

**Schnittstellenbeschreibung
für
SRC RS485 MODBUS**

Version 1.4, 03.01.2006

Änderungsindex

Version	Datum	Beschreibung
1.0,1.1	13.08.04	Erste Versionen
1.2	11.11.04	Update auf neue Firmware V1.2 <ul style="list-style-type: none">• Neue Steuerbefehle 2,4,5,6• Invertierung von Datenbyte 1 ist möglich mit den Konfigurationsbits 32 - 63
1.2a	15.11.04	4.1 Einlernen über...., Fehler in Beispiel: Sensor 1 in Sensor 2 geändert
1.3	04.11.05	Update auf neue Firmware V1.3 <ul style="list-style-type: none">• Selbsthalte-Funktion der Präsenstaste: Änderung der Präsenztaste wird gesendet• 2.5 Sensordaten Beschreibung erweitert
1.4	03.01.06	Update auf neue Firmware V1.4 <ul style="list-style-type: none">• Auswertung von Tastern möglich• Selbsthaltefunktion der Tasten• 2.5 Sensordaten Beschreibung erweitert

Änderungsindex.....	1
1 Einführung	3
2 Gerätebeschreibung	3
2.1 RS485 Transceiver	3
2.2 Protokoll	3
2.3 Konfigurationsmöglichkeiten	3
2.4 Unterstützte Steuerbefehle	4
2.5 Datenverwaltung	4
2.5.1 Registerzuordnung der Sensordaten	4
2.5.2 Registerzuordnung Modbus-Konfiguration	7
2.5.3 Bitzuordnung für Sensor-Lernmodus.....	7
2.5.4 Bitzuordnung zur Konfiguration „invertieren von Datenbyte 1“	8
3 Datenübertragung	9
3.1 Master/Slave Protokoll	9
3.2 Datenrahmen	9
3.3 Übertragungsmodus RTU	9
3.3.1 Telegrammaufbau.....	9
3.3.2 Berechnung der CRC-Prüfsumme.....	10
3.4 Übertragungsmodus ASCII	11
3.4.1 Telegrammaufbau.....	11
3.4.2 Berechnung der LRC-Prüfsumme	11
4 Einlernen von Sensoren.....	12
4.1 Einlernen über MODBUS - Schreibbefehl	12
4.2 Einlernen über Lerntaste des Funksensors.....	13
5 Daten auslesen	14
5.1 Register Auslesen.....	14
5.2 Bits Auslesen.....	15

1 Einführung

Das vorliegende Dokument beschreibt die serielle Schnittstelle des Funkempfängers SRC-RS485-MODBUS. Das von der Fa. Modicon entwickelte MODBUS-Protokoll ist ein offengelegtes Protokoll zur Kommunikation mehrerer intelligenter Geräte auf Master-Slave-Basis.

Weiterführende Informationen und Definitionen zum Thema MODBUS sind unter www.modbus.org erhältlich.

Ab der Firmware Version 1.4 können Taster ausgewertet werden.

2 Gerätebeschreibung

2.1 RS485 Transceiver

Die max. Anzahl der Busteilnehmer ohne Verwendung eines Repeaters wird durch den RS485-Transceiver vorgegeben. Der hier verwendete Transceiver gestattet max. 32 Geräte pro Bussegment.

2.2 Protokoll

Das Empfangsmodul SRC-RS485-MODBUS ist ein Slave-Busteilnehmer, der nur auf Anforderung des Masters auf den Bus senden darf. Das Protokoll entspricht den Vorgaben aus:

- MODBUS Application Protocol Specification V1.1
- MODBUS over Serial Line Specification & Implementation guide V1.0

2.3 Konfigurationsmöglichkeiten

Mittels Steckbrücken und 8pol. Dipschalter kann das Gerät an die jeweilige Bustopologie angepasst werden. Einstellbar sind:

- die Busadresse des Gerätes (1 - 247) über den 8po. Dipschalter
- Busabschlusswiderstand 120 Ohm
- Übertragungsmodus RTU oder ASCII
- Baudrate 9600 oder 19200
- Parität gerade, Parität ungerade oder kein Parität

Da das Geräte-Datenblatt eine detaillierte Beschreibung zu Position und Bedeutung der Steckbrücken enthält wird an dieser Stelle auf die Datei „Produktblatt_src_rs485.pdf“ verwiesen.

Wichtige Hinweise für den Betrieb im Master/Slave-System:

- !! Die Busadresse muss für jedes Gerät unterschiedlich eingestellt werden**
- !! Übertragungsmodus, Baudrate und Parität müssen gleich sein**
- !! Änderungen der Einstellungen werden erst nach Spannungsreset übernommen**

2.4 Unterstützte Steuerbefehle

Folgende MODBUS - Steuerbefehle werden unterstützt:

Beschreibung	Funktionscode	
Bitstelle(n) lesen	01 (hex)	1 (dez)
	02 (hex)	2 (dez)
Register lesen	03 (hex)	3 (dez)
	04 (hex)	4 (dez)
einzelnes Bit schreiben	05 (hex)	5 (dez)
einzelnes Register schreiben	06 (hex)	6 (dez)
mehrere Bits schreiben	0F (hex)	15 (dez)
mehrere Register schreiben	10 (hex)	16 (dez)

Tabelle 1

2.5 Datenverwaltung

Allen Daten in einem MODBUS-Slave sind Adressen zugeordnet. Der Zugriff auf die Daten (lesen oder schreiben) erfolgt durch den Entsprechenden Steuerbefehl und die Angabe der entsprechenden Datenadresse.

2.5.1 Registerzuordnung der Sensordaten

Ein Register besteht per Definition in MODBUS-Geräten aus 16 Bit. In den Registern 0-319 liegen hier die Daten zur Verwaltung von bis zu 32 Thermokon EasySens Sensoren, wobei jedem Sensor 10 Register zugeordnet sind (siehe Tabelle 2):

Sensor 1	Register 0 - 9 _{dez}
Sensor 2	Register 10 - 19 _{dez}
:	
Sensor 32	Register 310 - 319 _{dez}

Register	MSB								LSB								
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00	
0 R/W	not used								ORG								Daten Sensor 1
1 R/W	ID-Byte-3								ID-Byte-2								
2 R/W	ID-Byte-1								ID-Byte-0								
3 R	not used								Data-Byte-3								
4 R	not used								Data-Byte-2								
5 R	not used								Data-Byte-1								
6 R	not used								Data-Byte-0								
7 R	Receive-Time-Byte-1								Receive-Time-Byte-0								
8 R	not used								not used								
9 R	not used								not used								
:																	
:																	
310R/W	not used								ORG								Daten Sensor 32
311 R/W	ID-Byte-3								ID-Byte-2								
312 R/W	ID-Byte-1								ID-Byte-0								
313 R	not used								Data-Byte-3								
314 R	not used								Data-Byte-2								
315 R	not used								Data-Byte-1								
316 R	not used								Data-Byte-0								
317 R	Receive-Time-Byte-1								Receive-Time-Byte-0								
318 R	not used								not used								
319 R	not used								not used								

Tabelle 2: Registerzuordnung der Sensordaten

2.5.2 Identifikationscode

Die jeweils ersten 3 Register enthalten den Identifikationscode eines Sensors, der jeden Sensor eindeutig identifiziert. Er bestehend aus ORG-Byte (Gerätekennung Taster / 1Byte / 4Byte Sensor), und den ID-Bytes 0 bis 3.

Diese Register sind mit „R/W“ gekennzeichnet und haben sowohl Lese- als auch Schreibzugriff. Diese Daten werden im EEPROM abgespeichert und bleiben damit auch nach Spannungsreset erhalten.

2.5.3 Data-Bytes Sensoren

Die nachfolgenden vier Register enthalten die Sensordaten Data-Bytes 0 - 3. Die Bedeutung der Daten und wie diese weiterverarbeitet werden können ist abhängig vom Sensortyp. Bitte beachten sie hierzu die entsprechend Produktdatenblätter. Diese Register sind mit „R“ gekennzeichnet und können über den Modbus nur ausgelesen werden.

Data-Byte 0

- Für digitale Werte, z.B. SR04 xx mit Präsenztaste, Fensterkontakt SRW01
- Mit Selbsthalte-Funktion, Statusänderung der Präsenztaste wird im Gerät bis zur nächsten Modbus-Anfrage gespeichert und dann gesendet

Data-Byte 1

- Temperatur
- Auflösung 0 – 255 Bit, den Messbereich entnehmen Sie dem Datenblatt des Sensors
- Temperatur kann invertiert werden (siehe 2.5.8)

Schnittstellenbeschreibung SRC-RS485-Modbus

Data-Byte 2

- Sollwertsteller bei SR04 xx
- Feuchtwert bei SR04 rH

Data-Byte 3

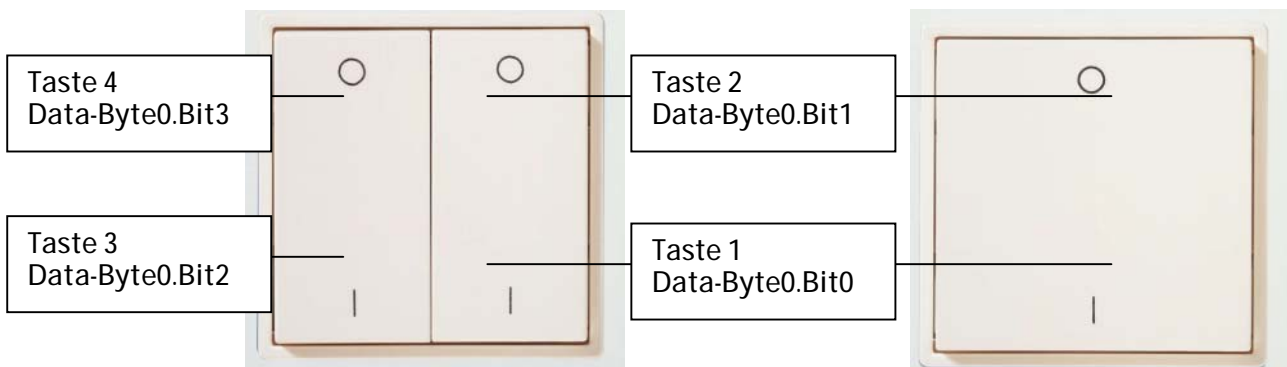
- Lüfterstufe bei SR04 xx
- Sollwertsteller bei SR04 rH

2.5.4 Data-Bytes Taster

Die nachfolgenden vier Register enthalten die Tasterdaten Data-Bytes 0 - 3.

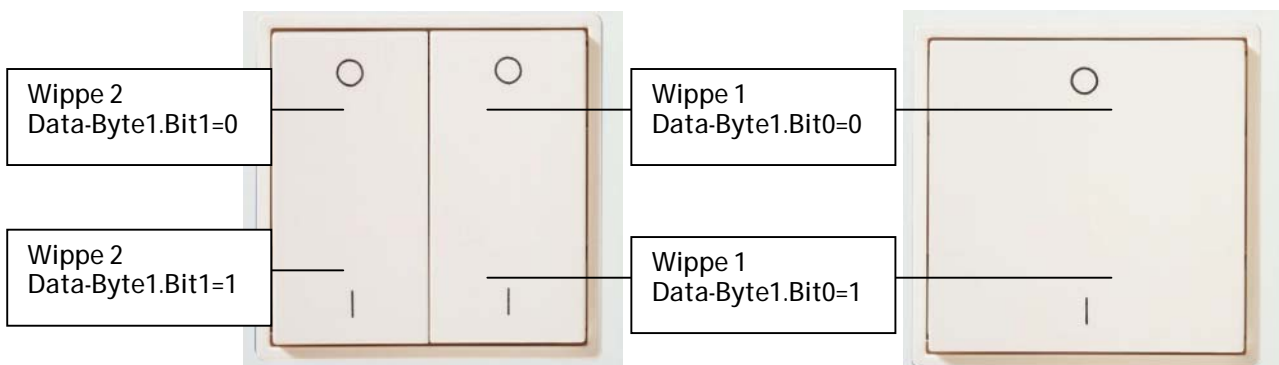
Data-Byte 0

- Aktueller Zustand der Tasten
- Taster-Funktion
- Alle Statusänderung des Tasters wird im Gerät bis zur nächsten Modbus-Anfrage gespeichert und dann gesendet
- Nach einer Abfrage des Registers wird das Data-Byte0 zurückgesetzt, außer eine Taste ist noch gedrückt
- bit = 1 ==> Taste gedrückt, bit = 0 ==> Taste nicht gedrückt



Data-Byte 1

- Aktueller Zustand der Wippe
- Schalten-Funktion
- Taste I: Bit0/Bit1 = 1
- Taste O: Bit0/Bit1 = 0



Data-Byte 2

- Aktueller Zustand des Tasters
- Taster-Funktion Statusänderung des Tasters wird im Gerät bis zur nächsten Modbus-Anfrage gespeichert und dann gesendet
- Gespeichert wird die zuletzt gedrückt Taste als Raw-Wert

Data-Byte 3

- Aktueller Zustand des Tasters

Da das Master-Slave-System beim Modbus zu langsam ist, kann es zu Verzögerungen bei Tasterbetätigungen kommen.

2.5.5 Sensor-Überwachungszeit

Das jeweils achte Register Receive-Time zeigt an, wie viel Zeit vergangen ist, seitdem das letzte Funktelegramm des Sensors empfangen wurde.

Daten die mit „not used“ gekennzeichnet sind, werden bei Datenausgabe immer mit dem Wert „0“ ausgegeben.

2.5.6 Registerzuordnung Modbus-Konfiguration

Register	MSB								LSB								
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 09	Bit 08	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00	
320R/W	Min-Response-Time-Byte-1								Min-Response-Time-Byte-0								min. Antwortzeit

Tabelle 3: Registerzuordnung für minimale Antwortzeit

Das Register 320 hat Lese- und Schreibzugriff und definiert die minimale Zeit (ms) die vergehen muss, bevor ein Slave auf eine Master-Anfrage antworten darf. Diese Daten werden im EEPROM abgespeichert und bleiben damit auch nach Spannungsreset erhalten. Voreingestellter Wert: 10 ms, kleinster erlaubter Wert 5 ms.

2.5.7 Bitzuordnung für Sensor-Lernmodus

Die in Tabelle 4 aufgelisteten Bitwerte sind mit „R/W“ gekennzeichnet und haben sowohl Lese- als auch Schreibzugriff. Wird z.B. Bit0 mit dem Wert „1“ beschrieben, dann ist der Lernmodus für Sensor1 aktiviert.

Im Lernmodus wartet der Empfänger auf ein Einlern- Funktelegramm eines Sensors, welches durch drücken der Lern Taste am Sensor erzeugt wird. Bei erfolgreicher Übertragung des Funktelegramms schreibt der Empfänger den Identifikationscode des Sensors in die entsprechenden Register (siehe Tabelle 2 und Kapitel 4 Einlernen von Sensoren“).

Bit	Wert = 1 ==> Lernmodus aktiv
0 R/W	Lernmodus Sensor 1
1 R/W	Lernmodus Sensor 2
	:
31 R/W	Lernmodus Sensor 32

Tabelle 4

2.5.8 Bitzuordnung zur Konfiguration „invertieren von Datenbyte 1“

Datenbyte 1 wird bei Thermokon - Temperatursensoren und Raumbediengeräten zur Übertragung des Temperaturwertes verwendet. Die Fühlertypen SR04 (ohne rel. Luftfeuchte) und SR65 senden den Temperaturwert invertiert, d.h. der minimale Temperaturwert entspricht in Datenbyte-1 dem Wert 255 und der maximale Temperaturwert entspricht in Datenbyte-1 dem Wert 0 (siehe hierzu die entsprechenden Produktdatenblätter).

Die Konfigurationsbits 32 bis 63 bieten für jeden Sensor einzeln die Möglichkeit den Temperaturwert zu invertieren, so dass die Temperatur proportional mit den Werten 0 bis 255 ausgegeben wird.

Diese Daten werden im EEPROM abgespeichert und bleiben damit auch nach Spannungsreset erhalten.

Bit	Speicherbereich	Wert = 1 ==> Datenbyte 1 invertieren
32 R/W	EEPROM	Sensor 1
33 R/W	EEPROM	Sensor 2
:		
63 R/W	EEPROM	Sensor 32

Tabelle 5

3 Datenübertragung

3.1 Master/Slave Protokoll

Ein Master und ein oder mehrere Slaves werden an den seriellen Bus angeschlossen. Die Kommunikation zwischen Master und Slave wird ausschließlich durch den Master geregelt. Die Slaves dürfen nur dann senden, wenn sie vorher vom Master angesprochen wurden. Slaves senden nur zurück zum Master, niemals an einen anderen Slave.

3.2 Datenrahmen

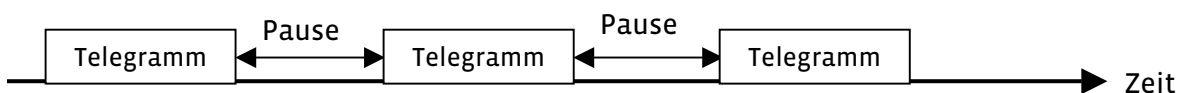
Die Daten werden nach streng definierten Vorgaben auf den Bus gesendet:

Adresse	Steuerbefehl	Daten	Prüfsumme
---------	--------------	-------	-----------

Allgemein startet ein MODBUS-Telegramm mit der Adresse des Slaves, gefolgt von einem Steuerbefehl (z.B. Register auslesen) und den Daten. Mit Hilfe der Prüfsumme am Telegrammende können die Busteilnehmer Übertragungsfehler erkennen.

3.3 Übertragungsmodus RTU

Im Übertragungsmodus RTU werden Telegramme durch Übertragungspausen voneinander getrennt:



Die Dauer der Übertragungspausen zur Trennung von Telegrammen ist abhängig von der eingestellten Baudrate und beträgt $3,5 \cdot \text{Wort-Übertragungszeit (11 Bit)}$. Bei 9600 Baud müssen damit mindestens 4 ms und bei 19200 mindestens 2 ms. zwischen zwei Telegrammen vergehen.

3.3.1 Telegrammaufbau

Adresse 1 Byte	Steuerbefehl 1 Byte	Daten 0 - 100 byte	Prüfsumme	
			CRC Low	CRC High

3.3.2 Berechnung der CRC-Prüfsumme

Die CRC - Prüfsumme (Cyclic Redundancy Check) wird vom Sender aus allen übertragenen Bytes berechnet und der Botschaft angehängt.

Der Empfänger berechnet dann die CRC-Prüfsumme erneut und vergleicht sie mit der empfangenen Prüfsumme. Stimmen die Werte nicht überein, dann ist von einem Übertragungsfehler auszugehen und die empfangenen Daten werden verworfen.

Das niederwertige Byte der 16 Bit großen Prüfsumme wird im Telegramm an vorletzter und das höherwertige Byte an letzter Stelle gesendet.

Berechnung der Prüfsumme (Programmbeispiel in C):

```

crc = 0xFFFF;           // CRC-Check, Initialisierung
for(i = 0; i < Telegrammlänge-2; i++)
    crc = crc_calc(crc, Telegrammdaten[i]);

crc_low = crc & 0x00FF; // Low-Byte
crc_high = (crc & 0xFF00) >> 8; // High-Byte

// Funktionsdefinition CRC Berechnen
unsigned int  crc_calc(unsigned int  crc_temp, unsigned int data)
{
    unsigned int  Index_CC=0;           // Schleifenzähler
    unsigned int  LSB=0;                // Hilfsvariable

    // Exclusive-Oder des 8Bit-Char mit den unteren 8Bit von CRC
    crc_temp = (( crc_temp ^ data) | 0xFF00) & (crc_temp | 0x00FF);

    for(Index_CC = 0; Index_CC<8; Index_CC++)
    {
        LSB = (crc_temp & 0x0001);
        crc_temp >>= 1;
        if(LSB)
            crc_temp = crc_temp ^ 0xA001; // calculation polynomial für CRC16
    }

    return(crc_temp);
}

```

3.4 Übertragungsmodus ASCII

Der ASCII-Übertragungsmodus stellt nicht so hohe Anforderungen an die Rechnergeschwindigkeit der Busteilnehmer. Die Telegramme werden hier nicht durch Pause-Zeiten voneinander getrennt, sondern durch ASCII-Steuerzeichen.

3.4.1 Telegrammaufbau

Das ASCII-Steuerzeichen „:“ bezeichnet immer den Anfang eines Telegramms und die ASCII-Steuerzeichen „CR“ und „LF“ dessen Ende. Die Telegramm Daten werden hexa-dezimal im ASCII-Format ausgegeben:

z.B.: 197dez (1Byte) = C5hex (1 Byte) = C (1 Byte) 5 (1 Byte) ASCII

Da ein Datenbyte durch 2 ASCII-Zeichen dargestellt wird, verdoppelt sich die Anzahl der zu übertragenden Datenbytes gegenüber dem RTU-Modus.

Start 1 char	Adresse 2 char	Steuerbefehl 2 char	Daten 0 - 2 x 100 char	Prüfsumme LRC 2 char	Ende 2 char
:					CR LF

3.4.2 Berechnung der LRC-Prüfsumme

Die LRC - Prüfsumme (Longitudinal Redundancy Check) wird vom Sender aus allen übertragenen Bytes berechnet (ohne „:“, „CR“, „LF“) und dann in der Botschaft vor „CR“ und „LF“ eingefügt.

Der Empfänger berechnet die LRC-Prüfsumme erneut und vergleicht sie mit der empfangenen Prüfsumme. Stimmen die Werte nicht überein, dann ist von einem Übertragungsfehler auszugehen und die empfangenen Daten werden verworfen.

Das höherwertige ASCII-Zeichen der 8 Bit großen Prüfsumme wird im Telegramm vor dem niederwertigen ASCII-Zeichen gesendet.

Berechnung der Prüfsumme (Programmbeispiel in C):

```
lrc = 0;
for(i = 1; i < Telegrammlänge -4; i++)
    lrc = lrc + Telegramm Daten [i];
```

```
lrc = 0xFF - lrc;
lrc = lrc + 1;
```

4 Einlernen von Sensoren

Der Empfänger verwaltet nur die Daten von Funksensoren deren Identifikationscode bekannt sind, d.h. im EEPROM abgespeichert wurden. Entsprechend Tabelle 2 sind jedem Sensor 10 Register zugeordnet, wobei jeweils die ersten drei Register den Identifikationscode enthalten.

Der Sensor-Identifikationscode wird entweder direkt über ein MODBUS-Telegramm in die Register geschrieben, oder aber im Lernmodus aus einem empfangenen „Lern-Funktelegramm“ selbstständig abgespeichert.

4.1 Einlernen über MODBUS - Schreibbefehl

Mit dem Steuerbefehlen „Register Schreiben“ (10hex oder 06hex) kann der Identifikationscode direkt in die entsprechenden Register geschrieben werden. Der Identifikationscode (ORG-Byte und ID-Bytes) identifiziert jeden Sensor eindeutig und ist auf dem Geräteetikett der Funk-sensoren vermerkt.

Beispiel: Sensor 2 mit ID = 01 23 D5 E7 (hex) und ORG-Byte = 07 (hex) einlernen

Master - Telegramm im Übertragungsmodus RTU:

Gerät	Befehl	Startadresse		Anzahl Register		Anzahl Bytes	Daten Register 0A		Daten Register 0B		Daten Register 0C		Prüfsumme	
		H Byte	L Byte	H Byte	L Byte		H Byte	L Byte	H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	10	00	0A	00	03	06	00	07	01	23	D5	E7	CRC	

Slave - Antworttelegramm im Übertragungsmodus RTU:

Gerät	Befehl	Startadresse		Anzahl Register		Prüfsumme	
		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	10	00	0A	00	03	CRC	

Wird nun ein Funktelegramm des Sensor mit der ID = 01 23 D5 E7 und ORG = 7 empfangen, dann werden die Messwerte in die entsprechenden Datenbytes geschrieben und der Überwachungstimer auf den Wert „0“ zurückgesetzt.

4.2 Einlernen über Lerntaste des Funksensors

Mit dem Steuerbefehl „Bit(s) Schreiben“ (0Fhex oder 05hex) kann ein Lernbit (oder mehrere) mit dem Wert „1“ beschrieben werden. Damit wird der Empfänger für einen ausgewählten Sensor in den Lernmodus versetzt. Im Lernmodus wartet der Empfänger auf ein Funktelegramm eines Sensors, bei dem der Lerntaster betätigt wurde und schreibt dann den empfangenen Identifikationscode direkt in die entsprechenden Register.

Beispiel: Sensor 29 in den Lernmodus schalten (Bit 28 = 1)

Master - Telegramm im Übertragungsmodus RTU:

Slave Adresse	Befehl	Startadresse		Anzahl Bitsr		Anzahl Bytes	Daten	Prüfsumme	
		H Byte	L Byte	H Byte	L Byte			L CRC	H CRC
02	0F	00	1C	00	01	01	01	CRC	

Slave - Antworttelegramm im Übertragungsmodus RTU:

Slave Adresse	Befehl	Startadresse		Anzahl Bits		Prüfsumme	
		H Byte	L Byte	H Byte	L Byte	L CRC	H CRC
02	0F	00	1C	00	01	CRC	

Nach Empfang eines Funk-Lerntelegramms wird das Lernbit automatisch gelöscht. Es ist somit nicht nötig ein neues Telegramm zum Zurücksetzen des Lernbits zu senden.

5 Daten auslesen

Alle im Kapitel 2.4 beschriebenen Register- und Bit-Werte besitzen Lesezugriff, wobei zum Auslesen von Registern und Bits über den Bus unterschiedliche Steuerbefehle verwendet werden.

5.1 Register Auslesen

Mit dem Steuerbefehl „Register lesen“ (03_{hex} oder 04_{hex}) können 1 - 50 Register ausgelesen werden. Versucht der Master mehr als 50 Register auszulesen, dann antwortet der Slave mit einem Fehlertelegramm (Fehlercode 02_{hex}).

Beispiel: Daten des Sensor 29 auslesen (Register 280_{dez} (0118_{hex}) bis 289_{dez} (0121_{hex}))

Master - Telegramm im Modus RTU		Slave - Antworttelegramm im Modus RTU	
Beschreibung	Wert (Hex)	Beschreibung	Wert (Hex)
Slave Adresse	02	Slave Adresse	02
Befehl	03	Befehl	03
Startadresse High	01	Anzahl Bytes	14
Startadresse Low	18	Register Wert High (0118) not used	00
Anzahl Register High	00	Register Wert Low (0118) ORG	07
Anzahl Register Low	0A	Register Wert High (0119) ID-Byte-3	01
Prüfsumme Low	CRC	Register Wert Low (0119) ID-Byte-2	23
Prüfsumme High		Register Wert High (011A) ID-Byte-1	D5
		Register Wert Low (011A) ID-Byte-0	E7
		Register Wert High (011B) not used	00
		Register Wert Low (011B) Data-Byte-3	E7
		Register Wert High (011C) not used	00
		Register Wert Low (011C) Data-Byte-2	2A
		Register Wert High (011D) not used	00
		Register Wert Low (011D) Data-Byte-1	5F
		Register Wert High (011E) not used	00
		Register Wert Low (011E) Data-Byte-0	0F
		Register Wert High (011F) Receive-Time	01
		Register Wert Low (011F) Receive-Time	20
		Register Wert High (0120) not used	00
		Register Wert Low (0120) not used	00
		Register Wert High (0121) not used	00
		Register Wert Low (0121) not used	00
		Prüfsumme Low	CRC
		Prüfsumme High	

5.2 Bits Auslesen

Mit dem Steuerbefehl „Bits lesen“ (01_{hex} oder 02_{hex}) können ein Bit oder mehrere Bits (siehe Tabellen 4 und 5 in Kap 2.4.3) ausgelesen werden.

Beispiel: Lernbits des Sensoren 29-30 auslesen (28_{dez} (0001C_{hex}) bis 29_{dez} (0001D_{hex}))

Master - Telegramm im Modus RTU		Slave - Antworttelegramm im Modus RTU	
Beschreibung	Wert (Hex)	Beschreibung	Wert (Hex)
Gerät	02	Gerät	02
Befehl	01	Befehl	01
Startadresse High	00	Anzahl Bytes	01
Startadresse Low	1C	Bitwerte 0,0,0,0,0,0,Bit29,Bit28	03
Anzahl Bits High	00	Prüfsumme Low	CRC
Anzahl Bits Low	02	Prüfsumme High	
Prüfsumme Low	CRC		
Prüfsumme High			