

isdnphoneDSP

Telefonmodul für die DSP-ISDN-Karten

- Billion BIPAC PCI Pro
- Eicon DIVA ISDN+CT
- Telekom T-Concept PC und T-View 100

Stand: 28.03.2003
Version 2.1.4.0

erstellt von: Jochen Tillmanns
erreichbar unter: Tel.: +49 (931) 88075952, Fax: +49 (931) 88075953, Email: info@isdnphone.de
Stand: 28.03.2003
© 2003 Jochen Tillmanns

Inhaltsverzeichnis

1	Kontakt	3
2	Lizenzinformationen	3
3	Zielsetzung	4
4	Übersicht	4
5	Schnittstelle	5
5.1	Funktion: ISDNCreate	5
5.2	Funktion: ISDNFree	5
5.3	Funktion: ISDNLicense	5
5.4	Funktion: ISDNInit	5
5.5	Funktion: ISDNDone	5
5.6	Funktion: ISDNMakeCall	6
5.7	Funktion: ISDNAcceptCall	6
5.8	Funktion: ISDNHangUp	6
5.9	Funktion: ISDNSendDTMF	6
5.10	Funktion: ISDNSendWaveFile (nur Billion BIPAC PCI Pro)	6
5.11	Funktion: ISDNSetVolume	6
5.12	Funktion: ISDNSetVolumeEx	7
5.13	Funktion: ISDNExplicitCallTransferTo	7
5.14	Funktion: ISDNGetLastError	8
5.15	Funktion: ISDNGetErrorText	8
5.16	Funktion: Callback TTraceProc	8
5.17	Funktion: Callback TCallStateProc	8
5.18	Funktion: Callback TQueryTapeProc	9
5.19	Funktion: Callback TDTMFProc	9
5.20	Funktion: ISDNSetDTMFParams	10
6	Programmablauf	10
6.1	Initialisierung / Deinitialisierung	10
6.2	Lizenz	10
6.3	Wann darf welche Funktion aufgerufen werden?	10
6.4	Callback-Funktionen	11
6.5	Kontinuierliche Ausgabe von Log-Informationen	11
6.6	Anruf tätigen	11
6.7	Anrufe annehmen	12
6.8	Anruf abbrechen	12
6.9	Anklopffunktion	12
6.10	Gesprächsaufzeichnung	12
6.11	Rufweitervermittlung	12
6.12	Dual-Tone Multi Frequency (DTMF) versenden und empfangen	12
6.13	Wave-Dateien senden (nur für Billion BIPAC PCI pro Karten!)	13
6.14	Hörerlautstärke	13
6.15	Deinitialisierung	13
7	Systemvoraussetzungen	13
7.1	Betriebssystem Microsoft Windows 98 / ME / NT 4 / 2000	13
7.2	ISDN-Karte	13
7.3	Hinweis zu den ISDN-Treibern der T-Concept Karte	14
7.4	Telefonhörer	14
7.5	Unzulässige Systemkomponenten	14
7.6	ISDN-Anschluß (S ₀ -Bus)	14
8	Technische Informationen	14
8.1	Das Telefonmodul wird in kompilierter Form zur Verfügung gestellt	14
8.2	Wichtige Information zur T-Concept PC ISDN-Karte	14
8.3	Wichtige Information zur Billion BIPAC-PCI Pro ISDN-Karte	15
8.4	Wichtige Information zur Eicon DIVA ISDN+CT ISDN-Karte	15
8.5	Folgende ISDN-Fehlermeldungen können während dem Betrieb des Telefonmoduls auftreten	15

1 Kontakt

Informationen erhalten sie unter folgenden Adressen:

- ISDNphone Homepage: www.isdnphone.de
- Email: info@isdnphone.de
- Postalisch:
Jochen Tillmanns
ISDNphone
Jakob-Riedinger-Str. 4
97074 Würzburg
- Tel.: 0931/88075952 Fax: 0931/88075953

2 Lizenzinformationen

Das ISDNphone DSP Telefonmodul erlaubt die Verwendung von DSP-ISDN-Karten zur Telefonie in professioneller Sprachqualität. Es basiert dabei auf der Technologie von ISDNphone 2.0, erhältlich auf <http://www.isdnphone.de>.

Das ISDNphone DSP Telefonmodul wird als Windows Dynamic Link Library (DLL) geliefert. Das Download-Archiv enthält die DLL, diese Dokumentation und ein Beispielpogramm.

Bestellungen richten Sie bitte an einer der in Abschnitt 1 genannten Adressen. Auf Wunsch können auch individuelle Telefonmodul-Versionen mit angepaßtem Funktionsumfang erstellt werden.

a) **Unlizenzierte Version**

Preis: kostenlos

Die gelieferte DLL kann ohne gültigen Lizenzschlüssel in eingeschränkter Form Verwendet werden:

- Keine kommerzielle Verwendung
- Nur zu Test- und Entwicklungszwecken gedacht
- Verbindungen sind auf max. 90 Sekunden limitiert

b) **Demo-Lizenz**

Preis: kostenlos, wird auf Anfrage ausgestellt

Auf Anfrage kann eine Demo-Lizenz ausgestellt werden. Diese Lizenz unterliegt folgenden Einschränkungen:

- Keine kommerzielle Verwendung
- Nur zu Test- und Entwicklungszwecken gedacht

c) **Einzelplatz-Lizenz**

Preis: 79 Euro (inkl. 16% Mwst.)

Nur diese Lizenz ist völlig frei von Einschränkungen und darf zeitlich unbegrenzt in kommerziellem Rahmen eingesetzt werden.

Die Lizenzdaten (Name und Schlüssel) werden der Funktion ISDNLicense übergeben, um die DLL zu lizenzieren.

3 Zielsetzung

Diese Dokumentation beschreibt die Schnittstelle des ISDNPHONE DSP-Telefonmoduls, damit es von externen Programmen ausgeführt werden kann. Das so angepaßte Programm wird auf Systemen mit folgender Ausstattung laufen:

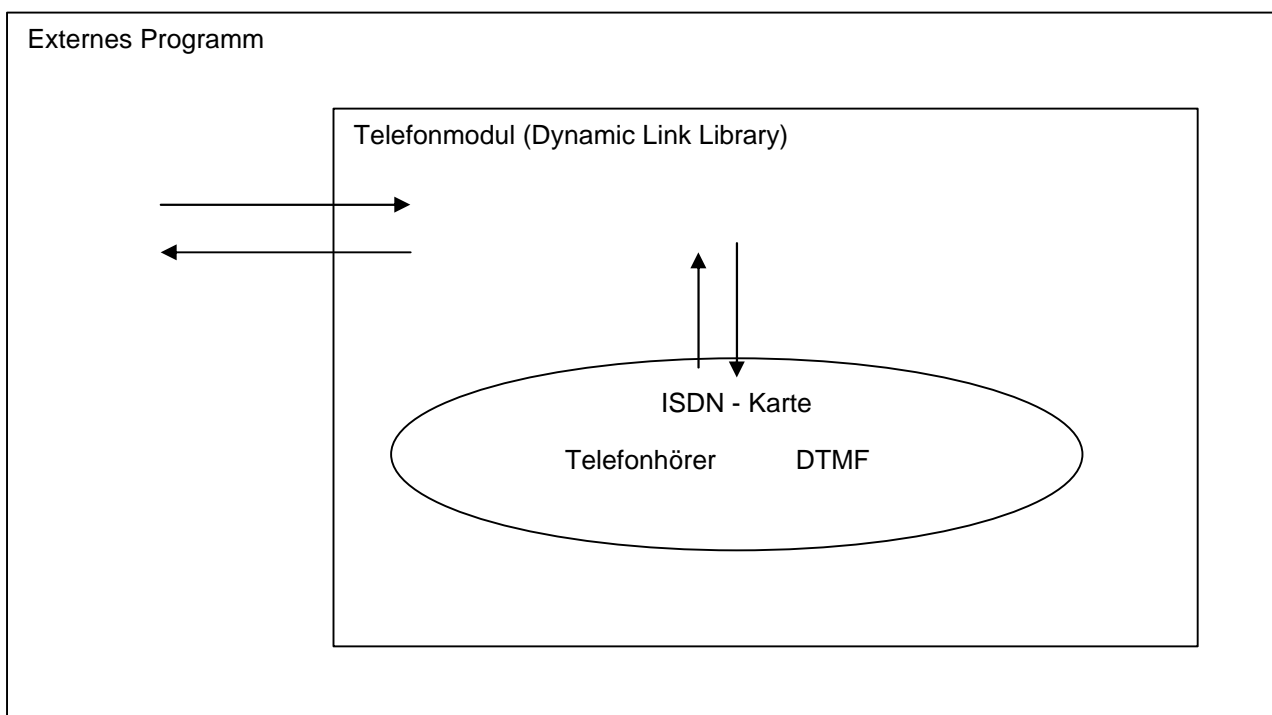
- Microsoft Windows 98, ME, NT 4 und 2000

und

- ISDN-Karte T-Concept ISDN PC (s. Abschnitt 8.2)
oder
- ISDN-Karte BIPAC PCI Pro (s. Abschnitt 8.3)
oder
- ISDN-Karte Eicon DIVA ISDN+CT (s. Abschnitt 8.4)

Für Informationen über den Bezug der genannten ISDN-Karten besuchen Sie bitte die ISDNphone Homepage (siehe Abschnitt 1).

4 Übersicht



5 Schnittstelle

Die Schnittstelle wird als Windows-Dynamic Link Library zur Verfügung gestellt. Die DLL wird mit isdnphoneDSP.DLL bezeichnet. Dieser Name darf nicht geändert werden, da es sonst zu Fehlfunktionen bei Verwendung von String-Ressourcen in der DLL kommt.

Die Library beinhaltet Funktionen, die mit der STDCALL-Aufrufkonvention deklariert sind: (Die Typen und Konstanten werden in der mitgelieferten Unit deklariert)

5.1 Funktion: ISDNCreate

```
function ISDNCreate:Pointer;
```

Parameter	Typ	Bemerkung
Rückgabewert	Pointer	Die Funktion liefert einen internen Zeiger zurück, der in den anderen Funktionen als DataHandle übergeben wird.

5.2 Funktion: ISDNFree

```
procedure ISDNFree(DataHandle:Pointer);
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle

5.3 Funktion: ISDNLicense

```
function ISDNLicense(DataHandle:Pointer; RegName, RegKey:
PChar):LongBool;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
RegName	PChar	0-100 Zeichen	Registrierungsname (siehe Lizenzdaten)
RegKey	PChar	0-100 Zeichen	Registrierungsschlüssel (siehe Lizenzdaten)
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.4 Funktion: ISDNInit

```
function ISDNInit(DataHandle:Pointer; InitParam: PInitParam):LongBool;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
InitParam	Pointer	PInitParam	Pointer auf Struktur mit erweiterten Parametern
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

TInitParam	Typ	Format/Werte	Bemerkung
CallStateProc	Pointer	Callback (s.u.)	Gibt Information über den aktuellen Verbindungsstatus
DTMFProc	Pointer	Callback (s.u.)	Gibt Informationen über DTMF
TraceProc	Pointer	Callback (s.u.)	optional, kann auch mit NULL bzw. nil aufgerufen werden
QueryTapeProc	Pointer	Callback (s.u.)	optional, kann auch mit NULL bzw. nil aufgerufen werden
AcceptMSN	Pchar	0-9	Gibt an, auf welche MSN reagiert werden soll (ALERT) NULL oder Leerstring: Alle MSN akzeptieren Sonst: Nur auf die angegebene MSN reagieren <i>Debug-Hinweis: Bei eingehendem Anruf wird im Log ein Hinweis auf das Ergebnis der Prüfung ausgegeben.</i>

5.5 Funktion: ISDNDone

```
function ISDNDone(DataHandle:Pointer):LongBool;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.6 Funktion: ISDNMakeCall

```
function ISDNMakeCall(DataHandle:Pointer;
szMSN,szNumber:PChar):LongBool;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
szMSN	PChar	0-25 Ziffern, 0-terminiert	Die an den Angerufenen übermittelte Anschlußnummer
szNumber	PChar	1-25 Ziffern, 0-terminiert	Die Rufnummer des Teilnehmers
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.7 Funktion: ISDNAcceptCall

```
function ISDNAcceptCall(DataHandle:Pointer):LongBool;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	Intern	Das von ISDNCreate zurückgegebene DataHandle
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.8 Funktion: ISDNHangUp

```
function ISDNHangUp(DataHandle:Pointer):LongBool; stdcall;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.9 Funktion: ISDNSendDTMF

```
function ISDNSendDTMF(DataHandle:Pointer; szDTMF:PChar):LongBool;
stdcall;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
szDTMF	Pchar	1-20 Zeichen, 0-terminiert	Die zu sendende DTMF Nummer (Eicon DIVA + T-Concept: 0-9, *, #, A, B, C, D) (BIPAC PCI Pro: 0-9, *, #, nur eine Ziffer!)
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.10 Funktion: ISDNSendWaveFile (nur Billion BIPAC PCI Pro)

```
function ISDNSendWaveFile(DataHandle:Pointer;
szFilename:PChar):LongBool; stdcall;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
szFilename	PChar	Dateipfad, 0-terminiert	Die Datei muß im Wave-Format vorliegen: Format PCM, 8000Hz, 16 bit, Mono
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.11 Funktion: ISDNSetVolume

Version 2.1: Für Eicon DIVA CT ISDNSetVolumeEx verwenden.

```
function ISDNSetVolume(DataHandle:Pointer; VolumeInfo:
PVolumeInfo):LongBool; stdcall;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	Intern	Das von ISDNCreate zurückgegebene DataHandle
VolumeInfo	PVolumeInfo	1-20 Zeichen, 0-terminiert	Enthält alle Lautstärke-Einstellungen
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

TVolumeInfo	Typ	Format/Werte	Bemerkung
AudioFrontEnd	Byte	0, 1	Auswahl des Sprechgerätes (Handset oder Headset) (nur für T-Concept)
Lautsprecher	Byte	0-7	Lautstärke des Hand/Headsets (0=Leise, 7=Laut)
Microphon	Byte	0-7	Lautstärke des Mikrophones (0=Leise, 7=Laut) <i>nur BIPAC PCI Pro und Eicon DIVA ISDN+CT</i>

5.12 Funktion: ISDNSetVolumeEx

Version 2.1: Erweiterte Spezial-Einstellmöglichkeiten für DSP-ISDN-Karten.

```
function ISDNSetVolumeEx(DataHandle:Pointer; Command: Word; Data:
Pointer):LongBool; stdcall;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
Command	Word	Konstante	Befehl zur Einstellung der DSP-ISDN-Karte. Definierte Konstanten sind: VOLUMEEX_SETDIVACTMODE_HANDSET=1 VOLUMEEX_SETDIVACTMODE_SPEAKERPHONE=2 VOLUMEEX_SETDIVACTVOLUME=3
Data	Pointer	Abhängig von <i>Command</i>	Bei Command=3: <i>PDivaCTVolumeInfo</i> (s.u.)
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

Command-Konstanten:

Command	Wert	Funktion
VOLUMEEX_SETDIVACTMODE_HANDSET	1	Schaltet die DIVA Karte in den Hand/Headset Modus (die RJ-Stecker am Karten-Slotblech sind aktiv). <i>Data</i> wird nicht verwendet
VOLUMEEX_SETDIVACTMODE_SPEAKERPHONE	2	Schaltet die DIVA Karte in den Speakerphone-Modus (die beiden Klinkenstecker am Karten-Slotblech sind aktiv). <i>Data</i> wird nicht verwendet
VOLUMEEX_SETDIVACTVOLUME	3	Stellt die ausgiebigen Lautstärkeparameter der DIVA ein <i>Data</i> ist vom Typ <i>PDivaCTVolumeInfo</i>

Strukturen:

TDivaCTVolumeInfo	Typ	Format/Werte	Bemerkung
Handset_MicVol	Byte	0-8	Lautstärke des Hand/Headset-Mikrophons am RJ-Stecker
Handset_SpkVol	Byte	0-5	Lautstärke des Hand/Headset-Lautsprechers am RJ-Stecker
Speakerphone_MicVol	Byte	0-8	Lautstärke des Mikrophons am MIC-Klinkenstecker
Speakerphone_SpkVol	Byte	0.-12	Lautstärke des Lautsprechers am Speaker-Klinkenstecker
HandsetandSpeakerphone_SpkGain	Byte	-33 bis +18	Zusätzliche digitale Verstärkung des Lautsprechers (Hand/Headset und Klinkenstecker)
HandsetandSpeakerphone_MicGain	Byte	-33 bis +18	Zusätzliche digitale Verstärkung des Mikrophons (Hand/Headset und Klinkenstecker)
reserved1	Word	0	Reserviert für zukünftige Erweiterungen
reserved2	Word	0	Reserviert für zukünftige Erweiterungen

5.13 Funktion: ISDNExplicitCallTransferTo

```
function ISDNExplicitCallTransferTo (DataHandle: Pointer;
szMSN,szNumber:PChar):LongBool; stdcall;
```

Initiiert die Anrufweitervermittlung an die angegebene Nummer.

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
szMSN	PChar	0-25 Ziffern, 0-terminiert	Die an den Angerufenen übermittelte Anschlußnummer
szNumber	PChar	1-25 Ziffern, 0-terminiert	Die Rufnummer des Teilnehmers
Rückgabewert	LongBool	False/True	0 bei Fehler, <> 0 wenn Erfolgreich

5.14 Funktion: ISDNGetLastError

```
function ISDNGetLastError(DataHandle:Pointer):Cardinal; stdcall;
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
Rückgabewert	Cardinal		Letzter Fehlerwert (CAPIERR_xxx)

5.15 Funktion: ISDNGetErrorText

```
procedure ISDNGetErrorText(DataHandle:Pointer; ErrorValue:Cardinal;
szTextBuf:PChar);
```

Parameter	Typ	Format/Werte	Bemerkung
DataHandle	Pointer	intern	Das von ISDNCreate zurückgegebene DataHandle
ErrorValue	Cardinal	Fehlerwerte	Von ISDNGetLastError zurückgelieferte Werte
szTextBuf	PChar	Mindest-Länge 2000 Byte	Rückgabe-Buffer für Fehlertext, 0-terminiert

5.16 Funktion: Callback TTraceProc

```
TTraceProc = procedure ( TraceType: integer; szText: PChar );
```

Parameter	Typ	Format/Werte	Bemerkung
TraceType	Integer	0 oder 1	Definition in der Deklarations-Unit
szText	Pchar	Info- oder Fehlertext	Enthält den Text (0-terminiert)

5.17 Funktion: Callback TCallStateProc

```
TCallStateProc = procedure (LineID: Byte; State: TCallState;
lpCallData: Pointer);
```

Parameter	Typ	Bemerkung
LineID	Byte	Eindeutige ID für die betreffende Leitung Folgende ID's sind definiert: LINEID_TELEPHONY = 1 (Telefonie-Leitung) LINEID_CALLWAITING = 2 (Leitung zur Anklopf-Anzeige) LINEID_ECTDESTINATION = 3 (Leitung für Weitervermittlung)
State	TCallState	Der aktuelle Status der Verbindung
lpCallData	Pointer	Nähere Informationen zur Verbindung, abhängig vom Status

```
TCallState = (csBereit, csAnruf, csVerbinde, csVerbunden,
csAufgelegt, csOnHold);
```

Inhalt von lpCallData abhängig von CallState:

CallState	Inhalt von lpCallData	Bemerkung
CsBereit	Nil	Es kann ein Anruf angenommen oder eine Nummer gewählt werden
CsAnruf	PAnrufInfo	Signalisierung eines Anrufes. Der Parameter enthält zusätzliche Informationen.
CsVerbinde	Nil	Eine ausgehende Verbindung wird aufgebaut
CsVerbunden	PConnectInfo	Der Anruf ist zustandegekommen. Der Parameter enthält zusätzliche Informationen.
CsAufgelegt	PDisconnectInfo	Ende einer Verbindung. Der Parameter enthält zusätzliche Informationen.
CsOnHold	Nil	Die Verbindung wird gehalten (nötig für Vermittlung)

TAnrufInfo	Typ	Bemerkung
MSN	PChar	Die angerufene MSN
AnruferNummer	PChar	Nummer des Anrufers, falls übertragen
TConnectInfo	Typ	Bemerkung (siehe auch Abschnitt "Programmablauf")
VerbindungsModus	Byte	0 = Keine weiteren Informationen (z.B. bei Nicht-Billion ISDN-Karten) Weitere Modi nur für Billion BIPAC PCI pro: 1 = Hörer AN, Datenübertragung AUS (Standard nach Verbindungsaufbau, Der Parameter „Fehler“ hat keine Bedeutung) 2 = Hörer AUS, Datenübertragung AN (Dateiübertragung erfolgreich initiiert? - > siehe „Fehler“)
Fehler	ByteBool	Nur Bei Verbindungsmodus=2 von Bedeutung
TDisconnectInfo	Typ	Bemerkung
Verbunden	ByteBool	Ist eine Verbindung zustande gekommen?
AuflegeRichtung	Byte	Welche Seite hat die Verbindung beendet? 0 = Aktive Beendigung der Verbindung mit ISDNHangUp 1 = Passive Beendigung der Verbindung durch das Netzwerk (dann ist Verbunden=FALSE) oder den anderen Teilnehmer (Verbunden=TRUE).
Fehler	ByteBool	TRUE, wenn die Verbindung aufgrund eines Fehlers getrennt wurde. Wenn FALSE, wurde die Verbindung normal getrennt. Der genaue Fehler-Code kann immer in <i>FehlerCode</i> nachgesehen werden
FehlerCode	Word	Bedeutung dieses Codes siehe Abs. 11.3

5.18 Funktion: Callback TQueryTapeProc

```
TQueryTapeProc = procedure ( lpQueryTapeData: Pointer );
```

Parameter	Typ	Bemerkung
lpQueryTapeData	Pointer	In dieser Struktur müssen die Rückgabewerte gespeichert werden. Falls die Verbindung aufgezeichnet werden soll, muß Enabled auf TRUE gesetzt werden.
TQueryTapeData	Typ	Bemerkung
Enable	ByteBool	Soll die aktuelle Verbindung aufgezeichnet werden?
MaxRecordTime	DWORD	Wie lange soll maximal aufgezeichnet werden (Angabe in Sekunden). 0 bedeutet keine maximale Aufzeichnungsdauer.
Filename	Char[260]	Datei, in die die Daten gespeichert werden sollen. Da die Daten im WAVE-format gespeichert werden, sollte die Dateierweiterung „.WAV“ verwendet werden.

5.19 Funktion: Callback TDTMFProc

```
TDTMFProc = procedure (DTMFType: TDTMFType; lpDTMFData: Pointer);
```

Parameter	Typ	Bemerkung
DTMFType	TDTMFType	Ist die Ursache der Meldung das Senden oder Empfangen von DTMF
lpDTMFData	Pointer	Nähere Informationen, abhängig vom DTMFType

```
TDTMFType = (dtmfSendResult, dtmfDetectInit, dtmfDetected);
```

Inhalt von lpDTMFData abhängig von DTMFType:

DTMFType	Inhalt von lpDTMFData	Bemerkung
DtmfSendResult	PLongInt	Mögliche Werte: 0 = DTMF erfolgreich versendet 1 = Ungültiges DTMF-Zeichen 2 = Unbekannter DTMF Befehl (sehr unwahrscheinlicher Fehler)
dtmfDetectInit	PLongInt	Mögliche Werte wie bei dtmfSendResult
dtmfDetected	PChar	Es wurden 1 oder mehrere Zeichen erkannt, der Parameter lpDTMFData enthält einen Pointer auf diese Zeichen

5.20 Funktion: ISDNSetDTMFParams

```
procedure ISDNSetDTMFParams(DataHandle:Pointer;  
DTMFDurations:PDTMFDurations);
```

Parameter	Typ	Bemerkung
DTMFDurations	PDTMFDurations	Ton- und Pausenlänge in Millisekunden beim Senden und Empfangen von DTMF (getrennt einstellbar). Sinnvolle Werte: 20-500 msek, Standard ist für alle Werte 70 msek.

TDTMFDurations	Typ	Bemerkung
SendToneDuration	Word	Länge eines Tones beim Senden
SendGapDuration	Word	Länge der Pause zwischen zwei Tönen beim Senden
DetectToneDuration	Word	(Hinweis auf) Länge eines Tones beim Empfangen
DetectGapDuration	Word	(Hinweis auf) Pause zwischen zwei Tönen beim Empfangen

6 Programmablauf

6.1 Initialisierung / Deinitialisierung

Das Telefonmodul wird vom externen Programm geladen und muß zunächst initialisiert werden (Funktionen ISDNCreate und ISDNInit). Danach können mehrere Anrufe getätigt/angenommen werden (ISDNMakeCall, ...AcceptCall, ...HangUp). Am Schluß wird das Modul mit ISDNDone und ISDNFree deinitialisiert.

Sollte während der Initialisierungsphase (ISDNInit) ein Fehler auftreten, wird dies über den Rückgabewert angezeigt (FALSE). Der Fehlerwert kann mit ISDNGetLastError und mit ISDNGetErrorText ein aussagekräftiger Text dazu geholt werden.

Das Telefonmodul erkennt automatisch, ob eine T-Concept PC ISDN, Eicon DIVA ISDN+CT oder eine Billion BIPAC PCI Pro verwendet wird. Voraussetzung dafür ist, daß auch nur diese Karten der jeweiligen Hersteller verwendet werden.

6.2 Lizenz

Falls gültige Lizenzdaten vorliegen (siehe Abschnitt 2), müssen diese noch vor dem Aufruf von ISDNInit an das Telefonmodul übergeben werden. Hierzu ist die Funktion ISDNLICENSE verantwortlich. Die Funktion liefert TRUE zurück, falls gültige Daten übergeben wurden. Ansonsten wird das Telefonmodul im unlizensierten Modus gestartet. Dies bedeutet, daß das Telefonmodul nicht routinemäßig oder kommerzielle Zwecke verwendet werden darf. Zudem ist die maximale Länge eines Gespräches auf 90 Sekunden limitiert.

6.3 Wann darf welche Funktion aufgerufen werden?

Wird eine der folgenden Funktionen im falschen Moment (= falscher CallState) aufgerufen, ist das Ergebnis FALSE und ISDNGetLastError liefert CAPIERR_INVALIDSTATE. Die Tabelle zeigt, wann welche Funktion aufgerufen werden darf:

Funktion ISDN...	CallState von LINEID_TELEPHONY				
	csBereit	csAnruf	csVerbinde	csVerbunden	csOnHold
MakeCall	X				
AcceptCall		X			
HangUp		X	X	X	X
SendDTMF				X	
SendWavefile				X	
ExplicitCallTransferTo				X	

Im CallState csAufgelegt darf keine Funktion aufgerufen werden. Er dient nur dazu, Informationen über den vorangegangenen Anruf zu übermitteln und wechselt sofort danach in den CallState csBereit.

6.4 Callback-Funktionen

Da das Telefonmodul asynchron läuft, gibt es den Verbindungsstatus über die Callback-Funktion TCallStateProc wieder. Der DTMF-Status wird über die Callbackfunktion TDTMFProc wiedergegeben. Die Callback-Funktion TraceProc dient nur der Fehlersuche und Information und wird nicht unbedingt benötigt. Mittels TQueryTapeProc können die Gesprächsdaten aufgezeichnet werden.

Wichtig dabei ist, daß keine ISDNxxx Funktion direkt aus einer Callback-Funktion aufgerufen wird. Dies kann zu einem „Deadlock“ der DLL führen. Die Aufrufe sollten deshalb durch Windows Messages entkoppelt werden.

6.5 Kontinuierliche Ausgabe von Log-Informationen

Die Ausgabe wird durch Übergeben einer gültigen Funktionsadresse in ISDNInit aktiviert. Durch Übergabe eines NULL-Pointers wird die Ausgabe deaktiviert.

Allgemeine Informationen (TraceType = CAPITRACE_INFO)

Es werden Informationen über die verwendete ISDN-Hardware, CAPI 2.0 Version und die gesamte Kommunikation mit der ISDN-Karte ausgegeben.

Nutzen: Es hat sich gezeigt, daß diese Informationen für die Behebung von Problemen sehr wichtig sind. Die Informationen sollten aber nicht an den Endbenutzer weitergegeben werden.

Fehler-Meldungen (TraceType = CAPITRACE_ERROR)

Hier werden alle Fehler des Telefonmoduls im Klartext ausgegeben. Dies schließt fehlerhafte Funktionsaufrufe (z.B. bei falschem CallState) sowie interne Probleme der Nachrichtenbehandlungsroutine ein.

Nutzen: Kann bei der Fehlersuche behilflich sein.

6.6 Anruf tätigen

Nach erfolgreicher Initialisierung des Moduls wird der Anruf eines Teilnehmers initiiert. Die Funktion kehrt sofort wieder zurück. Sollte ein Fehler aufgetreten sein, läßt er sich wie bei ISDNInit anzeigen. Bei erfolgreicher Ausführung wurde der Anruf initiiert. Die Callback-Funktion TCallStateProc wird aufgerufen, sobald entweder a) die Verbindung zustandegekommen ist (csVerbunden) oder b) beendet wurde (durch Fehler oder ISDNHangUp).

Erfolgreiche Verbindung mit ISDNMakeCall:

csBereit – csVerbinde – csVerbunden – csAufgelegt (lpCallData ist TRUE) – csBereit

Nicht Erfolgreiche Verbindung mit ISDNMakeCall:

csBereit – csVerbinde – csAufgelegt (lpCallData ist FALSE) – csBereit

6.7 Anrufe annehmen

Ein ankommender Ruf wird in TCallStateProc mit csAnruf signalisiert. (IpCallData enthält die Nummer des Anrufers). Der Anruf kann nun mit ISDNAcceptCall angenommen oder mit ISDNHangUp abgelehnt werden.

Erfolgreiche Verbindung mit ISDNAcceptCall:

csBereit – csAnruf – csVerbunden – csAufgelegt (IpCallData ist TRUE) – csBereit

Nicht Erfolgreiche Verbindung mit ISDNAcceptCall:

csBereit – csAnruf – csAufgelegt (IpCallData ist FALSE) – csBereit

6.8 Anruf abbrechen

Sollte der Teilnehmer nicht abheben, kann der Anwahlvorgang manuell abgebrochen werden (ISDNHangUp). Auch eine bestehende Verbindung und ein ankommender Anruf kann mit ISDNHangUp abgebrochen bzw. abgewiesen werden

6.9 Anklopffunktion

Sollte das Telefonmodul bei einem ankommenden Ruf bereits in Benutzung sein (die Haupt-Telefonie-Leitung LINEID_TELEPHONY hat einen Status ungleich csBereit), dann wird dieser Anruf über die zusätzliche CallWaiting-Leitung angezeigt. Sofort danach wird der Anruf abgewiesen und die CallWaiting-Leitung steht für einen weiteren anklopfenden Teilnehmer zur Verfügung.

6.10 Gesprächsaufzeichnung

Immer, wenn eine Verbindung zustandekommt wird die Callbackfunktion QueryTapeProc aufgerufen. Dabei kann man festlegen, ob das Gespräch aufgezeichnet werden soll oder nicht. Weiter kann auch die maximale Aufzeichnungsdauer und der Dateiname angegeben werden. Sollte beim Erstellen der Datei ein Fehler auftreten, so kann man das im Error-Trace erkennen.

6.11 Rufweitervermittlung

Wenn der Anrufer an eine andere Nummer weitervermittelt werden soll, kann dies mit der Funktion ISDNExplicitCalltransferTo geschehen. Zunächst wird das Gespräch Gehalten und dann der Zielteilnehmer angewählt. Nimmt dieser den Anruf an, werden Anrufer und Zielteilnehmer verbunden.

6.12 Dual-Tone Multi Frequency (DTMF) versenden und empfangen

Das versenden und empfangen von DTMF ist eine Funktion der ISDN-Karte.

Empfang (nur T-Concept PC und Eicon DIVA ISDN+CT)

Das Empfangen (= abhören des eingehenden Datenstroms) wird bei Beginn einer Verbindung gestartet und dauert bis Verbindungsende. Erkannte Töne werden über DTMFProc signalisiert.

Senden (alle Karten)

DTMF-Töne können bei Verbindungsstatus csVerbunden versandt werden. Dabei kehrt die Funktion ISDNSendDTMF sofort zurück. Das Ergebnis des DTMF-Auftrags wird über DTMFProc mitgeteilt.

Hinweis: DTMFProc kann schon VOR Übertragung mit dem Ergebnis dtmfSendResult zurückkehren. Dies bedeutet, daß die Übertragung von Tönen noch andauern kann, wenn DTMFProc signalisiert wird.

Wichtig für BIPAC PCI Pro: Es sollten immer nur einzelne Ziffern gesendet werden, da Ziffernfolgen von der Karte nicht vollständig übertragen werden.

Einstellungen

Damit alle Töne korrekt erkannt werden, müssen Sender und Empfänger aufeinander abgestimmt sein. Dazu lassen sich die Ton und Pausenlängen einstellen. (Die Werte müssen **vor** dem Aufbau einer Verbindung gesetzt werden).

6.13 Wave-Dateien senden (nur für Billion BIPAC PCI pro Karten!)

Wenn das Senden mit der Funktion SendWavefile erfolgreich war, wird zunächst der Datenmodus eingestellt. Bei Erreichen des Datenmodus (Callback CallStateProc, Parameter „PConnectInfo“, VerbindungsModus = 2) wird die Datei gesendet. Ob das Senden erfolgreich war, kann man im Parameter „Fehler“ nachsehen (wird auch im Error-Trace ausgegeben!). Nach erfolgreichem Senden wird die Datei geschlossen, der Datenmodus beendet und wieder der Telefon-Modus eingestellt. (Callback CallStateProc, Parameter „PConnectInfo“, VerbindungsModus = 2).

Achtung: Aus technischen Gründen werden die letzten 300-400 Millisekunden der Wavedatei NICHT übertragen. Die Wavedatei sollte deshalb am Ende noch 300-400 Millisekunden Stille enthalten.

6.14 Hörerlautstärke

Mit ISDNSetVolume kann die Hörerlautstärke eingestellt werden. Auch kann zwischen Handset und Headset umgeschaltet werden. Nur bei BIPAC PCI Pro und Eicon DIVA ISDN+CT kann auch die Mikrophonverstärkung eingestellt werden.

Eicon DIVA ISDN+CT:

Mit der neuen Funktion ISDNSetVolumeEx können bei der Eicon DIVA ISDN+CT DSP-Karte die Lautstärkeparameter sehr präzise eingestellt werden. Die Lautstärke kann so individuell an die verwendeten Hand/Headset bzw. Mikrophon und Lautsprecher Kombinationen und die Vorlieben des Anwenders angepaßt werden.

Nach dem Start des Telefonmoduls ist der RJ-Stecker „Headset 1“ am Slotblech der DIVA-Karte aktiviert. Alle Gesprächsdaten werden auf diesem Anschluß ausgegeben bzw. aufgenommen. Falls die Klinkenstecker verwendet werden sollen, müssen diese **vor** dem Aufbau einer Verbindung mit der Funktion ISDNSetVolumeEx und dem Command #2 aktiviert werden.

6.15 Deinitialisierung

Bei Programmende müssen interne Strukturen freigegeben werden. (ISDNDone, ISDNFree).

7 Systemvoraussetzungen

Das Telefonmodul benötigt folgende Komponenten:

7.1 Betriebssystem Microsoft Windows 98 / ME / NT 4 / 2000

Die Funktionsfähigkeit unter Windows 98 / ME / NT 4 / 2000 wird sichergestellt. Bei Verwendung von Windows 2000 müssen ausreichend getestete CAPI 2.0 Treiber vorliegen.

7.2 ISDN-Karte

Es werden die ISDN-Karten T-Concept PC und BIPAC PCI Pro und Eicon DIVA ISDN+CT unterstützt. Andere ISDN-Karten werden nicht unterstützt.

T-Concept ISDN PC Karte der Deutschen Telekom AG.

Die T-Concept PC Karte muß mit einem Kabel an den S₀-Bus angeschlossen sein. Aus technischen Gründen, die ihre Ursache im Aufbau der T-Concept PC ISDN Karte haben, darf während des Telefonierens mit dem Telefonmodul der 2. B-Kanal auf der ISDN-Karte nicht verwendet werden. Das gleichzeitige Telefonieren und z.B. faxen ist deshalb auf dem Rechner nicht möglich. Sollte der 2. B-Kanal belegt sein, so kann es a) zu Störungen der Sprachübertragung und b) zu Störungen beim Senden /Empfang von DTMF kommen.

BIPAC PCI Pro von Billion Electrics, Taiwan

Die BIPAC PCI Pro Karte muß mit einem Kabel an den S₀-Bus angeschlossen sein. Während einem Telefonat sollte der 2. B-Kanal frei bleiben. Bei der BIPAC PCI Pro ist die Erkennung von DTMF-Tönen nicht möglich.

Eicon DIVA ISDN+CT

Die Karte muß mit einem Kabel an den S₀-Bus angeschlossen sein.

7.3 Hinweis zu den ISDN-Treibern der T-Concept Karte

Die z.Z. mit der Karte mitgelieferten Treiber in der Version 1.0 build 41 enthalten die Capi 2.0 Release 2.61. Diese Version enthält einige Fehler und ist veraltet.

Voraussetzung für das einwandfreie Funktionieren des Telefonmoduls ist Capi 2.0 Release 2.70 vom Januar 2000, die in den aktuellen T-Concept Treibern (Version 2.0 build 80) enthalten ist. Diese Treiber können über den Support-Server der Telekom bezogen werden.

7.4 Telefonhörer

Der mitgelieferte Telefonhörer muß an die entsprechende Buchse angeschlossen werden. Bei Verwendung von alternativen Telefonhörern muß auf die einwandfreie Funktionalität des Hörerkontaktes geachtet werden.

7.5 Unzulässige Systemkomponenten

Nach Programmstart analysiert und initialisiert das Telefonmodul die ISDN-Karte. Sollte dabei ein Fehler auftreten (Fehlende CAPI, nicht installierte ISDN-Karte), wird dies am Rückgabewert der Funktion ISDNInit (FALSE) deutlich.

Leider kann das Telefonmodul die ISDN-Karten nicht direkt identifizieren. Es überprüft deshalb die Hersteller-Angaben des CAPI-Treibers. Es obliegt dem Anwender, ausschließlich die unterstützten DSP-ISDN-Karten zu verwenden, da nur so die Funktion des Telefonmoduls gewährleistet ist.

7.6 ISDN-Anschluß (S₀-Bus)

Das Programm benötigt einen einwandfrei funktionierenden und den Standards entsprechenden ISDN-Anschluß (Vermittlungsstelle und evtl. Telefonanlage). Vor allem durch fehlerhafte Firmware in Telefonanlagen können Fehler im Programmablauf entstehen, die nicht durch das Programm verursacht sind.

8 Technische Informationen

8.1 Das Telefonmodul wird in kompilierter Form zur Verfügung gestellt

Das Telefonmodul besteht aus einer compilierten Datei

- Dynamic Link Library isdnphoneDSP.dll
- Dazu wird ein Beispiel-Programm inkl. Sourcen geliefert (Borland Delphi 5)

8.2 Wichtige Information zur T-Concept PC ISDN-Karte

In diesem Pflichtenheft wird die derzeit (Oktober 2001) als „T-Concept ISDN PC“ erhältliche ISDN-Karte mit integriertem Audio-Chip und Telefonhörer als „T-Concept ISDN PC“ bezeichnet. Die Bezeichnung für diese Karte kann sich zukünftig ändern, genauso wie sich die zugrundeliegende Hardware bei gleichem Namen ändern kann. Grundsätzlich beziehen sich alle Angaben auf den aktuellen Wissensstand von Oktober 2001.

Das Telefonmodul benutzt eine proprietäre Erweiterung der CAPI 2.0 Spezifikation. Bei Änderungen dieser Erweiterung kann das Telefonmodul nicht mehr oder nur fehlerhaft funktionieren. Das Telefonmodul unterstützt Capi 2.0 Versionen ab 2.69.x . bis zur aktuellen Entwickler-Version 2.77.x. Es wird deshalb empfohlen, nach Installation des Telefonmoduls eine aktuelle Capi 2.0 Version zu installieren und diese beizubehalten. Weiterhin kann eine Änderung der ISDN-Hardware eine inkompatible Modifikation der Capi 2.0 verursachen. Die Verfügbarkeit von zur T-Concept PC ISDN-Karte kompatiblen ISDN-Karten in der Zukunft kann nicht garantiert werden.

8.3 Wichtige Information zur Billion BIPAC-PCI Pro ISDN-Karte

In diesem Pflichtenheft wird die derzeit (Oktober 2001) als „Billion BIPAC PCI Pro“ erhältliche ISDN-Karte mit integriertem Audio-Chip und Telefonhörer als „BIPAC PCI Pro“ bezeichnet. Die Bezeichnung für diese Karte kann sich zukünftig ändern, genauso wie sich die zugrundeliegende Hardware bei gleichem Namen ändern kann. Grundsätzlich beziehen sich alle Angaben auf den aktuellen Wissensstand von Oktober 2001.

8.4 Wichtige Information zur Eicon DIVA ISDN+CT ISDN-Karte

Die speziellen Audiofunktionen der DIVA ISDN+CT wurden unter Zuhilfenahme proprietärer Erweiterungen der CAPI-Spezifikation implementiert.

8.5 Folgende ISDN-Fehlermeldungen können während dem Betrieb des Telefonmoduls auftreten

Das ISDN ist ein komplexes Netzwerk mit vielen möglichen Fehlerquellen. Allerdings ist nicht jeder Fehler kritisch. In der folgenden Tabelle sind die möglichen Fehlercodes aufgelistet. Im Telefonmodul werden diese Fehlercodes nur in der *TDisconnectInfo*-Struktur im Parameter *FehlerCode* verwendet.

Häufige Fehler

Erfahrungsgemäß treten nur wenige Fehler wirklich auf:

Die **fettgedruckten** Fehler sind nicht kritisch. Sie treten im Zusammenhang mit Telefonaten auf und geben an, warum eine Verbindung nicht zustande kam. (z.B. User Busy: der Anschluß ist besetzt oder No circuit/channel available: Kein freier B-Kanal verfügbar).

Ein kritischer Fehler ist besonders wichtig

0x3301, Protocol error layer 1 (broken line or B-channel removed by signalling protocol)

Dieser Fehler tritt normalerweise auf, wenn die Verbindung von ISDN-Karte zur Telefonanlage oder generell zum S₀-Bus gestört ist (Kabel nicht eingesteckt etc.).

Fehler-Ursachen

Grundsätzlich werden die Fehler im Bereich 0x0??? bis 0x33?? von der CAPI oder vom Telefonmodul verursacht.

Die Fehler 0x34?? dagegen stammen alle aus dem ISDN und stellen kein Problem des Telefonmoduls oder der CAPI dar.

Fehler-Codes im ISDN:

0x0001	NCPI not supported by current protocol, NCPI ignored
0x0002	Flags not supported by current protocol, flags ignored
0x0003	Alert already sent by another application
0x1001	Too many applications
0x1002	Logical block size too small, must be at least 128 Bytes
0x1003	Buffer exceeds 64 kByte
0x1004	Message buffer size too small, must be at least 1024 Bytes
0x1005	Max. number of logical connections not supported
0x1006	Reserved
0x1007	The message could not be accepted because of an internal busy condition
0x1008	OS resource error (no memory ?)
0x1009	CAPI not installed
0x100A	Controller does not support external equipment
0x100B	Controller does only support external equipment
0x1101	Illegal application number
0x1102	Illegal command or subcommand or message length less than 12 bytes
0x1103	The message could not be accepted because of a queue full condition !! The error code does not imply that CAPI cannot receive messages directed to another controller, PLCI or NCCI
0x1104	Queue is empty
0x1105	Queue overflow, a message was lost !! This indicates a configuration error. The only recovery from this error is to perform a CAPI_RELEASE
0x1106	Unknown notification parameter
0x1107	The Message could not be accepted because of an internal busy condition
0x1108	OS Resource error (no memory ?)
0x1109	CAPI not installed
0x110A	Controller does not support external equipment
0x110B	Controller does only support external equipment
0x2001	Message not supported in current state
0x2002	Illegal Controller / PLCI / NCCI
0x2003	Out of PLCI
0x2004	Out of NCCI
0x2005	Out of LISTEN
0x2006	Out of FAX resources (protocol T.30)
0x2007	Illegal message parameter coding
0x3001	B1 protocol not supported
0x3002	B2 protocol not supported
0x3003	B3 protocol not supported
0x3004	B1 protocol parameter not supported
0x3005	B2 protocol parameter not supported
0x3006	B3 protocol parameter not supported
0x3007	B protocol combination not supported
0x3008	NCPI not supported
0x3009	CIP Value unknown
0x300A	Flags not supported (reserved bits)
0x300B	Facility not supported
0x300C	Data length not supported by current protocol
0x300D	Reset procedure not supported by current protocol
0x3301	<i>Protocol error layer 1 (broken line or B-channel removed by signalling protocol)</i>
0x3302	Protocol error layer 2
0x3303	Protocol error layer 3
0x3304	Another application got that call
0x3311	Connecting not successful (remote station is no FAX G3 machine)
0x3312	Connecting not successful (training error)
0x3313	Disconnected before transfer (remote station does not support transfer mode, e.g. resolution)
0x3314	Disconnected during transfer (remote abort)
0x3315	Disconnected during transfer (remote procedure error, e.g. unsuccessful repetition of T.30 commands)
0x3316	Disconnected during transfer (local tx data underrun)
0x3317	Disconnected during transfer (local rx data overflow)

0x3318	Disconnected during transfer (local abort)
0x3319	Illegal parameter coding (e.g. SFF coding error)
0x3480	Normal Disconnect
0x3481	Unallocated (unassigned) number
0x3482	No route to specified transit network
0x3483	No route to destination
0x3486	Channel unacceptable
0x3487	Call awarded and being delivered in an established channel
0x3490	Normal call clearing
0x3491	User busy
0x3492	No user responding
0x3493	No answer from user (user alerted)
0x3495	Call rejected
0x3496	Number changed
0x349A	Non-selected user clearing
0x349B	Destination out of order
0x349C	Invalid number format
0x349D	Facility rejected
0x349E	Response to STATUS ENQUIRY
0x349F	Normal, unspecified
0x34A2	No circuit / channel available
0x34A6	Network out of order
0x34A9	Temporary failure
0x34AA	Switching equipment congestion
0x34AB	Access information discarded
0x34AC	Requested circuit / channel not available
0x34AF	Resources unavailable, unspecified
0x34B1	Quality of service unavailable
0x34B2	Requested facility not subscribed
0x34B9	Bearer capability not authorized
0x34BA	Bearer capability not presently available
0x34BF	Service or option not available, unspecified
0x34C1	Bearer capability not implemented
0x34C2	Channel type not implemented
0x34C5	Requested facility not implemented
0x34C6	Only restricted digital information bearer capability is available
0x34CF	Service or option not implemented, unspecified
0x34D1	Invalid call reference value
0x34D2	Identified channel does not exist
0x34D3	A suspended call exists, but this call identity does not
0x34D4	Call identity in use
0x34D5	No call suspended
0x34D6	Call having the requested call identity has been cleared
0x34D8	Incompatible destination
0x34DB	Invalid transit network selection
0x34DF	Invalid message, unspecified
0x34E0	Mandatory information element is missing
0x34E1	Message type non-existent or not implemented
0x34E2	Message not compatible with call state or message type non-existent or not implemented
0x34E3	Information element non-existent or not implemented
0x34E4	Invalid information element contents
0x34E5	Message not compatible with call state
0x34E6	Recovery on timer expiry
0x34EF	Protocol error, unspecified
0x34FF	Interworking, unspecified