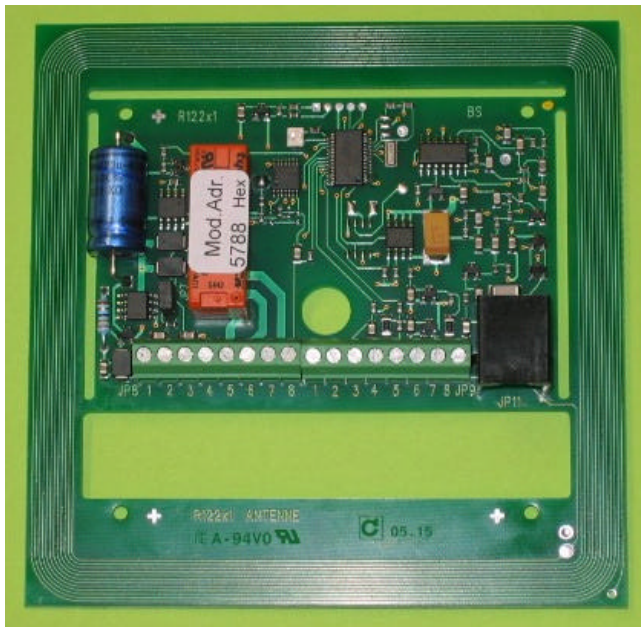


easyident-Q mit RS485 Interface

Für Firmware MT6B14

Art. Nr. FS-0097



Aufbau:

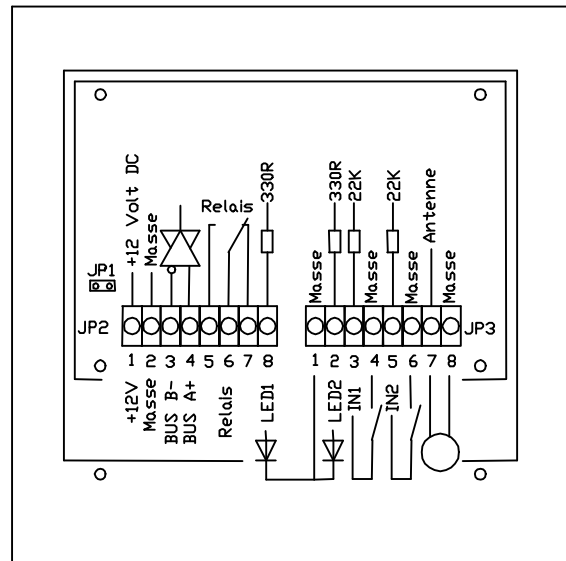


Abb. 1: Ansicht des easyident RS485 BUS Modul

Das easyident-RS-485 B318 Modul ist eine Lesemodul für berührungslose Transpondertechnologie. Die ausgelesenen Transponderdaten können über die RS-485 Schnittstelle abgefragt werden. Die Lesereichweite beträgt je nach Transponder bis zu ca. 70mm. Über die Schnittstelle können alle Zustände des Moduls ausgelesen und gesteuert werden. Unzählige Anwendungen lassen sich mit easyident-RS485 realisieren.

Beispiele: Elektronische Zeiterfassung, Zutrittskontrolle, Kostenabrechnungen für Kantinen, Getränkeautomaten und vieles mehr.

Das easyident-RS-485 Moduls benötigt für die ordnungsgemäße Funktion Transponder Typ H4002 z.B. EM Marin oder kompatible. Jeder Transponder ist ein Unikat. Die Nummer ist fest im Mikrochip programmiert und kann nicht geändert werden. Es stehen (2⁴⁰) verschiedene Codes zur Verfügung. Bei diesen Transpondern sind neben den 40 Nutzdatenbits zusätzlich 14 Paritätsbits eingebettet, um die Datenübertragung zum Leser über-prüfbar zu machen und dadurch Übertragungsfehler zu erkennen.

Die Transponder sind in mehreren Varianten erhältlich.
 ? Schlüsselanhänger 43 x 33 x 8 mm
 ? Scheckkarte nach ISO 7810/7816 .. 85,6 x 54,0 x 0,76 mm

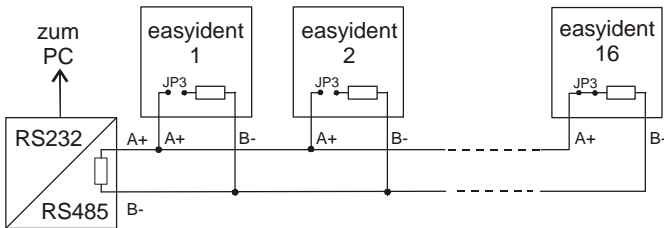
Technische Daten:

Stromaufnahme:	ca. 35 / 80mA
Betriebsspannung:	12V DC +/- 10%
Frequenz:	125 kHz
Anzahl der Codiervarianten:	4.294.967.294
Erkennungszeit:	ca. 0,2 Sekunden
Betriebstemperatur:	-10°C bis +65°C
Leseradius:	ca.60 -100mm
Sendeleistung Antenne:	ca. 150 mW
Maße:	103 * 103 x 15 mm
Zulassungen:	CE

Klemme	Pin	Bezeichnung
JP2	1	+ 12 Volt Versorgungsspannung
JP2	2	Masse
JP2	3	RS-485-BUS Anschluss TB-B
JP2	4	RS-485-BUS Aeschylus TB-A
JP2	5	Relais NO normally open
JP2	6	Relais Common
JP2	7	Relais NC normally closed
JP2	8	LED 1 Output 10 mA (grün)
JP3	1	Masse
JP3	2	LED 2 Output 10 mA (rot) oder (Relais 2I)
JP3	3	TTL Input 1 (Türkontakt)
JP3	4	Masse
JP3	5	TTL Input 2 (Sabotagekontakt)
JP3	6	Masse
JP3	7	Externe Antennen Eingang
JP3	8	Externe Antenne Masse

Tab. 1 : Anschlussbelegung Schraubklemmen JP2 und JP3

Mit dem Steckjumper JP1 kann der RS-485-BUS mit einem eingebauten 120 Ω abgeschlossen werden. Für den ordnungsgemäßen Betrieb muss das letzte Modul den Bus abschließen. In der Abb. 2 ist ein System mit insgesamt 32 Modulen dargestellt. Im Modul #32 ist JP1 gesteckt, bei allen anderen Modulen ist JP1 offen.



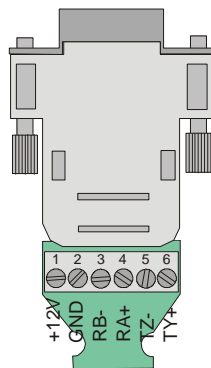
Installation und Hinweise:

- ? Die externe 12 Volt Stromversorgung **muss** den Richtlinien EN 60950 entsprechen.
- ? In die Zuleitung der Versorgungsspannung **muss** eine Sicherung 300mA träge eingebaut sein.
- ? easyident RS-485-BUS ist ein Regel - und Steuergerät der Wirkungsweise Typ1 nach EN60730 (VDE 0631).
- ? Bei Installation muss das easyident RS-485-BUS Modul frei von Verunreinigungen, wie grobem Staub oder Feuchtigkeit, sein.
- ? Die gesamte Leitungslänge eines Systems von RS-485-BUS Modulen kann bis zu 1000 Meter betragen.
- ? Bei Verwendung mehrerer Module ist zu beachten, dass der Abstand zwischen den Modulen **mindestens 1 Meter** sein muss, da sich die Systeme sonst gegenseitig beeinflussen. Als Folge verringert sich die Lesereichweite.
- ? Bei verschiedenen Anwendungen kann es erforderlich sein, Antenne und Elektronik zu trennen. Verbindungen zwischen Elektronik und Antenne sind bis max. 1m zulässig.
- ? In der Nähe der Antenne sind metallische Gegenstände zu meiden, da diese die Lesereichweite reduzieren.
- ? Das Modul ist nur für den stationären Betrieb geeignet.

Konverter A402-0051 Anschluss an das easyident RS485 Modul

Achtung:
Dieser Konverter ist nicht im Lieferumfang des easyident-RS485 enthalten.

Der easyident RS485 Konverter wird auf die COM Schnittstelle des Computers gesteckt und wandelt die RS232 Signale in RS485 BUS Signale um. Verbinden Sie die RS485-BUS-Anschlüsse des Konverters mit dem easyident RS485-BUS Modul wie auf der Zeichnung dargestellt.



Die Anschlussleitungen zwischen dem RS485-BUS Konverter und dem easyident RS485-BUS müssen verdreht sein. Standardmäßig erfolgt die Stromversorgung des RS485 Konverters über die Steuerleitungen RTS und DTR der RS232 Schnittstelle. Wenn Sie jedoch mehrere RS485 Busteilnehmer am RS485-BUS angeschlossen haben und eine Busleitung >100 Meter wendeln, so müssen Sie der Konverter mit Strom versorgen. Verbinden Sie dazu die beiden Schraubklemmen Pin 1 + 2 mit dem mitgelieferten Netzteil.

Konverter Pin 1 = +12 VDC +/- 10% ca. 30 mA
Konverter Pin 2 = Masse

Wenn Sie alle Kabel am RS485 Konverter angeschlossen haben, so sollten Sie diese mit einem Kabelbinder als Zugentlastung sichern. Verwenden Sie dazu die beiden Löcher am ende des Konverters.

Kommunikation:

Die Datenübertragung erfolgt über marktübliche RS485 Bustreiber – Maxim 487. Für die einwandfreie Funktion eines Systems mit easyident RS-485-BUS Modulen ist der **korrekte Busabschluss** am letzten Modul **unbedingt erforderlich**. Das System wird als Single - Master und Multi – Slave betrieben, daher antwortet bei korrekter Einstellung immer nur ein Modul am Bus, es kommt zu keinen Kollisionen. Die Datenübertragung erfolgt asynchron, die Einstellung der Masterschnittstelle ist:

- 9600 Baud
- 1 Startbit
- 8 Datenbits
- 2 Stoppbit

Nach der Aussendung eines Kommandos vom Master muss der Sender des Masters abgeschaltet werden. Das erledigt der Buskonverter IC 485IP-1M automatisch. Für die Dauer der Antwort wird der Sender im Slave (easyident RS-485-BUS Modul) durch dessen Mikroprozessor aktiviert, sendet die Daten und wird nach der Aussendung wieder abgeschaltet.

Hinweis: Aufgrund der Eigenschaft der Bustreiber und des Konverters sind die vom Master gesendeten Zeichen auf dem Empfangspin des RS232 Anschluss als Echo lesbar.

Zusatzfunktionen:

Das easyident-RS485 Modul ist auch offline funktionsfähig. Für die ordnungsgemäße Funktion des *offline* Betrieb müssen die entsprechenden Transponder im internen EEPROM hinterlegt werden. Die programmierte Transpondernummern werden mit den über die Antenne gelesenen Transpondernummern verglichen, bei Gleichheit wird das Relais geschaltet. Die entsprechenden Einstellungen sind unter Timer für *offline* Betrieb setzen - beschrieben. Die Funktion des *offline* Betriebs ist abschaltbar.

Logbuch:

Neben der beschriebenen Relaischaltaktion werden die letzten 26 Transaktionen mit gültigen Transpondern im EEPROM in einem Logbuchbereich hinterlegt und können über die RS485 Schittstelle abgefragt werden.

RS485 BUS Technik:

Wird keinen Standard RS232 to RS485 Bus Konverter benutzt, so muß sichergestellt sein, dass der RS485 BUS im Tristate Mode auf definierten Pegeln liegt. Dies erfolgt mit den beiden 620 Ω Widerständen gegen +5 Volt und Masse. Der 120 Ω Widerstand ist für den Abschluss und darf nur am letzten Modul aktiviert sein. Siehe Abb.2 Seite 1.

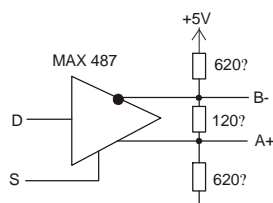


Abb. 4: RS 485 Bustreiber mit Pull-up und Pull down Widerstand

Kommandoübersicht:

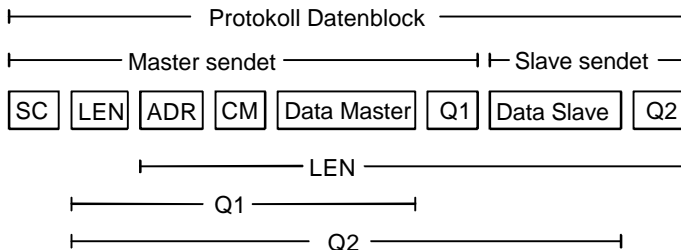
Für die Steuerung und den Datentransfer sind die folgenden Kommandos **CM** im easyident RS-485-BUS Modul implementiert. Die detaillierten Beschreibung folgen im Anschluss an die Übersicht.

CM	Bezeichnung
\$00	Get Version
\$01	Repeat Anser
\$02	Set Status Address
\$02	Reset all Status Address
\$03	Modul Reset
\$33	Global Status Request
\$7A	Get Modul Address
\$7B	Identifizierungskommando
\$7C	Timer für <i>offline</i> Betrieb setzen.
\$80	Get Modul Status
\$81	Set Relais and LED
\$88	Read Card Data
\$A8	Program Modul Address
\$AE	Write EEPROM Data
\$AF	Read EEPROM Data

Tab. 2: Übersicht Kommandos für easyident RS-485-BUS Modul

Aufbau des Protokolls:

Jedes Kommando an ein easyident-RS485 Modul wird in Form eines Datenblocks übermittelt, der folgendermaßen aufgebaut ist.



	Anzahl	Master sendet.	
SC	1	Start Character	Startzeichen \$2A
LEN	1	LENGth	Längenangabe: \$00...\$FF
ADR	2	Modul ADR esse	\$0000...\$FFFF
CM	1	Command Master	Kommando: \$00...\$FF
DM	n	Data Master	Masterdaten
Q1	1	Quersumme Master	1. Prüfsumme
		Slave sendet	
DS	n	Data Slave	Moduldaten
Q2	1	Quersumme Slave	2.Prüfsumme

Abb. 5: Allgemeines Datenprotokoll für easyident RS485 Modul

Datenübertragung:

Alle Übertragungen basieren auf der Grundlage des Protokolls in Abb. 5. Die Anzahl der Bytes **DM** und der **DS** wird durch das jeweilige Kommando bestimmt und liegt zwischen 0 und 8 Bytes. Der Slave führt den Befehl nur unter folgenden Bedingung aus und antwortet entsprechend:

- ? **SC** wurde korrekt empfangen
- ? Der Wert **ADR** entspricht der eigenen Moduladresse oder der Globaladresse \$0000.
- ? Der Befehl **CM** wird vom adressierten Modul unterstützt.
- ? Die Daten **DM** wurden eingelesen
- ? Die eingelesene Quersumme **Q1** ist identisch mit der vom Slave berechneten internen Prüfsumme, d.h. es sind keine Übertragungsfehler aufgetreten
- ? Es ist kein Timeout Fehler aufgetreten.

Wird eine dieser Bedingungen nicht erfüllt, führt das adressierte Modul den Befehl nicht aus und antwortet **nicht** am Bus.

Save Antwort:

Nach korrekt empfangenen Kommandos wird der Slave antworten, um dem Master die angeforderten Daten zu übertragen oder nur den korrekten Empfang der Daten mit der Summe Q2 zu bestätigen. Eine Ausnahme ist der Globale Status Request, bei dem nur ein Byte gesendet wird, sobald der Slave – entsprechend dem Wert der Status Adresse **SA** – an der Reihe ist und einige globaler Kommandos, die unbeantwortet bleiben. Für die Dauer der Antwort wird der Sender im Slavemodul vom Mikroprozessor aktiviert, sendet die Daten und wird nach der Aussendung wieder abgeschaltet.

Hinweis: Kommt innerhalb 200ms keine Antwort vom Slave, so muss die Anforderung vom Master wiederholt werden.

Modulspezifische Kommandos:

Modulspezifische Kommandos sind immer nur für **ein** Busmodul bestimmt. Der Wert **ADR** enthält eine 16 Bit Moduladresse, wobei das höherwertige Byte zuerst gesendet wird. Der Wert **LEN** bestimmt die Länge des gesamten, übertragenen Datenblocks inklusive Daten und Prüfsumme und ist abhängig von dem gesendeten Kommando. Er errechnet sich aus der Summe aller übertragenen Zeichen für ein Kommando, ausgenommen das Startzeichen **SC** und die Prüfsumme **Q2**. Der Wert **CM** enthält das entsprechende Kommando, welches ausgeführt werden soll.

Globale Kommandos:

Globale Kommandos sind immer für **alle** Busmodule gleichzeitig bestimmt. Der Wert **ADR** enthält 0, die globale Adresse. Der Wert **LEN** bestimmt die Länge des gesamten, übertragenen Datenblocks inklusive Daten und Prüfsumme und ist abhängig von dem Kommando **CM**. Bei globalen Kommandos gibt es keine Rückmeldung – weder als Daten noch als Q2 Prüfsumme. Eine Ausnahme bildet der globale Status Request, bei dem ein Byte \$0 oder \$FF vom Slave gesendet wird.

Quersummenberechnung:

Die Quersummenberechnung startet mit dem Wert \$0 nach der Übertragung des Startzeichens **SC**. Danach wird jedes übertragene Byte B nach folgenden Algorithmus in die Quersumme einbezogen, auch das Q1 Byte.

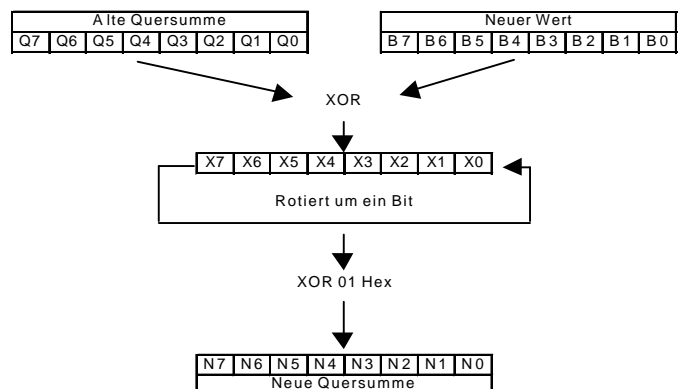


Abb. 6: Berechnung Prüfsummenbyte Q1 und Q2

Beispiel:

Byte Q XOR Byte B
 Ergebnis Wert X um ein Bit nach links rotieren, Bit 7 wird Bit 0.
 Ergebnis mit **XOR 01** Hex ergibt die Neue Quersumme.

\$00 – Get Version:

Mit diesem Befehl kann Modultype und Modulversion abgefragt werden.

Master sendet.		
SC	\$2A	Start Zeichen
LEN	\$07	Länge
ADR	MADR	Moduladresse
CM	\$00	Kommando Get Version
Q1	Q1	Prüfsumme Q1
Slave Sendet zurück.		
DS	MT	Byte Modul Type
DS	MV	Byte Daten für Modul Version
Q2	Q2	Prüfsumme Q2

Tab. 3: Datenblock für die Abfrage Modulversion und Typ.

Für die aktuellen Typ ist MT \$67 und die Version \$10. Der Wert für die Version kann sich bei neueren Modulen entsprechend ändern.

\$01 – Repeat Answer:

Mit diesem Befehl können die zuletzt gesendeten Rückgabedaten vom easyident RS-485-BUS Modul nochmals ausgegeben werden. Es können jedoch nur die ersten 8 Bytes wiederholt werden.

Master sendet.		
SC	\$2A	Startzeichen
LEN	LEN	Länge (\$5 + DS) z.B. \$0A für 5 Bytes DS
ADR	MADR	Moduladresse
CM	\$01	Kommando Repeat Answer
Q1	Q1	Prüfsumme Q1
Slave sendet zurück.		
DS	Data	1..8 Slave Daten
Q2	Q2	Prüfsumme Q2

Tab. 4: Datenblock für die Wiederholung einer Ausgabe.

\$02 – Set Status Address:

Mit diesem Befehl wird die Statusadresse SA des easyident RS-485-BUS Moduls für die Teilnahme Global Status Request GSR gesetzt. Werte ungleich 0 aktivieren die Teilnahme. Für die korrekte Funktion müssen die teilnehmenden Module **fortlaufende** Nummerierung besitzen. Der erste Teilnehmer erhält als SA den Wert \$01 Hex der zweite den Wert \$02 Hex usw.

Achtung: Doppelte Vergabe der SA werden von den Modulen **nicht erkannt** und führen zu einer Fehlfunktion des Global Status Request, da Module mit gleicher SA zur gleichen Zeit antworten werden.

Wert	Master sendet.	
SC	\$2A	Startzeichen
LEN	\$06	Länge nur mit schreiben der Status Adresse
ADR	MADR	Moduladresse
CM	\$02	Kommando Set Status Address
DM	SA	Byte Status Adresse schreiben
Q1	Q1	Prüfsumme Q1
Slave sendet zurück.		
Q2	Q2	Prüfsumme Q2

Oder mit Rückgabe Status Adresse !

Wert	Master sendet.	
SC	\$2A	Startzeichen
LEN	\$07	Länge für zusätzliche Ausgabe Status Adresse
ADR	MADR	Moduladresse
CM	\$02	Kommando Set Status Address
DM	SA	Byte Status Adresse schreiben
Q1	Q1	Prüfsumme Q1
Slave sendet zurück.		
DS	SA	Byte Status Adresse ausgelesen
Q2	Q2	Prüfsumme Q2

Tab. 5: Datenblock für die Einstellung der Status Adresse ohne und mit Antwort

\$02 – Reset all Status Address:

Mit diesem Befehl wird die Statusadresse SA aller am BUS angeschlossenen Busteilnehmer auf 0 gesetzt und damit die Teilnahme am Global Status Request gesperrt.

Master sendet.		
SC	\$2A	Startzeichen
LEN	\$04	Länge
ADR	\$0000	Global Moduladresse
CM	\$02	Kommando Reset all Status Address
Q1	Q1	Prüfsumme Q1

Tab. 6 Datenblock für Rücksetzung der Status Adresse

Bei diesem Kommando gibt es **keine** Rückgabewerte vom Slave. Eine Überprüfung der Statusadresse in jedem angeschlossenen Modul ist optional.

\$03 – Modul Reset:

Mit diesem Befehl wird das **angesprochene** easyident RS-485-BUS Modul in die Ausgangsstellung zurückgesetzt.

- ? Status Adresse SA wird auf 0 gesetzt.
- ? Modul Status ST wird zurückgesetzt.
- ? LED's und Relais sind aus.
- ? Offline Betrieb wird erlaubt.

Master sendet.		
SC	\$2A	Start Zeichen
LEN	\$05	Länge
ADR	MADR	Moduladresse
CM	\$03	Kommando Modul Reset
Q1	Q1	Prüfsumme Q1
Slave sendet zurück.		
Q2	Q2	Prüfsumme Q2

Tab. 7 Datenblock für Rücksetzung der Status Adresse

\$33 – Global Status Request:

Mit diesem Befehl kann ein schnelles Polling aller angeschlossenen Module am RS-485-BUS durchgeführt werden. Alle Module, deren Statusadresse ungleich Null ist, antworten in der Reihe entsprechend der eingestellten Status Adresse. Ist **keine** Änderung seit der letzten Abfrage – Eingangszustand oder Transponder im Lesebereich - so wird, wenn das entsprechenden Modul an der Reihe, eine \$00 senden. Bei einer Änderung seit der letzten Abfrage wird ein \$FF gesendet. Eine anschließende Statusabfrage liefert im Statusbyte ST den aktuellen Zustand.

Master sendet.		
SC	\$2A	Startzeichen
LEN	\$04+N	Länge des Datenblocks, N= Anzahl der Leser
ADR	\$0000	Globaladresse
CM	\$33	Kommando Global Status Request
Q1	Q1	Prüfsumme Q1
Slaves senden.		
S	Data	Modul antwortet mit \$00 oder \$FF

Tab. 8 Datenblock für Durchführung eines Global Status Request

Die Prüfsumme Q2 wird nicht gesendet.

\$7A – Get Modul Address:

Dieser globale Befehl funktioniert nur, wenn ein Modul am BUS hängt. Damit kann die individuelle Moduladresse ausgelesen werden. In DS ist die 2 byte Moduladresse, wobei das Highbyte zuerst kommt.

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$07	Länge
ADR	\$0000	Globaladresse
CM	\$7A	Kommando Get Modul Address
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück.
DS	MADR	Moduladresse
Q2	Q2	Prüfsumme Q2

Tab. 9 Datenblock für Auslesen der eigenen Modul Adresse

\$7B – Identifizierkommando:

Mit diesem globalen Kommando können die Moduladressen aller am Bus angeschlossenen easyident RS485-BUS Module abgefragt werden. Die Module antworten mit der eigenen Moduladresse als DS nur dann, wenn ein Transponder im Lesebereich der Antenne ist, wobei das Highbyte zuerst kommt. Ist kein Transponder lesbar, erfolgt keine Antwort. Auf diese Weise können alle Module am ? us ermittelt werden.

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$07	Länge
ADR	\$0000	Globaladresse
CM	\$7B	Kommando Identifizierungsmodus
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück
DS	Data	Moduladresse
Q1	Q1	Prüfsumme Q2

Tab. 10 Datenblock für Identifizierungskommando

Hinweis: Wird dieses Kommando mit der Aktivierung des Relais an dem entsprechenden Modul kombiniert, hat der Errichter eine Rückkopplung, dass sein Modul richtig erkannt wurde.

\$7C – Timer für offline Betrieb setzen.

Mit diesem Befehl kann das Zeitverhalten für den offline Betrieb des easyident RS-485-BUS Moduls festgelegt werden.

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$08	Länge
ADR	MADR	Moduladresse
CM	\$7C	Kommando Timer für offline Betrieb setzen
DM	TZ	Totzeit 1 Byte (Wert = 1 bis 10) *32 Sekunden
DM	PZ	Pause Zeit 1 Byte (Wert = 1 bis 50) in Sekunden
DM	RZ	Relais Zeit 1 Byte (Wert = 1 bis 250) in Sekunden
Q1	Q1	Prüfsumme Q1
		Slave sendet.
Q2	Q2	Prüfsumme Q2

Tab. 11 Datenblock für Timer für offline Betrieb setzen

Die Totzeit TZ ist die Zeit, die vergeht, nachdem das Modul in den offline Betrieb wechselt, sofern keine Buskommunikation stattfindet d.h. das Modul erkennt weder die eigene oder globale Adresse am Bus. Die Zeiteinheit ist in 32 s Schritten, Werte 1, entspricht 32s bis 10 entsprechend 320s, sind zulässig. Die Pausezeit PZ ist die Zeit zwischen dem Erkennen eines gültigen Transponders und der Relaischaltaktion. Werte von 1s bis 50s sind zulässig. Die Relaiszeit RZ bestimmt, wie lange das Relais angezogen bleibt. Werte von 1s bis 250s sind zulässig. Bei Datenwerte außerhalb des spezifizierten Wertes wird keine Prüfsumme Q2 gesendet, das

Kommando nicht ausgeführt. Nach erfolgreicher Beendigung des Kommandos wird Q2 gesendet und das easyident RS485 Bus Modul schaltet in den offline Betrieb. Nach der Auslieferung sind folgende offline Parameter gesetzt:

- TZ = 10 = 5,3 Minuten
- PZ = 30 = 30 Sekunden
- RZ = 3 = 3 Sekunden

\$80 - Get Modul Status:

Mit diesem Befehl kann der aktuelle Status des easyident RS-485-BUS Moduls abgefragt werden.

		Master sendet.
SC	\$2A	Start Zeichen
LEN	\$07	Länge
ADR	Modul	Moduladresse
CM	\$80	Kommando Get Modul Status
Q1	Q1	Quersumme Q1
		Slave sendet zurück.
DS	Data	Modulstatus ST und Status Adresse SA
Q2	Q2	Prüfsumme Q2

Tab. 12 Datenblock für Auslesen des Modulstatus

Modulstatus ST		
Bit 7	0 = Kein Flankenwechsel (Sabotagekontakt)	1 = Flankenwechsel (Sabotagekontakt)
Bit 6	0 = Sabotagekontakt zu	1 = Sabotagekontakt offen
Bit 5	0 = Türkontakt zu	1 = Türkontakt offen
Bit 4	0 = kein Transponder im Lesebereich	1 = Transponder im Lesebereich
Bit 3	0 = offline Betrieb gesperrt	1 = offline Betrieb erlaubt
Bit 2	0 = Relais aus	1 = Relais ein
Bit 1	0 = LED grün aus	1 = LED grün ein
Bit 0	0 = LED rot aus	1 = LED rot ein

Tab. 13 Ansicht des internen Statusbytes mit Einzelbitbeschreibung

Die entsprechenden Eingänge werden zyklisch alle 100us abgefragt. Änderungen zwischen 2 Abfragen werden gespeichert. Mit dem Get Modul Status kann der aktuelle Zustand abgefragt werden. Mit der Abfrage wird der Zustand des Bit 7 zurückgesetzt. Die Zustände der beiden TTL Inputs 1 und 2 werden zum Zeitpunkt der Abfrage ermittelt und über das Statusbyte über Bit 5 und 6 mitgeteilt. Für den Global Status Request sind Bit 4..7 maßgebend, danach richtet sich die Antwort des Moduls.

\$81 – Set Relais and LED

Mit diesem Befehl kann das Relais und die beiden LED's angesteuert werden. Der Befehl ist so aufgebaut, dass Relais und LED's unabhängig ein bzw. ausgeschaltet werden können. Der momentane Zustand ist über GET MODUL STATUS abrufbar.

		Master sendet.
SC	\$2A	Start Zeichen
LEN	\$06	Länge
ADR	MADR	Moduladresse
CM	\$81	Kommando Set Relais ans LED
DM	SB	Statusbyte
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück.
Q2	Q2	Prüfsumme Q2

Tab. 14 Datenblock für Auslesen des Status Byte

Folgende Abbildung zeigt die Zuordnung der Datenbits für das Statusbyte SB.

Bit 7	Keine Funktion	Keine Funktion
Bit 6	Keine Funktion	Keine Funktion
Bit 5	Keine Funktion	Keine Funktion
Bit 4	Keine Funktion	Keine Funktion
Bit 3	0 = offline Betrieb gesperrt	1 = offline Betrieb erlaubt
Bit 2	0 = Relais aus	1 = Relais ein
Bit 1	0 = LED grün aus	1 = LED grün ein
Bit 0	0 = LED rot aus	1 = LED rot ein

Tab. 15 Ansicht des internen Statusbytes mit Einzelbitbeschreibung

erhalten **alle angeschlossenen Module die gleiche Adresse**, mit dieser Einstellung wird das Bussystem **nicht** funktionieren.

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$09	Länge
ADR	\$0000	Global Moduladresse
CM	\$A8	Kommando Programm Modul Address
DM	4 Byte	Neue Moduladresse und komplement
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück.
Q2	Q2	Prüfsumme Q2

Tab. 18 Ansicht Protokoll für Program Modul Adress

Beispiel: Adresse **1234** Hex programmieren. Die entsprechende Bytefolge ist: \$2A \$09 \$00 \$00 \$A8 \$12 \$34 \$EE \$CC \$41, wobei sich das Komplement der neuen Moduladresse aus Highbyte : \$100 – \$12 = \$EE und Lowbyte \$100 – \$34 = \$CC berechnet.

\$88 - Read Card Data:

Mit diesem Befehl werden die Daten des Transponders gelesen. Die Übertragung erfolgt in einem Block mit fester Länge von insgesamt 12 Bytes. Damit können die 40 Bit Daten und 14 Prüfbits aus dem Transponder übertragen werden.

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$0C	Länge
ADR	MADR	Moduladresse
CM	\$88	Kommando Read Card Data
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück.
DS	TrData	7 Byte Transponder Datenblock
Q2	Q2	Prüfsumme Q2

Tab. 16 Ansicht Read Card Data

\$AE - Write EEPROM Data:

Mit diesem Befehl können Transponderdaten für den *offline* Betrieb ins EEPROM geladen werden. Bis zu 1142 Einträge sind möglich. **SPNR** ist die entsprechende Speichernummer, wobei das Highbyte zuerst gesendet wird. Die 7 Bytes für die Transponderdaten entsprechen dem Format von Tab.21. Nach korrekter Datenübertragung erfolgt die Programmierung, nach 8 bis 16 ms wird die Prüfsumme Q2 gesendet, das Kommando ist abgeschlossen.

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$0E	Länge
ADR	MADR	Moduladresse
CM	\$AE	Kommando Download EEPROM Data
DM	SPNR	Speichernummer
DM	TrData	7 Byte Transponder Datenblock
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück.
Q2	Q2	Prüfsumme Q2

Tab. 19 Ansicht Download EEPROM Data

Die 40 Bit Transponderdaten, entsprechend einer 10 Stelligen ASCII HEX Nummer werden im easyident RS485 BUS Modul zu einem 7 Byte langen Datenblock gepackt. Neben der eigentlichen Transpondernummer sind alle Paritybits in dem Datenblock. Mit diesem Datenformat werden auch die Vergleichsoperationen in *offline* Betrieb durchgeführt. Die eingelesenen Daten können unverändert – bis auf die Schalfunktion – unverändert in das EEPROM für den *offline* Betrieb gespeichert werden. Für genauere Informationen siehe Write EEPROM Data und Transponder Daten Format.

\$A5 Main Reset:

Mit diesem globalen Befehl werden **alle** easyident RS-485-BUS Module in die Ausgangsstellung gebracht.
 ? Status Adresse **SA** wird auf 0 gesetzt.
 ? Modul Status **ST** wird zurückgesetzt.
 ? LED's und Relais werden ausgeschaltet.
 ? *Offline* Betrieb wird erlaubt

		Master sendet.
SC	\$2A	Start Zeichen
LEN	\$04	Länge
ADR	0000	Global adresse
CM	\$A5	Kommando Main Reset
Q1	Q1	Prüfsumme Q1

Tab. 17 Datenblock für Main Reset

Bei diesem Kommando gibt es **keine** Rückgabewerte vom Slave.

\$A8 - Program Modul Address:

Dieses Kommando ist für Service und Prüffeld gedacht.
 Mit diesem globalen Befehl kann die Moduladresse des easyident RS-485-BUS Moduls geändert werden. Die Adresse sollte identisch mit der außen sichtbaren, individuellen 4 stelligen Nummer sein. Für die Programmierung muß die neue Moduladresse und das Komplement derselben gesendet werden, wobei das jeweilige Highbyte zuerst kommt. Nach der Programmierung im EEPROM wird Q2 gesendet.

Achtung: Dieser Befehl darf nur ausgeführt werden, wenn nur ein Modul am Bus hängt. Bei gleichzeitig mehreren Modulen am Bus

Das Datenformat für TrData wird wie folgt zusammen gestellt. Aus den 10 ASCII Hex Rohdaten (Byte 1 bis Byte 10 wird die Zeilenprüfsumme Q (Z1 bis Z10) und die Spalten Prüfsumme (S1 bis S4) berechnet. Die Prüfsummen werden immer auf gerade Daten ermittelt. Nun müssen Sie diese Daten in eine 7 Byte Information umwandeln. Folgendes Bild zeigt dabei die Zuordnung der einzelnen Bits zwischen den Rohdaten links und dem gepackten Format rechts.

Nr.	HEX Data				Q
Byte 1	A	A	A	A	Z1
Byte 2	B	B	B	B	Z2
Byte 3	C	C	C	C	Z3
Byte 4	D	D	D	D	Z4
Byte 5	E	E	E	E	Z5
Byte 6	F	F	F	F	Z6
Byte 7	G	G	G	G	Z7
Byte 8	H	H	H	H	Z8
Byte 9	I	I	I	I	Z9
Byte 10	J	J	J	J	Z10
Summe	S1	S2	S3	S4	

Nr. / Bit	7	6	5	4	3	2	1	0
Byte 1	A	A	A	A	Z1	B	B	B
Byte 2	B	Z2	C	C	C	C	Z3	D
Byte 3	D	D	D	Z4	E	E	E	E
Byte 4	Z5	F	F	F	F	Z6	G	G
Byte 5	G	G	Z7	H	H	H	H	Z8
Byte 6	I	I	I	I	Z9	J	J	J
Byte 7	J	Z10	S1	S2	S3	S4	X	Y

gültiger Transponder erkannt wird.. Die Zuordnung ist wie folgt.

Byte 7 Bit 0 (Y) 1 = Relais Anziehen 0 = Relais nicht anziehen
 Byte 7 Bit 1 (X) 1 = LED 2 (rot) EIN 0 = LED 2 (rot) nicht EIN

Über den LED 2 (rot) Ausgang kann ein 2. Relais angesteuert werden.

Achtung:

Alle nicht benutzten Speicher müssen mit FF FF FF FF FF FF FF FF belegt werden.

Die Einträge müssen mit der Nummer NR=0 Starten und fortlaufend programmiert werden. Für nicht belegte Speicherplätze kann die Transpondernummer 0000000000 belegt werden. Sobald Sie alle 7 Byte Daten von TrData = „FF“ setzen werden alle Speicherplätze die nach diesem Eintrag folgen nicht mehr für die Freischaltung berücksichtigt. Sie können daher alle gespeicherte Nottransponder deaktivieren in dem Sie in den Speicherplatz 0 alle 7 Bytes auf „FF“ Hex setzen.. Wenn Sie anschließend im Speicherplatz 0 eine neue Transpondernummer oder die Transponder Nummer 0000000000 schreiben sind alle folge Speicher wieder aktiviert.

Sollen einzelne Nottransponder aus dem easyident gelöscht werden, so schreiben Sie einfach in diesen Speicherplatz die Transpondernummer 0000000000.

Beispiel:

Aus einem Transponder mit der Nummer 010055EEAD ergibt sich ein Eintrag (7 Bytes) 00 C0 05 2B BD A6 DB Hex

Aus den beiden Abbildungen ist der Zusammenhang zwischen dem Nummernformat, rechte Abbildung und dem gepackten Format, linke Abbildung ersichtlich. Die 10 stellige Transpondernummer $N_{0..9}$, entsprechend 40 Bit, steht in den Bits 0..3 mit Index 0..9. $P_{0..9}$ sind die Reihen parity bits, Bit 0..3 mit Index P ist das Spalten parity halb byte, es besitzt kein Reihen parity bit. Es wird gerade Parity verwendet, sowohl bei den Reihen, wie auch bei der Spalten.

Das Bits X_R ist für die Schaltfunktion des Relais beim *offline* Modus entscheidend, das Bit X_L für den Ausgang LED rot. Eine 0 verhindert das Schalten des entsprechenden Ausgangs, eine 1 läßt einen Schaltvorgang zu, vorausgesetzt, der Transponder war gültig und der *offline* Betrieb aktiv.

Beispiel: Aus einem Transponder mit der Nummer \$010055EEAD ergeben sich 7 Bytes im gepackten Format \$00 \$C0 \$05 \$2B \$BD \$A6 \$DB. Die Bits X_R und X_L sind in diesem Beispiel 1, der Transponder mit der Nummer \$010055EEAD wird sowohl Relais, als auch den Ausgang LED rot schalten.

Offline Betrieb :

Der *offline* Betrieb ist nach dem bestromen des Moduls aktiv und schaltet sich beim ersten Erkennen der eigenen oder der globale Moduladresse ab, die Wartezeit für die Umschaltung in den *offline* Betrieb wird auf den eingestellten Wert für **TZ** zurückgesetzt. Zur Kennzeichnung des *offline* Betriebs leuchtet die grüne LED. Das Zeitverhalten im *offline* Betrieb kann mit dem Kommando - Timer für *offline* Betrieb setzen – eingestellt werden. Ob das easyident RS485 BUS Modul in den *offline* Modus schalten darf oder nicht, wird über die Schnittstelle mit dem Kommando Set Relais and LED -Tab. 14 eingestellt.

Hinweis: Für die ordnungsgemäße Funktion im *offline* Betrieb **müssen** Transponder im Modul gespeichert sein. Dies geschieht mit dem Kommando - Write EEPROM Data. Mit Kommando - SET RELAIS AND LED muss der *offline* Betrieb erlaubt werden

Bei aktivem *offline* Betrieb, die grüne LED leuchtet, wird jeder Transponder im Lesebereich kontinuierlich gelesen. Bei fehlerfreiem Lesen – Paritybits richtig – wird das interne EEPROM nach der gelesenen Transpondernummer durchsucht. Eine mit \$FF beginnende Nummer im EEPROM bricht den Suchvorgang ab, der Lesevorgang beginnt erneut. Wird die Nummer gefunden, so blinkt die grüne LED entsprechend der eingestellten **PZ**. Nach Ablauf der **PZ** wird das entsprechende Schaltelement (Relais oder LED rot) entsprechend den Bits XL und XR für die **RZ** aktiviert. Ist die Option bistabil eingestellt, schaltet der entsprechende Schaltausgang sofort um, unabhängig von der Einstellung der **PZ** und **RZ**. Dieser Schaltzustand bleibt bestehen, bis entweder ein erneuter Schaltvorgang durch eine Transponder erfolgt oder der *offline* Betrieb durch entsprechende Buskommunikation abgeschaltet wird. Für eine neue Schaltfunktion **muss** der Transponder aus dem Lesebereich entfernt werden, beim Verlassen des offline Betriebs werden Relais und LED Ausgang rot abgeschaltet.

\$AF – Read EEPROM Daten:

Mit diesem Befehl können die gespeicherten Transpondernummern und Logbuchinformationen für den *offline* Betrieb aus dem easyident RS-485-BUS Modul ausgelesen werden. Die Auswahl des entsprechenden Transponders erfolgt über die Speichernummer **SPNR**, wobei das Highbyte zuerst gesendet wird. Die 7 Datenbytes werden umgehend gesendet, gefolgt von Q2. Maximal sind 1168 Speichernummern möglich, wobei **SPNR #0..#1141** für Transponder im *offline* Betrieb vorgesehen sind. **SPNR #1142 ..#1167** ist der Logbuchbereich und **SPNR #1168** enthält den Zeiger zu den Logbuchdaten. Eine Überprüfung der Transponderdaten ist optional

		Master sendet.
SC	\$2A	Startzeichen
LEN	\$0E	Länge immer
ADR	MADR	Moduladresse
CM	\$AF	Kommando Read EEPROM Daten
DM	SPNR	Speicher Nummer
Q1	Q1	Prüfsumme Q1
		Slave sendet zurück.
DS	TrData	7 Byte Transponder Datenblock
Q2	Q2	Prüfsumme Q2

Tab. 20 Ansicht Read EEPROM Data

Transponder Daten Format:

Die 54 Bit Transponderdaten – bestehend aus 40 Bit Nutzdaten, 10 Reihen parity bits und 4 Spalten parity bits - werden in 7 Byte gepackt. In diesem Format werden Transponderdaten im easyident RS-485-BUS Modul intern verarbeitet. Zwischen der 10 stelligen Transpondernummer und dem gepackten Format besteht folgender Zusammenhang:

0 ₀	1 ₀	2 ₀	3 ₀	P ₀	0 ₁	1 ₁	2 ₁	TRData 0	0 ₀	1 ₀	2 ₀	3 ₀	P ₀	N ₀
3 ₁	P ₁	0 ₂	1 ₂	2 ₂	3 ₂	P ₂	0 ₃	TRData 1	0 ₁	1 ₁	2 ₁	3 ₁	P ₁	N ₁
1 ₃	2 ₃	3 ₃	P ₃	0 ₄	1 ₄	2 ₄	3 ₄	TRData 2	0 ₂	1 ₂	2 ₂	3 ₂	P ₂	N ₂
P ₄	0 ₅	1 ₅	2 ₅	3 ₅	P ₅	0 ₆	1 ₆	TRData 3	0 ₃	1 ₃	2 ₃	3 ₃	P ₃	N ₃
2 ₆	3 ₆	P ₆	0 ₇	1 ₇	2 ₇	3 ₇	P ₇	TRData 4	0 ₄	1 ₄	2 ₄	3 ₄	P ₄	N ₄
0 ₈	1 ₈	2 ₈	3 ₈	P ₈	0 ₉	1 ₉	2 ₉	TRData 5	0 ₅	1 ₅	2 ₅	3 ₅	P ₅	N ₅
3 ₉	P ₉	0 _P	1 _P	2 _P	3 _P	X _L	X _R	TRData 6	0 ₆	1 ₆	2 ₆	3 ₆	P ₆	N ₆
									0 ₇	1 ₇	2 ₇	3 ₇	P ₇	N ₇
									0 ₈	1 ₈	2 ₈	3 ₈	P ₈	N ₈
									0 ₉	1 ₉	2 ₉	3 ₉	P ₉	N ₉
									0 _P	1 _P	2 _P	3 _P	0	

Tab. 21 Transponderdaten im gepackten TrData und Nummernformat

Transponderspeicher und Logbuchdaten

In dieser Modulversion stehen insgesamt 1169 Speicherplätze für den *offline* Betrieb zur Verfügung. Der SPNR #1168 erlaubt den Zugriff auf den internen Konfigurationsspeicher. Darin sind die Parameter für diese Modulversion gespeichert.

SPNR	Bezeichnung
0	Offline Transponder #0
1	Offline Transponder #1
...	
1140	Offline Transponder #1141
1141	Offline Transponder #1142
1142	Logbuch Eintrag #25
1143	Logbuch Eintrag #24
...	
1166	Logbuch Eintrag #1
1167	Logbuch Eintrag #0
1168	Konfigurationsblock
1169	Reserviert

Tab. 22 Ansicht Aufteilung EEPROM SPNR's

Der Transponderbereich für den *offline* Betrieb erstreckt sich von **SPNR** 0..#1141 und kann **nicht** verändert werden. Der Bereich für die Logbuchdaten ist 26 Einträge lang, kann **nicht** verändert werden und beginnt bei **SPNR** #1167 – entsprechend dem Wert 0 des Logbuchzeigers und **SPNR** #1142 entsprechend dem Wert 25 des Logbuchzeigers. Der Wert des Logbuchzeigers steht unter **SPNR** #1168, Byte 4 enthält das Highbyte und Byte 5 das Lowbyte, zeigt auf die Stelle, an die der nächste Eintrag bei einem Ereignis - gültiger Transponder im *offline* Betrieb - geschrieben wird, das entspricht dem ältesten Eintrag im Logbuch. Der Wert des Logbuchzeigers wird anschließend um 1 erhöht. Erreicht der Wert des Zeigers 26, wird er zu 0 zurückgesetzt – Das Logbuch ist als Ringspeicher angelegt.

Master sendet.		
SC	\$2A	Startzeichen
LEN	\$0E	Länge
ADR	MADR	Moduladresse
CM	\$AF	Kommando Read EEPROM Daten
DM	SPNR	#1168
Q1	Q1	Prüfsumme Q1
Slave sendet zurück.		
DS	TrData	Konfigurationsblock
Q2	Q2	Prüfsumme Q2

Tab. 23 Ansicht Read EEPROM Data, 7 Byte Konfigurationsblock

Byte	7 Byte Konfigurationsblock
0	Letzte Speichernummer SPNR für die <i>offline</i> Transponder.
1	Reserviert
2	Logbuchlänge Highbyte, Lowbyte
3	Reserviert
4	Wert des Zeigers zum nächsten Logbucheintrag
5	Highbyte, Lowbyte
6	0: monostabile 3: bistabile Schaltfunktion im <i>offline</i> Mode

Tab. 24 Ansicht Daten im Konfigurationsblock mit SPNR #1168

Hinweis: Die Werte für Byte 0..3 können mit der Funktion Download EEPROM Data beliebig verändert werden, haben aber keine Einfluß auf die Funktion des easyident RS485 BUS Modul. Reserviert für spätere Modulvarianten. Mit Byte 4,5 kann der Wert des Logbuchzeigers geändert werden. Byte 6 bestimmt die Schaltfunktion im *offline* Betrieb.

Herstellerangaben.

FS Fertigungsservice

Waldweg 11 85777 Fahrenzhausen
Tel. 08133-9158 Fax. 08133-9159

eMail: Info@FSJR.de
Internet: www.easyident.de