thermokon
Sensortechnik GmbH

Schnittstellenbeschreibung
für
SRC/STC RS485 MODBUS

1 Änderungsindex

Version	Datum	Beschreibung
B	25.08.2008	(1) STC Firmware V2.0 • Sende Funktionalität • 8 Kanäle, die die Datenübertragung vom Modbus-Netzwerk ins EasySens-Netzwerk ermöglichen (2) SRC und STC-Beschreibung in gemeinsamer Datei
C	24.11.2011	Korrekturen Beispiel SAB02
D	30.11.2011	Ergänzungen
E	14.10.2013	Ergänzungen für neue Hardware (Rev B) und Firmware 3.0.0
F	02.12.2013	Ergänzungen bei der Registerbeschreibung

2 Einführung

Das vorliegende Dokument beschreibt die Funktionen des Funkempfängers SRC-RS485-MODBUS bzw. des Funktransceivers STC-RS485-MODBUS. Das von der Fa. Modicon entwickelte MODBUS-Protokoll ist ein offengelegtes Protokoll zur Kommunikation mehrerer intelligenter Geräte auf Master-Slave-Basis.
Beide Geräte unterstützen die Abbildung von bis zu 32 Easysens-Sensoren im Modbus-Netzwerk und der Funktransceiver zusätzlich bis zu 8 Easysens-Sender zur Datenübertragung vom Modbus- ins Easysens-Netzwerk.
Bei Verwendung eines SRC-RS485 Modbus entfallen die Kapitel 2.6.2 und 6 dieser Beschreibung.

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

Ab der Firmware Version 1.4 können auch Taster ausgewertet werden.

3 Gerätebeschreibung

3.1 Hardware Installation

Der Transceiver kann mittels eines Twisted-Pair-Kabels (Leitungswiderstand 120 Ohm) verbunden werden. Detaillierte Informationen zur Inbetriebnahme und Montage entnehmen Sie bitte dem Produktdatenblatt des SRC-RS485-Modbus bzw. des STC-RS485-Modbus und dem Datenblatt wiring_rs485_network.pdf.

3.2 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.

3.3 Protokoll

Der Funkempfänger SRC-RS485-MODBUS und der Funktransceiver STC-RS485-MODBUS sindSlave-Busteilnehmer, die nur auf Anforderung des Masters auf den Bus senden dürfen. Das Protokoll entspricht den Vorgaben aus:
- MODBUS Application Protocol Specification V1.1
- MODBUS over Serial Line Specification & Implementation guide V1.0

3.4 Konfigurationsmöglichkeiten

Mittels Dippschalter kann das Gerät an die jeweilige Bustopologie angepasst werden. Einstellbar sind:
- die Busadresse des Gerätes (1 - 247)
- Busabschlusswiderstand 120 Ohm
- Übertragungsmodus RTU oder ASCII
- Baudrate 9600, 19200, 38400 oder 57600 (38400 und 57600 erst ab Hardware Rev B und Firmware 3.0.0 möglich)
- Parität gerade, Parität ungerade oder keine 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“ bzw. „Produktblatt_STC_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

3.5 Unterstützte Steuerbefehle

Folgende MODBUS - Steuerbefehle werden unterstützt:

Beschreibung	Funktionscode	
Bitstelle(n) lesen	01 (hex)	1 (dez)
Register lesen	03 (hex)	3 (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

Hinweis:

Die Funktionscodes 02 und 04 sind ab Hardware Rev B und Firmware 3.0.0 nicht mehr verfügbar !

3.6 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.

3.6.1 Sensordaten

3.6.1.1 Registerzuordnung der Sensordaten

Ein Register besteht per Definition in MODBUS-Geräten aus 16 Bit. In den Registern 1-320 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 1 - 10_{dez}
Sensor 2 Register 11 - 20_{dez}
:
Sensor 32 Register 311 - 320_{dez}

Holding Register – Modbusbefehle 03_{hex}, 06_{hex} und 10_{hex}

Register	Daten-Adresse	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	
1 R/W	0	not used								ORG								Daten Sensor 1
2 R/W	1	ID-Byte-3								ID-Byte-2								
3 R/W	2	ID-Byte-1								ID-Byte-0								
4 R	3	not used								Data-Byte-3								
5 R	4	not used								Data-Byte-2								
6 R	5	not used								Data-Byte-1								
7 R	6	not used								Data-Byte-0								
8 R	7	Receive-Time-Byte-1								Receive-Time-Byte-0								
9 R/W	8	not used								Aktor Kanal								
10 R	9	not used								not used								
:																		
:																		
311 R/W	310	not used								ORG								Daten Sensor 32
312 R/W	311	ID-Byte-3								ID-Byte-2								
313 R/W	312	ID-Byte-1								ID-Byte-0								
314 R	313	not used								Data-Byte-3								
315 R	314	not used								Data-Byte-2								
316 R	315	not used								Data-Byte-1								
317 R	316	not used								Data-Byte-0								
318 R	317	Receive-Time-Byte-1								Receive-Time-Byte-0								
319 R/W	318	not used								Aktor Kanal								
320 R	319	not used								not used								

Tabelle 2: Registerzuordnung der Sensordaten

3.6.1.2 Identifikationscode

Die jeweils ersten 3 Register enthalten den Identifikationscode eines Sensors, der jeden Sensor eindeutig identifiziert. Er besteht 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.

3.6.1.3 Data-Bytes für Sensoren (ORG = 6 oder ORG = 7)

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
- 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 3.6.1.9)

Data-Byte 2

- Sollwertsteller bei SR04 xx
- Feuchtewert bei SR04 rH

Data-Byte 3

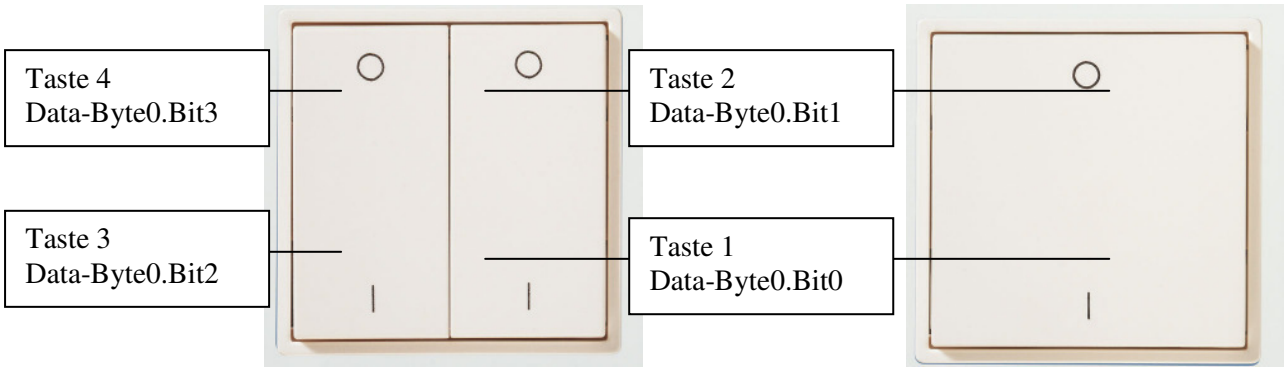
- Lüfterstufe bei SR04 xx
- Sollwertsteller bei SR04 rH
- Fensterkontakt SRW01

3.6.1.4 Data-Bytes für Taster (ORG = 5)

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

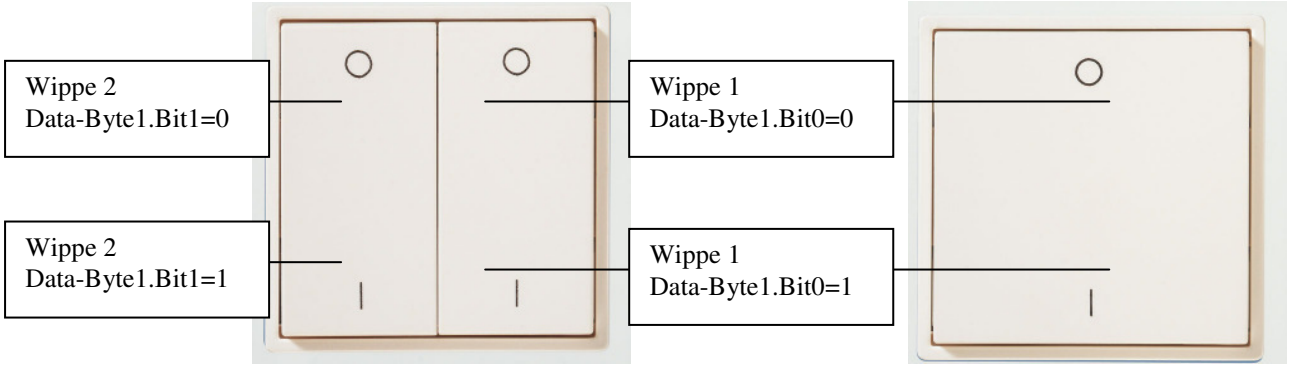
Data-Byte 0

- 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.

3.6.1.5 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.

3.6.1.6 Aktor Kanal

Ein Wert im Bereich 1…8 bewirkt, dass beim Telegramm-Empfang eines eingelernten Sensors automatisch die Daten eines Sendekanals (1…8) an den Aktor ausgesendet werden. Anwendungsbeispiel ist die direkte Kopplung eines Sensors mit einem SAB01-Stellantrieb.

3.6.1.7 Registerzuordnung Modbus-Konfiguration

Holding Register – Modbusbefehle 03_{hex}, 06_{hex} und 10_{hex}

Register	Daten-Adresse	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	
321R/W	320	Min-Response-Time-Byte-1								Min-Response-Time-Byte-0								min. Antwortzeit

Tabelle 3: Registerzuordnung für minimale Antwortzeit

Das Register 321 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.

Hinweis: Ab Hardware Rev B und Firmware 3.0.0 wirkt dieser Parameter nur noch im ASCII Modus. Im RTU Modus hat dieser Parameter keine auswirkungen.

3.6.1.8 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. Bit1 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 Lerntaste 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“).

Coils – Modbusbefehle 01_{hex}, 05_{hex} und $0F_{hex}$

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

Tabelle 4

3.6.1.9 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 33 bis 64 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.

Coils – Modbusbefehle 01_{hex}, 05_{hex} und $0F_{hex}$

Bit	Daten-Adresse	Speicherbereich	Wert = 1 ==> Datenbyte 1 invertieren
33 R/W	32	EEPROM	Sensor 1
34 R/W	33	EEPROM	Sensor 2
		:	
64 R/W	63	EEPROM	Sensor 32

Tabelle 5

3.6.2 EasySens-Senderdaten

(Dieses Kapitel ist nur für den STC-RS485 gültig!!)

3.6.2.1 Registerzuordnung der Senderdaten

Die Register der Sender sind gleich aufgebaut wie die der Sensoren. Ein Register besteht per Definition in MODBUS-Geräten aus 16 Bit. In den Registern 401-480 liegen die Daten zur Abbildung von bis zu 8 EasySens-Sendern, wobei jedem Sender 10 Register zugeordnet sind (siehe Tabelle 2):

Sender 1 Register 401 - 410dez

Sender 2 Register 411 -420dez

:

Sender 8 Register 471 - 480dez

Der Schreibbefehl „Mehrere Register schreiben(0x10)“ kann auf alle Register eines Senders angewendet werden. Übernommen werden aber nur die Daten für das ORG-Byte, die Data-Bytes und das STATUS-Byte! Die vom Modbus-Netzwerk empfangenen Daten werden dem EnOcean_Protokoll entsprechend versendet. Um einen Sendevorgang auszulösen, muss das entsprechen Sendebit gesetzt werden (s. Kapitel 2.6.2.5).

Holding Register – Modbusbefehle 03_{hex}, 06_{hex} und 10_{hex}

Register	Daten-Adresse	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	
401 R/W	400	not used								ORG								Daten Sender 1
402 R	401	ID-Byte-3								ID-Byte-2								
403 R	402	ID-Byte-1								ID-Byte-0								
404 R/W	403	not used								Data-Byte-0								
405 R/W	404	not used								Data-Byte-1								
406 R/W	405	not used								Data-Byte-2								
407 R/W	406	not used								Data-Byte-3								
408 R/W	407	not used								STATUS								
409 R	408	not used								not used								
410 R	409	not used								not used								
:																		
:																		
471 R/W	470	not used								ORG								Daten Sender 8
472 R	471	ID-Byte-3								ID-Byte-2								
473 R	472	ID-Byte-1								ID-Byte-0								
474 R/W	473	not used								Data-Byte-0								
475 R/W	474	not used								Data-Byte-1								
476 R/W	475	not used								Data-Byte-2								
477 R/W	476	not used								Data-Byte-3								
478 R/W	477	not used								STATUS								
479 R	478	not used								not used								
480 R	479	not used								not used								

Tabelle 6: Registerzuordnung der Senderdaten

3.6.2.2 Identifikationscode

Die Register 2 und 3 jedes Senders enthalten einen eindeutigen Identifikationscode, der vom Basis-Identifikationscode des EasySens-Moduls abgeleitet wird. Die Zuordnung ist: Sender 1 = BasisID+0, Sender1 = BasisID+1, …, Sender8 = BasisID+7

Diese Register sind mit „R“ gekennzeichnet und können nur gelesen werden.

3.6.2.3 ORG-Byte und Data-Bytes für Sender

Das ORG-Register legt fest welcher Telegrammtyp (ORG-Byte) versendet werden soll. Vier Register stehen für die Daten zur Verfügung.
Die Bedeutung der Datenbytes ist verschieden und abhängig von den zu übertragenden Werten. Beachten Sie dazu bitte die entsprechenden Beschreibungen.

Die Register sind mit „R/W“ gekennzeichnet und verfügen über Lese- und Schreibzugriff.

3.6.2.4 Status-Byte

Im Status-Byte können zusätzliche Informationen gemäß dem EnOcean-Protokoll übertragen werden.
Das Register ist mit „R/W“ gekennzeichnet und verfügt über Lese- und Schreibzugriff.

3.6.2.5 Telegramm senden

Die in Tabelle 7 aufgelisteten Bitwerte sind mit „R/W“ gekennzeichnet und haben sowohl Lese- als auch Schreibzugriff. Zum Senden eines Telegramms muss das Coil 1 gesetzt werden. Nach erfolgreichem Senden, wird das Coil automatisch auf 0 zurückgesetzt.

Coils – Modbusbefehle 01_{hex}, 05_{hex} und $0F_{hex}$

Bit	Daten-Adresse	Wert = 1 ==> Sendemodus aktiv
65 R/W	64	Sendebit Sender 1
66 R/W	65	Sendebit Sender 2
	:	
72R/W	71	Sendebit Sender 8

Tabelle 7: Sendebits

4 Datenübertragung

4.1 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.

4.2 Datenrahmen

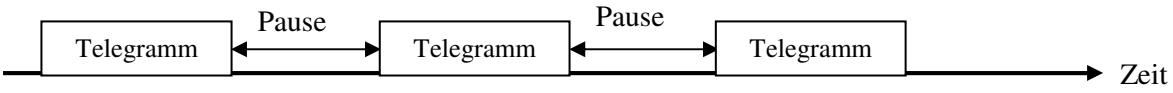
Die Daten werden nach streng definierten Vorgaben auf den Bus gesendet:

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 Busteilnehmern Übertragungsfehler erkennen.

4.3 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 * Wort-Übertragungszeit (11 Bit). Bei 9600 Baud müssen damit mindestens 4 ms und bei 19200 mindestens 2 ms. zwischen zwei Telegrammen vergehen.

4.3.1 Telegrammaufbau

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

4.3.2 Berechnung der CRC-Prüfsumme

Die CRC - Prüfsumme (Cyclicyal 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 polynominal für CRC16
        }

        return(crc_temp);
}
```

4.4 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.

4.4.1 Telegrammaufbau

Das ASCII-Steuerzeichen „ : “ bezeichnet immer den Anfang eines Telegramms und die ASCII-Steuerzeichen „CR“ und „LF“ dessen Ende. Die Telegrammdaten werden hexadezimal 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**

4.4.2 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 + Telegrammdaten [i];

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

5 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.

5.1 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 Funksensoren 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.

5.2 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 29 = 1, d.h. Daten-Adresse 28)

Master - Telegramm im Übertragungsmodus RTU:

Slave Adresse	Befehl	Startadresse		Anzahl Bits		Anzahl Bytes	Daten	Prüfsumme	
		H Byte	L Byte	H Byte	L Byte		H 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.

6 Daten auslesen

Alle im Kapitel 2.6 beschriebenen Register- und Bit-Werte besitzen Lesezugriff, wobei zum Auslesen von Registern und Bits über den Bus unterschiedliche Steuerbefehle verwendet werden.

6.1 Register Auslesen

Mit dem Steuerbefehl „Register lesen“ (03_{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 281_{dez} (Daten Adresse = 0118_{hex}) bis 290_{dez} (Daten Adresse = 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	

Tabelle 8

6.2 Bits Auslesen

Mit dem Steuerbefehl „Bits lesen“ (01_{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 (Daten Adresse = $0001C_{hex}$ $0001D_{hex}$)

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			

Tabelle 9

7 Daten versenden

(Dieses Kapitel ist nur für den STC-RS485 gültig!!)

7.1 Register schreiben

Mit dem Schreibbefehl „Write Multiple Registers(0x10)“ können die Register eines zu sendenden Telegramms beschrieben werden. Alternativ steht die Option jedes Register einzeln zu beschreiben (Befehl „Write Single Register“ (0x06)) zur Verfügung.

Sender 1 (Register 401-410)

Master - Telegramm im Modus RTU		**Slave - Antworttelegramm im Modus RTU**	
Beschreibung	Wert (Hex)	Beschreibung	Wert (Hex)
Slave Adresse	02	Slave Adresse	02
Befehl	10	Befehl	10
Startadresse High	01	Startadresse High	01
Startadresse Low	90	Startadresse Low	90
Anzahl Register High	00	Anzahl Register High	00
Anzahl Register Low	0A	Anzahl Register Low	0A
Anzahl Bytes	14	Prüfsumme Low	CRC
Wert Register1 High	00	Prüfsumme High	
Wert Register1 Low (ORG)	07		
Wert Register2 High	00		
Wert Register2 Low	00		
Wert Register3 High	00		
Wert Register3 Low	00		
Wert Register4 High	00		
Wert Register4 Low (DB3)	AB		
Wert Register5 High	00		
Wert Register5 Low (DB2)	08		
Wert Register6 High	00		
Wert Register6 Low (DB1)	13		
Wert Register7 High	00		
Wert Register7 Low (DB0)	00		
Wert Register8 High	00		
Wert Register8 Low (STATUS)	00		
Wert Register9 High	00		
Wert Register9 Low	00		
Wert Register10 High	00		
Wert Register10 Low	00		
Prüfsumme Low	CRC		
Prüfsumme High			

7.2 Auslösen eines Sendevorgangs

Mit dem Steuerbefehl „Bit(s) Schreiben“ (0Fhex oder 05hex) kann ein Sendevorgang durch Setzen eines oder mehrerer Sendebits mit dem Wert „1“ ausgelöst werden. Die entsprechenden Werte der Sender-Register werden in einem Easysens-Telegramm versendet. Anschließend wird das Sendebit automatisch vom Transceiver zurück auf 0 gesetzt, d.h. es ist nicht notwendig es über ein weiteres Telegramm zurückzusetzen.

Beispiel: Werte Sender 1 versenden (Bit 65 = 1, d.h. Daten-Adresse 64)

Master - Telegramm im Übertragungsmodus RTU:

Slave Adresse	Befehl	Startadresse		Anzahl Bits		Anzahl Bytes	Daten	Prüfsumme	
		H Byte	L Byte	H Byte	L Byte		H Byte	L CRC	H CRC
02	0F	00	40	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
2	0F	00	40	00	01	CRC	

7.3 Enocean-Telegramm

Folgendes Funktelegramm wird gemäß den zuvor übergebenen Werten übertragen. Die ID des Senders ist: 0xFFED8F00

SYNC-BYTE 1	0xA5
SYNC-BYTE 0	0x5A
H-SEQ LENGTH	0x6B
ORG	0x07
DATA-BYTE3	0xAB
DATA-BYTE2	0x08
DATA-BYTE1	0x13
DATA-BYTE0	0x00
ID-BYTE3	0xFF
ID-BYTE2	0xED
ID-BYTE2	0x8F
ID-BYTE0	0x00
STATUS	0x00
Checksumme	CS

8 Konfigurationssoftware

Mittels einer RS485-Schnittstelle (z.B. RS232-RS485-Pegelwandler z.B. ADAM-4520) kann mit der Konfigurationssoftware auf den Modbus zugegriffen werden. Die Konfigurationssoftware ist zur Inbetriebnahme des SRC-RS485-Modbus bzw. STC-RS485-Modbus nicht zwingend erforderlich. Sie können jedes beliebige Programm verwenden, welches Modbus-Telegramme erzeugt.

9 Software Installation

Zum Installieren der Konfigurationssoftware muss die Setup-Datei „setup.exe“ gestartet werden. Bitte beachten Sie, dass Sie zur Installation Administratorrechte besitzen müssen. Während der Installation folgen Sie den Bildschirmanweisungen.

Nach erfolgreicher Installation können Sie die Konfigurationssoftware über das Startmenü\Programme\Thermokon starten.

Unterstützte Betriebssysteme:	Windows9x; WindowsNT; WindowsMe; Windows2000; WindowsXP; WindowsServer, Windows 7

10 Konfiguration des Transceivers

10.1 Konfigurationssoftware

Mit der Konfigurationssoftware können Sensoren in die verschiedenen Register eingelernt und bei dem STC-RS485 die Senderkanäle geprüft werden. Register und somit die Daten der Sensoren und Sender können ausgelesen und zur Anzeige skaliert werden. Das SRC- und STC-RS485-Modbus haben insgesamt 320 Register für die Sensoren und das STC-RS485 Modbus zusätzlich 80 für die Sender. Ein Sensor belegt mit all seinen Daten 10 Register. Somit stehen die Register 1-10; 11-20 ... 311-320; jeweils für einen Sensor. Die Senderdaten belegen ebenfalls jeweils 10 Register, beginnend bei der Registeradresse 401. Sender1 belegt 401-410, Sender2 411-420,…,Sender8 471-480. Die Belegung der einzelnen Register mit den Datenbytes der Sensoren und Sender ist im Kapitel 3.6 beschrieben.

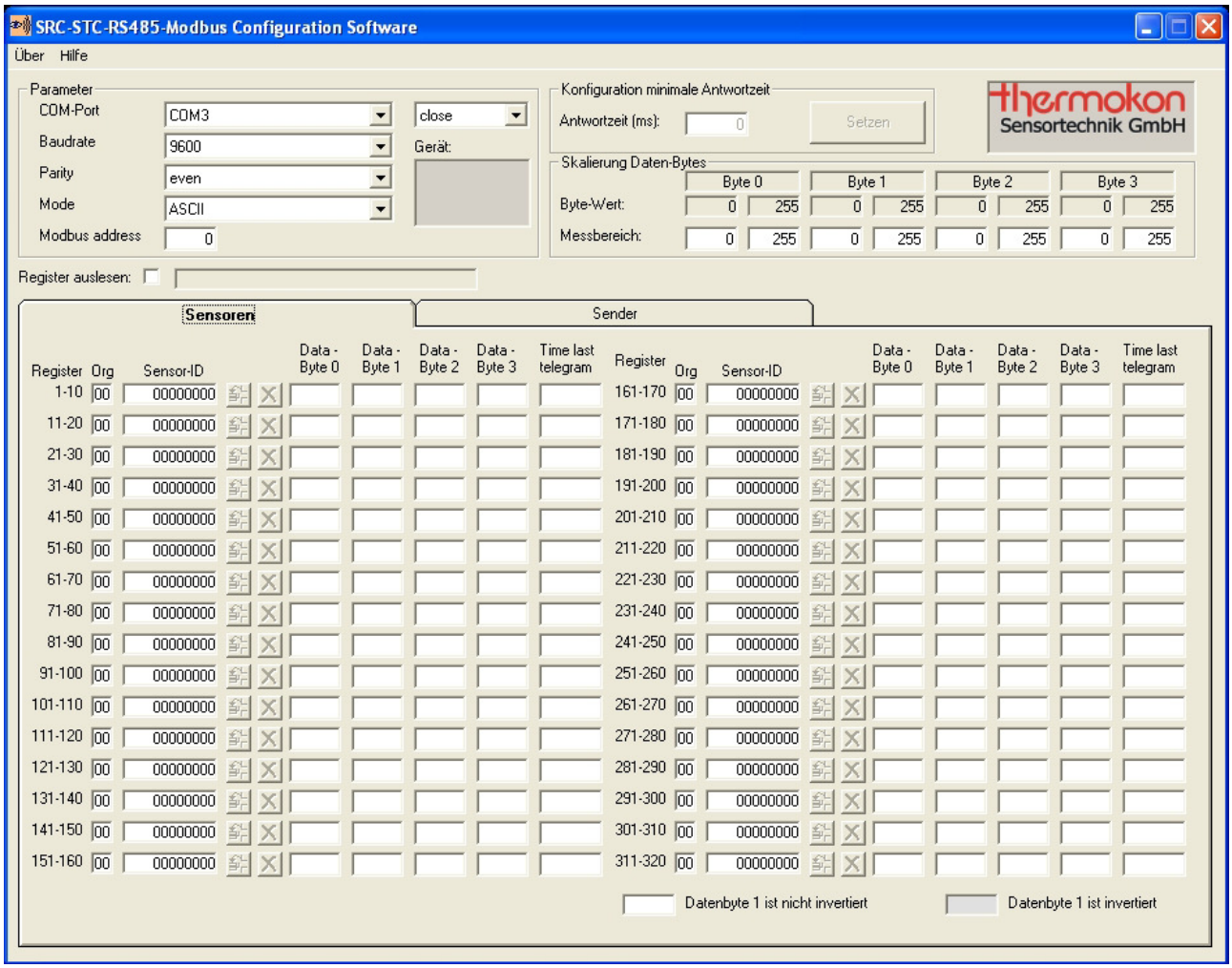


Abbildung 10-1: Konfigurationssoftware

10.2 Parameter-Frame

Mit der Konfigurationssoftware kann mittels eines COM-Ports auf den Modbus zugegriffen werden. Im „Parameter"-Frame können Hardware-Einstellungen getätigt werden. Diese müssen mit dem Modbus-Empfänger übereinstimmen, um eine Verbindung herzustellen.

Folgende Auswahlmöglichkeiten gibt es:

- COM-Port
- Baudrate
- Parität zur Einstellung ob keine, gerade oder ungerade Parität
- Modus zur Einstellung der Übertragung ASCII oder RTU
- Modbusadresse

Im Feld „Modbus address" geben Sie die Adresse des Modbus-Empfängers ein welcher konfiguriert werden soll (Wert zwischen 0 und 255).

Über das Auswahlmenü hinter „COM-Port" kann der Port geöffnet „open" und geschlossen „close" werden.

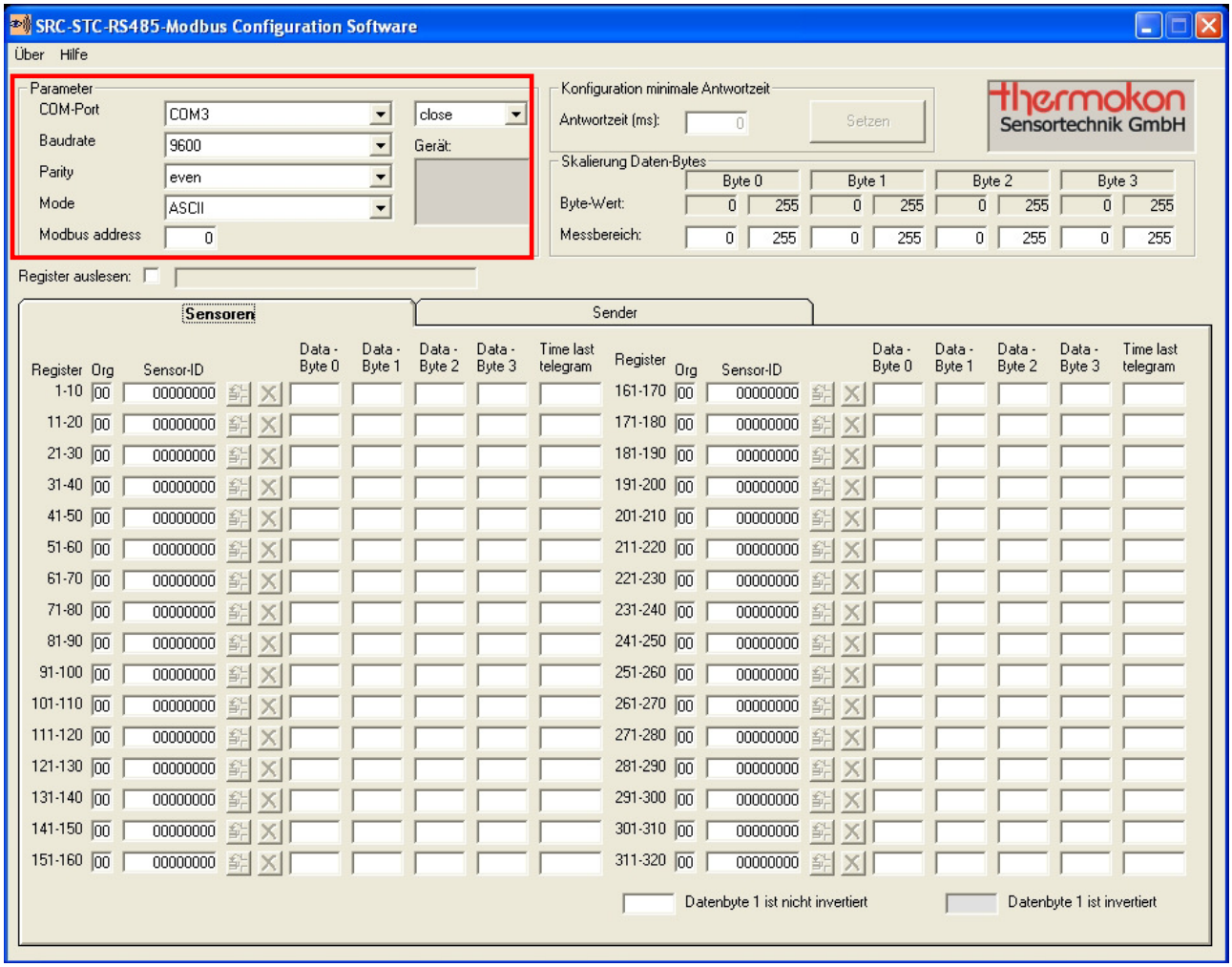


Abbildung 10-2: Parameter-Frame

Nach dem erfolgreichen Herstellen der Verbindung zum Gerät erscheint der Gerätetyp. Die Konfigurationssoftware erkennt automatisch, welcher Typ angeschlossen ist.

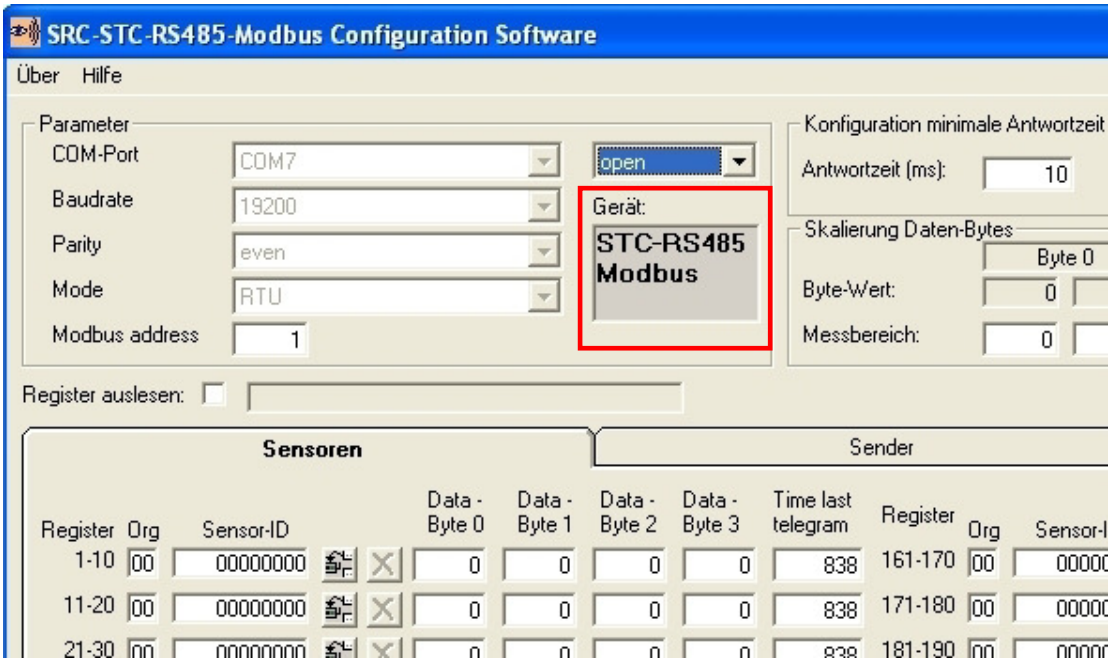


Abbildung 10-3: Geräte-Typ STC

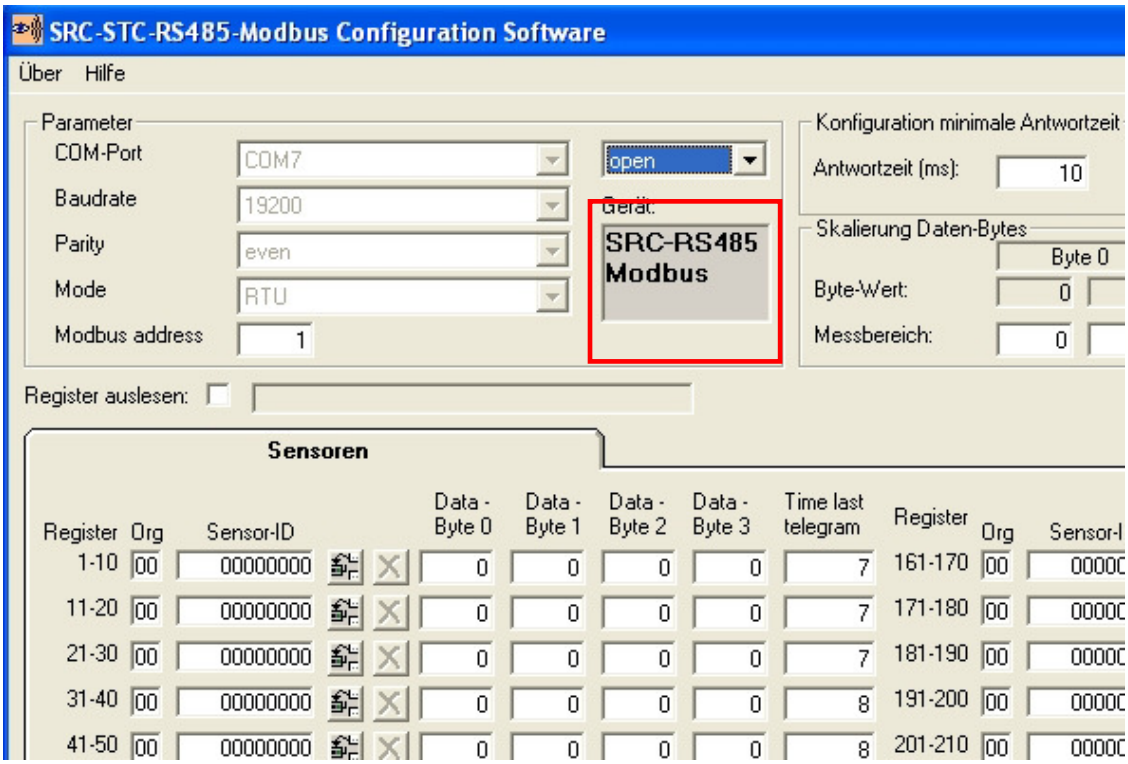


Abbildung 10-4: Geräte-Typ SRC

Minimale Antwortzeit

Im Frame „Konfiguration minimale Antwortzeit“ kann das Register 321 eingestellt werden. Diese Antwortzeit ist die minimale Zeit (ms) die vergehen muss, bevor ein Slave auf eine Master-Anfrage antworten darf. Voreingestellter Wert: 10 ms, kleinster erlaubter Wert 5 ms. Mit dem Button „Setzen“ werden die neuen Einstellungen der minimalen Antwortzeit übernommen.

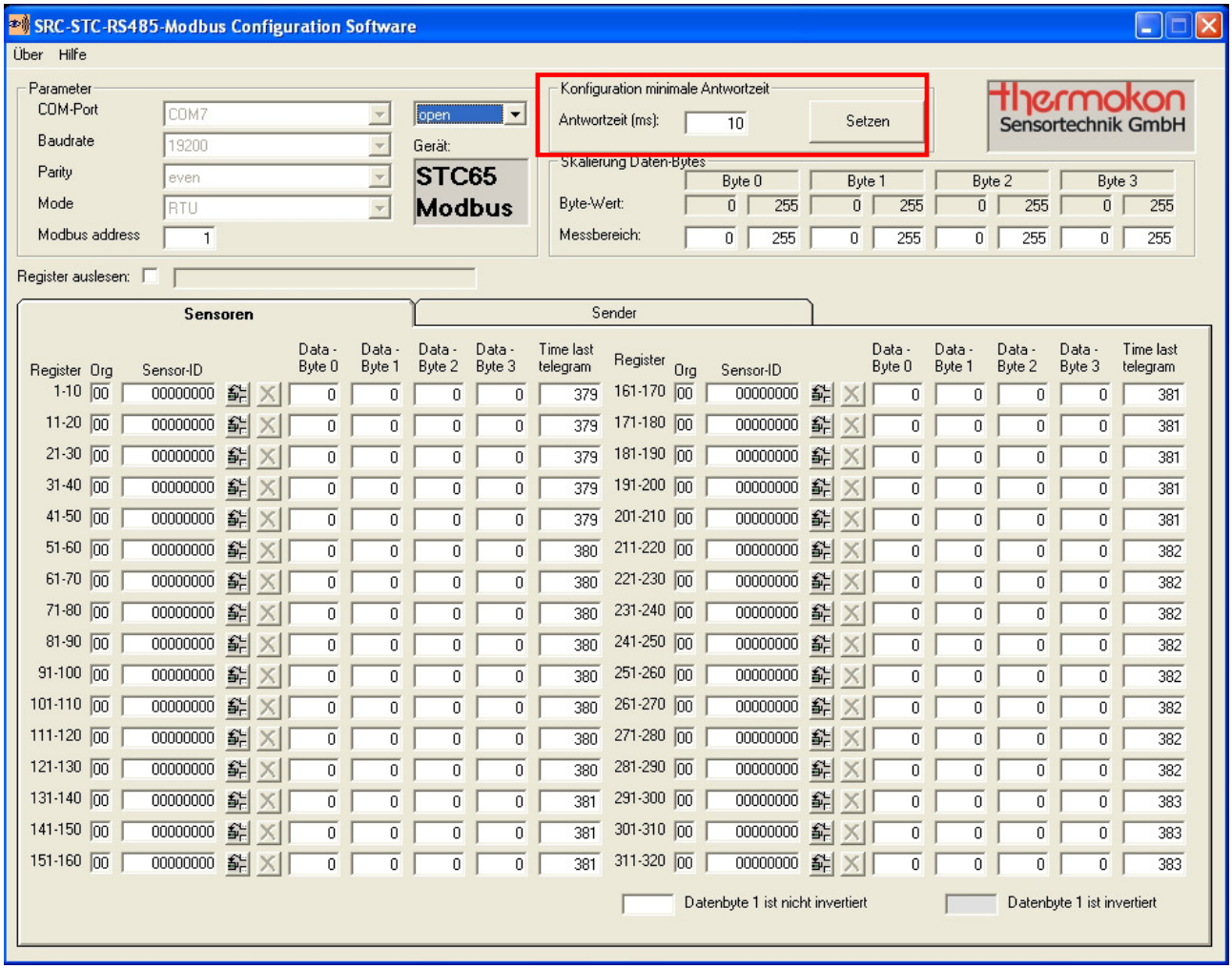


Abbildung 10-5: Minimale Antwortzeit

10.3 Register auslesen

Wird ein Häkchen bei „Register auslesen“ gesetzt, werden alle Register nacheinander ausgelesen und die Daten der Sensoren in der Konfigurationssoftware angezeigt. Ist eine Skalierung eingegeben worden, werden die Daten-Bytes skaliert.

Im Feld „Time last telegram“ wird die Zeit seit dem letzten empfangen Telegramm des Sensors angezeigt (in Sekunden).

Wenn die Temperatur invertiert ist (Daten-Byte 1), wird dies durch ein grau hinterlegtes Feld dargestellt. Nicht invertierte Werte sind weiß hinterlegt.

Bei Kommunikationsproblemen wird im Feld neben „Register auslesen“ eine Fehlermeldung ausgegeben.

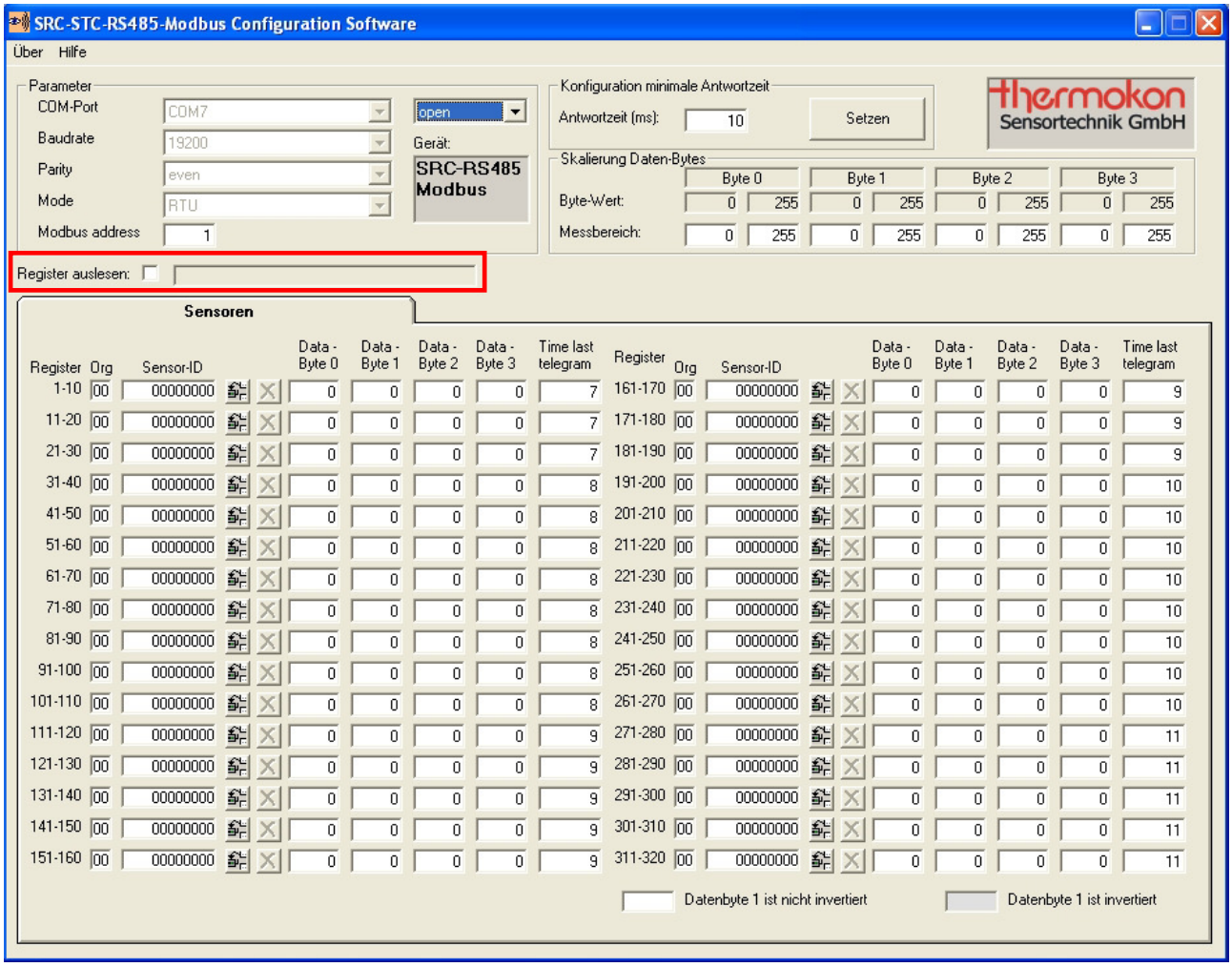


Abbildung 10-6: Sensor auslesen

10.4 Sensoren

10.4.1 Skalierung Daten-Bytes

Im „Skalierung Daten-Bytes“-Frame können die einzelnen Daten-Bytes der Sensoren skaliert werden. Die Skalierung dient nur zur einfacheren Anzeige der Sensorendaten.

Bsp.: Skalieren Sie z.B. für einen Außentemperaturfühler den Messbereich von –20°C bis +60°C.

Die Belegung der einzelnen Daten-Bytes können Sie dem Produktdatenblatt des Herstellers der Sensoren entnehmen.

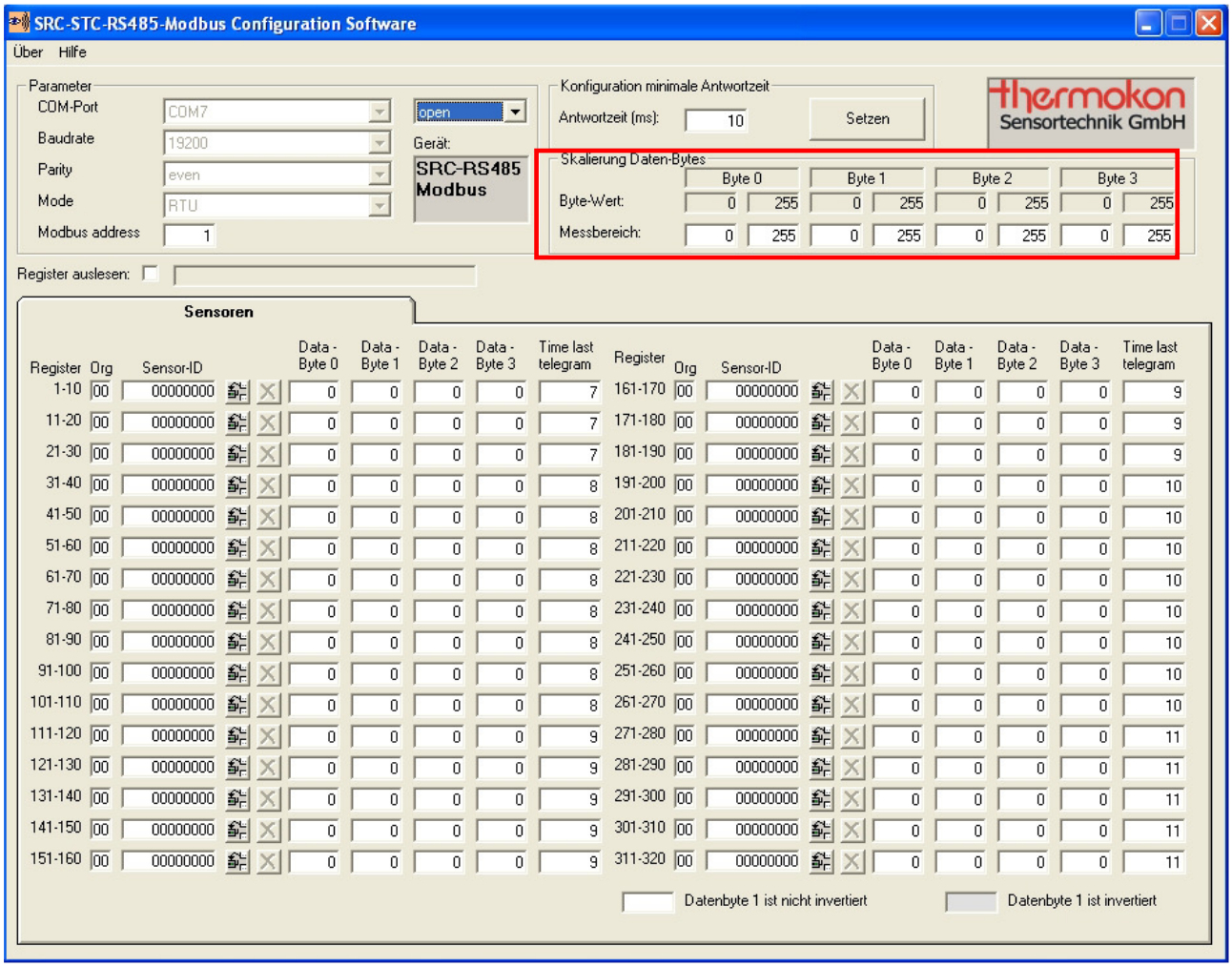


Abbildung 10-7: Skalierung

10.4.2 Sensor in den SRC/STC-RS485-Modbus einlernen

Sensoren können entweder manuell durch Eingabe der Sensor ID oder durch Drücken der Lerntaste eingelernt werden. Um ein Sensor einzulernen muss auf den Learn-in Button im Hauptmenü gedrückt werden.

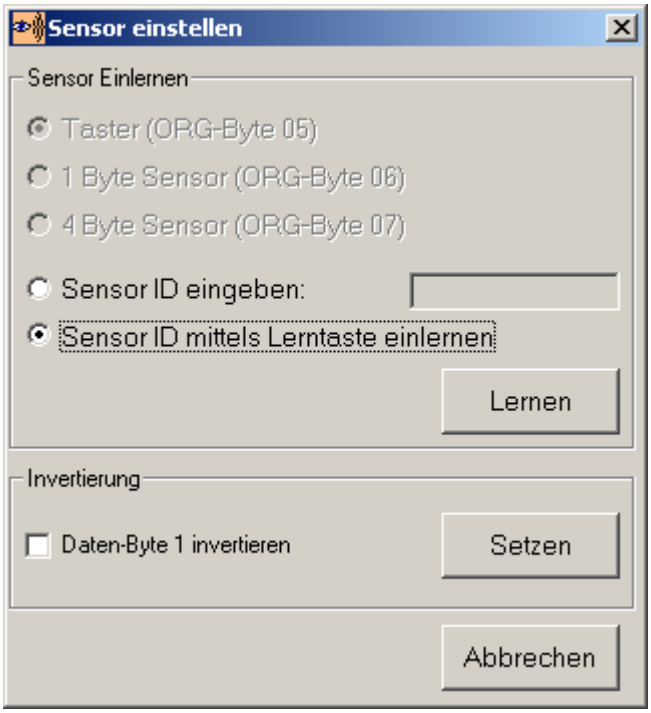


Abbildung 10-8: Sensor einstellen

- Sensor ID eingeben
  - Taster (ORG-Byte 05) z.B. PTM100
  - 1 Byte Sensor (ORG-Byte 06) z.B. Fensterkontakt
  - 4-Byte-Sensor (ORG-Byte 07) z.B. SR04x
  - Sensor ID eingeben
    - Besteht aus einer 4-Byte großen hexadezimalen Zahl
    - z.B. 0x00004E7A
  - Durch Drücken des „Lernen“ - Buttons wird der Sensor im Empfänger gespeichert
- Sensor ID mittels Lerntaste einlernen
  - Durch Drücken des „Lernen“ - Buttons wird der Empfänger in den Lernmodus gesetzt
  - Durch Drücken der Lerntaste am Sensor oder durch Drücken einer Taste am Schalter kann der Sensor / Schalter in den Empfänger eingelernt werden

10.4.3 Temperatur invertieren

Um das Daten-Byte 1 (Temperatur) eines Sensors zu invertieren muss auf den Learn-in Button gedrückt werden. Im Frame „Invertierung“ kann durch Aktivierung des Hakens die Temperatur invertiert werden. Durch Drücken des Buttons „Setzen“ werden die Einstellungen übernommen und zum Empfänger übertragen.

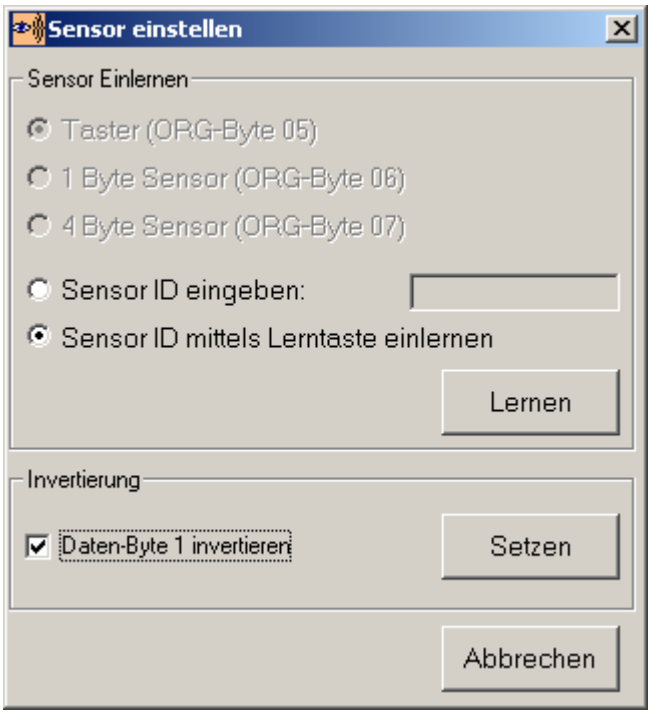


Abbildung 10-9: Sensor einstellen

10.5 Sender

In der Registerkarte „Sender“ können Funktelegramme über die 8 zur Verfügung stehenden Sendekanäle versendet werden.
Nach erfolgreichem Verbinden mit dem STC werden zuerst dessen Register ausgelesen. In den Feldern „Sender-ID“ der Kanäle erscheinen die jeweiligen Identifikationscodes. Diese ID´s werden automatisch von der ID des Sendemoduls abgeleitet und sind nicht veränderbar.
Die Felder ORG-BYTE, Data-Byte3-Data-Byte0 und Status sind veränderbar. Der gültige Werte-bereich liegt zwischen 0 und 255. Geänderte Werte werden rot markiert. Durch Drücken der Taste „Wert übernehmen“ werden die Werte in die entsprechenden Register des STC geschrieben. Das Funktelegramm wird durch Betätigen der Taste „Funktelegramm senden“ ausgelöst.

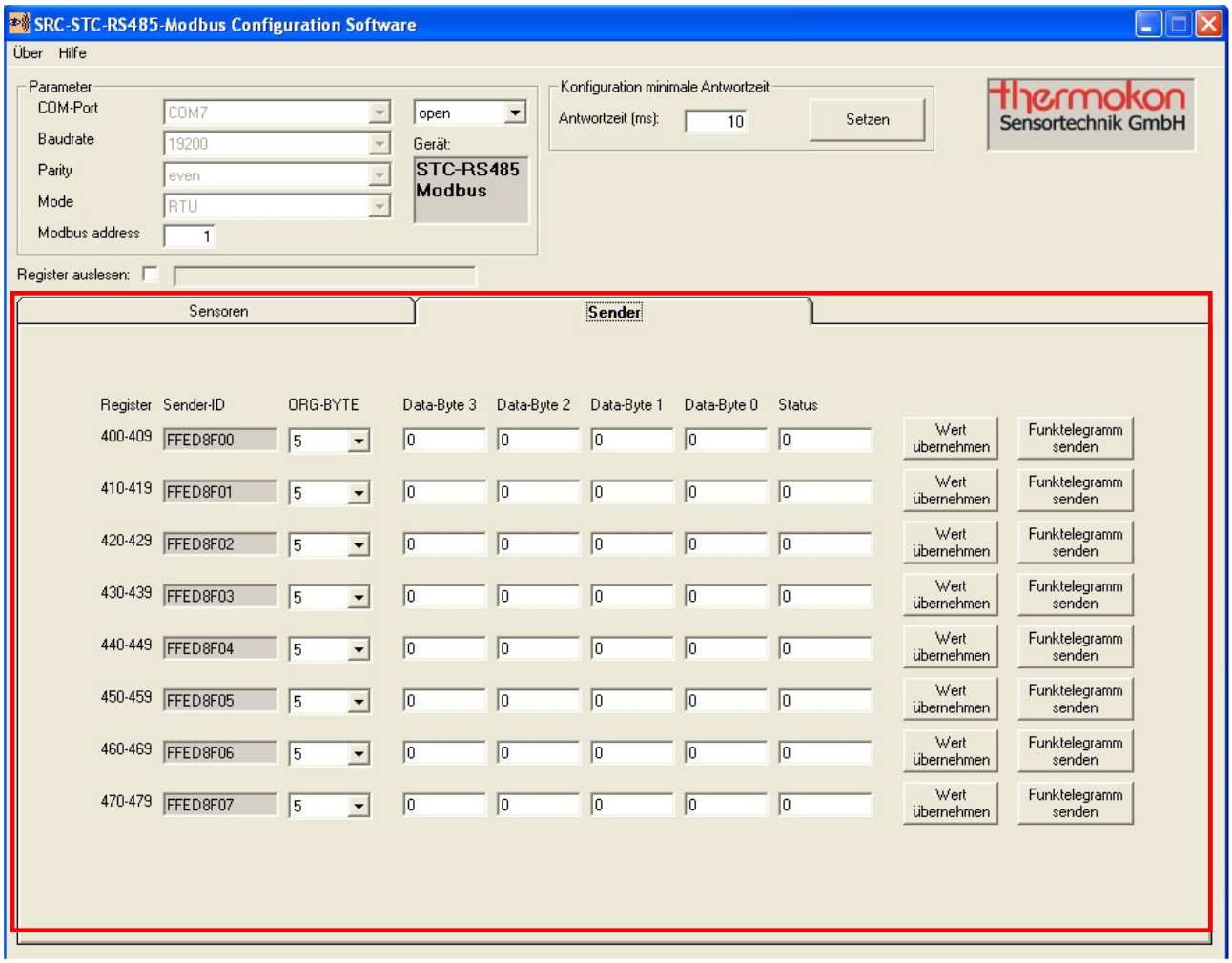


Abbildung 10.10 Registerkarte Sender

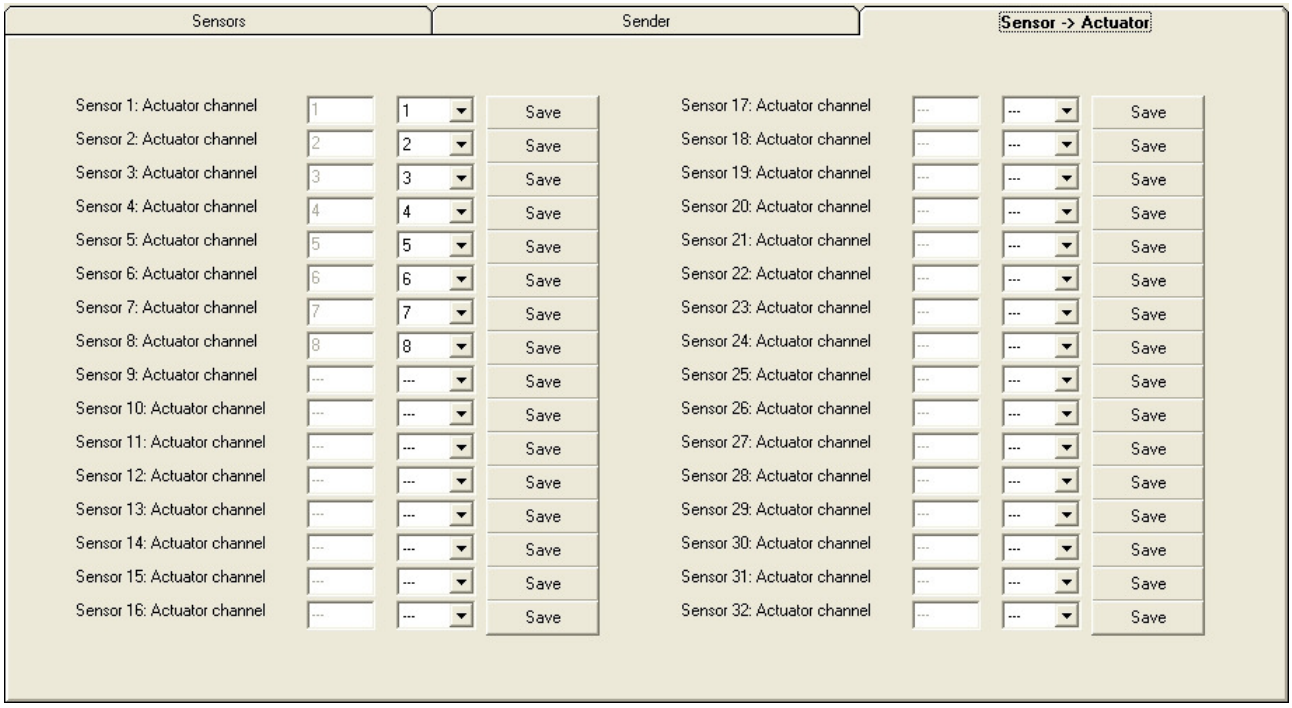


Abbildung 10-11 Zuordnung von Aktorkanal zu Sensor

10.6 Anhang

10.6.1 Einlernen eines SAB02

Folgende Sequenz muss nach einem Lerntelegramm des Aktors innerhalb der im SAB02-Datenblatt angegebenen Zeit vom Gateway (STC65) an den Aktor zurückgeschickt werden:

ORG-BYTE	=	0x07
DatenByte3	=	0x80
DatenByte2	=	0x08
DatenByte1	=	0x02,
DatenByte0	=	0xF0

Es gibt 2 Möglichkeiten zum Auslösen des obigen Telegramms:

(a) Benutzung der Thermokon-Konfigurationssoftware
1. Unter dem Reiter „Sender“ werden die oben angegebenen Werte eingetragen -> "Wert übernehmen"
2. Anschliessend ist das Lerntelegramm des Aktors, wie im SAB01-Datenblatt beschrieben, auszulösen
3. Innerhalb der im SAB01-Datenblatt angegebenen Zeit muss jetzt vom Gateway das Telegramm aus 1. weggeschickt werden -> "Funktelegramm senden"

(b) mit einer DDC oder einem Modbustool auf dem PC
1. Einem Sensorkanal wird ein Aktorkanal zugewiesen. Das geschieht über das achte Register eines Sensors (s.S.Kap 2.6.1.1 und Kap 10.5) => Register 8 *(Aktor Kanal)* für Sensor 1, Register 18 für Sensor 2, usw. In dieses Register wird der Aktorkanal (1-8) eingetragen
2. Jetzt müssen die Werte wie oben angegeben mittels DDC/Modbustool in die Register eingetragen werden(z.B. Aktorkanal1 => Register 401 und 404-407).
3. Das Gateway ist jetzt so eingestellt, dass bei Empfang des Sensors automatisch auf dem zugewiesenen Aktorkanal ein Telegramm mit den eingetragenen Werten versendet wird, d.h. hier besteht kein Zeitproblem mit der Generierung des Sendetelegramms (s. (a).3.)

Hinweis: Die Zuordnung sollte nach erfolgreichem Einlernen wieder entfernt werden, weil sonst nach jedem empfangenen Telegramm auf dem Sensorkanal das Gateway sofort wieder ein Telegramm an den Aktor zurückschickt!