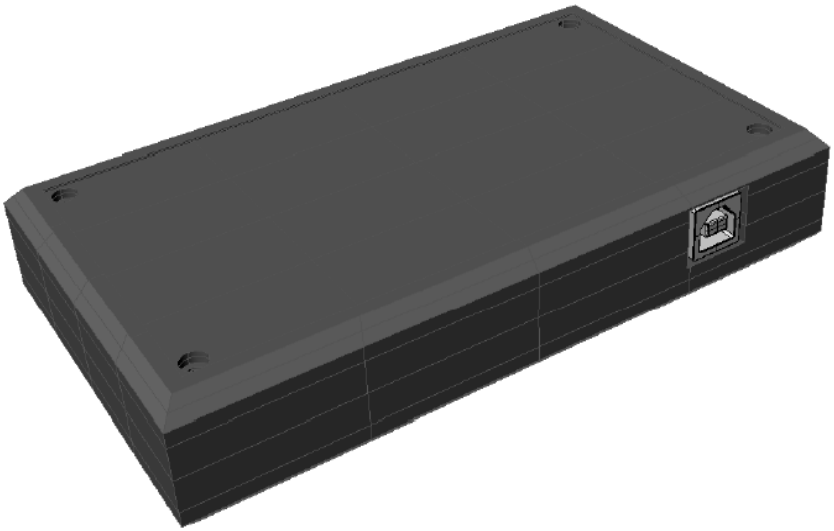
RFID-Leseeinheit mit USB-Schnittstelle

Version 1.10, Software- und Firmwareversion 1.20



Bedienungsanleitung

Dokument-Version A, erstellt am 26.05.2015

CGC Instruments
Hübschmannstr. 18 | D–09112 Chemnitz
Tel.: +49 (371) 355 098–55
Fax: +49 (371) 355 098–60
Internet: www.cgc-instruments.com
E–Mail: info@cgc-instruments.com

Inhalt

Technische Daten

Charakteristik

- Leseeinheit für RFID-Transponder
- Schnittstelle: USB
- 16-Bit Mikrocontroller
- Stromversorgung aus der USB-Schnittstelle
- ABS Gehäuse

Schnittstelle

- USB-Schnittstelle nach dem Standard USB 2.0
  Anschluss: USB-Buchse Typ B
  Datenübertragungsrate: bis zu 12 MBit/s (*Full Speed*)

RFID-Leseeinheit

- Arbeitsfrequenz: 125 kHz
- einsetzbare Transponder: EM 4102
  Zugriff: nur lesen,
  nutzbare Speicherkapazität: 40 Bit = 5 Byte,
  Sicherheit: CRC-Check
- Leseabstand für ISO-Karten: ca. 12 cm[†]

Stromversorgung

- Stromverbrauch:
  35 mA typ. mit deaktiviertem RFID-Empfänger
  130 mA max. mit aktivem RFID-Empfänger
- Einschaltstrombegrenzer

Allgemein

- ABS-Kunststoffgehäuse
  Abmessungen: 144,5 x 85 x 25 mm^3
  (Länge × Breite × Höhe)
- Gewicht: ca. 130 g

[†] Der Leseabstand hängt stark von der Umgebung ab. Wird beispielsweise ein Transponder direkt an eine mehrere mm dicke Aluminiumplatte befestigt, ist das Einlesen gar nicht mehr möglich. Auch eine dünnere direkt angebrachte metallische Unterlage bewirkt eine wesentliche Reduzierung des Leseabstandes. Sollen die Transponder an einen metallischen Gegenstand montiert werden, muss dies mittels einer Kunststoffunterlage geschehen. Ihre Dicke muss experimentell festgestellt werden.

Lieferungsumfang

- komplett getestetes Modul USB–RFID
- Diagnosesoftware
- Bedienungsanleitung
- optional:
  - USB-Anschlusskabel A-Stecker auf B-Stecker
  - USB-2.0-Schnittstellenkarte

Beschreibung

Anschlüsse und Konfigurationselemente

Das RFID-Modul verfügt über einen USB-Steckverbinder für die Daten-Schnittstelle und die Stromversorgung (siehe Abb. 1). Weiterhin befindet sich in dem Gerät eine mehrfarbige Leuchtdiode für die Statusanzeige, für die Bedeutung der Farben siehe Tab. 1. Nach dem Öffnen des Moduls ist ein Programmierstecker und zwei Kurzschlussbrücken zugänglich, mit denen das Aktualisieren der Firmware über die USB-Schnittstelle erlaubt wird.[‡]

Tab. 1. Farben der Leuchtdiode.

Farbe	Bedeutung
orange blinkend	Das Modul ist eingeschaltet, die RFID Leseeinheit ist inaktiv
orange	Die RFID Leseeinheit ist aktiv
grün	Die RFID Leseeinheit hat einen gültigen Transponder-Code eingelesen
rot	Diese Farbe wird von der RFID Leseeinheit nicht verwendet, kann nur über die Daten-Schnittstelle aktiviert werden

! **Vorsicht:** Das Modul ist ein elektronisches Gerät, das empfindlich gegen statische Elektrizität ist. Bei der Manipulation mit dem Modul, besonders beim geöffneten Gehäuse, müssen die ESD-Maßnahmen (*Electrostatic Discharge*) beachtet werden. Es muss unbedingt dafür gesorgt werden, dass bevor ein beliebiges Teil des Moduls angefasst wird, weder die Hände noch das Werkzeug elektrostatisch aufgeladen sind.

! **Vorsicht:** Beim Entfernen des Gehäusedeckels ist äußerste Vorsicht geboten. Die Schrauben zur Halterung des Deckels sind Blech-

[‡] Das Hochladen einer neuen Firmware in das RFID-Modul soll nicht direkt vom Kunden durchgeführt werden, wird daher in dieser Bedienungsanleitung nicht beschrieben. Die Kurzschlussbrücken dürfen nicht eingesetzt werden, denn bei einer falschen Bedienung kann das Modul beschädigt werden.

schrauben, ihr Halt im Gehäuse wird bei jedem Zerlegen des Moduls verringert.

USB-Schnittstelle

Das Modul verfügt über eine USB-Schnittstelle zur Datenübertragung zwischen dem Modul und einem Steuerrechner (üblicherweise einem PC). Die Datenübertragungsrate beträgt 230,4 kBaud, wodurch kurze Antwortzeiten gesichert sind.

Die USB-Schnittstelle erfordert einen kurzen Abstand zwischen dem Steuerrechner und dem Modul, durch die Datenkabel wird der Steuerrechner mit dem Modul galvanisch verbunden. Durch die galvanische Kopplung können Störsignale vom Steuerrechner übertragen werden oder können Störungen von Außen die Funktion des Steuerrechners beeinflussen.

Für die Funktion der USB-Schnittstelle ist an der PC-Seite ein virtueller Porttreiber erforderlich, welcher einen virtuellen COM-Port erstellt (siehe Abschnitt "USB-Schnittstelle"). Die Kommunikation mit dem RFID-Modul erfolgt über diesen Port, der Schnittstellenadapter kommuniziert intern mit dem Mikrocontroller mit der o.g. Datenübertra-

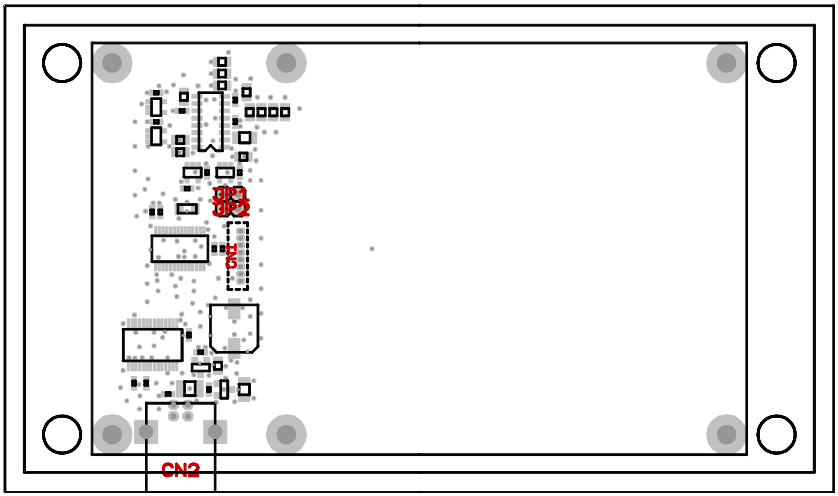


Abb. 1. Anschlüsse und Konfigurationselemente des Moduls.
CN1: Programmierstecker (optional), CN2: USB-Anschluss, JP1, JP2: Kurzschlussbrücken zum Programmieren des Mikrocontrollers.

gungsrate von 230,4 kBaud.

Die USB-Schnittstelle ist mit der USB-Version 2.0 kompatibel, sie nutzt den *Full-Speed Mode*, d.h. dass die Übertragung über USB mit der Datenrate von maximal 12 MBit/s erfolgt. Der Anschluss ist der B-Steckverbinder (CN2 in Abb. 1), für die Verbindung mit einem USB-Host (dem Steuerrechner) ist ein genormtes Kabel mit A- und B-Steckern erforderlich. Die USB-Schnittstelle nutzt die vom Host bereitgestellte Stromversorgung. Um den Einschaltstrom auf den von der USB-Norm geforderten Wert zu reduzieren, ist in dem RFID-Modul ein Strombegrenzer eingebaut.

RFID-Leseeinheit

Die RFID-Leseeinheit kann die Daten der Transponder des Typs EM 4102 einlesen. Diese nicht programmierbaren Transponder übermitteln einen 40 Bit-langen Code, welcher durch CRC gesichert ist. Die Arbeitsfrequenz der RFID-Leseeinheit liegt bei 125 kHz. Die RFID-Spule befindet sich direkt auf der Leiterplatte des Moduls, wodurch eine hervorragende Stabilität der RFID-Leseeinheit gewährleistet wird. Die Reichweite ist typischerweise höher als 10 cm und ist damit gut mit RFID-Leseeinheiten vergleichbar, welche gewickelte Luftspulen nutzen.

Befindet sich ein Transponder in der Reichweite der RFID-Leseeinheit, wird sein Code empfangen, durch den Mikrocontroller dekodiert und für den Datentransfer zum Steuerrechner bereitgestellt. Der Datentransfer kann automatisch oder nach Aufforderung erfolgen. Der automatische Datentransfer erfolgt nur, wenn ein neuer Transponder identifiziert oder aus der Reichweite der RFID-Leseeinheit entfernt wird. Eine direkte Abfrage ist zu jeder Zeit möglich.

Die RFID-Leseeinheit kann je nach Bedarf extern aktiviert oder deaktiviert werden. Im inaktiven Zustand sinkt die Stromaufnahme und dadurch auch die Wärmeentwicklung des Moduls. Das Aktivieren der RFID-Leseeinheit benötigt typischerweise einige 100 ms, bis der Code eines vorliegenden Transponders eingelesen wird.

Inbetriebnahme

Das Modul USB-RFID benötigt für seine Funktion eine funktionierende USB-Schnittstelle. Für den Anschluss ist ein passendes Datenkabel erforderlich.

Um die Funktion des Moduls zu überprüfen, gehen Sie wie folgt vor:

- Schließen Sie das Modul mit Hilfe eines Datenkabels an eine USB-Schnittstelle des PCs an. Besitzt der PC keine (freie) USB-Schnittstelle, muss eine Schnittstellenkarte installiert werden. Auf der PC-Seite muss weiterhin ein virtueller Porttreiber installiert werden (siehe Abschnitt "Treiberinstallation").
- Nach dem Anschließen des Moduls an eine USB-Schnittstelle muss die Leuchtdiode gelb blinken. Geschieht dies nicht, überprüfen Sie das Anschlusskabel und die USB-Schnittstelle des PCs. Lassen Sie bei einem solchen Problem das Modul über einige Minuten im stromlosen Zustand, damit sich die Kondensatoren in dem Modul entladen können. Fahren Sie keineswegs mit der Inbetriebnahme fort, wenn die Leuchtdiode auch nach wiederholtem Einschalten nicht blinkt. Es deutet auf ein grundlegendes Problem hin, das Modul muss zur Inspektion eingeschickt werden.
- Öffnen Sie ein Kommandozeilenfenster, wechseln Sie in das Verzeichnis mit dem Diagnoseprogramm `USB-RFID-Test.exe` (Verzeichnis "`Program`" des beiliegenden Softwarepakets) und starten Sie es. Das Programm zeigt beim Aufrufen die verfügbaren Optionen. Für eine erfolgreiche Ausführung muss mindestens die Nummer der virtuellen seriellen Schnittstelle angegeben werden, an die das Modul angeschlossen ist. Ist das Modul beispielsweise an die virtuellen Schnittstelle `COM5` angeschlossen, muss das Diagnoseprogramm durch die folgende Anweisung gestartet werden: "`USB-RFID-Test 5`". Wird die angegebene Schnittstelle nicht gefunden, endet das Programm mit einer Fehlermeldung. In einem solchen Fall muss im Gerätemanager von Windows überprüft werden, ob die virtuelle Schnittstelle existiert und ob sie funktionsfähig ist. Die Konfiguration der Schnittstelle ist dabei unerheblich, denn sie wird durch das Diagnoseprogramm selbst eingestellt.
- Konnte das Diagnoseprogramm erfolgreich gestartet werden, drücken Sie die Taste 'p' auf der Tastatur des PCs. Das Programm liest aus dem angeschlossenen Modul die Produktbezeichnung aus. Sollte anstelle davon ein Fehler gemeldet werden, wurde die Daten-

übertragung gestört. Wenn die Übertragung auch bei einer Wiederholung nicht korrekt funktioniert, versuchen Sie durch das Drücken der Taste 'Z' die Kommunikationspuffer zu löschen und die Kommunikation neu zu starten. Gelingt es auch dann nicht, eine fehlerfreie Antwort von dem RFID-Modul zu erhalten, liegt ein grundlegender Fehler vor. Überprüfen Sie das Anschlusskabel und die Funktion der benutzten USB-Schnittstelle des PCs.

- Funktioniert alles fehlerfrei, können Sie durch Drücken der Tasten 'n', 'd' und 'v' weitere Produktinformationen abfragen.
- Schalten Sie die RFID-Leseeinheit durch Drücken der Taste 'F' ein. Drücken Sie die Taste 'R', die automatische Abfrage der RFID-Leseeinheit wird gestartet. Bringen Sie einen Transponder in die Nähe des Moduls. Die Ebene des Transponders soll dabei parallel zu dem Deckel des Moduls abgebracht werden und der Transponder soll sich über der Lesespule des Moduls (markierte Fläche auf dem Deckel) befinden. Der Inhalt des Transponders wird eingelesen und auf dem PC-Bildschirm dargestellt. Stoppen Sie die automatische Abfrage durch Drücken der Taste 'r' und schalten Sie die RFID-Leseeinheit durch Drücken der Taste 'f' wieder aus.

Konnten die Tests fehlerfrei durchgeführt werden, ist das Modul für den Einsatz betriebsbereit.

Treiberinstallation

Installation des virtuellen Ports für die USB-Schnittstelle

Der virtuelle Porttreiber ist für den Anschluss des Moduls an eine USB-Schnittstelle erforderlich. Die Installationsdaten befinden sich im Verzeichnis "USB" des beiliegenden Softwarepakets. Beachten Sie dabei folgendes:

- Alternativ zu den Treibern aus der beiliegenden CD können aktuelle Treiber von der Homepage des Herstellers des USB-Adapters heruntergeladen werden. Die Treiber werden unter der Adresse http://www.ftdichip.com/Drivers/VCP.htm zur Verfügung gestellt, es soll die zum Betriebsystem passende Datei heruntergeladen werden.
- Zum Installieren der Treiber benötigen Sie Administrationsrechte.
- Die Installation ist in einer der sich in dem o.g. Verzeichnis befindlichen pdf-Datei detailliert beschrieben. Lesen Sie diese Beschreibung sorgfältig durch.
- Nach der Installation kann die Nummer der virtuellen Schnittstelle eingestellt werden. Dies wird im Gerätemanager bei dem Gerät *USB Serial Port* durchgeführt, es sind dazu Administrationsrechte erforderlich. Die Umstellung der Portnummer geschieht sofort nach der Bestätigung, der Rechner muss dabei nicht neu gestartet werden.

Softwareschnittstelle

Die Softwareschnittstelle zu dem RFID-Modul besteht aus einer dynamischen Linkbibliothek `USB-RFID.dll`. Sie befindet sich im Verzeichnis "Program" des beigelegten Softwarepakets. Die Softwareschnittstelle stellt ein selbständiges Softwarepaket dar, für ihre Funktion sind keine weiteren Treiber oder Bibliotheken erforderlich. Für die USB-Schnittstelle sind selbstverständlich geeignete Hardwaretreiber notwendig, um den Zugriff auf die Hardware überhaupt zu ermöglichen. Weiterhin muss ein entsprechender Treiber für den virtuellen COM-Port installiert werden (siehe Abschnitt "Treiberinstallation").

Die Benutzerfunktionen in der dynamischen Linkbibliothek `USB-RFID.dll` können aus allen gängigen Programmiersprachen aufgerufen werden. Die Details entnehmen Sie dem Benutzerhandbuch Ihres Compilers. Die Definitionen der Benutzerfunktionen sind in der Datei `USB-RFID.h` enthalten. Sollte Ihr Compiler keine Importbibliothek aus der dynamischen Linkbibliothek erstellen können, steht Ihnen die Datei `USB-RFID.lib` zur Verfügung.

Die am Ende dieses Abschnitts aufgeführten Programmbeispiele sind für den Compiler Borland C++ vorgesehen, sie sollen jedoch ohne eine erforderliche Änderung auch mit anderen Compilern kompatibel sein.

Funktion der Softwareschnittstelle

Die Softwareschnittstelle kann insgesamt 8 Kommunikationskanäle für die Datenübertragung verwalten. Jeder benutzte Kanal muss durch das Öffnen einer seriellen Schnittstelle zugeordnet werden. Ein geöffneter Kanal kann wieder geschlossen werden und danach durch ein erneutes Öffnen entweder der gleichen oder einer anderen Schnittstelle zugeordnet werden.

Das Öffnen eines Kommunikationskanals konfiguriert die zugeordnete serielle Schnittstelle und löscht ihre Puffer. Direkt nach dem Öffnen kann jede beliebige Funktion zur Datenübertragung aufgerufen werden.

Alle Funktionen erfordern als einen Parameter die Nummer des Kommunikationskanals `PortNumber` (siehe weiter), auf dem die gewünschte Operation ausgeführt werden soll. Es handelt sich um eine

Tab. 2. Rückgabewerte der Funktionen

Rückgabewert	Fehlermeldung	Beschreibung
0	No error	Die Datenübertragung endete erfolgreich.
1	No notification data is available	Es wurden keine automatisch übermittelten Daten empfangen. Diesen Rückgabewert liefert nur die Funktion `USB_RFID_CheckAutoInput`.
-1	PortNumber out of range	Nummer des Kommunikationskanals `PortNumber` war größer als maximal zulässig.
-2	Error opening the port	Die Schnittstelle konnte nicht geöffnet werden. Für mögliche Ursachen siehe Tab. 3.
-3	Error closing the port	Die Schnittstelle konnte nicht geschlossen werden. Für mögliche Ursachen siehe Tab. 3.
-4	Error purging the port	Die Puffer der Schnittstelle konnten nicht gelöscht werden. Für mögliche Ursachen siehe Tab. 3.
-5	Error setting the port control lines	Die Steuerleitungen der Schnittstelle konnten nicht angesteuert werden.
-6	Error reading the port status lines	Der Zustand der Statusleitungen der Schnittstelle konnte nicht ermittelt werden.
-7	Error sending command	Die Datenübertragung zum Modul ist fehlgeschlagen. Für mögliche Ursachen siehe Tab. 3.
-8	Error sending data	
-9	Error sending termination character	

Rückgabewert	Fehlermeldung	Beschreibung
-10	Error receiving command	Die Datenübertragung vom Modul ist fehlgeschlagen. Für mögliche Ursachen siehe Tab. 3.
-11	Error receiving data	
-12	Error receiving termination character	
-13	Wrong command received	Es wurde eine falsche Antwort vom Modul erhalten.
-14	Wrong argument received	Die vom Modul empfangene Antwort enthält falsche Daten.
-15	Wrong argument passed to the function	Der Funktion wurde ein Argument mit falschem Wert übergeben.
-100	Device not connected	Die Steuerleitungen der Schnittstelle zeigen, dass das Modul nicht angeschlossen ist.
-101	Device not ready	Die Steuerleitungen der Schnittstelle zeigen, dass das Modul nicht bereit ist. Dies kann durch einen laufenden Hintergrundprozess verursacht sein. Wiederholen Sie die Operation, um das Problem zu lösen.
-102	Device state could not be set to not ready	Das Modul reagiert nicht richtig. Versuchen Sie, die Kommunikation zurückzusetzen oder starten Sie das Modul neu, indem Sie es aus- und wieder einschalten.
-400	Error opening the file for debugging output	Die Textdatei für die Debugging-Ausgabe konnte nicht geöffnet werden.
-401	Error closing the file for debugging output	Die Datei für die Debugging-Ausgabe konnte nicht geschlossen werden.

vorzeichenlose 16 Bit-Zahl (`WORD`). Die Nummer 0 bezeichnet den ersten Kanal, die Nummer 7 den letzten. Die Anzahl der Kanäle ist durch die Konstante `USB_RFID_MAX_PORT` definiert.

Der Rückgabewert der meisten Funktionen enthält die Aussage über den Erfolg der jeweiligen Operation. Es handelt sich um eine 32 Bit-Zahl mit Vorzeichen (`int`). Der letzte Rückgabewert kann durch die Funktion `USB_RFID_State` neu abgefragt werden. Tabelle 2 listet die möglichen Rückgabewerte zusammen mit den Fehlermeldungen auf, wie sie durch die Funktion `USB_RFID_ErrorMessage` zurückgegeben werden. Ist ein Fehler bei der Datenübertragung entstanden, kann die Ursache durch Aufrufen der Funktionen `USB_RFID_IO_State` und `USB_RFID_IO_ErrorMessage` herausgefunden werden. Die erste gibt den letzten I/O-Fehler zurück, die zweite die entsprechende Fehlermeldung (siehe Tab. 3.).

Um eine einfachere Integration in Pascal-Softwarepakete zu vereinfachen, verwenden alle Funktionen der Softwareschnittstelle die Pascal-Aufrufkonvention.

Tab. 3. I/O-Fehler

Rückgabewert	Fehlermeldung	Beschreibung
0	No error	Die Funktion endete erfolgreich.
-1	`PortNumber` out of range	Nummer des Kommunikationskanals `PortNumber` war größer als maximal zulässig.
1	Port has not been opened yet	Es wurde versucht, einen Kommunikationskanal zu verwenden, der noch nicht geöffnet wurde.
2	Cannot open the port	Die spezifizierte Schnittstelle konnte nicht geöffnet werden. Die Schnittstelle existiert entweder nicht oder wird von einem anderen Programm verwendet.
3	Cannot get the state of the port	Der Zustand der Schnittstelle konnte nicht ermittelt werden.
4	Cannot set the state of the port	Der Zustand der Schnittstelle konnte nicht gesetzt werden.
5	Cannot set the timeouts for the port	Das Timeout für die Schnittstelle konnte nicht gesetzt werden.
6	Cannot clear the port	Die Puffer der Schnittstelle konnten nicht gelöscht werden.
7	Error reading data from the port	Der Empfang der Daten ist fehlgeschlagen.
8	Error writing data to the port	Das Senden der Daten ist fehlgeschlagen.
9	Wrong data amount written to the port	Es konnte nicht die richtige Anzahl an Zeichen gesendet werden.
10	Error setting the control lines of the port	Die Steuerleitungen der Schnittstelle konnten nicht angesteuert werden.
11	Error reading the status lines of the port	Der Zustand der Statusleitungen der Schnittstelle konnte nicht ermittelt werden.
12	Device is busy	Das Modul ist zur Kommunikation nicht bereit.

Steuerung der Kommunikation

Funktion USB_RFID_Open

```
int pascal USB_RFID_Open (WORD PortNumber,
  WORD ComNumber);
```

Öffnet die angegebene Schnittstelle für den Kommunikationskanal und liefert einen Fehlercode nach Tab. 2. Sie muss die erste Funktion sein, die während der Kommunikation auf dem jeweiligen Kommunikationskanal ausgeführt wird. Endet diese Funktion mit einem Fehler, ist keine Kommunikation auf dem Kanal möglich. Die Zahl `ComNumber` bezeichnet die Nummer der COM-Schnittstelle, die dem Kommunikationskanal zugeordnet werden soll. `ComNumber=1` steht beispielsweise für die Schnittstelle COM1.

Funktion USB_RFID_Close

```
int pascal USB_RFID_Close (WORD PortNumber);
```

Schließt den Kommunikationskanal und liefert einen Fehlercode nach Tab. 2. Sie kann verwendet werden, um den Kommunikationskanal freizugeben, um ihn später durch Öffnen der gleichen oder einen anderen Schnittstelle zuzuordnen.

Wird das Programm beendet, das die Softwareschnittstelle `USB-RFID.dll` exklusiv benutzt hat, werden alle geöffneten Kommunikationskanäle automatisch geschlossen. Der Programmierer muss in einem solchen Falle keine Sorge dafür tragen.

Funktion USB_RFID_Purge

```
int pascal USB_RFID_Purge (WORD PortNumber);
```

Löscht die Schnittstellenpuffer und liefert einen Fehlercode nach Tab. 2. Sie kann verwendet werden, um eine gestörte Kommunikation wiederherzustellen. Wird beispielsweise das RFID-Modul erst nach dem Start der Software eingeschaltet, soll als erstes diese Funktion aufgerufen werden, um evtl. fehlerhaft empfangene Daten zu löschen.

Die Funktion `USB_RFID_Purge` wird automatisch beim Ausführen der Funktion `USB_RFID_Open` aufgerufen.

Funktion USB_RFID_Buffer_State

```
int pascal USB_RFID_Buffer_State
  (WORD PortNumber, BOOL & empty);
```

Gibt in der Variable `empty` den Zustand des Empfangspuffers des RFID-Moduls zurück und liefert einen Fehlercode nach Tab. 2. Die Funktion kann verwendet werden, um vor einem Transfer einer größeren Datenmenge sicherzustellen, dass in dem Empfangspuffer genug Platz dafür vorhanden ist. Wird in der Variable `empty` durch den Wert `false` auf einen nicht leeren Empfangspuffer hingewiesen, besteht keine Garantie, dass das Modul neue Daten empfangen kann.

Funktion USB_RFID_Device_Purge

```
int pascal USB_RFID_Device_Purge
  (WORD PortNumber, BOOL & empty);
```

Löscht den Sendepuffer des RFID-Moduls und liefert einen Fehlercode nach Tab. 2. In der Variable `empty` wird der Zustand des Empfangspuffers zurückgegeben (siehe ebenfalls die Funktion `USB_RFID_Buffer_State`). Die Funktion kann verwendet werden, um eine gestörte Kommunikation wiederherzustellen. Antwortet beispielsweise das RFID-Modul durch gestörte Kommunikation nicht korrekt, soll diese Funktion aufgerufen werden, um den Sendepuffer zu löschen.

RFID-Leseeinheit

Funktion USB_RFID_EnableRFID

```
int pascal USB_RFID_EnableRFID
  (WORD PortNumber, BOOL Enable);
```

Schaltet den RFID-Empfänger ein oder aus. Ist der Empfänger ausgeschaltet, werden keine neuen Transponder identifiziert und der zuletzt eingelesene Code wird als ungültig markiert.

Nach dem Einschalten des Moduls ist der RFID-Empfänger ausgeschaltet. Der eingeschaltete Empfänger erhöht den Stromverbrauch des Moduls und dadurch seine Wärmeentwicklung. Wenn die RFID-Funktion nicht benötigt wird, soll der Empfänger ausgeschaltet werden. Beim Einschalten muss eine gewisse Verzögerung berücksichtigt werden, das Aktivieren des RFID-Empfängers dauert einige 100 ms.

Funktion USB_RFID_GetRFID

```
int pascal USB_RFID_GetRFID
  (WORD PortNumber, BYTE Value [8]);
```

Liest die zuletzt empfangenen Daten aus dem RFID-Empfänger ein und dekodiert sie in die Variable `Value`. Die Daten werden in der Form einer vorzeichenlosen 64 Bit-Zahl zurückgegeben. Aus Kompatibilitätsgründen mit älteren Compilern, die keine 64 Bit-Arithmetik unterstützen, wird diese Zahl als eine 8 Byte-Zeichenkette deklariert. Die Funktion verlangt einen Zeiger auf diese Zahl.

Der Rückgabewert in der Variable `Value` nutzt für gültige Transponder-Codes die niedrigsten 40 Bits der 64 Bit-Zahl, die restlichen Bits (die oberen 3 Bytes) werden mit 0 gefüllt. Befindet sich kein

Tab. 4. Konstanten für die RFID-Leseeinheit.

Wert	Bezeichnung	Beschreibung
FFh	USB_RFID_INVALID_BYTE	Rückgabewerte für einen nicht dekodierbaren oder nicht vorhandenen Transponder
FFFFFFFFh	USB_RFID_INVALID_DWORD	
-1	USB_RFID_INVALID	

Transponder in der Reichweite oder konnte er nicht fehlerfrei dekodiert werden, gleicht der Rückgabewert `-1` d.h. alle 8 Bytes gleichen 255 (FFh). Für eine Entscheidung über die Gültigkeit der übermittelten Zahl `Value` reicht es also das MSB auf `-1` zu testen. Für die Definition dieser Konstanten siehe auch Tab. 4.

Wird die Funktion `USB_RFID_GetRFID` aufgerufen, wird die automatische Übermittlung der Zustände des Tastenfelds unterbrochen, welche durch die Funktion `USB_RFID_AutoRFID` gestartet wurde.

Funktion USB_RFID_AutoRFID

```
int pascal USB_RFID_AutoRFID
  (WORD PortNumber);
```

Startet die automatische Übermittlung der Daten aus dem RFID-Empfänger. Die automatische Übermittlung wird durch den Aufruf der Funktion `USB_RFID_GetRFID` wieder unterbrochen.

Ist die automatische Übermittlung der RFID-Daten aktiv, soll die Funktion `USB_RFID_CheckAutoInput` periodisch in kurzen Abständen aufgerufen werden, um diese Daten zu bearbeiten.

Nach dem Einschalten des Moduls sind alle automatischen Übermittlungen ausgeschaltet.

Funktion USB_RFID_GetLastRFID

```
int pascal USB_RFID_GetLastRFID
  (WORD PortNumber, BYTE Value [8]);
```

Gibt analog zu der Funktion `USB_RFID_GetRFID` den Wert der zuletzt automatisch übermittelten Daten aus dem RFID-Empfänger zurück. Für mehrere Details siehe die Beschreibung der Funktionen `USB_RFID_GetRFID` und `USB_RFID_CheckAutoInput`.

Automatische Datenübermittlung

Funktion USB_RFID_CheckAutoInput

```
int pascal USB_RFID_CheckAutoInput
  (WORD PortNumber, WORD & CommandMask);
```

Prüft den Empfangspuffer des Steuerrechners auf automatisch übermittelte Daten. Sind solche Daten empfangen worden, werden sie dekodiert und zwischengespeichert.

Der Rückgabewert in der Variable `CommandMask` ist eine Bitmaske, die angibt, welche Daten seit dem letzten Aufruf dieser Funktion oder der Funktion `USB_RFID_GetAutoMask` empfangen wurden. Der Wert ist eine Bitkombination aus den in der Datei `USB-RFID.h` definierten Werten `AUTO_xxx`. (siehe Tab. 5). Je nach den gesetzten Bits in der Bitmaske `CommandMask` sollen entsprechende Funktionen zur Ermittlung der empfangenen Daten aufgerufen werden, diese sind ebenfalls in der Tabelle 5 als Empfangsfunktionen aufgelistet. Werden die Daten durch die entsprechende Empfangsfunktion nicht abgeholt, werden sie beim nächsten Empfang überschrieben.

Die Überprüfung auf automatisch übermittelte Daten erfolgt ebenfalls beim Aufruf von allen Funktionen der Softwareschnittstelle, welche Daten von dem RFID-Modul empfangen. In der Bitmaske `CommandMask` werden alle Daten wiederspiegelt, welche seit dem letzten Aufruf der Funktionen `USB_RFID_CheckAutoInput` oder `USB_RFID_GetAutoMask` gefunden wurden. Der Wert der Bitmaske `CommandMask` muss vom Benutzer zur Auswertung gespeichert werden, ein erneuter Aufruf einer der beiden o.g. Funktionen würde eine leere Bitmaske zurückgeben.

Die Rückgabewerte der Funktion `USB_RFID_CheckAutoInput` sind in der Tabelle 6 aufgelistet. Rückgabewerte der Funktion, welche ungleich Null sind, deuten auf einen Fehler hin und bedürfen einer ent-

Tab. 5. Bitmasken des automatischen Datenempfangs.

Wert	Bezeichnung	Empfangsfunktion	Datenquelle
64	`USB_RFID_AUTO`	`USB_RFID_GetLastRFID`	RFID-Empfänger

sprechenden Fehlerbehandlung. Ist der Rückgabewert gleich Null, entscheidet die Bitmaske `CommandMask`, ob automatisch übermittelte Daten empfangen wurden. Wenn die Bitmaske `CommandMask` ebenfalls gleich Null ist, lagen keine neuen Daten vor.

Die Funktion `USB_RFID_CheckAutoInput` soll periodisch aufgerufen werden, wenn das RDID-Modul Daten automatisch sendet. Die gesendeten Daten werden nach dem Versenden in dem Zwischenspeicher der Schnittstelle im Steuerrechner gespeichert. Die Funktion `USB_RFID_CheckAutoInput` soll so oft ausgeführt werden, dass der Zwischenspeicher nicht überlaufen kann. Je mehrere Daten automatisch gesendet werden, um so öfter soll diese Funktion aufgerufen werden. Die Häufigkeit der Aufrufe bestimmt ebenfalls die Zeitauflösung bei Ermittlung diverser Ereignisse. Ständige Aufrufe der Funktion `USB_RFID_CheckAutoInput` können auf der anderen Seite eine unnötige Prozessorlast auf dem Steuerrechner erzeugen. In der tatsächlichen Anwendung soll daher ein Kompromiss gefunden werden und die Funktion nur so oft aufgerufen werden, dass die beabsichtigte Zeitauflösung erreicht wird und dass keine Daten durch etwaige Überläufe des Zwischenspeichers verloren gehen. Bei modernen Rechnern kann die Funktion `USB_RFID_CheckAutoInput` in der Regel einmal pro Millisekunde aufgerufen werden, ohne dass dadurch eine nennenswerte Prozessorlast entsteht.

Tab. 6. Rückgabewerte der Funktion `USB_RFID_CheckAutoInput`.

Rückgabewert	Bedeutung
0	Die Schnittstelle hat möglicherweise Daten empfangen, die Bitmaske `CommandMask` wiederspiegelt, ob sie zwischengespeichert wurden
1	Die Schnittstelle hat Daten empfangen, welche keiner automatischen Datenübermittlung entsprechen, die Bitmaske `CommandMask` wiederspiegelt die davor korrekt empfangenen Daten
<0	siehe Tab. 2

Funktion USB_RFID_GetAutoMask

```
int pascal USB_RFID_GetAutoMask
  (WORD PortNumber, WORD & CommandMask);
```

Gibt die Bitmaske `CommandMask` zurück, die angibt, welche Daten seit dem letzten Aufruf dieser Funktion oder der Funktion `USB_RFID_CheckAutoInput` empfangen wurden.

Alle Funktionen der Softwareschnittstelle, welche Daten von dem RFID-Modul empfangen, können auch automatisch übermittelte Daten empfangen. Die Funktion `USB_RFID_GetAutoMask` kann verwendet werden, um festzustellen, ob ein solcher Empfang stattfand. Ist die Bitmaske `CommandMask` gleich Null, lagen keine Daten vor. Im anderen Falle sollen die Daten durch entsprechende Funktionen ermittelt werden (siehe Tab. 5 und die Beschreibung der Funktion `USB_RFID_CheckAutoInput`).

Der Wert der Bitmaske `CommandMask` muss vom Benutzer zur Auswertung gespeichert werden, ein erneuter Aufruf der Funktionen `USB_RFID_CheckAutoInput` oder `USB_RFID_GetAutoMask` würde eine leere Bitmaske zurückgeben.

Steuerung der Leuchtdiode

Funktion USB_RFID_SetLED

```
int pascal USB_RFID_SetLED (WORD PortNumber,
  BYTE Color, float Seconds);
```

Schaltet für die vorgegebene Dauer `Seconds` die Leuchtdiode auf die Farbe `Color` um. Die Dauer darf nicht negativ und muss kleiner als die Konstante `USB_RFID_LED_MAX_DELAY` sein. Für die erlaubten Farben sind ebenfalls Konstanten definiert (siehe Tab. 7).

Nach dem Ablauf der vorgegebenen Dauer wird die Leuchtdiode auf die Farbe umgeschaltet, welche dem jeweiligen Zustand des Moduls entspricht (siehe Tab. 1). Wird eine Dauer von `0` angegeben, wird die Leuchtdiode erst bei der nächsten Zustandsänderung des Moduls auf die Farbe nach Tab. 1 umgeschaltet.

Diese Funktion kann dazu verwendet werden, beispielsweise durch das Einschalten der roten Farbe einen Hinweis für den Benutzer des Moduls zu signalisieren.

Funktion USB_RFID_GetLED

```
int pascal USB_RFID_GetLED (WORD PortNumber,
  BYTE & Color);
```

Liest den aktuellen Zustand der Leuchtdiode ein und gibt die Farbe `Color` nach Tab. 7 zurück.

Tab. 7. Konstanten für die Leuchtdiodensteuerung.

Wert	Bezeichnung	Beschreibung
0	`USB_RFID_LED_NO`	Leuchtdiode aus
1	`USB_RFID_LED_RED`	rote Farbe
2	`USB_RFID_LED_GREEN`	grüne Farbe
3	`USB_RFID_LED_ORANGE`	gelbe Farbe
256	`USB_RFID_LED_MAX_DELAY`	maximal erlaubte Dauer

Fehlerbehandlung

Funktion USB_RFID_State

```
int pascal USB_RFID_State (WORD PortNumber);
```

Liefert einen Fehlercode nach Tab. 2. Die Funktion kann verwendet werden, um das Ergebnis der letzten Operation erneut abzufragen. Die Funktion hat keinen Einfluss auf die Kommunikation und kann zu jeder Zeit aufgerufen werden.

Funktion USB_RFID_ErrorMessage

```
const char * pascal USB_RFID_ErrorMessage
  (WORD PortNumber);
```

Liefert eine Fehlermeldung zu dem Ergebnis der letzten Operation. Der Rückgabewert ist ein Zeiger auf eine null-terminierte Zeichenkette, der Inhalt der Fehlermeldung kann Tab. 2 entnommen werden. Die Funktion hat keinen Einfluss auf die Kommunikation und kann zu jeder Zeit aufgerufen werden.

Funktion USB_RFID_IO_State

```
int pascal USB_RFID_IO_State
  (WORD PortNumber);
```

Liefert einen I/O-Fehlercode nach Tab. 3. Sie kann verwendet werden, um das Ergebnis der letzten I/O-Operation erneut abzufragen. Die Funktion hat keinen Einfluss auf die Kommunikation und kann zu jeder Zeit aufgerufen werden.

Funktion USB_RFID_IO_ErrorMessage

```
const char * pascal USB_RFID_IO_ErrorMessage
  (WORD PortNumber);
```

Liefert eine Fehlermeldung zu dem Ergebnis der letzten I/O-Operation. Der Rückgabewert ist ein Zeiger auf eine null-terminierte Zeichenkette. Der Inhalt der Fehlermeldung kann Tab. 3 entnommen werden. Die Funktion hat keinen Einfluss auf die Kommunikation und kann zu jeder Zeit aufgerufen werden.

Verschiedenes

Funktion USB_RFID_Version

```
WORD pascal USB_RFID_Version();
```

Liefert die Version der Softwareschnittstelle (der dynamischen Linkbibliothek `USB-RFID.dll`). Die Funktion soll verwendet werden, um festzustellen, ob eine Softwareschnittstelle mit der passenden Version verwendet wird. Die Funktion soll aufgerufen werden, bevor die restlichen Funktionen der Softwareschnittstelle verwendet werden.

Der Rückgabewert ist eine vorzeichenlose 16 Bit-Zahl (WORD), in dem höheren Byte wird die Hauptversionsnummer zurückgegeben, das niedrigere Byte bezeichnet die Reihenfolge innerhalb der Hauptversion. Eine identische Hauptversionsnummer zweier Bibliotheken `USB-RFID.dll` bedeutet, dass diese gleiche Funktionen implementieren und dass lediglich interne Korrekturen vorgenommen wurden. Sind die Hauptversionsnummern unterschiedlich, besteht keine Garantie, dass die Bibliothek eingesetzt werden kann, ohne dass das Programm neu kompiliert oder sogar modifiziert werden muss.

Funktion USB_RFID_GetUptime

```
int pascal USB_RFID_GetUptime
  (WORD PortNumber, DWORD & Seconds,
  WORD & Milliseconds);
```

Liefert die Zeit, welche seit dem Einschalten, bzw. seit dem Neustart des RFID-Moduls verstrichen ist. Sie kann verwendet werden, um festzustellen, ob das Modul einen ungeplanten Neustart durchgeführt hat. Die Rückgabewerte sind Sekunden in der Variable `Seconds` und Millisekunden in der Variable `Milliseconds`. Die Zeitauflösung beträgt 1/128 s.

Funktion USB_RFID_FW_Ver

```
int pascal USB_RFID_FW_Ver
  (WORD PortNumber, WORD & Version);
```

Liefert die Version der Firmware des RFID-Moduls. Sie soll verwendet werden, um festzustellen, ob die Softwareschnittstelle eine passende Version zu der Firmware hat.

Der Rückgabewert ist wie bei der Funktion `USB_RFID_Version` eine vorzeichenlose 16 Bit-Zahl `Version`. Die dynamische Linkbibliothek `USB-RFID.dll`. und die Firmware des RFID-Moduls sind miteinander kompatibel, wenn die Hauptversionsnummern identisch sind.

Funktion USB_RFID_FW_Date

```
int pascal USB_RFID_FW_Date
  (WORD PortNumber, char * DateString);
```

Liefert das Kompilationsdatum der Firmware des RFID-Moduls. Der Rückgabewert ist eine null-terminierte Zeichenkette, sie wird in den Puffer `DateString` kopiert. Der Puffer `DateString` muss vor dem Aufruf der Funktion definiert werden und muss groß genug sein, um alle Zeichen des Rückgabewerts (beispielsweise `12 OCT 2010`) und das Abschlusszeichen aufnehmen zu können. Die Funktion setzt einen Puffer der Länge von 16 Bytes voraus.

Funktion USB_RFID_Prod_No

```
int pascal USB_RFID_Prod_No
  (WORD PortNumber, DWORD & Number);
```

Liefert die Produktnummer des RFID-Moduls. Der Rückgabewert `Number` ist eine vorzeichenlose Zahl.

Funktion USB_DIO_Prod_ID

```
int pascal USB_DIO_Prod_ID
  (WORD PortNumber, char * Identification);
```

Liefert den Identifikationstext des RFID-Moduls. Der Rückgabewert `Identification` ist ein Zeiger auf eine null-terminierte Zeichenkette. Der Puffer `Identification` muss vor dem Aufruf der Funktion definiert werden und muss groß genug sein, um alle Zeichen des Rückgabewerts (beispielsweise "`USB-RFID Ver.1-00`") und das Abschlusszeichen aufnehmen zu können. Die Funktion setzt einen Puffer der Länge von 81 Bytes voraus.

Rumpf eines Kommunikationsprogramms

```
#include <stdio.h>
#include <windows.h>

#include "USB-RFID.h"

int main()
   {
   // check the DLL version:
   if ((USB_RFID_Version() & 0xFF00) != 0x010A)
      {
      printf ("Wrong DLL Version: expected 1.10, "
         "found %d.%02d\n", USB_RFID_Version() / 0x100,
         USB_RFID_Version() & 0xFF);
      return -1;
      }

   // define the port number:
   const unsigned short PortNum = 0;

   // open the port, attach it to COM1:
   if (USB_RFID_Open (PortNum, 1))
      {
      // handle a possible error:
      printf ("%s\n", USB_RFID_ErrorMessage (PortNum));
      return 1;
      }

   // setup the device: enable the RFID
   if (USB_RFID_EnableRFID (PortNum, TRUE))
      {
      // handle a possible error:
      printf ("%s\n", USB_RFID_ErrorMessage (PortNum));
      return 2;
      }

   // main program loop:
   while (true)
      {
      // check for automatic data:
      WORD CommandID;
      const int Result =
         USB_RFID_CheckAutoInput (PortNum, CommandID);

      if (Result != 0) // handle possible errors
         {
         if (Result < 0)
            printf ("%s\n", USB_RFID_ErrorMessage (PortNum));
         else
            printf ("Unexpected command received\n");
         continue; // do not proceed on errors
```

```
      }

   // check if RFID data is ready
   if (CommandID & USB_RFID_AUTO)
      {
      BYTE Data [8];
      if (USB_RFID_GetLastRFID (PortNum, Data))
         printf ("%s\n", USB_RFID_ErrorMessage (PortNum));
      else
         {
         // handle RFID data here
         }
      }

   // insert further cases here

   // insert service of other commands here

   }

//finish the program:
return 0;
}
```