

The logo for 'aitronic' is displayed in a white sans-serif font within a grey rectangular box. A small red circle is positioned above the letter 'i'.

aitronic



WinDECT v1.58

Windows-Softwarepaket für den Betrieb
eines DECT-Funknetzwerkes

Manual

Wir liefern Ihnen nicht nur unsere mobilen Terminals mit Standard-Software...

sondern entwickeln auch kundenspezifische

- Applikationen für diese Geräte
- PC-Applikationen
- Hardware

und beraten Sie bei

- der Erstellung von Konzepten für die mobile Datenerfassung
- Fragen rund um den Barcode
- Hardware-Problemen
- PC-Problemen

Wenden Sie sich mit Ihren Fragen und Problemen bitte an:



aitronic GmbH
Max-Planck-Str. 19
D-33104 Paderborn

Telefon: +49 (0) 5254 9969-0
Telefax: +49 (0) 5254 9969-40
Internet: <http://www.aitronic.de>
E-Mail: info@aitronic.de

Copyright © aitronic GmbH, 2010

Alle Rechte vorbehalten, insbesondere auch auszugsweise die der Übersetzung, des Nachdrucks, Wiedergabe durch Kopieren oder ähnliche Verfahren.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Ausgaben-Historie

Ausgabe	Änderungen
17.11.2009	Anpassungen an WinDECT v1.54
15.12.2009	<ul style="list-style-type: none">- Anpassungen an WinDECT v1.58- Beschreibungen der DLLs entfernt.- Anbindungen an MTFilseServer und MTDBAServer.
25.3.2010	Ausgaben-Historie und Mitgeltende Dokumente eingefügt.

Mitgeltende Dokumente

- Bedienungsanleitung der entsprechenden Applikation des mobilen Terminals, welche zusammen mit dieser Software betrieben wird.
- MTDBAServer - Mobile Terminal Database Access Server, Bedienungsanleitung
- MTFilseServer - Mobile Terminal File Server, Bedienungsanleitung
- Technisches Handbuch PocketScan, LogiScan, HTE

1. WinDECT 1-1

Funktionen von WinDect..... 1-3

Funktionen beim Start von WINDECT..... 1-3

Kontinuierliche Funktionen von WINDECT 1-4

Funktionen beim Beenden von WINDECT 1-4

Auswahl der seriellen Schnittstelle..... 1-4

WINDECT.INI Konfigurationsparameter..... 1-5

Anschluss der Access-Points 1-5

DECTComNo..... 1-5

DECTServer7 1-6

WuTReset 1-6

Weiterverarbeitung der Daten 1-7

ServerIP 1-7

ServerPort 1-7

Build_STX_ETX_Blocks..... 1-8

STX_ETX_Timeout..... 1-8

SerialNumberKilling..... 1-8

Programmkontrolle 1-9

RunFileControl 1-9

ExitFileControl 1-9

DisplayActiv..... 1-9

WindowMinimize..... 1-10

WindowHide 1-10

Datensicherungsprotokoll LAP-EC..... 1-10

LapECactiv 1-10

LAPECInfoWithPollFlag 1-10

Logging 1-11

LogToLogFile 1-11

LogFilePath 1-11

LogFileExistingTime 1-12

LogOpenClose 1-12

LogProgAction..... 1-12

LogFileHandling 1-12

LogSocketOpenClose 1-12

LogSocketWrite 1-13

LogSocketRead..... 1-13

LogWinDectProtocollnEx 1-13

LogWinDectProtocol..... 1-13

LogWinDectRCV 1-13

LogWinDectTX 1-14

LogWinDectProtocolError..... 1-14

LogLAPECProtocol 1-14

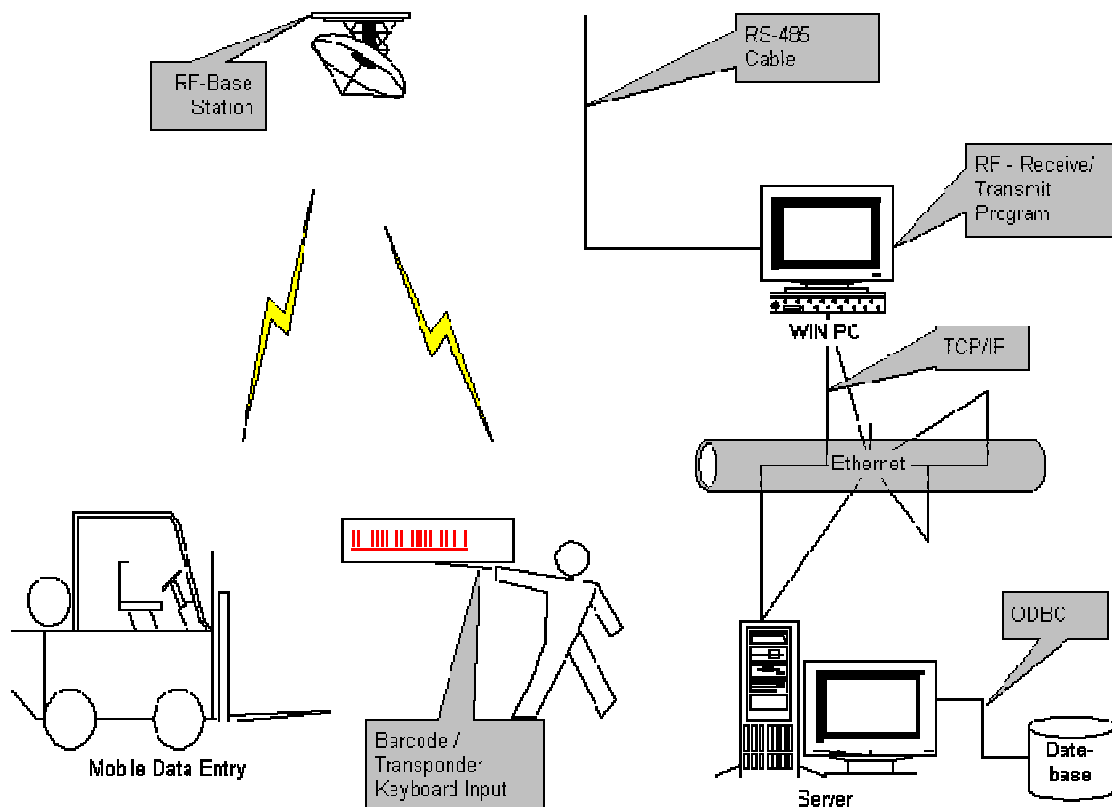
SIO Tracing	1-14
<i>LogSioOpenClose</i>	<i>1-14</i>
<i>LogSioInp</i>	<i>1-15</i>
<i>LogSioOut</i>	<i>1-15</i>
<i>LogSIODTR.....</i>	<i>1-15</i>
<i>LogSIORTS.....</i>	<i>1-15</i>
<i>LogSIORxChar</i>	<i>1-16</i>
<i>LogSIOTxChar</i>	<i>1-16</i>
<i>LogSIOTxEmpty</i>	<i>1-16</i>
<i>LogSIOErr</i>	<i>1-16</i>
<i>LogSIODSR.....</i>	<i>1-16</i>
<i>LogSIOCTS.....</i>	<i>1-17</i>
<i>LogSIOCarrier</i>	<i>1-17</i>
<i>LogSIORING</i>	<i>1-17</i>
WinDECT als Dienst einrichten	1-18
Fehlermeldungen.....	1-20
2. Daten-Schnittstellen.....	2-1
WinDECT als Client-Applikation.....	2-1
<i>MTFileServer</i>	<i>2-1</i>
<i>MTDBAServer</i>	<i>2-2</i>
WinDECT als Server-Applikation	2-3
<i>VB-Skript WinDECT.VBS</i>	<i>2-3</i>
<i>Sortierung der Datensätze nach Scanner-Adressen</i>	<i>2-5</i>
3. SIO-Kommandos für die mobilen Terminals.....	3-6
<i>CM Nachricht anzeigen</i>	<i>3-6</i>
<i>CI Info anzeigen</i>	<i>3-6</i>
<i>CBS Kurzer Ton</i>	<i>3-6</i>
<i>CBL Langer Ton</i>	<i>3-6</i>
4. Testbarcodes	4-7

1. WinDECT

Der digitale Funkstandard **DECT** (Digitally Enhanced Cordless Telecommunications), bekannt durch die schnurlosen Telefone, hat sich auch in der digitalen Datenübertragung bestens bewährt. Im Frequenzbereich von 1,88 bis 1,9 GHz kann dieses Übertragungsverfahren in über 90 Staaten (harmonisierte Normung) ohne Anmeldung problemlos eingesetzt werden.

Für die schnelle Funkanbindung von MDEs (Mobile Daten-Erfassungsgeräte) an Rechnersysteme, sowohl als Ersatz des Anschlusskabels, als auch im Servermode (Anschluss mehrerer Geräte an einen Access-Point) bietet sich dieses System an. Die Vielzahl an Mobilgeräten, die gleichzeitig betrieben werden können ohne sich gegenseitig zu stören, macht den Einsatz in weiten Bereichen des Handels und der Industrie interessant. Die Reichweite solcher Funkmodulanwendungen entspricht etwa einem Radius bis zu 300 Metern, je nach vorhandenen Umgebungsbedingungen (z.B. freier Platz, Stahlbetongebäude).

Die Firma aitronic hat sich bei ihren Anwendungen für die DECT-Engine MD32 von Siemens als Hardware-Plattform entschieden. Dieses zuverlässige und robuste Funk-Modul erlaubt sowohl einfache Punkt zu Punkt Verbindungen, als auch komplexe Punkt zu Mehrpunkt Server Anwendungen.



Mit WINDECT hat aitronic eine Windows-Software entwickelt, mit der (je nach Ausbaustufe) bis zu 64 mobile Geräte gleichzeitig über Funk mit einem PC kommunizieren können. WINDECT übernimmt die Steuerung der Access-Points an den seriellen Schnittstellen des PC und die Weiterleitung der Daten mit Hilfe von Socket-Programmierung über TCP/IP-Ports. Für die Speicherung der Daten in Dateien, Tabellen oder Datenbanken stehen die Windows-Programme MTFileServer und MTDBAServer zur Verfügung. Mit mehrstufigen, täglich neu angelegten LOG-Files kann die Tätigkeit, und Auslastung der Programme kontrolliert, sowie im Fehlerfall die Ursache dafür analysiert werden.

Die umfangreichen Funktionen des Programms WINDECT und des Funk-Netzwerkes werden durch Parameter aus der Datei WINDECT.INI konfiguriert. Darum ist große Sorgfalt angebracht, wenn Änderungen innerhalb der Parameterdatei vorgenommen werden.

Funktionen von WinDect

WINDECT hat die Aufgabe die an die seriellen Schnittstellen oder über das Netzwerk des PCs angeschlossenen Access-Points im Server-Modus zu steuern. Dabei bekommt der Anwender in einem Fenster die Aktivität des Programms angezeigt. Das Programm kann mehrere MD32-Module (Access-Points) an einem PC gleichzeitig verwalten. Jedem einzelnen MD32- Access-Point (FP: Fixed Part Server Mode) können bis zu 16 mit MD32-Modulen ausgestattete mobile Terminals (PP Portable Part) zugeordnet werden.

Die Zuordnung der mobilen Terminals (PP) zu den Access-Points (FP) geschieht durch ein separates Konfigurationsprogramm für die Funkmodule. Dabei werden die weltweit eindeutigen Adressen der Module einander bekannt gemacht und zugeordnet. Die "Abhör- und Fälschungssicherheit" ist dadurch auf der Ebene des DECT-Standards bereits garantiert.

Die von dem mobilen Terminal gesendeten Daten werden von dem Access-Point empfangen und in WINDECT kontrolliert. Die empfangenen Daten werden von WINDECT mittels Socket-Programmierung als Client-Server-Anwendung an ein entsprechendes Server-Anwendungsprogramm weitervermittelt. TCP-Socket-Programmierung ermöglicht als Datenübergabeschnittstelle eine Verknüpfung zweier Programme sowohl auf einem einzigen Rechner, als auch verteilt in einem IP-Netzwerk.

Funktionen beim Start von WINDECT

- Kontrolle einer laufenden Instanz von WINDECT auf dem Rechner
- Einlesen der Datei WINDECT.INI und Auswerten der Parameter
- Erzeugen und Öffnen des täglichen Logfiles `WINDECT_ jjj jmmdd.LOG`
- Anlegen der Laufzeitkontroll-Datei WINDECT.RUN
- Kontrollieren der Socket-Verbindung (z.B. MTFfileServer, MTDBAServer)
- Aktivieren der angegebenen seriellen COMs

Kontinuierliche Funktionen von WINDECT

- Einlesen der Daten von den aktivierten seriellen Schnittstellen
- Entflechten der eingelesenen seriellen Daten
- Übergabekontrolle der aktiven Socket-Ports
- Prüfen und Abhandeln des Datensicherungsprotokolls
- Kontrolle der Datei WINDECT.RUN als Abbruchkriterium
- Löschen alter Logfiles und Archivdateien

Funktionen beim Beenden von WINDECT

- Abbau der Socket-Verbindungen
- Schließen der aktiven seriellen Schnittstellen
- Laufzeitkontroll-Datei WINDECT.RUN löschen
- Erstellen der Programm-Rückgabewert-Datei WINDECT.END
- Schließen des aktiven Logfiles
- Beenden des Programms WINDECT.EXE

Auswahl der seriellen Schnittstelle

Für den Anschluss der DECT-Access-Points ist in WINDECT.INI die serielle Schnittstelle COM1 voreingestellt. Falls die Software für eine oder mehrere andere Schnittstellen konfiguriert werden soll, muss der Parameter `DECTComNo = "1"` in der Datei WINDECT.INI geändert werden. Soll zum Beispiel der Anschluss COM3 verwendet werden, so muss der Parameter `DECTComNo = "3"` gesetzt werden.

Im Abschnitt **WINDECT.INI Konfigurationsparameter** findet man weitere Einzelheiten für den Anschluss der Access-Points.

WINDECT.INI Konfigurationsparameter

Die folgenden Parameter sind in der Datei WINDECT.INI zu finden und werden von dem Programm WINDECT.EXE ausgewertet. Die Parameter werden nur beim Start des Programms einmalig ausgelesen. Änderungen der Parameterwerte in dieser Datei werden deshalb erst dann wirksam, wenn WINDECT erneut gestartet wird.

ACHTUNG: Nach Änderung von Parametern Neustart von WINDECT ausführen!

Unter der Sektion [WINDECT] findet man die nachfolgend beschriebenen Parameter für die Steuerung und das Erscheinungsbild von WINDECT. Änderungen innerhalb der Parameterdatei sollten mit Bedacht und Sorgfalt vorgenommen werden.

Anschluss der Access-Points

DECTComNo

Beispiel: DECTComNo = "2"

Dieser Parameter legt im Programm WINDECT fest, an welcher seriellen Schnittstelle des PC der Access-Point für das DECT-Funknetzwerk direkt angeschlossen ist.

Der Wert "2" bedeutet, dass WINDECT versuchen wird, an COM2 die Software zu installieren. Ist die hier vorgegebene Schnittstelle schon von einem anderen Programm belegt, so erfolgt ein entsprechender Fehlereintrag in die WINDECT-LOG-Datei und WINDECT wird beendet.

WINDECT bietet die Möglichkeit mehr als nur eine serielle Schnittstelle an einem PC zu unterstützen. Zu diesem Zweck können auch mehrere Nummern hinter dem Ausdruck DECTComNo angegeben werden. Die Nummern der Schnittstellen müssen dann mit Komma getrennt werden (Beispiel: DECTComNo = "1, 2, 4" für die seriellen Schnittstellen COM1, COM2 und COM4). Die Schnittstellen COM1 bis COM6 werden derzeit von WINDECT direkt unterstützt.

DECTServer7

Beispiel: DECTServer7 = "192.168.0.155:8000"

Eine andere Möglichkeit den Access-Point anzuschließen, ist die Verwendung eines COM-Servers. Dabei wird die serielle Schnittstelle des MD32-DECT-Moduls in eine TCP/IP-Verbindung umgewandelt. Diese kann dann im lokalen Netzwerk ähnlich einer PC-Verbindung genutzt werden, um Daten mit dem Access-Point auszutauschen. Über die sechs seriellen Schnittstellen (COM1 bis COM6) hinaus, kann mit DECTServer7 bis DECTServer24 die Zahl der anschließbaren Access-Points dadurch erweitert werden.

Die IP-Nummer des angesprochenen Access-Points und die Port-Nummer für den Datenaustausch wird hinter dem Parameter DECTServer_x angegeben. Die IP-Adresse und die Portnummer getrennt durch einen Doppelpunkt werden in der o. a. Weise eingetragen. Diese beiden Werte (IP und Port) sind bei der Auslieferung des Access-Points voreingestellt worden. Eine Änderung kann mit einer TELNET-Verbindung über einen speziellen Serviceport vorgenommen werden.

WuTReset

Beispiel: WuTReset = 1

Access-Points, die einen COM-Server der Firma „Wiesemann & Theis“ enthalten, können mit dem Starten des Programms WINDECT hardwaremäßig zurückgesetzt werden, wenn dieser Parameter auf "1" gesetzt wird. Der Programmstart von WINDECT verzögert sich dadurch um ca. 5 Sekunden pro definiertem Access-Point (DECTServer_X).

Weiterverarbeitung der Daten

Die Daten der mobilen Terminals aus dem DECT-Funknetzwerk können direkt durch eine im Programm WINDECT eingebundene DLL (**D**ynamic **L**ink **L**ibrary) weiterverarbeitet werden. Dazu kann eine entsprechende DLL angegeben werden, die dann beim Start von WINDECT eingebunden wird.

ServerIP

Beispiel: `ServerIP = "127.0.0.1"`

Von den mobilen Stationen empfangene Daten werden nach einer Vorprüfung über Socket-Verbindungen an andere Programme (zum Beispiel PORTSERV) weitergegeben. Rückmeldungen an die aktive mobile Station nutzen dieselbe Verbindung und stellen so einen eindeutigen Bezug zwischen dem mobilen Terminal und dem auswertenden Programm (z.B. Datenbank) her. Das Programm WINDECT ist bei dieser Socket-Verbindung als Client konfiguriert und muss zum Socket-Server eine Verbindung aufbauen.

Die Server IP-Nummer, die bei diesem Parameter eingegeben werden muss, adressiert den Rechner, auf dem das Socket-Server-Programm läuft. Der angesprochene Rechner muss innerhalb des erreichbaren IP-Adressbereichs liegen. Wenn auf demselben PC, auf dem WINDECT gestartet wird, auch das dazugehörige Port-Server Programm läuft, so wird die IP-Nummer "127.0.0.1" hier angegeben. Es wird dann lediglich der PC-interne Transportweg über die Socket-Schnittstellen genutzt und nicht das Netzwerk.

ServerPort

Beispiel: `ServerPort = 2001`

Zusätzlich zur IP-Adresse des anzugebenden Zielrechners muss auch einer der 65535 möglichen Socket-Ports definiert werden, mit dem kommuniziert werden soll. Die ersten 1024 Ports sind nicht frei wählbar, da dort schon Standard-Anwendungen mit ihren speziellen Protokollen darauf zugreifen. Die Firma aitronic nutzt voreingestellt den Port 2001 für diese Verbindung. Der Anwender kann aber ohne Probleme, falls erforderlich, eine andere Nummer (größer 1024) hier eintragen. Man muss lediglich darauf achten, dass bei beiden Programmen (WINDECT und PORTSERV) dieselbe Portadresse eingetragen wird.

Build_STX_ETX_Blocks

Beispiel: `Build_STX_ETX_Blocks = 1`

Bei Aktivierung dieses Parameters werden Datensätze zwischen %%STX... und %%ETX... gesammelt und als komplette Datei über den Server-Socket weitergereicht.

STX_ETX_Timeout

Beispiel: `STX_ETX_Timeout = 60000`

Dieser Parameter ist nur aktiv wenn `Build_STX_ETX_Blocks = 1` gesetzt ist. Nach diesem Timeout wird das Zwischenspeichern von Datensätzen zwischen %%STX... und %%ETX... vom mobilen Terminal oder vom Server-Socket abgebrochen.

SerialNumberKilling

Beispiel: `SerialNumberKilling = 1`

Bei Aktivierung dieses Parameters wird die Seriennummer (welche als Adresse für das mobile Terminal dient) vom Datensatz entfernt bevor dieser über den Server-Socket weitergereicht wird.

Programmkontrolle

RunFileControl

Beispiel: `RunFileControl = 1`

Wenn WINDECT.EXE gestartet wird, erzeugt es im gleichen Verzeichnis die Datei WINDECT.RUN. Diese Datei ist leer und zeigt lediglich an, das WINDECT aktiv ist. Beim Beenden des Programms wird auch WINDECT.RUN automatisch wieder gelöscht. Während WINDECT läuft wird permanent überwacht, ob die erzeugte Datei WINDECT.RUN noch existiert. Löscht man diese Datei im laufenden Betrieb, so beendet sich das Programm selbstständig.

Auf diese Weise können andere Programme leicht kontrollieren, ob WINDECT noch läuft. Andererseits können sie (durch Löschen von WINDECT.RUN) das Programm auch einfach beenden.

ExitFileControl

Beispiel: `ExitFileControl = 1`

Mit `ExitFileControl = 1` erzeugt WINDECT beim Beenden eine Datei WINDECT.END mit dem Rückgabewert des Programms als Inhalt. Der Rückgabewert und seine Bedeutung kann man im Kapitel **Fehlermeldungen** nachschauen.

DisplayActiv

Beispiel: `DisplayActiv = 1`

`DisplayActiv = 1` aktiviert das Anzeigen der Ein- und Ausgaben im Fenster des Programms. `DisplayActiv = 0` verhindert die Anzeige im Programmfenster.

WindowMinimize

Beispiel: `WindowMinimize = 1`

Dieser Parameter ermöglicht das Starten von WINDECT im minimierten Fenstermodus, wenn er auf 1 gesetzt wird. Beim Start des Programms wird nicht das vordefinierte, große Fenster geöffnet, sondern nur das minimierte Fenster in der Taskleiste angezeigt. Durch Anklicken des Programmsymbols in der Taskleiste kann auf die normale Größe umgeschaltet werden.

WindowHide

Beispiel: `WindowHide = 1`

Mit `WindowHide = 1` wird WINDECT ohne sichtbares Fenster gestartet. Es ist weder ein großes Fenster geöffnet, noch ein Symbol in der Taskleiste des Rechners sichtbar. Der Benutzer des Rechners kann lediglich im Taskmanager von Windows erkennen, ob eine Instanz des Programms gerade im Speicher läuft.

Datensicherungsprotokoll LAP-EC

LapECactiv

Beispiel: `LapECactiv = 1`

Zur Sicherung der übertragenen Daten hat die Firma aitronic ein Protokoll der Ebene 2 und 3 nach dem ISO 7-Schichtenmodell implementiert. Dieses Protokoll (angelehnt an HDLC und LAP-B) hat den Namen LAP-EC. Um die Datensicherung in WINDECT mit Hilfe dieses Protokolls zu aktivieren, muss der Parameter `LapECactiv = 1` gesetzt werden.

LAPECInfoWithPollFlag

Beispiel: `LAPECInfoWithPollFlag = 0`

Durch Setzen des Poll-Flags bei Infoblöcken innerhalb des LAP-EC-Protokolls kann der jeweilige Empfänger eines solchen Datenblocks aufgefordert werden, diesen sofort zu quittieren. Ein erhöhtes Datenaufkommen, aber unter bestimmten Umständen eine schnellere Validierung der übertragenen Daten ist die Folge davon.

Logging

LogToLogFile

Beispiel: `LogToLogFile = 1`

Dieser Parameter auf 1 gesetzt, ermöglicht das Mitschreiben der Programmaktionen von WINDECT in eine extra angelegte Ausgabedatei. Die Datei hat den Namen `WINDECT_yyyymm tt .LOG`. Dabei stehen die Buchstaben *yyyy* für die Jahreszahl, *mm* für den Monat und *tt* für das Tagesdatum im Monat. Die Datumsangabe erzeugt WINDECT aus der Systemzeit des PCs, auf dem das Programm ausgeführt wird. Die Log-Datei `WINDECT_20020204.LOG` wurde z.B. von WINDECT am 4. Februar 2002 erzeugt. Sie kann mit einem ASCII-Editor angezeigt werden.

Welche Aktionen des Programms zusätzlich zur Start- und Ende-Information in die LOG-Datei geschrieben werden, bestimmen die Parameter, die weiter unten beschrieben sind.

LogFilePath

Beispiel: `LogFilePath = ".\LogFiles\"`

Das Verzeichnis, in welches die LOG-Datei geschrieben wird, kann hinter dem Parameter `LogFilePath` angegeben werden. Absolute und relative Pfadangaben (relativ zum Speicherort von WINDECT.EXE) sind hier möglich. Es können auch Netzwerklaufwerke angegeben werden. Dabei ist aber zu beachten, dass diese dann immer verfügbar sein müssen.

Gibt man kein Verzeichnis an, aktiviert aber das Schreiben einer LOG-Datei mit `LogToLogFile = 1`, so wird die Datei in dem Verzeichnis erzeugt, in dem WINDECT.EXE gespeichert ist.

LogFileExistingTime

Beispiel: `LogFileExistingTime = 604800`

Damit durch das Schreiben von LOG-Dateien nicht endlos viel Plattenspeicher vollgeschrieben wird, kann durch Angabe eines Wertes hinter `LogFileExistingTime` die Aufbewahrungszeit der LOG-Dateien vorgegeben werden. Nach dieser Zeit (die Zahl definiert die Zeit in Sekunden) löscht WINDECT die LOG-Dateien wieder aus dem angegebenen Verzeichnis.

So werden in dem angegebenen Beispiel alle Dateien, die älter als 7 Tage (604800 Sekunden) sind, und `.\LogFiles\WinDect_???????.LOG` heißen, automatisch von WINDECT gelöscht.

LogOpenClose

Beispiel: `LogOpenClose = 1`

Es werden Einzelheiten beim Starten und Beenden des Programms geloggt.

LogProgAction

Beispiel: `LogProgAction = 1`

Alle wichtigen Aktionen des Programms WINDECT werden geloggt.

LogFileHandling

Beispiel: `LogFileHandling = 1`

Das Erstellen, Löschen, Öffnen, Lesen und Schreiben von und in Dateien wird geloggt.

LogSocketOpenClose

Beispiel: `LogSocketOpenClose = 1`

Wenn Socket-Ports geöffnet und geschlossen werden, so wird dies durch eine 1 bei dem Parameter `LogSocketOpenClose` mitgeloggt.

LogSocketWrite

Beispiel: `LogSocketWrite = 1`

Die Daten und die Handle-Nummer des Socket-Ports, auf den etwas herausgegeben wird, werden durch diesen Parameter mitprotokolliert.

LogSocketRead

Beispiel: `LogSocketRead = 1`

Wenn Daten an einem Socket-Port eingelesen werden, so wird dies zusammen mit der Handle-Nummer des Ports durch Angabe einer 1 hinter dem Parameter `LogSocketRead` im LOG aufgezeichnet.

LogWinDectProtocolInEx

Beispiel: `LogWinDectProtocolInEx = 1`

Jedes Aufrufen und wieder Verlassen der DECT-spezifischen ProgrammROUTINEN wird durch Einschalten dieses Parameters mitprotokolliert.

LogWinDectProtocol

Beispiel: `LogWinDectProtocol = 1`

Der gesamte DECT-spezifische Datenverkehr von und zu den MD32-Modulen wird in das Log-File geschrieben.

LogWinDectRCV

Beispiel: `LogWinDectRCV = 1`

Die empfangenen Daten aus den DECT-Modulen werden im Log aufgezeichnet.

LogWinDectTX

Beispiel: `LogWinDectTX = 1`

Alle Daten, die zu den DECT-Modulen gesendet werden, sind im Log-File wiederzufinden, wenn dieser Parameter auf "1" gesetzt wird.

LogWinDectProtocolError

Beispiel: `LogWinDectProtocolError = 1`

Wenn die DECT-Module Fehler melden, so wird das durch Angabe dieses Parameters mitprotokolliert.

LogLAPECProtocol

Beispiel: `LogLAPECProtocol = 1`

Einzelheiten des aktivierten Daten-Übertragungsprotokolls LAPEC (HDLC/LAP-B) werden in der Log-Datei mitgeschrieben, wenn dieser Parameter mit 1 besetzt wird.

SIO Tracing

Mehr als ein Dutzend weiterer Parameter zum Überwachen der seriellen Schnittstelle kann hier aktiviert werden. Diese hardwarenahen Einstellungen sollten nur mit entsprechender Vorsicht eingeschaltet werden, da sie eine große Datenflut verursachen könnten.

LogSioOpenClose

Beispiel: `LogSioOpenClose = 1`

Details über das Öffnen und Schließen der RS-232 Schnittstelle werden in der LOG-Datei aufgezeichnet.

LogSioInp

Beispiel: `LogSioInp = 1`

Die von der RS-232-Schnittstelle empfangenen Zeichen werden in der LOG-Datei aufgezeichnet.

ACHTUNG: Das Datenaufkommen kann dabei sehr schnell sehr groß werden. Wenn dieser Parameter überhaupt eingeschaltet wird, so sollte das Programm nur kurze Zeit gestartet werden, um eventuelle Fehleranalyse zu betreiben.

LogSioOut

Beispiel: `LogSioOut = 1`

Alle an die RS-232-Schnittstelle gesendeten Zeichen werden in der LOG-Datei aufgezeichnet.

ACHTUNG: Das Datenaufkommen kann dabei sehr schnell sehr groß werden. Wenn dieser Parameter überhaupt eingeschaltet wird, so sollte das Programm nur kurze Zeit gestartet werden, um eventuelle Fehleranalyse zu betreiben.

LogSIODTR

Beispiel: `LogSIODTR = 1`

Das Setzen und Rücksetzen des RS-232-Signals DTR wird geloggt.

LogSIORTS

Beispiel: `LogSIORTS = 1`

Das Setzen und Rücksetzen des RS-232-Signals RTS wird geloggt.

LogSIORxChar

Beispiel: `LogSIORxChar = 1`

Jedes von der RS-232-Schnittstelle empfangene Zeichen wird geloggt.

ACHTUNG: Das Datenaufkommen kann dabei sehr schnell sehr groß werden. Wenn dieser Parameter überhaupt eingeschaltet wird, so sollte das Programm nur kurze Zeit gestartet werden, um eventuelle Fehleranalyse zu betreiben.

LogSIOTxChar

Beispiel: `LogSIOTxChar = 1`

Alle an die RS-232-Schnittstelle gesendeten Zeichen werden geloggt.

ACHTUNG: Das Datenaufkommen kann dabei sehr schnell sehr groß werden. Wenn dieser Parameter überhaupt eingeschaltet wird, so sollte das Programm nur kurze Zeit gestartet werden, um eventuelle Fehleranalyse zu betreiben.

LogSIOTxEmpty

Beispiel: `LogSIOTxEmpty = 1`

Der Transmit Buffer Empty Interrupt der RS-232-Schnittstelle wird geloggt..

LogSIOErr

Beispiel: `LogSIOErr = 1`

Der Character Receive Error Interrupt der RS-232-Schnittstelle wird geloggt.

LogSIODSR

Beispiel: `LogSIODSR = 1`

Der DSR Interrupt der RS-232 Schnittstelle wird geloggt.

LogSIOCTS

Beispiel: `LogSIOCTS = 1`

Der CTS Interrupt der RS-232 Schnittstelle wird geloggt.

LogSIOCarrier

Beispiel: `LogSIOCarrier = 1`

Der DCD Interrupt der RS-232 Schnittstelle wird geloggt.

LogSIORING

Beispiel: `LogSIORING = 1`

Der Ring Indicator Interrupt der RS-232 Schnittstelle wird geloggt.

WinDECT als Dienst einrichten

Mit Hilfe der Utility `WINSERV.EXE` kann WinDECT als Dienstprogramm unter Windows eingerichtet werden.

Ein Dienst läuft für den Anwender unsichtbar im Hintergrund ab. Ein Dienst, der mit der Startart "Automatisch" eingetragen ist, hat den Vorteil, dass er ohne Anmeldung eines Benutzers an dem PC automatisch vom Betriebssystem gestartet wird. Selbst nach einem Absturz oder Stromausfall des PC, läuft das Programm dann ohne Anmeldung eines Benutzers mit dem Hochfahren des Rechners erneut los.

Die Alternative zum Dienstprogramm ist, das Programm im Autostart-Verzeichnis abzulegen. Wenn in der Datei `WINDECT.INI` dann noch der Parameter `WindowHide = 1` gesetzt wird, so läuft das Programm für den Benutzer unsichtbar im Hintergrund ab.

Die Utility `WINSERV.EXE` ist ein Kommandozeilen-Tool. Sie benötigt keine Installation, sondern wird direkt in einer DOS-Box mit den nötigen Parametern aufgerufen (Administratorrechte erforderlich). Mit dieser Utility lässt sich jedes Programm als Dienst einrichten.

Befehls-Struktur:

```
winserv <kommando> <service-name> <service-optionen>  
<programm> <programm-optionen>
```

Folgende Kommandos sind verfügbar:

<code>install</code>	richtet ein Programm als Dienst ein
<code>configure</code>	ändert einen vorhandenen Dienst
<code>uninstall</code>	entfernt den Dienst
<code>showconfig</code>	zeigt die Konfiguration des Dienstes an
<code>start</code>	startet den Dienst
<code>restart</code>	stoppt den Dienst und startet ihn neu
<code>stopp</code>	stoppt den Dienst
<code>pause</code>	hält den Dienst an
<code>continue</code>	setzt den Dienst fort

Alle Befehle außer den ersten beiden können übrigens auf alle Windows-Dienste angewandt werden.

Den Kommandos `install` und `configure` können folgende Parameter mitgegeben werden:

<code>-displayname</code>	Angezeigter Name des Dienstes
<code>-description</code>	Beschreibung (steht in der Windows-Dienstverwaltung in der zweiten Spalte)
<code>-start</code>	Start-Art des Dienstes : <code>auto</code> , <code>demand</code> oder <code>disabled</code>
<code>-depends</code>	Abhängigkeit zu anderen Diensten Syntax: <code>-depends service1,service2,...</code>
<code>-user</code>	Benutzer, unter dem der Dienst läuft
<code>-password</code>	Passwort des angegebenen Benutzers

Der typische Befehls-Aufbau ist wie folgt:

```
winserv install <service-name> <service-optionen> <programm>
<programm-optionen>
```

```
winserv configure <service-name> <service-optionen>
<programm> <programm-optionen>
```

```
winserv uninstall <service-name>
```

So sehen die Kommandos für das Installieren, Starten, Stoppen und Deinstallieren des WinDECT-Dienstes aus:

```
winserv install WINDECT c:\windect\windect.exe
```

```
winserv start WINDECT
```

```
winserv stop WINDECT
```

```
winserv uninstall WINDECT
```

Die Datei [WINSERV.HTM](#) enthält die komplette Dokumentation aller Optionen.

Fehlermeldungen

In der LOG-Datei wird beim Beenden des Programms eine Nummer, der Programm-Rückgabewert (RetVal: x) eingetragen. Dieser Programm-Rückgabewert wird auch in die Datei *program.END* eingetragen, wenn der entsprechende Parameter in der Datei *program.INI* aktiviert wurde (*program* repräsentiert den Namen des Programms, das zu der Datei gehört).

So kann zum Beispiel ein anderes Programm durch Einlesen der Datei *program.End* und Auswerten der dort eingetragenen Zahl beurteilen, ob und welcher Fehler vorliegt.

Code	Beschreibung
0	Programm beendet durch Bediener
1	Programm läuft bereits im Hauptspeicher
2	Initialisierungsfehler
3	Parameterdatei fehlt
4	Fehler beim Öffnen einer Datei "Open"
5	Fehler beim Schreiben in eine Datei "Write"
6	Fehler an serieller Schnittstelle "COMx"
7	maximale Anzahl Dateien erreicht
8	maximale Anzahl mobiler Clients erreicht
9	Programm mit Taste (z.B. ESC) beendet
10	Programm Laufzeit Endetermin erreicht
11	Laufzeit des Programms beendet
12	Laufzeitende für Leerlauf des Programms erreicht
14	Laufzeitende für "NETON_PAUSE" erreicht
20	"Exit" Kommando empfangen (nur bei NAC)
21	Laufzeit-Kontrolldatei <i>program.RUN</i> fehlt
22	PROTECT "NETOFF" empfangen (nur bei NAC)
23	DSR Signal fehlt
24	PROTECT Cascaden Fehler
30	PROTECT Netzwerk Fehler (Firmware nicht erkannt)
31	PROTECT Fehler des Access-Points bzw. NARF (Cascade)
40	Datei Fehler
50	Socket-Port "WSA Startup" Fehler
51	Socket-Port "No Socket" Fehler
52	Socket-Port "Socket" Fehler
53	Socket-Port "Connect" Verbindungsfehler
54	Socket-Port "No Name" Fehler
60	ODBC SQL "Connection" Verbindungsfehler
100	Programm "Thread" Erstellungsfehler
101	DLL Ladefehler "LoadLibrary"
102	DLL-Library Fehler bei "GetProgAddress"
180	Programm "Thread" Erstellungsfehler
181	Programm "Thread" Beendigungsfehler
198	Programm "Exeption"

2. Daten-Schnittstellen

WinDECT als Client-Applikation

MTFileServer

Der MTFileServer stellt die von WinDECT empfangenen Daten für die Weiterverarbeitung in Text-Dateien zur Verfügung. Eine detaillierte Beschreibung des MTFileServers ist in dem Dokument **MTFileServer.pdf** enthalten und kann von www.aitronic.de heruntergeladen werden.

Um WinDECT mit der Server-Applikation MTFileServer zu verbinden, ist IP-Adresse, auf dem MTFileServer ausgeführt wird, anzugeben. Die Angabe der Port-Nr. muss mit dem entsprechenden Parameter in MTFileServer.ini übereinstimmen. Z.B.:

```
ServerIP = "192.168.0.102:7676"
```

Sollte WinDECT und MTFileServer auf dem gleichen Host laufen, so ist

```
ServerIP = "127.0.0.1:7676"
```

anzugeben. Da MTFileServer die reinen Anwenderdaten (ohne die Seriennummer, die als Client-Adresse fungiert) erwartet, ist außerdem der Parameter

```
SerialNumberKilling = 1
```

anzugeben.

MTDBAServer

Der MTDBAServer dient zur Anbindung von WinDECT über eine ODBC-Connection an eine Datenbank. Eine detaillierte Beschreibung des MTDBAServer ist in dem Dokument **MTDBAServer.pdf** enthalten und kann von www.aitronic.de heruntergeladen werden.

Um WinDECT mit der Server-Applikation MTDBAServer zu verbinden, ist IP-Adresse, auf dem MTDBAServer ausgeführt wird, anzugeben. Die Angabe der Port-Nr. muss mit dem entsprechenden Parameter in MTDBAServer.ini übereinstimmen. Z.B.:

```
ServerIP = "192.168.0.102:7678"
```

Sollte WinDECT und MTFilerServer auf dem gleichen Host laufen, so ist

```
ServerIP = "127.0.0.1:7678"
```

anzugeben.

Da MTDBAServer die reinen Anwenderdaten (ohne die Seriennummer, die als Client-Adresse fungiert) erwartet, ist außerdem der Parameter

```
SerialNumberKilling = 1
```

anzugeben.

WinDECT als Server-Applikation

VB-Skript WinDECT.VBS

Voraussetzung für die Lauffähigkeit von VB-Skripten mit Winsock-Funktionalität ist die Installation/Registrierung folgender zum Testpaket gehörender Dateien notwendig:

- `Mswsock.dll`
- `Winsock.dll`
- `Mswinsck.ocx`

Die Installation/Registrierung dieser Dateien wird durch die Ausführung von

```
MS-Winsock.bat
```

durchgeführt.

Der im WinDECT-Software-Paket enthaltenen VB-Skript `WinDect.vbs` soll als Beispiel dienen und stellt folgende Funktionalität zur Verfügung:

- Herstellung einer Winsock-Verbindung zu WINDECT über die IP-Adresse LOCAL HOST Port 4004
- Herstellung einer Verbindung zur Microsoft-Access-Datenbank LAGER.MDB
- Empfang einer Artikelnummer von einem mobilen Terminal
- Abfrage der Artikelbezeichnung zu dieser Artikelnummer aus der Datenbank
- Senden der Artikelbezeichnung (oder der entsprechenden Rückmeldung, falls diese nicht in der Datenbank gefunden wird) an das mobile Terminal.

Die VB-Skripte können relativ einfach an spezielle Anforderungen angepasst werden. Es genügen ein einfacher Text-Editor und Kenntnisse in Windows-Scripting, Visual BASIC und COM-basierter Programmierung. Empfohlen werden können dafür die beiden Bücher **Windows-Scripting** und das **COM-Komponenten-Handbuch** von Holger Schwichtenberg aus dem Verlag ADDISON-WESLEY.

Damit WinDECT.exe eine Socket-Verbindung des als Client fungierenden Skripts WinDECT.VBS akzeptiert, ist folgende Einstellung in WinDECT.ini notwendig:

```
ConfigServicePort =4004
```

Um WinDECT zu aktivieren, wird nach dem Connect von WinDECT.VBS der Befehl „RUN“ gesendet. Als Seriennummer wird hierzu die „000000“ verwendet.

```
000000;RUN<CR>
```

In gleicher Weise kann mit dem Befehl „STOP“ WinDECT deaktiviert werden. Das Starten von WinDECT.EXE und dem gewünschten Skript erfolgt mit dem entsprechenden Batch-Programm. WinDect.BAT enthält z.B. folgende beide Zeilen:

```
start windect  
call cscript windect.vbs > cscript.log
```

Die Bildschirm-Ausgaben des Skripts werden in die Log-Datei CSCRIPT.LOG umgeleitet.

Sortierung der Datensätze nach Scanner-Adressen

Werden Daten von mehreren Scannern gleichzeitig übertragen, so werden diese „unsortiert“ in der zeitlichen Reihenfolge wie WinDECT sie empfängt, über die IP-Verbindung weitergereicht. Sollen die Daten nach Scanner-Adressen sortiert empfangen werden, so ist folgendes zu beachten:

- Die Daten müssen von jedem Scanner mit den Framing-Zeilen %%STX... und %%ETX..., (MTWin-Datenformat) gesendet werden.
- In WinDECT.INI ist der Parameter `Build_STX_ETX_Blocks = 1` zu setzen.

Sind diese Voraussetzungen erfüllt, puffert WinDECT die Daten getrennt nach der Scanner-Adresse und reicht diese erst nach Empfang des %%ETX-Satzes über die IP-Verbindung weiter.

Die Daten in der Gegenrichtung müssen ebenfalls mit %%STX/%%ETX-Zeilen gerahmt werden. Um zu verhindern, dass WinDECT direkt nach Empfang der %%ETX-Zeile evtl. gepufferte Daten eines anderen mobilen Terminals sendet, muss bereits jetzt (vor dem Senden der Daten) das Kommando

```
000000;XOFF<CR>
```

an WinDECT gesendet werden.

Erst nach Empfang der %%ETX-Zeile sendet WinDECT die Daten an das entsprechende mobile Terminal.

Ist die Bereitschaft zum Empfang weiterer Daten von mobilen Terminals wieder gegeben, muss das Senden mit dem Kommando

```
000000;XON<CR>
```

freigegeben werden.

3. SIO-Kommandos für die mobilen Terminals

Für Daten, die zum einem mobilen Terminal mit Standard-Programm 03000 geschickt werden, können folgende Kommandos verwendet werden:

CM Nachricht anzeigen

Ausgabe einer Nachricht auf dem Display des mobilen Terminals. Das Kommando muss folgendes Format haben:

```
CMccttt...ttt<CR>          cc      Cursorposition (00 - 31),  
                             tt...tt  Text.
```

Eine Nachricht muss am mobilen Terminal mit ENTER bestätigt werden, bevor andere Aktionen durchgeführt werden können. Auch weitere SIO-Kommandos werden erst nach dieser Bestätigung verarbeitet!

CI Info anzeigen

Ausgabe einer Info auf dem Display des mobilen Terminal. Das Kommando muss folgendes Format haben:

```
CIccttt...ttt<CR>          cc      Cursorposition (00 - 31),  
                             tt...tt  Text.
```

Ein Info-Text braucht entgegen der Nachricht (Kommando **CM**) am mobilen Terminal nicht mit ENTER bestätigt zu werden, sondern wird durch die nächste Display-Ausgabe überschrieben.

CBS Kurzer Ton

```
CBS<CR>
```

Ausgabe eines kurzen Tons auf dem Beeper des mobilen Terminals.

CBL Langer Ton

```
CBS<CR>
```

Ausgabe eines langen Tons auf dem Beeper des mobilen Terminals.

Für kundenspezifische Applikationen können eigene SIO-Kommandos definiert werden.

4. Testbarcodes

Folgende Testbarcodes sind in der Datenbank LAGER.MDB als Artikelnummern hinterlegt. WinDECT.VBS benutzt LAGER.MDB als Datenbank.



11111111



22222222



33333333



44444444



55555555