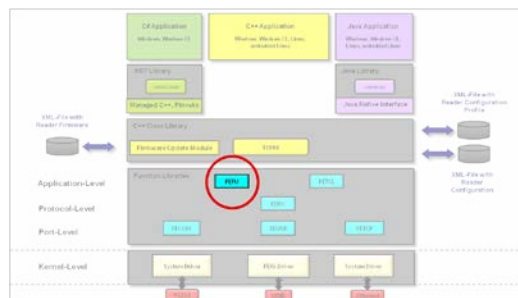


# ID FEFU

Version 2.00.00

## Software-Support für externe Funktionseinheiten (Function Units)



Betriebssystem	Ausführung		Anmerkungen
	32-Bit	64-Bit	
Windows XP	X	(X)	bei 64-Bit nur mit 32-Bit Laufzeitsystem
Windows Vista / 7 / 8	X	X	
Windows CE	X	-	
Linux	X	X	
Apple Max OS X	-	-	auf Anfrage

## Hinweis

© Copyright 2003-2014 by  
FEIG ELECTRONIC GmbH  
Lange Straße 4  
D-35781 Weilburg-Waldhausen  
Tel.: +49 6471 3109-0  
<http://www.feig.de>

Alle früheren Ausgaben verlieren mit dieser Ausgabe ihre Gültigkeit.  
Die Angaben in diesem Dokument können ohne vorherige Ankündigung geändert werden.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlung verpflichtet zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmuster-Eintragung vorbehalten.

Die Zusammenstellung der Informationen in diesem Dokument erfolgt nach bestem Wissen und Gewissen. FEIG ELECTRONIC GmbH übernimmt keine Gewährleistung für die Richtigkeit und Vollständigkeit der Angaben in diesem Dokument. Insbesondere kann FEIG ELECTRONIC GmbH nicht für Folgeschäden auf Grund fehlerhafter oder unvollständiger Angaben haftbar gemacht werden. Da sich Fehler, trotz aller Bemühungen nie vollständig vermeiden lassen, sind wir für Hinweise jederzeit dankbar.

Die in diesem Dokument gemachten Installationsempfehlungen gehen von günstigsten Rahmenbedingungen aus. FEIG ELECTRONIC GmbH übernimmt keine Gewähr für die einwandfreie Funktion in systemfremden Umgebungen.

FEIG ELECTRONIC GmbH übernimmt keine Gewährleistung dafür, dass die in diesem Dokument enthaltenden Informationen frei von fremden Schutzrechten sind. FEIG ELECTRONIC GmbH erteilt mit diesem Dokument keine Lizenzen auf eigene oder fremde Patente oder andere Schutzrechte.

OBID® and OBID i-scan® are registered trademarks of FEIG ELECTRONIC GmbH.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Windows Vista is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries

Linux® is a registered Trademark of Linus Torvalds.

## Allgemeine Hinweise zu diesem Dokument

- Die verwendeten Zahlenformate sind
  - 0...9: für dezimale Zahlenwerte und
  - 0x00...0xFF: für hexadezimale Zahlenwerte,
  - b0...1 für binäre Zahlenwerte.
- Hexadezimale Werte in eckigen Klammern [ ] kennzeichnen Kommandos (Steuerbefehle).

## Lizenzvertrag über die Nutzung der Software

Dies ist ein Vertrag zwischen Ihnen und der FEIG ELECTRONIC GmbH (nachfolgend "FEIG") über die Nutzung der überlassenen Programmbibliothek ID FEFU und die vorliegende Dokumentation, nachfolgend Lizenzmaterial genannt. Mit der Installation und Benutzung der Software erklären Sie sich mit allen Bestimmungen dieses Vertrages ausnahmslos und ohne Einschränkung einverstanden. Wenn Sie mit den Bestimmungen dieses Vertrages nicht oder nicht vollständig einverstanden sind, dürfen Sie das Lizenzmaterial nicht installieren oder anderweitig benutzen. Das überlassene Lizenzmaterial ist Eigentum der FEIG ELECTRONIC GmbH und ist international urheberrechtlich geschützt.

### §1 Vertragsgegenstand und Vertragsumfang

1. FEIG gewährt Ihnen das Recht, das überlassene Lizenzmaterial zu installieren und zu den nachstehenden Bedingungen zu nutzen.
2. Sie dürfen sämtliche Bestandteile des Lizenzmaterials auf einer Festplatte oder einem sonstigen Speichermedium installieren. Die Installation und Nutzung darf auch auf einem Netzwerk-Fileserver erfolgen. Sie dürfen Sicherheitskopien des Lizenzmaterials anfertigen.
3. FEIG gewährt Ihnen das Recht die dokumentierte Programmbibliothek ID FEFU für die Entwicklung eigener Anwendungsprogramme oder Programmbibliotheken zu verwenden und Sie dürfen die Laufzeitdatei FEFU.DLL, FEFUCE.DLL oder LIBFEFU.SO.x.y.z<sup>1</sup> ohne Abgabe von Lizenzgebühren vertreiben, unter der Voraussetzung, daß diese Anwendungsprogramme oder Programmbibliotheken nur zusammen mit von FEIG entwickelten Geräten verwendet werden.

### §2 Schutz des Lizenzmaterials

1. Das Lizenzmaterial ist geistiges Eigentum von FEIG und seinen Lieferanten. Es ist gemäß Urheberrecht, internationalen Verträgen und einschlägigen Gesetzen des Landes geschützt, in dem sie genutzt wird. Struktur, Organisation und Code der Software sind wertvolles Geschäftsgeheimnis und vertrauliche Information von FEIG und seinen Lieferanten.
2. Sie verpflichten sich, die Programmbibliothek sowie die Dokumentation nicht zu ändern, anzupassen, zu übersetzen, rückzuentwickeln, zu dekompileieren, zu disassemblieren oder auf andere Weise zu versuchen, den Quellcode dieser Software herauszufinden.
3. Soweit FEIG im Lizenzmaterial Schutzvermerke, wie Copyright-Vermerke und andere Rechtsvorbehalte angebracht hat, sind Sie verpflichtet, diese unverändert beizubehalten sowie in alle von Ihnen hergestellten vollständigen oder teilweisen Kopien in unveränderter Form zu übernehmen.
4. Die Weitergabe von Lizenzmaterial ist weder vollständig noch auszugsweise gestattet, solange dazu keine explizite anderslautende Vereinbarung zwischen Ihnen und FEIG getroffen wurde. Nicht betroffen von dieser Regelung sind solche Anwendungsprogramme oder Programmbibliotheken, die gem. §1 Absatz 3. dieser Vereinbarung erstellt und vertrieben werden.

### §3 Gewährleistung und Haftungsbeschränkungen

1. Sie stimmen mit FEIG darüber überein, dass es nicht möglich ist, EDV-Programme so zu entwickeln, dass sie für alle Anwendungsbedingungen fehlerfrei sind. FEIG weist Sie ausdrücklich darauf hin, dass die Installation eines neuen Programms bereits vorhandene Software beeinflussen kann, und zwar auch solche Software, die nicht gleichzeitig mit der neuen Software ausgeführt wird. FEIG haftet in keinem Fall für direkte oder indirekte Schäden, für Folgeschäden oder Sonderschäden, Einschließlich entgangenen Geschäftsgewinn oder entgangener Einsparungen. Wenn Sie sicherstellen wollen, dass es zu keinerlei Beeinflussung eines bereits installierten Programms kommt, dürfen Sie die vorliegende Software nicht installieren.
2. FEIG weist ausdrücklich darauf hin, dass mit der Software irreversible Einstellungen und Anpassungen an Geräten vorgenommen werden können, wodurch diese Geräte zerstört oder unbrauchbar gemacht werden können. FEIG übernimmt für derartiges Handeln unabhängig davon ob dies bewusst oder unbewusst erfolgte keinerlei Gewährleistung.
3. FEIG liefert Ihnen die Software "wie besehen" ohne jegliche Gewährleistung. FEIG kann für die Leistung oder die Ergebnisse, die Sie durch die Nutzung der Software erzielen, nicht garantieren. FEIG übernimmt keine Gewährleistung oder Garantie dafür, dass keine Schutzrechte Dritter verletzt werden, auch nicht dafür, dass die Software für irgendeinen bestimmten Zweck geeignet ist.
4. FEIG weist ausdrücklich darauf hin, dass das Lizenzmaterial nicht für den Einsatz mit oder in medizinischen Geräten oder für Geräte für lebenserhaltende Maßnahmen konzipiert ist, bei denen ein Fehler eine Gefahr für menschliches Leben oder für die gesundheitliche Unversehrtheit zur Folge haben kann.

---

<sup>1</sup> x.y.z repräsentiert die Versionsnummer

Der Anwender des Lizenzmaterials ist dafür verantwortlich, geeignete Maßnahmen zu ergreifen um Gefahren, Schäden oder Verletzungen zu vermeiden.

#### **§4 Schlussbestimmungen**

1. Dieser Vertrag enthält die vollständigen Lizenzbestimmungen und ersetzt alle eventuell vorangegangenen Regelungen und Absprachen. Änderungen und Ergänzungen bedürfen der Schriftform.
2. Sollte eine der in diesem Vertrag enthaltenen Bestimmungen unwirksam sein oder werden, so wird die Gültigkeit der übrigen Bestimmungen hierdurch nicht berührt. Beide Vertragsparteien verpflichten sich, die unwirksame Bestimmung durch eine solche wirksame Bestimmung zu ersetzen, die dem wirtschaftlichem Zweck der zu ersetzenden Bestimmung am nächsten kommt.
3. Dieser Vertrag unterliegt dem Recht der Bundesrepublik Deutschland. Gerichtsstand ist Frankfurt a. M.

## Inhalt

<b>Inhalt .....</b>	<b>5</b>
<b>1. Einleitung .....</b>	<b>7</b>
<b>1.1. Lieferumfang.....</b>	<b>8</b>
1.1.1. Windows XP / Vista / 7 / 8 .....	8
1.1.2. Windows CE .....	8
1.1.3. Linux .....	8
<b>2. Änderungen gegenüber der Vorversion .....</b>	<b>9</b>
<b>3. Installation.....</b>	<b>10</b>
3.1. 32- und 64-Bit Windows XP/Vista/7/8 .....	10
3.2. Windows CE.....	11
3.3. 32- und 64-Bit Linux .....	12
<b>4. Einbindung in das Anwendungsprogramm.....</b>	<b>13</b>
4.1. Unterstützte Entwicklungsumgebungen.....	13
4.2. Einbindung in Visual Studio .....	13
<b>5. Programmierschnittstelle.....</b>	<b>14</b>
5.1. Übersicht.....	14
5.2. Threadsicherheit.....	15
5.3. Liste der Funktionen .....	16
5.3.1. FEFU_GetDLLVersion .....	17
5.3.2. FEFU_GetErrorText.....	17
5.3.3. FEFU_GetStatusText.....	18
5.3.4. FEFU_GetLastState.....	18
5.3.5. FEFU_GetLastError .....	19
5.3.6. FEFU_MUX_SoftVersion .....	20
5.3.7. FEFU_MUX_CPUReset.....	20

5.3.8. FEFU_MUX_SelectChannel.....	21
5.3.9. FEFU_MUX_Detect .....	22
5.3.10. FEFU_DAT_SoftVersion .....	23
5.3.11. FEFU_DAT_CPUReset.....	23
5.3.12. FEFU_DAT_Detect .....	24
5.3.13. FEFU_DAT_GetValues .....	24
5.3.14. FEFU_DAT_SetOutput .....	25
5.3.15. FEFU_DAT_StartTuning .....	25
5.3.16. FEFU_DAT_ReTuning .....	26
5.3.17. FEFU_DAT_StoreSettings .....	26
5.3.18. FEFU_DAT_SetAddress .....	27
5.3.19. FEFU_DAT_SetMode .....	27
5.3.20. FEFU_UMUX_SoftVersion.....	28
5.3.21. FEFU_UMUX_CPUReset .....	28
5.3.22. FEFU_UMUX_SelectChannel .....	29
5.3.23. FEFU_UMUX_Detect_GetPower .....	30
<b>ANHANG .....</b>	<b>31</b>
<b>5.4. Fehlercodes .....</b>	<b>31</b>
<b>5.5. Historie.....</b>	<b>32</b>

## 1. Einleitung

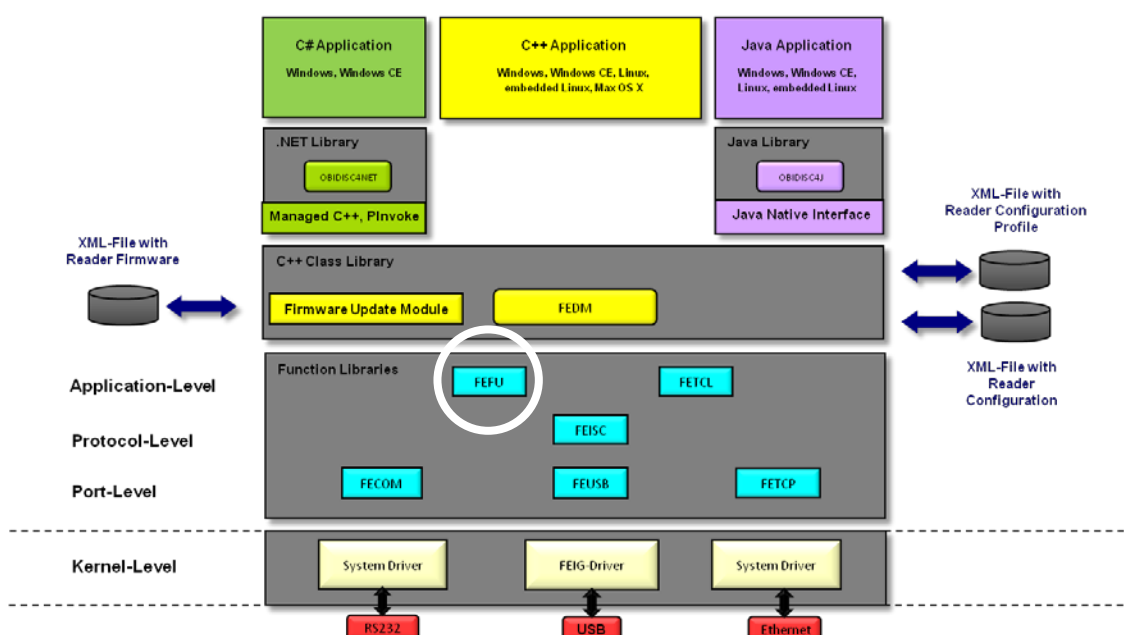
Das Supportpaket ID FEFU dient zur Unterstützung bei der Programmierung von Anwendungs-Software, die OBID i-scan®-Leser und externe Funktionseinheiten (Function Units) wie z. B. den Multiplexer ID ISC.ANT.MUX integrieren. Es unterstützt die Sprachen ANSI-C, ANSI-C++ und prinzipiell jede andere Sprache, die C-Funktionen aufrufen kann.

Mit dem Supportpaket wird eine einfache Funktionsschnittstelle zu OBID i-scan®-Funktionseinheiten angeboten. Für jedes im Systemhandbuch zur OBID i-scan®-Funktionseinheit dokumentierte Protokoll existiert eine eigene Funktion. Diese Kommandos werden über den OBID i-scan®-Leser an die Funktionseinheit gesendet. Für die Kommunikation mit dem OBID i-scan®-Leser wird das Supportpaket ID FEISC vorausgesetzt.

Verwendet werden kann die Bibliothek mit folgenden Betriebssystemen:

Betriebssystem	Ausführung		Anmerkungen
	32-Bit	64-Bit	
Windows XP	X	(X)	bei 64-Bit nur mit 32-Bit Laufzeitsystem
Windows Vista	X	X	
Windows 7	X	X	
Windows CE	X	-	
Linux	X	X	
Apple Max OS X	-	-	auf Anfrage

Die Bibliothek FEFU bildet die dritte Ebene in dem mehrschichtigen, hierarchisch strukturierten Aufbau von FEIG-Bibliotheken. Mit ihr werden Funktionseinheiten verwaltet und die Kommunikation mit diesen ermöglicht. Das nachfolgende Bild zeigt eine Übersicht über alle Bibliotheken.



---

## 1.1. Lieferumfang

---

Dieses Supportpaket besteht aus den nachfolgend aufgelisteten Dateien. In der Regel wird das Paket mit anderen Bibliotheken in einem speziell für das jeweilige Betriebssystem zusammengestellten Software Development Kit (SDK) – z.B. ID ISC.SDK.Win - ausgeliefert.

---

### 1.1.1. Windows XP / Vista / 7 / 8

---

Datei	Verwendung
FEFU.DLL	DLL mit allen Funktionen
FEFU.LIB	LIB-Datei zum Linken für C/C++-Projekte
FEFU.H	Header-Datei für C/C++-Projekte

---

### 1.1.2. Windows CE

---

Datei	Verwendung
FEFUCE.DLL	DLL mit allen Funktionen
FEFUCE.LIB	LIB-Datei zum Linken für C/C++-Projekte
FEFU.H	Header-Datei für C/C++-Projekte

---

### 1.1.3. Linux

---

Datei	Verwendung
LIBFEFU.SO.x.y.z <sup>2</sup>	Funktions-Bibliothek mit allen Funktionen
FEFU.H	Header-Datei für C/C++-Projekte

---

<sup>2</sup> x.y.z repräsentiert die Versionsnummer der Bibliotheksdatei



---

## 2. Änderungen gegenüber der Vorversion

---

- Windows:
  1. Migration der Entwicklungsumgebung von Visual Studio 2008 zu Visual Studio 2010.
  2. DLL jetzt ohne MFC
  3. Erstes Release der 64-Bit Version
  
- Linux:
  - Version für 64-Bit
  
- Funktional ohne Änderungen

Bitte beachten Sie auch die Änderungshistorie im Anhang.

---

### 3. Installation

---

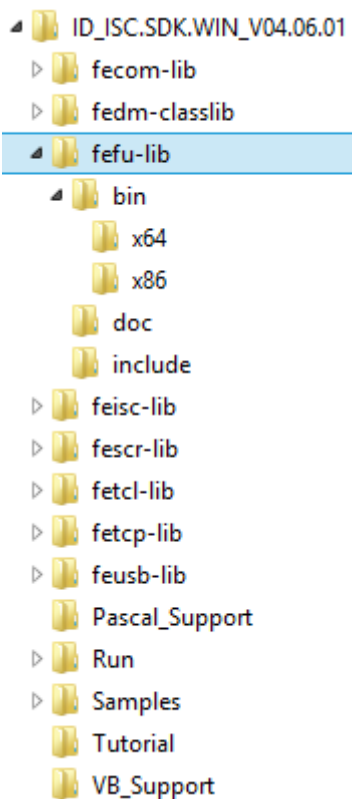
Das Supportpaket wird in der Regel mit einem Software Development Kit (SDK) ausgeliefert. Kopieren Sie das SDK in ein Verzeichnis Ihrer Wahl.

Die Dateien dieses Supportpakets finden sich im Verzeichnis fefu-lib.

---

#### 3.1. 32- und 64-Bit Windows XP/Vista/7/8

---



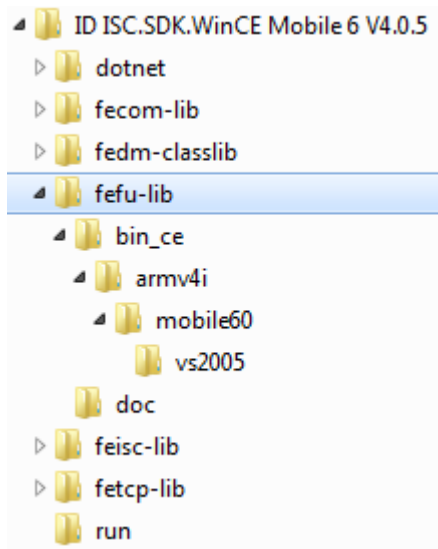
Wenn eigene Projekte nicht im SDK-Verzeichnis angelegt werden sollen, dann empfiehlt sich folgende Vorgehensweise:

- Kopieren Sie FEFU.DLL in das Verzeichnis des Anwendungsprogramms (empfohlen) oder in das Systemverzeichnis von Windows.
- Kopieren Sie FEFU.LIB in das Projekt- oder LIB-Verzeichnis
- Kopieren Sie FEFU.H in das Projekt- oder INCLUDE-Verzeichnis

---

## 3.2. Windows CE

---



Wenn eigene Projekte nicht im SDK-Verzeichnis angelegt werden sollen, dann empfiehlt sich folgende Vorgehensweise:

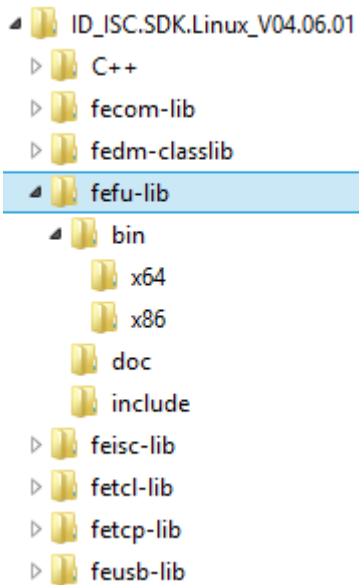
- Kopieren Sie die Datei FEFUCE.DLL in das Anwendungs- oder Systemverzeichnis des Windows CE Rechners.
- Kopieren Sie FEFUCE.LIB in das Projekt- oder LIB-Verzeichnis.
- Kopieren Sie FEFU.H in das Projekt- oder INCLUDE-Verzeichnis

Hinweis: die DLL kann nicht mit eMbedded Visual Basic 3.0 verwendet werden.

---

### 3.3. 32- und 64-Bit Linux

---



Zur Installation gibt es zwei Optionen:

Option 1: Falls eine `install.sh` im SDK-Verzeichnis vorliegt, führen Sie diese aus. Damit werden alle Bibliotheken in das Verzeichnis `/usr/lib` bzw. `/usr/lib64` kopiert und alle symbolischen Links angelegt. Die Headerdatei können Sie in ein Verzeichnis Ihrer Wahl kopieren.

Option 2: Kopieren Sie die Dateien dieses Supportpakets in Verzeichnisse Ihrer Wahl und Erzeugen Sie symbolische Links auf die Bibliotheksdatei `libfefu.so.x.y.z`<sup>3</sup> im Verzeichnis `/usr/lib` bzw. `/usr/lib64` durch folgende Aufrufe:

```
cd /usr/lib (für 64 Bit : /usr/lib64)
```

```
ln -s /<Verzeichnis>/libfefu.so.x.y.z libfefu.so.x
```

```
ln -s /<Verzeichnis>/libfefu.so.x libfefu.so
```

```
ldconfig
```

#### Anmerkung:

x86 : Die Bibliothek wurde unter SuSE Linux 11.1 mit der GNU Compiler Collection V4.3.2 erstellt.

X64: Die Bibliothek wurde unter SuSE Linux 11.2 mit der GNU Compiler Collection V4.4.1 erstellt.

---

<sup>3</sup> *x.y.z repräsentiert die Versionsnummer der Bibliotheksdatei*

---

## 4. Einbindung in das Anwendungsprogramm

---

---

### 4.1. Unterstützte Entwicklungsumgebungen

---

Betriebssystem	Entwicklungsumgebung	Unterstützung
Windows XP / Vista / 7 / 8	Visual Studio	ja
	Borland C++ Builder	ja
	Embarcadero C++ Builder	ja
Windows CE	eMbedded Visual C++ 4	ja
	Visual Studio 2005 / 2008	ja
Linux	GCC	ja
Mac OS X	GCC	ja, für Projekte mit x86_64 Architektur
	Xcode ≥ V4.3.2	ja, für Projekte mit x86_64 Architektur

---

### 4.2. Einbindung in Visual Studio

---

1. Include-Pfad zur Headerdatei in den Projekteinstellungen (Kategorie C/C++) hinzufügen
2. die LIB-Datei in den Projekteinstellungen (Kategorie Linker) eintragen

ID FEISC und eine der Kommunikationsbibliotheken (FECOM, FEUSB, FETCP) muss ebenfalls in Ihr Projekt eingebunden werden.

## 5. Programmierschnittstelle

### 5.1. Übersicht

Die FEFU-DLL kapselt für den Programmierer alle notwendigen Funktionen zur einfachen Kommunikation mit externen Funktionseinheiten, die über die Leser der OBID i-scan®-Familie angesprochen werden. Abb. 1 zeigt die Kommunikationskette innerhalb und außerhalb des Host.

Zur Ansteuerung der Funktionseinheit benötigt man den Handle eines Reader-Objekt aus der FEISC Bibliothek (s. FEISC\_NewReader) und die Kenntnis der Busadresse des Lesers. Das Reader-Objekt innerhalb der FEISC benötigt wiederum den Handle eines Porttreibers (z.B. von FECOM).

Innerhalb der FEFU.DLL werden keine Objekte erzeugt. Die Bibliothek ist als reine Funktionsammlung angelegt.

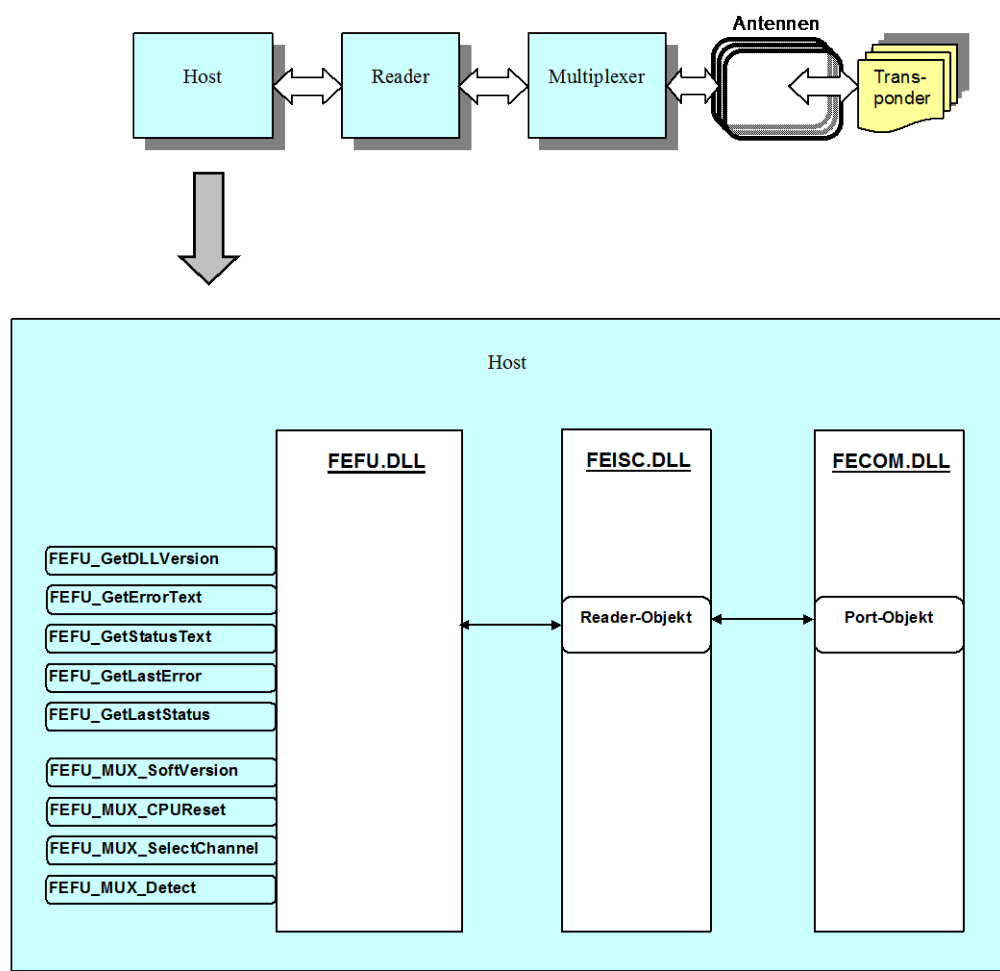


Abbildung 1: Kommunikationskette für FEFU.DLL am Beispiel der seriellen Schnittstelle

---

## 5.2. Threadsicherheit

---

Alle FEIG-Bibliotheken sind prinzipiell nicht vollständig threadsicher. Unter Beachtung einiger Regeln kann man dennoch Parallelität in der Ausführung von Kommunikationsaufgaben und damit praktische Threadsicherheit erreichen. Man muss auch wissen, dass alle OBID® RFID-Leser immer nur eine Aktion ausführen können, also synchron arbeiten.

Auf der Ebene der Transportschicht (FECOM, FEUSB, FETCP) kann über jede Verbindung nur synchron kommuniziert werden, weil auch die OBID-Leser nur synchron arbeiten. Threadsicher sind die Port-Objekte untereinander, weil diese unabhängig voneinander sind. Es ist demnach möglich, dass z. B. zwei Threads mit zwei OBID-Lesern über zwei verschiedene TCP-Verbindungen kommunizieren.

Auf der Ebene der Protokollschicht (FEISC) ist Parallelität über separate Leser-Objekte möglich, wenn jedes Leser-Objekt mit einer eigenen Kommunikations-Schnittstelle verbunden ist. Eine Ausnahme gilt für die vier speziellen Funktionen FEISC\_BuildxxProtocol, FEISC\_SplitxxProtocol, die einen globalen Puffer für Protokolldaten nutzen.

Die Bibliothek FEFU für Externe Funktionseinheiten bietet keine Threadsicherheit. Über sie kann immer nur ein Thread kommunizieren und die Threadsicherheit muss in der Applikation hergestellt werden.

---

### 5.3. Liste der Funktionen

---

In dem Support-Paket sind Funktionen für unterschiedliche Aufgabestellungen enthalten. Zur besseren Orientierung sind sie in Gruppen aufgeteilt.

#### Allgemeine Funktionen

- **void FEFU\_GetDLLVersion( char\* cVersion )**
- **int FEFU\_GetErrorText( int iErrorCode, char\* cErrorText )**
- **int FEFU\_GetStatusText( UCHAR ucStatus, char\* cStatusText )**
- **int FEFU\_GetLastState( int iReaderHnd, char\* cStatusText )**
- **int FEFU\_GetLastError( int iReaderHnd , int\* iErrorCode, char\* cErrorText )**

#### Spezielle Kommunikations-Funktionen für HF-Multiplexer

- **int FEFU\_MUX\_CPUReset ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr )**
- **int FEFU\_MUX\_SoftVersion ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR\* ucVersion )**
- **int FEFU\_MUX\_SelectChannel ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucIn1, UCHAR ucIn2 )**
- **int FEFU\_MUX\_Detect ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr )**

#### Spezielle Kommunikations-Funktionen für Dynamischen Antennentuner (HF)

- **int FEFU\_DAT\_CPUReset ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )**
- **int FEFU\_DAT\_SoftVersion ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR\* ucVersion )**
- **int FEFU\_DAT\_Detect ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )**
- **int FEFU\_DAT\_GetValues ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR\* ucValues )**
- **int FEFU\_DAT\_SetOutput( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR ucOut )**
- **int FEFU\_DAT\_StartTuning( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )**
- **int FEFU\_DAT\_ReTuning( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )**
- **int FEFU\_DAT\_StoreSettings( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )**
- **int FEFU\_DAT\_SetAddress( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR ucNewDatAdr )**
- **int FEFU\_DAT\_SetMode ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR ucMode )**

#### Spezielle Kommunikations-Funktionen für UHF-Multiplexer

- **int FEFU\_UMUX\_CPUReset ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags )**
- **int FEFU\_UMUX\_SoftVersion ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags, UCHAR\* ucVersion )**
- **int FEFU\_UMUX\_SelectChannel ( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags, UCHAR ucChannelNo )**
- **int FEFU\_UMUX\_Detect\_GetPower( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags, UCHAR\* ucData )**



---

5.3.1. FEFU\_GetDLLVersion

---

<b>Funktion</b>	Ermittelt die Versionsnummer der DLL.
<b>Syntax</b>	<b>void FEFU_GetDLLVersion( char* cVersion )</b>
<b>Beschreibung</b>	<p>Die Funktion gibt die Versionsnummer der DLL zurück.</p> <p><i>cVersion</i> ist eine leere, nullterminierte Zeichenkette zur Rückgabe der Versionsnummer. Die Zeichenkette sollte wenigstens 256 Zeichen aufnehmen können.</p> <p>In der Zeichenkette wird aktuelle Versionsnummer zurückgegeben (z.B. "02.00.00"). Neuere Versionen könnten aber weitere Informationen liefern.</p>
<b>Rückgabewert</b>	ohne
<b>Beispiel</b>	<pre>... #include "fefu.h" ... char cVersion[256]; FEFU_GetDLLVersion( cVersion );     // hier Code zum Anzeigen der Versionsnummer ... </pre>

---

5.3.2. FEFU\_GetErrorText

---

<b>Funktion</b>	Ermittelt Fehlertext zum Fehlercode
<b>Syntax</b>	<b>int FEFU_GetErrorText( int iErrorCode, char* cErrorText )</b>
<b>Beschreibung</b>	<p>Die Funktion übergibt in <i>cErrorText</i> den zum <i>iErrorCode</i> zugehörigen englischen Fehlertext.</p> <p>Der Puffer für <i>cErrorText</i> sollte 256 Zeichen aufnehmen können.</p>
<b>Rückgabewert</b>	Im fehlerfreien Fall liefert die Funktion Null und im Fehlerfall einen Wert kleiner als Null zurück. Die Liste der Fehlercodes findet sich im Anhang.
<b>Beispiel</b>	<pre>... #include "fefu.h" ... char cErrorText[256];  int iBack = FEFU_GetErrorText(FEFU_ERR_TIMEOUT, cErrorText )     // hier Code zum Anzeigen des Textes ... </pre>

---

5.3.3. FEFU\_GetStatusText

---

<b>Funktion</b>	Ermittelt Kurztext zum Statusbyte
<b>Syntax</b>	<b>int FEFU_GetStatusText( UCHAR ucStatus, char* cStatusText )</b>
<b>Beschreibung</b>	Die Funktion übergibt in <i>cStatusText</i> den zum <i>ucStatus</i> zugehörigen englischen Kurztext.  Der Puffer für <i>cStatusText</i> sollte 128 Zeichen aufnehmen können.
<b>Rückgabewert</b>	Im fehlerfreien Fall liefert die Funktion Null und im Fehlerfall einen Wert kleiner als Null zurück. Die Liste der Fehlercodes findet sich im Anhang.
<b>Beispiel</b>	<pre>... #include "fefu.h" ... char cStatusText[128]; ...  int iBack = FEFU_GetStatusText(0x11, cStatusText )            // hier Code zum Anzeigen des Textes ... ...</pre>

---

5.3.4. FEFU\_GetLastState

---

<b>Funktion</b>	Das im letzten Empfangsprotokoll enthaltene Statusbyte wird zurückgegeben.
<b>Syntax</b>	<b>int FEFU_GetLastStatus( char* cStatusText )</b>
<b>Beschreibung</b>	Mit dieser Funktion kann man das Statusbyte und einen Kurztext zum Statusbyte des letzten Empfangsprotokolls ermitteln.  Der Puffer für den Kurztext <i>cStateText</i> sollte 256 Zeichen aufnehmen können.
<b>Rückgabewert</b>	Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte.  Die Liste der Fehlercodes findet sich im Anhang.

---

5.3.5. FEFU\_GetLastError

---

<b>Funktion</b>	Ermittelt den letzten Fehlercode und übergibt Fehlertext
<b>Syntax</b>	<b>int FEFU_GetLastError( int* iErrorCode, char* cErrorText )</b>
<b>Beschreibung</b>	<p>Die Funktion übergibt in <i>iErrorCode</i> den letzten Fehlercode und in <i>cErrorText</i> den zugehörigen englischen Fehlertext zurück.</p> <p>Der Puffer für <i>cErrorText</i> sollte 256 Zeichen aufnehmen können.</p>
<b>Rückgabewert</b>	Im fehlerfreien Fall liefert die Funktion Null und im Fehlerfall einen Wert kleiner als Null zurück. Die Liste der Fehlercodes findet sich im Anhang.
<b>Beispiel</b>	<pre>... #include "fefu.h" ... char cErrorText[256]; int iErrorCode = 0; ...  int iBack = FEFU_GetLastError( &amp;iErrorCode, cErrorText )     // hier Code zum Anzeigen des Textes ... ...</pre>

---

5.3.6. FEFU\_MUX\_SoftVersion

---

<b>Funktion</b>	Funktion liest die Versionsnummer des HF-Multiplexers aus.
<b>Syntax</b>	<b>int FEFU_MUX_SoftVersion( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR* ucVersion )</b>
<b>Beschreibung</b>	<p>Die Versionsnummer des Multiplexers wird ermittelt und in <i>ucVersion</i> hinterlegt.</p> <p>Der Puffer für die Version muß mindestens 8 Byte aufnehmen können. Dabei ist 1 Byte für das NUL-Zeichen vorgesehen.</p> <p><i>ucMuxAdr</i> ist die per DIP-Schalter eingestellte Kaskadierungsebene.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Querverweis</b>	Die Struktur von <i>ucVersion</i> ist im Systemhandbuch H30701-xx-ID-B zum Multiplexer dokumentiert.
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

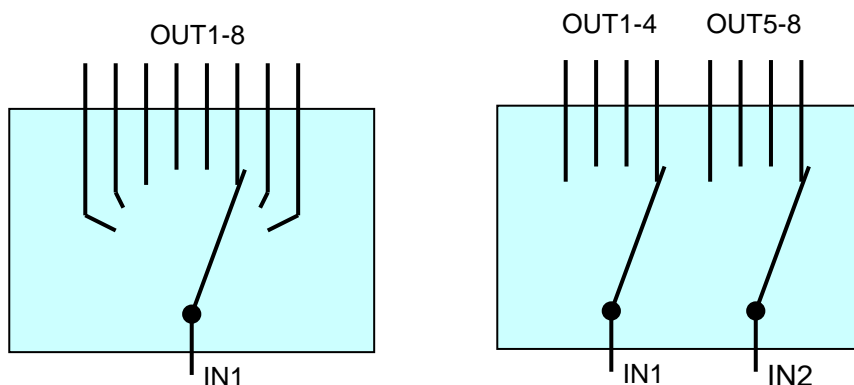
5.3.7. FEFU\_MUX\_CPUReset

---

<b>Funktion</b>	Funktion löst einen Reset in der CPU des HF-Multiplexers aus.
<b>Syntax</b>	<b>int FEFU_MUX_CPUReset( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr )</b>
<b>Beschreibung</b>	<p>Funktion löst einen Reset in der CPU des Multiplexers aus.</p> <p><i>ucMuxAdr</i> ist die per DIP-Schalter eingestellte Kaskadierungsebene.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

## 5.3.8. FEFU\_MUX\_SelectChannel

<b>Funktion</b>	Funktion setzt Ausgänge des HF-Multiplexers.
<b>Syntax</b>	<code>int FEFU_MUX_SelectChannel( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucIn1, UCHAR ucIn2 )</code>



<b>Beschreibung</b>	<p>Die Funktion schaltet für jeden Eingang ucIn1 und ucIn2 einen Ausgang des Multiplexers. In ucIn1 (ucIn2) übergibt man die Nummer des Ausgangs, mit dem der Eingang 1 (2) verbunden wird. Wird der Eingang 2 nicht benötigt, setzt man seinen Wert ucIn2 auf 0.</p> <p><i>ucMuxAdr</i> ist die per DIP-Schalter eingestellte Kaskadierungsebene.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.9. FEFU\_MUX\_Detect

---

<b>Funktion</b>	Funktion detektiert einen HF-Multiplexer.
<b>Syntax</b>	<b>int FEFU_MUX_Detect( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr )</b>
<b>Beschreibung</b>	<p>Funktion detektiert einen Multiplexer.</p> <p><i>ucMuxAdr</i> ist die Kaskadierungsebene, in der detektiert werden soll.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.10. FEFU\_DAT\_SoftVersion

---

<b>Funktion</b>	Funktion liest die Versionsnummer des Dynamischen Antennentuners aus.
<b>Syntax</b>	<b>int FEFU_MUX_SoftVersion( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR* ucVersion )</b>
<b>Beschreibung</b>	<p>Die Versionsnummer des Dynamischen Antennentuners wird ermittelt und in <i>ucVersion</i> hinterlegt.</p> <p>Der Puffer für die Version muß mindestens 8 Byte aufnehmen können. Dabei ist 1 Byte für das NUL-Zeichen vorgesehen.</p> <p><i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Querverweis</b>	Die Struktur von <i>ucVersion</i> ist im Systemhandbuch H30701-xx-ID-B zum Dynamischen Antennentuners dokumentiert.
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.11. FEFU\_DAT\_CPUReset

---

<b>Funktion</b>	Funktion löst einen Reset in der CPU des Dynamischen Antennentuners aus.
<b>Syntax</b>	<b>int FEFU_DAT_CPUReset( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )</b>
<b>Beschreibung</b>	<p>Funktion löst einen Reset in der CPU des Dynamischen Antennentuners aus.</p> <p><i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.12. FEFU\_DAT\_Detect

---

<b>Funktion</b>	Funktion detektiert einen Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_Detect( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )</b>
<b>Beschreibung</b>	<p>Funktion detektiert einen Dynamischen Antennentuner.</p> <p><i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.13. FEFU\_DAT\_GetValues

---

<b>Funktion</b>	Funktion liest Abgleichwerte von einem Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_GetValues( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR* ucValues )</b>
<b>Beschreibung</b>	<p>Funktion liest die Abgleichwerte von einem Dynamischen Antennentuner aus und speichert die Werte in <i>ucValues</i>.</p> <p>Der Puffer für die Abgleichwerte muß mindestens 7 Byte aufnehmen können. Dabei ist 1 Byte für das NUL-Zeichen vorgesehen.</p> <p><i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Querverweis</b>	Die Struktur von <i>ucValues</i> ist im Systemhandbuch H30701-xx-ID-B zum Dynamischen Antennentuners dokumentiert.
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>



---

5.3.14. FEFU\_DAT\_SetOutput

---

<b>Funktion</b>	Funktion schaltet Ausgänge in einem Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_SetOutput( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR ucOut )</b>
<b>Beschreibung</b>	Funktion schaltet Ausgänge in einem Dynamischen Antennentuner. <i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse. <i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek. <i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.
<b>Rückgabewert</b>	Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls. Die Liste der Fehlercodes findet sich im Anhang.

---

5.3.15. FEFU\_DAT\_StartTuning

---

<b>Funktion</b>	Funktion startet den Abgleichprozess in einem Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_StartTuning( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )</b>
<b>Beschreibung</b>	Die Funktion startet den Abgleichprozess in einem Dynamischen Antennentuner. Der Tuner sendet erst die Antwort zurück und startet anschließend den Abgleichprozess. Dieser Prozess kann länger andauern. Während dieser Zeit ist der Tuner nicht ansprechbar. <i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse. <i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek. <i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.
<b>Rückgabewert</b>	Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls. Die Liste der Fehlercodes findet sich im Anhang.

---

5.3.16. FEFU\_DAT\_ReTuning

---

<b>Funktion</b>	Funktion initiiert einen Feinabgleich in einem Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_ReTuning( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )</b>
<b>Beschreibung</b>	Funktion initiiert einen Feinabgleich in einem Dynamischen Antennentuner. <i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse. <i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek. <i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.
<b>Rückgabewert</b>	Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls. Die Liste der Fehlercodes findet sich im Anhang.

---

5.3.17. FEFU\_DAT\_StoreSettings

---

<b>Funktion</b>	Funktion speichert die Abgleichwerte eines Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_StoreSettings( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )</b>
<b>Beschreibung</b>	Die Funktion speichert die Abgleichwerte eines Dynamischen Antennentuner in dessen EEPROM. <i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse. <i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek. <i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.
<b>Rückgabewert</b>	Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls. Die Liste der Fehlercodes findet sich im Anhang.

---

5.3.18. FEFU\_DAT\_SetAddress

---

<b>Funktion</b>	Funktion stellt eine Adresse in einem Dynamischen Antennentuner ein.
<b>Syntax</b>	<b>int FEFU_DAT_SetAddress( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr )</b>
<b>Beschreibung</b>	<p>Funktion stellt eine Adresse in einem Dynamischen Antennentuner ein.</p> <p><i>ucNewDatAdr</i> ist die neue Tuner Adresse.</p> <p><i>ucDatAdr</i> ist die aktuelle Tuner Adresse.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.19. FEFU\_DAT\_SetMode

---

<b>Funktion</b>	Funktion wechselt den Modus für einen Dynamischen Antennentuner.
<b>Syntax</b>	<b>int FEFU_DAT_SetMode( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucDatAdr, UCHAR ucMode )</b>
<b>Beschreibung</b>	<p>Funktion wechselt den Modus für einen Dynamischen Antennentuner.</p> <p><i>ucMode</i> ist das Mode-Byte.</p> <p><i>ucDatAdr</i> ist die per <b>FEFU_DAT_SetAddress</b> eingestellte Adresse.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

5.3.20. FEFU\_UMUX\_SoftVersion

---

<b>Funktion</b>	Funktion liest die Versionsnummer des UHF-Multiplexers aus.
<b>Syntax</b>	<b>int FEFU_UMUX_SoftVersion( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags, UCHAR* ucVersion )</b>
<b>Beschreibung</b>	<p>Die Versionsnummer des UHF-Multiplexers wird ermittelt und in <i>ucVersion</i> hinterlegt.</p> <p>Der Puffer für die Version muß mindestens 6 Byte aufnehmen können. Dabei ist 1 Byte für das NUL-Zeichen vorgesehen.</p> <p><i>ucMuxAdr</i> ist die per DIP-Schalter eingestellte Kaskadierungsebene.</p> <p><i>ucFlags</i> enthält Steuerflags.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Querverweis</b>	Die Struktur von <i>ucVersion</i> ist im Systemhandbuch H80302-xx-ID-B zum UHF-Multiplexer dokumentiert.
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

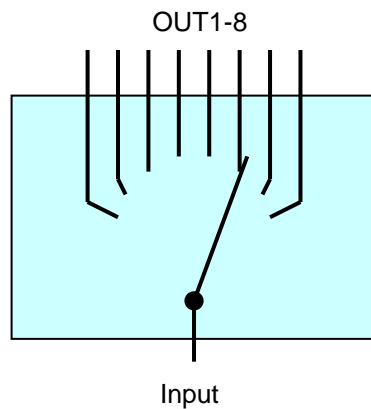
5.3.21. FEFU\_UMUX\_CPUReset

---

<b>Funktion</b>	Funktion löst einen Reset in der CPU des UHF-Multiplexers aus.
<b>Syntax</b>	<b>int FEFU_UMUX_CPUReset( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags )</b>
<b>Beschreibung</b>	<p>Funktion löst einen Reset in der CPU des Multiplexers aus.</p> <p><i>ucMuxAdr</i> ist die per DIP-Schalter eingestellte Kaskadierungsebene.</p> <p><i>ucFlags</i> enthält Steuerflags.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

## 5.3.22. FEFU\_UMUX\_SelectChannel

<b>Funktion</b>	Funktion setzt Ausgänge des UHF-Multiplexers.
<b>Syntax</b>	<b>int FEFU_UMUX_SelectChannel( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags, UCHAR ucChannelNo )</b>
<b>Beschreibung</b>	<p>Die Funktion schaltet den Eingang auf den Ausgang <i>ucChannelNo</i> des Multiplexers.</p> <p><i>ucMuxAdr</i> ist die per DIP-Schalter eingestellte Kaskadierungsebene.</p> <p><i>ucFlags</i> enthält Steuerflags.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>



---

5.3.23. FEFU\_MUX\_Detect\_GetPower

---

<b>Funktion</b>	Funktion detektiert einen UHF-Multiplexer und ermittelt die aktuellen Leistungswerte
<b>Syntax</b>	<b>int FEFU_MUX_Detect_GetPower( int iReaderHnd, UCHAR ucReaderBusAdr, UCHAR ucMuxAdr, UCHAR ucFlags, UCHAR* ucData )</b>
<b>Beschreibung</b>	<p>Funktion detektiert einen UHF-Multiplexer und ermittelt die aktuellen Leistungswerte vom geschalteten Ausgang, die in <i>ucData</i> zurückgegeben werden.</p> <p><i>ucMuxAdr</i> ist die Kaskadierungsebene, in der detektiert werden soll.</p> <p><i>iReaderHnd</i> ist der Handle zum Leser-Objekt in der FEISC Bibliothek.</p> <p><i>ucReaderBusAdr</i> ist die im Leser eingestellte Busadresse.</p>
<b>Querverweis</b>	Die Struktur von <i>ucData</i> ist im Systemhandbuch H80302-xx-ID-B zum UHF-Multiplexer dokumentiert.
<b>Rückgabewert</b>	<p>Der Rückgabewert enthält im fehlerfreien Fall das Statusbyte des Antwortprotokolls.</p> <p>Die Liste der Fehlercodes findet sich im Anhang.</p>

---

**ANHANG**

---

---

**5.4. Fehlercodes**

---

Fehler-Konstante	Wert	Beschreibung
FEFU_ERR_POINTER_IS_NULL	-4102	Zeiger auf Übergabeparameter ist NULL
FEFU_ERR_NO_MORE_MEM	-4103	Kein Systemspeicher mehr
FEFU_ERR_UNSUPPORTED_FUNCTION	-4105	nicht unterstützte Funktion
FEFU_ERR_PROTLEN	-4130	Protokolllängenfehler
FEFU_ERR_CHECKSUM	-4131	Checksummenfehler
FEFU_ERR_TIMEOUT	-4132	Kommunikations-Timeout
FEFU_ERR_UNKNOWN_STATUS	-4133	unbekanntes Statusbyte
FEFU_ERR_NO_RECDATA	-4134	keine Empfangsdaten eingetroffen
FEFU_ERR_UNKNOWN_PARAMETER	-4150	Übergabeparameter ist nicht bekannt
FEFU_ERR_PARAMETER_OUT_OF_RANGE	-4151	Übergabeparameter zu groß oder zu klein
FEFU_ERR_UNKNOWN_ERRORCODE	-4153	unbekannter Fehlercode

---

## 5.5. Historie

---

### V2.00.00

- Windows / Windows CE: Migration der Entwicklungsumgebung von Visual Studio 6 zu Visual Studio 2008.
- Funktional ohne Änderungen

### V1.03.00

- Neue Funktionen für den UHF-Multiplexer ID ISC.UMUX.

### V1.02.00

- Modifikationen für das Timing der Funktion FEFU\_DAT\_GetValues.
- Modifizierte Lizenzbestimmung
- Die Linux-Version wurde mit dem Compiler GCC 3.3.3 unter SuSE Linux 9.1 erstellt.

### V1.01.00

- Unterstützung für den Dynamischen Antennentuner ID ISC.DAT.
- Erstes Linux Release (SuSE Linux 8.2, GNU Compiler Collection V3.3-23, glibc V2.3.2-6)

### V1.00.00

- Dies ist die erste Version.