

CiTouchD

Touch-Treiber für DOS

V1.0.01x

Benutzerhandbuch

Dokumenten Revision

Rev.	Beschreibung	Bearb.	Datum
001	Erste Ausgabe	wh	09.01.1996
002	Anpassung an CiTouchD V1.0.001	tt	23.01.1996
003	Anpassung an Textformat	pk	10.02.1996
004	Anpassung an CiTouchD V1.0.005	wh	27.03.1996
005	Anpassung an CiTouchD V1.0.006	wh	29.03.1996
006	Anpassung und Korrektur nach Übersetzung	pk	11.11.1996
007	Anpassung an CiTouchD V1.0.008	wh	28.01.1997
008	Anpassung an CiTouchD V1.0.009	wh	07.02.1997
009	Anpassung an CiTouchD V1.0.010, Änderung der Vorgabewerte in CITOUCHE.D	wh	14.07.1997
010	Anpassung an neue Firmenadresse	wh	29.08.1997
012	Anpassung an neues Firmenlogo	pk	25.02.1998

Haftungsausschluß

Der Inhalt dieses Handbuchs dient nur zu Informationszwecken. Die Citron GmbH behält sich das Recht vor, ohne vorherige Ankündigung, Änderungen am Inhalt dieses Handbuchs vorzunehmen. Bei der Erstellung dieses Handbuchs wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Die Citron GmbH übernimmt deshalb keinerlei Haftung für fehlende oder fehlerhafte Informationen in diesem Handbuch, oder für Fehler, die durch Anwendung der in diesem Handbuch enthaltenen Informationen entstanden sind.

Für Verbesserungsvorschläge und Hinweise auf Fehler ist die Citron GmbH dankbar.

Dieses Handbuch und die zugrundeliegende Software unterliegen dem Urheberrecht
 ©Copyright 1995 - 1998 CITRON GmbH, Anwaltinger Str. 14, 86165 Augsburg
 Tel. (0821) 74945-0 FAX (0821) 74945-99
 E-mail: info@citron.de
<http://www.citron.de>

ALLE RECHTE VORBEHALTEN

Dokumenten-Info

Dateiname: h:\manuals\citouchd\ctd_r011.doc
 Datum: 29.08.97 11:25
 Dokumenten Rev.: 2
 Dokumenten-Vorlage: h:\dformat.dot\cidoku.dot

1 Inhalt

1 Inhalt	3
2 Einführung	4
3 Aufruf von CiTouchD	4
4 Konfiguration	4
4.1 Kommandozeilenparameter	5
4.2 Vordefinierte Maustasten-Emulationsmodi	6
5 Softwareschnittstelle	11
5.1 Grundlagen	11
5.1.1 Koordinatensysteme	11
5.1.2 Unterstützte Videomodi	11
5.1.3 Benutzerdefinierte Tastenemulation	12
5.1.4 Weitere Betriebsparameter	13
5.1.5 Aufruf der API-Funktionen	17
5.2 Emulierte Maustreiberfunktionen der Int33-API	17
5.2.1 Numerische Übersicht	17
5.2.2 Besonderheiten der Int 33 API-Emulation	18
5.3 Erweiterte API-Funktionen	21
5.3.1 Ermittlung der Kennnummer von CiTouchD	21
5.3.2 Übersicht der erweiterten API-Funktionen	25
5.3.3 Referenz der erweiterten API-Funktionen	26
5.3.4 Referenz der Strukturen	37
5.3.5 Benachrichtigung durch CiTouchD	42
6 CITOUCHE.INI	42
6.1.1 [Acceleration]	42
6.1.2 [Calibration]	43
6.1.3 [Commands]	44
6.1.4 [Hardware]	45
6.1.5 [Settings]	46
6.1.6 [Sound]	48

2 Einführung

Der CiTouchD-Maustreiber (Citron Touch-Treiber für MS-DOS) ermöglicht den Einsatz eines Citron Infrarot-Touchs (im Folgenden kurz IRT genannt) als Mausersatz unter dem Betriebssystem MS-DOS. Voraussetzung für den Betrieb des CiTouchD ist MS-DOS Version 3.0 oder höher, sowie mindestens ein i386 Prozessor.

Der in CiTouchD integrierte Maustreiber bietet die Funktionen eines Standard-DOS-Maustreibers an, die zur Bestimmung der Cursorposition (Position der Touch-Unterbrechung) und der Simulation einer Maustaste nötig sind. Ein bereits zuvor installierter Treiber für eine normale Maus wird durch den CiTouchD-Treiber abgeschaltet, so daß diese Maus nicht mehr als Eingabemedium verwendet werden kann.

Bevor mit der Installation begonnen wird, muß der IRT an eine serielle Schnittstelle des Rechners angeschlossen werden. Dabei ist zu beachten, daß für den Betrieb des CiTouchD-Treibers unbedingt eine interruptfähige serielle Schnittstelle benötigt wird.

Wird eine Citron-LDVGA Grafikkarte zusammen mit der LDRI-Empfängerkarte eingesetzt ist ein geeigneter IRT-Anschluß bereits über das Verbindungskabel zur Display-Einheit erfolgt. In diesem Fall muß nur noch die Basisadresse und der Interruptkanal für die SIO1 der LDVGA-Karte eingestellt werden. Dies erfolgt über das im LDVGA-Handbuch beschriebene Utility LDVINST.

Im Gegensatz zu einer normalen Maus besteht bei CiTouchD die Möglichkeit mit absoluten Koordinaten zu arbeiten. Dabei wird die Bewegung des Fingers auf der Bildschirmoberfläche nicht in eine relative Änderung der Cursorposition umgesetzt, sondern der Cursor erscheint immer an derjenigen Stelle, an welcher der Bildschirm gerade berührt wird. Zur Erhöhung der Positioniergenauigkeit kann jedoch auch mit relativen Koordinaten gearbeitet werden. Die Umschaltung des Koordinatenmodus erfolgt dabei entweder statisch während der Konfiguration oder dynamisch im laufenden Betrieb. Dabei wird aufgrund einer speziellen Fingerbewegung, z.B. durch eine Zweifachberührung, der Koordinatenmodus gewechselt. Die größte Herausforderung bei der Emulation einer Maus durch einen Touch ist die Nachbildung der Maustasten. Es lassen sich nicht alle Eigenschaften einer Maustaste gleichzeitig und gleich gut nachempfinden. Allerdings werden je nach Anwendungsprogramm unterschiedliche Anforderungen gestellt (z.B. exaktes Timing des Tastendrucks, hohe Sicherheit gegen Fehlbedienung, Drag-And-Drop-Fähigkeit, ...), so daß meist bereits ein bestimmter Teilaspekt der Mausfunktion ausreichend für die Bedienung des jeweiligen Programms ist. Die Emulation der Maustasten ist beim CiTouchD in weiten Grenzen durch den Benutzer konfigurier- und programmierbar.

Sämtliche Parameter, welche die Arbeitsweise des CiTouchD beeinflussen, werden in einer INI-Datei im ASCII-Format abgespeichert. Alle Einstellungen lassen sich sowohl während der Installation, als auch im laufenden Betrieb verändern.

3 Aufruf von CiTouchD

Der CiTouchD-Treiber wird durch einen Aufruf der Datei CITOUCHEX.EXE von der MS-DOS Kommandozeile oder aus der AUTOEXEC.BAT resident in den Speicher des Computers geladen.

Bevor CiTouchD gestartet wird, muß der IRT an eine serielle Schnittstelle des Rechners angeschlossen werden. Dabei ist zu beachten, daß für den Betrieb des CiTouchD-Treibers unbedingt eine interruptfähige serielle Schnittstelle benötigt wird.

Wird eine Citron-LDVGA Grafikkarte zusammen mit der LDRI-Empfängerkarte eingesetzt, ist ein geeigneter IRT-Anschluß bereits über das Verbindungskabel zur Display-Einheit erfolgt. In diesem Fall muß nur noch die Basisadresse und der Interruptkanal für die SIO1 der LDVGA-Karte eingestellt werden. Dies erfolgt über das im LDVGA-Handbuch beschriebene Utility LDVINST.

4 Konfiguration

Die Parameter zur Konfiguration des CiTouchD Verhaltens können auf drei verschiedene Arten gesetzt werden:

1. Übergabe als Kommandozeilenparameter
2. Einträge in der Datei CITOUCHEX.INI
3. Aufrufe der Programmierschnittstelle (Int 33h bzw. Int 2Fh API)

Dabei verwendet CiTouchD die Parameter in der folgenden Reihenfolge: Zunächst liest der Treiber seine Parameter aus der Datei CITOUCHEX.INI. Fehlt dort ein Eintrag, so wird der Vorgabewert verwendet. Wurden beim Programmstart zusätzlich Kommandozeilenparameter angegeben, so überschreiben diese die Werte aus der CITOUCHEX.INI Datei. Dabei wird auch der Inhalt der Datei auf die neuen Werte aktualisiert. Welche Einträge in die Datei CITOUCHEX.INI möglich sind, ist im Kapitel "CITOUCHEX.INI" auf Seite 42 beschrieben.

Aufrufe der Programmierschnittstelle verändern lediglich die internen Variablen des Treibers. Der Inhalt der CITOUCD.INI Datei wird dadurch nicht geändert.

4.1 Kommandozeilenparameter

CiTouchD versteht die folgenden Kommandozeilenparameter in beliebiger Reihenfolge, unter Berücksichtigung von Groß-/Kleinschreibung:

Parameter	Bereich	CITOUCD.INI	Vorgabe	Beschreibung
/? [/h] [/H]	-	-	-	Gibt alle zulässigen Kommandozeilenparameter aus
?	-	-	-	Installationsstatus von CiTouchD
?D	-	-	-	Aktuelle Einstellungen des resident installierten Teils von CiTouchD
?T	-	-	-	Hard- und Softwarestand des angeschlossenen IRT
/b:	1..3	[Settings] Button =	1	Gibt an, welche Maustaste emuliert werden soll: 1 = linke Maustaste 2 = rechte Maustaste 3 = beide Maustasten gleichzeitig
/c:	1..4	[Hardware] Interrupt = IO_Base	1	Nummer der vom IRT verwendeten seriellen Schnittstelle. Die Option kombiniert /i: und /p:, wobei von den PC Standard-Schnittstellen ausgegangen wird: COM1 (/c:1) -> Port=3F8h, IRQ 4 COM2 (/c:2) -> Port=2F8h, IRQ 3 COM3 (/c:3) -> Port=3E8h, IRQ 4 COM4 (/c:4) -> Port=2E8h, IRQ 3
/d	-	-	-	Entfernt den residenten Teil des CiTouchD-Treibers wieder aus dem Speicher
/k:	0..64	[Citouchd] Stack =	6	Größe des von CiTouchD eingerichteten Stacks in Schritten von 256 Byte. Im Falle von /k:0 verwendet CiTouchD den Stack des gerade laufenden Programms.
/i:	0..15	[Hardware] Interrupt =	4	Nummer des von der seriellen Schnittstelle verwendeten IRQ
/m:	0..7	[Commands] mögliche Einträge sh. nächstes Kapitel	3D-Touch: 7 2D-Touch: 5	Maustasten-Emulationsmodus, je nach Modus sind weitere Parameter erforderlich. Diese werden durch Kommas, ohne Leerzeichen, abgetrennt
/p:	1..FFFF	[Hardware] IO_Base =	3F8	Basisadresse der seriellen Schnittstelle. Der Parameter wird als Hexadezimalzahl interpretiert. Es sind keine führenden Nullen erforderlich.
/r:	0..2	[Citouchd] HardReset =	0	Einstellung, ob Touch durch die Maus-API-Funktion 00h und 2Fh zurückgesetzt wird (Rücksetzen dauert bis zu 10 Sekunden). 0 = keine Zurücksetzung 1 = Rücksetzen nur in Fkt. 2Fh 2 = Rücksetzen nur in Fkt. 00h

/t:	0,1,2,128	[Hardware] IRT_Mode =	0	Kommunikationsprotokoll des angeschlossenen IRT: 0 = automatische Erkennung 1 = Mode-C (2D-Touch) 2 = CTS1 (3D-Touch) Wird über diesen Parameter das richtige Kommunikationsprotokoll angegeben erfolgt das Ankoppeln des IRT schneller als bei der automatischen Erkennung.
-----	-----------	--------------------------	---	--

Die Spalten in obiger Tabelle haben dabei die folgende Bedeutung:

Parameter	Syntax des Kommandozeilenparameters
Bereich	Erlaubter Wertebereich
CITOUCHD.INI	[Abschnitt] und zugehöriger Eintrag in der Datei CITOUCHD.INI
Vorgabe	Wert, der verwendet wird, wenn sowohl der Kommandozeilenparameter als auch der Eintrag in die CITOUCHD.INI Datei fehlt
Beschreibung	Kurze Erläuterung der Funktion des Kommandozeilenparameters

4.2 Vordefinierte Maustasten-Emulationsmodi

Es existieren 8 vordefinierte Emulationsmodi. Einer davon wird über den Aufrufparameter /m: beim Start von CiTouchD ausgewählt. Wird kein Maustasten-Emulationsmodus beim Aufruf von CiTouchD angegeben, wird der im Abschnitt [Commands] der CITOUCHD.INI-Datei festgelegte Emulationsmodus verwendet.

Im folgenden werden die vordefinierten Modi einzeln erläutert.

/m:0

Enter

Beschreibung Sobald Koordinaten gemeldet werden, wird die emulierte Maustaste gedrückt. Die Taste bleibt so lange gedrückt, bis der Touchbereich wieder verlassen wird.

Parameter keine

Beispiel `c:\>citouchd.exe /m:0`

Vorzüge

- Genaues Timing des Maustastendrucks
- Einfache Bedienung
- Bewegung des Cursors bei gedrückter Maustaste möglich

Nachteile

- Ungenaue Positionierung des Maustastendrucks bei Verwendung absoluter Koordinaten
- Relative Koordinaten nicht sinnvoll
- Geringe Sicherheit gegen Fehlbedienung

CITOUCHD.INI

```
[Commands]
Time1           = 0
Time2           = 0
Time3           = 0
Idle_T1         = 2
T1_Trigger      = 1
Trigger_T2      = 6
T2_UPT3         = 0
UPT3_Idle       = 0
UPT3_Trigger    = 0
```

/m:1

Exit

Beschreibung Sobald der Touchbereich wieder verlassen wird erfolgt ein kurzer Maustastendruck.

Parameter keine

Beispiel `c:\>citouchd.exe /m:1`

Vorzüge

- Genaues Timing des Maustastendrucks
- Einfache Bedienung
- Gute Positionierung des Maustastendrucks

Nachteile

- Relative Koordinaten nicht sinnvoll
- Keine Bewegung des Cursors bei gedrückter Maustaste möglich

CITOUCHD.INI

[Commands]

Time1	= 0
Time2	= 0
Time3	= 0
Idle_T1	= 1
T1_Trigger	= 6
Trigger_T2	= 1
T2_UPT3	= 0
UPT3_Idle	= 0
UPT3_Trigger	= 0

/m:2,t

Tap

Beschreibung

Wird der Touchbereich verlassen und innerhalb einer einstellbaren Zeit wieder unterbrochen, so wird ein Tastendruck emuliert. Dieser Vorgang wird als "Tap" bezeichnet. Die Taste bleibt solange gedrückt bis Touchbereich erneut verlassen wird.

Parameter

Tap-Time (t): Zeit innerhalb derer der IRT wieder unterbrochen werden muß um einen Tap zu erzeugen. Die Tap-Time läßt sich in Schritten von 55 ms zwischen 0 ms und 2 s einstellen.

Beispiel

```
c:\>citouchd.exe /m:2,300
```

Vorzüge

- Gute Sicherheit gegen Fehlbedienung
- Bewegung des Cursors bei gedrückter Maustaste möglich

Nachteile

- Ungenaue Positionierung des Mausdrucks bei Verwendung absoluter Koordinaten
- Relativ komplizierte Bedienung

CITOUCHD.INI

```
[Commands]
Time1           = 0
Time2           = 0
Time3           = 0
Idle_T1         = 2
T1_Trigger      = 5
Trigger_T2      = 6
T2_UPT3         = 0
UPT3_Idle       = 0
UPT3_Trigger    = 0
[Settings]
TapTime         = t
```

/m:3,t1,t2

Time

Beschreibung

Ist der Touchbereich unterbrochen und erfolgt für eine Zeit T1 keine Mausbewegung, wird ein Maustastendruck emuliert. Die Taste bleibt solange gedrückt bis der Touchbereich wieder verlassen wird. Wird der Touchbereich innerhalb einer Zeit T2 erneut unterbrochen, so erfolgt sofort ein Maustastendruck. Auf diese Weise ist es möglich einen Doppelklick zu erzeugen.

Parameter

Time to Click (t1): Nach dieser Zeit erfolgt der erste Maustastendruck. T1 ist in Schritten zu 55 ms zwischen 0 ms und 2 s einstellbar.

Time to Idle (t2): Wird der Touchbereich innerhalb dieser Zeit erneut unterbrochen, so erfolgt ein sofortiger Maustastendruck. T2 ist in Schritten zu 55 ms zwischen 0 ms und 2 s einstellbar.

Beispiel

```
c:\>citouchd.exe /m:3,500,300
```

Vorzüge

- Einfache Bedienung
- Bewegungen des Cursors bei gedrückter Maustaste möglich

Nachteile

- Ungenaues Timing des Maustastendrucks
- Geringe Sicherheit gegen Fehlbedienung

CITOUCHD.INI

```
[Commands]
Time1           = t1
Time2           = 0
Time3           = t2
Idle_T1         = 2
T1_Trigger      = 2
Trigger_T2      = 6
T2_UPT3         = 0
UPT3_Idle       = 0
UPT3_Trigger    = 2
```


/m:4,n**Dual Touch**Beschreibung

Ist der Touchbereich unterbrochen und erfolgt gleichzeitig eine zweite Berührung, so wird ein Maustastendruck emuliert. Die Taste bleibt solange gedrückt wie die Zweifachberührung besteht.

Parameter

Dual Touch Skip Count (n): Legt fest wieviel Zweifachberührungsmeldungen verworfen werden bevor der Maustastendruck emuliert wird, einstellbar zwischen 0 und 255.

Beispiel

```
c:\>citouchd.exe /m:4,1
```

Vorzüge

- Einfache Bedienung
- Genaue örtliche und zeitliche Positionierung des Maustastendrucks

Nachteile

- Geringe Sicherheit gegen Fehlbedienung
- Keine Bewegung des Cursors bei gedrückter Maustaste möglich

CITOUCHD.INI

```
[Commands]
Time1           = 0
Time2           = 0
Time3           = 0
Idle_T1         = 2
T1_Trigger      = 4
Trigger_T2      = 8
T2_UPT3         = 0
UPT3_Idle       = 0
UPT3_Trigger    = 0
[Settings]
DbErrSkip       = n
```

m:5,n**Dual / Exit**Beschreibung

Ist der Touchbereich unterbrochen und erfolgt gleichzeitig eine zweite Berührung, so wird ein Maustastendruck emuliert. Im Gegensatz zu "Dual Touch" bleibt die Taste solange gedrückt bis der Touchbereich wieder verlassen wird.

Parameter

Dual Touch Skip Count (n): Legt fest wieviel Zweifachberührungsmeldungen verworfen werden bevor der Maustastendruck emuliert wird, einstellbar zwischen 0 und 255.

Beispiel

```
c:\>citouchd.exe /m:5,1
```

Vorzüge

- Einfache Bedienung
- Genaue örtliche und zeitliche Positionierung des Maustastendrucks
- Bewegung des Cursors bei gedrückter Maustaste möglich

Nachteile

- Geringe Sicherheit gegen Fehlbedienung
- Schwieriges Erzeugen von Doppelklicks

CITOUCHD.INI

```
[Commands]
Time1           = 0
Time2           = 0
Time3           = 0
Idle_T1         = 2
T1_Trigger      = 4
Trigger_T2      = 6
T2_UPT3         = 0
UPT3_Idle       = 0
UPT3_Trigger    = 0
[Settings]
DbErrSkip       = n
```

m:6,t1,t2**Time / Time**Beschreibung

Ist der Touchbereich unterbrochen und erfolgt für eine Zeit T1 keine Mausbewegung, so wird ein Maustastendruck emuliert. Wird der Cursor weiterhin nicht bewegt, so wird nach einer Zeit T2 die Maustaste kurz angehoben und sofort wieder gedrückt.

Parameter

Time to Click (t1): Nach dieser Zeit erfolgt der Maustastendruck. T1 ist in Schritten zu 55 ms zwischen 0 ms und 2 s einstellbar.

Time to Second Click (t2): Nach dieser Zeit erfolgt der zweite und weitere Tastendrücke, bis der Touchbereich wieder verlassen wird. T2 ist in Schritten zu 55 ms zwischen 0 ms und 2 s einstellbar.

Beispiel

```
c:\>citouchd.exe /m:6,500,300
```

Vorzüge

- Einfache Bedienung
- Einfaches und positionsgenaues Erzeugen von Doppelklicks

Nachteile

- Ungenaueres Timing des Maustastendrucks
- Geringe Sicherheit gegen Fehlbedienung

CITOUCHD.INI

[Commands]

```
Time1           = t1
Time2           = t2
Time3           = 0
Idle_T1         = 2
T1_Trigger      = 2
Trigger_T2      = 1
T2_UPT3         = 6
UPT3_Idle       = 6
UPT3_Trigger    = 2
```

m:7,p**Z-Press**Beschreibung

Dieser Wahlpunkt steht nur zur Verfügung, falls der IRT mit Drucksensoren ausgerüstet ist. Ein Maustastendruck wird emuliert, sobald ein einstellbarer Druck auf die Frontscheibe überschritten wird. Die Taste bleibt solange gedrückt, bis dieser Druck, abzüglich einer Hysterese, wieder unterschritten wird.

Parameter

Pressure Sensitivity (p): Höhe des notwendigen Drucks auf die Frontscheibe, einstellbar zwischen 0 und 255. Die tatsächliche Druckstärke ist abhängig vom Einbau des IRT.

Beispiel

```
c:\>citouchd.exe /m:7,10
```

Vorzüge

- Einfache Bedienung
- Bewegung des Cursors bei gedrückter Maustaste möglich
- Hohe Sicherheit gegen Fehlbedienung

Nachteile

- IRT muß mit Drucksensoren ausgerüstet sein

CITOUCHD.INI

[Commands]

```
Time1           = 0
Time2           = 0
Time3           = 0
Idle_T1         = 3
T1_Trigger      = 1
Trigger_T2      = 7
T2_UPT3         = 0
UPT3_Idle       = 0
UPT3_Trigger    = 0
```

[Settings]

```
Pressure        = p
```

5 Softwareschnittstelle

Es folgt eine Einführung in die Funktion des CiTouchD-Treibers und eine Beschreibung der Anwenderprogrammen zur Verfügung gestellten Softwareschnittstellen.

5.1 Grundlagen

Um die API-Funktionen des CiTouchD-Treiber einsetzen zu können ist ein Verständnis der Funktion der Mausemulation erforderlich. Dies soll in den folgenden Kapiteln vermittelt werden.

5.1.1 Koordinatensysteme

Positionen werden innerhalb des Touchtreibers immer auf einen virtuellen Grafikbildschirm bezogen, dessen Auflösung von dem jeweiligen Videomodus und der ihm zugrundeliegenden Grafik-Karte abhängt. Da dieser virtuelle Grafikbildschirm auch innerhalb der verschiedenen Textmodi zur Bestimmung der Touchunterbrechung herangezogen wird und die Grundlage für die Kommunikation mit dem Touch-Interface darstellt, muß jeweils eine Umrechnung zwischen diesen Grafikkordinaten und der Spalten- bzw. Zeilenkoordinate des Cursors vorgenommen werden. Da jede Spalte und Zeile acht Punkten entspricht, müssen die Grafikkordinaten durch acht geteilt werden.

Der Ursprung des virtuellen Bildschirms liegt in der linken oberen Ecke. Die Koordinaten werden nach unten und nach rechts positiv gezählt, sind also immer positiv.

Die Funktionen 0Fh und 0Bh der Int33-API liefern relative Koordinaten. Aus Kompatibilitätsgründen zur Maus wurde hier die Einheit „Mickey“ (= 1/400 Zoll) beibehalten. Funktion 0Fh erlaubt die Einstellung des Verhältnisses zwischen Mickeys und 8 virtuellen Pixeln.

Relative Koordinaten sind vorzeichenbehaftet. Negative Vektoren stehen für eine Bewegung nach links oben, positive Vektoren für eine Bewegung nach rechts unten.

5.1.2 Unterstützte Videomodi

Unterstützt werden alle über Funktion 00h des EGA/VGA-BIOS einstellbare Video-Modi:

Bildschirmmodus	Text/Grafik	Auflösung	Virtueller Bildschirm
00h	Text	40*25 Zeichen	320*200 Punkte
01h	Text	40*25 Zeichen	320*200 Punkte
02h	Text	80*25 Zeichen	640*200 Punkte
03h	Text	80*25 Zeichen	640*200 Punkte
04h	Grafik	320*200 Punkte	320*200 Punkte
05h	Grafik	320*200 Punkte	320*200 Punkte
06h	Grafik	640*200 Punkte	640*200 Punkte
07h	Text	80*25 Zeichen	640*200 Punkte
0Dh	Grafik	320*200 Punkte	320*200 Punkte
0Eh	Grafik	640*200 Punkte	640*200 Punkte
0Fh	Grafik	640*350 Punkte	640*350 Punkte
10h	Grafik	640*350 Punkte	640*350 Punkte
11h	Grafik	640*480 Punkte	640*480 Punkte
12h	Grafik	640*480 Punkte	640*480 Punkte
13h	Grafik	320*200 Punkte	320*200 Punkte

5.1.3 Benutzerdefinierte Tastenemulation

Im CiTouchD-Treiber werden Maustastenergebnisse von der sogenannten Button-Machine erzeugt. Dabei handelt es sich um einen programmierbaren, asynchronen Zustandsautomaten. Der Übergang von einem Zustand zum nächsten erfolgt sobald alle erforderlichen Bedingungen eintreffen. Das genaue Zustands-Übergangs-Diagramm der Button-Machine ist in Abbildung 5-1 dargestellt:

Nach der Initialisierung des Treibers befindet sich die Button-Machine im Zustand "IDLE". Alle Maustasten sind im losgelassenen Zustand. Die Zustände werden in Pfeilrichtung durchlaufen.

In den Zuständen "T1", "T2" und "T3" kann eine Zeit eingestellt werden nach deren Ablauf der jeweilige Zustand verlassen wird. Für den Zustand "T1" muß dazu gleichzeitig die int "T1_Trigger" angegebene Bedingung erfüllt sein. Dagegen ist in den Zuständen "T2" und "T3" entweder die angegebene Bedingung oder ein Ablauf der eingestellten Zeit ausreichend um in den Folgezustand zu wechseln.

Beim Eintreten in den Zustand "TRIGGER" wird ein Maustastendruck emuliert. Beim Übergang in den Zustand "T3" wird die Maustaste wieder losgelassen.

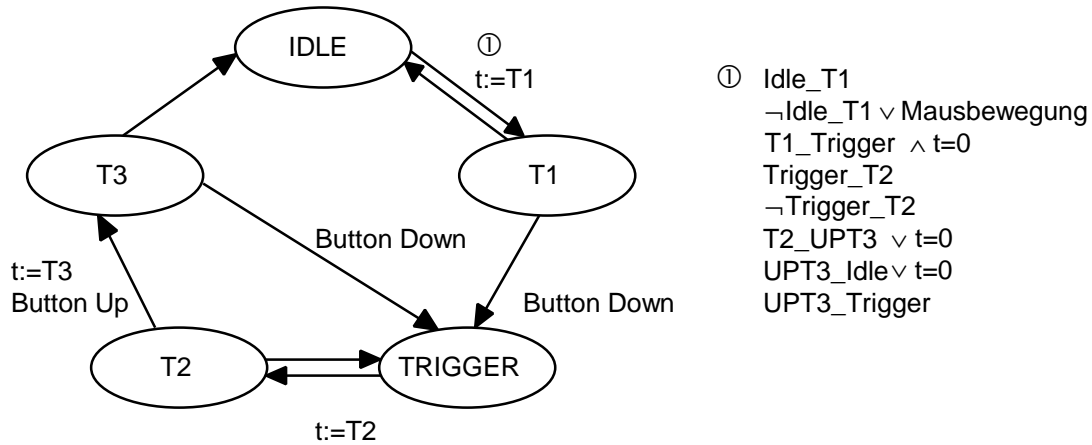


Abbildung 5-1, Zustandsautomat zur Maustastenemulation

Mögliche Bedingungen für Zustandsübergänge lauten:

- **Never:** Diese Bedingung tritt niemals ein
- **Immediately:** Diese Bedingung tritt immer und sofort ein
- **Enter:** Es werden Koordinaten gemeldet
- **Z-Press:** Der eingestellte Druck wurde überschritten
- **Dual Touch:** Es wurde eine Zweifachberührung festgestellt
- **Tap:** Ein "Tap" wurde erkannt
- **Leave:** Es werden keine Koordinaten mehr gemeldet
- **Z-Release:** Der eingestellte Druck wurde unterschritten
- **No Dual Touch:** Es werden keine Zweifachberührungen mehr gemeldet

Der Tastenemulationsmodus "**Time**" kann zum Beispiel folgendermaßen programmiert werden:

```
T1 = 550 ms
T2 = 0 ms
T3 = 550 ms
Idle_T1    = "Enter"
T1_Trigger = "Enter"
Trigger_T2 = "Leave"
T2_UPT3    = "Never"
UPT3_Idle  = "Never"
UPT3_Trigger = "Enter"
```

Ein weiteres Beispiel wäre ein neuer Emulationsmodus mit Namen "Z-Press / Exit". Dabei erfolgt, ähnlich dem Modus "Dual / Exit", das Loslassen der Maustaste nicht bereits beim Unterschreiten eines Drucks sondern erst nachdem der Touchbereich vollständig verlassen wird. Die Parameter hierfür lauten:

```
T1 = 0 ms
T2 = 0 ms
T3 = 0 ms
Idle_T1      = "Enter"
T1_Trigger   = "Z-Press"
Trigger_T2   = "Leave"
T2_UPT3      = "Never"
UPT3_Idle    = "Never"
UPT3_Trigger = "Never"
```

5.1.4 Weitere Betriebsparameter

Damit die Mausemulation durch einen IRT so komfortabel wie möglich zu bedienen ist, existieren eine Reihe weiterer Parameter mit denen das Verhalten der Mausemulation auf die Bedürfnisse des Benutzers und der Applikation angepaßt werden können. Diese Parameter werden entweder in der CITOUCHE.D.INI Datei dauerhaft eingestellt, oder von der jeweiligen Applikation durch Aufruf der erweiterten API-Funktionen temporär verändert.

5.1.4.1 Glättung

Die Auflösung des IRT im Mode-C Kommunikationsprotokoll ist bei Verwendung absoluter Koordinaten nicht ausreichend um im Grafikmodus jeden Pixel des Bildschirms anzusteuern, vielmehr entspricht eine IRT-Koordinatenänderung bei einer Bildschirmauflösung von 640 mal 480 Bildpunkten in etwa 8 Pixel. Befindet sich der Berührungspunkt nun gerade am Übergang von einer Koordinate auf die nächste, springt der Cursor ständig einige Pixel hin und her. Um diesen störenden Effekt zu vermindern kann hier ein Glättungsfaktor eingestellt werden. Dabei wird der Mittelwert über die eingestellte Zahl an IRT-Koordinatenmeldungen gebildet. Die Anzahl der bei der Mittelwertbildung zu berücksichtigenden Koordinatenmeldungen kann für die X- und die Y-Achse getrennt eingestellt werden.

Die API-Funktionen und zugehörigen Strukturelemente für die Glättungsparameter lauten im Einzelnen:

citGetDriverSettings (Funktion 0Bh)	Abfrage der Einstellungen
citSetDriverSettings (Funktion 14h)	Verändern der Einstellungen

Die zugehörigen Elemente der Struktur DRIVERSETTINGS (sh. S. 40) lauten:

<i>dsSmoothX</i>	Glättungsfaktor für die X-Achse
<i>dsSmoothY</i>	Glättungsfaktor für die Y-Achse
<i>dsSmoothAlways</i>	0 = mit der Mittelwertbildung wird nach jeder neuen Unterbrechung des Touchbereichs frisch begonnen. Das bedeutet der Cursor wird zunächst unmittelbar auf den Berührungspunkt gesetzt und erst weitere Cursorbewegungen werden gemittelt. 1 = der alte Mittelwert bleibt nach dem Verlassen des Touchbereichs erhalten. Wird der Touchbereich wieder unterbrochen, wandert der Cursor, ausgehend von seiner alten Position, entsprechend dem eingestellten Glättungsfaktor schrittweise zum neuen Berührungspunkt hin.
<i>dsCoordSkip</i>	Legt fest, wieviele Koordinatenmeldungen des IRT nach dem Unterbrechen des Touchbereichs verworfen werden, bevor eine neue Cursorposition gemeldet wird. Das Verwerfen der ersten Koordinatenmeldungen ist zum Beispiel dann sinnvoll, wenn der IRT in einem größeren Abstand zur Bildschirmoberfläche angebracht ist. Der IRT erkennt in diesem Fall eine gültige Unterbrechung bevor der Finger die Bildschirmoberfläche berührt. In der Regel verschiebt sich die Position des Fingers aber auf dem Weg zwischen Unterbrechungserkennung durch den IRT und der Bildschirmoberfläche. Der Benutzer erwartet jedoch erst dann eine Veränderung der Cursorposition wenn er die Bildschirmoberfläche tatsächlich berührt. Durch das Verwerfen der ersten Koordinatenmeldungen kann das erwartete Verhalten des Cursors erzielt werden.

Der zulässige Wertebereich für *dsSmoothX* und *dsSmoothY* bewegt sich von 0 bis *dsSmoothMax*. Dieser Wert kann über **citGetDriverConstants** (Funktion **0Eh**) abgefragt werden.

In der Datei CITOUCHE.D.INI steuern die folgenden Einträge das Glättungsverhalten:

[Settings]

X_Smoothing	entspricht <i>dsSmoothX</i>
Y_Smoothing	entspricht <i>dsSmoothY</i>
SmoothAlways	entspricht <i>dsSmoothAlways</i>
CoordinateSkip	entspricht <i>dsCoordSkip</i>

5.1.4.2 Koordinatenmodus

Der CiTouchD-Treiber kann sowohl mit absoluten als auch mit relativen Koordinaten arbeiten. Werden absolute Koordinaten verwendet, springt der Cursor direkt an die Stelle der Unterbrechung des Touchbereichs. Bei Verwendung relativer Koordinaten bewegt sich der Cursor dagegen immer relativ zu seiner aktuellen Position weiter. Die Richtung der Bewegung entspricht dabei der Richtung des sich bewegenden Unterbrechungspunktes. Da die zurückgelegte Strecke geringer ausfallen kann als es der Fingerbewegung entspricht, läßt sich eine Ortsauflösung erreichen die der einer konventionellen Maus entspricht. Die Geschwindigkeit mit der sich der Cursor bei Verwendung relativer Koordinaten bewegt wird über die folgenden Funktionen eingestellt:

citGetCalibrationRel (Funktion 0Ah)	Abfrage der Kalibrierung für rel. Koordinaten
citSetCalibrationRel (Funktion 13h)	Ändern der Kalibrierung relativer Koordinaten

Die Parameter werden in der Struktur CALIBRATIONREL (sh. S. 38) übergeben.

In welchem Koordinatenmodus CiTouchD arbeiten soll wird über die folgende Funktion festgelegt:

citGetDriverSettings (Funktion 0Bh)	Abfrage des Koordinatenmodus
citSetDriverSettings (Funktion 14h)	Ändern des Koordinatenmodus

Die zugehörigen Elemente der Struktur DRIVERSETTINGS (sh. S. 40) lauten:

<i>dsAbsolute</i>	0 = Es wird der relative Koordinatenmodus verwendet
	1 = Es wird der absolute Koordinatenmodus verwendet

Es ist weiterhin möglich im laufenden Betrieb zwischen den verschiedenen Koordinatenmodi zu wechseln. Unter welcher Bedingung ein Wechsel zwischen absoluten und relativen Koordinaten erfolgen soll wird über die folgende Funktion festgelegt:

citGetCommands (Funktion 07h)	Abfrage der Bedingung für Moduswechsel
citSetCommands (Funktion 10h)	Ändern der Bedingung für Moduswechsel

Die zugehörigen Elemente der Struktur COMMANDS (sh. S. 39) lauten:

<i>cmModeChange</i>	0 = Während des Betriebs ist kein Wechsel zwischen absoluten und relativen Koordinaten möglich.
	3 = Diese Option ist nur verfügbar, wenn der IRT über Drucksensoren verfügt. Immer wenn der eingestellte Druck auf die Frontscheibe überschritten wird findet ein Wechsel zwischen absoluten und relativen Koordinaten statt.
	4 = Bei jeder Zweifachberührung wird zwischen absoluten und relativen Koordinaten gewechselt.
	5 = Bei jedem "Tap" wird zwischen absoluten und relativen Koordinaten gewechselt (die Erläuterung eines "Tap" befindet sich auf Seite 8).

Um bei kritischen Anwendungen die Gefahr einer Fehlbedienung zu verringern kann festgelegt werden, wann sich die Cursorposition ändern darf. Dabei existieren die folgenden Möglichkeiten:

- **Enter:** *cmCoordEnterZ* = 0
cmCoordSignalZ = 0
Es wird eine neue Cursorposition gemeldet sobald eine gültige Unterbrechung des Touchbereichs vorliegt.
- **Z-Press/Enter:** *cmCoordEnterZ* = 1
cmCoordSignalZ = 0
Diese Option ist nur verfügbar, wenn der IRT über Drucksensoren verfügt. Damit nach dem Eintauchen eine neue Cursorposition gemeldet wird, muß

zuerst der eingestellte Druck auf die Frontscheibe überschritten werden. Für weitere Änderungen der Cursorposition genügt es dann, daß der Touchbereich unterbrochen bleibt.

- **Z-Press:** $cmCoordEnterZ = 1$
 $cmCoordSignalZ = 1$
 Diese Option ist nur verfügbar, wenn der IRT über Drucksensoren verfügt. Neue Cursorpositionen werden nur gemeldet, solange der eingestellte Druck auf die Frontscheibe überschritten wird.

Das Verhalten wird mit den folgenden Funktionen eingestellt

citGetCoordMode (Funktion 08h)	Abfrage der Parameter
citSetCoordMode (Funktion 11h)	Ändern der Parameter

Die Parameter werden in der Struktur COORDMODE (sh. S. 40) übergeben.

In der Datei CITOUCHE.INI steuern die folgenden Einträge den Koordinatenmodus:

[Calibration]

XRel_Div	Divisor für X-Koordinatenskalierung (entspricht $crDivX$)
XRel_Mul	Multiplikator für X-Koordinatenskalierung (entspricht $crMulX$)
YRel_Div	Divisor für Y-Koordinatenskalierung (entspricht $crDivY$)
YRel_Mul	Multiplikator für Y-Koordinatenskalierung (entspricht $crMulY$)
SmoothAlways	entspricht $dsSmoothAlways$
CoordinateSkip	entspricht $dsCoordSkip$

[Commands]

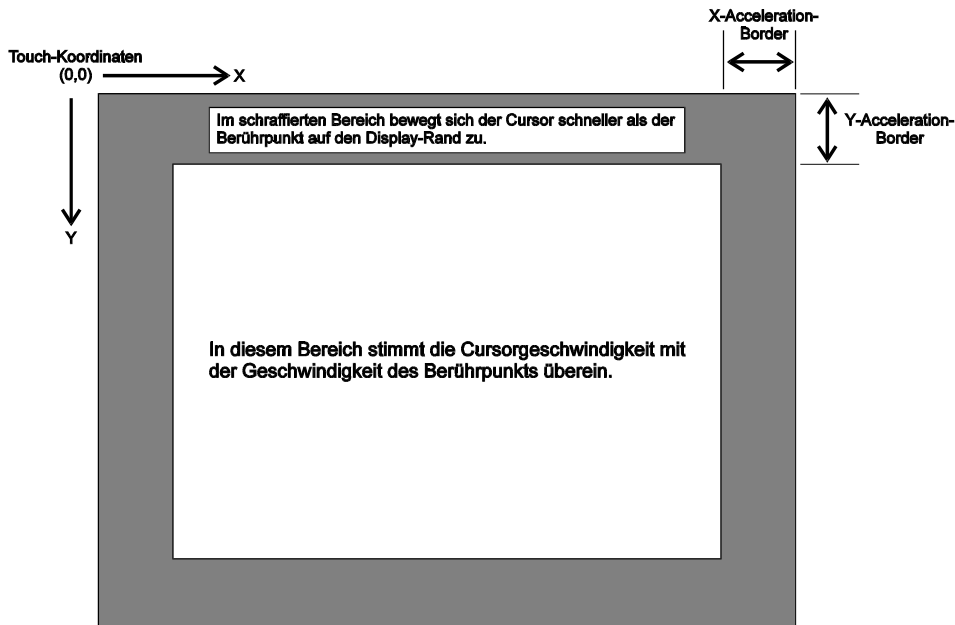
ModeChange	entspricht $cmModeChange$
-------------------	---------------------------

[Settings]

AbsoluteMouse	entspricht $dsAbsolute$
CoordEnterZ	entspricht $cmCoordEnterZ$
CoordSignalZ	entspricht $cmCoordSignalZ$

5.1.4.3 Cursor Beeinflussung

Mit diesen Parametern kann ein Abstand zwischen der tatsächlichen Cursorposition gegenüber dem Berührungspunkt und eine beschleunigte Bewegung des Cursors gegenüber dem Finger eingestellt werden. Je nach Einbaulage des IRT ist es nicht immer möglich den äußersten Bildschirmrand mit dem Finger zu erreichen. Durch die Cursorbeschleunigung des CiTouchD-Treibers bewegt sich die Cursorposition außerhalb eines Bereichs einstellbarer Größe schneller als der Finger auf den Bildschirmrand zu. Somit kann in jedem Fall der äußerste Bildschirmrand erreicht werden. Die Cursorbeschleunigung wird nur zusammen mit absoluten Koordinaten verwendet.



Zum Einstellen der Beschleunigungsparameter werden die folgenden API-Funktionen verwendet:

citGetAcceleration (Funktion 20h)	Abfrage der Beschleunigungsparameter
citSetAcceleration (Funktion 21h)	Ändern der Beschleunigungsparameter

Die Parameter werden in der Struktur ACCELERATION (sh. S. 37) übergeben.

Zum Einstellen des Abstands zwischen Berührungspunkt und Cursorposition werden die folgenden API-Funktionen verwendet:

citGetDriverSettings (Funktion 0Bh)	Abfrage des Abstands
citSetDriverSettings (Funktion 14h)	Ändern des Abstands

Die zugehörigen Elemente der Struktur DRIVERSETTINGS (sh. S. 40) lauten:

<i>dsOfsX</i>	X-Abstand zwischen Berührungspunkt und Cursorposition
<i>dsOfsY</i>	Y-Abstand zwischen Berührungspunkt und Cursorposition

In der Datei CITOUCHD.INI steuern die folgenden Einträge die Cursorbeeinflussung:

[Acceleration]

X_Border	X-Grenze, außerhalb derer die Beschleunigung greift (entspricht <i>acBorderX</i>)
Y_Border	Y-Grenze, außerhalb derer die Beschleunigung greift (entspricht <i>acBorderY</i>)
X_Mul	X-Beschleunigungsfaktor (entspricht <i>acMulX</i>)
Y_Mul	Y-Beschleunigungsfaktor (entspricht <i>acMulY</i>)

[Settings]

X_Offset	entspricht <i>dsOfsX</i>
Y_Offset	entspricht <i>dsOfsY</i>

5.1.4.4 Steuerung der Hintergrundbeleuchtung

Der IRT besitzt eine "Touch-Saver" Funktion die immer dann aktiviert wird, wenn der Touchbereich für eine bestimmte, einstellbare Zeit nicht unterbrochen wurde. Ist der Touch-Saver aktiv, so wird gleichzeitig die Abtastrate der IRT-Lichtschranken verringert. Daher reagiert der IRT bei aktivem Touch-Saver langsamer als bei inaktivem Touch-Saver auf Unterbrechungen des Touchbereichs.

Des Weiteren besitzt der IRT einen PWM-Ausgang, der z.B. zur Helligkeitseinstellung der Hintergrundbeleuchtung von TFT-Displays eingesetzt werden kann.

Der CiTouchD-Treiber ordnet nun den zwei Zuständen "aktiv" und "inaktiv" des Touch-Savers je ein bestimmtes Puls-/Pausenverhältnis des PWM-Ausgangs, und somit unterschiedliche Helligkeiten der Hintergrundbeleuchtung, zu. Die Aktivierungszeit des Touch-Savers lässt sich über die folgenden API-Funktionen einstellen:

citGetTouchSettings (Funktion 0Ch)	Abfrage der Aktivierungszeit
citSetTouchSettings (Funktion 15h)	Ändern der Aktivierungszeit

Die zugehörigen Elemente der Struktur TOUCHSETTINGS (sh. S. 42) lauten:

<i>tsTSaver</i>	Aktivierungszeit des Touch-Savers (0=sofort aktiv, 65535 = abgeschaltet)
<i>tsTScan</i>	Abstand zwischen zwei Scanvorgängen bei aktivem Touch-Saver

Die Helligkeit der Hintergrundbeleuchtung läßt sich über die folgenden API-Funktionen einstellen:

citGetDimming (Funktion 1Dh)	Abfrage der Helligkeitswerte
citSetDimming (Funktion 1Eh)	Ändern der Helligkeitswerte

Die Parameter werden in der Struktur DIMMING (sh. S. 40) übergeben.

In der Datei CITOUCHD.INI steuern die folgenden Einträge die Touch-Saver Funktion:

[Settings]

DimmingHigh	Helligkeit bei inaktivem Touchsaver (Normalbetrieb, entspricht <i>tsDimmHigh</i>)
DimmingLow	Helligkeit bei aktivem Touchsaver (entspricht <i>tsDimmLow</i>)
SaverScan	entspricht <i>tsTScan</i>
SaverTime	entspricht <i>tsTSaver</i>

5.1.5 Aufruf der API-Funktionen

Die Kommunikation eines Anwenderprogramms mit dem CiTouchD-Treiber erfolgt über Softwareinterrupts. Dabei wird die Funktionsnummer im AL Register übergeben und der entsprechende Softwareinterrupt ausgelöst. Weitere Parameter werden in den anderen Prozessorregistern übergeben. Die genaue Registerbelegung ist bei der jeweiligen Funktionsbeschreibung angegeben.

5.2 Emulierte Maustreiberfunktionen der Int33-API

Die Int 33-API (Application Program Interface, Interrupt 33h) des CiTouchD-Treibers ist kompatibel zum Microsoft[®] Maustreiber Version 8.0. Daher wird auf eine detaillierte Beschreibung jeder Funktion dieser API verzichtet und nur auf Besonderheiten bei einzelnen Funktionen eingegangen.

5.2.1 Numerische Übersicht

Die nachfolgende Tabelle zeigt die unterstützten Funktionen der Int33-API in numerischer Reihenfolge. Die mit einem Kreuz (†) markierten Funktionen können zwar aufgerufen werden, besitzen jedoch keinerlei Funktionalität. Die mit einem Stern (*) markierten Funktionen unterscheiden sich in Details vom Microsoft[®] Maustreiber und werden weiter unten genauer beschrieben.

Nummer	Bezeichnung	Beschreibung
00h	int33ResetAndStatus	Reset des Treibers und der Touch-Hardware
01h	int33ShowCursor	Maus-Cursor auf dem Bildschirm anzeigen
02h	int33HideCursor	Maus-Cursor ausblenden
03h	int33GetPosBtn	Mausposition und Status der emulierten Maustaste ermitteln
04h	int33SetCursorPos	Maus-Cursor bewegen
05h	int33GetBtnPress	Anzahl der Betätigungen der emulierten Maustaste ermitteln
06h	int33GetBtnRelease	Wie oft wurde die emulierte Maustaste losgelassen?
07h	int33SetMinMaxHor	Horizontalen Bewegungsbereich für Maus-Cursor festlegen
08h	int33SetMinMaxVer	Vertikalen Bewegungsbereich für Maus-Cursor festlegen
09h	int33SetGraphicsCursor	Maus-Cursor im Grafikmodus definieren
0Ah	int33SetTextCursor	Maus-Cursor im Textmodus definieren
0Bh	int33ReadMotionCounters	Bewegungswerte ermitteln
0Ch	int33InstallEvent	Event-Handler installieren
0Dh	int33LightPenOn	Emulation des Lichtgriffels anschalten †
0Eh	int33LightPenOff	Emulation des Lichtgriffels abschalten †
0Fh	int33SetMickeyPixelRatio	Verhältnis zwischen Miceys und Pixeln setzen
10h	int33ConditionalOff	Ausschluß-Bereich definieren
13h	int33SetDoubleThreshold	Schwelle für Verdoppelung der Mausgeschwindigkeit setzen*
14h	int33SwapEvents	Austauschen der Event-Handler
15h	int33GetStateSize	Größe des Touch-Status-Puffers ermitteln
16h	int33SaveState	Touch-Status sichern
17h	int33RestoreState	Touch-Status restaurieren
18h	int33SetAltEvent	Alternativen Event-Handler installieren
19h	int33GetAltEvent	Adresse eines alternativen Event-Handlers ermitteln
1Ah	int33SetSensitivity	Maus-Sensitivität einstellen*

1Bh	int33GetSensitivity	Maus-Sensitivität ermitteln
1Ch	int33SetInterruptRate	Interruptrate der Maus-Hardware einstellen*
1Dh	int33SetCrtPage	Bildschirmseite für Maus-Cursor setzen
1Eh	int33GetCrtPage	Bildschirmseite des Maus-Cursor ermitteln
1Fh	int33Disable	Touch-Treiber deaktivieren
20h	int33Enable	Touch-Treiber aktivieren
21h	int33SoftReset	Softwarereset des CiTouchD-Treibers
22h	int33SetLanguage	Sprache für Meldungen einstellen*
23h	int33GetLanguage	Sprache für Meldungen abfragen
24h	int33Inquire	Version, Maustyp und IRQ ermitteln*
25h	int33InquireEx	Allgemeine Informationen abfragen
26h	int33GetMouseScreen	Ausdehnung des virtuellen Maus-Bildschirms abfragen
27h	int33GetCursorMask	Bit-Masken des Bildschirm-Cursors abfragen*
28h	int33SetVideoMode	Videomodus setzen †
29h	int33InquireVideoModes	Liste der verfügbaren Video-Modi abfragen*
2Ah	int33GetCursorInfo	Informationen über Maus-Cursor Abfragen
2Bh	int33SetAcceleration	Beschleunigungskurve einstellen*
2Ch	int33GetAcceleration	Aktuelle Beschleunigungskurve auslesen*
2Dh	int33GetSelectAccel	Aktuelle Beschleunigungskurve einstellen/abfragen*
2Fh	int33HardReset	Reset der Touch-Hardware
30h	int33InquireBallpoint	Einstellungen der Ballpoint-Maus setzen/abfragen*
31h	int33GetVirtualScreen	Ausdehnung des virtuellen Bildschirms abfragen
32h	int33InquireFunc	Unterstützte Funktionen ermitteln
33h	int33InquireAll	Einstellungen ermitteln
34h	int33GetIni	Lage der Datei CITOUCHE.D.INI ermitteln

5.2.2 Besonderheiten der Int 33 API-Emulation

Einige Funktionen der Int 33 API unterscheiden sich in Details vom Verhalten des Microsoft® Treibers. Diese Funktionen werden im folgenden einzeln erläutert.

Funktion 13h Schwelle für Verdoppelung der Mausgeschwindigkeit setzen*

Eingabe: AX = 0013h
 DX = Schwelle für Verdoppelung der Mausgeschwindigkeit

Rückgabe: keine

Besonderheiten: Die Schwelle für die Verdoppelung der Mausgeschwindigkeit in DX wird zwar intern gespeichert, aber ansonsten ignoriert.

Funktion 1Ah Maus-Sensitivität einstellen*

Eingabe: AX = 001Ah
 BX = Anzahl horizontaler Mickeys, die 8 Pixeln entsprechen
 CX = Anzahl vertikaler Mickeys, die 8 Pixeln entsprechen
 DX = Schwelle für Verdoppelung der Mausgeschwindigkeit

Rückgabe: keine

Besonderheiten: Die Schwelle für die Verdoppelung der Mausgeschwindigkeit in DX wird zwar intern gespeichert, aber ansonsten ignoriert.

Funktion 1Ch**Interruptrate der Maus-Hardware einstellen***Eingabe:

AX = 001Ch
 BX = Interrupt-Rate

Rückgabe:

keine

Besonderheiten:

Die übergebene Rate wird intern gespeichert, ansonsten ignoriert.

Funktion 22h**Sprache für Meldungen einstellen***Eingabe:

AX = 0022h
 BX = Codenummer der Sprache

Rückgabe:

keine

Besonderheiten:

Es wird nur die Codenummer 0 (=Englisch) unterstützt. Alle anderen Codenummern werden ignoriert.

Funktion 24h**Version, Maustyp und IRQ ermitteln***Eingabe:

AX = 0024h

Rückgabe:

BH = Vorkommteile der Versionsnummer
 BL = Nachkommteile der Versionsnummer
 CH = Maustyp (immer 2 = serielle Maus)
 CL = IRQ-Nummer

Besonderheiten:

CiTouchD unterstützt prinzipiell jede auf einem PC verfügbare IRQ-Nummer, so daß in CL auch Werte größer 7 zurückgeliefert werden können.

Funktion 27h**Bit-Masken des Bildschirm-Cursors abfragen***Eingabe:

AX = 0027h

Rückgabe:

AX = UND-Maske bei Software-/Startzeile bei Hardware-Cursor
 BX = XOR-Maske bei Software-/Endzeile bei Hardware-Cursor
 CX = Länge der horizontalen Bewegung in Mickeys
 DX = Länge der vertikalen Bewegung in Mickeys

Besonderheiten:

Die Werte in CX und DX sind stets 0, da ein IRT mit absoluten Koordinaten arbeitet.

Funktion 29h**Liste der verfügbaren Video-Modi abfragen***Eingabe:

AX = 0029h
 CX = 0 (ersten Video-Mode) bzw. <> 0 (nächste Modi) abfragen

Rückgabe:

BX = Segmentadresse eines Strings
 CX = Codenummer des Videomodus
 DX = Offsetadresse eines Strings

Besonderheiten:

In CX wird stets der Wert 0 zurückgeliefert, um anzuzeigen, daß keine Modi abgefragt werden können.

Funktion 2Bh	Beschleunigungskurve einstellen*
<u>Eingabe:</u>	AX = 002Bh BX = Nummer der zu aktivierenden Beschleunigungskurve ES = Segmentadresse der Kenndaten SI = Offsetadresse der Kenndaten
<u>Rückgabe:</u>	AX = FFFFh (Fehler) oder 0021h (ok) BX = Anzahl der Mausknöpfe
<u>Besonderheiten:</u>	In AX wird stets FFFFh zurückgeliefert, da Beschleunigungskurven in Zusammenhang mit einem IRT keinen Sinn machen.

Funktion 2Ch	Aktuelle Beschleunigungskurve auslesen*
<u>Eingabe:</u>	AX = 002Ch
<u>Rückgabe:</u>	AX = Funktionsstatus: <> 0 (Fehler) oder 0000h (ok) BX = Nummer der aktuellen Beschleunigungskurve (0..3) ES = Segmentadresse des Buffers SI = Offsetadresse des Buffers
<u>Besonderheiten:</u>	In AX wird stets FFFFh zurückgeliefert, da Beschleunigungskurven in Zusammenhang mit einem IRT keinen Sinn machen.

Funktion 2Dh	Aktuelle Beschleunigungskurve einstellen/abfragen*
<u>Eingabe:</u>	AX = 002Dh BX = -1 (aktuelle Beschleunigungskurve abfragen) oder BX = 1..4 (diese Beschleunigungskurve aktivieren)
<u>Rückgabe:</u>	AX = Funktionsstatus: <> 0 (Fehler) oder 0000h (ok) BX = Nummer der aktuellen Beschleunigungskurve (0..3) ES = Segmentadresse des Buffers mit Namen der Kurve SI = Offsetadresse des Buffers mit Namen der Kurve
<u>Besonderheiten:</u>	In AX wird stets FFFFh zurückgeliefert, da Beschleunigungskurven in Zusammenhang mit einem IRT keinen Sinn machen.

Funktion 30h	Einstellungen der Ballpoint-Maus setzen/abfragen*
<u>Eingabe:</u>	AX = 0030h BX = Drehwinkel CX = Kommando-Code
<u>Rückgabe:</u>	AX = Funktionsstatus: -1 (keine Ballpoint-Mouse) oder <> 0 (ok) BX = Drehwinkel CX = aktive Buttons
<u>Besonderheiten:</u>	In AX wird stets FFFFh zurückgeliefert, da CiTouchD keine Ballpoint-Mäuse unterstützt.

5.3 Erweiterte API-Funktionen

Die erweiterten API-Funktionen sind über den sogenannten Multiplex-Interrupt 2Fh erreichbar. Hierüber lassen sich verschiedene Parameter und Arbeitsmodi des Treibers ermitteln bzw. einstellen. Dabei muß die Kennnummer des CiTouchD-Treibers im AH-Register und die Nummer der angesprochenen Funktion im AL-Register übergeben werden. Da es sich hierbei um spezielle API-Funktionen des CiTouchD-Treibers handelt werden diese im folgenden genau beschrieben.

Die Mausemulation des CiTouchD-Treibers ist vollständig parametrisierbar. Diese Parameter werden beim Programmstart von der Kommandozeile oder aus der Datei CITOUCHE.INI gelesen. Der Inhalt dieser Datei ist in Kapitel „CITOUCHE.INI“ auf Seite 42 dokumentiert. Zum Ändern der Parameter zur Laufzeit sind eine Anzahl API-Funktionen vorgesehen, die alle über eine Int 2Fh (Multiplexer-Interrupt) Schnittstelle implementiert sind. Diese Funktionen werden im Folgenden beschrieben.

5.3.1 Ermittlung der Kennnummer von CiTouchD

Die Kennnummer des CiTouchD-Treibers muß bei jedem Aufruf einer erweiterten API-Funktion über den Multiplex-Interrupt im Register AH angegeben werden. Seinen Namen verdankt der Multiplex-Interrupt der Tatsache, daß über ihn nicht nur ein einziges (DOS-) Programm erreicht werden kann, sondern daß er für alle TSR-Programme offen ist, die eine Kommunikationsschnittstelle nach außen benötigen. Nachdem ein TSR-Programm bei seinem ersten Aufruf installiert ist, können weitere Aufrufe von der DOS-Kommandozeile aus dazu genutzt werden, bestimmte Parameter in der bereits installierten Kopie des Programms einzustellen oder das Programm wieder aus dem Speicher zu entfernen. Das neuerlich aufgerufene Programm tritt mit dem bereits installierten Teil dabei über den Multiplex-Interrupt in Kontakt. Ein Programm, das den Multiplex-Interrupt (kurz: MUX) nutzen will, muß sich zunächst selbst eine 8-Bit-Kennnummer geben. Die Kennnummern von 00h bis BFh sind dabei von Microsoft® für eigene DOS-Programme reserviert, der Bereich von C0h bis FFh ist jedoch frei und kann von Anwendungsprogrammen, d. h. von eigenen TSR-Programmen, genutzt werden.

Da die Wahl der Kennnummer innerhalb von C0h bis FFh freigestellt ist, besteht die Möglichkeit, daß mehrere Programme dieselbe Nummer besitzen. Um diesem Problem beizukommen muß jedes Programm, das den Multiplex-Interrupt nutzen will, eine Funktion für den Installations-Check bereitstellen. Diese wird ausgeführt, wenn beim Multiplexer-Aufruf im AH-Register die Kennnummer des jeweiligen Programms und im AL-Register der Wert 00h übergeben werden. Entdeckt einer der bereits installierten Handler seine Kennnummer, muß er daraufhin einen Wert ungleich 00h (in der Regel FFh) im AL-Register zurückliefern.

CiTouchD durchläuft auf diese Weise alle Kennnummern von C0h an aufwärts und „hängt“ sich an die erste freie Stelle.

Da bei dieser Vorgehensweise die Kennnummer von CiTouchD nicht im voraus festliegt, muß ein Programm, das die über den Multiplex-Interrupt 2Fh bereitgestellten Treiberfunktionen nutzen will, zuerst die momentan gültige Kennnummer ermitteln. Zur Ermittlung dieser ruft das Programm ausgehend von der Kennnummer C0h (im AH-Register zu speichern) die Funktion 00h (im AL-Register zu speichern) des Multiplex-Interrupts 2Fh auf, bis das Registerpaar ES:DI auf die Zeichenkette „CiTouchD“ zeigt. Die beim erfolgreichen Aufruf angegebene Kennnummer entspricht dann der aktuellen Kennnummer des CiTouchD-Treibers. Ausgehend von dieser Kennziffer können nun die über den Multiplex-Interrupt zur Verfügung gestellten Funktionen von CiTouchD aufgerufen werden.

Bleibt die Suche bis Kennnummer FFh erfolglos, ist CiTouchD nicht installiert.

Die folgenden Beispiele in den Programmiersprachen C, Pascal und Assembler nutzen diese Methode, um über die weiter unten beschriebene Funktion 25h (citDetectLDVGA) das Vorhandensein einer LDVGA Karte abzufragen.

C-Beispiel:

```

/*****
 detectLDVGA - Check if LDVGA graphics card is present (Microsoft C example)
 *****/

#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <string.h>

#define IDENSTR      "CiTouchD"
#define IDSTRLEN    8
#define PRESENCE    0x00 /* Literal for interrupt 2f/function 00h
*/

```

```

#define DETECTLDVGA    0x25 /* Literal for interupt 2F/function 25h/37d          */

int main( void )
{
    union _REGS ir, or;
    struct _SREGS sregs;
    int i;
    unsigned char idenNr; /* Identify number of CiTouchD
                           */
    void far *pIdenStr;

    for( i=0xC0; i<0x100; i++ )          /* ID number from 0xC0...0xFF are allowed
    /*                                     */
    {                                     /* for non system TSRs using MUX interupt
    /*                                     */
        ir.h.al = PRESENCE;                /* Verify presence function
        /*                                     */
        ir.h.ah = i;                       /* Identify number
        /*                                     */
        _int86x( 0x2F, &ir, &or, &sregs );

        /* Look, if a tsr with multiplex support is installed under
        this identify number

        */
        if( or.h.al == 0xFF )
        {
            /* point to given identifier string
            */
            pIdenStr = (void far*) ( ((unsigned long)sregs.es << 16) + or.x.di);

            /* Look, if identifier string matches searched one
            */
            if( !_fstrncmp( pIdenStr, IDENSTR, IDSTRLEN ) )
            {
                break;
            }
        }
    }

    if( i == 256 )
        printf( "\nCiTouchD not installed!\n" );
    else
    {
        idenNr = ( unsigned char )i;        /* store identify number
        /*                                     */

        /* detect LDVGA          */
        ir.h.al = DETECTLDVGA;
        ir.h.ah = idenNr;
        _int86x( 0x2F, &ir, &or, &sregs );
        if( or.x.ax )
            printf( "LDVGA detected.\n" );
        else
            printf( "No LDVGA detected.\n" );
    }

    return( 0 );
}

```

Pascal-Beispiel:

```

(*****
detectLDVGA - Check if LDVGA graphics card is present (Turbo Pascal example)
*****)

uses dos;                                (* For 'Intr' *)

const
    Idenstr      = 'CiTouchD';
    PRESENCE     = $00;
    citDetectLDVGA = $25;

var
    Regs        : Registers;
    Idenstr     : Byte;
    IdenstrPtr  : Pointer;

```

```

    TmpIden    : String;
    i          : Integer;
    j          : Byte;
    Found      : Boolean;

begin

    Found := False;
    i := $C0;                                (* ID number from $C0...$FF are
                                              allowed *)

    repeat
        Regs.AL := PRESENCE;                 (* Verify presence function      *)
        Regs.AH := Byte( i );               (* Identify number              *)
        Intr( $2F, Regs );

        (* Look, if a tsr with multiplex support is installed under this identify number
        *)
        if Regs.AL = $FF then begin
            IdenStrPtr := Ptr( Regs.ES, Regs.DI ); (* Point to given identifier string *)

            (* Convert given null terminated C-String in Pascal string and compare *)
            TmpIden := '';
            for j := 1 to 8 do begin
                TmpIden := TmpIden + Char( Ptr( Regs.ES, Regs.DI )^ );
                Regs.DI := Regs.DI + 1;
            end;

            if TmpIden = IdenStr then
                Found := TRUE;
            end;
            i := i + 1;
        until Found OR ( i > $FF );

        if Not Found then
            writeln( 'CiTouchD not installed!' )
        else begin
            idenNr := Byte( i - 1 );          (* store identify number *)

            (* detect LDVGA *)
            Regs.AL := citDetectLDVGA;
            Regs.AH := idenNr;
            Intr( $2F, Regs );
            if Regs.AX <> 0 then
                writeln( 'LDVGA found!' )
            else
                writeln( 'No LDVGA detected.' );
            end;
        end;
    end.

```

Assembler-Beispiel:

```

;////////////////////////////////////
;/// S w i t c h A - Switch back to TBT/Assembler example
;/// Assemble with MASM or TASM
;////////////////////////////////////

        .MODEL small, pascal
        .DOSSEG

CR          EQU 13t
LF          EQU 10t
IDLEN      EQU 8t
PRESENCE   EQU 000h
DETECTLDVGA EQU 025h

        .STACK
        .DATA

MsgErr    BYTE    CR, LF, "CiTouchD not installed!", CR, LF, '$'
MsgFound  BYTE    CR, LF, "LDVGA found!", CR, LF, '$'
MsgNoLDVGA BYTE    CR, LF, "No LDVGA detected.", CR, LF, '$'

IDStr     BYTE    'CiTouchD', 0                ;// Multiplex indentifier string
IDNr      BYTE    ?                            ;// Storage for found MUX code

```

```

.CODE

Start PROC NEAR

    mov     ax, DGROUP           ;// Initialize DS
    mov     ds, ax

    ;// Start searching with number 0C0h...0FFh
    mov     bx, 0C0h           ;// only bl is of interest
                                ;// (bx is used for easier overflow
                                ;// test)
    cld                               ;// String compare from left to right

SearchIDLoop:
    mov     al, PRESENCE       ;// verify presence function
    mov     ah, bl            ;// next identifier number to search for
    int     02Fh

    ;// Look if a tsr with multiplex support is installed under this identify number
    cmp     al, 0FFh
    jne     SearchIDLoopNext

    ;// Look, if identify string matches
    ;// (ES:DI points to 'CiTouchD' if right tsr)
    mov     cx, IDLEN
    mov     si, OFFSET IDStr   ;// DS:SI points to compare string
    REPE   cmpsb
    jnz     SearchIDLoopNext   ;// No match -> carry on search
    mov     IDNr, bl           ;// else store ID number and
    jmp     Detect             ;// search LDVGA

SearchIDLoopNext:
    inc     bx
    cmp     bx, 0100h         ;// look if search for all possible
                                ;// ID numbers
    jne     SearchIDLoop     ;// -> if not next try

    ;// ID String not found -> quit with error message
    mov     dx, OFFSET MsgErr   ;// DS:DX points to message
    jmp     PrintOut

    ;// check presence of LDVGA via multiplex function DETECTLDVGA
Detect:
    mov     al, DETECTLDVGA
    mov     ah, IDNr           ;// use examined ID number
    int     02Fh
    or     ax, ax             ;// have we found the LDVGA
    jz     NotFound          ;// -> NO
    mov     dx, OFFSET MsgFound ;// DS:DX points to message
    jmp     PrintOut

NotFound:
    mov     dx, OFFSET MsgNoLDVGA ;// DS:DX points to message

PrintOut:
    mov     ah, 09h           ;// Request print string function 09h
    int     21h

MainExit:
    mov     ax, 04C00h        ;// end program
    int     21h

Start ENDP

    END Start

```


5.3.2 Übersicht der erweiterten API-Funktionen

Es folgt eine Kurzübersicht aller CiTouchD-API-Funktionen, zusammengefaßt nach Funktionsgruppen.

5.3.2.1 Abfrage von CiTouchD-Parametern

Mit diesen Befehlen lassen sich sämtliche Konfigurationsparameter des CiTouchD-Treibers auslesen.

Nr.	Bezeichnung	Zurückgelieferte Daten	sh. Seite
05h	citGetFlags	Aktueller Treiberzustand	27
06h	citGetSerialHardware	Parameter der seriellen Schnittstelle	27
07h	citGetCommands	Parameter für Maustastenemulation	27
08h	citGetCoordMode	Parameter zur Koordinatenberechnung	28
09h	citGetCalibrationAbs	Kalibrierungsparameter für absolute Koordinaten	28
0Ah	citGetCalibrationRel	Kalibrierungsparameter für relative Koordinaten	28
0Bh	citGetDriverSettings	Veränderliche Treiberparameter	28
0Ch	citGetTouchSettings	Veränderliche Parameter des IRT	29
0Dh	citGetTouchHardware	Unveränderliche Parameter des IRT	29
0Eh	citGetDriverConstants	Unveränderliche Treiberparameter	29
1Bh	citGetVersion	Versionsnummer von CITOUCHEX.E	33
1Dh	citGetDimming	Puls-/Pausenverhältnisse des IRT PWM-Ausgangs	34
20h	citGetAcceleration	Parameter für die Cursorbeschleunigung	34
22h	citGetButtonBeep	Parameter für Maustastenklicks	35
27h	citGetCitdFlags	Erweiterter Treiberzustand	36

5.3.2.2 Verändern von CiTouchD-Parametern

Mit diesen Befehlen lassen sich sämtliche Konfigurationsparameter des CiTouchD-Treibers verändern.

Nr.	Bezeichnung	Veränderte Parameter	sh. Seite
0Fh	citSetSerialHardware	Parameter der seriellen Schnittstelle	29
10h	citSetCommands	Parameter für Maustastenemulation	30
11h	citSetCoordMode	Parameter zur Koordinatenberechnung	30
12h	citSetCalibrationAbs	Kalibrierungsparameter für absolute Koordinaten	30
13h	citSetCalibrationRel	Kalibrierungsparameter für relative Koordinaten	31
14h	citSetDriverSettings	Veränderliche Treiberparameter	31
15h	citSetTouchSettings	Veränderliche Parameter des IRT	31
1Eh	citSetDimming	Puls-/Pausenverhältnisse des IRT PWM-Ausgangs	34
21h	citSetAcceleration	Parameter für die Cursorbeschleunigung	35
23h	citSetButtonBeep	Parameter für Maustastenklicks	35
28h	citSetCitdFlags	Setzen des Hardware-Reset-Handlings	36

5.3.2.3 Kommunikation mit dem IRT

Der CiTouchD-Treiber stellt eine einfach zu handhabende Schnittstelle für das Senden von Befehlen zum IRT und das Empfangen von Berichten vom IRT zur Verfügung.

Bevor ein Anwenderprogramm mit dem IRT kommunizieren kann, muß der Empfangskanal mittels der Funktion **citOpen()** geöffnet werden. Ein CiTouchD-Treiber mit geöffnetem Empfangskanal meldet allerdings keine Cursorbewegungen oder Maustastendrucke mehr. Daher darf nach Empfang der gewünschten Berichte ein Aufruf von **citClose()** nicht vergessen werden! Es werden stets nur komplette Berichte des IRT an ein Anwenderprogramm übergeben. Für das Anwenderprogramm entfällt dadurch der Aufwand für die Erkennung von Berichtsgrenzen. Es können allerdings nur IRT-Funktionen verwendet werden die keine Neuinitialisierung des IRT erfordern. Insbesondere eine Neuprogrammierung des FLASH-Memorys auf dem IRT ist somit nicht möglich.

Für das Empfangen von IRT-Berichten werden zwei Methoden angeboten: *Polling* und *Notification*.

Beim *Polling* muß ständig über **citReceiveStatus()** abgefragt werden ob bereits ein kompletter Bericht vom IRT vorliegt. Ist dies der Fall, so kann der Bericht mit der Funktion **citReceive()** gelesen werden. Bei der *Notification* wird hingegen ein Event-Handler des Anwenderprogramms aufgerufen, sobald ein kompletter Bericht empfangen wurde. Die Adresse des Event-Handlers legt der Benutzer beim Aufruf der **citOpen()** Funktion fest. Der Aufbau der Benachrichtigung ist auf Seite 42 beschrieben.

Wird ein weiterer Bericht vom IRT empfangen obwohl der Empfangspuffer noch nicht ausgelesen wurde, wird der neu empfangene Bericht verworfen. Auf diese Weise ist sichergestellt, daß ein explizit angeforderter Bericht nicht von nachfolgenden Koordinatenmeldungen des IRT überschrieben werden kann. Mögliche Befehle für den IRT und der Aufbau von IRT-Berichten unterscheiden sich im Mode-C Kommunikationsprotokoll und im CTS1-Protokoll. Die jeweils gültigen Befehle sind im Handbuch zum IRT erläutert.

Nr.	Bezeichnung	Funktion	Seite
16h	citSend	Sendet ein Byte zum IRT	31
17h	citReceive	Empfängt einen kompletten Bericht vom IRT	32
18h	citOpen	Öffnet dem Empfangskanal vom IRT zum Rechner	32
19h	citClose	Schließt den Empfangskanal	33
1Ah	citReceiveStatus	Ermittelt den Zustand des Empfangskanals	33
1Ch	citCheckBreak	Prüft eine serielle Schnittstelle auf 100ms-Breaks	34

5.3.2.4 Allgemeine Hilfsfunktionen

Der CiTouchD-Treiber stellt neben den für die Mausemulation benötigten Schnittstellen auch noch einige nützliche Zusatzfunktionen zur Verfügung

Nr.	Befehl	Funktion	Seite
00h	citPresence	Installationsstatus abfragen	26
01h	citGetPSP	PSP des residenten Treibers ermitteln	26
25h	citDetectLDVGA	Prüft, ob eine LDVGA-Karte vorhanden ist	35
26h	citPlaySound	Gibt einen Ton auf dem PC-Lautsprecher aus	36

5.3.3 Referenz der erweiterten API-Funktionen

Es folgt eine Referenz aller erweiterten API-Funktionen in numerischer Reihenfolge. Diese sind über den Int 2Fh (Multiplex-Interrupt) erreichbar. Die Bestimmung der erforderlichen Kennung des CiTouchD-Treibers wurde weiter oben bereits beschrieben.

Funktion 00h

citPresence

Eingabe:

AL = 00h
AH = Kennung des CiTouchD-Treibers

Rückgabe:

AL = 0FFh Kennung gültig, ES:DI zeigt auf einen 8 Zeichen langen String
AL = 00h Kein residentes Programm unter dieser Kennung vorhanden, ES:DI ist in diesem Fall ungültig.

Beschreibung:

Wenn CiTouchD installiert ist, liefert die Funktion im AL-Register den Wert FFh und im Registerpaar ES:DI einen Zeiger auf den Identifizierungsstring „CiTouchD“ (ASCII-Z-Format) zurück. Dann, und nur dann können auch die anderen Funktionen des Multiplex-Interrupts aufgerufen werden. Der in AH anzugebende Wert hängt von der Rechnerkonfiguration ab. Eine genaue Erklärung findet sich im Kapitel "Ermittlung der Kennnummer" auf Seite 21.

Funktion 01h

citGetPSP

Eingabe:

AL = 01h
AH = Kennung des CiTouchD-Treibers

Rückgabe:

ES:DI = Segment und Offset des residenten Teils von CiTouchD

Beschreibung:

Diese Funktion liefert den PSP des residenten Teils des Treibers.

Funktion 05h**citGetFlags**Eingabe:

AL = 05h
 AH = Kennung des CiTouchD-Treibers

Rückgabe:

AX = Momentaner Zustand des CiTouchD-Treibers

Beschreibung:

Im Rückgabewort liegt der aktuelle Treiberzustand in Form von Bit-Flags vor. Die einzelnen Bits haben dabei die folgende Bedeutung:

Bit 3 = IF_SMOOTH_ALWAYS	Glättung auch zwischen zwei Berührungen
Bit 4 = IF_FIRST_COORDINATE	Erste Koordinatenmeldung
Bit 5 = IF_BEEP_MODE	Akkustische Rückmeldung bei Maustasteneignissen
Bit 6 = IF_ABSOLUTE_MODE	Es werden absolute Koordinaten verwendet
Bit 7 = IF_Z_TOUCH	Es wurde ein 3D-IRT gefunden
Bit 8 = IF_ENABLED	Der Treiber ist aktiviert
Bit 11 = IF_INT_ON	Die Hardware Interrupts sind freigegeben
Bit 12 = IF_BREAK	Es wurden BREAKs vom IRT empfangen
Bit 13 = IF_AUTOREINIT	Automatische Neuinitialisierung ist eingeschaltet
Bit 14 = IF_ON_SLAVEPIC	Der verwendete Interruptvektor befindet sich auf dem zweiten PIC des Rechners.
Bit 15 = IF_IRT_EXISTS	Ein IRT wurde gefunden

Funktion 06h**citGetSerialHardware**Eingabe:

AL = 06h
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine SERIALHARDWARE-Struktur zur Übernahme der seriellen Schnittstellenparameter.

Rückgabe:

AX = Anzahl der in die SERIALHARDWARE-Struktur geschriebenen Bytes

Beschreibung:

Es werden die seriellen Schnittstellenparameter in die übergebene SERIALHARDWARE-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 41

Funktion 07h**citGetCommands**Eingabe:

AL = 07h
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine COMMANDS-Struktur zur Übernahme der Parameter für die Maustastemulation.

Rückgabe:

AX = Anzahl der in die COMMANDS-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Parameter für die Maustastemulation in die übergebene COMMANDS-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 39. Eine Beschreibung der Funktion der Maustastemulation befindet sich in Kapitel "Benutzerdefinierte Tastemulation" auf Seite 12.

Funktion 08h**citGetCoordMode**Eingabe:

AL = 08h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine COORDMODE-Struktur zur Übernahme der Parameter zur Koordinatenausgabe.

Rückgabe:

AX = Anzahl der in die COORDMODE-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Parameter zur Koordinatenausgabe in die übergebene COORDMODE-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 40

Funktion 09h**citGetCalibrationAbs**Eingabe:

AL = 09h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine CALIBRATIONABS-Struktur zur Übernahme der absoluten Kalibrierungsparameter.

Rückgabe:

AX = Anzahl der in die CALIBRATIONABS-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Kalibrierungsparameter des absoluten Koordinatenmodus in die übergebene CALIBRATIONABS-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 38

Funktion 0Ah**citGetCalibrationRel**Eingabe:

AL = 0Ah
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine CALIBRATIONREL-Struktur zur Übernahme der relativen Kalibrierungsparameter.

Rückgabe:

AX = Anzahl der in die CALIBRATIONREL-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Kalibrierungsparameter des relativen Koordinatenmodus in die übergebene CALIBRATIONREL-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 38

Funktion 0Bh**citGetDriverSettings**Eingabe:

AL = 0Bh
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine DRIVERSETTINGS-Struktur zur Übernahme der Treiberparameter.

Rückgabe:

AX = Anzahl der in die DRIVERSETTINGS-Struktur geschriebenen Bytes

Beschreibung:

Es werden die veränderbaren Treiberparameter in die übergebene DRIVERSETTINGS-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 40

Funktion 0Ch**citGetTouchSettings**Eingabe:

AL = 0Ch
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine TOUCHSETTINGS-Struktur zur Übernahme der veränderbaren IRT-Parameter.

Rückgabe:

AX = Anzahl der in die TOUCHSETTINGS-Struktur geschriebenen Bytes

Beschreibung:

Es werden die veränderbaren IRT-Parameter in die übergebene TOUCHSETTINGS-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 42

Funktion 0Dh**citGetTouchHardware**Eingabe:

AL = 0Dh
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine TOUCHHARDWARE-Struktur zur Übernahme der konstanten IRT-Parameter.

Rückgabe:

AX = Anzahl der in die TOUCHHARDWARE-Struktur geschriebenen Bytes

Beschreibung:

Es werden die unveränderlichen IRT-Parameter in die übergebene TOUCHHARDWARE-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 41

Funktion 0Eh**citGetDriverConstants**Eingabe:

AL = 0Eh
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine DRIVERCONSTANTS-Struktur zur Übernahme der Treiberkonstanten.

Rückgabe:

AX = Anzahl der in die DRIVERCONSTANTS-Struktur geschriebenen Bytes

Beschreibung:

Es werden die unveränderlichen Treiberparameter in die übergebene DRIVERCONSTANTS-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 40

Funktion 0Fh**citSetSerialHardware**Eingabe:

AL = 0Fh
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine SERIALHARDWARE-Struktur mit den neuen seriellen Schnittstellenparametern.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
 AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die seriellen Schnittstellenparameter auf die übergebene SERIALHARDWARE-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 41
 Falls ein Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten.

Funktion 10h**citSetCommands**Eingabe:

AL = 10h
AH = Kennung des CiTouchD-Treibers
ES:DI= Zeiger auf eine COMMANDS-Struktur mit den neuen Parameter für die Maustastenemulation.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Parameter für die Maustastenemulation auf die übergebene COMMANDS-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 39. Eine Beschreibung der Funktion der Maustastenemulation befindet sich in Kapitel "Benutzerdefinierte Tastenemulation" auf Seite 12.
Falls ein Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten

Funktion 11h**citSetCoordMode**Eingabe:

AL = 11h
AH = Kennung des CiTouchD-Treibers
ES:DI= Zeiger auf eine COORDMODE-Struktur mit den neuen Parametern zur Koordinatenausgabe.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Parameter zur Koordinatenausgabe auf die übergebene COORDMODE-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 40
Falls ein Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten

Funktion 12h**citSetCalibrationAbs**Eingabe:

AL = 12h
AH = Kennung des CiTouchD-Treibers
ES:DI= Zeiger auf eine CALIBRATIONABS-Struktur mit den neuen absoluten Kalibrierungsparametern.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
EX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Kalibrierungsparameter des absoluten Koordinatenmodus auf die übergebene CALIBRATIONABS-Struktur geändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 38
Falls das Ändern der Parameter nicht möglich war, bleibt der alte Zustand erhalten.

Funktion 13h**citSetCalibrationRel**Eingabe:

AL = 13h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine CALIBRATIONREL-Struktur mit den neuen relativen Kalibrierungsparametern.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Kalibrierungsparameter des relativen Koordinatenmodus auf die übergebene CALIBRATIONREL-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 38. Falls das Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten.

Funktion 14h**citSetDriverSettings**Eingabe:

AL = 14h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine DRIVERSETTINGS-Struktur mit den neuen Treiberparametern.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Treiberparameter auf die übergebene DRIVERSETTINGS-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 40. Falls ein Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten.

Funktion 15h**citSetTouchSettings**Eingabe:

AL = 15h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine TOUCHSETTINGS-Struktur mit den neuen IRT-Parametern.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die IRT-Parameter auf die übergebene TOUCHSETTINGS-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 42. Falls ein Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten.

Funktion 16h**citSend**Eingabe:

AL = 16h
AH = Kennung des CiTouchD-Treibers
BL = Datenbyte welches zum IRT gesendet werden soll

Rückgabe:

keine

Beschreibung:

Es wird ein Byte zum IRT gesendet. Falls der Sendepuffer des Schnittstellenbausteins nicht leer sein sollte, wird solange gewartet bis dieser leer ist. Befehle im CTS1-Protokoll müssen bereits kodiert sein, d.h. die DC2/DC4 und SYN-Sequenzen müssen vom Anwenderprogramm eingefügt werden.

Funktion 17h**citReceive**Eingabe:

AL = 17h
 AH = Kennung des CiTouchD-Treibers
 ES:DI= Zeiger auf einen Puffer in welchen der empfangene Bericht kopiert wird.

Rückgabe:

AX = Zustand des Empfangspuffers
 Mögliche Rückgabewerte sind:

00h = CRS_IDLE	Kein Bericht verfügbar
01h = CRS_READY	Ein Bericht wurde empfangen
02h = CRS_OVER	Es wurden weitere Berichte empfangen. Der zuerst empfangene Bericht befindet sich aber noch vollständig im Puffer.
04h = CRS_CLOSE	Der Treiber wurde nicht durch citOpen (Funktion 18h) geöffnet.

Beschreibung:

Wurde ein kompletter Bericht vom IRT empfangen, so kann er mit dieser Funktion ausgelesen werden. Falls kein Bericht vorliegt wird nicht gewartet, sondern mit einem entsprechenden Zustandscode sofort zurückgekehrt. Die maximal erforderliche Größe für den Empfangspuffer kann mit der Funktion **citGetDriverConstants** (Funktion **0Eh**) angefragt werden. Berichte im CTS1-Protokoll sind bereits dekodiert, d.h. sie enthalten keine DC2/DC4 und SYN-Sequenzen mehr.

Funktion 18h**citOpen**Eingabe:

AL = 18h
 AH = Kennung des CiTouchD-Treibers
 ES:DI= Zeiger auf einen Eventhandler, oder NULL, falls keine Benachrichtigungen verschickt werden sollen.

Rückgabe:

AX = 0001h Der Empfangskanal konnte geöffnet werden
 AX = 0000h Der Empfangskanal konnte nicht geöffnet werden. Mögliche Gründe hierfür sind ein bereits geöffneter Empfangskanal oder ein nicht initialisierter Treiber.

Beschreibung:

Bevor ein Anwenderprogramm Berichte des IRT empfangen kann muß zuerst der Empfangskanal geöffnet werden. Ein CiTouchD-Treiber mit geöffnetem Empfangskanal meldet allerdings keine Cursorbewegungen oder Maustastendrucke mehr. Daher darf nach Empfang der gewünschten Berichte ein Aufruf von **citClose** (Funktion **19h**) nicht vergessen werden!

Funktion 19h**citClose**Eingabe:

AL = 19h
 AH = Kennung des CiTouchD-Treibers

Rückgabe:

AX = 0001h Empfangskanal konnte geschlossen werden
 AX = 0000h Empfangskanal konnte nicht geschlossen werden.
 Mögliche Ursachen dafür sind ein bereits geschlossener
 Empfangskanal oder ein nicht initialisierter Treiber.

Beschreibung:

Damit der CiTouchD-Treiber wieder seine normale Funktion als Maustreiber aufnehmen kann muß der Empfangskanal, nachdem er geöffnet und alle interessierenden Berichte gelesen wurden, wieder geschlossen werden.

Funktion 1Ah**citReceiveStatus**Eingabe:

AL = 1Ah
 AH = Kennung des CiTouchD-Treibers

Rückgabe:

AX = Zustand des Empfangspuffers
 Mögliche Rückgabewerte sind:

00h = CRS_IDLE	Kein Bericht verfügbar
01h = CRS_READY	Ein Bericht wurde empfangen
02h = CRS_OVER	Es wurden weitere Berichte empfangen. Der zuerst empfangene Bericht befindet sich aber noch vollständig im Puffer.
04h = CRS_CLOSE	Der Treiber wurde nicht durch citOpen (Funktion 18h) geöffnet.

Beschreibung:

Es wird der momentane Zustand des Empfangspuffers ermittelt.

Funktion 1Bh**Versionsnummer von CITOUCHD.EXE**Eingabe:

AL = 1Bh
 AH = Kennung des CiTouchD-Treibers

Rückgabe:

DH = Hauptversionsnummer
 DL = Subversionsnummer
 AH = Revisionsnummer
 AL = Stadium

Beschreibung:

Es wird die Versionsnummer von CiTouchD ermittelt. Über die Funktion int33Inquire wird hingegen die Microsoft® Maustreiber kompatible Version zurückgeliefert.

Funktion 1Ch**citCheckBreak**Eingabe:

AL = 1Ch
 AH = Kennung des CiTouchD-Treibers

Rückgabe:

AX = 0001h Es wurden 100ms-Breaks gefunden
 AX = 0000h Es wurden keine Breaks gefunden

Beschreibung:

Ein uninitialisierter IRT sendet in Abständen von 100 ms BREAK-Signale. Mit Hilfe dieser Funktion kann eine serielle Schnittstelle auf solche Signale untersucht werden.

Funktion 1Dh**citGetDimming**Eingabe:

AL = 1Dh
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine DIMMING-Struktur zur Übernahme der Dimming-Parameter.

Rückgabe:

AX = Anzahl der in die DIMMING-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Parameter zur Ansteuerung des PWM-Ausgangs des IRT in die übergebene DIMMING-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 40

Funktion 1Eh**citSetDimming**Eingabe:

AL = 1Eh
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine DIMMING-Struktur mit den neuen Parametern für die Ansteuerung des PWM-Ausgangs des IRT.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
 AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Parameter für die Ansteuerung des PWM-Ausgangs des IRT auf die übergebene DIMMING-Struktur abgeändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 40
 Falls ein Ändern der Parameter nicht möglich war bleibt der alte Zustand erhalten.

Funktion 20h**citGetAcceleration**Eingabe:

AL = 20h
 AH = Kennung des CiTouchD-Treibers
 ES:DI = Zeiger auf eine ACCELERATION-Struktur zur Übernahme der Parameter für die Cursorbeschleunigung

Rückgabe:

AX = Anzahl der in die ACCELERATION-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Parameter für die Cursorbeschleunigung in die übergebene ACCELERATION-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 37

Funktion 21h**citSetAcceleration**Eingabe:

AL = 21h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine ACCELERATION-Struktur mit den neuen Parametern für die Cursorbeschleunigung.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Parameter für die Cursorbeschleunigung auf die übergebene ACCELERATION-Struktur geändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 37
Falls das Ändern der Parameter nicht möglich war, bleibt der alte Zustand erhalten.

Funktion 22h**citGetButtonBeep**Eingabe:

AL = 22h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine BUTTONBEEP-Struktur zur Übernahme der Maustastenklick-Parameter.

Rückgabe:

AX = Anzahl der in die BUTTONBEEP-Struktur geschriebenen Bytes

Beschreibung:

Es werden die Parameter für die Erzeugung des Maustastenklicks in die übergebene BUTTONBEEP-Struktur eingetragen. Eine Beschreibung dieser Struktur befindet sich auf Seite 38

Funktion 23h**citSetButtonBeep**Eingabe:

AL = 23h
AH = Kennung des CiTouchD-Treibers
ES:DI = Zeiger auf eine BUTTONBEEP-Struktur mit den neuen Parametern für die Maustastenklickerzeugung.

Rückgabe:

AX = 0001h Die neuen Parameter konnten eingestellt werden
AX = 0000h Die Parameter konnten nicht geändert werden

Beschreibung:

Es werden die Parameter für die Maustastenklickerzeugung auf die übergebene BUTTONBEEP-Struktur geändert. Eine Beschreibung dieser Struktur befindet sich auf Seite 38
Falls das Ändern der Parameter nicht möglich war, bleibt der alte Zustand erhalten.

Funktion 25h**citDetectLDVGA**Eingabe:

AL = 25h
AH = Kennung des CiTouchD-Treibers

Rückgabe:

AX = 0001h Es wurde eine Citron LDVGA-Karte gefunden
AX = 0000h Eine Citron LDVGA-Karte ist nicht vorhanden.

Beschreibung:

Diese Funktion stellt fest, ob eine Citron Long-Distance-VGA-Karte vorhanden ist.

Funktion 26h**citPlaySound**Eingabe:

AL = 26h
 AH = Kennung des CiTouchD-Treibers
 BX = Tonhöhe in Hertz [Hz]
 CX = Dauer in Millisekunden [ms]

Rückgabe:

keine

Beschreibung:

Diese Funktion gibt einen Ton mit der gewünschten Höhe und Dauer auf dem PC-Lautsprecher aus. Die Funktion kehrt zurück sobald die Tonausgabe gestartet wurde. Der Rechner wird daher nicht für die Dauer des Tons blockiert.

Funktion 27h**Erweiterter Treiberzustand**Eingabe:

AL = 27h
 AH = Kennung des CiTouchD-Treibers

Rückgabe:

AX = Momentaner Arbeitsmodus des CiTouchD-Treibers

Beschreibung:

Der Rückgabewert stellt den Arbeitsmodus als Bitfeld vor. Die Bits haben dabei folgende Bedeutung

Bit 0 = CTDF_VESA -> VGA-Karte unterstützt VESA Standard
 Bit 1 = CTDF_CHANGEMODE -> Video Modus wurde geändert
 Bit 2 = CTDF_TEXT -> Grafikkarte arbeitet im Textmodus
 Bit 3 = CTDF_SOFT -> Verwendung des Software-Textcursors
 Bit 4 = CTDF_EGA -> EGA-Karte gefunden
 Bit 5 = CTDF_VGA -> VGA-Karte gefunden
 Bit 6 = CTDF_MONO -> monochromer EGA/VGA Video Modus
 Bit 7 = CTDF_EXTVIDEO -> Funktion des erweiterten Video-BIOS
 Bit 8 = CTDF_STACK -> CiTouchD nutzt eigenen Stack
 Bit 9 = CTDF_HARDRESET0 -> Touch-Reset in Fkt. Int33h/00h
 Bit 10 = CTDF_HARDRESET2F -> Touch-Reset in Fkt. Int33/2Fh
 Bit 11 = CTDF_HIDE -> Keine eigener Cursor unter WINDOWS
 Bit 12 = CTDF_SYS -> CiTouchD in CONFIG.SYS geladen
 Bit 13 = CTDF_CONDOFF -> Conditional off ist aktiv

Diese Version von CiTouchD unterstützt CTDF_EGA, CTDF_EXTVIDEO und CTDF_SYS noch nicht. Diese Bits sind stets 0.

Funktion 28h**Setzen des Hardware-Reset-Handlings**Eingabe:

AL = 28h
 AH = Kennung des CiTouchD-Treibers
 BX = Zu ändernde Flags

Rückgabe:

AX = 0, falls Read-Only-Flags geändert werden sollten.

Beschreibung:

Über diese Funktion läßt sich das Reset-Verhalten des Touches bei Aufruf der Int33h-Funktionen 00h und 2Fh über die Flags CTDF_HARDRESET0 Bit 9 und CTDF_HARDRESET2F Bit 10 ändern.

Alle anderen Flags sind Read-Only und müssen in BX auf 0 gesetzt sein.

5.3.4 Referenz der Strukturen

Zur Parameterübergabe an die erweiterten API-Funktionen sind zahlreiche Strukturen definiert. Es folgt eine alphabetische Auflistung dieser Strukturen. Dabei ist zu beachten, daß von den API-Funktionen keinerlei zusätzliche Füllbytes zwischen den Strukturelementen erwartet werden. Es werden die folgenden Datentypen verwendet:

Typ	Länge	Wertebereich	Beschreibung
char	1 Byte	-128...127	vorzeichenbehafteter Integerwert
BYTE	1 Byte	0...255	vorzeichenloser Integerwert
short	2 Byte	-32768...32767	vorzeichenbehafteter Integerwert
WORD	2 Byte	0...65535	vorzeichenloser Integerwert
BOOL	2 Byte	0, 1	vorzeichenloser Integerwert, Werte ungleich 0 werden als WAHR, der Wert 0 wird als FALSCH interpretiert.
DWORD	4 Byte	0...4294967295	vorzeichenloser Integerwert

ACCELERATION

Diese Struktur enthält die Parameter für die Cursorbeschleunigung.

Typ	Name	Beschreibung
short	acMulX	Beschleunigungsfaktor für X-Koordinaten
short	acMulY	Beschleunigungsfaktor für Y-Koordinaten
WORD	acBorderX	Breite des Beschleunigungsbereichs (in Touch-Koord.!))
WORD	acBorderY	Höhe des Beschleunigungsbereichs (in Touch-Koord.!))

Die Größen *acMulX* und *acMulY* sind vorzeichenbehaftete Festkommazahlen. Ihr Wertebereich wird über die Formeln

$$acMulX_{\max} = \frac{32767 * dcFixedBias}{caMulX}$$

$$acMulY_{\max} = \frac{32767 * dcFixedBias}{caMulY}$$

berechnet. Der Skalierungsfaktor *dcFixedBias* für die Festkommazahlen kann mittels **citGetDriverConstants** (Funktion **0Eh**) und *caMulX* bzw. *caMulY* mittels **citGetCalibrationAbs** (Funktion **09h**) abgefragt werden.

Die Größen *acBorderX* und *acBorderY* werden in Touch-Koordinaten angegeben. Der Wertebereich beträgt:

$$acBorderX_{\max} = 32767$$

$$acBorderY_{\max} = 32767$$

BUTTONBEEP

Diese Struktur enthält die Parameter für die Maustastenklickerzeugung.

Typ	Name	Beschreibung
WORD	bbFreqDown	Frequenz des Tones beim Niederdrücken der Maustaste in Hertz [Hz]
WORD	bbTimeDown	Dauer des Tones beim Niederdrücken der Maustaste in Millisekunden [ms]
WORD	bbFreqUp	Frequenz des Tones beim Loslassen der Maustaste in Hertz [Hz]
WORD	bbTimeUp	Dauer des Tones beim Loslassen der Maustaste in Millisekunden [ms]
WORD	bbClickMode	Eine Kombination der folgenden Konstanten die festlegt, wann ein Ton erzeugt wird: 00h = BC_OFF Keine Tonerzeugung 01h = BC_DOWN Tonausgabe beim Drücken der emulierten Maustaste 02h = BC_UP Tonausgabe beim Loslassen der emulierten Maustaste

CALIBRATIONABS

Diese Struktur enthält die Parameter für die Kalibrierung absoluter Koordinaten. Die verwendete Formel zur Kalibrierung absoluter Koordinaten ist eine einfache Geradengleichung:

$$y = mx + t, \text{ mit } y = \text{kalibrierte Koordinate und } x = \text{vom IRT gelieferte Koordinate}$$

CiTouchD erwartet kalibrierte Koordinaten mit einem Wertebereich von 0 bis 65535.

Typ	Name	Beschreibung
DWORD	caMulX	"m" für die X-Koordinate
short	caAddX	"t" für die X-Koordinate
DWORD	caMulY	"m" für die Y-Koordinate
short	caAddY	"t" für die Y-Koordinate
WORD	caOrientation	Orientierung des IRT gegenüber dem Display. Dabei wird einer der folgenden Werte, entsprechend der Position des IRT Anschlußsteckers gegenüber der linken oberen Display-Ecke, angegeben: 00h = OR_TOPLEFT linke, obere Ecke 01h = OR_TOPRIGHT rechte, obere Ecke 02h = OR_BOTTOMRIGHT rechte, untere Ecke 03h = OR_BOTTOMLEFT linke, untere Ecke

Die Größen *caMulX* und *caMulY* sind vorzeichenlose Festkommazahlen. Der Skalierungsfaktor für die Festkommazahlen kann mittels **citGetDriverConstants** (Funktion **0Eh**) abgefragt werden.

CALIBRATIONREL

Diese Struktur enthält die Parameter für die Kalibrierung relativer Koordinaten. Die verwendete Formel zur Kalibrierung relativer Koordinaten lautet:

$$y = x \cdot \frac{m}{d}$$

Typ	Name	Beschreibung
short	crMulX	"m" für die X-Koordinate
WORD	crDivX	"d" für die X-Koordinate
short	crMulY	"m" für die Y-Koordinate
WORD	crDivY	"d" für die Y-Koordinate

COMMANDS

Diese Struktur enthält die Parameter für die Maustastenemulation. Die Erzeugung eines Maustastendrucks ist in Kapitel "Benutzerdefinierte Tastenemulation" auf Seite 12 beschrieben

Typ	Name	Beschreibung
WORD	cmdT1	erste Zeitkonstante in Einheiten von 1 ms
WORD	cmdT2	zweite Zeitkonstante in Einheiten von 1 ms
WORD	cmdT3	dritte Zeitkonstante in Einheiten von 1 ms
BYTE	cmdIdleT1	Bedingung 1
BYTE	cmdT1Trigger	Bedingung 2
BYTE	cmdTriggerT2	Bedingung 3
BYTE	cmdT2UpT3	Bedingung 4
BYTE	cmdUpT3Idle	Bedingung 5
BYTE	cmdUpT3Trigger	Bedingung 6
BYTE	cmdModeChange	Bedingung für Wechsel zwischen absoluten und relativen Koordinaten
BYTE	cmdReserved1	reserviert

Mögliche Werte für die Bedingungen 1 bis 6 lauten:

Wert	Bezeichnung	Bedeutung
00h	BM_NEVER	niemals WAHR
01h	BM_IMMED	immer WAHR
02h	BM_ENTER	WAHR, sobald der Touchbereich unterbrochen wird
03h	BM_ZPRESS	WAHR, sobald ein bestimmter Druck auf den Touchbereich überschritten wird
04h	BM_DUAL	WAHR, sobald eine Doppelberührung des Touchbereichs vorliegt
05h	BM_TAP	WAHR, sobald der Touchbereich für kurze Zeit verlassen und dann erneut unterbrochen wird.
06h	BM_LEAVE	WAHR, sobald der Touchbereich wieder verlassen wird
07h	BM_ZRLSE	WAHR, sobald der Druck auf den Touchbereich wieder unterschritten wird
08h	BM_NODUAL	WAHR, sobald keine Doppelberührung des Touchbereichs mehr vorliegt

Für das Feld "cmdModeChange" können die folgenden Werte verwendet werden:

Wert	Bezeichnung	Bedeutung
00h	MC_NEVER	Niemals umschalten
03h	MC_ZPRESS	WAHR, sobald ein bestimmter Druck auf den Touchbereich überschritten wird
04h	MC_DUAL	WAHR, sobald eine Doppelberührung des Touchbereichs vorliegt
05h	MC_TAP	WAHR, sobald der Touchbereich für kurze Zeit verlassen und dann erneut unterbrochen wird.

COORDMODE

Diese Struktur enthält die Parameter für die Koordinatenerzeugung.

Typ	Name	Beschreibung	
BOOL	cmCoordEnterZ	TRUE	Für die erste Koordinatenmeldung nach dem Eintauchen ist die Überschreitung des eingestellten Drucks auf die Frontscheibe erforderlich.
		FALSE	Die erste Koordinatenmeldung erfolgt sofort nach dem Eintauchen.
BOOL	cmCoordSignalZ	TRUE	Für alle weiteren Koordinatenmeldung ist die Überschreitung des eingestellten Drucks auf die Frontscheibe ebenfalls erforderlich.
		FALSE	Für weitere Koordinatenmeldungen genügt es, daß der Touchbereich unterbrochen bleibt.

DIMMING

Diese Struktur enthält die Parameter für die Ansteuerung des PWM-Ausgangs des IRT.

Typ	Name	Beschreibung
WORD	blDimmingHigh	Puls-/Pausenverhältnis bei inaktivem Touch-Saver
WORD	blDimmingLow	Puls-/Pausenverhältnis bei aktivem Touch-Saver
BOOL	blSaverActive	TRUE bei aktivem Touch-Saver

Der Wertebereich für *blDimmingHigh* bzw. *blDimmingLow* reicht von 0 bis 255. Sind beide Werte gleich 0, so wird das Puls-/Pausenverhältnis unabhängig vom Zustand des Touch-Savers auf den Maximalwert eingestellt. Die Aktivierungszeit des Touch-Savers wird über **citSetTouchSettings** (Funktion **15h**) festgelegt.

Der *blSaverActive*-Wert wird von **citGetDimming** (Funktion **1Dh**) gesetzt. Die Funktion **citSetDimming** (Funktion **1Eh**) ignoriert diesen Parameter. Der Touch-Saver kann hierüber nicht ein- oder ausgeschaltet werden!

DRIVERCONSTANTS

Diese Struktur enthält die unveränderlichen Parameter des CiTouchD-Treibers.

Typ	Name	Beschreibung
WORD	dcSmoothMax	Obere Grenze für die Glättungsfaktoren. Glättungsfaktoren müssen <u>kleiner</u> als der hier angegebene Wert sein.
WORD	dcFixedBias	Skalierungsfaktor für das Festkommaformat zur Kalibrierung absoluter Koordinaten.
WORD	dcReportMax	Maximale Länge des Puffers für empfangene IRT-Berichte.
WORD	dcKeyNum	Der IRT wird im Tastenmodus betrieben wobei eine einzige Taste mit der hier angegebenen Nummer definiert wird.
WORD	dcKeyMode	Betriebsart der Taste im Mode-C Protokoll

DRIVERSETTINGS

Diese Struktur enthält die einstellbaren Parameter des CiTouchD-Treibers.

Typ	Name	Beschreibung
WORD	dsSmoothX	Glättungsfaktor für die X-Achse
WORD	dsSmoothY	Glättungsfaktor für die Y-Achse
short	dsOfsX	X-Abstand zwischen Berührungspunkt und Cursorposition
short	dsOfsY	Y-Abstand zwischen Berührungspunkt und Cursorposition

WORD	dsTapTime	Zeit, innerhalb derer der Touchbereich erneut unterbrochen werden muß damit ein "Tap" erkannt wird. Die TapTime wird in Einheiten von 55 ms angegeben.
WORD	dsCoordSkip	Anzahl der Koordinatenmeldungen die nach dem Eintauchen verworfen werden bevor eine neue Cursorposition gemeldet wird.
WORD	dsDberrSkip	Anzahl der Zweifachberührungsmeldungen die verworfen werden bevor eine Zweifachberührung erkannt wird.
WORD	dsButtonNum	Nummer der zu emulierenden Maustaste. 1 = linke Maustaste 2 = rechte Maustaste 3 = beide Maustasten gleichzeitig
BOOL	dsAutoInIt	TRUE, wenn nach einer Unterbrechung der Verbindung zwischen IRT und Rechner automatisch neu angekoppelt werden soll.
BOOL	dsButtonClick	TRUE, wenn bei einem Maustastendruck zusätzlich eine akustische Rückmeldung erfolgen soll.
BOOL	dsAbsolute	TRUE, wenn absolute Koordinaten verwendet werden sollen.
BOOL	dsSmoothAlways	TRUE, wenn die Koordinatenglättung auch über ein Verlassen und erneutes Eintauchen hinweg erfolgen soll.

SERIALHARDWARE

Diese Struktur enthält die Parameter der seriellen Schnittstelle.

Typ	Name	Beschreibung
WORD	shPort	Basisadresse des seriellen Schnittstellenbausteins
short	shInterrupt	Zugehöriger ISA-Bus Interruptkanal
WORD	shBaudDiv	Teiler für die Baudrate

TOUCHHARDWARE

Diese Struktur enthält die unveränderlichen Parameter des IRT.

Typ	Name	Beschreibung
WORD	thBeamsX	Anzahl der physikalisch vorhandenen X-Lichtschranken
WORD	thBeamsY	Anzahl der physikalisch vorhandenen Y-Lichtschranken
WORD	thResolutionX	Maximale vom IRT gelieferte X-Koordinate
WORD	thResolutionY	Maximale vom IRT gelieferte Y-Koordinate
WORD	thProtocol	Vom IRT verwendetes Kommunikationsprotokoll. Dieses Feld kann einen der durch die TP_??? Konstanten definierten Werte annehmen.
char	thDesignator[33]	Bezeichnung des IRT (0-terminierter String)
char	thAssy[17]	ASSY-Nummer des IRT (0-terminierter String)
char	thMem	'E' = EPROM, 'F' = FLASH-Memory
BYTE	thReserved1	reserviert
char	thComment[257]	Im Mode-C Protokoll: Optionaler Kommentar (0-terminierter String) Im CTS1-Protokoll: Seriennummer des IRT (0-terminierter String)

TOUCHSETTINGS

Diese Struktur enthält die einstellbaren Parameter des IRT.

Typ	Name	Beschreibung
WORD	tsMinBeamsX	Minimale Anzahl unterbrochener X-Lichtschranken
WORD	tsMinBeamsY	Minimale Anzahl unterbrochener Y-Lichtschranken
WORD	tsBeamTimeout	Ausblendzeit für fehlerhafte Lichtschranken, in Einheiten von 1 s.
WORD	tsTCont	Abstand zwischen zwei Koordinatenmeldungen in Einheiten von 1 ms.
WORD	tsPressLevel	Druckempfindlichkeit
WORD	tsTSaver	Zeit nach welcher der Touch-Saver aktiviert wird, in Einheiten von 1 s.
WORD	tsTScan	Abstand zwischen zwei Scanvorgängen bei aktivem Touch-Saver, in Einheiten von 1 ms.

5.3.5 Benachrichtigung durch CiTouchD

Wurde beim Aufruf von **citOpen** (Funktion **18h**) ein Zeiger auf einen Event-Handler übergeben, so ruft CiTouchD, sobald ein kompletter Bericht des IRT empfangen wurde, diesen Event-Handler auf. Dabei ist zu beachten, daß der Event-Handler innerhalb der Interrupt-Service-Routine von CiTouchD aufgerufen wird und daher möglichst schnell wieder zurückkehren sollte. Insbesondere dürfen keine DOS- bzw. BIOS-Funktionen aufgerufen werden! Die Register sind beim Aufruf des Event-Handlers auf die folgenden Werte gesetzt:

AX	Zustand des Empfangspuffers (wie bei citReceive , Funktion 17h)
CX	Anzahl der Bytes im Empfangspuffer
ES:DI	Zeiger auf den Empfangspuffer, von hier darf nur gelesen werden!

Der Event-Handler muß einen der folgenden Werte in AX zurückliefern:

00h	Empfangspuffer wurde nicht gelesen, Zustand des Puffers soll nicht verändert werden.
01h	Empfangspuffer wurde gelesen, Zustand der Empfangspuffer soll gelöscht werden.

6 CITOUCHD.INI

In der Datei CITOUCHD.INI speichert der CiTouchD-Treiber seine Parameter. Beim Start von CiTouchD werden die hier abgelegten Werte verwendet, sofern sie nicht durch Kommandozeilenparameter überschrieben werden.

Im folgenden werden die einzelnen Abschnitte und Einträge der Datei CITOUCHD.INI in alphabetischer Reihenfolge aufgeführt. Ist ein Eintrag nicht vorhanden, so wird der im Treiber fest eingespeicherte Vorgabewert, oder der entsprechende Kommandozeilenparameter, verwendet.

Die folgenden Abschnitte werden in CITOUCHD.INI erkannt:

Abschnitt	Funktion
[Acceleration]	Parameter zur Cursorbeschleunigung
[Calibration]	Kalibrierung absoluter und relativer Koordinaten
[Commands]	Maustastenemulation
[Hardware]	Parameter der serielle Schnittstelle
[Settings]	Allgemeine Einstellungen für den Treiber und IRT
[Sound]	Einstellungen für die Maustastenklickerzeugung
[Citouchd]	Einstellungen für Verhalten von CiTouchD als Maustreiber

6.1.1 [Acceleration]

In diesem Abschnitt sind die Parameter für die Beschleunigung der Cursorbewegung, bei Verwendung absoluter Koordinaten, enthalten.

Im Abschnitt **[Acceleration]** sind die folgenden Einträge möglich:

$$X_Border = 0..32767$$

Dieser Eintrag bestimmt die Breite des Bereichs innerhalb dessen der Cursor gegenüber dem Finger beschleunigt bewegt wird. Der Vorgabewert ist **7864**, entsprechend 24%.

$$Y_Border = 0..32767$$

Dieser Eintrag bestimmt die Höhe des Bereichs innerhalb dessen der Cursor gegenüber dem Finger beschleunigt bewegt wird. Der Vorgabewert ist **7864**, entsprechend 24%.

$$X_Mul = 0..(32767*dcFixedBias)/caMulX$$

Dieser Eintrag bestimmt den Faktor um den der Cursor gegenüber dem Finger in X-Richtung beschleunigt wird. Der Vorgabewert ist **256**, entsprechend dem Faktor 1,0.

$$Y_Mul = 0..(32767*dcFixedBias)/caMulY$$

Dieser Eintrag bestimmt den Faktor um den der Cursor gegenüber dem Finger in Y-Richtung beschleunigt wird. Der Vorgabewert ist **256**, entsprechend dem Faktor 1,0.

6.1.2 [Calibration]

In diesem Abschnitt sind die Kalibrierungsparameter für absolute und relative Koordinaten enthalten. Im Abschnitt **[Calibrations]** sind die folgenden Einträge möglich:

$$X_Add = -32767..32767$$

Dieser Eintrag bestimmt den Offset der Geradengleichung zur Kalibrierung absoluter X-Koordinaten. Der Vorgabewert ist **0**.

$$X_Mul = 0..65535$$

Dieser Eintrag bestimmt die Steigung der Geradengleichung zur Kalibrierung absoluter X-Koordinaten. X_Mul ist eine vorzeichenlose Festkommazahl. Der Skalierungsfaktor kann mit der Funktion **citGetDriverConstants** abgefragt werden. Der Vorgabewert ist **256**.

$$XRel_Div = 0..65535$$

Dieser Eintrag bestimmt den Teiler zur Skalierung relativer X-Koordinaten. Der Vorgabewert ist **1**.

$$XRel_Mul = -32768..+32767$$

Dieser Eintrag bestimmt den Multiplikator zur Skalierung relativer X-Koordinaten. Der Vorgabewert ist **-1**.

$$Y_Add = -32767..32767$$

Dieser Eintrag bestimmt den Offset der Geradengleichung zur Kalibrierung absoluter Y-Koordinaten. Der Vorgabewert ist **0**.

Y_Mul = 0..65535

Dieser Eintrag bestimmt die Steigung der Geradengleichung zur Kalibrierung absoluter Y-Koordinaten. Y_Mul ist eine vorzeichenlose Festkommazahl. Der Skalierungsfaktor kann mit der Funktion **citGetDriverConstants** (Funktion **0Eh**) abgefragt werden. Der Vorgabewert ist **256**.

YRel_Div = 0..65535

Dieser Eintrag bestimmt den Teiler zur Skalierung relativer Y-Koordinaten. Der Vorgabewert ist **1**.

YRel_Mul = -32768..+32767

Dieser Eintrag bestimmt den Multiplikator zur Skalierung relativer Y-Koordinaten. Der Vorgabewert ist **-1**.

6.1.3 [Commands]

In diesem Abschnitt sind die Parameter zur Maustastemulation enthalten. Eine Beschreibung der Maustastemulation findet sich in Kapitel "Benutzerdefinierte Tastemulation" auf Seite 12. Die Zahlenwerte für die Zustandsübergangsbedingungen besitzen die folgende Bedeutung:

Wert	Bedeutung
0	Never
1	Immediately
2	Enter
3	Z-Press
4	Dual Touch
5	Tap
6	Leave
7	Z-Release
8	No Dual Touch

Im Abschnitt **[Commands]** sind die folgenden Einträge möglich:

Idle_T1 = 0..8

Dieser Eintrag bestimmt die Bedingung für den Übergang vom Zustand "IDLE" in den Zustand "T1". Der Vorgabewert ist **2**.

ModeChange= 0 | 3 | 4 | 5

Bedingung für den Wechsel zwischen absoluten und relativen Koordinaten. Der Vorgabewert ist **0**.

T1_Trigger = 0..8

Dieser Eintrag bestimmt die Bedingung für den Übergang vom Zustand "T1" in den Zustand "TRIGGER". Der Vorgabewert ist **4**.

T2_UPT3 = 0..8

Dieser Eintrag bestimmt die Bedingung für den Übergang vom Zustand "T2" in den Zustand "T3". Der Vorgabewert ist **0**.

Time1 = 0..65535

Dieser Eintrag bestimmt die Zeitkonstante für den Zustand "T1". Die Angabe entspricht der gewünschten Zeit in Millisekunden. Der Vorgabewert ist **0**.

Time2 = 0..65535

Dieser Eintrag bestimmt die Zeitkonstante für den Zustand "T2". Die Angabe entspricht der gewünschten Zeit in Millisekunden. Der Vorgabewert ist **0**.

Time3 = 0..65535

Dieser Eintrag bestimmt die Zeitkonstante für den Zustand "T3". Die Angabe entspricht der gewünschten Zeit in Millisekunden. Der Vorgabewert ist **0**.

Trigger_T2 = 0..8

Dieser Eintrag bestimmt die Bedingung für den Übergang vom Zustand "TRIGGER" in den Zustand "T2". Der Vorgabewert ist **6**.

UPT3_Idle = 0..8

Dieser Eintrag bestimmt die Bedingung für den Übergang vom Zustand "T3" in den Zustand "IDLE". Der Vorgabewert ist **0**.

UPT3_Trigger = 0..8

Dieser Eintrag bestimmt die Bedingung für den Übergang vom Zustand "T3" in den Zustand "TRIGGER". Der Vorgabewert ist **0**.

6.1.4 [Hardware]

In diesem Abschnitt werden die Parameter der seriellen Schnittstelle eingetragen. Im Abschnitt **[Hardware]** sind die folgenden Einträge möglich:

BaudDivisor = 0..65535

Dieser Eintrag bestimmt den Teiler für den Baudratengenerator. Die erzielte Baudrate errechnet sich nach der Formel

$$\text{BaudRate} = \frac{f_{\text{Quarz}}}{16 \cdot \text{BaudDivisor}} ; \quad \text{üblicherweise ist } f_{\text{Quarz}} = 1.8432 \text{ MHz}$$

Der Vorgabewert für BaudDivisor ist **6**, das entspricht 19200 Baud bei $f_{\text{Quarz}} = 1.8432 \text{ MHz}$.

Interrupt = 0..15

Dieser Eintrag bestimmt die Nummer des ISA-Bus Interruptkanals der seriellen Schnittstelle. Der Vorgabewert ist **4**.

IO_Base = 0..65535

Dieser Eintrag bestimmt die Basisadresse des seriellen Schnittstellenbausteins. Der Vorgabewert ist **0x3f8**.

IRT_Mode = 0 | 1 | 2

Dieser Eintrag enthält das zuletzt vom Treiber detektierte Kommunikationsprotokoll des IRT. Dadurch kann der IRT schneller angekoppelt werden. Fehlt dieser Eintrag, oder enthält er eine falsche Angabe, versucht der CiTouchD-Treiber das vorhandene Kommunikationsprotokoll automatisch zu erkennen. Es wird die folgende Zuordnung zwischen dem erkannten Protokoll und dem hier angegebenen Zahlenwert verwendet:

Wert	Bedeutung
0	kein Protokoll erkannt
1	Mode-C Protokoll
2	CTS1 Protokoll

6.1.5 [Settings]

In diesem Abschnitt sind alle veränderbaren Betriebsparameter des Treibers und des IRT eingetragen. Im Abschnitt **[Settings]** sind die folgenden Einträge möglich:

AbsoluteMouse = *Ja* | *Nein*

Dieser Eintrag legt fest ob nach der Initialisierung des Treibers mit absoluten oder relativen Koordinaten gearbeitet werden soll. Ist im Abschnitt **[Commands]** unter dem Eintrag "ModeChange" ein anderer Wert als 0 angegeben, kann der Koordinatenmodus auch dynamisch im laufenden Betrieb gewechselt werden. Der Vorgabewert ist **Ja**.

AutoReinit = *Ja* | *Nein*

Dieser Eintrag legt fest ob nach einer Unterbrechung der Verbindung zwischen IRT und Rechner automatisch neu angekoppelt werden soll. Der Vorgabewert ist **Ja**.

BeamTimeout = 0..65535

Dieser Eintrag bestimmt die Ausblendzeit für defekte Lichtschranken. Ist eine Lichtschranke des IRT länger als die hier eingestellte Zeit in Sekunden unterbrochen, so wird sie von der Koordinatenberechnung ausgeschlossen. Ein Wert von 0 verhindert das Ausblenden der Lichtschranken. Der Vorgabewert ist **20**.

Button = 1..3

Dieser Eintrag bestimmt die zu emulierende Maustaste. Dabei bedeutet 1 die linke Taste, 2 die rechte Taste und 3 beide Taste gleichzeitig. der Vorgabewert ist **1**.

ButtonClick = *Ja* | *Nein*

Dieser Eintrag legt fest ob bei einem Maustastendruck eine akustische Rückmeldung über den PC-Lautsprecher erfolgen soll. Der Vorgabewert ist **Ja**.

ContTime = 0..65535

Dieser Eintrag bestimmt den Abstand zwischen zwei Koordinatenmeldungen des IRT. Der zeitliche Abstand errechnet sich aus dem hier angegebenen Wert in Millisekunden. Dabei ist die zur Übertragung einer Koordinatenmeldung benötigte Zeit (abhängig von der Baudrate) zu beachten. Der Vorgabewert ist **22**.

CoordEnterZ = *Ja* | *Nein*

Dieser Eintrag legt fest ob zur Meldung der ersten Cursorposition nach dem Eintauchen zusätzlich die Überschreitung des eingestellten Drucks auf die Frontscheibe nötig ist. Dazu muß der IRT allerdings mit einer Z-Achse ausgerüstet sein. Der Vorgabewert ist **Nein**.

CoordinateSkip = 0..65535

Dieser Eintrag bestimmt die Anzahl von Koordinatenmeldungen des IRT die nach den Eintauchen ignoriert werden sollen. Der Vorgabewert ist **1**.

CoordSignalZ = Ja | Nein

Dieser Eintrag legt fest ob zur Meldung weiterer Cursorpositionen nach dem Eintauchen zusätzlich die Überschreitung des eingestellten Drucks auf die Frontscheibe nötig ist. Dazu muß der IRT allerdings mit einer Z-Achse ausgerüstet sein. Der Vorgabewert ist **Nein**.

DbErrSkip = 0..65535

Dieser Eintrag bestimmt die Anzahl von Zweifachberührungsmeldungen des IRT die ignoriert werden sollen bevor eine Zweifachberührung erkannt wird. Der Vorgabewert ist **1**.

DimmingHigh = 0..255

Dieser Eintrag bestimmt das Puls-/Pausenverhältnis des IRT PWM-Ausgangs bei inaktivem Touch-Saver. Der Vorgabewert ist **0**.

DimmingLow = 0..255

Dieser Eintrag bestimmt das Puls-/Pausenverhältnis des IRT PWM-Ausgangs bei aktivem Touch-Saver. Der Vorgabewert ist **0**.

MinXBeams = 1..5

Dieser Eintrag bestimmt die Anzahl nebeneinanderliegender X-Lichtschranken des IRT die gleichzeitig unterbrochen werden müssen um als Berührung erkannt zu werden. Der Vorgabewert ist **1**.

MinYBeams = 1..5

Dieser Eintrag bestimmt die Anzahl nebeneinanderliegender Y-Lichtschranken des IRT die gleichzeitig unterbrochen werden müssen um als Berührung erkannt zu werden. Der Vorgabewert ist **1**.

Pressure = 0..255

Dieser Eintrag bestimmt den Druck auf die Frontscheibe der nötig ist um die druckgesteuerten Ereignisse des Treibers auszulösen. Ein Wert von 0 schaltet die Z-Achse des IRT aus. Der Vorgabewert ist **20**.

SaverScan = 1..65535

Dieser Eintrag legt die Abtastrate des IRT bei aktivem Touch-Saver fest. Die Abtastrate entspricht dem hier angegebenen Wert in Millisekunden. Der Vorgabewert ist **500**.

SaverTime = 0..65535

Dieser Eintrag bestimmt die Zeit bis zur Aktivierung des Touch-Savers. Die Aktivierungszeit entspricht dem hier eingetragenen Wert in Sekunden. Bei einem Wert von 0 wird der Touch-Saver sofort aktiviert. Bei einem Wert von 65535 wird der Touch-Saver niemals aktiviert. Der Vorgabewert ist **65535**.

SmoothAlways = Ja | Nein

Dieser Eintrag legt fest, ob die Mittelwertbildung für absolute Koordinaten auch über das Verlassen und erneute Unterbrechen des Touchbereichs hinweg erfolgen soll. Der Vorgabewert ist **Nein**.

TapTime = 0..65535

Dieser Eintrag bestimmt die Zeit innerhalb derer der Touchbereich erneut unterbrochen werden muß um ein Tap-Ereignis auszulösen. Die Zeit entspricht dem hier angegebenen Wert in Millisekunden. Der Vorgabewert ist **500**.

X_Offset = -32768..+32767

Dieser Eintrag bestimmt den X-Abstand zwischen dem Berührungspunkt und der tatsächlichen Cursorposition. Positive Werte verschieben die Cursorposition nach rechts, negative Werte nach links. Der Vorgabewert ist **0**.

X_Smoothing = 0..dcSmoothMax-1

Dieser Eintrag bestimmt die Anzahl an Koordinatenmeldungen über die zur Glättung absoluter X-Koordinaten gemittelt wird. Die obere Grenze kann mit der Funktion **citGetDriverConstants** abgefragt werden. Der Vorgabewert ist **10**.

Y_Offset = -32768..+32767

Dieser Eintrag bestimmt den Y-Abstand zwischen dem Berührungspunkt und der tatsächlichen Cursorposition. Positive Werte verschieben die Cursorposition nach unten, negative Werte nach oben. Der Vorgabewert ist **0**.

Y_Smoothing = 0..dcSmoothMax-1

Dieser Eintrag bestimmt die Anzahl an Koordinatenmeldungen über die zur Glättung absoluter Y-Koordinaten gemittelt wird. Die obere Grenze kann mit der Funktion **citGetDriverConstants** angefragt werden. Der Vorgabewert ist **10**.

6.1.6 [Sound]

In diesem Abschnitt sind die Parameter zum Erzeugen des Maustastenklicks eingetragen. Im Abschnitt **[Sound]** sind die folgenden Einträge möglich:

ButtonClick = 0 | 1 | 2 | 3

Dieser Eintrag legt fest, wann ein Ton erzeugt werden soll. Dabei kann eine Kombination der folgenden Werte verwendet werden:

Wert	Bedeutung
0	keine Tonerzeugung
1	Tonerzeugung, wenn die Maustaste gedrückt wird
2	Tonerzeugung, wenn die Maustaste losgelassen wird
3	Tonerzeugung, sowohl beim Drücken, als auch beim Losgelassen der Maustaste.

Der Vorgabewert ist **1**.

FreqDown = 0..65535

Dieser Eintrag legt die Höhe des Tons, der beim Drücken der Maustaste erzeugt wird, in 1 Hz Schritten fest. Der Vorgabewert ist **783**.

FreqUp = 0..65535

Dieser Eintrag legt die Höhe des Tons, der beim Loslassen der Maustaste erzeugt wird, in 1 Hz Schritten fest. Der Vorgabewert ist **1046**.

TimeDown = 0..65535

Dieser Eintrag legt die Dauer des Tons, der beim Drücken der Maustaste erzeugt wird, in Millisekunderschritten fest. Der Vorgabewert ist **30**.

TimeUp = 0..65535

Dieser Eintrag legt die Dauer des Tons, der beim Loslassen der Maustaste erzeugt wird, in Millisekunderschritten fest. Der Vorgabewert ist **30**.

6.1.7 [Citouchd]

In diesem Abschnitt sind die Parameter für das Verhalten von CiTouchD als Maustreiber eingetragen. Im Abschnitt **[Citouchd]** sind die folgende Einträge möglich.

Stack = 0..64

Dieser Eintrag legt fest, wieviel Stack CiTouchD für sich reserviert. Der Wert entspricht der Stackgröße in 256-Byte-Blöcken. Im Falle von Stack=0 arbeitet CiTouchD auf dem Stack des gerade laufenden Programms. Der Vorgabewert ist **6**.

HardReset = 0 | 1 | 2.

Dieser Eintrag legt fest, ob in den Maus-API-Funktionen (Software-Interrupt 33h) 00h (int33ResetAndStatus) und 2Fh (int33HardReset) ein Reset des Touches durchgeführt werden soll. Da dieser Vorgang bis zu 10 Sekunden dauern kann (im Falle eines Fehlschlagens), läßt sich der Reset abschalten. Der Vorgabewert ist **0**.

Wert	Bedeutung
0	Weder in Funktion 00h noch in Funktion 2Fh wird ein Reset des Touches ausgeführt.
1	Nur in Funktion int33HardReset wird ein Reset des Touches ausgeführt
2	Sowohl in Funktion 00h als auch in Funktion 2Fh wird ein Reset des Touches ausgeführt

7 INDEX

A

absolute Koordinaten	14
AbsoluteMouse	46; 49
ACCELERATION	37; 42
AutoReinit	46

B

Backlight Dimming	16
BaudDivisor	45
BeamTimeout	46
Button	46
ButtonClick	46
Button-Machine	12

C

Calibration	43
CALIBRATIONABS	38
CALIBRATIONREL	38
CITOUCHW.INI	42
Citron Infrarot-Touch	4
COMMANDS	38; 39; 44
ContTime	46
CoordEnterZ	46
CoordinateSkip	47
COORDMODE	40
CoordSignalZ	47
Cursor Control	15
Cursorbeschleunigung	15

D

DbErrSkip	47
DIMMING	40
DRIVERCONSTANTS	40
DRIVERSETTINGS	40
Dual / Exit	9
Dual Touch	9; 12
Dual Touch Skip Count	9

E

Enter	6; 12; 14
Exit	7

G

Glättung	13
----------------	----

H

Hardware	45
----------------	----

I

Idle_T1	44
Immediately	12
Interrupt	45
IO_Base	45
IRT	4

K

Koordinatenmodus	14
------------------------	----

L

Leave	12
--------------------	----

M

MinXBeams	47
MinYBeams	47
ModeChange	44

N

Never	12
No Dual Touch	12
<i>Notification</i>	25

P

<i>Polling</i>	25
Pressure	47
Pressure Sensivity	10

S

SaverScan	47
SaverTime	48
SERIALHARDWARE	41
Settings	46; 48
SmoothAlways	48
Steuerung der Hintergrundbeleuchtung	16

T

T1_Trigger	44
T2_UPT3	44
Tap	8; 12
TapTime	48
Tap-Time	8
Time	8
Time / Time	10
Time to Click	8; 10
Time to Idle	8
Time to Second Click	10
Time1	45
Time2	45
Time3	45
TOUCHHARDWARE	41
Touch-Saver	16
TOUCHSETTINGS	42
Trigger_T2	45

U

UPT3_Idle	45
UPT3_Trigger	45

X

X_Add	43
X_Border	43
X_Mul	43
X_Offset	48
X_Smoothing	48
XRel_Div	43
XRel_Mul	43

Y

Y_Add	43
Y_Border	43
Y_Mul	43; 44
Y_Offset	48
Y_Smoothing	48
YRel_Div	44
YRel_Mul	44

Z

Z-Press	10; 12; 15
Z-Press / Enter	15
Z-Release	12