

Beschreibung zum Kommunikationsmodul DFComDLL

- 1 Einleitung
 - 1.1 Hauptthemen des Dokumentes
 - 1.2 Vorteile einer DLL
- 2 Einbindung der DLL
 - 2.1 Einbindung unter Visual Basic
 - 2.2 Einbindung unter Visual Basic .NET
 - 2.3 Einbindung unter C++
 - 2.4 Einbindung unter C#
- 3 Verbindungsarten
 - 3.1 Passive Geräteverbindungen
 - 3.2 Aktive Geräteverbindung
- 4 Abrufen von Informationen über das Datafox-Terminal
 - 4.1 Abfragen der Firmwareversion und des Gerätetyps
 - 4.2 Ermitteln der Seriennummer
 - 4.3 Ermitteln von Informationen zu eingebaute / angeschlossene Module
- 5 Geräteeinrichtung
 - 5.1 Übertragen von Firmware oder Benutzerspezifischer Textdaten (Firmware, Textdaten)
 - 5.2 Abgleich der Uhrzeit
 - 5.3 Übertragen vom Setup (Setupdatei)
 - 5.4 Setzen der Konfiguration für GSM, GPRS
 - 5.5 Setzen der Konfiguration zur Aktiven Verbindung
 - 5.6 Übertragen von Listen (Listendaten, auch für Zutritt)
 - 5.7 Zugriff auf vorhandene Listendaten
- 6 Abrufen der erzeugten Datensätze
 - 6.1 Datensatzstruktur ermitteln
 - 6.2 Daten zu einem Datensatz abrufen und bestätigen
- 7 Nette Hilfsfunktionen
 - 7.1 Schalten eines Relais
 - 7.2 Anzeigen von Text am Display

1 Einleitung

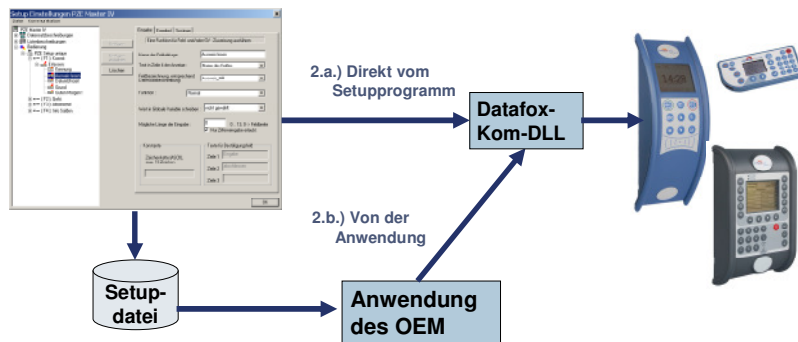
- Dieses Dokument erklärt die Einbindung und Funktionsweise des Kommunikationsmoduls, um mit den Datafox-MasterIV Geräten zu kommunizieren. Es richtet sich in erster Linie an Interessierte, die einen Groben Überblick über das Kommunikationsmodul erhalten wollen.
- Bei dem Kommunikationsmodul handelt es sich um eine Win32-DLL, die Sie in Ihre Softwarelösung integrieren, um mit den Datafox-MasterIV Geräten zu kommunizieren.
- DLL steht für Dynamic Link Library und bezeichnet eine Zusammenstellung von Funktionen zu einem bestimmten Bereich. Es ist ein Programmmodul, welches Sie in Ihre Softwarelösung einbinden um mit dessen Funktionalität die Datafox-MasterIV Geräte anzusprechen. Eine genauere Erklärung der Technik hinter DLL's finden Sie unter http://de.wikipedia.org/wiki/Dynamic_Link_Library

Setup für die Geräte erstellen und übertragen



1. Setup erstellen

2. Setup übertragen

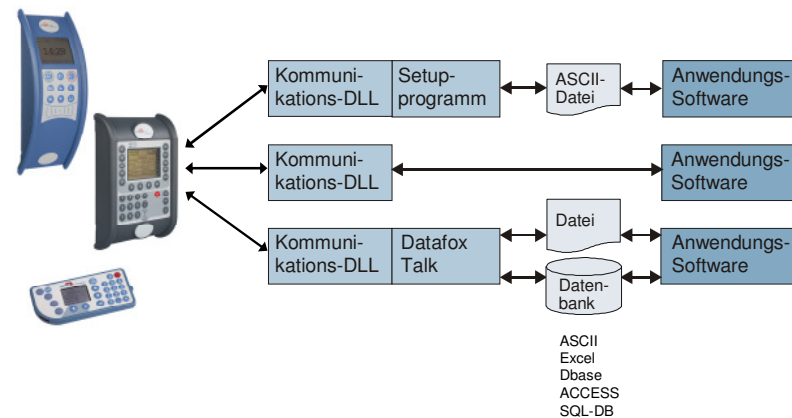


Übertragungsmöglichkeiten für das Setup per:

- ▶ Datafox Setup
- ▶ Datafox Talk
- ▶ Datafox Kommunikations-DLL direkt von der Anwendung

Folie 22 © Datafox GmbH – www.datafox.de

Kommunikationswege



Folie 23 © Datafox GmbH – www.datafox.de

Notizen :

[illegible]

- Bitte beachten Sie, dass es sich bei dem Dokument um eine kurze Beschreibung grundlegender Herangehensweisen handelt. Das Dokument ersetzt nicht die beiliegende DLL-Dokumentation in der alle Funktionen der DLL mit deren Parameter beschrieben sind.**

- Kann von der Programmiersprache unabhängig verwendet werden.
- Aktualisierbar ohne Neuerstellung Ihrer Softwarelösung.

2 Einbindung der DLL

Um die Funktionen der DLL ansprechen zu können, muss sie in Ihre Softwarelösung eingebunden werden. Hierbei ist je nach Entwicklungsumgebung eine gewisse Vorgehensweise notwendig. Prinzipiell ist allen eines gemeinsam, die benötigten Funktionen müssen Ihrer Softwarelösung bekannt gegeben werden (sie sind zu Deklarieren).

Mit der Funktion DFCComOpenIV wird eine Verbindung zum Datafox-Terminal hergestellt. Wie Sie diese Funktion nun je nach verwendeter Programmiersprache Deklarieren wird in den nun folgenden Kapiteln beschrieben. Ein Komplette Liste der DLL-Funktionen finden Sie in den jeweiligen Anwendungsbeispielen zur DFComDLL.

2.1 Einbindung unter Visual Basic

Um z.B. die Funktion DFCComOpenIV unter Visual Basic Classic verwenden zu können muss sie wie folgt Deklariert werden:

```
Public Declare Function DFCComOpenIV Lib "DFComDll" Alias _  
    "_DFCComOpenIV@24" (ByVal channelId As Long, ByVal deviceID As Long, ByVal commType As Long,  
    ByVal commString As String, ByVal commValue As Long, ByVal commTimeout As Long) As Long
```

Nach Bekanntgabe (Deklaration) der Funktion kann Sie verwendet werden um eine Verbindung zum Datafox-Terminal aufzubauen.

2.2 Einbindung unter Visual Basic .NET

Um z. B. die Funktion DFCComOpenIV unter Visual Basic .NET verwenden zu können kann sie wie folgt Deklariert werden:

```
Public Declare Function DFCComOpenIV Lib "DFComDll" Alias _  
    "_DFCComOpenIV@24" (ByVal channelId As Integer, ByVal deviceID As Integer, ByVal commType As  
    Integer, ByVal commString As String, ByVal commValue As Integer, ByVal commTimeout As Integer) As  
    Integer
```

Notizen :

Nach Bekanntgabe (Deklaration) der Funktion kann Sie verwendet werden um eine Verbindung zum Datafox-Terminal aufzubauen.

2.3 Einbindung unter C++

Um z. B. die Funktion DFComOpenIV unter C++ / Visual C++ verwenden zu können muss sie wie folgt Deklariert werden:

```
extern "C" __declspec(dllimport)
int __stdcall DFComOpenIV(int channelID, int deviceID, int commType, char *commString, int
commValue, int commTimeout);
```

Nach Bekanntgabe (Deklaration) der Funktion kann Sie verwendet werden um eine Verbindung zum Datafox-Terminal aufzubauen.

Hinweis:

Hierbei muss noch beachtet werden, ob die Bindung implizit oder explizit sein soll. Für eine implizite Bindung müssen die Win32-API Funktionen LoadLibrary, GetProcAddress und FreeLibrary verwendet werden. Um eine explizite Bindung anzuwenden liegt dem DLL-Paket eine DFComDLL.lib bei.

2.4 Einbindung unter C#

Um z. B. die Funktion DFComOpenIV unter C# verwenden zu können kann sie wie folgt Deklariert werden:

```
[DllImport("DFComDLL.dll", EntryPoint="_DFComOpenIV@24", CharSet=CharSet.Ansi)]
public static extern int ComOpenIV(int channelID, int deviceID, int type, string commString, int
commValue, int commTimeout);
```

Nach Bekanntgabe (Deklaration) der Funktion kann Sie verwendet werden um eine Verbindung zum Datafox-Terminal aufzubauen.

Notizen :

3 Verbindungsarten

Es gibt zwei Verbindungsarten der Datafox-Terminals.

- Bei der Passiven Geräteverbindung wird durch die DLL die Verbindung zu den Terminals aufgebaut,
- bei der Aktiven Geräteverbindung ist es umgekehrt, dort wird durch das Datafox-Terminal die Verbindung zur DLL aufgebaut.

Hinweis:

Eine Geräteverbindung wird in der DLL – Dokumentation als Kanal bezeichnet. Jede Geräteverbindung hat eine eindeutige ID, welche bei Funktionsaufrufen angegeben werden muss. Durch diese Kanal-ID „weiß“ die DLL mit welchem Gerät Sie kommunizieren möchten.

3.1 Passive Geräteverbindungen

Eine Passive Geräteverbindung stellen Sie mit der Funktion *DFCComOpenIV* her und beenden Sie mit der Funktion *DFCComClose*. Je nach Kommunikationsart übergeben Sie der Funktion entsprechende Parameter. Grundsätzlich wird zwischen zwei Verbindungstypen unterschieden.

3.1.1 Verbindungstyp: Serielle Kommunikation

- RS232 (über Umsetzer auch RS485)
- USB
- Modem

Hinweis:

Modemverbindungen sind über die Funktionen *DFCWrite* und *DFCRead* mittels AT-Kommandos zu steuern. Über diese Funktionen können Sie eine Initialisierung, Anwahl und das Auflegen realisieren. Wenn sich das Modem im Transparenten Modus befindet, können die DLL-Funktionen wie bei jedem anderen Verbindungstyp aufgerufen werden.

Zusätzlich zu Funktionsaufrufen, muss neben der Kanal-ID auch eine sogenannte Busnummer angegeben werden. Die Busnummer ist nur in RS485 Netzwerken ungleich 254.

Notizen :

3.1.2 Verbindungstyp: TCP / IP

- TCP / IP
- GPRS

3.2 Aktive Geräteverbindung

Die Aktive Geräteverbindung bedingt eine entsprechende Einrichtung der Geräte und der DLL.

3.2.1 Aus Sicht des Datafox-Terminals

Die Verwendung der Aktiven Geräteverbindung bedingt eine entsprechende Einrichtung der Datafox-Terminals. Hierbei belegen Sie über die Funktion *DFCSetGlobVar* die notwendigen Systemvariablen der Aktiven Verbindung mit den notwendigen Verbindungsdaten. Das setzen der Systemvariablen beinhaltet Informationen wie z. B. wohin sich das Gerät nach Systemstart verbinden soll. Die Einrichtung kann selbstverständlich auch über das DatafoxStudioIV erfolgen.

3.2.2 Aus Sicht der DLL

Durch Aufruf der Funktion *DFCStartActiveConnection* startet die DLL intern die Verarbeitung der Aktiven Verbindung. Sie wartet hiernach auf eingehende Verbindungsanfragen von Datafox-Terminals und weist jedem zu verbindenden Datafox-Terminal eine entsprechende Kanal-ID zu.

Über Funktionen wie *DFCGetFirstActiveChannelID* können Sie alle verfügbaren Verbindungen abrufen. Über die zugewiesene Kanal-ID können Sie das verbundene Datafox-Terminal ansprechen.

Geschlossen wird eine Aktive Verbindung normalerweise nie, außer Sie nutzen die Aktive Verbindung im Kontext eines Wartungsservers, hierbei kann über die Funktion *DFCComClose* ein schließen der Verbindung erfolgen.

Das Beenden der gesamte interne Verarbeitung, von Aktiven Verbindungen, geschieht über die Funktion *DFCStopActiveConnection*.

Notizen :



3.2.3 Datensätze über GPRS per HTTP versenden

Eine Sonderstellung nimmt das Versenden von Datensätzen per HTTP ein. Hierbei kann, wie bei der Aktiven Verbindung, über die Funktion *DFCSetGlobVar* das Versenden eingerichtet werden. Das setzen der Systemvariablen beinhaltet Informationen wie z. B. wohin die Datensätze gesendet werden sollen. Die Einrichtung kann selbstverständlich auch über das DatafoxStudioIV erfolgen.

Notizen :

4 Abrufen von Informationen über das Datafox-Terminal

Um in Ihrer Softwarelösung die Datafox-Terminals verwalten zu können, stehen Ihnen verschiedene Funktionen zur Verfügung um Informationen über das jeweils angesprochene Datafox-Terminal abzurufen.

4.1 Abfragen der Firmwareversion und des Gerätetyps

Die Abfrage der aktuell verwendeten Firmwareversion und des vorliegenden Gerätetyps geschieht über den Aufruf der Funktion *DFCGetVersionFirmware*. Hierbei erhalten Sie eine Zeichenfolge „04.01.06.30.PZE“ die angibt, dass die zum Einsatz kommende Firmwareversion 04.01.06.30 ist, und es sich um einen PZE-MasterIV handelt.

4.2 Ermitteln der Seriennummer

Die Seriennummer stellt ein wichtiges Attribut des Gerätes dar. Einen eindeutigen Schlüssel, der das Gerät kennzeichnet ist jedoch die Kombination aus Gerätetyp und Seriennummer. Die Seriennummer bezieht sich immer auf den Gerätetyp und kann somit bei verschiedenen Gerätetypen gleich sein. Abrufen können Sie die Seriennummer unter Verwendung der Funktion *DFCGetSeriennummer*.

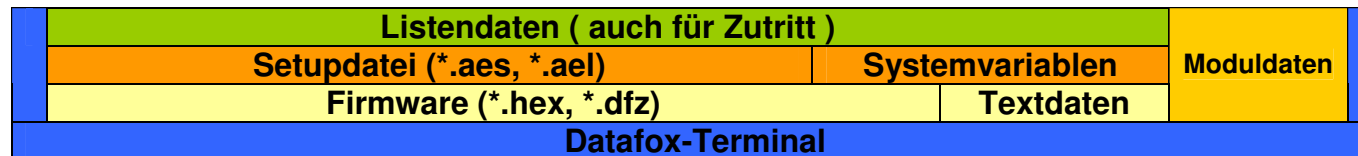
4.3 Ermitteln von Informationen zu eingebaute / angeschlossene Module

Um Informationen über eingebaute oder angeschlossene Module zu ermitteln steht Ihnen die Funktion *DFCGetInfo* zur Verfügung. Über die Funktion lassen sich Informationen wie die eingesetzte Firmwareversion des eingebauten Fingerprintmoduls oder angeschlossenen Türmoduls abrufen.

Notizen :

5 Geräteeinrichtung

Die Geräteeinrichtung besteht aus mehreren darstellbaren Schichten, die aufeinander aufbauen.



Die Firmware stellt die Anwendungsschicht dar, sie arbeitet mit der Setupdatei welche die Hauptkonfiguration des Datafox-Terminals beinhaltet. Ein kleiner Teil der Gerätekonfiguration geschieht über zusätzliche Systemvariablen. Zur Anzeige von Informationen oder Auswahl von Werten stehen die Listendaten zur Verfügung. Zusätzlich sind je nach Terminalausstattung auch Moduldaten, wie Fingertemplates bei der Biometrischen Erfassung, auf eingebauten Modulen notwendig. Alle Daten zum Einrichten des Terminals können über entsprechende Funktionen der DLL auf das Datafox-Terminal übertragen werden.

5.1 Übertragen von Firmware oder Benutzerspezifischer Textdaten (Firmware, Textdaten)

Die Firmware liegt in Dateien der Endung *.hex (einzelne Firmwaredatei) oder *.dfz (Gruppe von Firmwaredateien) vor. Ein Aufspielen der Firmware erfolgt über die Funktion *DFCUpload*. Jede Firmware enthält ihren eigenen Satz von Textdaten in deutsch, somit ist kein gesondertes Übertragen der Textdaten notwendig, wenn Sie jedoch Textanpassungen vorgenommen haben und die vorliegende Datei mit der Endung *.dfl einspielen möchten, können Sie diese ebenfalls über *DFCUpload* auf das Terminal übertragen. Textanpassungen können über das DatafoxStudioIV vorgenommen werden.

5.1.1 Übertragen von Firmware auf eingebaute Module

Über die Funktion *DFCUploadModule* können Sie in das Terminal eingebaute oder am Terminal angeschlossene Module aktualisieren. Z. b. Aktualisierung der Firmware von einem Fingerprintmodul. Hierzu ist auch die Funktion *DFCGetInfo* interessant, da durch sie bestimmte Informationen (wie aktuell eingesetzte Firmwareversion) der eingebaut oder angeschlossenen Module abgefragt werden können.

Notizen :

5.2 Abgleich der Uhrzeit

Ein Abgleich des Datums und der Uhrzeit ist in regelmäßigen Zyklen empfohlen. Ein Abgleich mit der PC-Uhr kann über einen Aufruf der Funktion *DFComSetTime* geschehen.

5.3 Übertragen vom Setup (Setupdatei)

Die Konfiguration, welche das Verhalten des Terminals auf Tastendrücke oder Signaländerungen beschreibt, liegt in einer Setupdatei mit der Endung (*.aes oder *.ael) vor. Übertragen kann man eine solche Datei durch Aufruf der Funktion *DFCSetupLaden*. Falls Sie die Aktuelle Konfiguration des Terminals auslesen möchten, können Sie dieses über die Funktion *DFCDownload*.

5.4 Setzen der Konfiguration für GSM, GPRS

Um eine SIM-Karte in Betrieb zu nehmen, muss ggf. eine PIN / PUK Angegeben werden. Diese Werte können Sie über Systemvariablen mittels *DFCSetGlobVar* setzen. Zusätzlich zu diesen Informationen sind für die Einwahl beim Provider und versenden von Datensätzen zusätzliche Systemvariablen zu konfigurieren. Die aktuell gesetzten Werte können Sie mit der Funktion *DFCGetGlobVar* abrufen.

Für eine Liste der Verfügbaren Systemvariablen sehen Sie bitte in der DLL-Dokumentation nach. Selbstverständlich können Sie die Systemvariablen auch über das DatafoxStudioIV konfigurieren.

5.5 Setzen der Konfiguration zur Aktiven Verbindung

Um dem Terminal mitzuteilen wohin es eine Verbindung aufbauen soll, werden die Werte in Systemvariablen hinterlegt. Mittels *DFCSetGlobVar* können Sie auf die Werte der Systemvariablen setzen und mit *DFCGetGlobVar* können Sie die aktuellen Werte ermitteln.

Für eine Listen der Verfügbaren Systemvariablen sehen Sie bitte in der DLL-Dokumentation nach. Selbstverständlich können Sie die Systemvariablen auch über das DatafoxStudioIV konfigurieren.

5.6 Übertragen von Listen (Listendaten, auch für Zutritt)

Listen beinhalten Daten aus denen der Anwender oder das System auswählen kann und deren Werte bei

Notizen :

Bedarf in Datensätze übernommen werden können. Um Listendaten nutzen zu können, müssen diese durch Ihre Softwarelösung importiert und auf das Terminal übertragen werden.

5.6.1 Listenstruktur ermitteln

Die Struktur der Liste, Name, Anzahl Felder und Feldnamen sowie Typen, wird im Setup festgelegt. Um auf die Strukturinformationen des im Terminal vorliegenden Setups zuzugreifen, stehen folgende Funktionen zur Verfügung.

- Auslesen der Strukturinformation in die DLL mit *DFCLoadListenbeschreibung*.
- Abruf der Anzahl definierter Listen mit *DFCListBCnt*.
- Ermitteln des Listennamens und Anzahl Felder mit *DFCListBDatensatz*.
- Die Information zu jedem einzelnen Feld, kann mit der Funktion *DFCListBFeld* abgerufen werden.

Durch Kombinierte Anwendung dieser Funktionen können die im Terminal im Einsatz befindlichen Listenstrukturen aus der DLL abgerufen werden.

5.6.2 Listenimport und Übertragung

Für jede mögliche Liste steht ein eigener Listepuffer zur Verfügung. Das importieren und Übertragen von Listendaten geschieht in drei Schritten.

- Löschen und Initialisieren aller internen Listenpuffer mit *DFCClrListenBuffer*,
- übergeben der Listendaten an den entsprechenden internen Listenpuffer mit *DFCMakeListe*,
 - die Funktion ist für jede zu übertragende Liste aufzurufen,
- übertragen der übergebenen Listendaten auf das Terminal durch *DFCLoadListen*.

5.6.3 Zutrittslisten

Für jede mögliche Liste steht ein eigener Listenspuffer zur Verfügung. Das importieren und Übertragen von Listendaten geschieht in drei Schritten.

- Löschen und initialisieren eines internen Listenpuffers mit *DFCClearEntrance2List*,
- übergeben der Listendaten an den entsprechenden internen Listenpuffer mit *DFCMakeEntrance2List*,
- übertragen der Listendaten auf das Terminal durch *DFCLoadEntrance2List*.

Notizen :

Für jede mögliche Liste steht ein eigener Listenpuffer zur Verfügung. Das importieren und Übertragen von Listendaten geschieht in drei Schritten.

- ## 5.7 Zugriff auf vorhandene Listendaten

Verfügbare Listen können über die Funktion *DFCOpenTable* geöffnet und *DFCCloseTable* wieder geschlossen werden. Es stehen weitere Funktionen zum setzen von Selektierkriterien und verschieben des Datensatzzeigers sowie lesen und setzen von Feldwerten zur Verfügung.

[illegible]

6 Abrufen der erzeugten Datensätze

Die mittels Eingabeketten erzeugten Datensätze folgen der im Setup definierten Struktur. Die Struktur und Daten können über entsprechende Funktionen der DLL abgerufen werden. Die Datensätze selbst werden immer in der chronologischen Reihenfolge geliefert.

6.1 Datensatzstruktur ermitteln

Die Struktur der Datensätze, Name, Anzahl Felder und Feldnamen sowie Typen, wird im Setup festgelegt. Um auf die Strukturinformationen des im Terminal vorliegenden Setups zuzugreifen, stehen folgende Funktionen zur Verfügung.

- Auslesen der Strukturinformation in die DLL mit *DFCLoadDatensatzbeschreibung*.
- Abruf der Anzahl definierter Listen mit *DFCDatBCnt*.
- Ermitteln des Datensatznamens und Anzahl Felder mit *DFCDatBDatensatz*.
- Die Information zu jedem einzelnen Feld, kann mit der Funktion *DFCDatBFeld* abgerufen werden.

Durch Kombinierte Anwendung dieser Funktionen können die im Terminal im Einsatz befindlichen Datensatzstrukturen aus der DLL abgerufen werden.

6.2 Daten zu einem Datensatz abrufen und bestätigen

Das Abrufen eines Datensatzes geschieht in zwei Schritten,

- Lesen der Daten mittels *DFCReadRecord*,
- bestätigen der erfolgreichen Verarbeitung mittels *DFCQuitRecord*.

Erst nach Bestätigung durch *DFCQuitRecord* kann mit *DFCReadRecord* der nächste Datensatz abgerufen werden. *DFCReadRecord* liefert solange den gleichen Datensatz bis ein erfolgreicher Aufruf von *DFCQuitRecord* erfolgte.

6.2.1 Wiederherstellen von bereits abgerufenen Datensätzen

Solange keine Setupübertragung stattgefunden hat, stehen die Erzeugten Datensätze noch zur Verfügung, solange keine Überschreibung aufgrund des zur Verfügung stehenden Speicherplatzes stattgefunden hat. Die zugewiesene Speichergröße stellen Sie über das verwendete Setup ein. Ein wiederherstellen der Datensätze ist durch ein Aufruf der Funktion *DFCRestoreRecords* möglich.

Bitte beachten Sie, dass hiernach ggf. eine größere Datensatzanzahl abgerufen wird.

Notizen :

7 Nette Hilfsfunktionen

7.1 Schalten eines Relais

Die in den Terminals eingebauten Relais können per Funktionsaufruf angesprochen werden. Sie können mittels *DFCCloseRelay* geschlossen werden. Mit *DFCOpenRelay* geöffnet, es kann über *DFCGetRelayState* auch der aktuelle Zustand des Relais abgefragt werden.

7.2 Anzeigen von Text am Display

Um Text am Terminal einblenden zu können steht die Funktion *DFCComSendMessage* zur Verfügung. Aktuelle Auftragsstände die in einer Art Memo immer abrufbar sein sollen, steht die Funktion *DFCComSendInfotext* zur Verfügung. Diese Funktion ist jedoch nur bei Geräten für die AE / BDE nutzbar.

7.3 Zutrittskontrolle

Mittels der Funktion *DFCEntrance2Identification* können Sie virtuelle Buchungen in der Zutrittskontrolle ausführen. Hierbei handelt es sich um Buchung, wobei getan wird als ob ein Anwender an einem Leser Bucht. Die Funktion kann bei einem Pförtner sinnvoll genutzt werden. Die Zutrittskontrolle kann in einem Offlinemodus betrieben werden, um hierbei die Relais und LED's der Türmodule schalten zu können steht Ihnen die Funktion *DFCEntrance2OnlineAction* zur Verfügung.

Notizen :