Antwort: Gesamtes KVK-Template + Response Code '62 82'
4	Eject ICC1	01	20 15 01 00	ohne Wartezeit (alternativ: mit Wartezeit)

3	Read Binary	00	00 b0 00 00 00 00 00	Alternative mit "extended length"
---	-------------	----	----------------------	-----------------------------------

#### Dateninterpretation

Bei den Versichertendaten handelt es sich um eine Darstellung im Format ASN.1. Die Anzeige bei aktivierter Kartensteckfunktion oder den lokalen Funktionen basiert auf der Interpretation dieser Daten. Zur Alternative "Festformat" wird auf die KBV-Spezifikation verwiesen.

#### 4.3 Lesen der Krankenversichertenkarte (KVK) für *mobil*

Hinweis: Die Antwort '90 00' beim Befehl REQUEST ICC weist darauf hin, dass der Datensatz aus dem Einlesen einer KVK entstanden ist.

Die folgenden Terminal- bzw. Karten-Kommandos führen zur Ausgabe der Versichertendaten an den Host



Adressierung: Es existiert nur die Adresse '00' (dad) für den Datenspeicher („virtuelle Karte“), sowie '01' als Terminal-Adresse.

Innerhalb der Befehle (Request, Eject) ist auch nur die Slot-Adresse '01' zulässig.

	Kommando	dad	APDU	Bemerkung
1	Request ICC1	01	20 12 01 00	ohne Wartezeit (alternativ: mit Wartezeit nur Wert 01)
2	Select File (KVK)	00	00 a4 04 00 06 d2 76 00 00 01 01	
3	Read Binary	00	00 b0 00 00 00	Antwort: Gesamtes KVK-Template + Response Code '90 00'
			Die Antwort aus READ BINARY ist das übliche Versichertendaten-Template, ergänzt durch drei Elemente: 1. Tag '91': Einlesedatum 2. Tag '92': Zulassungsnummer des mobilen Terminals 3. Tag '93': Checksumme über die Tags '91', '92' und Tag+Länge von '93'	
4	Erase Binary	00	00 0e 00 00	Löschen des soeben übertragenen Datensatzes
5	Eject ICC1	01	20 15 01 00	ohne Wartezeit (alternativ: mit Wartezeit nur Wert 01)

3	Read Binary	00	00 b0 00 00 00 00 00	Alternative mit "extended length"
---	-------------	----	----------------------	-----------------------------------

#### Dateninterpretation

Bei den Versichertendaten handelt es sich um eine Darstellung im Format ASN.1. Die Anzeige bei aktivierter Kartensteckfunktion oder den lokalen Funktionen basiert auf der Interpretation dieser Daten. Zur Alternative "Festformat" wird auf die KBV-Spezifikation verwiesen.

#### 4.4 Lesen der elektronischen Gesundheitskarte (eGK) für *stationär*

Das Terminal arbeitet konsequent als MKT, einen besonderen Status gibt es nicht.

Hinweis: Die Antwort '90 01' beim Befehl REQUEST ICC bezeichnet eine Asynchronkarte, so dass dann eine eGK angenommen werden sollte.

Die folgenden Terminal- bzw. Karten-Kommandos führen zum Auslesen der Versicherten- und Personendaten (es wird der vordere Kartenschlitz adressiert):

	Kommando/Schritt	dad	APDU	Bemerkung
1	Request ICC1	01	20 12 01 00	ohne Wartezeit (alternativ: mit Wartezeit)
2	Select File (HCA)	00	00 a4 04 0c 06 d2 76 00 00 01 02	
3	Read Binary (Status)	00	00 b0 8c 00 00	Antwort: Kartenstatus (25 Byte) + Response Code '90 00'
			Die Antwort aus READ STATUS muss geprüft werden: Das erste Byte muss den Wert '30' haben. Anderenfalls dürfen die Kartendaten nicht verwendet werden!	
4	Read Binary (Personal Data)	00	00 b0 81 00 00	Personendaten (max. 256 Byte) + Response Code '90 00'
5	Read Binary (Subsequent Data)	00	00 b0 01 00 00	Folgende Daten (max. 256 Byte) mit Offset 256 + '90 00'
6	Read Binary (Subsequent Data)	00	00 b0 0n 00 00	Folgende Daten (max. 256 Byte) mit Offset n*256 + '90 00'
			Die Daten aus den einzelnen Antworten werden zu einer Datei zusammengesetzt, um weiter verarbeitet werden zu können. Sind keine Daten mehr lesbar, antwortet die Karte mit '6b 00' (Offset falsch).	
7	Read Binary (Insurance Data)	00	00 b0 82 00 00	Versichertendaten (max. 256 Byte) + '90 00'
8	Read Binary (Subsequent Data)	00	00 b0 0n 00 00	Folgende Daten (max. 256 Byte) mit Offset n*256 + '90 00'
			Die Daten aus den einzelnen Antworten werden zu einer Datei zusammengesetzt, um weiter verarbeitet werden zu können. Sind keine Daten mehr lesbar, antwortet die Karte mit '6b 00' (Offset falsch).	
9	Eject ICC1	01	20 15 01 00	ohne Wartezeit (alternativ: mit Wartezeit)

### Hinweise

Befehl 6 und 8 werden mit fortschreitendem Offset so oft wiederholt, bis alle Daten eingelesen sind. Hierzu ist der Wert '0n' jeweils um 1 zu erhöhen, beginnend mit '01'.

Die Alternative hierzu, das Kommando READ BINARY mit erweiterter Längenangabe ("*extended length*") zu senden, ist nicht empfehlenswert: *Extended length* ist bei der eGK optional, somit ist dieser Weg nicht für alle Karten praktikabel. Diese Alternative bietet jedoch den Vorteil, dass alle Daten in einem einzigen Block (genauer: in einer Befehlsantwort) versendet werden.

Beispiel: READ BINARY für PD mit extended length: 00 b0 81 00 00 00 00

Dateninterpretation siehe Abschnitt 4.6

#### 4.5 Lesen der elektronischen Gesundheitskarte (eGK) für mobil

Hinweis: Die Antwort '90 01' beim Befehl REQUEST ICC weist darauf hin, dass der Datensatz aus dem Einlesen einer eGK entstanden ist.

Die folgenden Kommandos führen zum Auslesen der Versicherten- und Personendaten.

Adressierung: Es existiert nur die Adresse '00' (dad) für den Datenspeicher („virtuelle Karte“), sowie '01' als Terminal-Adresse.

Innerhalb der Befehle (Request, Eject) ist auch nur die Slot-Adresse '01' zulässig.

	Kommando/Schritt	dad	APDU	Bemerkung
1	Request ICC1	01	20 12 01 00	ohne Wartezeit (oder: 20 12 01 00 01 01)
2	Select File (HCA)	00	00 a4 04 0c 06 d2 76 00 00 01 02	
3	Read Binary (Status)	00	00 b0 8c 00 00	Antwort: Kartenstatus (min. 45 Byte) + Response Code '90 00'
			Die Antwort aus READ STATUS ist eine TLV-Struktur und muss geprüft werden (gemäß der Spezifikation für mobile Terminals der Gematik): 1. Tag 'A0': Das erste Byte hierin muss den Wert '30' haben. Anderenfalls dürfen die Kartendaten nicht verwendet werden! 2. Tag '91': Einlesedatum 3. Tag '92': Zulassungsnummer des mobilen Terminals 4. Tag '93': Checksumme über die Tags '91', '92' und Tag+Länge von '93'	
4	Read Binary (Personal Data)	00	00 b0 81 00 00 00 00	Antwort: Personendaten + Response Code '90 00'
5	Read Binary (Insurance Data)	00	00 b0 82 00 00 00 00	Antwort: Versichertendaten + Response Code '90 00'
6	Erase Binary	00	00 0e 00 00	Löschen des soeben über- tragenen Datensatzes
7	Eject ICC1	01	20 15 01 00	ohne Wartezeit (oder: 20 15 01 00 01 01)

#### Hinweise

Befehl 4 und 5 verwenden "*extended length*", was für Mobilgeräte zwingend erforderlich ist. Offsets in den Parametern des Befehls sind nicht zulässig.

Dateninterpretation siehe Abschnitt 4.6

## 4.6 Interpretation der eingelesenen Daten der elektronischen Gesundheitskarte

### Hinweise:

- Dieser Abschnitt bietet nur eine Hilfestellung für die Weiterverarbeitung der eingelesenen Daten. Das Entpacken ist jedoch nicht mehr im Verantwortungsbereich des Terminalherstellers! Entpacken und Interpretieren der übertragenen Dateien sind also Aufgaben der Primärsoftware.
- Das Entzippen der beiden Datenbereiche führt zu drei (!) XML-Dateien.
- Für die Interpretation der XML-Dateien sind allein die VSDM-Dokumente der Gematik relevant. Hierfür kann durch den Terminalhersteller keine weitere Unterstützung erfolgen.

Die Basis für das Zippen ist die verbreitete Software "gzip". Diese Software wird mit dem Parameter „-d“ (für decompress) aufgerufen.

Die eingelesenen Dateien müssen jedoch zuvor noch wie folgt bearbeitet werden:

### Personendaten:

Die ersten beiden Byte bezeichnen die Länge der relevanten Daten (gezählt ab dem dritten Byte der Datei).

Diese Länge und nachfolgende Daten (meist Nullen) sollten entfernt werden, bevor entzippt werden kann.

### Versichertendaten:

Die ersten acht Byte sind Offset-Informationen:

1.+2. Byte: Startadresse der Versichertendaten

3.+4. Byte: Endadresse der Versichertendaten

5.+6. Byte: Startadresse der geschützten Daten

7.+8. Byte: Endadresse der geschützten Daten

Mit diesen Informationen können die beiden enthaltenen Dateien separiert werden.

Anschließend können sie einzeln entzippt werden.

<u>Herstellung:</u> CCV Deutschland GmbH Gewerbering 1 D-84072 Au i.d. Hallertau	<u>Vertrieb:</u> CCV Allcash ecm GmbH Eurotec-Ring 7 D-47445 Moers
Umfangreiche aktuelle Informationen bietet auch unsere Internetseite: <a href="http://www.celectronic.de">www.celectronic.de</a>	
Unser Hotline ist erreichbar unter 02841 / 1796 - 155 oder per eMail: <a href="mailto:hotline@ccv-allcash.de">hotline@ccv-allcash.de</a>	



**celectronic**  
BERLIN