

SMA-Data

Spezifikation

Ausgabe 1.25

**Definition und Beschreibung
des Telegrammformats und
Kommunikationsprotokolls**

Änderungsübersicht

Dokumenten- Nummer SMADAT	Ausgabe und Änderungstyp ¹⁾		Bemerkungen	Autor
- SWRNET -11:KD0897	1.0	C	Vorgängerversion	
-11:ZD2897	1.1	B	Erstausgabe, aufbauend auf SWRNET-11:KD	L.K. Müller
-12:ZD0498	1.2	B	Verbesserungen und Ergänzungen	L.K. Müller M. Kurth
-12:ZD0803	1.25	B	Erweiterung um SMA-Net Telegrammrahmen, Berichtigungen und Ergänzungen	W. Rei- chenbä- cher

¹⁾ A: Änderung aufgrund fehlerhafter Unterlagen oder Verbesserung der Unterlagen

B: Änderung, die eine volle oder Vorwärts- Austauschbarkeit sicherstellt

C: Änderungen, die die Austauschbarkeit einschränken oder ausschließen

	Name	Datum	Unterschrift
Geprüft	Bröring	03.03.2003	

Erklärungen zu den verwendeten Symbolen

Um Ihnen einen optimalen Gebrauch dieses Handbuchs und einen sicheren Baugruppeneinsatz in den Phasen der Inbetriebnahme, des Betriebs und der Wartung zu gewährleisten, beachten Sie bitte die folgenden Erklärungen zu den verwendeten Symbolen.



Unter dem Symbol Hinweis wird ein Sachverhalt aufgeführt, dessen Nichtbeachtung zu einem Verlust an Komfort oder zur Beeinträchtigung der Funktion führen kann.

Beispiel: „Fertige C-Routinen hierzu finden Sie auf der Support-Diskette.“



Unter dem Symbol Achtung wird ein Sachverhalt aufgeführt, dessen Nichtbeachtung zur Beschädigung von Bauteilen oder zur Gefährdung von Personen führen kann.

Beispiel: „Vor Öffnen des Gerätes auf jeden Fall Netzstecker ziehen!“



Dieses Symbol kennzeichnet ein Beispiel.

Inhaltsverzeichnis

1	Einleitung.....	5
2	Telegrammrahmen	6
2.1	Sunny-Net Telegrammrahmen.....	6
2.2	SMA-Net Telegrammrahmen	9
2.2.1	PPP Rahmenformat.....	10
2.2.2	Das SMA-Net Rahmenformat.....	13
3	Übertragungsmedien und Zeitverhalten	20
3.1	RS485-Übertragung	20
3.1.1	Carrier Sense (CS)	20
3.1.2	Multiple Acces (MA).....	21
3.1.3	Collision Detection (CD)	21
3.2	Powerline-Übertragung	22
4	SMA-Data Übertragungsprotokoll.....	23
4.1	Telegrammformat.....	24
4.2	Kommunikationsverlauf.....	28
4.2.1	Datenanforderung vom Zielgerät (Einzeltelegramm).....	28
4.2.2	Datenanforderung vom Zielgerät (Mehrfachtelegramm).....	28
4.3	Kommandos.....	32
4.3.1	Kommandos zur Anlagenkonfiguration.....	33
4.3.2	Kommandos zur Datenerfassung	44
4.3.3	Kommandos zur Konfiguration der Datenspeicherung	56
4.3.4	Kommandos zur Binärübertragung.....	60
4.3.5	Kommandos zu Variablen	68
4.3.6	Sonstige Kommandos	71
5	Literatur	73

1 Einleitung

Die Telegramme der SMA-Data Kommunikation bestehen aus zwei Teilen, dem Telegrammrahmen und dem eigentlichen Telegramminhalt (SMA-Data).

2 Telegrammrahmen

Es werden zwei verschiedene Telegrammrahmen eingesetzt:

Sunny-Net kann als Inhalt nur SMA-Data Telegramme übertragen und wird bei Neuentwicklungen nicht mehr eingesetzt.

SMA-Net kann aufgrund seiner Multiprotokoll-Fähigkeit neben SMA-Data Telegrammen auch viele andere bis hin zu TCP/IP übertragen und wird daher bei Neuentwicklungen bevorzugt eingesetzt.

Nicht jede Funktionalität wird von allen Geräten genutzt bzw. unterstützt. Daher ist nicht jedes, der in diesem Dokument spezifizierten Kommandos, zwangsläufig implementiert.

2.1 Sunny-Net Telegrammrahmen

Der Sunny-Net Telegrammrahmen ist nur zum Einsatz mit dem SMA-Data Protokoll gedacht.

- Der **Telegrammrahmen** beinhaltet ein Startzeichen, 2-mal die Nutzdatenlänge, eine 16 Bit Checksumme und ein Stoppzeichen.
- Der **Telegramminhalt** setzt sich aus 7 Byte Protokollkopf (Adressierung, Kommando) und maximal 255 Byte für die eigentlichen Nutzdaten zusammen.

Optional wird jedem Telegramm eine Synchronisationssequenz, bestehend aus zwei aufeinander folgenden AAH Bytes, vorausgeschickt. Diese Sequenz gehört nicht zum Telegrammumfang und dient nur zum „Trainieren“ des FSK (Frequency Shift Keying-Demodulators).

Sunny-Net Telegramm									
Powerline Synchronisation (optional)		Rahmen				Inhalt SMA-Data Telegramm		Rahmen	
Sync	Sync	Start	Länge	Länge	Start	Protokollkopf	Nutzdaten	16 Bit CS	Stopp
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	7 Byte	0..255 Byte	2 Byte	1 Byte
AA	AA	68			68				16

Abk.	Name	Beschreibung	Wert
Start	Startbyte	Telegrammstartzeichen	68H
Länge(2x)	Längenbyte	Anzahl der Nutzdatenbytes	0-255
Start	Startbyte	Telegrammstartzeichen	68H
<i>Inhalt</i>	<i>Telegramminhalt</i>	SMA-Data Telegramm bestehend aus: 7 Byte Protokollkopf 0 bis max. 255 Byte Nutzdaten	
CS	Checksumme	16 Bit Prüfsumme Die Prüfsummenbildung erfolgt durch eine byteweise Addition des kompletten Telegramminhaltes (Protokollkopf und Nutzdaten)	
Stopp	Stoppbyte	Telegrammabschlusszeichen	16H

Startbyte

Um eine Synchronisation auf den Telegrammstart zu ermöglichen, beginnen alle Telegramme mit dem Startzeichen (68H). Dieses Zeichen erscheint jeweils als erstes und als viertes Zeichen in einer Telegrammsequenz.

Längenbyte

Das Längenbyte gibt Auskunft über die Länge der im Telegramm enthaltenen Nutzdaten ohne Protokollkopf. Diese Information erscheint doppelt als zweites und drittes Zeichen in einem Telegramm.

Telegramminhalt

Als Telegramminhalt sind nur SMA-Data Telegramme zulässig. Diese setzen sich aus einem Protokollkopf und Nutzdaten zusammen:

Protokollkopf

Der Protokollkopf beinhaltet die Adressierung und das Kommando.

Nutzdaten

Der Nutzdateninhalt eines Telegramms hängt von dem jeweiligen Kommando ab. Es können maximal 255 Zeichen pro Telegramm übertragen werden.

Prüfsumme

Die Prüfsumme ist ein 16 Bit Wert. Die Erzeugung erfolgt durch die vorzeichenlose byteweise Addition des Telegramminhaltes (Protokollkopf und Nutzdaten).

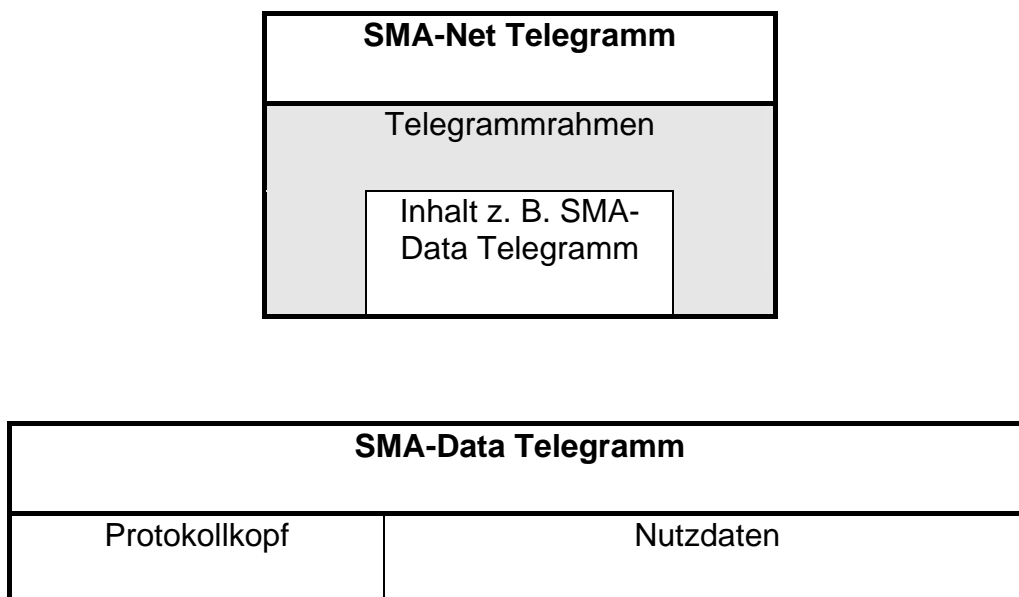
Stoppzeichen

Um ein eindeutiges Telegrammende zu erkennen, werden alle Telegramme mit einem Stoppzeichen (16H) abgeschlossen.

2.2 SMA-Net Telegrammrahmen

In diesem Kapitel werden ein Rahmenformat (Frameformat), ein Zugriffsverfahren und Protokoll-Konventionen dargestellt, die für den Einsatz auf seriellen Verbindungen einschließlich Mehrpunktverbindungen wie etwa RS485 geeignet sind. Das Frameformat orientiert sich an dem Punkt-zu-Punkt Protokoll (PPP) der TCP/IP Protokoll-Familie. PPP selbst ist in RFC (Request for Comments) 1661 [1] dargestellt, ein zugehöriges Frameformat in RFC1662 [2]. Dieses Frameformat orientiert sich seinerseits an dem internationalen HDLC-Standard (ISO 3309, [3]), der in vielen paketorientierten Netzen (X.25, Datex-P) eingesetzt wird.

Ferner beschreibt dieses Kapitel die Einbettung von SMA-Data Telegrammen in das SMA-Net Rahmenformat.



SMA-Net kann Dank seiner Multiprotokoll-Fähigkeit, neben SMA-Data-Telegrammen, auch viele andere bis hin zu TCP/IP übertragen.

Aus der Einbettung von SMA-Data Telegrammen in das SMA-Net Rahmenformat ergeben sich eine Reihe von Vorteilen gegenüber dem klassischen Sunny-Net Rahmenformat.

- Kompatibilität zum Internet- und zum HDLC-Standard
- Option der Nutzung von TCP/IP Protokollen ohne Änderung des Frameformats
- Integrierte Multi-Protokoll-Fähigkeit
- Sichere Datenübertragung durch 16-Bit-CRC
- Einfache Paketerkennung durch eindeutiges Start-Flag
- Minimaler Overhead von nur 8 Bytes bei bis zu 1500 Bytes Nutzdaten
- Transparente Übertragung erlaubt XON/XOFF Software-Handshake

2.2.1 PPP Rahmenformat

Das Rahmenformat für PPP (vgl. RFC 1662, [2]) hat folgenden Aufbau:

PPP Telegramm						
Rahmen			Inhalt		Rahmen	
Start	Adresse	Control	Protokollkopf	Nutzdaten	FCS	Stopp
1 Byte	1 Byte	1 Byte	2 Byte	n Byte	2 Byte	1 Byte
7E	FF	03				7E

Abk.	Name	Beschreibung	Wert
Start	Startbyte	Startflag	0x7E
Adresse	Adressbyte		0xFF
Control	Controlfield		0x03
<i>Inhalt</i>	<i>Telegramm-inhalt</i>	Telegramm bestehend aus: 2 Byte Protokollkopf n Byte Nutzdaten	s.u.
FCS	Frame-Check-Sequence	16 Bit Prüfsumme Die Frame-Check-Sequence ist ein 16-Bit-CRC gemäß dem Polynom $X^{16} + X^{12} + X^5 + 1$	
Stopp	Stoppbyte	Stoppflag	0x7E

Dieses Rahmenformat orientiert sich in folgendem Sinne am HDLC-Format:

- Das Startflag ist unter HDLC identisch
- Als Adresse benutzt HDLC die Werte 3, 5, 7 und 15
- Das Controlfield ist ein bitcodiertes HDLC Steuerfeld: 0x03 steht für unnummerierte Datenböcke.
- Das FCS-Feld (Frame Check Sequence) und seine Berechnung sind unter HDLC identisch.
- Das Stoppflag ist unter HDLC identisch.

Protokoll

Aus RFC1661 [1]: Die Struktur dieses Feldes ist zur ISO 3309 „Extension Mechanism for Address Fields“ konsistent:

- Alle Protokolle müssen ungerade sein. Das niederwertigste Bit des niederwertigsten Oktets muss „1“ sein. Auch müssen alle Protokolle so festgelegt sein, dass das niederwertigste Bit des höchstwertigen Oktets „0“ ist.
- Rahmen, die die obigen Regeln nicht entsprechen, müssen als unbekannte Protokolle behandelt werden.
- Protokollfeldwerte, im Bereich "0xxx" bis "3xxx" gehören zum Netzwerk-Layer-Protokoll bestimmter Pakete und Werte im Bereich "8xxx" bis "3xxx" gehören zum zugehörigen Netzwerk-Control-Protokoll (NCPs).
- Protokollfeldwerte im Bereich "4xxx" bis "7xxx" werden für Protokolle mit geringen Datenmengen, ohne zugehöriges NCP, verwendet.
- Protokollfeldwerte im Bereich "Cxxx" bis "Fxxx" werden benutzt, um Link-Layer-Control-Protokolle zu verwenden.

Protokoll-Einträge

Von dem 2 Byte langen Protokoll-Feld wird das MSB (Most Significant Byte) zuerst übertragen, anschließend das LSB (Least significant Byte).

Aktuelle Werte für das Protokoll-Feld sind jeweils in der neuesten RFC-Ausgabe ([4]) mit dem Titel "Assigned Numbers" enthalten.

2.2.2 Das SMA-Net Rahmenformat

SMA-Net Telegramm						
Rahmen			Inhalt		Rahmen	
Start	Adresse	Control	Protokollkopf	SMA-Data	FCS	Stopp
1 Byte	1 Byte	1 Byte	2 Byte	255 Byte	2 Byte	1 Byte
7E	FF	03				7E

Abk.	Name	Beschreibung	Wert
Start	Startbyte	Startflag / Beginn eines neuen Telegramms	0x7E
Adresse	Adressbyte	Broadcast – Adresse des SMA-Net Rahmens	0xFF
Control	Controlfield		0x03
<i>Inhalt</i>	<i>Telegramm-inhalt</i>	Telegramm bestehend aus: 2 Byte Protokollkopf 255 Byte Nutzdaten (SMA-Data) ($7 \leq n \leq 262$)	s.u.
FCS	Frame-Check-Sequence	16 Bit Prüfsumme Die Frame-Check-Sequence ist ein 16-Bit-CRC gemäß dem Polynom $X^{16} + X^{12} + X^5 + 1$	
Stopp	Stoppbyte	Stoppflag	0x7E

Start- und Stoppflag

Jeder Datenrahmen beginnt und endet mit dem Zeichen 0x7E. Alle Teilnehmer im SMA-Net überwachen den Datenstrom kontinuierlich auf dieses Zeichen und synchronisieren sich damit eindeutig auf den Beginn eines neuen Telegramms.

Im Falle aufeinander folgender Rahmen ist nur ein Zeichen 0x7E erforderlich. Treten trotzdem zwei Zeichen 0x7E auf, gilt dies als (zulässiges) leeres Paket.

Adresse

Die Adresse 0xFF ist im SMA-Net die Broadcast-Adresse und muss von allen SMA-Net Teilnehmern empfangen werden.

Control

Das Control-Field ist ein Byte mit dem Wert 0x03. Es ist nach dem HDLC-Standard ein „Unnumbered Information (UI) command“, in dem das Poll/Final (P/F) Flag auf 0 gesetzt ist.

Protokoll

Von dem 2 Byte langen Protokoll-Feld wird das MSB (Most Significant Byte) zuerst übertragen, anschließend das LSB (Least Significant Byte).

Allgemeine Richtlinien für das Protokoll-Feld, gemäß RFC1661, sind im Kapitel über die PPP-Protokolle aufgeführt. Im Rahmen von SMA-Net sind folgende Protokoll-Kennungen definiert.

- ID 1 4041 SMA-Data Telegramme
- ID 2 4051 TCP / IP Zusatzbaugruppe
- ID 3 4043 Software Update System, SUSy

Frame Check Sequence (FCS)

Die Frame-Check-Sequence ist ein 16-Bit-CRC gemäß dem Polynom

- $X^{16} + x^{12} + x^5 + 1$

Von dem 2 Byte langen FCS-Wert wird das LSB (Least Significant Byte) zuerst übertragen.

Der FCS-Wert wird berechnet über die Felder Adresse, Control, Protokoll und Daten. Das Start- und Stoppflag sowie Escape-Zeichen, die im Rahmen der Transparenz (s. u.) eingefügt wurden, gehen nicht in die Berechnung ein. Dies gilt auch für Escape-Zeichen innerhalb des FCS-Wertes selbst.

Wenn Zeichen empfangen werden, die in der ACCM (s. u.) als Steuerzeichen ausgewiesen sind, werden diese vor der FCS-Berechnung aus dem Datenstrom entfernt.

Escape-Zeichen und ACCM

Zur Erreichung einer transparenten Übertragung, in deren Datenstrom das Start-/Stoppflag nirgends vorkommt, wird das Escape-Zeichen 0x7D verwendet. Für den Einsatz des Escape-Zeichens gelten folgende Regeln:

- 0x7E wird als 0x7D / 0x5E übertragen
- 0x7D wird als 0x7D / 0x5D übertragen
- Alle Zeichen mit einem gesetzten Bit in der ACCM, werden als 0x7D / 0xNN übertragen, wobei 0xNN das exclusive Oder aus dem ursprünglichen Wert und 0x20 ist.

Die Async-Control-Character-Map (ACCM)

Die ACCM ist eine 32-Bit-Zahl, deren Bits mit den ASCII-Steuerzeichen 0x00 (Bit 0) bis 0x1F (Bit 31) korrespondieren.

- Der Defaultwert der ACCM im SMA-Net ist 0x000E0000

In diesem Defaultwert sind die Bits 17 (0x11, XON), 18 (0x12, DC2) und 19 (0x13, XOFF) gesetzt. Die im XON/XOFF Verfahren benutzten Steuerzeichen 0x11 und 0x13 werden auf diesem Wege als 0x7D / 0x31 und 0x7D / 0x33 übertragen.

Der Defaultwert ist so ausgelegt, dass die Felder Adresse und Control im SMA-Net stets ohne Escape-Zeichen übertragen werden, um die Rahmen nicht unnötig zu verlängern.

SMA-Net Sender

Der Sender stellt das zu übertragende Telegramm zunächst vollständig zusammen (einschließlich FCS) und ersetzt dann alle Zeichen 0xNN zwischen dem Start- und dem Stoppflag durch das Paar 0x7D / (0xNN ^ 0x20), die entweder dem Flag-Zeichen oder dem Escape-Zeichen entsprechen oder die ein ASCII-Steuerzeichen mit gesetztem ACCM-Bit sind.

SMA-Net Empfänger

Der Empfänger ersetzt alle eintreffenden Zeichen mit vorausgehendem Escape-Zeichen (0x7D) durch ihr mit 0x20 xor-tes Äquivalent. In dem Sonderfall 0x7D / 0x7E bricht er den Empfang ab und interpretiert 0x7E als den Beginn eines neuen Telegramms.

Ferner testet der Empfänger für alle Zeichen, die kleiner als 0x20 sind, das zugehörige Bit in der ACCM. Wenn es gesetzt ist, wird das Zeichen aus dem Datenstrom entfernt. Damit können z. B. Telegramme vom Sender durch angehängte Nullen auf eine Mindestlänge aufgefüllt werden.

Für den auf diesem Wege modifizierten Datenstrom wird anschließend der FCS Wert berechnet und ausgewertet.

Fast Frame Check Sequence (FCS)-Implementation

Die Frame-Check-Sequence wurde ursprünglich als Hardware-Implementation erdacht. Während der serielle Bit-Strom über die Leitung herausgeht, wird FCS berechnet und als Komplement an den seriellen Strom angehängt. Abschließend folgt das Stopplflag.

Der Empfänger hat keine Möglichkeit festzustellen, dass die FCS des empfangenen Datenstroms abgeschlossen ist, bevor er das Stopplflag erkennt. Aus diesem Grund wurde die FCS so definiert, dass ein spezielles Bit-Muster entsteht, wenn die FCS-Berechnung über die empfangene komplementäre FCS läuft. Ein gültiger Rahmen wird durch einen „guten“ FCS-Wert angezeigt.

FCS-Tabellengenerator

Der folgende Quelltext erzeugt eine Tabelle zur Berechnung des FCS:

```
/*
 * Generate a FCS-16 table.
 *
 * Drew D. Perkins at Carnegie Mellon University.
 *
 * Code liberally borrowed from Mohsen Banan and D. Hugh Redelmeier.
 */

/*
 * The FCS-16 generator polynomial:  $x^{16} + x^{12} + x^5 + 1$ .
 */
#define P          0x8408

main()
{
    register unsigned int b, v;
    register int i;

    printf("typedef unsigned short ul6;\n");
    printf("static ul6 fcstab[256] = {");
    for (b = 0; ; )
    {
        if (b % 8 == 0)
            printf("\n");

        v = b;
```

```

    for (i = 8; i--; )
        v = v & 1 ? (v >> 1) ^ P : v >> 1;

    printf("\t0x%04x", v & 0xFFFF);
    if (++b == 256)
        break;
    printf(",");
}
printf("\n};\n");
}

```

FCS-Berechnungsmethode

Der folgende Quelltext stellt einen schnelleren Algorithmus zur Berechnung der FCS unter Zuhilfenahme einer Nachschlagtabelle bereit:

```

/*
 * u16 represents an unsigned 16-bit number. Adjust the typedef for
 * your hardware.
 */
typedef unsigned short u16;

/*
 * FCS lookup table as calculated by the table generator.
 */
static u16 fcstab[256] = {
    0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,
    0x8c48, 0x9dc1, 0xaf5a, 0xbed3, 0xca6c, 0xdbe5, 0xe97e, 0xf8f7,
    0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,
    0x9cc9, 0x8d40, 0xbfdb, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,
    0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,
    0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,
    0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,
    0xbdcb, 0xac42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,
    0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,
    0xce4c, 0xdfc5, 0xed5e, 0xfcd7, 0x8868, 0x99e1, 0xab7a, 0xbaf3,
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,
    0xdec d, 0xcf44, 0xfddf, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,
    0xef4e, 0xfec7, 0xcc5c, 0xdd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,
    0xffcf, 0xee46, 0xdcdd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,
    0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,
    0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,
    0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,
    0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,
    0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,
    0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,
    0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,
    0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,
    0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
    0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
    0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
    0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
    0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
    0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
    0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
    0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78
}

```

```
};

#define PPPINITFCS16    0xffff /* Initial FCS value */
#define PPPGOODFCS16   0xf0b8 /* Good final FCS value */

/*
 * Calculate a new fcs given the current fcs and the new data.
 */
u16 pppfcs16(fcs, cp, len)
    register u16 fcs;
    register unsigned char *cp;
    register int len;
{
    ASSERT(sizeof (u16) == 2);
    ASSERT(((u16) -1) > 0);
    while (len--)
        fcs = (fcs >> 8) ^ fcstab[(fcs ^ *cp++) & 0xff];

    return (fcs);
}

/*
 * How to use the fcs
 */
tryfcs16(cp, len)
    register unsigned char *cp;
    register int len;
{
    u16 trialfcs;

    /* add on output */
    trialfcs = pppfcs16( PPPINITFCS16, cp, len );
    trialfcs ^= 0xffff; /* complement */
    cp[len] = (trialfcs & 0x00ff); /* least significant byte first */
    cp[len+1] = ((trialfcs >> 8) & 0x00ff);
    /* check on input */
    trialfcs = pppfcs16( PPPINITFCS16, cp, len + 2 );
    if ( trialfcs == PPPGOODFCS16 )
        printf("Good FCS\n");
}
}
```

3 Übertragungsmedien und Zeitverhalten

3.1 RS485-Übertragung

Die Datenübertragung im SMA-Net benötigt nur die Sende- und Empfangsleitungen. Ein Handshake über weitere Signalleitungen wie etwa „Request To Send“ (RTS), „Clear To Send“ (CTS), „Data Carrier Detect“ (DCD) oder „Data Terminal Ready“ (DTR) ist nicht vorgesehen.

Das SMA-Net Rahmenformat ist für serielle Mehrpunkt-Verbindungen nach RS485 ausgelegt, wobei alle Teilnehmer als gleichberechtigt angesehen werden. Es benötigt deshalb zusätzliche Regeln zum Zugriff auf das Medium (Media Access Control). Diese Regeln orientieren sich am CSMA/CD Verfahren des Ethernets.

3.1.1 Carrier Sense (CS)

Ein Teilnehmer im SMA-Net muss vor jedem Sendeversuch eines Anfrage oder Mitteilungs-Telegramms (Bit 6 in Ctrl ist 0) die RxD-Leitung 30 ms lang ohne Aktivität (d. h. mit High-Pegel) gesehen haben. Zwischen dem Erhalt einer Antwort und dem Versenden der nächsten Anfrage müssen mindestens 50 ms liegen.

Ein Antwort-Telegramm (Bit 6 in Ctrl ist 1) soll frühestens nach 25 (der Anfrager muss seinen Sender abschalten) und spätestens nach 30 ms gesendet werden. Dadurch hat die Antwort auf eine Anfrage stets Vorrang vor weiteren Anfragen (da die notwendige Ruhephase von 30 ms nicht erreicht wird). Frage-Antwort-Zyklen können also kollisionsfrei abgeschlossen werden, sobald der Anfrager Sendeberechtigung erlangt hat.



Das Verfahren ist bewusst so gewählt, dass jeder Polling-Teilnehmer die RxD-Leitung immer im starren 5 ms Raster prüft, d. h. MSR ausliest und 5 ms später auf Aktivität prüft. Wenn also mehrere Teilnehmer senden

wollen, werden sie in der Regel zeitlich gegeneinander versetzte Polling-Zeitpunkte haben, so dass mit großer Wahrscheinlichkeit einer der Teilnehmer die Leitung so früh als frei erkennt, dass die anderen seine Sendeaktivität bereits erkennen und somit weiter im Polling-Status bleiben. Der Gewinner dieser Runde hat anschließend gegenüber den noch wartenden Teilnehmern den Nachteil, dass er nun 40 ms aussetzen muss.



Das CS-Polling ist nur effektiv wirksam, wenn während der Übertragung eines Telegramms keine Lücke von mehr als 30 ms zwischen zwei Zeichen entsteht. Darauf haben die Teilnehmer beim Senden zu achten.

3.1.2 Multiple Acces (MA)

Alle Stationen, die sich nach dem zuvor beschriebenen CS-Polling davon überzeugt haben, dass die Leitung frei ist, dürfen anfangen zu senden.

3.1.3 Collision Detection (CD)

Jeder Teilnehmer muss die von ihm gesendeten Zeichen selbst empfangen und im Falle eines Fehlers (Differenz zwischen TX- und RX-Daten) das Senden sofort einstellen. Das Verfahren soll dabei so organisiert sein, dass einerseits Kollisionen möglichst schnell (unmittelbar nach jedem TX-Zeichen) erkannt werden und andererseits der Betrieb von seriellen Bausteinen mit FIFO unterstützt wird.

Im Falle einer Kollision erfolgt ein CS-Polling mit wahlweise 0 ms oder 5 ms Pausenzeit nach n Kollisionen ($n = 1, 2, ..$) in Folge.

- Wenn Bit $n-1$ der eigenen Teilnehmernummer 0 ist, dann wählen Sie 0 ms für das CS-Polling, sonst 5 ms.
- Nach 16 Kollisionen in Folge gilt die Übertragung als endgültig gescheitert.



Die Sende-Konkurrenz erfolgt hier nur zwischen den Kollisions-Partnern, da alle Ruhepausen auf der Leitung unter 30 ms liegen.

3.2 Powerline-Übertragung

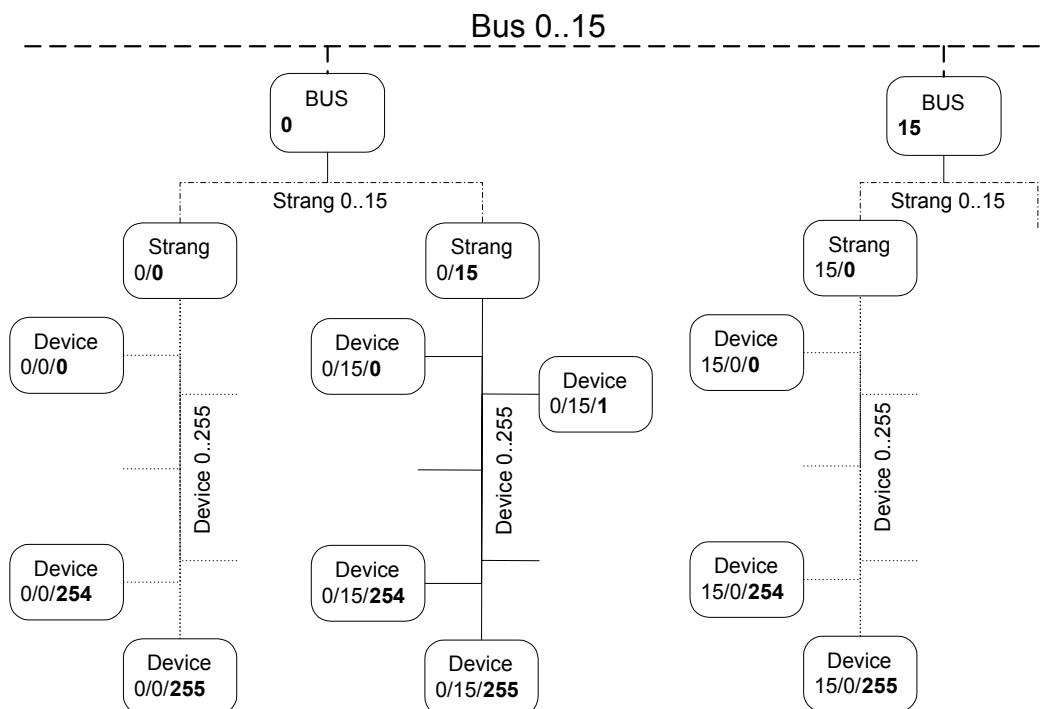
Die Powerline-Übertragung ist im EN 50065-1 „Signalübertragung auf elektrischen Niederspannungsnetzen im Frequenzbereich 3 kHz bis 148,5 kHz“ mit den folgenden Randbedingungen definiert:

- Es darf nicht länger als 1 Sekunde dauernd gesendet werden.
- Nach jeder Übertragung ist eine Zwangspause von 125 ms einzuhalten.
- Innerhalb einer Sendung dürfen keine Pausen von mehr als 80 ms ohne Signalübertragung enthalten sein.
- Jeder Teilnehmer muss Belegungen des Sendebandes erkennen. Das Frequenzband ist belegt, wenn ein Signal von mindestens 4 ms Dauer vorhanden ist.
- Jeder Teilnehmer darf senden, wenn für eine Zeitspanne von 85 ms bis 125 ms mit mindestens sieben möglichen Abfragezeitpunkten das Band als nicht belegt detektiert wurde.

4 SMA-Data Übertragungsprotokoll

Das nachfolgend beschriebene Übertragungsprotokoll ist darauf ausgelegt, eine größtmögliche Flexibilität und Übertragungssicherheit zu gewährleisten.

Das Protokoll basiert auf folgender Netzwerktopologie:



Jeder Netzwerkteilnehmer besitzt eine eindeutige Netzwerkadresse, die sich aus einer Busadresse, einer Strangadresse und einer Geräteadresse zusammensetzt. Über diese Adressierungsart können pro logischem Strang 256 Geräte adressiert werden. Bei maximal 16 Strängen ergibt sich somit ein Adressbereich von bis zu 4096 (16 x 256) Geräten. Die Busadresse dient zur Trennung von benachbarten Anlagen (maximal 15).

Eine weitere Adressierungsart stellt die Gruppenadressierung dar. Es können bis zu $2^{16} = 65536$ Gruppen definiert werden. Dabei kann jeder Netzwerkteilnehmer, unabhängig von Strang und Bus, einer oder mehreren beliebigen Gruppe/n zugeteilt werden. Um die Möglichkeit von netzglobalen Broadcastmeldungen zu schaffen, gehören alle Busteilnehmer automatisch der Gruppe 0 an.

Der Datenaustausch zwischen den einzelnen Netzteilnehmern erfolgt über einzelne Datentelegramme, die sowohl Telegrammquell- als auch Zieladresse enthalten. Diese Art der Adressierung erlaubt es, dass im Prinzip jeder Netzwerkteilnehmer mit jedem anderen Netzwerkteilnehmer kommunizieren kann. Es gibt also keine strenge Master- / Slave-Kommunikation.



Die Daten werden „Little Endian, low-Byte first“ übertragen (Intel-Format).

4.1 Telegrammformat

Der Telegramminhalt eines SMA-Data Telegramms besteht aus zwei Teilen, dem Protokollkopf und den eigentlichen Nutzdaten.

- Der **Protokollkopf** beinhaltet die Telegrammquell- und Zieladresse, ein Control-Byte zum Paket-Handling, einen Paketzähler und das jeweilige Kommando.
- Die **Nutzdaten** setzen sich aus den zum jeweiligen Kommando gehörenden Daten zusammen.

SMA-Data Telegramminhalt					
Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Data
2 Byte	2 Byte	1 Byte	1 Byte	1 Byte	max. 255 Byte

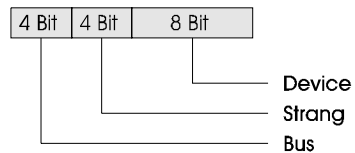
Abk.	Name	Beschreibung	Wert
Quelle	Quelladresse	Telegrammabsender <i>Netzwerkadresse</i>	
Ziel	Zieladresse	Telegrammempfänger <i>Netzwerkadresse oder Gruppenadresse</i>	
Ctrl	Control-Byte	Bit 7 Adressierungsart (0=Netzwerkadresse / 1=Gruppenadresse) Bit 6 Acknowledge (0=Anfrage / 1=Antwort) Bit 4 Gatewaysperre (0=nicht gesetzt / 1=gesetzt) Bit 0-3,5 Reserve (0)	
PktCnt	Paketzähler	Anzahl der Telegrammfolgekupete	0-255H
Cmd	Kommando	Dies ist der Befehl des Telegrammabsenders an die Ziel- adresse	0-255H
Data	Nutzdaten	Der Nutzdateninhalt eines Telegramms kann in Abhängig- keit vom Kommando zwischen 0 und 255 Byte liegen.	

Telegrammadressen (Quelle / Ziel)

Die Quelladresse eines Telegramms ist immer die Netzwerkadresse des Telegrammabsenders. Die Zieladresse eines Telegramms kann entweder die Netzwerkadresse eines einzigen Empfängers oder die Gruppenadresse mehrerer Empfänger sein. Die Festlegung der Zieladressierungsart erfolgt über das Bit 7 des Ctrl-Bytes eines Telegramms.

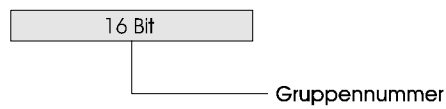
Netzwerkadresse (als Quell- oder Zieladresse)

Die Netzwerkadresse setzt sich aus der Bus- (4 Bit), Strang- (4 Bit) und Geräteadresse (8 bit) zusammen:



Gruppenadresse (nur als Zieladresse)

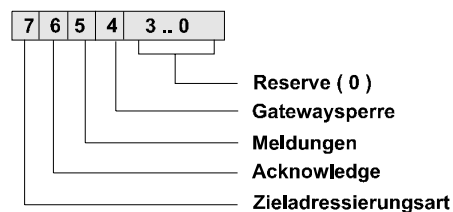
Eine Gruppenadresse besteht aus einer 16-Bit Zahl.



Jedes Gerät gehört automatisch zur Gruppe 0.

Control-Byte

Das Control-Byte hat folgenden Aufbau:



- Bit 7: Dieses Bit entscheidet über die Art der Zieladresse. Ist es gesetzt, so ist die Zieladresse eine Gruppenadresse, anderenfalls ist die Zieladresse eine Netzwerkadresse.
0 = Netzwerkadresse
1 = Gruppenadresse
- Bit 6: 0 = Anfrage Telegramm
1 = Antwort Telegramm
- Bit 4: Dieses Bit entscheidet darüber, ob ein Datenlogger den Strang sperrt oder transparent arbeitet.
0 = Gatewaysperre nicht gesetzt (Telegrammweitergabe)
1 = Gatewaysperre gesetzt (Telegrammweitergabe gesperrt)
- Bit 0..3: Reserve (=0)

Paketzähler

Dieses Byte enthält einen Wert zwischen Null und 255. Ist die Nutzdatenmenge größer als 255 Bytes, sind mehrere Pakete zur Übertragung notwendig. Die Zählung der Pakete beginnt bei einem Wert $n > 0$. Der Paketzähler wird pro Paket dekrementiert. Das letzte Paket enthält den Wert Null. Es kann mit oder ohne Nutzdaten gesendet werden. Falls mehr als 255 Pakete notwendig sind, wird bis eins dekrementiert und dann wieder bei 255 fortgesetzt.

Kommando-Byte

Das Kommando-Byte enthält den eigentlichen Befehl an einen oder mehrere Netzwerkteilnehmer.

Nutzdaten

Der Nutzdateninhalt eines Telegramms hängt von dem jeweiligen Kommando ab. Es können maximal 255 Zeichen pro Telegramm übertragen werden. Je nach Kommando kann der Nutzdateninhalt aber auch die Länge Null haben.

4.2 Kommunikationsverlauf

Es gelten folgende Regeln:

- Es sendet jeweils nur ein Netzwerkteilnehmer, alle anderen Teilnehmer verhalten sich passiv und hören mit.
- Grundsätzlich wird jedes per Netzwerkadresse übertragene Telegramm mit gültigem Nutzdateninhalt vom Empfänger bestätigt. Diese Bestätigung erfolgt unabhängig vom Telegramminhalt mit gesetztem Acknowledge Bit im Control-Byte des Telegramms.

4.2.1 Datenanforderung vom Zielgerät (Einzeltelegramm)



Anfrage vom Quellgerät:	Acknowledge Bit	= 0
	Nutzdaten	= laut Kommando

Antwort vom Zielgerät:	Acknowledge Bit	= 1
	Paketzähler	= 0
	Nutzdaten	= laut Kommando

4.2.2 Datenanforderung vom Zielgerät (Mehrfachtelegramm)



Beispiel 1:

Es sollen 600 Bytes Nutzdaten vom Zielgerät an das Quellgerät übertragen werden. Da pro Telegramm max. 255 Zeichen übertragen werden können, muss die Übertragung in 3 Paketen erfolgen:

Paket Nr. 1:

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Paketzähler	= 0
Antwort vom Zielgerät	Acknowledge Bit	= 1
	Paketzähler	= 255
	Nutzdaten	= 255 Byte

Paket Nr. 2:

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Paketzähler	= 255
Antwort vom Zielgerät	Acknowledge Bit	= 1
	Paketzähler	= 254
	Nutzdaten	= 255 Byte

Paket Nr. 3:

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Paketzähler	= 254
Antwort vom Zielgerät	Acknowledge Bit	= 1
	Paketzähler	= 0
	Nutzdaten	= 90 Byte

**Beispiel 2:**

Dieser Mechanismus erlaubt es auch, vom Quellgerät fehlerhafte Pakete nachzufordern:

Paket Nr. 1:

Anfrage vom Quellgerät	Acknowledge Bit	= 0
------------------------	-----------------	-----

Antwort vom Zielgerät	Acknowledge Bit	= 1
	Paketzähler	= 255
	Nutzdaten	= 255 Byte

Paket Nr. 2:

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Paketzähler	= 255

Fehlerhaftes Paket erkannt oder Timeout !!!

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Paketzähler	= 255

Antwort vom Zielgerät	Acknowledge Bit	= 1
	Paketzähler	= 254
	Nutzdaten	= 255 Byte

Paket Nr. 3:

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Paketzähler	= 254

Antwort vom Zielgerät	Acknowledge Bit	= 1
	Paketzähler	= 0
	Nutzdaten	= 90 Byte

**Beispiel 3:**

Datenanforderung von mehreren Zielgeräten per Gruppenadresse.

Mit Gruppenadressen können mehrere (Multicast) oder alle (Broadcast) Teilnehmer erreicht werden.

Zur Vermeidung von Buskollisionen erfolgt die Rückmeldung auf Gruppenanforderungen jeweils nach einer Übertragungspause von 85 + Zu-

fallswert (0-4765) ms.

Anfrage vom Quellgerät	Acknowledge Bit	= 0
	Nutzdaten	= laut Kommando

Übertragungspause

Antwort vom Gruppenmitglied 1	Acknowledge Bit	= 1
	Nutzdaten	= laut Kommando

Übertragungspause

Antwort vom Gruppenmitglied 2	Acknowledge Bit	= 1
	Nutzdaten	= laut Kommando

Übertragungspause

Antwort vom Gruppenmitglied 3	Acknowledge Bit	= 1
	Nutzdaten	= laut Kommando

Übertragungspause

4.3 Kommandos

Die SMA-Data Kommandos lassen sich in solche zur Erfassung / Konfiguration der Anlage, zur Datenerfassung, zur Binärübertragung und zur Übertragung von Variablen zusammenfassen.

Ein SMA-Data Teilnehmer unterstützt nicht immer alle Kommandos, sondern jeweils nur die für den entsprechenden Gerätetyp zutreffenden Kommandos.

Übersicht der SMA-Data Kommandos:

Befehl:	Cmd:	Beschreibung:	Kapitel
Kommandos zur Anlagenkonfiguration:			
CMD_GET_NET_START	06	Beginn Netz-Konfigurationsanforderung	4.3.1
CMD_GET_NET	01	Netz-Konfiguration anfordern	4.3.2
CMD_SEARCH_DEV	02	Geräte suchen	4.3.1.3
CMD_CFG_NETADR	03	Netzadresse vergeben	4.3.1.4
CMD_GET_CINFO	09	Kanalinformation anfordern	4.3.1.5
CMD_SET_GRPADR	04	Gruppenadresse vergeben (reserviert)	
CMD_DEL_GRPADR	05	Gruppenadresse löschen (reserviert)	
CMD_SET_MPARA	15	Datenerfassung parametrieren (reserviert)	
CMD_TNR_VERIFY	50	Verifizieren der Teilnehmernummer (reserviert)	
Kommandos zur Datenerfassung:			
CMD_SYN_ONLINE	10	Online-Daten synchronisieren	4.3.2.1
CMD_GET_DATA	11	Datenanforderung	4.3.2.2
CMD_SET_DATA	12	Daten senden	4.3.2.3
CMD_GET_SINFO	13	Datenspeicher-Info abfragen	4.3.2.4
Kommandos zur Konfiguration der Datenspeicherung:			
CMD_GET_MTIME	20	Speicherungsintervalle lesen	4.3.3.1
CMD_SET_MTIME	21	Speicherungsintervalle setzen	4.3.3.2
Kommandos zur Binärübertragung:			
CMD_GET_BINFO	30	Binärbereich-Info anfordern	4.3.4.1
CMD_GET_BIN	31	Binärdaten anforderung	4.3.4.2
CMD_SET_BIN	32	Binärdaten senden	4.3.4.3

Befehl:	Cmd:	Beschreibung:	Kapitel
Kommandos zu Variablen:			
CMD_VAR_VALUE	51	System Variablen erfragen	4.3.5.1
CMD_VAR_FIND	52	Besitzer einer Variablen (reserviert)	
CMD_VAR_STATUS_OUT	53	Zuordnung Variable – Kanal (reserviert)	
CMD_VAR_DEFINE_OUT	54	Zuordnen Variable – Kanal (reserviert)	
CMD_VAR_STATUS_IN	55	Zuordnung Eingangsgröße – Status Variable (reserviert)	
CMD_VAR_DEFINE_IN	56	Zuordnen Eingangsgröße – Variable (reserviert)	
Sonstige Kommandos:			
CMD_PDELIMIT	40	Begrenzung der Geräteleistung	4.3.6.1
CMD_TEAM_FUNCTION	60	Team-Funktion für Solarwechselrichter	4.3.6.2



Reservierte Kommandos sollten aus historischen Gründen nicht verwendet werden und sind in diesem Dokument nicht beschrieben.

4.3.1 Kommandos zur Anlagenkonfiguration

Diese Kommandos dienen zum Erfassen und Konfigurieren einer Anlage. Jedes passive Gerät innerhalb des Netzwerkes muss diese Kommandos beantworten können und entsprechend reagieren. Aktive Geräte, z. B. ein zentraler PC zum Verwalten einer Anlage, müssen diese Kommandos unterstützen, um eine Anlage ordnungsgemäß erfassen zu können.

4.3.1.1 Beginn Netz-Konfigurationsanforderung (CMD_GET_NET_START)

Dieser Befehl dient zum Start einer SMA-Data Netzwerk-Konfiguration. Alle Geräte, die diesen Befehl empfangen, setzen ihren internen Merker für „schon gemeldet“ zurück und antworten. Der Dateninhalt der Antwortpakete entspricht denen des Befehls CMD_GET_NET. Nach Empfang der (neuen) Netzwerk-Adresse durch

CMD_CFG_NETADR wird der interne Merker der Geräte für „schon gemeldet“ gesetzt.

Der Befehl CMD_GET_NET_START ist nur einmal zum Beginn einer Erfassung abzusetzen. Für alle folgenden Erfassungszyklen muss der Befehl CMD_GET_NET verwendet werden, da sich sonst bereits erfasste Geräte erneut melden.

Anforderung:

Cmd: 06
 Ctrl: 10000000 b
 Dateninhalt: -
 Datenlänge: 0 Byte
 Quelladresse: Eigene Netzwerkadresse
 Zieladresse: Gruppe 0



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	-
01 00	00 00	80	00	06	-

Antwort:

Cmd: 06
 Ctrl: 01000000 b
 Dateninhalt: Seriennummer des Gerätes (dword)
 Gerätetyp z. B. „WR700-07“ (8 char)
 beinhaltet der Typname weniger als 8 Zeichen, so ist die
 Differenz mit ASCII-Nullen aufzufüllen
 Datenlänge: 12 Byte
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Seriennummer, Gerätetyp SN=9380933, Typ= "WR700-07"
02 00	01 00	40	00	06	45 24 8F 00, 57 52 37 30 30 2D 30 37

4.3.1.2 Netz-Konfiguration anfordern (CMD_GET_NET)

Dieser Befehl dient der Erfassung der SMA-Data Netzwerk-Konfiguration. Alle SMA-Data Teilnehmer mit nicht gesetztem Merker für ‚schon gemeldet‘ antworten mit ihrer derzeitigen Netzwerkadresse, Seriennummer und Gerätetypbezeichnung. Zur Vermeidung von Buskollisionen erfolgt die Rückmeldung jeweils nach einer Übertragungspause von 85 + Zufallswert (0-4765) ms. Nach Empfang der (neuen) Netzwerk-Adresse durch CMD_CFG_NETADR wird der interne Merker der Geräte für „schon gemeldet“ gesetzt.

Anforderung:

Cmd:	01
Ctrl:	10000000 b
Dateninhalt:	-
Datenlänge:	0 Byte
Quelladresse:	Eigene Netzwerkadresse
Zieladresse:	Gruppe 0



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	-
01 00	00 00	80	00	01	-

Antwort:

Cmd: 01
Ctrl: 01000000 b
Dateninhalt: Seriennummer des Gerätes (dword)
 Gerätetyp z. B. „WR700-07“ (8 char)
 beinhaltet der Typname weniger als 8 Zeichen, so ist die
 Differenz mit ASCII-Nullen aufzufüllen
Datenlänge: 12 Byte
Quelladresse: Netzwerkadresse des Gerätes
Zieladresse: Eigene Netzwerkadresse



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Seriennummer, Gerätetyp SN=9380933, Typ= "WR700-07"
02 00	01 00	40	00	01	45 24 8F 00, 57 52 37 30 30 2D 30 37



Ablauf des Erfassungszyklus einer Anlage ohne Datenlogger:

Schritt:	Befehl:	Erklärung:
1	CMD_GET_NET_START	Start der SMA-Data Netzwerk-Konfiguration, Geräte setzen den internen Merker für „schon gemeldet“ zurück und antworten.
2	CMD_CFG_NET_ADR	Allen erkannten Geräten wird eine Netzwerkadresse zugeordnet. Diese setzen den internen Merker für „schon gemeldet“.
3	CMD_GET_NET	Fortsetzung der SMA-Data Netzwerk-Konfiguration. Es melden sich nur Geräte, deren interner Merker für „schon gemeldet“ noch nicht gesetzt ist.
4	CMD_CFG_NET_ADR	Allen erkannten Geräten wird eine Netzwerkadresse zugeordnet. Diese setzen den internen Merker für „schon gemeldet“.
5		Schritte 3 und 4 wiederholen, bis alle Geräte erfasst sind.



Ablauf der Erfassung einer Anlage, welche Datenlogger beinhaltet:

Schritt:	Befehl:	Erklärung:
1	CMD_GET_NET_START mit 'Ctrl-Bit' 'Gatewaysperre gesetzt'	Es melden sich: a) Geräte, welche direkt erreichbar sind b) alle Datenlogger
2	CMD_CFG_NET_ADR	Allen erkannten Geräten wird eine Netzwerkadresse zu- geordnet. Diese setzen den internen Merker für „schon gemeldet“. Datenloggern wird mit der Netzadresse ein Strang zuge- wiesen.
3	CMD_GET_NET mit 'Ctrl-Bit' 'Gatewaysperre gesetzt' und Netzadresse des Datenloggers	Der Datenlogger meldet sich für jedes verwaltete Gerät in der Form: „Info verwaltetes Gerät“ . „Info Datenlogger“ z. B.: S/N-SWR, „WR-850-01“, NetAdr-SBC, S/N-SBC, „SunBC-01“
4	CMD_CFG_NET_ADR	Allen vom Datenlogger verwalteten Geräten wird eine Netzwerkadresse zugeordnet (Bus- und Strangadresse entsprechend Datenlogger). Diese setzen den internen Merker für „schon gemeldet“.
5	GET_CINFO mit 'Ctrl-Bit' 'Gatewaysperre gesetzt' und Netzadresse des vom Da- tenlogger verwalteten Gerä- tes	Der PC fragt die Kanalliste der unbekanntenen Geräte ab. Der Datenlogger antwortet mit der Kanalliste für das Gerät

4.3.1.3 Geräte suchen (CMD_SEARCH_DEV)

Dieser Befehl dient zum Ansprechen eines bestimmten Gerätes über seine Serien-
nummer. Das Gerät antwortet mit seiner derzeitigen Netzwerkadresse, Seriennum-
mer und Gerätetypbezeichnung.

Anforderung:

Cmd:	02
Ctrl:	10000000 b
Dateninhalt:	Seriennummer des Gerätes (dword)
Datenlänge:	4 Byte
Quelladresse:	Eigene Netzwerkadresse
Zieladresse:	Gruppe 0

Antwort:

Cmd:	02
Ctrl:	01000000 b
Dateninhalt:	Seriennummer des Gerätes (dword) Gerätetyp z. B. 'WR700-7' (8 char) beinhaltet der Typname weniger als 8 Zeichen, so ist die Differenz mit ASCII-Nullen aufzufüllen
Datenlänge:	12 Byte
Quelladresse:	Netzwerkadresse des Gerätes
Zieladresse:	Eigene Netzwerkadresse

4.3.1.4 Netzadresse vergeben (CMD_CFG_NETADR)

Dieser Befehl dient zum Konfigurieren eines bestimmten Gerätes über seine Seriennummer. Das Gerät antwortet mit seiner neuen Netzwerkadresse und Seriennummer. Nach Erhalt dieses Befehls wird eine Reaktion auf CMD_GET_NET durch das Setzen des internen Merkers für „schon gemeldet“ unterdrückt. Durch einen anderen Broadcast-Befehl wie CMD_SYN_ONLINE oder CMD_GET_NET_START wird der Merker für „schon gemeldet“ zurückgesetzt und somit die Reaktion auf CMD_GET_NET wieder freigegeben. Dieser Befehl wird benötigt, um neue oder ausgetauschte Geräte, die keine oder eine ungültige Adresse haben, in das SMA-Data Netzwerk zu integrieren.

Anforderung:

Cmd: 03
 Ctrl: 10000000 b
 Dateninhalt: Seriennummer des Gerätes (dword)
 Neue Netzwerkadresse für das Gerät (word)
 Datenlänge: 6 Byte
 Quelladresse: Eigene Netzwerkadresse
 Zieladresse: Gruppe 0



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Seriennummer, Netzadresse SN=9380933, NetAdr=3
01 00	02 00	80	00	03	45 24 8F 00, 03 00

Antwort:

Cmd: 03
 Ctrl: 01000000 b
 Dateninhalt: Seriennummer des Gerätes
 Datenlänge: 4 Byte
 Quelladresse: neue Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Seriennummer SN=9380933
03 00	01 00	40	00	03	45 24 8F 00

4.3.1.5 Kanalinformation anfordern (CMD_GET_CINFO)

Um eine größtmögliche Flexibilität in Bezug auf Speicherkanalerweiterungen und Änderungen zu erhalten, kann die Speicher- und Parameterkonfiguration eines SMA-Data-Teilnehmers angefordert werden.

Der Teilnehmer antwortet mit der Auflistung seiner (Speicher-)Kanalinformation, d. h., es erfolgt die Übertragung der Beschreibung sämtlicher Ein- / Ausgänge und Parameter. Eine solche Beschreibung teilt sich in einen typunabhängigen Kopf und eine typspezifische Erweiterung auf.

Anforderung:

Cmd: 09
 Ctrl: 00000000 b
 Dateninhalt: -
 Datenlänge: 0 Byte

Quelladresse: Eigene Netzwerkadresse
 Zieladresse: Netzwerkadresse des Gerätes



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	-
01 00	02 00	00	00	09	-

Antwort:

Cmd: 09
 Ctrl: 01000000 b
 PktCnt: nnnn = (n-1) Pakete
 Dateninhalt: siehe Tabelle(n)
 Datenlänge: je nach Länge der Kanalliste des Gerätes
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	je Kanal: Index, Kanaltyp, Datenformat, Level, Name, Einheit, LoVal, HiVal, .. z.B.: 1, 01 04, 02 01, 02 00, 'SMA-SN', ' ', 0.0, 1.0E6
02 00	01 00	40	3B	09	01, 01 04, 02 01, 02 00, 53 4D 41 2D 53 4E 20 20 20 20 20 20 20 20 00, 20 20 20 20 20 20 00, 00 00 00 00, 00 24 74 49

Aufbau einer Kanalbeschreibung:**Typunabhängiger Teil:**

Abk.	Name	Beschreibung	Typ	Wert
No.	Index	laufende Kanalnummer diese Kanaltyps	byte	1..255
cTyp	Kanaltyp	<p>Bit FEDC BA98 7654 3210 0000 0000 0000 0000</p>	word	siehe Beschreibung
nType	Datenformat	<p>Bit FEDC BA98 7654 3210 0000 0000 0000 0000</p>	word	
nFill			word	
Name	Kanalname	Kanalname (Klartext)	char 16	
		SUMME	23	Byte

Analoge Größen:

Abk.	Name	Beschreibung	Typ	Wert
Unit	Einheit	Einheit des Kanals	char 8	
Gain	Verstärkung	Verstärkung des Kanals (Parameter LoVal)	float 4	
Ofs	Offset	Offset des Kanals (Parameter HiVal)	float 4	
		SUMME	16	Byte

Digitale Größen:

Abk.	Name	Beschreibung	Typ	Wert
TxtLo	Lo-Text	Text Signal = 0	char 16	
TxtHi	Hi-Text	Text Signal <> 0	char 16	
		SUMME	32	Byte

Zähler Größen:

Abk.	Name	Beschreibung	Typ	Wert
Unit	Einheit	Einheit des Kanals	char 8	
Gain	Verstärkung	Verstärkung des Kanals	float 4	
		SUMME	12	Byte

Status Größen:

Abk.	Name	Beschreibung	Typ	Wert
SizeT	Größe Textfeld	Länge der nachfolgenden Stringliste	word	
StatT	Statustexte	Liste von nullterminierten Strings entsprechend der maximalen Anzahl der Zustände. (die Länge eines einzelnen Eintrags ist auf max. 16 Zeichen begrenzt)	dynamisch	
		SUMME	2+SizeT	Byte

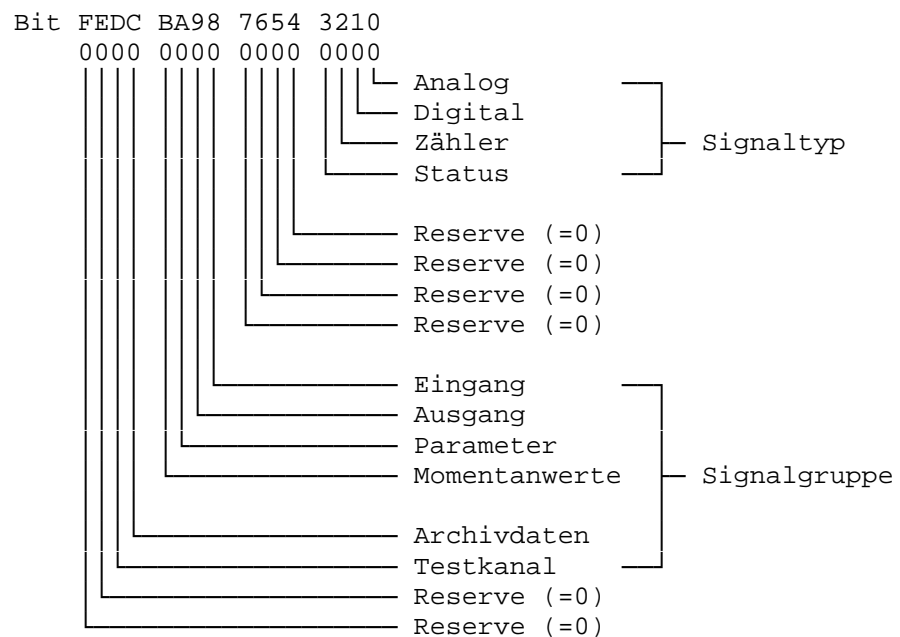
4.3.2 Kommandos zur Datenerfassung

Mit den Kommandos zur Datenerfassung können von einem Gerät Daten angefordert bzw. gesetzt werden. Es können sowohl Daten eines einzelnen Kanals als auch Daten einer Gruppe von Kanälen übertragen werden. Die Selektion, welche Kanäle übertragen werden sollen, wird durch die Übertragungsmaske (3 Byte) mitgeteilt.

Die Übertragungsmaske ist als ein Filter anzusehen. Alle Kanäle, die diesem Filter entsprechen, sind von dem auszuführenden Kommando betroffen.

Aufbau der Übertragungsmaske

1. Maske für Kanaltyp (word)



2. Kanalindex (byte)

= 0 ⇒ alle Kanäle laut 'Maske für Kanaltyp'

> 0 ⇒ nur ein Kanal (laut Kanal Nr.)

Im Prinzip ist jede beliebige Bitkombination in der Maske für den Kanaltyp zulässig, obwohl dies nicht unbedingt einen Sinn ergibt. Einige sinnvolle Kombinationen sind:

Bit	FEDC	BA98	7654	3210		
=	0000	1001	0000	1111	⇒	Anforderung Momentanwerte Ein-
	0	9	0	F	(Hex)	gangssignale aller Signaltypen
=	0001	0001	0000	1111	⇒	Anforderung Archivdaten Ein-
	1	1	0	F	(Hex)	gangssignale alle Signaltypen
=	0000	0001	0000	0100	⇒	Anforderung aktuelle Zähler-
	0	1	0	4	(Hex)	stände
=	0000	0100	0000	1111	⇒	Anforderung der Parameter
	0	4	0	F	(Hex)	aller Signaltypen

Soll nur ein einzelner Kanal angesprochen werden, so setzt sich die Übertragungsmaske aus dem Kanaltyp und dem Kanalindex des anzusprechenden Kanals zusammen.



Bitte beachten Sie, dass bei Übertragung des Kanaltyps zuerst der niederwertige Teil und dann der höherwertige Teil übertragen wird.

4.3.2.1 Onlinedaten synchronisieren (CMD_SYN_ONLINE)

Um eine zeitgleiche Erfassung von Onlinedaten zu gewährleisten, muss vor jedem Abfragezyklus ein Synchronisationsbefehl mit der aktuellen Zeit als Broadcastmeldung an alle Busteilnehmer gesendet werden. Alle Geräte, die diesen Befehl empfangen haben, frieren eine Kopie der aktuellen (Momentan-) Speicherwerte zusammen mit der übertragenen Uhrzeit ein. Eine Rückmeldung findet nicht statt. Nach dem Absetzen des Synchronisationskommandos erfolgt die Online-Datenabfrage an jedem Gerät.

Anforderung:

Cmd: 10
 Ctrl: 10000000 b
 Dateninhalt: Lokale Zeit in Sekunden seit dem 1.1.1970-00:00:00
 als Longint (keine Berücksichtigung der Sommer- /
 Winterzeit bzw. Zeitzonen).
 Datenlänge: 4 Byte
 Quelladresse: Eigene Netzwerkadresse
 Zieladresse: Gruppe x



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Unix-Zeitformat Zeit=843504044s = 0 x 32 46 09 AC
01 00	00 00	80	00	0A	AC D9 46 32

Antwort: keine

4.3.2.2 Datenanforderung (CMD_GET_DATA)

Dieser Befehl ermöglicht die Datenabfrage nach einer beliebigen Anforderungsmaske. Diese Maske ist identisch mit der Kanaltypdefinition der jeweiligen (Speicher-) Kanäle und setzt sich immer aus dem Kanaltyp (2 Byte) und dem Kanalindex (1 Byte) zusammen.

Die Antwort erfolgt dann entsprechend der in der Maske ausgewählten Kanäle. Entsprechend der Anforderungsmaske ergibt sich der Aufbau der zu übertragenden Datensätze:

Anforderung:

Cmd: 11
 Ctrl: 00000000 b

Dateninhalt: Übertragungsmaske (3 Byte)
 Datenlänge: 3 Byte
 Quelladresse: Eigene Netzwerkadresse = 1
 Zieladresse: Netzwerkadresse des Gerätes = 2



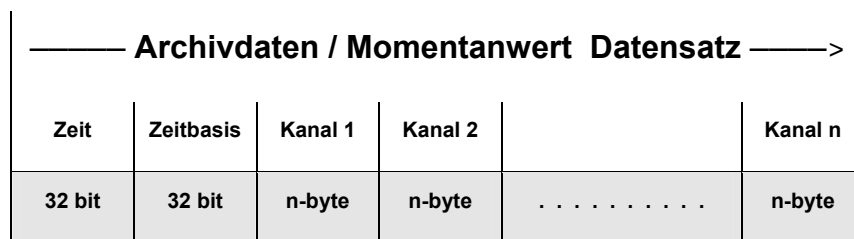
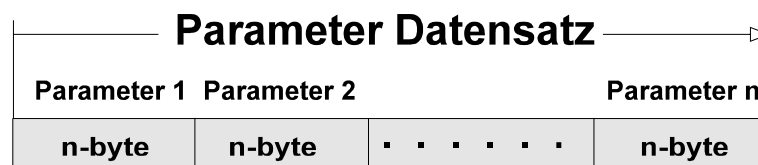
Protokollkopf					Nutzdaten		
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=09 0F, Kanal-Idx=00	opt. Unix time 1 843517290	opt. Unix time 2 843603690
01 00	02 00	00	00	0B	0F 09, 00	6A 0D 47 32	EA 5E 48 32



Unix time 1 und Unix time 2 sind nur bei Abfrage von Archivdaten enthalten (siehe Beispiel unten).

Antwort:

Cmd: 11
 Ctrl: 01000000 b
 Dateninhalt: Übertragungsmaske (3 Byte)
 Anzahl der Datensätze (word) (zu berechnen)
 Datensätze (nn Byte) (zu berechnen)
 Datenlänge: 3 Byte + nn Byte Datensätze
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse

Datensatzformat Archivdaten- / Momentanwertanforderung**Datensatzformat Parameteranforderung**



Antwortpaket Momentanwertanforderung

Protokollkopf					Nutzdaten ...
					Bytenummer, dezimal: 00 01 02
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske, ... Kanal-Typ=09 0F, Kanal-Idx=00;
02 00	01 00	40	00	0B	0F 09, 00;

...Nutzdaten...		
03 04	05 06 07 08	09 10 11 12
Anzahl Datensätze	Sek. seit 1.1.70	Zeitbasis
1	843517290	1s
01 00	6A 0D 47 32	01 00 00 00

...Nutzdaten...					
13 14	15 16	17 18	19 20	21 22	23 24
Analogdaten K1 Upv-Ist	Analogdaten K2 Upv-Soll	Analogdaten K3 lac-Ist	Analogdaten K4 lac-Soll	Analogdaten K5 Uac	Analogdaten K6 Fac
117	196	3748	3	223	4983
75 00	C4 00	A4 0E	03 00	DF 00	77 13

...Nutzdaten...					
25 26	27 28	29 30	31 32	33 34	35 36
Analogdaten K7 Pac	Analogdaten K8 Zac	Analogdaten K9 d-Zac	Analogdaten K10 R-Iso	Analogdaten K11 Uac-Srr	Analogdaten K12 Fac-Srr
835	37	2954	2954	221	4983
43 03	25 00	7C 13	8A 0B	DD 00	77 13

...Nutzdaten...					
37 38	39 40	41 42	43 44 45 46	47 48 49 50	51 52 53 54
Analogdaten K13 Zac-Srr	Analogdaten K14 lZac	Analogdaten K15 TKK	Zählerdaten K16 E-Total	Zählerdaten K17 h-Total	Zählerdaten K18 Netz-Ein
37	4765	605	4361490	296068	75
25 00	9D 12	5D 02	12 8D 42 00	84 84 04 00	4B 00 00 00

...Nutzdaten			
55 56 57 58	59 60 61 62	63	64
Zählerdaten K19 FehlerCnt	Zählerdaten K20 Snr	Statusdaten K21 Status	Statusdaten K22 Fehler
86	9380933	= MPP	= -----
56 00 00 00	45 24 8F 00	07	00

Archivdaten Abfragen (mit CMD_GET_DATA)

GET_DATA ermöglicht mit gesetztem Bit „Archivdaten“ die Abfrage von Datenaufzeichnungen.

Der Übertragungsmaske von 3 Byte kann hierbei (bei gesetztem Bit „Archivdaten“) noch ein Zeitbereich (8 Byte) angefügt werden. Dieser gibt den Zeitbereich in der Form „von Zeit“, „bis Zeit“ an. „Zeit“ ist eine Unixzeit und enthält somit sowohl Datum als auch Uhrzeit. Wird kein Zeitbereich angegeben (0 bis 0), so wird der maximale Zeitbereich angenommen (alle vorhandenen Archivdaten).



Für jeden Archivdatenkanal muss die Datenanforderung einzeln erfolgen.

Entsprechend der Übertragungsmaske ergibt sich der Aufbau der zu übertragenden Datensätze:

Anforderung Archivdaten:

Cmd:	11	
Ctrl:	00000000 b	
Dateninhalt:	Übertragungsmaske	(3 Byte)
	Zeitbereich:	
	von Zeit	(unsigned long)
	bis Zeit	(unsigned long)
Datenlänge:	11 Byte	
Quelladresse:	Eigene Netzwerkadresse	
Zieladresse:	Netzwerkadresse des Gerätes	



Beispiel für gesetztes Bit „Archivdaten“ bei Kanaltyp:

Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske; Zeitbereich Kanal-Typ=01 19, Kanal-Idx=04; von Zeit= 819936000, bis Zeit=843517290
01 00	02 00	00	00	0B	19 01, 04; 00 3B DF 30, 6A 0D 47 32

Antwort Archivdaten:

Cmd: 11
 Ctrl: 01000000 b
 Dateninhalt: Übertragungsmaske (3 Byte)
 Anzahl der Datensätze (word)
 Datensätze (nn Byte) (zu berechnen)
 Datenlänge: 3 Byte
 + 2 Byte Anzahl der Datensätze
 + (8+n) Byte je Datensatz
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse



Protokollkopf					Nutzdaten ...
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske, ... Kanal-Typ=01 19, Kanal-Idx=04;
02 00	01 00	40	00	0B	19 01, 04;

...Nutzdaten...				
Anzahl Datensätze	Speicherzeit 1	Zeitbasis 1	Wert 1	Speicherzeit 2
3	843517290	900s (15min)	114	843518190
02 00	6A 0D 47 32	84 03 00 00	72	EE 10 47 32

...Nutzdaten				
Zeitbasis 2	Wert 2	Speicherzeit 3	Zeitbasis 3	Wert 3
900s (15min)	121	843519090	900s (15min)	110
84 03 00 00	C4 00	27 14 47 32	84 03 00 00	6E

Aufbau der Antwortdatensätze:

— Archivdaten Datensatz —>		
Speicherzeit	Zeitbasis	Wert
32 bit	32 bit	n-byte

Speicherzeit: Speicherzeitpunkt, im Unix-Zeitformat

Zeitbasis: Speicherintervall in Sekunden (z. B. Tageswerte = 24*60*60)

Wert: Wert des Kanals, abhängig vom Kanaltyp

4.3.2.3 Daten senden (CMD_SET_DATA)

Der Befehl ermöglicht die Datenübertragung zu einem beliebigen SMA-Data-Netzwerkteilnehmer. Hiermit können Parameter, Zählerstände, digitale Ausgänge und Betriebsarten (Status) gesetzt werden. Der Dateninhalt setzt sich entsprechend der Übertragungsmaske zusammen. Der Empfänger antwortet mit einem Acknowledge bei korrekter Übertragung.

Anforderung:

Cmd:	12	
Ctrl:	00000000 b	
Dateninhalt:	Übertragungsmaske	(3 Byte)
	Anzahl Datensätze	(word)
	Datensatz (zu berechnen)	(nn Byte)
Datenlänge:	5 Byte + nn Byte	
Quelladresse:	Eigene Netzwerkadresse	
Zieladresse:	Netzwerkadresse des Gerätes	



Analogen Parameterkanal 2 auf 160 V setzen

Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=04 01, Kanal-Idx=02; Anzahl Datensätze=01, Wert=160
01 00	02 00	00	00	0C	01 04, 02; 01 00, A0 00

Antwort:

Cmd: 12
 Ctrl: 01000000 b
 Dateninhalt: Übertragungsmaske (3 Byte)
 Anzahl Datensätze (word)
 Datenlänge: 5 Byte
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=04 01, Kanal-Idx=02; Anzahl Datensätze=01
02 00	01 00	40	00	0C	01 04, 02; 01 00

4.3.2.4 Datenspeicher-Info abfragen (CMD_GET_SINFO)

Dieser Befehl ermöglicht die Abfrage der Speicherinformation evtl. vorhandener Archivdatenaufzeichnungen. Die Antwort enthält neben dem Aufzeichnungsintervall den Aufzeichnungsbereich (von, bis), für den Speicherwerte vorhanden sind.

Sind Speicherwerte vorhanden, können diese mit dem Kommando CMD_GET_DATA und gesetztem Archivdaten-Bit ausgelesen werden.

Anforderung:

Cmd:	13
Ctrl:	00000000 b
Dateninhalt:	Übertragungsmaske (3 Byte)
Datenlänge:	3 Byte
Quelladresse:	Eigene Netzwerkadresse
Zieladresse:	Netzwerkadresse des Gerätes

Antwort:

Cmd:	13
Ctrl:	01000000 b
Dateninhalt:	Übertragungsmaske (3 Byte) Datensätze (je 12 Byte)
Datenlänge:	3 Byte Übertragungsmaske + 12 Byte je Datensatz
Quelladresse:	Netzwerkadresse des Gerätes
Zieladresse:	Eigene Netzwerkadresse



Aufbau eines Antwort-Datensatzes:

Name	Speicherungsintervall	vonDat - Unix	bisDat - Unix
Byte	4	4	4
Beispiel	60	01.01.2002-00:00	04.01.2002-00:00

Die Nutzdaten der Antwort bestehen aus der Übertragungsmaske (3 Byte) und der zugehörigen Anzahl von Datensätzen. Diese enthalten jeweils 12 Byte. Ihre Anzahl errechnet sich entsprechend der Übertragungsmaske.

Speicherungsintervall

Das Speicherungsintervall wird in Sekunden angegeben. Somit ergibt sich, z. B. für Stunden-Archivdaten ein Speicherungsintervall von 3600.

4.3.3 Kommandos zur Konfiguration der Datenspeicherung

Diese Kommando-Gruppe ermöglicht das Festlegen der von einem Gerät aufzuzeichnenden Kanäle über das zugehörige Speicherungsintervall. Dabei werden nur Kanäle mit gesetztem Flag „Archivdaten“ berücksichtigt (siehe Aufbau der Übertragungsmaske), andere Kanäle können nicht aufgezeichnet werden.

4.3.3.1 Speicherungsintervalle lesen (CMD_GET_MTIME)

Mit diesem Befehl können die derzeit für die Archivdatenkanäle eingestellten Speicherungsintervalle ausgelesen werden. Über die Übertragungsmaske kann ein einzelner Kanal oder eine Kanalgruppe selektiert werden. Die Antwort enthält eine Liste der Speicherungsintervalle (in Sekunden) entsprechend der Übertragungsmaske.

Anforderung:

Cmd: 20
 Ctrl: 10000000 b
 Dateninhalt: Übertragungsmaske (3 Byte)
 Datenlänge: 3 Byte
 Quelladresse: Eigene Netzwerkadresse
 Zieladresse: Netzwerkadresse des Gerätes

Die Anforderung beinhaltet die Übertragungsmaske der Kanäle, deren Speicherungsintervalle ausgelesen werden sollen.



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=19 01, Kanal-Idx=01;
01 00	02 00	00	00	14	04 21, 02

Antwort:

Cmd: 20
 Ctrl: 01000000 b
 Dateninhalt: Übertragungsmaske (3 Byte)
 Datensätze (Speicherungsintervalle) (je 4 Byte)
 Datenlänge: 3 Byte Übertragungsmaske
 + 4 Byte je Datensatz
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse

Die Nutzdaten der Antwort bestehen aus der Übertragungsmaske (3 Byte) und der zugehörigen Anzahl von Datensätzen. Jeder Datensatz beinhaltet das Speicherungsintervall für einen durch die Übertragungsmaske selektierten Kanal.



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=19 01, Kanal-Idx=01; Speicherungsintervall (in Sekunden)
02 00	01 00	40	00	14	01 19, 01; 3C 00 00 00

4.3.3.2 Speicherungsintervalle setzen (CMD_SET_MTIME)

Mit diesem Befehl können die derzeit zur Datenaufzeichnung ausgewählten Archivdatenkanäle verändert werden. Wird einem Kanal das Speicherungsintervall 0 zugewiesen, so wird er nicht mehr aufgezeichnet. Die Speicherungsintervalle werden vom Gerät bestätigt. Unzulässige Speicherungsintervalle werden dabei auf zulässige Werte geändert. Kann ein Gerät nur ein für alle Kanäle gleiches Speicherungsintervall verarbeiten, so wird das erste gültige Speicherungsintervall verwendet. Alle weiteren davon abweichenden Intervalle (> 0) werden auf dieses Speicherungsintervall gesetzt und bestätigt.

Anforderung:

Cmd:	21
Ctrl:	10000000 b
Dateninhalt:	Übertragungsmaske (3 Byte) Datensätze (Speicherungsintervalle) (je 4 Byte)
Datenlänge:	3 Byte + 4 Byte je Datensatz
Quelladresse:	Eigene Netzwerkadresse
Zieladresse:	Netzwerkadresse des Gerätes



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=19 01, Kanal-Idx=01; Speicherungsintervall (in Sekunden)
01 00	02 00	00	00	15	01 19, 01; 3C 00 00 00

Antwort:

Cmd: 21
 Ctrl: 01000000 b
 Dateninhalt: Übertragungsmaske (3 Byte)
 Datensätze (Speicherungsintervalle) (je 4 Byte)
 Datenlänge: 3 Byte Übertragungsmaske
 + 4 Byte je Datensatz
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Netzwerkadresse

Die Antwort bestätigt die Anforderung, wobei unzulässige Speicherungsintervalle korrigiert werden.



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Übertragungsmaske Kanal-Typ=19 01, Kanal-Idx=01; Speicherungsintervall (in Sekunden)
02 00	01 00	40	00	15	01 19, 01; 3C 00 00 00

4.3.4 Kommandos zur Binärübertragung

Diese Kommando-Gruppe ermöglicht das direkte Auslesen bzw. Setzen von Speicherbereichen eines SMA-Data-Teilnehmers. Diese Kommandos sind optional und werden nicht von allen Geräten unterstützt.

4.3.4.1 Binärbereich-Info anfordern (CMD_GET_BINFO)

Dieser Befehl ermöglicht die Abfrage der vom Gerät für Binärdatenübertragung zur Verfügung stehenden Speicherbereiche und Datenfiles.

Das Antwortpaket enthält eine Liste der Speicherbereiche und Datenfiles, in deren Rahmen Binärdatenübertragungen zulässig sind. Werden die Kommandos CMD_GET_BIN oder CMD_SET_BIN mit unzulässigen Bereichsangaben verwendet, so werden diese vom Zielsystem verweigert.

Anforderung:

Cmd:	30
Ctrl:	00000000 b
Dateninhalt:	-
Datenlänge:	0 Byte
Quelladresse:	Eigene Netzwerkadresse
Zieladresse:	Netzwerkadresse des Gerätes

Antwort:

Cmd: 30
 Ctrl: 01000000 b

Dateninhalt:

Liste der Binärbereiche:

Binärbereich Nr.	(byte)
Änderungszähler	(byte)
Bereich Name	(16 byte)
Startadresse	(unsigned long)
Größe	(unsigned long)
Modus	(byte)

Datenlänge: 27 Byte je Bereich
 Quelladresse: Netzwerkadresse des Gerätes
 Zieladresse: Eigene Data Netzwerkadresse



Be- reich Nr.	Änderungs- zähler	Bereich Na- me	Startadresse	Größe	Modus
1	0	Daten RTC	0x0000 0000	0x0000 0080	0
2	0	Program-Flash	0x0004 0000	0x0004 0000	2
3	0	IO-Adressen	0x0000 0000	0x0000 FFFF	1

Modus: 0 := nur lesen
 1 := nur schreiben
 2 := lesen / schreiben

Wenn statt physikalischer Speicherbereiche Datenfiles transferiert werden, ist die Startadresse 0x0000 0000.

Die in der Antwort auf CMD_GET_BINFO enthaltene Größe gibt an, wie groß das File maximal werden kann. Bei dem Transfer mit CMD_GET_BIN wird diese Größe nicht unbedingt erreicht.

Durch den Änderungszähler ist zu erkennen, ob sich seit dem letzten Filetransfer der Inhalt des Datenbereiches verändert hat. Er wird bei jeder Änderung inkrementiert.

4.3.4.2 Binärdatenanforderung (CMD_GET_BIN)

Dieser Befehl ermöglicht die gezielte Datenabfrage einzelner Speicherbereiche und Datenfiles.

Die Anforderungsmaske enthält einen Index für den anzusprechenden Speicherbereich (Device: SRAM, Flash1,...), die Startadresse des Speicherbereiches und die Endadresse.

Die Anforderung der Daten erfolgt wie beschrieben, für die Startadresse und Bereichs-Länge gilt:

Die Startadresse enthält den Wert, bei dem begonnen werden soll oder bei der fortgefahren werden soll mit dem Transfer, wenn nicht alle Daten mit einem Paket gesendet werden konnten, d. h. die Startadresse der zweiten Anforderung ergibt sich aus der Summe der Startadresse der ersten Anforderung und Bereichslänge aus dem ersten Antworttelegramm. Zur neuen Startadresse wird also immer die Bereichslänge aus der vorherigen Antwort addiert.

Bei der Anforderung muss unter Bereichs-Länge die maximale Datensatzlänge angegeben werden, die aufgenommen werden kann, ohne die Bytes für Binärbereich Nr., Startadresse und Bereich-Länge.

Die Antwort liefert unter Bereichs-Länge die tatsächliche Länge des zurückgelieferten Datensatzes (ohne Binärbereich Nr., Startadresse und Bereich-Länge). Die Startadresse enthält den Wert aus der Anforderung.

Die Geräte überprüfen bei Folgepaketen die fortlaufende Startadresse und brechen den Transfer ab, wenn die Folge-Startadresse nicht der berechneten entspricht.

Enthält die Anforderung unter Bereichlänge den Wert Null, wird dieser Mechanismus nicht angewandt.

Entsprechend den Bereichsangaben der Anforderung ergibt sich der Aufbau der zu übertragenden Datensätze.

Anforderung:

Cmd: 31
 Ctrl: 00000000 b
 Dateninhalt:
 Bereichsangaben:
 Binärbereich Nr. (byte)
 Startadresse (unsigned long)
 Bereich-Länge (unsigned short)
 Datenlänge: 7 Byte
 Quelladresse: Eigene Netzwerkadresse
 Zieladresse: Netzwerkadresse des Gerätes

1. Beispiel:



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Binärbereich-Nr =01, Startadresse= 0000 0000, Bereichslänge=1000 (4096 Byte)
01 00	02 00	00	00	1F	01, 0000 0000, 00 10

Antwort:

Cmd: 31

Ctrl: 01000000 b

Dateninhalt:

Bereichsangaben:

Binärbereich Nr. (byte)

Startadresse (unsigned long)

Bereich-Länge (unsigned short)

Datensatz (Dateninhalt des Speicherbereiches)

Datenlänge: 7 Byte + nn Byte Datensatz

Quelladresse: Netzwerkadresse des Gerätes

Zieladresse: Eigene Netzwerkadresse

2. Beispiel: Transfer eines Textfiles**Format 1.Telegramm:****Anforderung:**

- Protokollkopf:
 - Quelle : 01 00 (2 Byte)
 - Ziel: 02 00 (2Byte)
 - Ctrl: 00 (1Byte)
 - PktCnt: 00 (1Byte)
 - Cmd: 1F (1Byte)
- Dateninhalt:

Bereichsangaben:

 - Binärbereich Nr: 03 (1Byte)

- Startadresse : 00 00 00 00 (4 Byte)
- Bereichslänge: 00 04 (2 Byte)

Antwort:

- Protokollkopf:
 - Quelle : 02 00 (2 Byte)
 - Ziel: 01 00 (2Byte)
 - Ctrl: 40 (1Byte)
 - PktCnt: FF (1Byte)
 - Cmd: 1F (1Byte)
- Dateninhalt:
Bereichsangaben:
 - Binärbereich Nr: 03 (1Byte)
 - Startadresse : 00 00 00 00 (4 Byte)
 - Bereichslänge: C8 00 (2 Byte)

Datensatz (Textfile Bsp.) : 00 46 01 02 20 46 61 68

**Format 2. Telegramm:****Anforderung:**

- Protokollkopf:
 - Quelle : 01 00 (2 Byte)
 - Ziel: 02 00 (2Byte)
 - Ctrl: 00 (1Byte)
 - PktCnt: FF (1Byte)
 - Cmd: 1F (1Byte)

- Dateninhalt:
Bereichsangaben:
 - Binärbereich Nr: 03 (1Byte)
 - Startadresse : C8 00 00 00 (4 Byte)
 - Bereich-Länge: 00 04 (2 Byte)

Antwort:

- Protokollkopf:
 - Quelle : 02 00 (2 Byte)
 - Ziel: 01 00 (2Byte)
 - Ctrl: 40 (1Byte)
 - PktCnt: FE (1Byte)
 - Cmd: 1F (1Byte)
- Dateninhalt:
Bereichsangaben:
 - Binärbereich Nr: 03 (1Byte)
 - Startadresse : C8 00 00 00 (4 Byte)
 - Bereich-Länge: C8 00 (2 Byte)

Datensatz (Textfile Bsp.) : 00 46 01 03 20 46 61 68

4.3.4.3 Binärdaten senden (CMD_SET_BIN)

Der Befehl ermöglicht das gezielte Schreiben einzelner Speicherbereiche.

Die Anforderungsmaske enthält einen Index für den anzusprechenden Speicherbereich, die Startadresse und die Endadresse des zu schreibenden Bereiches.

Entsprechend den Bereichsangaben der Anforderung ergibt sich der Aufbau der zu übertragenden Datensätze:

Anforderung:

Cmd: 32
Ctrl: 00000000 b
Dateninhalt:
 Bereichsangaben:
 Binärbereich Nr. (byte)
 Startadresse (unsigned long)
 Bereich-Länge (unsigned short)
 Datensatz (Dateninhalt des Speicherbereiches)
Datenlänge: 7 Byte + nn Byte Datensatz
Quelladresse: Eigene Netzwerkadresse
Zieladresse: Netzwerkadresse des Gerätes

Antwort:

Cmd: 32
Ctrl: 01000000 b
Dateninhalt:
 Bereichsangaben:
 Binärbereich Nr. (byte)
 Startadresse (unsigned long)
 Bereich-Länge (unsigned short)
Datenlänge: 7 Byte
Quelladresse: Netzwerkadresse des Gerätes
Zieladresse: Eigene Netzwerkadresse

4.3.5 Kommandos zu Variablen

Die Kommandos beinhalten im Wesentlichen Variablennummer, Inhalte von Variablen und Speicherkanalnummern.

Zur Identifizierung des Typs und Index eines Kanals wird eine Kanalnummer verwendet. Die Kanalnummer ist eine vierstellige Zahl. Aus der ersten Stelle geht der Kanaltyp hervor:

- 1: Analoger Kanal
- 2: Digitaler Kanal
- 3: Zähler Kanal
- 4: Status Kanal
- 5: Alle digitalen Kanäle, bitmaskiert, max. 32 Bit breit
- 6-9: nicht definiert.

Die restlichen Zahlen bilden den Index des Kanals in der Kanalliste (siehe Tabellen Speicherkanäle).



- Kanalnummer 1011: analoger Kanal mit dem Index Nr. 11
- Kanal 5001 Bit 3: digitaler Eingang mit dem Index 4

4.3.5.1 System Variablen erfragen (CMD_VAR_VALUE)

Dieses Kommando fordert den Inhalt einer oder mehrerer Variablen an.

Auf dieses Kommando können mehrere Teilnehmer antworten. Jede Komponente beantwortet ihre Variablen (Ausgangsgrößen).

Anforderung:

Cmd: 51
 Ctrl: 10000000 b
 Dateninhalt: Anzahl der Variablen: n , 1 bis max. 25 (2 Byte)
 Variablennummer $X_1 \dots X_n$ (n*2 Byte)
 Quelladresse: Netzwerkadresse
 Zieladresse: Broadcast



Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Anzahl , Variablennr. X_1 bis X_n
03 00	00 00	80	00	33	XX XX XX XX ... XX XX

Antwort:

Cmd: 51
 Ctrl: 11000000 b
 Dateninhalt: Anzahl der Variablen: m, 1 bis max. 25 (2 Byte)
 Variablennummer X_1 (2 Byte) - Inhalt W_1 (4 Byte) ...
 Variablennummer X_m - Inhalt W_m
 (insgesamt: 2 Byte + m * 6 Byte)
 Quelladresse: Netzwerkadresse
 Zieladresse: Broadcast

Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Anzahl , Variablennr. X ₁ , Inhalt W ₁ Variablennr. X _n , Inhalt W _n
03 00	00 00	C0	00	33	XX XX XX XX xx xx xx xx XX XX xx xx xx xx



Variablen - Anfrage

Variablen: 21 01 (Lüfter Gerät 1)

22 01 (Lüfter Gerät 2)

Anforderung:

Protokollkopf					Nutzdaten		
Quelle	Ziel	Ctrl	PktCnt	Cmd	Anzahl	Variable 1	Variable 2
03 00	00 00	80	00	33	02 00	01 21	01 22

Antworten:

Von Gerät 1 (Lüfter EIN):

Protokollkopf					Nutzdaten		
Quelle	Ziel	Ctrl	PktCnt	Cmd	Anzahl	Variable	Wert
01 00	03 00	C0	00	33	01 00	01 21	01 00 00 00

Von Gerät 2 (Lüfter AUS):

Protokollkopf					Nutzdaten		
Quelle	Ziel	Ctrl	PktCnt	Cmd	Anzahl	Variable	Wert
02 00	03 00	C0	00	33	01 00	01 22	00 00 00 00

4.3.6 Sonstige Kommandos

4.3.6.1 Begrenzung der Geräteleistung (CMD_PDELIMIT)

Um eine schnelle Begrenzung von Geräten zu ermöglichen, kann eine Broadcastmeldung an alle Busteilnehmer gesendet werden. Alle Geräte, die diesen Befehl empfangen haben (und das Kommando unterstützen), begrenzen ihre Ausgangsleistung.

Anforderung:

Cmd:	40
Ctrl:	10000000 b
Dateninhalt:	Typ der Begrenzung (1 Byte) 0: relative Begrenzung (bezogen auf aktuellen Wert –100 ... +100 %) 1: absolute Begrenzung (auf Begrenzungswert 0 .. 100 %) Begrenzungswert (in % vom Nennwert) (1 Byte – signed)
Datenlänge:	2 Byte
Quelladresse:	Eigene Netzwerkadresse
Zieladresse:	Gruppe x



Relative Begrenzung des aktuellen Ist-Wertes um 5 % vom Nennwert nach unten

Protokollkopf					Nutzdaten
Quelle	Ziel	Ctrl	PktCnt	Cmd	Typ relativ = 0 Begrenzung –5%
01 00	00 00	80	00	28	00 FB

Antwort: keine

4.3.6.2 Team-Funktion für Solarwechselrichter (CMD_TEAM_FUNCTION)

Dieses Kommando ist reserviert.

Anforderung:

Cmd: 60

5 Literatur

[1] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 50, RFC 1661, Daydreamer, July 1994.

[2] Simpson, W., Editor, "PPP in HDLC Framing", STD 51, RFC 1662, Daydreamer, July 1994.

[3] ISO/IEC 3309:1991(E), "Information Technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures - Framestructure", International Organization For Standardization, Fourth edition 1991-06-01.

[4] Postel, J. ASSIGNED NUMBERS, RFC 1700, , Daydreamer, October 1994.

[S] Europäische Normen:

EN 50065-1: 1991 +

A1: 1992 + A2: 1995 +

A3: 1996 D

