

RFID-Leseinheit mit analogen und digitalen Ein- und Ausgängen, LCD Anzeige, Tastenfeld und Signalgeber

Version 3.00



Bedienungsanleitung

Dokument-Version A, erstellt am 25.11.2009

Inhalt

Technische Daten	5
Charakteristik	5
Schnittstellen.....	5
Anzeige	5
Tastenfeld	5
RFID-Leseeinheit	6
Digitale Aus- und Eingänge	6
Analoge Aus- und Eingänge	6
Signalgeber	6
Stromversorgung, Version 12 V	7
Stromversorgung, Version 24 V	7
Allgemein	7
Lieferungsumfang	8
Beschreibung	9
Anschlüsse und Konfigurationselemente.....	9
Abb. 1. Anschlüsse und Konfigurationselemente des Moduls.....	9
Schnittstellen.....	10
Tab. 1. Wahl der Schnittstelle durch Kurzschlussbrücken JP3 und JP4.....	10
Tab. 2. Pinbelegung des RS-232-Steckers CN9	12
Stromversorgung.....	13
RFID-Leseeinheit	14
Anzeige	14
Tastenfeld	15
Signalgeber	15
Analoge Ein- und Ausgänge	16
Tab. 3. Pinbelegung der D-SUB-Buchse CN3.....	16
Digitale Ein- und Ausgänge	17
Abb. 2. Beschaltung der Eingänge (links) und der Ausgänge (rechts).....	18
Tab. 4. Pinbelegung der D-SUB-Buchse CN7	19
Inbetriebnahme	21
Schaltbeispiele	25

Abb. 3. Anschluss eines Schalters oder Tasters (SW1, SW2) und eines Umschalters (SW3) an die digitalen Eingänge. Der Schalter SW1 nutzt die an dem Stecker CN7 bereitgestellte Stromversorgung des RFID-Moduls, die restlichen Schalter SW2 und SW3 schalten an die Eingänge eine Spannung aus einem externen Netzteil. Der Schalter SW1 ist durch die Beschaltung mit dem RFID-Modul galvanisch verbunden, die Schalter SW2 und SW3 sind dagegen von dem RFID-Modul galvanisch isoliert.....	25
Abb. 4. Anschluss eines photoelektrischen Näherungsschalters der Firma Contrinex an die digitalen Eingänge.....	26
Abb. 5. Anschluss einer Kontrollleuchte an die digitalen Ausgänge.....	27
Treiberinstallation.....	28
Installation des virtuellen Ports für die USB-Schnittstelle.....	28
Installation des virtuellen Ports für die LAN-Schnittstelle.....	28
Installation der RS-232-Schnittstellenkarte.....	29
Softwareschnittstelle.....	31
Funktion der Softwareschnittstelle.....	31
Tab. 5. Rückgabewerte der Funktionen.....	32
Tab. 6. I/O-Fehler.....	33
Steuerung der Kommunikation.....	35
Auslesen der analogen Eingänge.....	37
Tab. 7. Konstanten für die Wahl der analogen Eingänge.....	37
Ansteuerung der analogen Ausgänge.....	39
Tab. 8. Konstante für die analogen Anschlüsse.....	39
Auslesen der digitalen Eingänge.....	41
Tab. 9. Konstanten für die digitalen Eingänge.....	41
Ansteuerung der digitalen Ausgänge.....	43
Tab. 10. Konstanten für die digitalen Ausgänge.....	43
Tastenfeld.....	44
Tab. 11. Zuordnung der Tasten und der Bitmaske.....	44
Tab. 12. Konstanten für das Tastenfeld.....	46
RFID-Leseinheit.....	47

Tab. 13. Konstanten für die RFID-Leseinheit	47
Signalgeber	49
Tab. 14. Konstanten für den Signalgeber.	50
LCD Anzeige	51
Tab. 15. Konstanten für die LCD Anzeige.	51
Tab. 16. Der Zeichensatz.....	56
Automatische Datenübermittlung.....	59
Tab. 17. Bitmasken des automatischen Datenempfangs.	59
Tab. 18. Rückgabewerte der Funktion COM_RFID_CheckAutoInput.	60
Fehlerbehandlung	62
Verschiedenes	63
Rumpf eines Kommunikationsprogramms.....	65
Beispiel eines Kommunikationsprogramms zur Bedienung der digitalen Ein- und Ausgänge.....	67

Technische Daten

Charakteristik

- Leseeinheit für RFID-Transponder
- LCD Anzeige 128x64 Pixel
- Tastenfeld mit 12 Tasten
- vier analoge Eingänge, ein analoger Ausgang
- vier digitale Eingänge, vier digitale Ausgänge
- programmierbarer akustischer Signalgeber
- Schnittstellen: RS-232, USB, LAN
- 16-Bit Mikrocontroller
- Stromversorgung aus einem externen Netzteil (12 V- und 24 V-Versionen) oder aus der RS-232-Schnittstelle (12 V-Version)
- ABS Gehäuse

Schnittstellen

- serielle Schnittstelle nach dem RS-232-Standard
Anschluss: D-SUB-Stecker, 9 Pins
Datenübertragungsrate: 115,2 kBaud
Schnittstellenparameter: 7 Datenbits, 1 Stopbit, gerade Parität
- USB-Schnittstelle nach dem Standard USB 2.0
Anschluss: USB-Buchse Typ B
Datenübertragungsrate: bis zu 12 MBit/s (*Full Speed*)
- LAN-Schnittstelle nach dem 100BaseT Ethernet-Standard
Anschluss: RJ-45
Datenübertragungsrate: 10 oder 100 MBit/s

Anzeige

- LCD Anzeige 128x64 Pixel
- Pixelgröße: 0,5 mm
- Pixelfarbe: gelb, Hintergrund: blau
- Hintergrundbeleuchtung: Intensität frei programmierbar

Tastensfeld

- 12 Tasten, Beschriftung frei wählbar
- Anordnung: Matrix mit 3 Reihen und 4 Spalten
- dekodierbare Tastenkombination:
garantiert zwei gleichzeitig gedrückte Tasten

RFID-Leseinheit

- Arbeitsfrequenz: 125 kHz
- einsetzbare Transponder: EM 4102
Zugriff: nur lesen,
nutzbare Speicherkapazität: 40 Bit = 5 Byte,
Sicherheit: CRC-Check
- Leseabstand für ISO-Karten: ca. 10 cm[†]

Digitale Aus- und Eingänge

- Anschluss: Buchse D-SUB-25
- Nennspannung der Eingänge: 12 V= oder 24 V=
(je nach der Spannungsversion)
- Schaltleistung der Ausgänge: 150 V= / 125 V~, 1 A, 30 W / 60 VA

Analoge Aus- und Eingänge

- Anschluss: Buchse D-SUB-9
- Messbereich der Eingänge: 0...2,5 V
- maximal erlaubte Eingangsspannung: 3,5 V
- Messverfahren: Mittelwert aus 4 Messungen im Zeitabstand
von 4,15 ms (60,2 Messwerte pro Sekunde)
- Spannungsbereich des Ausgangs: 0...2,5 V
- Auflösung: 12 Bit
- Referenzspannung: 2,5 V

Signalgeber

- interner Lautsprecher: ø40 mm, 200 Hz...20 kHz
- Ausgabe: frei programmierbarer Ton ca. 62 Hz...5,6 kHz
- Anschluss für einen externen Lautsprecher:
Buchse für ø3,5 mm Mono-Klinkenstecker,
Kurzschlussfest, Innenwiderstand: 10 Ω

[†] 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.

Stromversorgung, Version 12 V

- Externes Netzteil:
Nennspannung: 13 bis 25 V=
Anschluss:
Hohlstecker mit Stift $\varnothing 2,1$ mm,
Polarität: Innen +, Außen –
- RS-232-Schnittstelle:
Anschluss: Pin 9 des D-SUB-Steckers
Nennspannung: 12 V=
- Stromverbrauch:
50 mA typ. mit deaktivierten Ein- und Ausgängen,
Signalgeber, RFID-Empfänger und LAN-Adapter
350 mA max. mit deaktiviertem LAN-Adapter
400 mA max. mit aktivem LAN-Adapter
- Verpolungs- und Überspannungsschutz

Stromversorgung, Version 24 V[‡]

- Externes Netzteil:
Nennspannung: 24 V=
Anschluss: wie bei der 12 V-Variante
- Stromverbrauch:
53 mA (1,25 W) typ. mit deaktivierten Ein- und Ausgängen,
Signalgeber, RFID-Empfänger und LAN-Adapter
190 mA (4,5 W) max. mit deaktiviertem LAN-Adapter
210 mA (5,0 W) max. mit aktivem LAN-Adapter
- Verpolungs- und Überspannungsschutz

Allgemein

- ABS-Kunststoffgehäuse
Abmessungen: 195 x 151,5 x 44 mm³
(Länge x Breite x Höhe ohne Befestigungslaschen)
- Gewicht: ca. 450 g

[‡] Das Gerät in der 24 V-Variante kann ebenfalls wie die 12 V-Variante aus einer Schnittstellenkarte mit 12 V oder aus einem externen Netzteil mit 13-24 V versorgt werden. Bei Versorgungsspannungen kleiner als 24 V kann die Funktion der digitalen Eingänge jedoch nicht garantiert werden, wenn sie aus dieser Spannung versorgt werden.

Lieferungsumfang

- komplett getestetes Modul COM–RFID–DIO4+4–AIO4
- Diagnosesoftware
- Bedienungsanleitung
- optional:
 - trittfestes RS-232-Anschlusskabel 2x D-SUB-Buchse, 3 Pin
 - USB-Anschlusskabel A-Stecker auf B-Stecker
 - LAN-Kabel
 - externes Netzteil 12 V=, 500 mA
 - externes Netzteil 24 V=, 400 mA
 - RS-232-Schnittstellenkarte mit 1 bis 8 Schnittstellen
 - USB-2.0-Schnittstellenkarte
 - LAN-Adapterkarte

Beschreibung

Anschlüsse und Konfigurationselemente

Das RFID-Modul verfügt über externe Steckverbinder für die Daten-Schnittstellen, analoge und digitale Ein- und Ausgänge, einen externen Lautsprecher und die Stromversorgung (siehe Abb. 1). Weiterhin befinden sich in dem Gerät Steckverbinder für die Anzeige, das Tastenfeld und zur Diagnose des Mikrocontrollers. Nach dem Öffnen des Moduls sind ferner mehrere Kurzschlussbrücken zugänglich, mit denen das Gerät konfiguriert werden kann oder die Firmware aktualisiert werden. Die Einstellung der Kurzschlussbrücken sowie die detaillierte Beschreibung der einzelnen Funktionsgruppen befindet sich in den

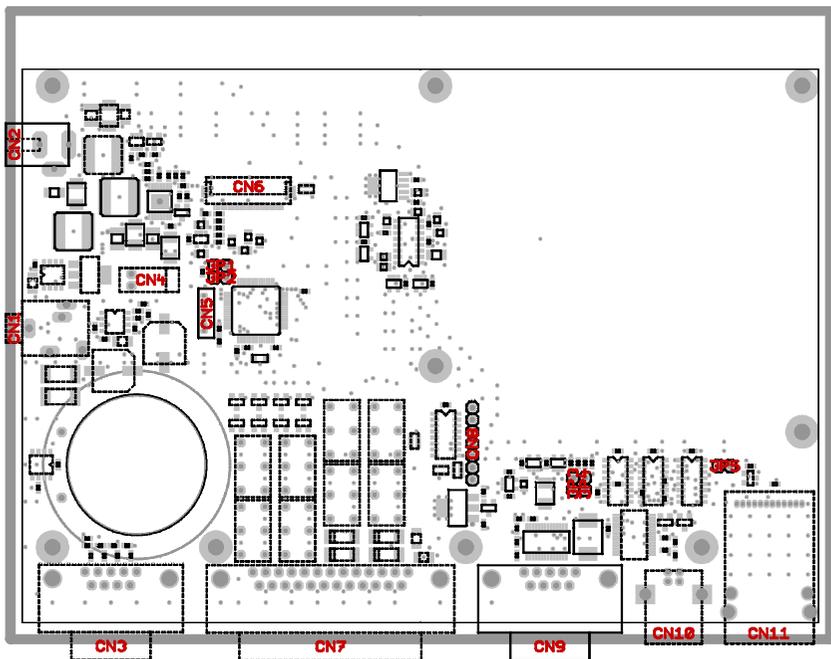


Abb. 1. Anschlüsse und Konfigurationselemente des Moduls.

CN1: Klinkenstecker-Buchse für einen externen Signalgeber, CN2: Hohlstecker für das externe Netzteil, CN3: D-SUB-Buchse für die analogen Ein- und Ausgänge, CN7: D-SUB-Buchse für die digitalen Ein- und Ausgänge, CN9: D-SUB-Stecker für die RS-232-Schnittstelle, CN10: USB-Anschluss, CN11: LAN-Anschluss, JP1, JP2: Kurzschlussbrücken zum Programmieren des Mikrocontrollers, JP3, JP4: Kurzschlussbrücken zur Schnittstellenwahl, JP5: Kurzschlussbrücke zum Programmieren des LAN-Adapters.

nächsten Abschnitten.

! **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 Blechschrauben, ihr Halt im Gehäuse wird bei jedem Zerlegen des Moduls verringert. Der Deckel muss nach der Entfernung der Schrauben zuerst im hinteren Teil angehoben werden und erst danach vorne bei den Steckverbindern gelöst werden. Der Deckel kann zunächst nur geringfügig von dem Bodenteil entfernt werden, ansonsten können die Verbindungskabel zur Anzeige und zum Tastenfeld beschädigt werden. Der Abstand reicht aber in der Regel zur vorsichtigen Manipulation mit den Kurzschlussbrücken. Um den Deckel komplett vom Bodenteil zu entfernen, müssen die Kabel von den Steckern gelöst werden.

Schnittstellen

Das Modul verfügt über eine serielle Schnittstelle nach dem RS-232-Standard, eine USB- und eine LAN-Schnittstelle. Diese Schnittstellen dienen der Datenübertragung zwischen dem Modul und einem Steuerrechner (üblicherweise einem PC). Die Datenübertragungsrates beträgt 115,2 kBaud, wodurch auch ein Transfer der für die serielle Schnittstelle umfangreicheren Displaydaten in einer kurzen Zeit erfolgen kann. Die maximale Datenmenge der gesamten Anzeige beträgt

Tab. 1. Wahl der Schnittstelle durch Kurzschlussbrücken JP3 und JP4

JP3	JP4	Schnittstelle
geschlossen	geschlossen	USB
offen	geschlossen	-
geschlossen	offen	LAN
offen	offen	RS-232

1 KB, diese Daten können mit Berücksichtigung der Datenkodierung im idealen Falle innerhalb von 200 ms übertragen werden.

Welche der drei Schnittstellen für die Kommunikation verwendet wird, wird durch die Kurzschlussbrücken JP3 und JP4 festgelegt (siehe Abb. 1 und Tab. 1). Wird die LAN-Schnittstelle ausgewählt, wird gleichzeitig ihre Stromversorgung aktiviert. Dabei leuchtet eine grüne Kontrollleuchte an dem LAN-Stecker auf, der Stromverbrauch des Moduls steigt und das Gehäuse kann sich in der Nähe des LAN-Steckers nach einer Weile erwärmen. Die USB-Schnittstelle wird ebenfalls nur dann aktiviert, wenn sie ausgewählt wird; der angeschlossene Steuerrechner wird sie auch nur dann erkennen. Die RS-232-Schnittstelle ist permanent aktiv, wird jedoch nur dann von dem Mikrocontroller bedient, wenn sie ausgewählt wird. Ihr Stromverbrauch und auch der Stromverbrauch der USB-Schnittstelle ist im Vergleich mit der LAN-Schnittstelle gering und bewirkt im normalen Betrieb keine erkennbare Erwärmung.

Die RS-232- und die USB-Schnittstellen erfordern 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. Die LAN-Schnittstelle bietet in diesen beiden Aspekten einen gravierenden Vorteil, denn die Entfernung zwischen dem Steuerrechner und dem Modul ist praktisch unbegrenzt und das Modul ist durch die LAN-Schnittstelle von dem Datenkabel isoliert. Dies ermöglicht, dass das Modul an frei wählbare Referenzpotentiale angeschlossen werden kann, und ferner besteht keine Gefahr von Erdschleifen, welche die Messung oder Erzeugung der Analogspannungen stören würden.

Wird die USB- oder die LAN-Schnittstelle genutzt, wird dazu an der PC-Seite ein virtueller Porttreiber installiert, welcher einen virtuellen COM-Port erstellt (siehe Abschnitt "Schnittstellen"). Die Kommunikation mit dem RFID-Modul erfolgt über diesen Port, so dass für alle drei Schnittstellenarten eine identische Softwareschnittstelle verwendet werden kann. Die Schnittstellenadapter kommunizieren intern mit dem Mikrocontroller mit der o.g. Datenübertragungsrate von 115,2 kBaud, die Datenübertragung kann auch über eine schnellere externe Schnittstelle wie LAN nicht schneller stattfinden als mit der internen Datenrate, durch diverse Latenzzeiten ist die resultierende Datenrate sogar geringer als über die RS-232-Schnittstelle.

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 (CN10 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 nicht, diese wird nur abgetastet, um den USB-Controller zu aktivieren, wenn ein USB-Host angeschlossen wird. Der USB-Controller kann nur dann aktiviert werden, wenn die USB-Schnittstelle durch die Kurzschlussbrücken ausgewählt wird.

Die LAN-Schnittstelle entspricht dem Standard *100BaseT Ethernet*, die Datenübertragungsrate beträgt 10 oder 100 MBit/s, der Anschluss ist ein RJ-45 Stecker (CN11 in Abb. 1). Da die LAN-Schnittstelle erst dann versorgt wird, wenn sie durch die Kurzschlussbrücken ausgewählt wird, kann der LAN-Adapter nur in diesem Falle mit dem LAN

Tab. 2. Pinbelegung des RS-232-Steckers CN9

Pin	Bezeichnung	Funktion
1	DCD	nicht verwendet
2	RxD	Datenleitung Modul → PC
3	TxD	Datenleitung Modul ← PC
4	DTR	intern angeschlossen an DSR, optional
5	GND	Masse, Abschirmung der Leitungen
6	DSR	intern verbunden mit DTR, optional
7	RTS	intern angeschlossen an CTS, optional
8	CTS	intern verbunden mit RTS, optional
9	RI	Stromversorgung +12V, optional

Bemerkungen:

1. Das Datenkabel soll für eine höhere Kommunikationssicherheit abgeschirmt sein. Die Abschirmung soll mit der Masse (Pin 5) verbunden sein.
2. Die Handshake-Leitungen DTR (Pin 4) und RTS (Pin 7) werden nur zum Hochladen einer neuen Firmware benutzt. Die Leitungen DSR (Pin 6) und CTS (Pin 8) werden von dem Modul nicht verwendet und sind daher nicht erforderlich.
3. Die Leitung RI (Pin 9) ist nur dann notwendig, wenn das Modul über die Schnittstellenkarte versorgt wird.

kommunizieren und beispielsweise die TCP/IP-Adresse automatisch beziehen. Ebenfalls die Kontrollleuchten des LAN-Adapters sind nur dann funktionsfähig, wenn die LAN-Schnittstelle ausgewählt wurde.

Der LAN-Anschluss besitzt mehrere Kontrollleuchten: Auf der rechten Seite signalisiert die grüne Leuchtdiode, dass ein aktives LAN-Kabel angeschlossen wurde, sie blinkt, wenn Daten übertragen werden. Die gelbe Leuchtdiode zeigt an, dass die Datenübertragung mit 100 MBit/s stattfindet. Die grüne Leuchtdiode auf der linken Seite meldet, dass der LAN-Adapter aktiv ist, die rote Leuchtdiode zeigt Fehler in der Datenübertragung.

Die serielle Schnittstelle entspricht dem RS-232-Standard, sie nutzt einen D-SUB-Stecker mit 9 Pins (CN9 in Abb. 1). Die Pinbelegung des Steckers zeigt Tabelle 2. Aufgrund der hohen Datenrate soll ein hochwertiges abgeschirmtes Datenkabel eingesetzt werden. Tabelle 2 zeigt die Leitungen, welche für die Funktion des Moduls erforderlich sind; die Leitungen verbinden gleichnamige Pins beider D-SUB-Buchsen. Das Datenkabel kann ein 1:1-Kabel mit zwei D-SUB-Buchsen mit 9 Pins sein, für die Kommunikation reicht jedoch ein einfacheres Kabel mit zwei Adern. Das komplett bestückte Kabel ist nur dann erforderlich, wenn die Firmware des Moduls über die RS-232-Schnittstelle hochgeladen werden soll.[§]

Stromversorgung

Zur Versorgung des Moduls wird an den Hohlstecker CN2 in Abb. 1 ein externes Netzteil mit der Gleichspannung je nach der Spannungsvariante angeschlossen. Optional kann die Stromversorgung des Moduls über die RS-232-Schnittstelle erfolgen. Die Versorgungsspannung von ca. 12 V wird an den Pin 9 des D-SUB-Steckers CN9 angelegt. Um diese Option verwenden zu können, muss das Modul an eine spezielle Schnittstellenkarte angeschlossen werden, die über die Möglichkeit einer Stromversorgung verfügt (für weitere Details siehe den Abschnitt "Installation der RS-232-Schnittstellenkarte").** Das Umschalten zwischen den beiden möglichen Stromversorgungsarten

[§] Das Hochladen einer neuen Firmware in das RFID-Modul soll nicht direkt vom Kunden durchgeführt werden, wird daher in der Bedienungsanleitung nicht beschrieben.

** Die Stromversorgung aus der Schnittstellenkarte kann nur bei der 12 V-Variante ohne Funktionseinschränkungen verwendet werden. Bei der 24 V-Variante steht dabei an der Buchse CN7 eine für die digitalen Eingänge nicht ausreichende Spannung zur Verfügung.

erfolgt völlig automatisch: Werden Versorgungsspannungen an beide Stecker angelegt, verwendet das Modul je nach deren Höhe nur eine der beiden Versorgungsspannungen. Wenn die Schnittstellenkarte die Stromversorgung über den Pin 9 nicht unterstützt, wird sie von dem Modul nicht beeinflusst, auch wenn das Anschlusskabel die Pins 9 des Moduls und der Schnittstellenkarte verbindet.

RFID-Leseinheit

Die RFID-Leseinheit 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-Leseinheit liegt bei 125 kHz. Die RFID-Spule befindet sich direkt auf der Leiterplatte des Moduls, wodurch eine hervorragende Stabilität der RFID-Leseinheit gewährleistet wird. Die Reichweite ist typischerweise höher als 10 cm und ist damit gut mit RFID-Leseinheiten vergleichbar, welche gewickelte Luftspulen nutzen.

Befindet sich ein Transponder in der Reichweite der RFID-Leseinheit, 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-Leseinheit entfernt wird. Eine direkte Abfrage ist zu jeder Zeit möglich.

Die RFID-Leseinheit 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-Leseinheit benötigt typischerweise einige 100 ms, bis der Code eines vorliegenden Transponders eingelesen wird.

Anzeige

Das RFID-Modul besitzt eine einfarbige graphische LCD-Anzeige mit 128 x 64 Pixeln. Die Anzeige kann im Text- oder im Graphikmodus extern gesteuert werden. Der Steuerrechner kann nach Belieben beide Modi kombinieren, um Text und Graphiksymbole nebeneinander darzustellen. Der implementierte Datensatz nutzt Zeichen mit der Größe 6 x 8 Pixel, verfügbar sind alle ASCII-Zeichen mit den Codes von 32 (Leerzeichen) bis 127 (©, siehe Tab. 16). Weiterhin können

Bitmaps in beide Richtungen transferiert werden, dabei ist jedoch die Datenübertragungsrate zu beachten (siehe Abschnitt "Schnittstellen"). Die Datenkommunikation nutzt hinreichend große Puffer^{††}, so dass jede Darstellung erst nach dem Abschluss des Datentransfers geschieht. Dadurch ergibt sich zwar eine Zeitverzögerung zwischen dem Beginn des Datentransfers und der Darstellung der Daten, diese erfolgt aber mit der vollen Geschwindigkeit des Mikrocontrollers ohne störendes Blinken oder Flimmern.

Die Hintergrundbeleuchtung der Anzeige kann ebenfalls extern gesteuert werden. Die Lichtintensität kann stufenlos zwischen 0 und 100% eingestellt werden. Weiterhin kann die Einstellung des Sollwerts mit einer vorgegebenen Zeitkonstante erfolgen, dies ergibt eine allmähliche für den Benutzer angenehmere Lichtanpassung als die sofortige Einstellung der Lichtintensität.

Tastenfeld

Das Modul ist mit einem Tastenfeld mit 12 Tasten ausgestattet. Die Tasten sind mechanisch und auch elektrisch in einer Matrix aus 3 Reihen und 4 Spalten angeordnet. Die elektrische Beschaltung ergibt gewisse Einschränkungen beim Dekodieren des Tastenzustandes. Während eine oder zwei gleichzeitig gedrückte Tasten immer dekodiert werden können, ist ab drei gleichzeitig gedrückten Tasten nicht jede Tastenkombination eindeutig dekodierbar. Diese Einschränkung gilt für die Fälle, wenn drei gedrückte Tasten in den Ecken eines Rechtecks liegen.

Das Tastenfeld wird kontinuierlich abgetastet und der aktuelle Tastenzustand ermittelt. Dieser wird entweder automatisch oder nach Aufforderung zum Steuerrechner übertragen. Der automatische Datentransfer erfolgt nur bei einer Änderung des Tastenzustandes. Eine direkte Abfrage ist zu jeder Zeit möglich.

Signalgeber

Das Modul besitzt einen eingebauten Lautsprecher, welcher frei programmierbare Töne oder einfache Melodien abspielen kann. Alternativ kann an die Klinkenstecker-Buchse CN1 in Abb. 1 ein externer

^{††} Sowohl der Empfangs- als auch der Sendepuffer des Moduls besitzen eine Größe von 3 KB.

Lautsprecher oder ein Verstärker angeschlossen werden. Der interne Lautsprecher wird dadurch ausgeschaltet. Wird in die Buchse CN1 ein unbestückter Klinkenstecker hereingesteckt, wird der Signalgeber gänzlich abgeschaltet.

Der Signalgeber kann extern ein- oder ausgeschaltet werden. Dabei wird sein Verstärker aktiviert oder deaktiviert. Das Aktivieren dauert etwa eine Sekunde, sie soll daher rechtzeitig vor dem Abspielen eines Tons erfolgen, damit der Tonanfang nicht verzerrt wird. Sowohl das Aktivieren als auch das Deaktivieren des Signalgebers kann mit Klickgeräuschen verbunden sein. Der aktivierte Signalgeber kann weiterhin verschiedene Störsignale wiedergeben.

Der Signalgeber kann nur einfache Töne abspielen, die Tonhöhe und auch die Tondauer sind frei programmierbar. Zum Wiedergeben einer Melodie muss der Steuerrechner eine Sequenz der Töne mit der richtigen Zeitverzögerung abspielen lassen.

Analoge Ein- und Ausgänge

Die analogen Ein- und Ausgänge sind zusammen mit der Gerätemasse sowie der Versorgungs- und Referenzspannung an die D-SUB-Buchse CN3 angeschlossen (siehe Tab. 3). Die vier Eingänge werden sequentiell durch einen 12-Bit A/D-Wandler abgetastet, die Messun-

Tab. 3. Pinbelegung der D-SUB-Buchse CN3

Pin	Bezeichnung	Funktion
1	+12V	Versorgungsspannung
2	Gnd	Gerätemasse
3	Gnd	Gerätemasse
4	Gnd	Gerätemasse
5	+Vref	Referenzspannung
6	AIN0	analoger Eingang 0
7	AIN1	analoger Eingang 1
8	AIN2	analoger Eingang 2
9	AIO3	analoger Ein- oder Ausgang 3

gen erfolgen in einem Zeitabstand von 4,15 ms. Um das Rauschen zu unterdrücken, wird aus jeweils 4 Messwerten an jedem Eingang ein Mittelwert berechnet und dieser an den Steuerrechner weitergegeben. Der komplette Messzyklus für alle Eingänge dauert daher 66,4 ms, dies entspricht etwa 15 Messungen pro Sekunde. Der letzte Kanal kann als analoger Ausgang konfiguriert werden. In diesem Falle wird die Spannung durch einen 12-Bit D/A-Wandler erzeugt.

Die analogen Eingänge können positive Spannungen messen, die maximal messbare Spannung beträgt 2,5 V, sie gleicht der Referenzspannung. Der analoge Ausgang besitzt den gleichen Spannungsbereich wie die Eingänge. Die Anschlüsse mit der Versorgungs- und Referenzspannung ermöglichen den Anschluss externer Verstärkerschaltungen zur Anpassung des Signalpegels oder zum Filtern der Signale (siehe Abschnitt "Schaltbeispiele").

! **Vorsicht:** Wird eine Spannung falscher Polarität oder eine Spannung höher als die maximal erlaubte angelegt, kann dies zu einer dauerhaften Beschädigung des gesamten Gerätes führen.

Der A/D-Wandler kann extern ein- oder ausgeschaltet werden. Wenn eingeschaltet, läuft die Messung mit der o.g. Rate von etwa 15 Messungen pro Kanal und pro Sekunde. Die zuletzt gemessenen Werte können zu jeder Zeit abgefragt werden. Weiterhin kann die Übermittlung der Messwerte automatisch erfolgen. Dabei werden nur dann neue Daten verschickt, wenn sich der Messwert geändert hat. Aufgrund des Eigenrauschens des Wandlers geschieht dies jedoch in der Regel mehrmals pro Sekunde. Der Ausgang kann ebenfalls zu jeder Zeit aktiviert oder deaktiviert werden und die Spannung eingestellt werden. Ferner kann die aktuell eingestellte Spannung abgefragt werden.

Digitale Ein- und Ausgänge

Die digitalen Ein- und Ausgänge sind zusammen mit der Gerätemasse und der Versorgungsspannung an die D-SUB-Buchse CN7 angeschlossen. Die Tabelle 4 zeigt die Pinbelegung der Buchse. Die vier digitalen Ausgänge sind als galvanisch getrennte Umschalter ausgeführt. Die vier digitalen Eingänge sind ebenfalls vom Gerät galvanisch getrennt und benötigen zur Aktivierung das Anlegen einer Spannung von 12 oder 24 V der in Tabelle 4 angedeuteten Polarität. Dazu kann entweder die vorliegende Versorgungsspannung oder eine externe Spannung gleicher Polarität und gleichen Wertes genutzt werden.

Abb. 2 zeigt die Beschaltung der Ein- und Ausgänge. Die Ausgänge sind an die Kontakte (Umschalter) des jeweiligen Relais angeschlossenen, das durch das Modul angesteuert wird. Die Abbildung zeigt den Zustand der Kontakte, wenn der jeweilige Kanal inaktiv ist oder wenn die Stromversorgung des Moduls abgeschaltet ist.

Die Eingangsanschlüsse steuern direkt die Spule des jeweiligen Relais. Das Modul ermittelt den Schaltzustand anhand der Kontakte des Relais. Die Eingänge werden durch das Anlegen der entsprechenden Steuerspannung aktiviert, im stromlosen Zustand sind die Eingänge inaktiv. Bei der Beschaltung muss die Polarität der Steuerspannung beachtet werden.

Da beim Ausschalten der Eingänge durch die in der Induktivität der Relais-Spule akkumulierte Energie hohe Spannungen entstehen können (Selbstinduktion), wurden die Eingänge mit einem aus jeweils einer Diode (siehe Abb. 2) bestehenden Überspannungsschutz ausgestattet. Dadurch werden die angeschlossenen Schaltkreise ausreichend geschützt, weitere Schutzmaßnahmen sind auch bei der Verwendung von Halbleiter-Schaltern (beispielsweise von Näherungssensoren) nicht erforderlich.

! Vorsicht: Wird eine Steuerspannung falscher Polarität oder eine Steuerspannung höher als die nominale angelegt, kann dies zu einer dauerhaften Beschädigung des jeweiligen Eingangs oder sogar des gesamten Gerätes führen. Das Anlegen der Steuerspannung ohne angeschlossene Stromversorgung des Moduls ist dagegen ungefährlich und hat keinen negativen Einfluss auf die Zuverlässigkeit oder Lebensdauer des Moduls.

Der Zustand der digitalen Eingänge wird kontinuierlich abgetastet und der aktuelle Relaiszustand ermittelt. Dieser wird entweder automatisch oder nach Aufforderung zum Steuerrechner übertragen. Der automatische Datentransfer erfolgt nur bei einer Änderung des Relaiszustand. Eine direkte Abfrage ist zu jeder Zeit möglich. Die Ausgänge

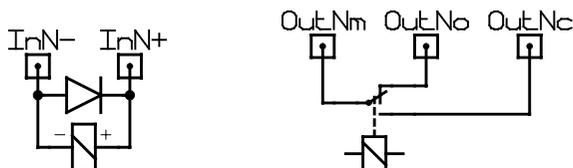


Abb. 2. Beschaltung der Eingänge (links) und der Ausgänge (rechts).

Tab. 4. Pinbelegung der D-SUB-Buchse CN7

Pin	Bezeichnung	Funktion
1	Gnd	Gerätemasse
2	Gnd	Gerätemasse
3	DIN0-	digitaler Eingang 0, negativ
4	DIN1-	digitaler Eingang 1, negativ
5	DIN2-	digitaler Eingang 2, negativ
6	DIN3-	digitaler Eingang 3, negativ
7	Gnd	Gerätemasse
8	DOUT0O	digitaler Ausgang 0, Anschluss o
9	DOUT1O	digitaler Ausgang 1, Anschluss o
10	DOUT1C	digitaler Ausgang 1, Anschluss c
11	DOUT2O	digitaler Ausgang 2, Anschluss o
12	DOUT3O	digitaler Ausgang 3, Anschluss o
13	DOUT3C	digitaler Ausgang 4, Anschluss c
14	+V	Versorgungsspannung +12 V/+24 V
15	+V	Versorgungsspannung +12 V/+24 V
16	DIN0+	digitaler Eingang 0, positiv
17	DIN1+	digitaler Eingang 1, positiv
18	DIN2+	digitaler Eingang 2, positiv
19	DIN3+	digitaler Eingang 3, positiv
20	DOUT0M	digitaler Ausgang 0, Anschluss m
21	DOUT0C	digitaler Ausgang 0, Anschluss c
22	DOUT1M	digitaler Ausgang 1, Anschluss m
23	DOUT2M	digitaler Ausgang 2, Anschluss m
24	DOUT2C	digitaler Ausgang 2, Anschluss c
25	DOUT3M	digitaler Ausgang 3, Anschluss m

können ebenfalls zu jeder Zeit aktiviert oder deaktiviert werden, ferner kann der aktuelle Zustand abgefragt werden.

Inbetriebnahme

Das Modul COM-RFID-DIO4+4-AIO4 benötigt für seine Funktion eine funktionierende RS-232-, USB- oder LAN-Schnittstelle. Für den Anschluss ist ein passendes Datenkabel erforderlich, weiterhin muss das Gerät über ein geeignetes Netzteil verfügen.

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

- Wählen Sie mit Hilfe der Kurzschlussbrücken JP3 und JP4 die Schnittstelle aus, welche für die Kommunikation verwendet werden soll (siehe Abb. 1 und Tab. 1).
- Schalten Sie das Modul ein, die Anzeige muss aufleuchten und für eine kurze Zeit das Firmenlogo erscheinen. Geschieht dies nicht, überprüfen Sie die Stromversorgung. Fahren Sie keineswegs mit der Inbetriebnahme fort, wenn die Anzeige nicht funktioniert. Es deutet auf ein grundlegendes Problem hin, das Modul muss zur Inspektion eingeschickt werden.
- Schließen Sie das Modul mit Hilfe eines Datenkabels an eine passende Schnittstelle des PCs, bzw. an das LAN an. Besitzt der PC keine (freie) serielle Schnittstelle, muss eine Schnittstellenkarte installiert werden (siehe Abschnitt "Installation der RS-232-Schnittstellenkarte"). Soll die USB- oder die LAN-Schnittstelle verwendet werden, muss dazu auf der PC-Seite ein virtueller Porttreiber installiert werden (siehe Abschnitte "Schnittstellen" und "Treiberinstallation").
- Öffnen Sie ein Kommandozeilenfenster, wechseln Sie in das Verzeichnis mit dem Diagnoseprogramm `COM-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 seriellen Schnittstelle angegeben werden, an die das Modul angeschlossen ist. Ist das Modul beispielsweise an die Schnittstelle COM5 angeschlossen, muss das Diagnoseprogramm durch die folgende Anweisung gestartet werden: `"COM-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 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 Datenübertragung 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 Schnittstelle des PCs.
- Funktioniert alles fehlerfrei, können Sie durch Drücken der Tasten 'n', 'd' und 'v' weitere Produktinformationen abfragen.
- Testen Sie die Anzeige, indem Sie die Taste 'g' drücken. Auf der Anzeige muss das Firmenlogo erscheinen, durch Drücken der Taste 'l' kann es wieder gelöscht werden. Drücken Sie anschließend die Taste 'S' und geben einen kurzen Text ein. Dieser muss daraufhin auf der Anzeige erscheinen.
- Drücken Sie die Taste 'K' auf der PC-Tastatur, die automatische Abfrage des Tastenfeldes des Moduls wird gestartet. Drücken Sie verschiedene Tasten auf dem Tastenfeld des Moduls und beobachten die Rückmeldung auf dem PC-Bildschirm. Zum Kodieren der Daten lesen Sie den Abschnitt "Tastenfeld" im Kapitel "Softwareschnittstelle", die Werte sind in der Tabelle 11 zusammengefasst. Stoppen Sie die automatische Abfrage durch Drücken der Taste 'k' auf der PC-Tastatur.
- Schalten Sie den Signalgeber durch Drücken der Taste 't' und durch anschließende Eingabe von 'y' ein. Drücken Sie erneut die Taste 't' und geben jetzt zweimal nacheinander '0' ein. Der interne Lautsprecher fängt an den Kammerton a¹ (Frequenz 440 Hz) abzuspielen. Drücken Sie die Taste 'T' um die Wiedergabe anzuhalten. Für die Kodierung der Tonhöhe siehe Abschnitt "Signalgeber" im Kapitel "Softwareschnittstelle". Schalten Sie anschließend den Signalgeber durch Drücken der Taste 't' und durch darauffolgende Eingabe von 'n' wieder aus.
- Schalten Sie die RFID-Leseinheit durch Drücken der Taste 'F' ein. Drücken Sie die Taste 'R', die automatische Abfrage der RFID-Leseinheit 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 Transpon-

der 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-Leseinheit durch Drücken der Taste 'f' wieder aus.

- Drücken Sie die Taste 'o' und geben Sie einen neuen Wert für die Ausgänge an. Sie können dekadische, durch 'h' endende hexadezimale (beispielsweise A_h) oder durch 'b' endende binäre (beispielsweise 1011_b) Zahlen eingeben. Sie können die Ausgänge einzeln durchtesten, indem Sie Werte von 0001_b bis 1000_b eingeben. Die Schaltgeräusche der Ausgangsrelais kann man dabei gut hören, so dass man auf die Funktion der Ausgänge auch ohne einen externen Anschluss schließen kann. Für die Kodierung der Ausgangswerte siehe Abschnitt "Ansteuerung der digitalen Ausgänge" im Kapitel "Softwareschnittstelle".
- Drücken Sie die Taste 'l', die automatische Abfrage der digitalen Eingänge wird gestartet. Aktivieren Sie die Eingänge durch Anschließen einer externen Spannung von 12 oder 24 V (je nach der Spannungsvariante) oder der an dem Steckverbinder CN7 anliegenden Spannung (siehe Abb. 1 und Tab. 4). Eine Änderung des Zustandes der digitalen Eingänge wird durch das Diagnoseprogramm über die Anzeige der aktuell eingelesenen Werte signalisiert. Für die Kodierung der Ausgangswerte siehe Abschnitt "Auslesen der digitalen Eingänge" im Kapitel "Softwareschnittstelle". Beenden Sie die automatische Abfrage durch Drücken der Taste 'i'.
- Schalten Sie den Analog-Digital-Wandler durch Drücken der Taste 'a' und durch anschließende Eingabe von 'y' ein. Drücken Sie die Taste 'a' und geben '0' ein, das Diagnoseprogramm liest die am analogen Eingang Nr. 0 anliegende Spannung aus. Sind die Eingänge offen, kann der Messwert einen praktisch beliebigen Wert zwischen 0 und 2500 mV besitzen. Schließen Sie eine externe Spannungsquelle an den analogen Eingang Nr. 0 an (Steckverbinder CN3, siehe Abb. 1 und Tab. 3) und stellen die Spannung auf einen Wert zwischen 0 und 2500 mV ein. Achten Sie dabei unbedingt darauf, dass die maximal erlaubte Eingangsspannung nicht überschritten wird (siehe Abschnitt "Technische Daten"). Wiederholen Sie die Messung für verschiedene Spannungen und auch für andere Kanäle. Trennen Sie anschließend die Spannungsquelle von den Eingängen.

- Schalten Sie den analogen Anschluss Nr. 3 durch Drücken der Taste 'c' und durch anschließende Eingabe von 'y' auf Ausgang um. Drücken Sie erneut die Taste 'c' und geben nun eine neue Spannung für den Ausgang ein. Drücken Sie die Taste 'a' und geben '3' ein, das Diagnoseprogramm liest die Spannung aus, welche der Analog-Digital-Wandler an dem analogen Anschluss Nr. 3 gemessen hat. Sie können die Spannung auch direkt mit einem Voltmeter an dem Anschluss AIO3 (Steckverbinder CN3, siehe Abb. 1 und Tab. 3) messen. Probieren Sie den Test für verschiedene Spannungen, der eingegebene und der gemessene Wert dürfen sich dabei um einige Millivolts unterscheiden. Schalten Sie anschließend den Analog-Digital-Wandler durch Drücken der Taste 'a' und durch darauffolgende Eingabe von 'n' wieder aus. Schalten Sie den analogen Anschluss Nr. 3 durch Drücken der Taste 'c' und durch anschließende Eingabe von 'n' erneut auf Eingang um.

Konnten alle Tests fehlerfrei durchgeführt werden, ist das Modul für den Einsatz betriebsbereit. Anschließend kann noch die Kommunikation über eine andere Schnittstelle getestet werden, um auch die restlichen Kommunikationskanäle zu überprüfen. Zu diesem Zweck kann neben dem Programm `COM-RFID-Test.exe` ebenfalls das Diagnoseprogramm `Diagnose.exe` eingesetzt werden (beide im Verzeichnis "Program" des beiliegenden Softwarepakets). Beide Programme werden ähnlich ausgeführt, sie benötigen ähnliche Kommandozeilenparameter: Für eine erfolgreiche Ausführung muss mindestens die Nummer der seriellen Schnittstelle angegeben werden, an die das Modul angeschlossen ist. Das Programm `Diagnose.exe` läuft zwar auf dem PC, die Ein- und Ausgabe erfolgt jedoch auf dem RFID-Modul. Das Programm besitzt eine einfache selbsterklärende Menüführung, der Benutzer kann zwischen verschiedenen Tests des Funktionsgruppen des RFID-Moduls wählen.

Schaltbeispiele

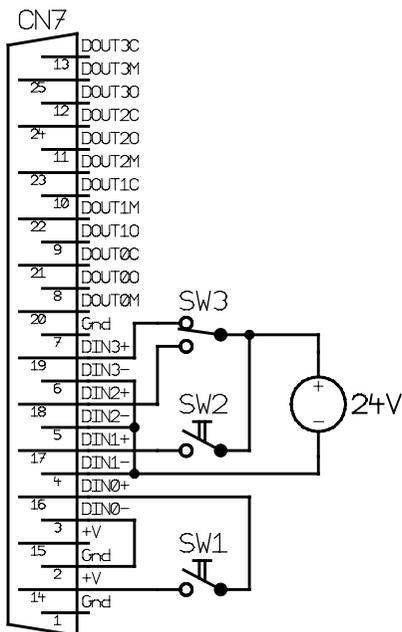


Abb. 3. Anschluss eines Schalters oder Tasters (SW1, SW2) und eines Umschalters (SW3) an die digitalen Eingänge. Der Schalter SW1 nutzt die an dem Stecker CN7 bereitgestellte Stromversorgung des RFID-Moduls, die restlichen Schalter SW2 und SW3 schalten an die Eingänge eine Spannung aus einem externen Netzteil. Der Schalter SW1 ist durch die Beschaltung mit dem RFID-Modul galvanisch verbunden, die Schalter SW2 und SW3 sind dagegen von dem RFID-Modul galvanisch isoliert.

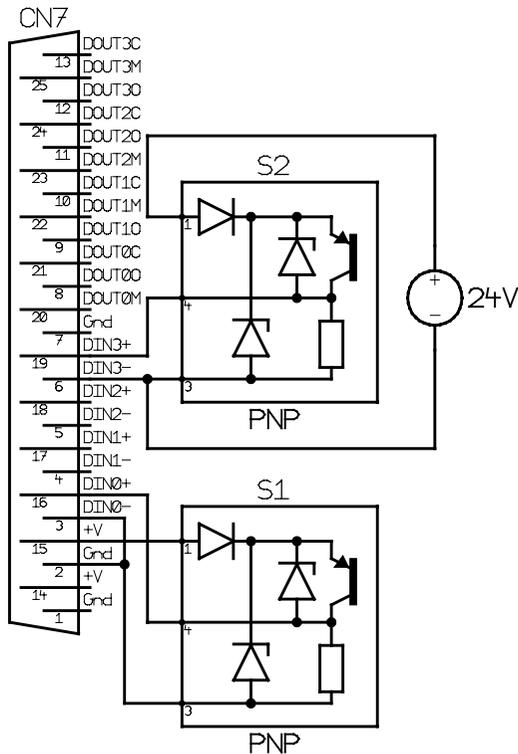


Abb. 4. Anschluss eines photoelektrischen Näherungsschalters der Firma Contrinex an die digitalen Eingänge.

Die Schalter müssen ein PNP-Typ sein. Der Schalter S1 wird über den Stecker CN7 aus dem Netzteil des RFID-Moduls, der Schalter S2 aus einem externen Netzteil versorgt. Der Schalter S1 ist durch die Beschaltung mit dem RFID-Modul galvanisch verbunden, der Schalter S2 ist dagegen von dem RFID-Modul galvanisch isoliert.

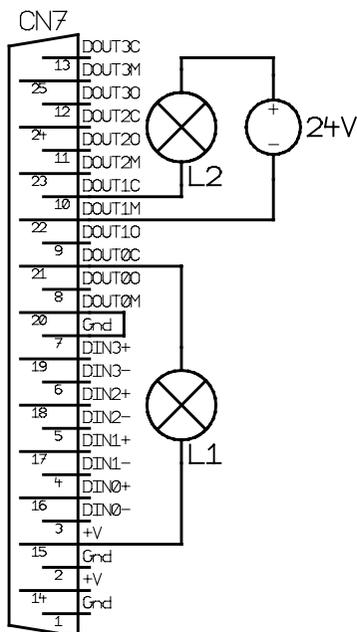


Abb. 5. Anschluss einer Kontrollleuchte an die digitalen Ausgänge. Die Lampen müssen so dimensioniert werden, dass die erlaubte Schaltlast der Kontakte des Moduls nicht überschritten wird. Die Lampe L1 wird über den Stecker CN7 aus dem Netzteil des RFID-Moduls, die Lampe L2 aus einem externen Netzteil versorgt. Beide Netzteile müssen über eine ausreichende Leistung verfügen. Die Lampe L1 ist durch die Beschaltung mit dem RFID-Modul galvanisch verbunden, die Lampe L2 ist dagegen von dem RFID-Modul galvanisch isoliert.

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 "RS232-USB" des beiliegenden Softwarepakets. Beachten Sie dabei folgendes:

- Vor der Installation muss die USB-Schnittstelle durch die Kurzschlussbrücken JP3 und JP4 ausgewählt werden (siehe Abb. 1 und Tab. 1).
- 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 Betriebssystem 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. Dabei muss beachtet werden, dass die Softwareschnittstelle nur Schnittstellen COM1 bis COM9 ansprechen kann. Die Einstellung 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.

Installation des virtuellen Ports für die LAN-Schnittstelle

Der virtuelle Porttreiber ist für den Anschluss des Moduls über die LAN-Schnittstelle erforderlich. Die Installationsdaten befinden sich im Verzeichnis "RS232-LAN" des beiliegenden Softwarepakets. Beachten Sie dabei folgendes:

- Vor der Installation muss die LAN-Schnittstelle durch die Kurzschlussbrücken JP3 und JP4 ausgewählt werden (siehe Abb. 1 und Tab. 1), die Schnittstelle muss an das LAN angeschlossen werden.

Die rechte grüne Leuchtdiode an dem LAN-Adapter muss dabei aufleuchten (siehe Abschnitt "Schnittstellen").

- Alternativ zu den Treibern aus der beiliegenden CD können aktuelle Treiber von der Homepage des Herstellers des LAN-Adapters heruntergeladen werden. Die Treiber werden unter der Adresse <http://tibbo.com/downloads/soi/tdst.html> zur Verfügung gestellt, empfehlenswert ist es nicht den neusten, sondern einen der bewährten Treiber herunterzuladen.
- Zum Installieren der Treiber benötigen Sie Administrationsrechte.
- Bei der Installation sollen die vorgegebenen Werte übernommen werden, es soll jedoch ein vom PC erreichbarer Geräteserver abgeschlossen werden. Bei diesem Schritt soll nach dem Geräteserver gesucht werden und das gefundene Gerät ausgewählt werden. An den aufgelisteten Geräteservern können in diesem Schritt diverse Einstellungen vorgenommen werden. Bei dem LAN-Adapter wird vor der Auslieferung beispielsweise der DHCP-Server aktiviert, d.h. der Geräteserver versucht seine TCP/IP-Adresse automatisch zu beziehen. Soll eine feste Adresse erwünscht sein, kann dies in den Einstellungen geändert werden.
- Nach der Installation kann die Nummer der virtuellen Schnittstelle eingestellt werden. Dabei muss beachtet werden, dass die Softwareschnittstelle nur Schnittstellen COM1 bis COM9 ansprechen kann. Die Einstellung wird im Gerätemanager bei dem Gerät *Virtual 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.
- Nach der Installation wird automatisch beim Anmelden das Programm *Virtual Serial Port Monitor* geöffnet. Das Programm ist für die Funktion des RFID-Moduls nicht erforderlich und sein automatischer Start kann in den Einstellungen verhindert werden.

Installation der RS-232-Schnittstellenkarte

Die RS-232-Schnittstellenkarten werden optional als Zubehör zu dem RFID-Modul geliefert. Die Beschreibung der Installation bezieht sich auf die Schnittstellenkarten der Firma EXSYS. Als Beispiel wird hier die Installation der Karte EX-41052 dargestellt, die zwei serielle Schnittstellen zur Verfügung stellt. Die Installationsdaten befinden sich im Verzeichnis "EX-41052" des beiliegenden Softwarepakets.

Die Installation ist in dem beiliegenden Faltblatt beschrieben. Beachten Sie dabei jedoch folgendes:

- Die Konfiguration der Kurzschlussbrücken auf der Karte soll nur dann geändert werden, wenn die Karte zur optionalen Stromversorgung der RFID-Module verwendet werden soll. In solchem Falle müssen die Kurzschlussbrücken so eingestellt werden, dass die Karte an den Pins 9 der beiden D-SUB-Stecker die Versorgungsspannung von +12 V zur Verfügung stellt.
- Alternativ zu den Treibern aus der beiliegenden CD können aktuelle Treiber von der Homepage des Herstellers der RS-232-Schnittstellenkarte heruntergeladen werden. Die Treiber werden für die o. g. Karte EX-41052 unter der Adresse <http://www.exsys.ch/download/driver/EX-41052.zip> zur Verfügung gestellt.
- Zum Installieren der Treiber benötigen Sie Administrationsrechte.
- Die Installation ist in der sich im jeweiligen Verzeichnis befindlichen pdf-Datei detailliert beschrieben. Lesen Sie diese Beschreibung sorgfältig durch.
- Nach der Installation sollen die Nummern der Schnittstellen eingestellt werden. Standardmäßig stellt die Karte die Schnittstellen COM3 und COM4 zur Verfügung. Aus programmtechnischen Gründen kann es günstig sein, für die Schnittstellen höhere Nummern zu verwenden, um Kollisionen mit anderen seriellen Schnittstellen oder virtuellen Porttreibern zu vermeiden. Dabei muss jedoch beachtet werden, dass die Softwareschnittstelle des RFID-Moduls nur Schnittstellen COM1 bis COM9 ansprechen kann. Die Einstellung wird im Gerätemanager 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 `COM-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 jede Schnittstelle sind selbstverständlich geeignete Hardwaretreiber notwendig, um den Zugriff auf die Hardware überhaupt zu ermöglichen. Wird zur Kommunikation die USB- oder die LAN-Schnittstelle genutzt, müssen weiterhin entsprechende Treiber für die virtuellen COM-Ports installiert werden (siehe Abschnitt "Treiberinstallation").

Die Benutzerfunktionen in der dynamischen Linkbibliothek `COM-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 `COM-RFID.h` enthalten. Sollte Ihr Compiler keine Importbibliothek aus der dynamischen Linkbibliothek erstellen können, steht Ihnen die Datei `COM-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. 5. 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 <code>COM_RFID_CheckAutoInput</code> .
-1	PortNumber out of range	Nummer des Kommunikationskanals <code>PortNumber</code> war größer als maximal zulässig.
-2	Memory for the port data could not be allocated	Bei der Initialisierung konnte dem Kommunikationskanal kein Speicher zugeteilt werden.
-3	Error opening the port	Die Schnittstelle konnte nicht geöffnet werden. Für mögliche Ursachen siehe Tab. 6.
-4	Error closing the port	Die Schnittstelle konnte nicht geschlossen werden. Für mögliche Ursachen siehe Tab. 6.
-5	Error purging the port	Die Puffer der Schnittstelle konnten nicht gelöscht werden. Für mögliche Ursachen siehe Tab. 6.
-6	Error sending command	Die Datenübertragung zum Modul ist fehlgeschlagen. Für mögliche Ursachen siehe Tab. 6.
-7	Error receiving command	Die Datenübertragung vom Modul ist fehlgeschlagen. Für mögliche Ursachen siehe Tab. 6.
-8	Wrong command received	Es wurde ein falsche Antwort vom Modul erhalten.
-9	Wrong argument received	Die vom Modul empfangene Antwort enthält falsche Daten.

Tab. 6. 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.
-2	Port not allocated	Bei der Initialisierung konnte dem Kommunikationskanal kein Speicher zugeteilt werden.
1	Port has not been opened yet	Es wurde versucht, einen Kommunikationskanal zu verwenden, der noch nicht geöffnet wurde.
2	Port has already been opened	Es wurde versucht, einen Kommunikationskanal zu öffnen, der bereits geöffnet wurde.
3	Cannot open the port	Die spezifizierte Schnittstelle konnte nicht geöffnet werden. Die Schnittstelle existiert entweder nicht oder wird von einem anderen Programm verwendet.
4	Cannot get the state of the port	Der Zustand der Schnittstelle konnte nicht ermittelt werden.
5	Cannot set the state of the port	Der Zustand der Schnittstelle konnte nicht gesetzt werden.
6	Cannot set the event mask for the port	Die Ereignismaske für die Schnittstelle konnte nicht gesetzt werden.
7	Cannot set the timeouts for the port	Das Timeout für die Schnittstelle konnte nicht gesetzt werden.
8	Cannot clear the port	Die Puffer der Schnittstelle konnten nicht gelöscht werden.
9	Error reading data from the port	Der Empfang der Daten ist fehlgeschlagen.
10	Error writing data to the port	Das Senden der Daten ist fehlgeschlagen.
11	Wrong data amount written to the port	Es konnte nicht die richtige Anzahl an Zeichen gesendet werden.

Rückgabewert	Fehlermeldung	Ursache
12	Error setting the control lines of the port	Die Steuerleitungen der Schnittstelle konnten nicht angesteuert werden.
13	Error reading the status lines of the port	Der Zustand der Statusleitungen der Schnittstelle konnte nicht ermittelt werden.

vorzeichenlose 16 Bit-Zahl (WORD). Die Nummer 0 bezeichnet den ersten Kanal, die Nummer 7 den letzten.

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 `COM_RFID_State` neu abgefragt werden. Tabelle 5 listet die möglichen Rückgabewerte zusammen mit den Fehlermeldungen auf, wie sie durch die Funktion `COM_RFID_ErrorMessage` zurückgegeben werden. Ist ein Fehler bei der Datenübertragung entstanden, kann die Ursache durch Aufrufen der Funktionen `COM_RFID_IO_State` und `COM_RFID_IO_ErrorMessage` herausgefunden werden. Die erste gibt den letzten I/O-Fehler zurück, die zweite die entsprechende Fehlermeldung (siehe Tab. 6.).

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

Steuerung der Kommunikation

Funktion COM_RFID_Open

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

Öffnet die angegebene Schnittstelle für den Kommunikationskanal und liefert einen Fehlercode nach Tab. 5. 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 COM_RFID_Close

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

Schließt den Kommunikationskanal und liefert einen Fehlercode nach Tab. 5. 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 `COM-RFID.dll` exklusiv benutzt hat, werden alle geöffneten Kommunikationskanäle automatisch geschlossen. Der Programmierer muss in einem solchen Falle keine Sorge dafür tragen.

Funktion COM_RFID_Purge

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

Löscht die Schnittstellenpuffer und liefert einen Fehlercode nach Tab. 5. 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 `COM_RFID_Purge` wird automatisch beim Ausführen der Funktion `COM_RFID_Open` aufgerufen.

Funktion COM_RFID_Buffer_State

```
int pascal COM_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. 5. 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. Dies kann geschehen, wenn das Modul gerade eine größere Datenmenge empfangen hat und die Bearbeitung noch nicht abgeschlossen ist. In einem solchen Falle soll die Abfrage nach wenigen ms wiederholt werden. Die Bearbeitung des gesamten LCD Speichers dauert ca. 50 ms, dies stellt etwa die Obergrenze für die Dauer einer möglichen internen Datenbearbeitung dar.

Funktion COM_RFID_Device_Purge

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

Löscht den Sendepuffer des RFID-Moduls und liefert einen Fehlercode nach Tab. 5. In der Variable `empty` wird der Zustand des Empfangspuffers zurückgegeben (siehe ebenfalls die Funktion `COM_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.

Auslesen der analogen Eingänge

Funktion COM_RFID_EnableAnalogInput

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

Schaltet den Analog-Digital-Wandler für die Analogeingänge ein oder aus. Ist der Wandler ausgeschaltet, werden keine neuen Spannungswerte gemessen.

Nach dem Einschalten des Moduls ist der Analog-Digital-Wandler ausgeschaltet.

Funktion COM_RFID_GetAnalogInput

```
int pascal COM_RFID_GetAnalogInput  
(WORD PortNumber, WORD Channel,  
WORD & Voltage);
```

Gibt den Wert der zuletzt gemessenen Eingangsspannung an dem jeweiligen analogen Anschluss zurück. Die Rückgabe der Spannung erfolgt in Millivolt in der Variable `Voltage`, die Variable `Channel` bestimmt die Nummer des analogen Anschlusses (siehe Tab. 7). Der Messwert der Spannung kann zwischen 0 V und der Referenzspannung liegen (siehe Tab. 8).

Wird diese Funktion aufgerufen, wird die automatische Übermittlung der Messwerte für den spezifizierten analogen Anschluss unterbrochen, welche durch die Funktion `COM_RFID_AutoAnalogInput` gestartet wurde.

Tab. 7. Konstanten für die Wahl der analogen Eingänge.

Wert	Bezeichnung	Beschreibung
0	AIN_0	Analogeingang AIN0
1	AIN_1	Analogeingang AIN1
2	AIN_2	Analogeingang AIN2
3	AIN_3	Analogeingang AIO3
4	AIN_NUM_CHANNEL	Anzahl der analogen Eingänge

Funktion COM_RFID_AutoAnalogInput

```
int pascal COM_RFID_AutoAnalogInput  
(WORD PortNumber, WORD Channel);
```

Startet die automatische Übermittlung der Messwerte für den spezifizierten analogen Anschluss `Channel` (siehe Tab. 7). Die automatische Übermittlung der Messwerte wird durch den Aufruf der Funktion `COM_RFID_GetAnalogInput` für den entsprechenden Anschluss `Channel` wieder unterbrochen.

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

Nach dem Einschalten des Moduls sind alle automatischen Übermittlungen der Messwerte ausgeschaltet.

Funktion COM_RFID_GetLastAnalogInput

```
int pascal COM_RFID_GetLastAnalogInput  
(WORD PortNumber, WORD Channel,  
WORD & Voltage);
```

Gibt analog zu der Funktion `COM_RFID_GetAnalogInput` den Wert der zuletzt automatisch übermittelten Eingangsspannung an dem jeweiligen analogen Anschluss zurück. Die Rückgabe der Spannung erfolgt in Millivolt in der Variable `Voltage`, die Variable `Channel` bestimmt die Nummer des analogen Anschlusses (siehe Tab. 7). Der Messwert der Spannung kann zwischen 0 V und der Referenzspannung liegen (siehe Tab. 8). Für mehrere Details siehe die Beschreibung der Funktion `COM_RFID_CheckAutoInput`.

Ansteuerung der analogen Ausgänge

Funktion COM_RFID_EnableAnalogOutput

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

Konfiguriert den analogen Anschluss AIO3 als Ausgang (`Enable = true`) oder als Eingang (`Enable = false`). Der Analoge Eingang an diesem Anschluss bleibt in beiden Fällen aktiv, wird der Anschluss AIO3 als Ausgang konfiguriert, kann durch den Eingang die ausgegebene Spannung überprüft werden.

Nach dem Einschalten des Moduls ist der Anschluss AIO3 als Eingang konfiguriert.

Funktion COM_RFID_SetAnalogOutput

```
int pascal COM_RFID_SetAnalogOutput
  (WORD PortNumber, WORD Voltage);
```

Setzt die Ausgangsspannung an dem analogen Anschluss AIO3. Die Spannung wird in Millivolt in der Variable `Voltage` übergeben. Die Ausgabe wird erst dann wirksam, wenn der Anschluss durch die Funktion `COM_RFID_EnableAnalogOutput` als Ausgang konfiguriert wird. Die Ausgangsspannung kann auf einen beliebigen Wert zwischen 0 V und der Referenzspannung eingestellt werden (AIO_REF, siehe Tab. 8).

Funktion COM_RFID_GetAnalogOutput

```
int pascal COM_RFID_GetAnalogOutput
  (WORD PortNumber, WORD & Voltage);
```

Gibt den Wert der durch die Funktion `COM_RFID_SetAnalogOutput` zuletzt gesetzten Ausgangsspannung an dem analogen Anschluss AIO3 zurück. Die Rückgabe der Spannung erfolgt in Millivolt in der

Tab. 8. Konstante für die analogen Anschlüsse.

Wert	Bezeichnung	Beschreibung
2500	AIO_REF	Referenzspannung in mV

Variable `Voltage`. Der Rückgabewert kann zwischen 0 V und der Referenzspannung liegen (siehe Tab. 8).

Die Funktion `COM_RFID_GetAnalogOutput` kann verwendet werden, um den Zustand des analogen Ausgangs abzufragen oder den aktuellen Zustand des Moduls zu ermitteln, nachdem die Software neu gestartet wurde.

Auslesen der digitalen Eingänge

Funktion COM_RFID_GetDigitalInput

```
int pascal COM_RFID_GetDigitalInput
(WORD PortNumber, WORD & Value);
```

Liest die digitalen Eingänge aus und speichert den Zustand in die Variable `Value`. Die vier niedrigsten Bits entsprechen dabei jeweils jedem der Eingänge (siehe Tab. 9), der Rückgabewert ist eine Bitmaske, welche die Werte der Eingänge kombiniert. Ist das jeweilige Bit gesetzt (1), ist der Eingang aktiv, d.h. wurde durch eine Spannung angesteuert.

Wird diese Funktion aufgerufen, wird die automatische Übermittlung der Zustände der digitalen Eingänge unterbrochen, welche durch die Funktion `COM_RFID_AutoDigitalInput` gestartet wurde.

Funktion COM_RFID_AutoDigitalInput

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

Startet die automatische Übermittlung der Zustände der digitalen Eingänge aus dem RFID-Modul. Die automatische Übermittlung wird durch den Aufruf der Funktion `COM_RFID_GetDigitalInput` wieder unterbrochen.

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

Tab. 9. Konstanten für die digitalen Eingänge.

Wert	Bezeichnung	Beschreibung
1	DIN_0	Digitaleingang DIN0
2	DIN_1	Digitaleingang DIN1
4	DIN_2	Digitaleingang DIN2
8	DIN_3	Digitaleingang DIN3
4	DIN_NUM_CHANNEL	Anzahl der digitalen Eingänge

Nach dem Einschalten des Moduls sind alle automatischen Übermittlungen ausgeschaltet.

Funktion COM_RFID_GetLastDigitalInput

```
int pascal COM_RFID_GetLastDigitalInput  
(WORD PortNumber, WORD & Value);
```

Gibt analog zu der Funktion `COM_RFID_GetDigitalInput` den Wert des zuletzt automatisch übermittelten Zustandes der digitalen Eingänge zurück. Für mehrere Details siehe die Beschreibung der Funktion `COM_RFID_CheckAutoInput`.

Ansteuerung der digitalen Ausgänge

Funktion COM_RFID_SetDigitalOutput

```
int pascal COM_RFID_SetDigitalOutput
(WORD PortNumber, WORD Value);
```

Steuert die digitalen Ausgänge anhand der Variable `Value` an. Die vier niedrigsten Bits entsprechen dabei jeweils jedem der Ausgänge (siehe Tab. 10), der Wert ist eine Bitmaske, welche die Werte für die Ausgänge kombiniert. Ist das jeweilige Bit gelöscht (0), wird das Ausgangsrelais ausgeschaltet und die Kontakte `m` und `o` des Ausgangs geschlossen. Ist das jeweilige Bit gesetzt (1), wird das Ausgangsrelais aktiviert und die Kontakte `m` und `o` des Ausgangs geschlossen.

Funktion COM_RFID_GetDigitalOutput

```
int pascal COM_RFID_GetDigitalOutput
(WORD PortNumber, WORD & Value);
```

Liest den Zustand der digitalen Eingänge aus und speichert ihn in die Variable `Value`. Die Bits der Variable `Value` haben die gleiche Bedeutung wie bei der Funktion `COM_RFID_SetDigitalOutput`.

Die Funktion `COM_RFID_GetDigitalOutput` kann verwendet werden, um den Zustand der früher gesetzten Ausgänge abzufragen oder den aktuellen Zustand des Moduls zu ermitteln, nachdem die Software neu gestartet wurde.

Tab. 10. Konstanten für die digitalen Ausgänge.

Wert	Bezeichnung	Beschreibung
1	DOUT_0	Digitalausgang DOUT0
2	DOUT_1	Digitalausgang DOUT1
4	DOUT_2	Digitalausgang DOUT2
8	DOUT_3	Digitalausgang DOUT3
4	DOUT_NUM_CHANNEL	Anzahl der digitalen Ausgänge

Tastefeld

Funktion COM_RFID_GetKeypad

```
int pascal COM_RFID_GetKeypad
  (WORD PortNumber, WORD & Value);
```

Gibt den Zustand des Tastenfelds in der Variable `Value` zurück. Die zwölf niedrigsten Bits entsprechen dabei jeweils jeder der Tasten des Tastenfelds. Für die Zuordnung der Bits zu den Tasten siehe Tab. 11, jede Zelle der Tabelle symbolisiert eine Taste des Tastenfeldes (siehe auch Tab. 12).

Das Abtasten des Tastenfelds ermöglicht die Ermittlung von einem gleichzeitigen Tastendruck mindestens zweier Tasten. Werden drei oder mehrere Tasten gedrückt, ist das eindeutige Dekodieren nur dann möglich, wenn sich keine drei gedrückten Tasten in den Ecken eines Rechtecks befinden. Wird eine solche nichtdekodierbare Tastenkombination festgestellt, gibt die Funktion `COM_RFID_GetKeypad` in der Variable `Value` den Wert `-1 (FFFFh)` zurück (`KEY_INVALID`, siehe Tab. 12).

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

Funktion COM_RFID_AutoKeypad

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

Startet die automatische Übermittlung der Zustände des Tastenfelds aus dem RFID-Modul. Die automatische Übermittlung wird durch den Aufruf der Funktion `COM_RFID_GetKeypad` wieder unterbrochen.

Ist die automatische Übermittlung der Tastenfeld-Zustände aktiv, soll

Tab. 11. Zuordnung der Tasten und der Bitmaske.

Bit 2 (004h)	Bit 5 (020h)	Bit 8 (100h)	Bit 11 (800h)
Bit 1 (002h)	Bit 4 (010h)	Bit 7 (080h)	Bit 10 (400h)
Bit 0 (001h)	Bit 3 (008h)	Bit 6 (040h)	Bit 9 (200h)

die Funktion `COM_RFID_CheckAutoInput` periodisch in kurzen Abständen aufgerufen werden, um diese Daten zu bearbeiten.

Nach dem Einschalten des Moduls sind alle automatischen Übermittlungen ausgeschaltet.

Funktion `COM_RFID_GetLastKeypad`

```
int pascal COM_RFID_GetLastKeypad  
    (WORD PortNumber, WORD & Value);
```

Gibt analog zu der Funktion `COM_RFID_GetKeypad` den Wert des zuletzt automatisch übermittelten Zustandes des Tastenfelds zurück. Für mehrere Details siehe die Beschreibung der Funktion `COM_RFID_CheckAutoInput`.

Tab. 12. Konstanten für das Tastenfeld.

Wert	Bezeichnung	Beschreibung
0000h	KEY_NO	Tastencode, wenn keine Taste gedrückt ist
0001h	KEY_L_B	Code der Taste in der unteren Reihe links
0002h	KEY_L_M	Code der Taste in der mittleren Reihe links
0004h	KEY_L_T	Code der Taste in der oberen Reihe links
0008h	KEY_LM_B	Code der Taste in der unteren Reihe in der Mitte links
0010h	KEY_LM_M	Code der Taste in der mittleren Reihe in der Mitte links
0020h	KEY_LM_T	Code der Taste in der oberen Reihe in der Mitte links
0040h	KEY_RM_B	Code der Taste in der unteren Reihe in der Mitte rechts
0080h	KEY_RM_M	Code der Taste in der mittleren Reihe in der Mitte rechts
0100h	KEY_RM_T	Code der Taste in der oberen Reihe in der Mitte rechts
0200h	KEY_R_B	Code der Taste in der unteren Reihe rechts
0400h	KEY_R_M	Code der Taste in der mittleren Reihe rechts
0800h	KEY_R_T	Code der Taste in der oberen Reihe rechts
FFFFh	KEY_INVALID	ungültige Tastenkombination wurde festgestellt, d.h. der Zustand des Tastenfeldes konnte nicht eindeutig dekodiert werden
12	KEY_NUM	Anzahl der Tasten

RFID-Leseinheit

Funktion COM_RFID_EnableRFID

```
int pascal COM_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 COM_RFID_GetRFID

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

Liest die zuletzt empfangenen Daten aus dem RFID-Empfänger 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 Transponder in der Reichweite oder konnte er nicht fehlerfrei deko-

Tab. 13. Konstanten für die RFID-Leseinheit.

Wert	Bezeichnung	Beschreibung
FFh	RFID_INVALID_BYTE	Rückgabewerte für einen nicht dekodierbaren oder nicht vorhandenen Transponder
FFFFFFFFh	RFID_INVALID_DWORD	
-1	RFID_INVALID	

diert 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. 13.

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

Funktion `COM_RFID_AutoRFID`

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

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

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

Nach dem Einschalten des Moduls sind alle automatischen Übermittlungen ausgeschaltet.

Funktion `COM_RFID_GetLastRFID`

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

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

Signalgeber

Funktion COM_RFID_EnableBeeper

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

Schaltet den Verstärker des Signalgebers ein- oder aus. Ist der Verstärker ausgeschaltet, werden keine Töne abgespielt.

Nach dem Einschalten des Moduls ist der Verstärker des Signalgebers ausgeschaltet. Der eingeschaltete Verstärker erhöht den Stromverbrauch des Moduls und dadurch seine Wärmeentwicklung. Wenn der Signalgeber nicht benötigt wird, soll der Verstärker ausgeschaltet werden. Beim Einschalten muss eine Verzögerung von etwa eine Sekunde berücksichtigt werden.

Funktion COM_RFID_PlayTone

```
int pascal COM_RFID_PlayTone  
(WORD PortNumber, short Height,  
WORD Duration);
```

Spielt ein Ton der vorgegebenen Länge und Höhe ab. Die Variable *Duration* bestimmt die Tonlänge in Millisekunden, wird sie auf 0 gesetzt, wird die Tonwiedergabe erst unterbrochen, wenn die Funktion *COM_RFID_StopTone* aufgerufen wird. Die maximale programmierbare Tonlänge beträgt etwa 4 Sekunden (*LENGHT_MAX*, siehe Tab. 14).

Die Tonhöhe wird als eine ganze Zahl in der Variable *Height* übergebenen. Die Zahl bestimmt die Tonhöhe in Halbtönen (d.h. 100 Cent) relativ zum Kammerton a^1 (Frequenz 440 Hz). Beispielsweise eine Oktave entspricht daher einer um 12 höheren oder kleineren Zahl (siehe Tab. 14).

Funktion COM_RFID_StopTone

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

Stoppt die Tonwiedergabe. Diese Funktion kann verwendet werden, um die Wiedergabe eines beliebigen Tons zu unterbrechen, welche durch die Funktion *COM_RFID_PlayTone* gestartet wurde.

Tab. 14. Konstanten für den Signalgeber.

Wert	Bezeichnung	Beschreibung
-9	TONE_c	Ton c ¹
-8	TONE_cis	Ton cis ¹
-8	TONE_des	Ton des ¹
-7	TONE_d	Ton d ¹
-6	TONE_dis	Ton dis ¹
-6	TONE_es	Ton es ¹
-5	TONE_e	Ton e ¹
-4	TONE_f	Ton f ¹
-3	TONE_fis	Ton fis ¹
-3	TONE_ges	Ton ges ¹
-2	TONE_g	Ton g ¹
-1	TONE_gis	Ton gis ¹
-1	TONE_as	Ton as ¹
0	TONE_a	Kammerton a ¹ (Frequenz 440 Hz)
1	TONE_b	Ton b ¹
2	TONE_h	Ton h ¹
12	TONE_OCTAVE	Tonabstand für eine Oktave
-34	TONE_MIN	Minimale erlaubte Tonhöhe
44	TONE_MAX	Maximale erlaubte Tonhöhe
3984	LENGHT_MAX	Maximale erlaubte Tonlänge in ms

LCD Anzeige

Funktion COM_RFID_SetDisplayBrightness

```
int pascal COM_RFID_SetDisplayBrightness
  (WORD PortNumber, WORD Value);
```

Stellt die Helligkeit der LCD Anzeige ein. Die Funktion kann benutzt werden, um die Anzeige an die gegebenen Umstände anzupassen. Die Variable `Value` darf Werte zwischen 0 und 63 (LCD_MAX_BRIGTHESS, siehe Tab. 15) besitzen.

Nach dem Einschalten des Moduls wird der Wert auf 30 eingestellt.

Tab. 15. Konstanten für die LCD Anzeige.

Wert	Bezeichnung	Beschreibung
128	LCD_WIDTH_PIXEL	Breite der LCD Anzeige in Pixel
64	LCD_HEIGHT_PIXEL	Höhe der LCD Anzeige in Pixel
1024	LCD_MEM_SIZE	Größe des LCD Speichers in Bytes
6	LCD_CHAR_WIDTH	Zeichenbreite in Pixel
8	LCD_CHAR_HEIGHT	Zeichenhöhe in Pixel
20h	LCD_CHAR_FIRST	erstes Zeichen im Zeichensatz
7Fh	LCD_CHAR_LAST	letztes Zeichen im Zeichensatz
21	LCD_WIDTH	Breite der LCD Anzeige in Zeichen-Einheiten
8	LCD_HEIGHT	Höhe der LCD Anzeige in Zeichen-Einheiten
3984	LENGHT_MAX	Maximale Zeitkonstante in ms
63	LCD_MAX_BRIGTHESS	Maximal möglicher Wert der Helligkeit (siehe Funktion COM_RFID_SetDisplayBrightness)

Funktion COM_RFID_SetBacklight

```
int pascal COM_RFID_SetBacklight  
(WORD PortNumber, WORD Value);
```

Stellt die Helligkeit der Hintergrundbeleuchtung der LCD Anzeige ein. Die Variable `Value` stellt die Helligkeit in Prozent ein, d.h. sie darf Werte zwischen 0 und 100 besitzen.

Nach dem Einschalten des Moduls wird der Wert auf 100 eingestellt.

Funktion COM_RFID_SetBacklightSmooth

```
int pascal COM_RFID_SetBacklightSmooth  
(WORD PortNumber, WORD Value,  
WORD TimeConst);
```

Stellt die Helligkeit der Hintergrundbeleuchtung der LCD Anzeige ähnlich wie die Funktion `COM_RFID_SetBacklight` ein. Die zusätzliche Variable `TimeConst` stellt die Zeitkonstante für die Einstellung der Helligkeit in Millisekunden ein. Sie darf Werte bis zu etwa 4 Sekunden besitzen (`LENGHT_MAX`, siehe Tab. 15).

Funktion COM_RFID_GetBacklight

```
int pascal COM_RFID_GetBacklight  
(WORD PortNumber, WORD & Value);
```

Gibt die aktuell eingestellte Helligkeit der Hintergrundbeleuchtung der LCD Anzeige zurück. Die Rückgabe der Helligkeit erfolgt in Prozent in der Variable `Value`.

Die Funktion `COM_RFID_GetBacklight` kann verwendet werden, um die früher eingestellte Helligkeit abzufragen oder den aktuellen Zustand des Moduls zu ermitteln, nachdem die Software neu gestartet wurde.

Funktion COM_RFID_ClearDisplay

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

Löscht den Inhalt der LCD Anzeige und setzt die Zeichenposition für die nächste Operation in die obere linke Ecke. Dies ist der Zustand nach dem Einschalten des RFID-Moduls.

Funktion COM_RFID_SetDisplay

```
int pascal COM_RFID_SetDisplay  
(WORD PortNumber, BYTE Pattern);
```

Setzt den gesamten Inhalt der LCD Anzeige auf den vorgegebenen Byte `Pattern` und setzt die Zeichenposition für die nächste Operation in die obere linke Ecke. Die Funktion kann genutzt werden, um verschiedene Hintergrundmuster auf der LCD Anzeige zu erzeugen. Die Anzeige wird zeilenweise gesetzt, analog zum wiederholten Aufrufen der Funktion `COM_RFID_SetDisplayRow`.

Funktion COM_RFID_ClearDisplayRow

```
int pascal COM_RFID_ClearDisplayRow  
(WORD PortNumber, WORD Row);
```

Löscht den Inhalt der Zeichenzeile Nummer `Row` der LCD Anzeige und setzt die Zeichenposition für die nächste Operation auf den Anfang dieser Zeile. Die Zeilennummer darf Werte zwischen 0 für die oberste und 7 (`LCD_HEIGHT-1`, siehe Tab. 15) für die unterste Zeichenzeile besitzen.

Funktion COM_RFID_SetDisplayRow

```
int pascal COM_RFID_SetDisplayRow  
(WORD PortNumber, WORD Row, BYTE Pattern);
```

Setzt den Inhalt der Zeichenzeile Nummer `Row` der LCD Anzeige auf den vorgegebenen Byte `Pattern` und setzt die Zeichenposition für die nächste Operation auf den Anfang dieser Zeile. Die Funktion kann genutzt werden, um ein Muster auf der vorgegebenen Zeichenzeile zu erzeugen. Das Bitmuster `Pattern` entspricht den Pixeln in der gesamten Zeile. Das LSB entspricht der oberen, das MSB der unteren Pixelreihe. Für die Beschreibung der Zeilennummer `Row` siehe die Funktion `COM_RFID_ClearDisplayRow`.

Funktion COM_RFID_SetPixel

```
int pascal COM_RFID_SetPixel  
(WORD PortNumber, WORD Row, WORD Column);
```

Setzt das vorgegebene Pixel. Die Variablen `Row` und `Column` bestimmen die Pixelkoordinaten der LCD Anzeige. Die Zeilennummer

Row darf Werte zwischen 0 für die oberste und 63 (LCD_HEIGHT_PIXEL-1, siehe Tab. 15) für die unterste Pixelzeile, die Spaltennummer Column Werte zwischen 0 für die linkste und 127 (LCD_WIDTH_PIXEL-1, siehe Tab. 15) für die rechteste Pixelspalte besitzen. Die Zeichenposition für die nächste Operation befindet sich eine Pixelreihe rechts von dem gesetzten Pixel.

Funktion COM_RFID_ClearPixel

```
int pascal COM_RFID_ClearPixel  
(WORD PortNumber, WORD Row, WORD Column);
```

Löscht das vorgegebene Pixel. Für die Beschreibung der Pixelkoordinaten Row und Column siehe die Funktion COM_RFID_SetPixel. Die Zeichenposition für die nächste Operation befindet sich eine Pixelreihe rechts von dem gelöschten Pixel.

Funktion COM_RFID_SetDisplayXY

```
int pascal COM_RFID_SetDisplayXY  
(WORD PortNumber, WORD Row, WORD Column);
```

Setzt die Zeichenposition für die nächste Operation. Die beiden Werte sind Zeichenkoordinaten. Die Zeilennummer Row darf Werte zwischen 0 für die oberste und 7 (LCD_HEIGHT-1, siehe Tab. 15) für die unterste Zeichenzeile, die Spaltennummer Column Werte zwischen 0 für die linkste und 20 (LCD_WIDTH-1, siehe Tab. 15) für die rechteste Zeichenspalte besitzen.

Funktion COM_RFID_SetDisplayXYfine

```
int pascal COM_RFID_SetDisplayXYfine  
(WORD PortNumber, WORD Row, WORD Column);
```

Setzt die feine Zeichenposition für die nächste Operation. Die Zeilennummer Row ist eine Zeichenkoordinate, die Spaltennummer Column eine Pixelkoordinate. Die Zeilennummer Row darf Werte zwischen 0 für die oberste und 7 (LCD_HEIGHT-1, siehe Tab. 15) für die unterste Zeichenzeile, die Spaltennummer Column Werte zwischen 0 für die linkste und 127 (LCD_WIDTH_PIXEL-1, siehe Tab. 15) für die rechteste Pixelspalte besitzen.

Funktion COM_RFID_DisplayString

```
int pascal COM_RFID_DisplayString  
(WORD PortNumber, const char * String);
```

Gibt die vorgegebene Zeichenkette auf der Anzeige aus. Die Variable `String` ist ein Zeiger auf die zur Ausgabe bestimmte null-terminierte Zeichenkette.

Die Ausgabe fängt bei der aktuellen Zeichenposition für die nächste Operation an und wird mit jedem ausgegebenen Zeichen um eine Zeichenposition nach rechts verschoben. Ist die Zeichenkette länger, als die jeweilige Zeile darstellen kann, werden nur die darstellbaren Zeichen ausgegeben. Um eine mehrzeilige Ausgabe zu erreichen, müssen die Funktionen `COM_RFID_DisplayString` und `COM_RFID_SetDisplayXY`, bzw. `COM_RFID_SetDisplayXYfine` kombiniert werden.

Der implementierte Datensatz nutzt Zeichen mit der Größe 6 x 8 (LCD_CHAR_WIDTH x LCD_CHAR_HEIGHT, siehe Tab. 15) Pixel. Verfügbar sind alle ASCII-Zeichen mit den Codes von 32 (Leerzeichen) bis 127 (©, siehe Tab. 16).

Funktion COM_RFID_SetDisplayInversion

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

Schaltet die invertierte Zeichenausgabe ein oder aus. Ist die invertierte Ausgabe eingeschaltet, werden nur Zeichen mit der Funktion `COM_RFID_DisplayString` invertiert ausgegeben, alle anderen Funktionen bleiben unbeeinflusst.

Nach dem Einschalten des Moduls ist die invertierte Ausgabe ausgeschaltet.

Funktion COM_RFID_DisplayLogo

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

Gibt das Firmenlogo aus und setzt die Zeichenposition für die nächste Operation in die obere linke Ecke.

Tab. 16. Der Zeichensatz.

Wert	Zeichen	Wert	Zeichen	Wert	Zeichen	Wert	Zeichen
20h		38h	8	50h	P	68h	h
21h	!	39h	9	51h	Q	69h	i
22h	"	3Ah	:	52h	R	6Ah	j
23h	#	3Bh	;	53h	S	6Bh	k
24h	\$	3Ch	<	54h	T	6Ch	l
25h	%	3Dh	=	55h	U	6Dh	m
26h	&	3Eh	>	56h	V	6Eh	n
27h	'	3Fh	?	57h	W	6Fh	o
28h	(40h	@	58h	X	70h	p
29h)	41h	A	59h	Y	71h	q
2Ah	*	42h	B	5Ah	Z	72h	r
2Bh	+	43h	C	5Bh	[73h	s
2Ch	,	44h	D	5Ch	\	74h	t
2Dh	-	45h	E	5Dh]	75h	u
2Eh	.	46h	F	5Eh	^	76h	v
2Fh	/	47h	G	5Fh	_	77h	w
30h	0	48h	H	60h	`	78h	x
31h	1	49h	I	61h	a	79h	y
32h	2	4Ah	J	62h	b	7Ah	z
33h	3	4Bh	K	63h	c	7Bh	{
34h	4	4Ch	L	64h	d	7Ch	
35h	5	4Dh	M	65h	e	7Dh	}
36h	6	4Eh	N	66h	f	7Eh	~
37h	7	4Fh	O	67h	g	7Fh	©

Funktion COM_RFID_WriteDisplayMemory

```
int pascal COM_RFID_WriteDisplayMemory  
(WORD PortNumber, WORD Size,  
const BYTE * Buffer);
```

Beschreibt eine Zeichenzeile mit den vorgegebenen Daten. Die Daten werden auf die Anzeige ab der Zeichenposition für die nächste Operation kopiert. Die Variable `Size` bestimmt die Datenlänge und dadurch auch die Anzahl der kopierten Pixelspalten. Sie darf den Wert zwischen 1 und 128 (`LCD_WIDTH_PIXEL`, siehe Tab. 15) besitzen. Die Daten befinden sich in dem Byte-Array `Buffer`, diese Variable muss vom Benutzer zur Verfügung gestellt werden und die erforderliche Länge besitzen. Das jeweilige Byte wird als eine Pixelspalte kopiert, das LSB entspricht dabei der oberen, das MSB der unteren Pixelreihe.

Funktion COM_RFID_ReadDisplayMemory

```
int pascal COM_RFID_ReadDisplayMemory  
(WORD PortNumber, WORD Size,  
BYTE * Buffer);
```

Liest Daten aus einer Zeichenzeile. Die Daten werden von der Anzeige ab der Zeichenposition für die nächste Operation kopiert. Für die Beschreibung der Parameter siehe die Funktion `COM_RFID_WriteDisplayMemory`.

Funktion COM_RFID_WriteDisplayRow

```
int pascal COM_RFID_WriteDisplayRow  
(WORD PortNumber, WORD Row,  
const BYTE Buffer [LCD_WIDTH_PIXEL]);
```

Beschreibt eine gesamte Zeichenzeile mit den vorgegebenen Daten. Die Variable `Row` bestimmt die Nummer der Zeichenzeile, sie darf Werte zwischen 0 für die oberste und 7 (`LCD_HEIGHT-1`, siehe Tab. 15) für die unterste Zeichenzeile besitzen. Die Daten befinden sich in dem Byte-Array `Buffer`, diese Variable muss vom Benutzer zur Verfügung gestellt werden und die vorgegebene Länge von 128 (`LCD_WIDTH_PIXEL`, siehe Tab. 15) Byte besitzen. Die Funktion arbeitet ähnlich wie die Funktion `COM_RFID_WriteDisplayMemory`.

Funktion COM RFID ReadDisplayRow

```
int pascal COM_RFID_ReadDisplayRow  
(WORD PortNumber, WORD Row,  
BYTE Buffer [LCD_WIDTH_PIXEL]);
```

Liest Daten aus einer gesamten Zeichenzeile. Für die Beschreibung der Parameter siehe die Funktion `COM_RFID_WriteDisplayRow`. Die Funktion arbeitet ähnlich wie die Funktion `COM_RFID_ReadDisplayMemory`.

Funktion COM RFID SetDisplayMemory

```
int pascal COM_RFID_SetDisplayMemory  
(WORD PortNumber, WORD Row, WORD Column,  
WORD Height, WORD Width,  
const BYTE * Buffer);
```

Beschreibt einen rechteckigen Bereich der Anzeige mit den vorgegebenen Daten. Die Variablen `Row` und `Column` bestimmen die linke obere Ecke des Bereichs, die Variablen `Height` und `Width` seine Abmessungen. Die Zeilen sind Zeichenkoordinaten im Bereich 0 bis 7 (`LCD_HEIGHT-1`, siehe Tab. 15), die Spalten Pixelkoordinaten im Bereich 0 bis 127 (`LCD_WIDTH_PIXEL-1`, siehe Tab. 15). Die Daten befinden sich in dem Byte-Array `Buffer`, erste Zeile zuerst. Die Variable `Buffer` muss vom Benutzer zur Verfügung gestellt werden und die erforderliche Länge besitzen. Für die komplette Anzeige sind insgesamt 1024 (`LCD_MEM_SIZE`, siehe Tab. 15) Bytes erforderlich, deren Übertragung kann eine Dauer erfordern, die im Bereich von mehreren 100 ms liegt. Die Funktion arbeitet ähnlich wie die Funktion `COM_RFID_WriteDisplayMemory`.

Funktion COM RFID GetDisplayMemory

```
int pascal COM_RFID_GetDisplayMemory  
(WORD PortNumber, WORD Row, WORD Column,  
WORD Height, WORD Width, BYTE * Buffer);
```

Liest Daten aus einer gesamten Zeichenzeile. Für die Beschreibung der Parameter siehe die Funktion `COM_RFID_SetDisplayMemory`.

Automatische Datenübermittlung

Funktion COM_RFID_CheckAutoInput

```
int pascal COM_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 `COM_RFID_GetAutoMask` empfangen wurden. Der Wert ist eine Bitkombination aus den in der Datei `COM-RFID.h` definierten Werten `AUTO_xxx`. (siehe Tab. 17). 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 17 als Empfangsfunktionen aufgelistet. Werden die Daten durch die entsprechende Empfangsfunktion nicht ab-

Tab. 17. Bitmasken des automatischen Datenempfangs.

Wert	Bezeichnung	Empfangsfunktion	Datenquelle
1	AUTO_AIN_0	COM_RFID_GetLast AnalogInput	Analogeingang AIN0
2	AUTO_AIN_1	COM_RFID_GetLast AnalogInput	Analogeingang AIN1
4	AUTO_AIN_2	COM_RFID_GetLast AnalogInput	Analogeingang AIN2
8	AUTO_AIN_3	COM_RFID_GetLast AnalogInput	Analogeingang AIO3
16	AUTO_DIN	COM_RFID_GetLast DigitalInput	Digitaleingänge
32	AUTO_KEYPAD	COM_RFID_GetLast Keypad	Tastenfeld
64	AUTO_RFID	COM_RFID_GetLast RFID	RFID-Empfänger

geholt, 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 widerspiegelt, welche seit dem letzten Aufruf der Funktionen `COM_RFID_CheckAutoInput` oder `COM_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 `COM_RFID_CheckAutoInput` sind in der Tabelle 18 aufgelistet. Rückgabewerte der Funktion, welche ungleich Null sind, deuten auf einen Fehler hin und bedürfen einer entsprechenden 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 `COM_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 `COM_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 Zeit-

Tab. 18. Rückgabewerte der Funktion `COM_RFID_CheckAutoInput`.

Rückgabewert	Bedeutung
0	Die Schnittstelle hat möglicherweise Daten empfangen, die Bitmaske <code>CommandMask</code> widerspiegelt, ob sie zwischengespeichert wurden
1	Die Schnittstelle hat Daten empfangen, welche keiner automatischen Datenübermittlung entsprechen, die Bitmaske <code>CommandMask</code> widerspiegelt die davor korrekt empfangenen Daten
<0	siehe Tab. 5

auflösung bei Ermittlung diverser Ereignisse. Ständige Aufrufe der Funktion `COM_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 `COM_RFID_CheckAutoInput` in der Regel einmal pro Millisekunde aufgerufen werden, ohne dass dadurch eine nennenswerte Prozessorlast entsteht.

Funktion COM_RFID_GetAutoMask

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

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

Alle Funktionen der Softwareschnittstelle, welche Daten von dem RFID-Modul empfangen, können auch automatisch übermittelte Daten empfangen. Die Funktion `COM_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. 17 und die Beschreibung der Funktion `COM_RFID_CheckAutoInput`).

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

Fehlerbehandlung

Funktion COM_RFID_State

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

Liefert einen Fehlercode nach Tab. 5. 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 COM_RFID_ErrorMessage

```
const char * pascal COM_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. 5 entnommen werden. Die Funktion hat keinen Einfluss auf die Kommunikation und kann zu jeder Zeit aufgerufen werden.

Funktion COM_RFID_IO_State

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

Liefert einen I/O-Fehlercode nach Tab. 6. 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 COM_RFID_IO_ErrorMessage

```
const char * pascal COM_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. 6 entnommen werden. Die Funktion hat keinen Einfluss auf die Kommunikation und kann zu jeder Zeit aufgerufen werden.

Verschiedenes

Funktion COM_RFID_Version

```
WORD pascal COM_RFID_Version();
```

Liefert die Version der Softwareschnittstelle (der dynamischen Linkbibliothek COM-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 COM-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 COM_RFID_GetUptime

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

Liefert die Zeit zurück, 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 1/64 s.

Funktion COM_RFID_FW_Ver

```
int pascal COM_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 `COM_RFID_Version` eine vorzeichenlose 16 Bit-Zahl `Version`. Die dynamische Linkbibliothek `COM-RFID.dll` und die Firmware des RFID-Moduls sind miteinander kompatibel, wenn die Hauptversionsnummern identisch sind.

Funktion `COM_RFID_FW_Date`

```
int pascal COM_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 `COM_RFID_Prod_No`

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

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

Funktion `COM_DIO_Prod_ID`

```
int pascal COM_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 `COM-RFID-DIO4+4-AIO4`) 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 "COM-RFID.h"

int main()
{
    // check the DLL version:
    if ((COM_RFID_Version() & 0xFF00) != 0x0100)
    {
        printf ("Wrong DLL Version: expected 1.x, found %d.%02d\n",
            COM_RFID_Version() / 0x100, COM_RFID_Version() & 0xFF);
        return -1;
    }

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

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

    // insert setup here

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

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

```

```
// check if digital input data is ready
if (CommandID & AUTO_DIN)
{
    WORD Data;
    if (COM_RFID_GetLastDigitalInput (PortNum, Data))
        printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
    else
    {
        // handle digital input data here
    }
}

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

// insert further cases here

// insert service of other commands here

}

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

Beispiel eines Kommunikationsprogramms zur Bedienung der digitalen Ein- und Ausgänge

```

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

#include "COM-RFID.h"

int main()
{
    // check the DLL version:
    if ((COM_RFID_Version() & 0xFF00) != 0x0100)
    {
        printf ("Wrong DLL Version: expected 1.x, found %d.%02d\n",
            COM_RFID_Version() / 0x100, COM_RFID_Version() & 0xFF);
        return -1;
    }

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

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

    // Set automatic transmission of digital input state:
    if (COM_RFID_AutoDigitalInput (PortNum))
        printf ("%s\n", COM_RFID_ErrorMessage (PortNum));

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

        if (Result < 0) // handle possible errors
            printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
        if (Result == 1) printf ("Unexpected command received\n");
    }
}

```

```
if (Result == 0) // if data is ready, print it
{
    if (CommandID & AUTO_DIN) {
        WORD Data;
        if (COM_RFID_GetLastDigitalInput (PortNum, Data))
            printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
        else
            printf ("Digital input data received: "
                "%d (%02Xh)\n", Data, Data);
    }
}

if (kbhit() == 0) continue; // repeat if no key is pressed

const int Key = getch(); // read the pressed key

// set the output state:
if (Key == 'o')
{
    printf ("Enter new hexadecimal output state: ");
    WORD Data; scanf ("%X", &Data);
    if (COM_RFID_SetDigitalOutput (PortNum, Data))
        printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
}

// get identification data:
if (Key == 'i')
{
    // get firmware version:
    WORD Version;
    if (COM_RFID_FW_Ver (PortNum, Version))
        printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
    else
        printf ("Firmware version: %d.%02d\n",
            Version / 0x100, Version & 0xFF);

    // get firmware date:
    char DateString [16];
    if (COM_RFID_FW_Date (PortNum, DateString))
        printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
    else
        printf ("Firmware date: %s\n", DateString);

    // get product identification:
    char IDString [81];
    if (COM_RFID_Prod_ID (PortNum, IDString))
        printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
    else
        printf ("Product identification: %s\n", IDString);
}
```

```
// get product number:
DWORD Number;
if (COM_RFID_Prod_No (PortNum, Number))
    printf ("%s\n", COM_RFID_ErrorMessage (PortNum));
else
    printf ("Product number: %d\n", Number);
}

// stop on 'Esc':
if (Key == 27)
    break;
}

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