

Tixi SPS TiXML Handbuch

Tixi Alarm Modem

V 2.5

© 2005 Tixi.Com GmbH, Berlin

Redaktionsschluß: 7. Mai 2007. Unterstützte Tixi Alarm Modem Firmware: 2.2.12.0 (BETA 3.0.1.276)
Dieses Handbuch ist durch Copyright geschützt. Jede weitere Veräußerung ist nur mit der Zustimmung des Herausgebers gestattet. Dies gilt auch für Kopien, Mikrofilme, Übersetzungen sowie die Speicherung und Verarbeitung in elektronischen Systemen.

In diesem Handbuch verwendete Firmen- und Markennamen sind eigenständige Markenzeichen der betreffenden Firmen, auch wenn sie nicht explizit als solche gekennzeichnet sind.

Inhalt

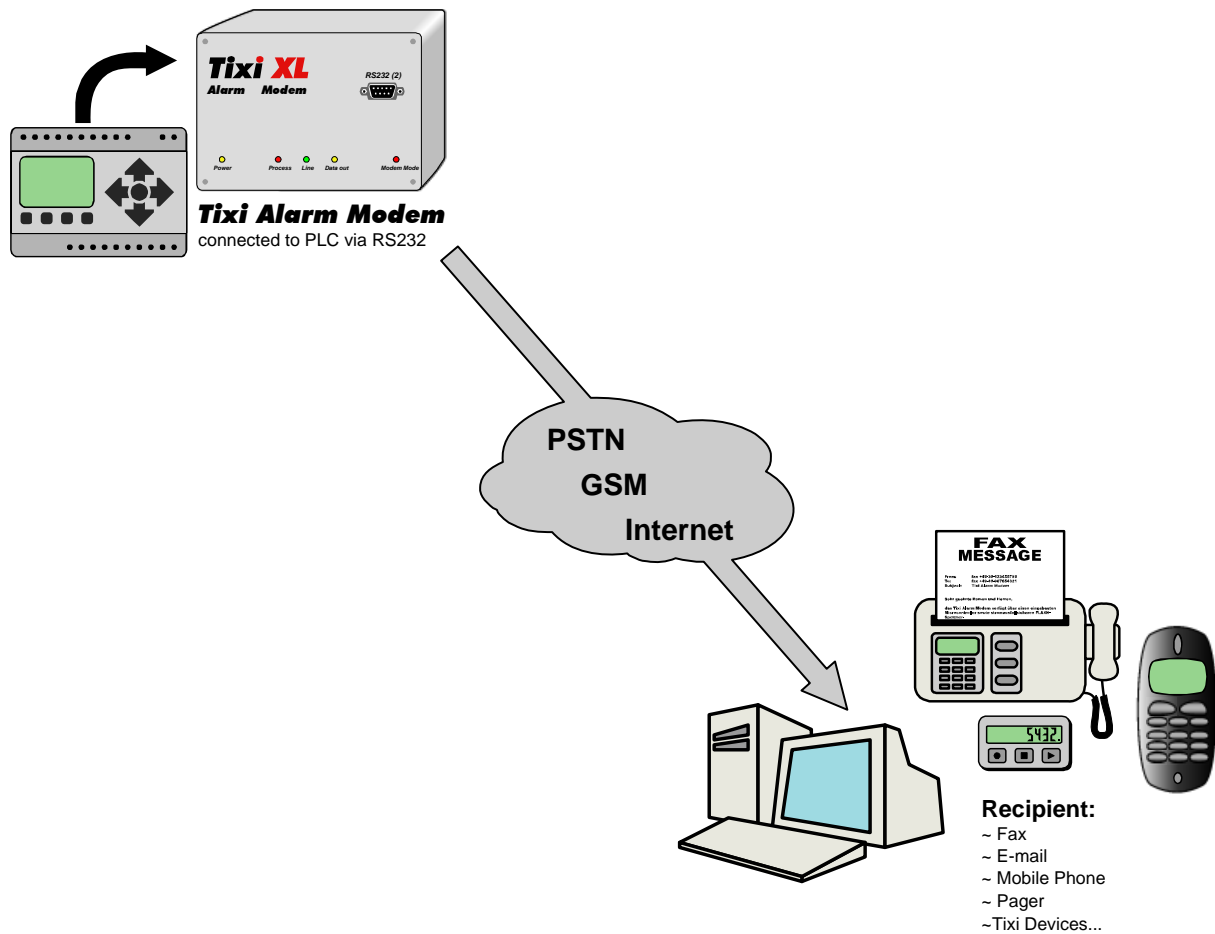
1	ÜBERSICHT	3
2	TIXI FÜR SPS-SYSTEME PARAMETRIEREN	4
2.1	SPS- UND BUSSYSTEME KONFIGURIEREN	4
2.2	GERÄTEVARIABLEN LESEN	11
2.3	GERÄTEVARIABLEN SETZEN	12
2.4	ARRAYS	13
2.4.1	<i>Arrays lesen.....</i>	<i>13</i>
2.4.2	<i>Arrays schreiben</i>	<i>13</i>
2.5	FEHLERZUSTÄNDE BEI GERÄTEVARIABLEN VERARBEITEN	14
2.5.1	<i>Fehlerzustand auslesen</i>	<i>14</i>
2.5.2	<i>Fehlerzustand verarbeiten</i>	<i>15</i>
3	UNTERSTÜTZTE SPS-SYSTEME	16
3.1	MITSUBISHI ALPHA XL	16
3.2	MITSUBISHI MELSEC FX.....	18
3.3	SIEMENS SIMATIC S7-200	20
3.4	SIEMENS SIMATIC S7-300/400	22
3.5	VIPA CPU 100/200/300	24
3.6	MOELLER EASY 400 / 500 / 600 / 700 / 800 / MFD	26
3.7	MOELLER PS30 & PS4/40.....	29
3.8	SAIA BURGESS S-BUS	30
3.9	CAREL MACROPLUS	32
3.10	ABB AC010, AC031, CL REIHE.....	33
3.11	ALLEN BRADLEY PICO	33
3.12	THEBEN PHARAO II	33
4	FELDBUS UNTERSTÜTZUNG.....	34
4.1	TIXI-BUS	34
4.2	ASCII PROTOCOL.....	35
4.2.1	<i>Definition der Variablenabfrage-Strings</i>	<i>36</i>
4.2.2	<i>Auswertung der Strings.....</i>	<i>36</i>
4.2.3	<i>Modifizierung der Werte</i>	<i>37</i>
4.3	MODBUS RTU	38
4.4	MODBUS ASCII	40
4.5	M-BUS	40
4.6	CAN-BUS	42
5	FORMATIEREN VON SPS VARIABLENWERTEN	42
6	VERWENDEN DER SPS-VARIABLEN IM TIXI ALARM MODEM.....	47
6.1	STANDARDADRESSIERUNG	47
6.2	ADRESSIERUNG ÜBER BUSNAME UND STATIONSNAME.....	47
6.3	ÜBERWACHUNG DER SPS-KOMMUNIKATION	47

1 Übersicht

Das **Tixi Alarm Modem** ist ein völlig neuartiger Typ von Kommunikationsgerät, welcher ohne großen Aufwand in bestehende Systeme eingebunden werden kann.

Die Kommunikationsprotokolle verbreiteter SPS-Systeme sind bereits im Tixi Alarm Modem vorhanden, somit ist keine Anpassung in der SPS notwendig. Andere SPS-Systeme können das Tixi Alarm Modem über einfache **TiXML**-Textbefehle steuern.

Eine einfache Anwendung des Tixi Alarm Modem:



Dieses Handbuch beschreibt die notwendigen Einstellungen des Tixi Alarm Modem zum Anschluß an die SPS. Zusätzlich stellt es eine Übersicht der derzeit unterstützten SPS-Systeme und deren Variablenumfang, sowie Befehle zur Formatierung von SPS-Variablenwerten dar.

2 Tixi für SPS-Systeme parametrieren

2.1 SPS- und Bussysteme konfigurieren

Um das Tixi Alarm Modem mit der SPS kommunizieren zu lassen, muss die External-Datenbank des Tixi Alarm Modems parametrieren werden. Diese Beschreibung setzt die grundlegende Kenntnis der TiXML-Sprache und des Tixi Alarm Modems voraus.

Die Datenbank der SPS-Konfiguration ist PROCCFG/External und sieht wie folgt aus:

```
<External>
  <Bus _="Bus" BusId="BusId" {Name="Alias"} protocol="Proto"
  type="BType" {baud="Speed"} {format="Dataformat"}
  {handshake="handshake"} {Mem="Memory"} [TS="OwnID"]
  MAXADR="Range" [GUF="Factor"] [RC="Retries"]>
  {<Condition _="Name" Variable="Path" Pollrate="Rate"/>}
  <Device _="ID" {Name="Alias"} {Pollrate="Rate TUnit"}
  [CharTimeout="CT TUnit "] [Pause="Wait TUnit "]
  [Timeout="Timeout TUnit"] [DWordInc="AI"] [DwordSwap="Swap"]
  [ForceSingleWordWrite="Funct"] [PrimaryAddr="PA"]
  [SecondaryAddr="SA"] [FabricationAddr="FA"]
  [ManufactoryCode="MC"] [Generation="Gen"] [Medium="Med"]
  [devType="DType"] [UseCache="Cache"] [MaxElements="Elements"]
  {Condition="Name"}>
    <VName _="VType" {simpleType="BasicType"} [exp="Exp"]
    [precision="Precision"] [size="ArraySize"] acc="Rights
    Storage" [ind="Index"] [subind="Index2"] [no="Array"]
    {def="default"} {multip="Factor"} {format="Format"}
    {write="wFunct"} {read="rFunct"}/>
  </Device>
</Bus>
</External>
```

Attribute in {...} sind optional.

Attribute in [...] werden nur für bestimmte Geräte oder Bussysteme benötigt.

Übersicht der möglichen Parameter:

<Bus>-Parameter

<i>Bus</i>	Definiert die Schnittstelle, an der die SPS angeschlossen ist. Mögliche Werte: MB SPS an der RS232 des Mainboards C0 SPS an der RS422/485 der Erweiterungskarte C1 SPS an der RS232 (2) der Erweiterungskarte COM1 SPS an COM1 (Nur Hx-Modems ab FW 1.80.0.0, erfordert BusId) COM2 SPS an COM2 (Nur Hx-Modems ab FW 1.80.0.0, erfordert BusId)
<i>BusId</i>	Ermöglicht die Adressierung des SPS-Bus unabhängig von der Schnittstelle. Darf nicht mit „Name“ vermischt werden.
<i>Alias</i>	Ermöglicht die Benennung des SBS-Bus und dadurch eine Schnittstellenunabhängige Adressierung. Darf nicht mit „BusId“ vermischt werden. (max. 20 alphanumerische Zeichen, keine Sonderzeichen, darf nicht mit einer Zahl beginnen)

Proto Bestimmt das SPS-Protokoll oder Bus-Protokoll, das ist abhängig vom der angeschlossenen Gerät, SPS oder Bussystem angeschlossen ist. Die gültigen Werte können Kapitel 3 entnommen werden.

BType Bestimmt den Kommunikationsmodus gegenüber der SPS: *Master* oder *Slave*. Die gültigen Werte können Kapitel 3 entnommen werden.

Speed Bestimmt die Baudrate in bps zwischen dem Tixi Alarm Modem und der SPS bzw. beschreibt die Baudrate auf dem Bussystem (z.B. 19200). Die gültigen Werte können Kapitel 3 entnommen werden.

Dataformat

Bestimmt das Datenformat auf der seriellen Schnittstelle. Das Datenformat wird abhängig von der verwendeten SPS oder dem Bussystem (z.B. 8E1 bei S7-200) gewählt. Die gültigen Werte können Kapitel 3 entnommen werden.

Syntax: **DataBitsParityBitsStopBits**

DataBits:

8..8 Datenbits

7..7 Datenbits

ParityBits:

N..kein Paritätsbit

E..gerade Parität

O..ungerade Parität

StopBits:

1..1 Stopbit

2..2 Stopbits

Handshake

Bestimmt das Soft- oder Hardwarehandshake zwischen Alarm Modem und SPS oder Bussystem.

None	Kommunikation ohne Handshake
XONXOFF	Software Handshake
XONXOFFPASS	software Handshake, XONXOFF wird an Anwendung weitergeleitet
RTSCTS	Hardware Handshake mit RTS CTS
DTRDSR	Hardware Handshake mit DTR DSR
HALF	Halbduplex RS 485 Kommunikation
FULL	Vollduplex RS 485/422
HALFX	Halbduplex RS 485 Kommunikation mit XON XOFF
FULLX	Vollduplex RS 485/422 mit XON XOFF
noDTR	deaktiviert die DTR

Memory Definiert, wie viel Speicher (in byte) für den Bus reserviert wird. Insgesamt stehen für alle möglichen Bus-Definitionen 131072 byte zur Verfügung.

OwnID Legt die Stationnummer des Alarm Modems fest. Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.

<i>Range</i>	Maximal abzufragende Stationensnummern. Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Factor</i>	"Gap update factor" zur Erkennung weiterer Slaves Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Retries</i>	Anzahl Wiederholungen bei Kommunikationsfehlern Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.

Bis zur Tixi Alarm Modem Firmware 1.72.x.x ist nur ein BUS-Eintrag zulässig.

Ab Firmware-Version 1.80.x.x können zwei verschiedene SPS-Systeme an zwei Schnittstellen definiert werden. Wird eine der Schnittstellen für MPI-Kommunikation genutzt, so muss dieser an erster Stelle in der External parametrisiert sein.

<Device>-Parameter

<i>ID</i>	Legt die Stationsnummer des abzufragenden Gerätes oder SPS fest. Diese muss mit der in der SPS-Station (Geräte-Station) parametrisierten Kennung übereinstimmen, (z.B. 1). Einige Bus-Systeme erlauben mehrere Stationen (Netzwerk).
<i>Alias</i>	Ermöglicht die Benennung der SPS-Station und dadurch eine Stationsnummernunabhängige Adressierung.
<i>Rate</i>	Abfragezyklus des Tixi Alarm Modems im Master-Mode bzw. Kommunikationstimeout im Slave-Mode in der angegebenen Zeiteinheit (siehe <i>TUnit</i> , default ist 60s).
<i>CT</i>	Timeout zwischen den Zeichen in der angegebenen Zeiteinheit (siehe <i>TUnit</i>) Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Wait</i>	Pause zwischen den Nachrichten in der angegebenen Zeiteinheit (siehe <i>TUnit</i>) Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Timeout</i>	Timeout für Antwort in der angegebenen Zeiteinheit (siehe <i>TUnit</i>) Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>AI</i>	AdressIncrement zwischen zwei aufeinanderfolgenden DWORDSs (2, nur Modbus RTU). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Swap</i>	Muss gesetzt werden, wenn Low vor High Word in DWORD gesendet wird (0, nur Modbus RTU). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Funct</i>	Setzen, wenn Funct 0x06 anstelle von 0x10 für einzel WORD Schreiben verwendet werden soll (0, nur Modbus RTU). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>PA</i>	Primäradresse (Nur M-Bus). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>SA</i>	Sekundäradresse (Nur M-Bus). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.

<i>FA</i>	Fabrikationsadresse (Nur M-Bus). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>MC</i>	Herstellercode (Nur M-Bus). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Gen</i>	Gerätegeneration (Nur M-Bus). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Med</i>	Gerätemedium (Nur M-Bus). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>DType</i>	CPU-Typ (Nur Mitsubishi FX). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Cache</i>	Ist der Wert auf 0 gesetzt, wird das Zusammenfassen von aufeinanderfolgenden Variablen zu blockweisen Abfragen (Caching) deaktiviert. Alle Variablen werden in einzelnen Abfragen geholt. (Default: 1, nur Modbus RTU). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Elements</i>	Anzahl der Variablen die pro Modbus-Telegramm abgefragt werden (caching). Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>Name</i>	Name der Bedingung zur Änderung der Pollrate.
<i>TUnit</i>	Zeiteinheit für Zeitangaben (ms, s, m, h).

<Condition>-Parameter

Conditions werden verwendet, um die Pollrate abhängig von einer Bedingung zu verändern

<i>Name</i>	Name der Bedingung, auf die im Device-Abschnitt verwiesen wird.
<i>Path</i>	Variable, durch welche die Bedingung als erfüllt gilt (=1). Wird ohne Referenz-Zeichen eingegeben.
<i>Rate</i>	Abfragezyklus des Tixi Alarm Modems im Master-Mode bzw. Kommunikationstimeout im Slave-Mode (z.B. 1s , 2m , 6h)

<Variablen>-Parameter

Aus Performancegründen wird empfohlen, nicht mehr als 100 Variablen zu definieren. In einer Station darf es keine doppelten Variablen mit identischem Typ und Index geben (z.B. zweimal Merker 2 mit unterschiedlicher Formatanweisung).

<i>VName</i>	Name der im Folgenden parametrisierten Variable. Der Name darf bis 20 Zeichen lang sein.
<i>VType</i>	Bestimmt den Variablentyp der SPS bzw. des Busprotokolls, z.B. einen Zähler oder Merker. Die gültigen Typen können in Kapitel 3 nachgeschlagen werden.
<i>BasicType</i>	Bestimmt den Basistyp der Variablen, wie er in der jeweiligen Anwendung verwendet wird. Der Basistyp bestimmt die zugehörige Formatierungsmöglichkeiten und die native Darstellung im TiXML-Protokoll. Einem Variablen Typ (Attribute = “_“) können mehrere Basistypen (Attribut „simpleType“) zugeordnet sein von denen je Variableneintrag eine ausgewählt werden muss. Für verschiedene Basistypen müssen weitere Attribute angegeben werden (siehe „exp“, „size“). Folgende Werte sind möglich:

BasicType Wert	Bedeutung	Weitere Attrib.	native Darstellung Beispiele
Uint8	Vorzeichenloser 8 Bit Wert (0...255)	exp	"123" (exp = 0) "12.3" (exp = -1) "12300" (exp = 2) "0.0123" (exp = -4)
Uint16	Vorzeichenloser 16 Bit Wert (0...65535)	exp	"12345" (exp = 0) "1234.5" (exp = -1) "1234500" (exp = 2)
Uint32	Vorzeichenloser 32 Bit Wert (0...4294967295)	exp	"1234567" (exp = 0) "123456.7" (exp = -1) "123456700" (exp = 2)
Int8	Vorzeichenbehafteter 8 Bit Wert (-128...+127)	exp	„123“ (exp=0) „-123“ (exp=0) "1.23" (exp=-2) "12300" (exp=2)
Int16	Vorzeichenbehafteter 16 Bit Wert (-32768...32767)	exp	"-12345" (exp = 0) " 12345" (exp = 0) "-1234.5" (exp = -1) " 1234.5" (exp = -1) "-1234500" (exp = 2) " 1234500" (exp = 2)
Int32	Vorzeichenbehafteter 32 Bit Wert (-2147483648...2147483647)	exp	"-1234567" (exp = 0) " 1234567" (exp = 0) "-123456.7" (exp = -1) " 123456.7" (exp = -1) "-123456700" (exp = 2) " 123456700" (exp = 2)
String	Text (0...size Zeichen)	size	„Das ist ein Text „ (nur die ersten 100 Zeichen)
Blob	Binärdaten Array (0...size Byte) currently not supported	size	"AF037FFF" Byteweise hexadezimal (nur die ersten 100 Byte)
Bit	Digitale Wert (0...1)		„0“ „1“
Float	Gleitkommazahl einfache Genauigkeit ($\pm 3.402823466 \cdot 10^{38}$)	exp	12.34567 -0.001234 -0.123456 E-7 0.123456 E+7 (Exponentialdarstellung bei Exponent > 6)
Double	Gleitkommazahl doppelte Genauigkeit ($\pm 1.7976931348623158 \cdot 10^{308}$)	exp	12.345678910 -0.0012345678 -0.1234567890 E-11 0.1234567890 E+11 (Exponentialdarstellung bei Exponent > 11)

Rights

Legt die Zugriffsrechte des Tixi Alarm Modems auf die Variable fest:

R Lesezugriff
W Schreibzugriff
RW Lese-/Schreibzugriff

Die gültigen Werte dieses Attributwertes sind vom Variablentyp abhängig und können Kapitel 3 entnommen werden.

<i>Storage</i>	<p>Legt weitere Speicher und Zugriffsoptionen fest:</p> <p><i>L</i> (RWL) gemeinsamer Speicher (nur bei S-Bus Variablen mit Lese-Schreibzugriff)</p> <p><i>A</i> (RA, WA, RWA) Aktiviert aktiven Variablenzugriff (Siemens)</p> <p><i>C</i> Aktiviert den gecacheten Zugriff auf Variablenblöcke, falls der generelle Cache via 'UseCache' (s.o.) deaktiviert wurde. Ist das Flag gesetzt, wird die betreffende Variable nicht sofort abgefragt, sondern geprüft, ob die folgende auch noch in der Abfrage mitgelesen werden kann. Hat diese einen anderen Typ, dann erfolgt die Abfrage trotzdem einzeln.</p> <p>Für verschiedene SPS bzw. Geräte werden weitere Zugriffsoptionen definiert. Die gültigen Werte dieses Attributswertes sind vom Variablentyp abhängig und können Kapitel 3 entnommen werden.</p>
<i>Index</i>	Variablenadresse. Sie ist abhängig vom gewählten Variablentyp und muss mit der Parametrierung des Geräts bzw. Busses übereinstimmen. Die gültigen Adressen können Kapitel 3 entnommen werden.
<i>Index2</i>	Variablensubadresse ist abhängig vom gewählten Variablentyp und muss mit der Parametrierung des Geräts bzw. Busses übereinstimmen. Die gültigen Adressen können Kapitel 3 entnommen werden.
<i>Array</i>	Anzahl der Elemente, die als Array abgefragt werden.
<i>Default</i>	Startwert der Variable. Bei Variablen mit Schreibzugriff wird dieser bei jedem (!) Systemstart in die SPS geschrieben. Bei Variablen mit Lesezugriff wird der Wert beim Systemstart verwendet, bis das Modem den tatsächlichen Wert aus der SPS erhält. Der Startwert muss passend zum „exp“ und simpleType angegeben werden (siehe Tabelle Kap. 2.3).
<i>ArraySize</i>	<p>1. simpleType = String (siehe BasicType) Die maximale Anzahl von ASCII Zeichen in einem Textwert (nur gültig für den Typ simpleType="String", 0...65535). Bei null-terminierten Strings muss das Null-Zeichen mit berechnet werden. Der Wert ist abhängig vom Gerätetyp bzw. vom Busprotokoll (optional, abhängig vom Basistyp, Gerät bzw. Bussystem).</p> <p>2. simpleType= Blob (siehe BasicType) Die maximale Anzahl von Bytes in einem Bytearray (nur gültig für den Typ simpleType="Blob", 0...65535). Der Wert ist abhängig vom Gerätetyp bzw. vom Busprotokoll (optional, abhängig vom Basistyp, Gerät bzw. Bussystem).</p>
<i>Factor</i>	<p>Der vom Gerät empfangene Wert wird mit diesem Faktor multipliziert, bevor er weiter verarbeitet wird:</p> $\text{valueTAM} = \text{Factor} * \text{valueDevice}$ $\text{valueDevice} = 1/\text{Factor} * \text{valueTAM}$ <p>Der Faktor wird durch einen Bruch dargestellt z.B.: „1/1000“ oder „3600/1“ der Nenner und der Zähler darf nicht Null sein. Die Verwendung dieses Attributs ist vom Variablentyp abhängig Die gültigen Werte können Kapitel 3 entnommen werden</p>
<i>Exp</i>	<p>Exponent zur Basis 10 der die Auflösung einer Festkommazahl vom Typ simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (siehe BasicType) beschreibt. Der im TAM gespeicherte Werte wird mit $10^{\text{exp}(\text{Exp})}$ multipliziert (nachdem</p>

der *Factor* angewandt wurde), um den Wert des Parameters zu ermitteln.

$\text{valueParameter} = 10^{\text{Exp}} * \text{valueTAM}$.

Der Exponent bestimmt damit die Position des Kommas bei einer Festkommazahl an.

Es gibt folgende Werte für den Exponenten

Exp Wert	Beschreibung
-6	Auflösung = 0,000001
-5	Auflösung = 0,00001
-4	Auflösung = 0,0001
-3	Auflösung = 0,001
-2	Auflösung = 0,01
-1	Auflösung = 0,1
0	Auflösung = 1 (Standard)
1	Auflösung = 10
2	Auflösung = 100
3	Auflösung = 1000
4	Auflösung = 10000
5	Auflösung = 100000
6	Auflösung = 1000000

Precision Genauigkeit der Integerdarstellung eines Wertes mit **simpleType = Float , Double (siehe BasicType)**.

Der im TAM gespeicherte Wert wird mit $10^{\text{exp}(\text{Exp})}$ multipliziert, um den Wert in die Integerdarstellung zu wandeln. Die Integerdarstellung wird bei der Berechnung der Prozessvariablen z.B. mit den Befehlen (GT, LT etc.) verwendet. Er gibt somit die Genauigkeit bei der Berechnung der Prozessvariablen an und ist Abhängig von der Anwendung zu definieren.

$\text{Integerdarstellung} = \text{Ganzzahl}(10^{\text{Exp}} * \text{valueParameter})$.

Die Werte des entsprechen der folgenden Tabelle.

Precision Wert	Beschreibung
-6	Faktor = 0,000001
-5	Faktor = 0,00001
-4	Faktor = 0,0001
-3	Faktor = 0,001
-2	Faktor = 0,01
-1	Faktor = 0,1
0	Faktor = 1 (Standard)
1	Faktor = 10
2	Faktor = 100
3	Faktor = 1000
4	Faktor = 10000
5	Faktor = 100000
6	Faktor = 1000000

Format Die Formatoptionen kennzeichnen die Standardformatierung des Wertes, wie sie bei Ausgaben z.B. in E-Mails oder beim Get Befehl (siehe Kapitel 5) erfolgt.

- wFunct* Funktionscode, der beim Schreiben einer Variable verwendet wird. Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
- rFunct* Funktionscode, der beim Lesen einer Variable verwendet wird. Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.

2.2 Gerätevariablen lesen

Der Wert einer Variablen aus der External Datenbank kann, wie jeder andere Variablenwert, per TiXML Befehl „Get“ abgefragt werden. Dieser Befehl wird für die Variablen aus der External Datenbank um das Attribut „format“ erweitert.

Befehl:

[<Get _="Vpath" format ="Format"/>]

Übersicht der möglichen Parameter (kursiv geschriebene Werte):

Vpath: Pfad zur Adressierung des Parameters oder des Gerätes oder des Busses.

Format Formatangabe, folgende Werte sind möglich:

Formatwert	Beschreibung
„integer“	<p>1. simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (siehe 2.1 BasicType). Der Wert wird in der Integer-Darstellung ausgegeben. Die Integer-Darstellung berechnet sich mit Hilfe des für die Variable definierten Exponenten aus dem Wert der Variablen (siehe 2.1 <i>Exp</i>) :</p> $\text{Wert in Integer Darstellung} = 10^{-Exp} * \text{Wert}$ <p>2. simpleType = float, double (siehe 2.1 BasicType). Der Wert wird in der Integer-Darstellung ausgegeben. Die Integer-Darstellung berechnet sich mit Hilfe der für die Variable definierten Präzision aus dem Wert der Variablen (siehe 2.1. <i>Precision</i>) :</p> $\text{Wert in Integer Darstellung} = 10^{-Precision} * \text{Wert}$ <p>3. alle anderen Datentypen (siehe 2.1 BasicType). Der Wert wird in der nativen Darstellung ausgegeben (native Darstellung siehe 2.1 <i>BasicType</i>).</p>
Leer(„“) oder „native“	Der Wert wird in der nativen Darstellung ausgegeben (native Darstellung siehe 2.1 <i>BasicType</i>)
Formatstring	Der Wert wird entsprechend der angegebenen Formatierung ausgegeben (5, Formatieren von SPS Variablenwerten).

Fehlt das Formatattribut, wird der Wert in dem Format ausgegeben, das als Standardformat in der Variablendefinition (siehe Anforderung 2.1) angegeben wurde. Ist kein Standardformat definiert worden wird der Wert in diesem Fall in der nativen Darstellung ausgegeben (native Darstellung siehe 2.1 *BasicType*).

Antwort:

```
[<Get _="Value" />]
```

Value: Wert in der eingestellten Formatierung bzw. Darstellung.

2.3 Gerätevariablen setzen

Der Wert einer Variablen aus der External Datenbank kann, wie jeder andere Variablenwert, per TiXML Befehl „Set“ gesetzt werden. Dieser Befehl wird für die Variablen aus der External Datenbank um das Attribut „format“ erweitert.

Befehl:

```
[<Set _="Vpath" value="Value" format="Format"/>]
```

Übersicht der möglichen Parameter (kursiv geschriebene Werte):

Vpath: Pfad zur Adressierung des Parameters.
Value: Wert des Parameters .
Format: Formatangabe, folgende Werte sind möglich:

Formatwert	Beschreibung
„integer“	<p>1. simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (siehe 2.1 BasicType). Der Wert wird in der Integer-Darstellung eingegeben. Die Integer-Darstellung berechnet sich mit Hilfe des für die Variable definierten Exponenten (siehe 2.1 <i>Exp</i>) :</p> $\text{Wert in Integer Darstellung} = 10^{-Exp} * \text{Wert}$ <p>2. simpleType = float, double (siehe 2.1 BasicType). Der Wert wird in der Integer-Darstellung eingegeben. Die Integer-Darstellung berechnet sich mit Hilfe der für die Variable definierten Präzision :</p> $\text{Wert in Integer Darstellung} = 10^{-Precision} * \text{Wert}$ <p>3. alle anderen Datentypen (siehe 2.1 BasicType). Der Wert wird in der nativen Darstellung eingegeben (native Darstellung siehe 2.1 <i>BasicType</i>)</p>
Leer(„“) oder „native“	Der Wert wird in der nativen Darstellung eingegeben (native Darstellung siehe 2.1 <i>BasicType</i>)
Formatstring	Der Wert wird entsprechend der angegebenen Formatierung eingegeben (z.Z. nicht implementiert).

Fehlt das Formatattribut, wird der Wert wie im Falle des Formatwertes „native“ eingegeben.

Antwort:

```
[<Set/>]
```

2.4 Arrays

Sind Variablen in der External als Array angelegt (Attribut "no"), z.B.

```
<Variable_0 _="B" ind="22" no="8" acc="RW"/>
```

so lassen sich die einzelnen Werte der Arrays durch einen Variablensuffix in eckigen Klammern adressieren:

2.4.1 Arrays lesen

Das adressierte Element im Array wird in eckige Klammern angehängt,

```
[<Get _="Vpath[element]"/>]
```

vpath: Pfad zur Adressierung des Arrays.

element: Adressiertes Element des Arrays.

z.B. der dritte Wert im Array:

```
[<Get _="/Process/Aux?/D?/Variable_0[3]"/>]
```

Wird kein Suffix angegeben, werden bei der Ausgabe alle Elemente durch Komma getrennt aufgelistet:

```
[<Get _="Vpath" >]
```

Antwort:

```
[<Get _="Value1,Value2,Value3,...Value8" />]
```

2.4.2 Arrays schreiben

Das adressierte Element im Array wird in eckige Klammern angehängt,

```
[<Set _="Vpath[element]" value="Value"/>]
```

vpath: Pfad zur Adressierung des Arrays.

element: Adressiertes Element des Arrays.

value: Wert des Elements .

z.B. der dritte Wert im Array:

```
[<Set _="/Process/Aux?/D?/Variable_0[3]" value="20"/>]
```

Wird kein Suffix angegeben, müssen im Value alle Elemente durch Komma getrennt aufgelistet werden. Lücken sind nicht zulässig:

```
[<Set _="Vpath" value="Value1,Value2,Value3,Value4,Value5,Value6,Value7,Value8">]
```

Antwort:

```
[<Set/>]
```

2.5 Fehlerzustände bei Gerätevariablen verarbeiten

Die in der `External` Datenbank definierten Variablen können, abhängig vom Gerätetyp bzw. Bussystem, auch unterschiedliche Fehlerzustände speichern. Insbesondere bei der Abfrage über ein Kommunikationsprotokoll können beispielsweise Kommunikationsfehler oder Protokollfehler auftreten, so dass der Variablenwert (Wert des letzten fehlerfreien Zugriffs bzw. initialer Wert) ungültig ist.

Der Fehlerzustand der Variablen wird durch die beiden Fehlercodes `ErrorClass` und `ErrorNumber` dargestellt. Beide können ausgelesen bzw. für die Generierung eines Alarms weiterverarbeitet werden.

2.5.1 Fehlerzustand auslesen

Ist der Wert einer Variablen ungültig, d.h. es ist ein Fehler erkannt und im Fehlerzustand gespeichert worden, so wirkt sich dies in folgender Weise auf den Lesebefehl „Get“ aus.

1. TiXML „Get“ Befehl zum Auslesen der **Variablengruppe** (z.B. [`<Get _="/Process/Aux1/D2"/>`]):
Parametereinträge mit ungültigen Werten werden in der Antwort **nicht aufgeführt**.
2. TiXML „Get“ Befehl zum Auslesen **einer Variablen**:
(z.B. [`<Get _="/Process/Aux1/D2/Sprache"/>`])

In der Antwort wird ein TiXML Fehler ausgegeben (siehe TiXML Error Frame in TiXML Reference Manual) :
ErrorText: „Variable exists but does not contain data“
ErrNo: -2194

Hinweis:

Beide Befehle liefern den zuletzt fehlerfrei gelesenen Wert anstelle einer Fehleranzeige, wenn mindestens ein Zugriff nach dem Upload der Variablenkonfiguration oder einem Systemstart (Reset, Power On) fehlerfrei war.

Für das direkte Auslesen des Fehlerzustandes eines Parameters oder von Parametergruppen wird der TiXML „Get“ Befehl (siehe TiXML Reference Manual) um ein XML Attribut erweitert, das den „Get“ Befehl veranlasst anstelle des Parameterwertes den Wert einer Zusatzinformation zu einem Parameter auszugeben in diesem Falle den Fehlerzustand :

Befehl:

```
[<Get _="VPath" AddInfo="AddInfo" />]
```

Übersicht der möglichen Parameter (kursiv geschriebene Werte):

vpath: Pfad zur Adressierung des Parameters.

AddInfo:

ErrorGibt Fehlerzustand des Wertes zurück.

Das Attribut „AddInfo“ wird bei Werten, die keine Parameter von externen Geräten sind ignoriert. Und der Wert des Parameters zurückgegeben.

Antwort:

AddInfo = Error

```
[<Get _="ErrorClass,ErrorValue" />]
```

ErrorClass: Fehler Klasse
0... Kein Fehler
>0... Fehler

ErrorValue: Fehlerwert

ErrorClass	ErrorValue	Bedeutung
0	0	Kein Fehler
1	$n > 0$	Fehler in der Zugriffsimplementation des TAM. Die Nummer n ist > 0 und abhängig vom jeweiligen Gerät bzw. Bussystem.
$c > 1$	n	Fehlermeldung des angeschlossenen Gerätes bzw. des Busprotokolls. Die Nummern n und c sind abhängig vom jeweiligen Gerät bzw. Bussystem.

Hinweis:

Der Fehlerzustand wird durch einen nachfolgenden fehlerfreien Zugriff auf die SPS bzw. auf den Bus wieder gelöscht (d.h. auf 0,0 gesetzt).

2.5.2 Fehlerzustand verarbeiten

Um den Fehlerzustand zur Erzeugung von Alarmen (oder allgemein Ereignissen) zu nutzen wird ein spezieller Ladebefehl in der Instruction List des „<Value/>“ Eintrags bei der Definition einer Prozessvariablen (siehe TiXML Reference Manual) definiert:

Befehl:

<LDS _=*Vpath*“ AddInfo=**“AddInfo”** />

Übersicht der möglichen Parameter (kursiv geschriebene Werte):

Beschreibung:

Liest den Fehlerzustand des Parameters aus, der in *Vpath* referenziert wird und schreibt in den Verarbeitungsstack folgende Werte:

High-Teil (Bit 16 - 31)	Low Teil (Bit 0 - 15)
Wert der Fehlerklasse	Wert des Fehlers

Vpath: Pfad zur Adressierung des Parameters.

AddInfo:

ErrorCodeGibt Fehlerzustand des Wertes zurück.

3 Unterstützte SPS-Systeme

3.1 Mitsubishi Alpha XL

Die zu überwachenden Variablen der angeschlossenen Mitsubishi Alpha XL müssen im Tixi Alarm Modem definiert sein.

Die Mitsubishi Variablen sind in der External-Gruppe der 'PROCCFG' Datenbank gespeichert:

```
<External>
  <Bus _="C1" protocol="Mitsubishi,Alpha2" type="Master"
    baud="9600">
    <Device _="0" Pollrate="1s">
      <Input1 _="I" ind="1"/>
      <ExtInput129 _="EI" ind="129" acc="R"/>
      <M1 _="M" ind="1" acc="R"/>
      <Keypad1 _="K" ind="1" acc="R" />
      <Display _="N" ind="3" acc="R" />
      <InAnalog7 _="AI" ind="7" acc="R" />
      <Counter1 _="CB" ind="1" acc="R" />
      <CW2 _="CW" ind="2" />
      <Out2 _="O" ind="2" acc="W"/>
      <ExtOut129 _="EO" ind="129" acc="W"/>
    </Device>
  </Bus>
</External>
```

Verwende COM-Port auf der Erweiterungskarte C1

Master-Kommunikation

Variablenliste

Der BUS-Parameter enthält die Adresse der Erweiterungskarte, den Protokoll-Hersteller "Mitsubishi", den Typ der angeschlossenen Steuerung "Alpha2", den Kommunikationsmodus "Master" und die verwendete Baudrate.

Die Device-ID (Stationskennung) muss mit der SPS-Adresse übereinstimmen (Standard 0).

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Es kann eine Liste von Variablen definiert werden: Variablentyp in der Alpha-XL (I,O,AI...)

```
<Alarm11 _="I" ind="5" acc="R"/>
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Mitsubishi SPS (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Mitsubishi Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für Alpha XL:

Typ	index	access	Kommentar
M	1-14	R	Systembits
I	1-15	R	Digitale Eingänge
EI	129-132	R	Eingänge Erweiterungsmodul
O	1-9	RW	Ausgänge
EO	129-132	RW	Ausgänge Erweiterungsmodul
K	1-8	R	BTasten
E	1-4	R	Link Eingänge
A	1-4	RW	Link Ausgänge
N	1-4	RW	Kontrollbits
AI	1-8	R	Analogeingang
CB	1-100	RW	Bitoperanden
CW	1-100	RW	Wordoperanden

Wenn nur Lesezugriff ‚R‘ möglich ist, kann der Parameter ‚acc‘ weggelassen werden. Wenn die SPS im „RUN“ ist, können einige Variablen durch das SPS-Programm überschrieben werden. In diesem Fall müssten Merker vorgeschaltet werden.

Hinweis: Wenn der automatisch generierte DeviceState (siehe Kapitel 6.3) „1“ ist, aber keine Variablenwerte empfangen werden, so existieren im Alarm Modem oder der Alpha XL Bit- oder Wordoperanden, welche auf einer der beiden Seiten nicht definiert sind. Diese Variablen müssen immer auf beiden Geräten vorhanden sein, ansonsten antwortet die Alpha XL nicht.

Eine Alpha XL anschließen

Das Tixi Alarm Modem muss an die Alpha XL über ein Mitsubishi GSM-CAB verbunden werden.

Beachten Sie folgende Hinweise:

1. In der Alpha XL muss ein Program mit aktivierter „serieller Kommunikation“ auf 9600/8N1 vorhanden sein. (siehe Alpha Programming Software online Hilfe). Nach dem Aktivieren muss die Alpha neu gestartet werden.
2. Das GSM-CAB kann direkt an die Mainboard RS232 Schnittstelle (MB) angeschlossen werden.
3. Wenn Sie das GSM-CAB an eine Tixi Alarm Modem Erweiterungskarte RS232-2 (C1) anschließen, müssen Sie ein Nullmodem-Kabel zwischen Alarm Modem und GSM-CAB verwenden.

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender „TransMode“-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

Alpha XL:

```
[<TransMode baud="9600" format="8N1" handshake="noDTR" com="MB"/>]  
(verwenden Sie com="C1", wenn die Alpha XL an der Erweiterungskarte RS232-2  
angeschlossen ist)
```

3.2 Mitsubishi MELSEC FX

Die zu überwachenden Variablen der angeschlossenen Mitsubishi FX Steuerung müssen im Tixi Alarm Modem definiert sein.

Die Mitsubishi Variablen sind in der External-Gruppe der 'PROCCFG' Datenbank gespeichert:

Verwende COM-Port auf Erweiterungskarte C1

```
<External>
  <Bus _="C1" protocol="Mitsubishi,Format1" type="Master"
    baud="9600">
    <Device _="0" Pollrate="1s" devType="FX1N">
      <Input1 _="X" ind="1"/>
      <M1 _="M" ind="1" acc="R"/>
      <Timer _="TS" ind="1" acc="R" />
      <Counter _="CS" ind="3" acc="R" />
      <CN2 _="CN" ind="2" />
      <Out2 _="Y" ind="2" acc="W"/>
    </Device>
  </Bus>
</External>
```

Master-Kommunikation

Variablenliste

Der BUS-Parameter enthält die Adresse der Erweiterungskarte, den Protokoll-Hersteller "Mitsubishi", den Protokoll-Typ "Format1", den Kommunikationsmodus "Master" und die verwendete Baudrate.

Die Device-ID (Stationskennung) muss mit der SPS-Adresse übereinstimmen (Standard 0). Das Format1-Protokoll ist Netzwerkfähig (über RS485), wodurch mehrere Devices mit unterschiedlichen IDs existieren können. Der Parameter „devType“ bestimmt den CPU-Typ: FX1S,FX1N oder FX2N.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Es kann eine Liste von Variablen definiert werden: Variablentyp in der FX (X,Y,..)

```
<Alarm11 _="Y" ind="5" acc="R"/>
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Mitsubishi SPS (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Mitsubishi Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für MELSEC FX „Format1“:

Typ	index	access	Kommentar
M	0-8255	RW	Merker
Y	0-377	RW	Ausgänge (oktal)
X	0-377	R	Eingänge (oktal)
S	0-999	RW	Schrittstatus
CS	0-255	RW	Zähler Kontakt
TS	0-255	RW	Timer Kontakt
TN	0-255	RW	Timer Istwert vorzeichenlos
TNI	0-255	RW	Timer Istwert vorzeichenbehaftet
CN	0-199	RW	Zähler Istwert (16Bit)
CD	200-255	RW	Zähler Istwert (32Bit) vorzeichenlos
CDI	200-255	RW	Zähler Istwert (32Bit) vorzeichenbehaftet
D	0-8255	RW	Register (16 Bit) vorzeichenlos
DI	0-8255	RW	Register (16 Bit) vorzeichenbehaftet
DW	0-8254	RW	Register (32 Bit) vorzeichenlos
DWI	0-8254	RW	Register (32 Bit) vorzeichenbehaftet

Wenn nur Lesezugriff ‚R‘ möglich ist, kann der Parameter ‚acc‘ weggelassen werden.
 Wenn die SPS im „RUN“ ist, können einige Variablen durch das SPS-Programm überschrieben werden. In diesem Fall müssten Merker vorgeschaltet werden.

Zusätzlich zu dem offenen Protokoll „Format1“ ist das FX-interne Protokoll in den Alarm Modems integriert. Die Auswahl erfolgt über den Bus-Parameter `protocol="Mitsubishi,FX"`. Dieses Protokoll ist nicht netzwerkfähig, bietet jedoch zusätzlich Zugriff auf die Contacts und Resets der Timer/Counter:

Liste der zusätzlich unterstützten Variablen für MELSEC FX „FX Protocol“:

Typ	index	access	Kommentar
CC	0-255	R	Zähler Spule
TC	0-255	R	Timer Spule
CR	0-255	R	Zähler Reset
TR	0-255	R	Timer Reset

Eine Mitsubishi FX anschließen

Das Tixi Alarm Modem kann an die FX interne RS422 Schnittstelle oder über eine zusätzliche Schnittstellenerweiterung RS232-BD / RS422-BD / RS485-BD angeschlossen werden.

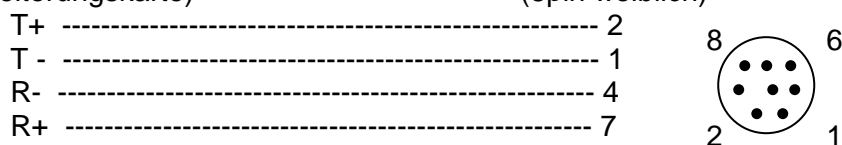
Wenn Sie eine BD-Erweiterung verwenden, muss diese Schnittstelle über die GX Developer Software mit den Parametern 9600/7E1 aktiviert werden.

Beide Schnittstellen können simultan verwendet werden, um an die FX z.B. ein Alarm Modem und ein Display gleichzeitig anzuschließen.

RS422 Anschluß:

Tixi Alarm Modem
 (RS422 Erweiterungskarte)

MELSEC FX Port
 (8pin weiblich)



RS485 Anschluß:

Tixi Alarm Modem

(RS485 Erweiterungskarte)

MELSEC FX RS485-BD

(5pol Schraubklemme)

T+ -----	RDA
T- -----	RDB
R- -----	SDB
R+ -----	SDA
0V -----	SG

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

MELSEC FX:

Tixi Alarm Modem RS232 Schnittstelle:

```
[<TransMode baud="9600" format="7E1" com="MB"/>]
```

(Verwenden Sie com="C1" wenn die SPS an einer RS232 Erweiterungskarte steckt.)

Tixi Alarm Modem RS422 Schnittstelle:

```
[<TransMode baud="9600" format="7E1" handshake="FULL" com="C0"/>]
```

3.3 Siemens Simatic S7-200

Die zu überwachenden Variablen der angeschlossenen Siemens Simatic S7-200 müssen im Tixi Alarm Modem definiert sein.

Die Siemens Variablen sind in der External-Gruppe der 'PROCCFG' Datenbank gespeichert:

```
<External>
  RS485 Schnittstelle auf Erweiterungskarte C0
  <Bus _="C0" protocol="Siemens,S7-200" type="Master"
    baud="9600" handshake="HALF" TS="0" MAXADR="15" GUF="1"
    RC="1">
    <Device _="2" Pollrate="1s">
      <M30 _="M" ind="30" acc="R" />
    </Device>
    <Device _="3" Pollrate="60s">
      <V100 _="V" ind="100" acc="R" />
      <VS50 _="VS" size="4" ind="100" acc="R" />
    </Device>
  </Bus>
</External>
```

Station #2

Station #3

Der BUS-Parameter enthält die Adresse der Erweiterungskarte, den Protokoll-Hersteller "Siemens", den Typ der angeschlossenen Steuerung "S7-200", den Kommunikationsmodus "Master", die verwendete Baudrate und das notwendige Handshake (bei Verwendung der RS485 Erweiterungskarte). TS ist die Stationsnummer des Alarm Modems, MAXADR der Bereich der abzufragenden Stationsnummern, GUF der "gap update factor" um andere Slaves zu erkennen und RC die Anzahl der Wiederholungen bei Kommunikationsfehlern.

Das Tixi Alarm Modem kann gegenüber der S7-200 nur Master sein. Die Standardbaudrate ist 9600.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Für jede Steuerung muss ein 'Device'-Abschnitt eingefügt werden, welcher die Stationsnummer ('_' – Attribut) und den Abfragezyklus enthält. Nach jedem Abfragezyklus werden die Variablenwerte erneut eingelesen.

Es kann eine Liste von Variablen definiert werden:

```
<AlarmM10 _="M" ind="10" acc="R"/>
```

Variablentyp in der S7-200

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Siemens SPS (siehe: Liste der unterstützten Variablen).

Der 'ind' Parameter bestimmt die Adresse der Variable in der Simatic Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Es muss das zusätzliche Zugriffsrecht 'A' angefügt werden, welches den aktiven Zugriff auf die Variablen erlaubt.

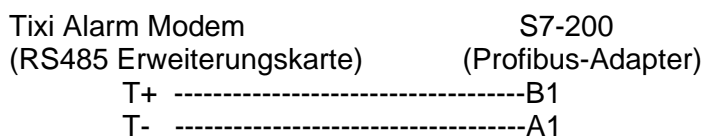
Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für S7-200 CPUs

Typ	index	access	Kommentar
V	0-10239.7	R/W	Variablenspeicher (Bit)
VB	0-10239	R/W	Variablenspeicher (Byte)
VW	0-10238	R/W	Variablenspeicher (Word)
VD	0-10236	R/W	Variablenspeicher (Doppel-Word)
VS	0-10239	R/W	Variablenspeicher (string, erfordert Parameter "size")
I	0-15.7	R	Eingänge
Q	0-15.7	R/W	Ausgänge
M	0-31.7	R/W	Merker (Bit)
MB	0-31	R/W	Merker (Byte)
MW	0-30	R/W	Merker (Word)
MD	0-28	R/W	Merker (Doppel-Word)
SM	0-549.7	R	Sondermerker
S	0-31.7	R/W	Ablaufsteuerungsrelais
T	0-255	R/W	Timer
C	0-255	R/W	Zähler
AI	0-62	R	Analogeingang
AQ	0-62	R/W	Analogausgang
HC	0-5	R	High speed Zähler

Eine S7-200 anschließen

Die S7-200 kann über ein PPI-Kabel (RS232) oder über einen Profibus-Adapter (RS485 Erweiterungskarte) angeschlossen werden.



Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference Manual für weitere Informationen)

Tixi Alarm Modem RS232 Schnittstelle:

```
[<TransMode baud="9600" format="8E1" handshake="noDTR" com="MB" />]
```

(Verwenden Sie com="C1" wenn die SPS an einer RS232 Erweiterungskarte steckt.)

Tixi Alarm Modem RS485 Schnittstelle:

```
[<TransMode baud="9600" format="8E1" handshake="HALF" com="C0" />]
```

Um sich mit einer S7-200 zu verbinden, muss lokal ein 11-Bit modem (8E1 Datenformat) verwendet werden. In der MicroWin Software müssen zudem über einen auf der Tixi CD mitgelieferten Registry-Patch die Timings angepasst werden.

3.4 Siemens Simatic S7-300/400

Die zu überwachenden Variablen der angeschlossenen Siemens Simatic S7-300/400 müssen im Tixi Alarm Modem definiert sein.

Die Siemens Variablen sind in der External-Gruppe der 'PROCCFG' Datenbank gespeichert:

```
<External>
  <Bus Name="Bus1" _="COM2" family="Siemens" protocol="Siemens,S7-300/400-A"
    type="Master" TS="0" MAXADR="15" GUF="1" RC="1">
    <Device _="2" Name="Device_2" Pollrate="2s">
      < AlarmS7300 _="M" ind="0.0" acc="RA" def="0" format="?Alarm,OK"/>
      < Confirmation _="M" ind="0.1" acc="RWA" def="1" />
      < I124 _="VM" ind="0" db="15" acc="RA" def="0" />
      < Test _="M" ind="0.3" acc="RA" def="0" format="?Alarm,OK"/>
    </Device>
    <Device _="3" Name="Device_3" Pollrate="60s">
      < AlarmS7300 _="M" ind="0.0" acc="RA" def="0" format="?Alarm,OK"/>
      < Confirmation _="M" ind="0.1" acc="RWA" def="1" />
    </Device>
  </Bus>
</External>
```

Der BUS-Parameter enthält die Adresse der Erweiterungskarte, den Protokoll-Hersteller "Siemens", den Typ der angeschlossenen Steuerung "S7-300/400-A" und den Kommunikationsmodus "Master" oder „Slave“. TS ist die Stationsnummer des Alarm Modems, MAXADR der Bereich der abzufragenden Stationsnummern, GUF der "gap update factor" um andere Slaves zu erkennen und RC die Anzahl der Wiederholungen bei Kommunikationsfehlern.

Beim Laden einer External-Definition werden die von der Siemens Teleservice Software im TS-Adapter hinterlegten Parameter deaktiviert.

Das Tixi Alarm Modem kann gegenüber der S7-300/400 Master oder Slave sein.

Für jede Steuerung muss ein 'Device'-Abschnitt eingefügt werden, welcher die Stationsnummer ('_' – Attribut) und den Abfragezyklus enthält. Nach jedem Abfragezyklus werden die Variablenwerte erneut eingelesen (Master) oder ein Kommunikationstimeout erkannt (Slave).

Es kann eine Liste von Variablen definiert werden: Variablentyp in der S7-300

```
<AlarmM10 _="DBX" db="2" ind="0.0" acc="R"/>
```

Datenbaustein der Variablen

Jede Zeile definiert einen logischen Namen (Alias, z.B. AlarmM10) und den Typ der Variable in der Siemens SPS (siehe: Liste der unterstützten Variablen).

Der 'ind' Parameter bestimmt die Adresse der Variable in der Simatic Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'db' Parameter bestimmt die Nummer des Datenblocks in dem sich die Variable befindet. Der Wert kann zwischen 1 und 65535 liegen.

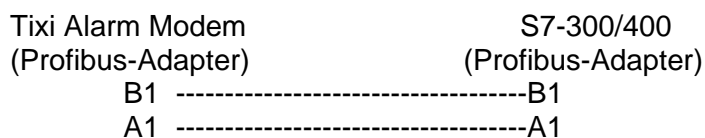
Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für S7-300/400 CPUs

Typ	index	access	Kommentar
DBX	0.0-65535.7	R/W	Datenbaustein (Bit, erfordert Parameter "db")
DBB	0-65535	R/W	Datenbaustein (Byte, erfordert Parameter "db")
DBW	0-65534	R/W	Datenbaustein (Word, erfordert Parameter "db")
DBD	0-65532	R/W	Datenbaustein (DoubleWord, erfordert Parameter "db")
DBS	0-65535	R/W	Datenbaustein (string, erfordert Parameter "size" und "db")
I	0-16384.7	R	Eingänge
Q	0-16384.7	R/W	Ausgänge
M	0-65535.7	R/W	Merker Bit
MB	0-65535	RW	Merker Byte
MW	0-65534	RW	Merker Wort
MD	0-65532	RW	Merker DWort
T	0-2074	R/W	Timer
C	0-2074	R/W	Zähler

Eine S7-300/400 anschließen

Die S7-300/400 wird über einen Profibus-Adapter an das Tixi Alarm Modem MPI angeschlossen werden.



Fernzugriff

Der Fernzugriff erfolgt völlig transparent über die TS-Adapter Funktion.

Folgende Bedingungen gelten für den TS-Adapter-Zugriffschutz:

1. Der lokale Zugriff auf die S7 durch das Modem (COM1->COM2) lässt sich nicht sperren. (Verhält sich also wie der Original Siemens-TS-Adapter).
2. Die über die SIMATIC-Software eingestellten TS-Benutzer (Menü "TS-Adapter-Parametrieren") funktionieren wie beim Original Siemens-TS-Adapter, d.h. es lässt sich ein ADMIN und 2 User anlegen, sowie eine CallBack-Funktion für die Benutzer.
3. Die TS-Adapter-Einstellungen werden beim Factory-Reset gelöscht.
4. Sobald im Modem eine S7 Anbindung (External) UND (!) AccRights User (siehe TiXML-Reference-Manual) definiert wurden, gelten nurnoch die in den AccRights

angelegten User mit "TSAdapter" Dienst. Bei diesen läßt sich auch ein Callback über das gleichnamige Userattribut anlegen. Die im TS-Adapter parametrisierten User bleiben erhalten, sind aber inaktiv. Erst nach Löschen der AccRights werden diese wieder wirksam.

Weiterer Vorteil: In den AccRights lassen sich quasi beliebig viele User anlegen, eine beschränkung auf 3 User (wie im TS-Adapter) gibt es hier nicht.

Beispiel TS-Adapter AccRights:

```
[<SetConfig _="USER" ver="y">
  <AccRights>
    <Groups>
      <Hausmeister>
        <TSAdapter AccLevel="1"/>
      </Hausmeister>
    </Groups>
    <User>
      <Krause Plain="Dackel" Group="Hausmeister" Callback="123"/>
    </User>
  </AccRights>
</SetConfig>]
```

3.5 VIPA CPU 100/200/300

Die zu überwachenden Variablen der angeschlossenen VIPA CPU müssen im Tixi Alarm Modem definiert sein.

Die VIPA-CPU's können natürlich auch an ein Alarm Modem mit MPI Schnittstelle angeschlossen werden. Der Funktionsumfang ist identisch (siehe 3.4)

Die VIPA Variablen sind in der External-Gruppe der '**PROCCFG**' Datenbank gespeichert:

```
<External>
  <Bus _="C1" protocol="Vipa,GreenCable" type="Master"
    baud="38400" TS="1">
    <Device _="2" Pollrate="1s">
      <M30 _="M" ind="30" acc="R" />
    </Device>
  </Bus>
</External>
```

Verwende COM-Port auf Erweiterungskarte C1

Station #2

Der BUS-Parameter enthält die Adresse der Erweiterungskarte "C1", den Protokoll-Hersteller "Vipa", das Protokoll "GreenCable", den Kommunikationsmodus "Master", die Baudrate "38400" und den Parameter TS, welcher die Stationsnummer des Tixi Alarm Modems enthält.

Das Tixi Alarm Modem ist gegenüber der VIPA immer "Master". Die Device-ID (Stationskennung) muss mit der SPS-Adresse übereinstimmen.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Für jede Steuerung muss ein 'Device'-Abschnitt eingefügt werden, welcher die Stationsnummer ('_' – Attribut) und den Abfragezyklus enthält. Nach jedem Abfragezyklus werden die Variablenwerte erneut eingelesen (Master) oder ein Kommunikationstimeout erkannt (Slave).

Es kann eine Liste von Variablen definiert werden: Variablentyp in der VIPA

```
<AlarmM10 _="M" ind="10" acc="R"/>
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. AlarmM10) und den Typ der Variable in der VIPA CPU (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Simatic Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für VIPA System 200V CPU

Typ	index	access	Kommentar
M	0-2047.7	R/W	Merker (Bit)
MB	0-2047	R/W	Merker (Byte)
MW	0-2046	R/W	Merker (Word)
MD	0-2044	R/W	Merker (Doppel-Word)
I	0-15.7	R	Eingänge
Q	0-15.7	R/W	Ausgänge

Das Tixi Alarm Modem unterstützt auch VIPA System 100V CPU und VIPA System 300V CPU.

Bei Alarm Modems der Aluline Reihe wird die VIPA nur bis Firmware Version 2.0 unterstützt.

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

```
[<TransMode baud="38400" format="801" com="MB"/>]
(Verwenden Sie com="C1" wenn die VIPA an der RS232 Erweiterungskarte angeschlossen ist)
```

Um eine Verbindung zu einer VIPA CPU herzustellen, ist lokal ein 11-Bit modem (Unterstützung für 801 Datenformat) zu verwenden.

3.6 Moeller Easy 400 / 500 / 600 / 700 / 800 / MFD

Die zu überwachenden Variablen der angeschlossenen Moeller Easy (z.B. easy 400/600/800/MFD) müssen im Tixi Alarm Modem definiert sein.

Die Moeller Variablen sind in der External-Gruppe der **'PROCCFG'** Datenbank gespeichert:

```
<External>
  Verwende COM-Port auf Erweiterungskarte C1
  <Bus _="C1" protocol="Moeller,Easy 800" type="Master"
    baud="9600" handshake="noDTR">
    <Device _="0" Pollrate="1s">
      <Input1 _="I" ind="1"/>
      <UIn _="AI" ind="7" acc="R" format="F.1;°C" def="1"/>
      <Out4 _="M" ind="2" acc="RW"/>
      <OutClock _="OU" ind="0" acc="R" />
      <OutCount _="OC" ind="0" acc="R" />
      <OutAnalog _="OA" ind="0" acc="R" />
      <OutTimer _="OT" ind="0" acc="R" />
      <Counter _="IC" ind="0" acc="R" />
      <TextMarker _="D" ind="0" />
      <Timer _="IT" ind="0" />
      <Out2 _="Q" ind="2" acc="W"/>
      <Flag _="M" ind="8"/>
      <Clock _="U" ind="0" acc="RW"/>
    </Device>
  </Bus>
</External>
```

Easy 800 Master

Variablenliste

Der BUS-Parameter enthält die Adresse der Erweiterungskarte "C1", den Protokoll-Hersteller "Moeller", das angeschlossene Gerät "Easy 400/600", "Easy 500/700" oder "Easy 800", den Kommunikationsmodus "Master", die Baudrate und das notwendige „noDTR“ Handshake.

Für Easy 400//600 wählen Sie "Moeller, Easy 400/600", Baudrate: 4800.

Für Easy 500//700 wählen Sie "Moeller, Easy 500/700", Baudrate: 4800.

Für Easy 800/MFD wählen Sie "Moeller, Easy 800", Baudrate: 9600 oder 19200.

Easy 400//600: Der 'Device' Eintrag muss "1" sein und den Abfragezyklus enthalten.

Easy 500/700/800/MFD: Der 'Device' Eintrag muss die Stationskennung der angeschlossenen Easy enthalten. Bei einer einzelnen Easy 500/700/800/MFD ist dies immer "0". Die Easy 800/MFD unterstützt mehrere Stationen (1-8).

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Es kann eine Liste von Variablen definiert werden:

Variablentyp in der Easy

```
<Alarm11 _="I" ind="16" acc="R"/>
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Moeller Easy (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Moeller Easy und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der Easy erhalten hat.

Liste der unterstützten Variablen für Easy 400/600:

Typ	index	access	Kommentar
IK	1-16	R	Eingänge Erweiterung
I	1-8	R	Eingänge
OU	1-4	R	Zeitschaltuhr Ausgang
OC	1-8	R	Zähler Ausgang
OA	1-8	R	Analogwertvergleichler Ausgang
OT	1-8	R	Timer Ausgang
IC	1-8	R	Zähler Istwert
D	1-8	RW	Textanzeige
IT	1-8	R	Timer Istwert
Q	1-8	R(W)	Ausgänge (Schreiben nur im STOP)
M	1-16	RW	Merker
U	-	RW	Uhr
ST	1-8	R	Timer Sollwert
SC	1-8	R	Counter Sollwert
AW	1-8	R	Analogwertvergleichler Sollwert
QK	1-8	R(W)	Ausgänge Erweiterung (Schreiben nur im STOP)
Key	1-4	R	Testen
AI	7,8	R	Analogeingang

Liste der unterstützten Variablen für Easy 500/700:

Typ	index	access	Kommentar
I	1-16	R	Eingänge
Q	1-8	R	Ausgänge
R	1-16	R	Eingang EASY-LINK
S	1-8	R	Ausgang EASY-LINK
OU	1-8	R	Zeitschaltuhr Ausgang, wie OUW (Easy 400/600 Kompatibilität)
OUW	1-8	R	Wochenzeitschaltuhr Ausgang
OUY	1-8	R	Jahreszeitschaltuhr Ausgang
OC	1-16	R	Zähler Ausgang
OA	1-16	R	Analogwertvergleichler Ausgang
OT	1-16	R	Timer Ausgang
IC	1-16	RW	Zähler Istwert
D	1-16	R	Textanzeige
IT	1-16	RW	Timer Istwert
IW	1-4	RW	Betriebsstundenzähler Istwert
SW	1-4	RW	Betriebsstundenzähler Sollwert
N	1-16	RW	Merker N
M	1-16	RW	Merker M
U	-	RW	Uhr
ST	1-16	RW	Timer Sollwert 1
ST2	1-16	RW	Timer Sollwert 2
SC	1-16	RW	Zähler Sollwert
AW	1-16	RW	Analogwertvergleichler Sollwert, wie AWI2 (Easy 400/600 Kompatibilität)
AWI1	1-16	RW	Analogwertvergleichler Sollwert 1
AWI2	1-16	RW	Analogwertvergleichler Sollwert 2
AWF1	1-16	RW	Analogwertvergleichler Verstärker
AWF2	1-16	RW	Analogwertvergleichler Verstärker 2
AWOS	1-16	RW	Analogwertvergleichler Offset
AWHY	1-16	RW	Analogwertvergleichler Hysterese

Key	1-4	R	Tasten
AI	7,8,11,12	R	Analogeingang

Liste der unterstützten Variablen für Easy 800 / MFD:

Typ	index	access	Kommentar
I	1-16	R	Eingänge
Q	1-8	RW	Ausgänge (Schreiben nur im STOP)
R	1-16	R	Eingänge Erweiterung
S	1-8	RW	Ausgänge Erweiterung (Schreiben nur im STOP)
M	1-96	RW	Merker
MB	1-96	RW	Byte Merker
MW	1-96	RW	Word Merker
MD	1-96	RW	Doppelword Merker
Key	1-4	R	Tasten
IA	1-4	R	Analogeingänge
QA	-	RW	Analogausgänge
U	-	RW	Uhr

Wenn für eine Variable nur eine Adresse existiert, z.B. "U" ,kann der Parameter ,ind' weggelassen werden.

Wenn nur Lesezugriff ,R' möglich ist, kann der Parameter ,acc' weggelassen werden.

Wenn die SPS im "RUN" ist, können einige Variablen durch das Easy-Programm überschrieben werden. In diesem Fall müssten Merker vorgeschaltet werden.

Für jedes parametrisierte Gerät wird automatisch eine **Type** Variablen eingefügt:

```
[<Get _="/Process/Aux?/D?/Type"/>]
```

Diese Variable enthält den Typ der angeschlossenen Easy, z.B. "412-DC-Rx".

Fernzugriff

Für den Fernzugriff auf die Easy ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

Easy 400/500/600/700

```
[<TransMode baud="4800" format="8N1" handshake="noDTR" com="MB"/>]
```

(Verwenden Sie com="C1" wenn die SPS an einer RS232 Erweiterungskarte steckt.

Easy 800/MFD

```
[<TransMode baud="9600" format="8N1" handshake="noDTR" com="MB"/>]
```

(Verwenden Sie com="C1" wenn die SPS an einer RS232 Erweiterungskarte steckt.

Um sich via GSM mit einer Easy zu verbinden, ist mindestens Easy-Soft 5.01 notwendig (Wartezeit einstellbar).

3.7 Moeller PS30 & PS4/40

Die zu überwachenden Variablen der angeschlossenen PS30 oder PS4/40 müssen im Tixi Alarm Modem definiert sein.

Die Moeller Variablen sind in der External-Gruppe der 'PROCCFG' Datenbank gespeichert:

```

<External>
  <Bus _="C1" protocol="Moeller,SucomA" type="Master"
    baud="9600">
    <Device _="7" Pollrate="1s">
      <Value1 _="S" ind="1" acc="R"/>
      <Alarmflag _="F" ind="7.0" acc="R"/>
      <Temp _="B" ind="2" acc="RW"/>
      <Power _="R" ind="127" acc="RW" />
      <CountVal _="D" ind="3212" acc="RW" />
    </Device>
  </Bus>
</External>

```

Verwende COM-Port auf Erweiterungskarte C1

SucomA Master

Variablenliste

Der BUS-Parameter enthält die Adresse der Erweiterungskarte "C1", den Protokoll-Hersteller "Moeller", das Protokoll "SucomA", den Kommunikationsmodus "Master" und die Baudrate (4800 bis 57600).

Die Stationsnummer muss "7" sein.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Es kann eine Liste von Variablen definiert werden: Variablentyp in der PS30 & PS4/40

```

<Alarm11 _="F" ind="16" acc="R"/>

```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Moeller SPS (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Moeller SPS und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für PS30 & PS4/40:

Typ	index	access	Kommentar
S	0-65535	R	Status WORD
F	0-14999.7	R	Merker BIT
B	0-14999	RW	Merker BYTE
R	0-14998	RW	Merker WORD
D	0-14996	RW	Merker DWORD

Wenn nur Lesezugriff ,R' möglich ist, kann der Parameter ,acc' weggelassen werden. Wenn die SPS im "RUN" ist, können einige Variablen durch das SPS-Programm überschrieben werden. In diesem Fall müssten Merker vorgeschaltet werden.

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

```
[<TransMode baud="9600" format="8N1" com="MB"/>]
```

(Verwende com="C1" wenn die SPS an der RS232 Erweiterung angeschlossen ist)

3.8 SAIA Burgess S-Bus

Die zu überwachenden Variablen der angeschlossenen SAIA-Steuerung (z.B. PCD2) müssen im Tixi Alarm Modem definiert sein.

Die SAIA Variablen sind in der External-Gruppe der '**PROCCFG**' Datenbank gespeichert:

```
<External>
  <Bus _="C0" protocol="Saia,SBus-DataMode" type="Master"
    baud="19200" handshake="HALF">
    <Device _="1" Pollrate="1s">
      <F100 _="F" ind="100" acc="R" />
    </Device>
    <Device _="3" Pollrate="60s">
      <R102 _="R" ind="102" acc="R" />
    </Device>
  </Bus>
</External>
```

Verwende RS485-Port auf Erweiterungskarte

aktiviert RS485 Modus

S-Bus Station #1

S-Bus Station #3

Das Tixi Alarm Modem kann als S-BUS Master oder Slave parametrierbar werden. Alle S-BUS Baudraten werden unterstützt, die Standardbaudrate (kein Eintrag) ist 19200.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Der Parameter handshake="HALF" aktiviert den RS485 Modus auf speziellen Erweiterungskarten.

Für jede S-Bus Station muss ein 'Device'-Eintrag erstellt werden, welcher die Stationsnummer der angeschlossenen Slave-PCD2s bzw. die eigene Stationsnummer als S-BUS-Slave enthält ('_' - Attribute), sowie den Abfragezyklus.

Nach Ablauf des Abfragezyklus werden die Variablenwerte neu eingelesen (Master) oder ein Kommunikationstimeout erkannt (Slave).

Es kann eine Liste von Variablen definiert werden: Variabletyp im S-BUS (F,T,C,I,O,R)

```
<Alarm11 _="F" ind="11" acc="R"/>
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable auf dem S-BUS (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der PCD2 und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Ein zusätzliches Zugriffsrecht ist 'RWL', welches einen gemeinsamen Lese-und Schreibspeicher aktiviert, welcher jedoch nur im Slave-Modus verwendet werden darf. Mit dem 'RW' Zugriffsrecht hat das Tixi Alarm Modem einen getrennten Speicher für Schreib- und Lesezugriffe. In diesem Fall kann die SPS einen Variablenwert in den Lesespeicher des Tixi Alarm Modems schreiben und das Tixi Alarm Modem schreibt in den Schreibspeicher

der gleichen Variable. Dadurch kann es zu der Situation kommen, daß die SPS z.B. eine 1 schreibt, beim nachfolgenden Auslesen jedoch eine 0 erhält.

Mit 'RWL' verwendet das Tixi Alarm Modem für beide Zugriffe den gleichen Speicher, d.h. es gilt immer der zuletzt geschriebene Wert, egal von wem er gesetzt wurde.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für PCD2

Typ	index	access	Kommentar
F	0-8191	R/W	Flag
R	0-4095	R/W	Register
T	0-1600	R	Timer
C	0-1600	R	Counter
O	0-256	R/W	Ausgang
I	0-256	R	Eingang (nur als Master)

PCD2 anschließen

Das Tixi Alarm Modem kann an der PCD2 an allen 3 seriellen Schnittstellen S0-S2 angeschlossen werden. Es ist lediglich eine 3-Draht-Leitung (RX,TX,GND) notwendig.

Beachten Sie folgende Hinweise:

1. Wenn Sie das Tixi Alarm Modem an den PGU-Port (S0) der PCD2 anschließen, darf die DSR-Leitung nicht mitgeführt werden, da die PCD2 sonst den S-BUS deaktiviert.
2. Wenn das Tixi Alarm Modem mit der Mainboard RS232 (MB) an der PCD2 angeschlossen wird, darf die DTR-Leitung nicht mitgeführt werden, da der S-BUS sonst deaktiviert wird.

Tixi-MB	PCD2	Tixi-Cx	PCD2	Tixi-RS485	PCD2
2-RxD-----	RxD-12/32	2-RxD-----	TxD-12/32	1-GND-----	GND-10/35
3-TxD-----	TxD-11/31	3-TxD-----	RxD-11/31	4-R(-)-----	RX-TX-11/36
5-GND-----	GND-10/30	5-GND-----	GND-10/30	5-R(+)-	/RX-/TX-12/37

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

Tixi Alarm Modem RS232 Schnittstelle:

```
[<TransMode baud="9600" format="8N1" com="MB"/>]
```

(Verwenden Sie com="C1" mit einer RS232 Erweiterungskarte)

Tixi Alarm Modem RS485 Schnittstelle:

```
[<TransMode baud="9600" format="8N1" handshake="HALF" com="C0"/>]
```

Wählen Sie die gleiche Baudrate wie die SPS.

3.9 Carel Macroplus

Die zu überwachenden Variablen der angeschlossenen Carel-Steuerung (z.B. Macroplus) müssen im Tixi Alarm Modem definiert sein.

Die Carel Variablen sind in der External-Gruppe der 'PROCCFG' Datenbank gespeichert:

```

<External>
  <Bus _="C0" protocol="Carel,PC2" type="Master"
    handshake="FULL">
    <Device _="1" Pollrate="1s">
      <Alarm11 _="D" ind="22" acc="RW" />
    </Device>
    <Device _="3" Pollrate="60s">
      <Alarm31 _="D" ind="22" acc="RW" />
    </Device>
  </Bus>
</ External >

```

Verwende RS422-Port der Erweiterungskarte C0
Aktiviert RS422-Modus
CarelBus Steuerung #1
CarelBus Steuerung #3

Der Parameter `handshake="FULL"` aktiviert den RS422 Modus. Ohne diesen Eintrag wird RS232 Kommunikation verwendet.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Für jede auf dem Carel-Bus befindliche Steuerung muss ein 'Device' Eintrag eingefügt werden, welcher die Stationsnummer auf dem Carel-Bus ('_' – Attribut) sowie den Abfragezyklus enthält. Das Tixi Alarm Modem fragt die Steuerung nach geänderten Variable ab solange es keinen NULL-Frame erhält, der dem Modem mitteilt, daß keine weiteren Änderungen vorhanden sind.

Es kann eine Liste von Variablen definiert werden:

```
<Alarm11 _="D" ind="22" acc="RW" />
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Carel-Steuerung (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Carel Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für Macroplus

Typ	index	access	Kommentar
D	1-183	R/W	Bits
I	1-50	R/W	Integer
A	1-50	R/W	Analog

Macroplus anschließen

Die Macroplus kann über einen RS422-RS232 Adapter oder direkt an eine RS422 Erweiterungskarte angeschlossen werden.

Tixi Alarm Modem (RS422 Erweiterungskarte)	Macroplus (9pin RS422 weiblich)
T+ -----	R+ (4)
T- -----	R- (5)
R- -----	T- (1)
R+ -----	T+ (2)

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

Tixi Alarm Modem RS232 Schnittstelle:

```
[<TransMode baud="1200" format="8N2" com="MB"/>]
```

(Verwenden Sie com="C1" mit einer RS232 Erweiterung)

Tixi Alarm Modem RS422 Schnittstelle:

```
[<TransMode baud="1200" format="8N2" handshake="FULL" com="C0"/>]
```

3.10 ABB AC010, AC031, CL Reihe

Der AC010 Logikcontroller ist eine OEM-Version der Moeller EASY 400/600 Produkte. Siehe Kapitel 3.6 für Details.

Die AC031 wird über das MODBUS RTU Protokoll unterstützt. Siehe Kapitel 4.3 für Details.

Die CL Reihe ist eine OEM-Version der Moeller EASY 500/700 Produkte. Siehe Kapitel 3.6 für Details.

3.11 Allen Bradley Pico

Diese Serie intelligenter Kleinststeuerrelais ist eine OEM-Version der Moeller Kleinststeuerungen.

Pico Serie A = Moeller Easy 400/600

Pico Serie B = Moeller Easy 500/700

Pico GFX = Moeller MFD

Siehe Kapitel 3.6 für Details.

3.12 Theben PHARAO II

Die PHARAO II Kleinststeuerung ist eine OEM-Version der Mitsubishi Alpha2 (XL).

Siehe Kapitel 3.1 für Details.

4 Feldbus Unterstützung

4.1 Tixi-Bus

Tixi-Bus ist ein einfaches Feldbus Protokoll um Variablen effektiv auch mit mehreren SPS-Systemen auszutauschen.

(Schauen Sie auf der Tixi.Com Webseite nach weiteren Informationen, oder senden Sie eine E-Mail an Developer@Tixi.Com)

Die Tixi-Bus Variablen werden in der External Gruppe der 'PROCCFG' Datenbank definiert.

```
<External> _____ Verwende RS422-Schnittstelle der Erweiterungskarte C0
  <Bus _="C0" protocol="Tixi.Com,Tixi-Bus" type="Master"
    handshake="none" TS="0">
    <Device _="1" Pollrate="1s"> _____ SPS #1
      <Alarm11 _="F" ind="22" acc="RW"/>
    </Device>
    <Device _="3" Pollrate="60s"> _____ SPS #3
      <Alarm31 _="R" ind="243" acc="RW"/>
      <Array _="B" ind="1" no="8" acc="RW"/>
    </Device>
  </Bus>
</External>
```

Das Modem kann nur als Tixi-Bus "Master" agieren.

Der Parameter `handshake="mode"` aktiviert verschiedene Kommunikationsmodi:

Bei einer RS232 Schnittstelle ist keine Angabe notwendig.

Mit `handshake="HALF"` wird RS485-Kommunikation, und mit "FULL" wird RS 422 Kommunikation verwendet. Für RS 485/422 ist eine spezielle Erweiterungskarte notwendig.

TS ist die Stationsnummer des Alarm Modems.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Für jede TixiBus Station muss ein 'Device'-Eintrag erstellt werden, welcher die Stationsnummer der angeschlossenen Slave-SPS-Systeme sowie den Abfragezyklus enthält. Nach Ablauf des Abfragezyklus werden die Variablenwerte neu eingelesen.

Es kann eine Liste von Variablen definiert werden:

```
<Alarm11 _="F" ind="22" acc="RW"/> _____ Variablentyp im TixiBus Protokoll (F,W,...)
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Steuerung (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter bestimmt die Adresse der Variable in der Steuerung und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Bei Tixi-Bus können Variablen als Arrays über das Attribut „no“ angelegt werden. Siehe dazu auch Kapitel 2.4.

Liste der unterstützten Variablen für Tixi-Bus

Typ	index	access	Kommentar
C	0-65535	R/W	Flag (Coil)
W	0-65535	R/W	Word
DW	0-65535	R/W	DWord
F	0-65535	R/W	Float
DF	0-65535	R/W	Double
B	0-65535	R/W	Byte
S	0-65535	R/W	String (erfordert Angabe von "size")

4.2 ASCII Protocol

Das ASCII protocol ist eine einfache Möglichkeit mit dem Tixi Alarm Modem Werte aus Geräten auszulesen, die Daten über ein Textprotokoll zu Verfügung stellen.

Das ASCII Protokoll wird in der External Gruppe der 'PROCCFG' Datenbank definiert.

```

<External> _____ Verwende RS422-Schnittstelle der Erweiterungskarte C0
  <Bus _="C0" protocol="Tixi.Com,ASCII" type="Master"
    handshake="none" baud="115200" format="8N1">
      <Device _="1" Pollrate="1s">
        <Float _="DF" Pos="14" End="24" acc="R" Radix="K"
          Request="&#13;" ResEnd="Ende&#13;" ResTime="10s"/>
        <Flag _="C" Pos="0" acc="R" />
        <Word _="W" Pos="3" End="5" acc="R" />
        <BinWord _="W" Pos="7" Radix="B" acc="R" />
        <String _="S" Size="30" Pos="3" Width="10" acc="R" />
      </Device>
    </Bus>
  </External>

```

Variablen des abzufragenden Gerätes

Der Parameter handshake="mode" aktiviert verschiedene Kommunikationsmodi:

Bei einer RS232 Schnittstelle ist keine Angabe notwendig.

Mit handshake="HALF" wird RS485-Kommunikation, und mit "FULL" wird RS 422 Kommunikation verwendet. Für RS 485/422 ist eine spezielle Erweiterungskarte notwendig.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die ASCII-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Für das angeschlossene Gerät muss ein 'Device'-Eintrag erstellt werden, welcher die Stationsnummer sowie den Abfragezyklus enthält. Nach Ablauf des Abfragezyklus werden die Variablenwerte neu eingelesen.

Es kann eine Liste von Variablen definiert werden:

```

<Float _="DF" Pos="14" End="24" acc="R" Radix="K"
Request="&#13;" ResEnd="Ende&#13;" ResTime="10s"/>

```

Variablentyp im ASCII Protokoll (C,W,...)

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11), den Typ der Variable im Gerät (siehe: Liste der unterstützten Variablen) und der 'acc' Parameter das Zugriffsrecht 'R' für "Lesen". Die weiteren Parameter werden in den folgenden Kapiteln beschrieben.

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

4.2.1 Definition der Variablenabfrage-Strings

Das Format der Variablenabfragen kann für jede Variable mittels folgender Parameter bestimmt werden:

Request

String, der an das Gerät zum Abfordern des Datenpaketes gesendet werden soll.

Wait

Gibt die Zeit an, die das Tixi Alarm Modem wartet, bevor der Requeststring an das Gerät geschickt wird.

ResTime

Gibt die Zeit an, die das Tixi Alarm Modem wartet bis das ersten Zeichen der Antwort kommt.

CharTime

Gibt die Zeit an, die maximal zwischen den Zeichen vergehen darf. Wird diese Zeit überschritten, gilt die Antwort als beendet.

Wenn **ResTime** ohne **CharTime** verwendet wird, ist ResTime die Gesamtzeit für den Empfang der Antwort. Nach dieser Zeit gilt die Antwort als abgeschlossen, egal ob noch weitere Zeichen kommen.

ResEnd

String, der das Ende der Nachricht markiert. Wird ResEnd nicht angegeben, gilt nur das angegebene Timeout (ResTime/CharTime). Ansonsten müssen beide erfüllt sein.

4.2.2 Auswertung der Strings

Die Parameter gelten solange, d.h. für alle nach einer Anfrage erhaltenen Antworten, bis ein neues Format einer anderen Variablenabfrage bestimmt wurde.

Find

Im Empfangspuffer wird nach diesem String gesucht.

Pos

Das einzulesende Datenfeld befindet sich an der angegebenen Position. Diese bezieht sich auf den Anfang des empfangenen Textes oder, wenn angegeben, auf das Ende des Suchstrings (Find).

FindPos

Damit kann die Suche ab der angegebenen Position gestartet werden.

End

Das Ende des einzulesende Datenfeldes befindet sich an der angegebenen Position. Diese bezieht sich auf den Anfang des empfangenen Textes oder, wenn angegeben, auf das Ende des Suchstrings (Find).

Width

Alternativ zur Endposition kann die Größe des Datenfeldes angegeben werden.

4.2.3 Modifizierung der Werte

Die Datentypen können durch die Angabe von "**Radix**" modifiziert werden.

"D" dezimal
"H" hexadezimal
"O" oktal
"B" binaer

Fließkommaformate:

"K" Vor- und Nachkommastellen durch Kommata getrennt. Punkte werden als Tausendertrennzeichen gewertet und ignoriert.
Ein 'E/e' wird als Exponent gewertet. '+/-' als Vorzeichen. Alles andere außer Ziffern stoppt die Wandlung.
"P" Vor- und Nachkommastellen durch Punkt getrennt. Die Kommata werden als Tausendertrennzeichen gewertet und ignoriert.
Ein 'E/e' wird als Exponent gewertet. '+/-' als Vorzeichen. Alles andere außer Ziffern stoppt die Wandlung.

Die Default-Radix sind abhängig vom Datentyp:

Datentyp	Radix
B,W,DW,I	D
F,DF	P
C,S	"" (leerer Text)

Anmerkung zu den Flags:

Die Flags werden als ein Zeichen eingelesen. Ist dieses "J/j/Y/y/1/0" ergibt es ein TRUE (1).
Die Zeichen "N/n/0" ergeben ein FALSE (0).

Liste der unterstützten Variablen für das ASCII Protokoll

Typ	index	access	Kommentar
C	0-65535	R/W	Flag
W	0-65535	R/W	Word
DW	0-65535	R/W	DWord
F	0-65535	R/W	Float
DF	0-65535	R/W	Double
B	0-65535	R/W	Byte
S	0-65535	R/W	String
I	0-65535	R/W	Integer

4.3 Modbus RTU

Die zu überwachenden Variablen der angeschlossenen Modbus-Steuerung müssen im Tixi Alarm Modem definiert sein.

Die Modbus Variablen sind in der External-Gruppe der '**PROCCFG**' Datenbank gespeichert:

```
<External>
  <Bus _="C1" protocol="Modbus,RTU" type="Master" baud=19200">
    <Device _="1" Pollrate="1s" DWordInc="1" Timeout="3000s">
      <Coil1 _="C" ind="0x2000" acc="RW"/>
      <Input1 _="I" ind="0x03E0" acc="R"/>
      <InputReg5 _="R" ind="0x2005" acc="R"/>
      <Register10 _="H" ind="0x200A" acc="RW" def="1"/>
      <Register3 _="D" ind="0x4003" acc="RW" def="1"/>
    </Device>
  </Bus>
</ External >
```

Verwende COM-Port auf Erweiterungskarte C1

Variablenliste

Der BUS-Parameter enthält die Portadresse der Erweiterungskarte, das Protokoll "Modbus,RTU", den Kommunikationsmodus "Master" und die verwendete Baudrate.

Unterstützte Baudraten: 1200, 4800, 9600, 14400, 19200, 38400

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Für die Modbus Station muss ein 'Device'-Eintrag erstellt werden, welcher die Stationsnummer der angeschlossenen Slave-SPS enthält ('_' – Attribute), sowie den Abfragezyklus. Nach Ablauf des Abfragezyklus werden die Variablenwerte neu eingelesen.

Im Device Abschnitt sind spezielle Parameter möglich, die die Modbus Kommunikation zwischen Modem und SPS regeln:

<i>CharTimeout</i>	Timeout zwischen den Zeichen (50ms,)
<i>Pause</i>	Pause zwischen den Nachrichten (50ms)
<i>Timeout</i>	Timeout für Antwort (300ms)
<i>DWordInc</i>	AdressIncrement zwischen zwei aufeinanderfolgenden DWORDSs (2)
<i>DwordSwap</i>	Muss gesetzt werden, wenn Low vor High Word in DWORD gesendet wird (0)
<i>ForceSingleWordWrite</i>	Setzen, wenn Funct 0x06 anstelle von 0x10 für einzel WORD Schreiben verwendet werden soll (0)
<i>UseCache</i>	Ist der Wert auf 0 gesetzt, wird das Zusammenfassen von aufeinanderfolgenden Variablen zu blockweisen Abfragen (Caching) deaktiviert. Alle Variablen werden in einzelnen Abfragen geholt. (Default: 1)
<i>MaxElements</i>	Begrenzt die beim Caching in einer Modbus-Nachricht abgefragten Elemente.

Es kann eine Liste von Variablen definiert werden:

```
<Alarm11 _="C" ind="0x03E3" acc="R" read="1" write="5"/>
```

Variablentyp in der SPS (C,I,H,...)

Jede Zeile definiert einen logischen Namen (Alias, z.B. Alarm11) und den Typ der Variable in der Modbus-Steuerung (siehe: Liste der unterstützten Variablen)

Der 'ind' Parameter definiert die Adresse der Modbus RTU Variable als HEX-Code und der 'acc' Parameter das Zugriffsrecht. Das Zugriffsrecht kann entweder 'R' oder 'RW' für Lese oder Lese-/Schreibzugriff sein, abhängig von der gewählten Variable. Fügt man dem acc Attributwert ein C hinzu (z.B. 'RWC'), wird die betreffende Variable nicht sofort abgefragt sondern geprüft, ob die folgende auch noch in der Abfrage mitgelesen werden kann (caching, falls UseCache=0).

Der 'def' Parameter bestimmt den Startwert der Variable. Eine Variable mit Schreibzugriff enthält diesen Startwert bis zum ersten Schreibzugriff. Eine Variable mit Lesezugriff enthält diesen Wert bis das Alarm Modem den tatsächlichen Wert von der SPS erhalten hat.

Liste der unterstützten Variablen für Modbus RTU:

Typ	index	access	Kommentar
C	0-65535	RW	Coil (single bit)
I	0-65535	R	Discrete input
R	0-65535	R	Input register (unsigned)
H	0-65535	RW	Holding register (WORD Marker, unsigned)
D	0-65534	RW	Holding Register (DWORD Marker, unsigned)
RI	0-65535	R	Input register (signed integer)
HI	0-65535	RW	Holding register (WORD Marker, signed integer)
DI	0-65534	RW	Holding Register (DWORD Marker, signed integer)

Wenn nur Lesezugriff ,R' möglich ist, kann der Parameter ,acc' weggelassen werden.

Modbus function codes

Das Tixi Alarm Modem verwendet, abhängig vom Variablentyp, folgende Modbus Function-Codes:

Code (dezimal)	Variablentyp
1 - Read Coil Status	C
2 - Read Input Status	I
3 - Read Holding Registers	H, HI, D, DI
4 - Read Input Registers	R, RI
5 - Force Single Coil	C
6 - Preset Single Register	H, R, D, HI, RI, DI (wenn ForceSingleWordWrite=1)
15 - Force Multiple Coil	C (wenn size>1)
16 - Preset Multiple Registers	H, R, D, HI, RI, DI

Über die Variablenattribute "read" und "write" können die Function-Codes geändert werden. Register werden standardmäßig über FC16 geschrieben. Wenn FC6 benötigt wird, kann dieser mit "ForceSingleWordWrite" aktiviert werden.

Fernzugriff

Für den Fernzugriff auf die SPS ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

```
[<TransMode baud="19200" format="8N1" com="MB"/>]
```

(Verwenden Sie com="C1" wenn die SPS an einer RS232-2 Erweiterungskarte steckt.

4.4 Modbus ASCII

Die Modbus ASCII Parametrierung ist ähnlich der Modbus RTU (Kapitel 4.3).

Der `Bus protocol` Parameter muss "Modbus,ASCII" lauten.

Die speziellen Modbus RTU Device-Parameter (`CharTimeout`, `Pause`, `Timeout`, `DWordInc`, `DwordSwap`, `ForceSingleWordWrite`, `UseCache`) sind bei Modbus ASCII nicht gültig.

4.5 M-Bus

M-Bus (Meter-Bus) ist ein Feldbus Protokoll um Energie- und Klimageräte auf effiziente Art und Weise zu überwachen, auch wenn mehrere Geräte am Tixi Alarm Modem angeschlossen sind.

Die M-Bus Variablen sind in der External Gruppe der 'PROCCFG' Datenbank registriert.

```
<External>
  <Bus _="C0" protocol="Meterbus">
    <Device _="1" PrimaryAddr="123" Pollrate="1s">
      <SecondaryAddr _="ident" acc="R"/>
      <Var01 ind="1"/>
      <Var02 ind="2"/>
    </Device>
    <Device _="2" SecondaryAddr="12345678" Pollrate="60s">
      <Var01 ind="1"/>
      <Time _="DateTime"/>
    </Device>
  </Bus>
</External>
```

Verwende M-BUS Schnittstelle an Erweiterungskarte C0

Gerät 1

Gerät 2

Für den einfachen Anschluß von M-Bus Geräten ist eine spezielle M-Bus Erweiterungskarte verfügbar. Die Standardadresse ist "C0". M-Bus Geräte können ebenso über einen externen M-Bus/RS232 Konverter angeschlossen werden.

Wenn das Tixi Alarm Modem an der "MB"-Schnittstelle (Mainboard) angeschlossen ist, startet die SPS-Kommunikation direkt nach Abziehen des PC-Kabels, und unterbricht automatisch bei Erkennung eines TiXML-Befehls.

Die M-BUS Implementierung arbeitet als Bus-Master mit der Baudrate 2400, 8 Datenbits, 1 Stopbit, gerader Parität und ohne Handshake.

Für jedes M-BUS Gerät muss ein 'Device'-Eintrag erstellt werden, welcher mindestens die Primäradresse (`PrimaryAddr`, dezimal), Sekundäradresse (`SecondaryAddr`, 8 dezimals) oder Fabrikationsadresse (`FabricationAddr`, 8 dezimals) des Gerätes sowie den Abfragezyklus enthalten muss.

Optionale Parameter sind "ManufactoryCode" (3 ASCII Zeichen), "Generation" (hex) und "Medium" (hex) welche als weitere Unterscheidungsmerkmale für Geräte mit gleicher Adresse verwendet werden können.

Es kann eine Liste von Variablen definiert werden:

```
<Var01 ind="1"/>
```

Jede Zeile definiert einen logischen Namen (Alias, z.B. Var01) und der "ind"-Parameter die Position der Variable im M-BUS Telegramm.

Ausgabe der Gerätekennungen

Die Gerätekennungen „Primäradresse“, „Sekundäradresse“ sowie der „Herstellercode“ können bei der Abfrage des Gerätes z.B. fürs Logging mit ausgegeben werden. Dazu müssen spezielle Einträge in der Variablenliste vorgenommen werden:

```
<PrimaryAddr _="primary" />
<SecondaryAddr _="ident" />
<Manufacturer _="manufacturer" />
```

Spezielle Initialisierungen

Time:

Um die Tixi Alarm Modem RTC Zeit an ein M-BUS Gerät zu übertragen, muss die Variable "Time" (nicht lesbar) im Device-Abschnitt des Gerätes definiert werden:

```
<Time _="DateTime" />
```

Reset:

Zu Beginn der Kommunikation kann ein "Reset Code" zum M-BUS Gerät gesendet werden. Die notwendige Variable "ResCode" ist nicht lesbar und enthält den Reset-Code als Parameter "def" (Wert=0-255):

```
<ResCode _="Reset" def="114" />
```

Rohdaten Initialisierung:

Zu Beginn der Kommunikation wird ein benutzerdefiniertes Datagram (Parameter def) zum M-BUS Gerät gesendet:

```
<Init _="Raw" def="7304FD0834120000" />
```

Die def Zeichenkette muss in hex Bytes angegeben werden. Das erste Byte ist das CI-Feld gefolgt von den Rohdaten ohne Prüfsumme.

VIF – Value Information Field: Medium/Einheit:

Beim Lesen des M-BUS Variablen kann das Alarm Modem die Informationen des VIF bzgl. Medium und Einheit ausgeben. Weitere Informationen stehen im Kapitel 5.

Datenlogging:

Beim Logging von M-Bus Werten werden, um keine M-Bus Daten zu verlieren, im Binärlogfile standardmäßig 37Byte pro Wert beansprucht (32Byte String + 5Byte Datentyp). Sollen reine M-Bus Zahlenwerte geloggt werden, kann im Logfilerecord durch Angabe des Attribut size der verwendete Speicher auf die tatsächlich notwendige Größe reduziert werden.

Bei reinen Zahlenwerten empfehlen wir eine Größe von 9Byte (4Byte Wert + 5Byte Datentyp), z.B.:

```
<Datalogging_0>
<Variable_0 _="meterbus" path="/Process/Bus1/Device_0/Variable_0"/>
<Variable_1 _="meterbus" size="9" path="/Process/Bus1/Device_0/Variable_1"/>
</Datalogging_0>
```

Weitere Informationen zum Datenlogging und zur Berechnung der Logfilegröße finden sie im TiXML-Reference-Manual.

Fernzugriff

Für den Fernzugriff auf die Geräte ist folgender "TransMode"-Befehl notwendig: (siehe TiXML-Reference manual für weitere Informationen)

Tixi Alarm Modem M-BUS Schnittstelle:

```
[<TransMode baud="2400" format="8E1" com="C1" />]
```

4.6 CAN-Bus

Noch nicht freigegeben.

5 Formatieren von SPS Variablenwerten

Ohne Formatierung werden die Variablen so dargestellt, wie sie von der SPS übermittelt werden. Das Tixi Alarm Modem kann diese Werte in Zahlenformate ändern und boolesche Variablenwerte durch Zeichenketten ersetzen. Die formatierte Variable wird in E-Mails eingebunden und bei Get-Befehlen (ohne eigenes Formatattribut) ausgegeben.

Formatieren von SPS-Variablenwerten

Bei der Variablendefinition:

```
<Variable ...simpleType="Uint8" exp="2"... format="Elements;Text"/>
```

Bei der Variablenwertabfrage:

```
<Get ... format="Elements;Text"/>
```

Beim Setzen des Variablenwertes:

```
<Set ... format="Elements"/>
```

Beschreibung:

Der Parameter `format` besteht aus zwei Teilen, die durch ein **Semikolon** getrennt sind:

1. Teil:

Enthält **Format-Elemente**, die die Aus- oder Eingabe von Zahlen beschreiben. Abgesehen vom Tausendertrennzeichen „T“ und dem Zahlenformat „F“ sind die Formatelemente nicht miteinander kombinierbar. Die Position des Tausendertrennzeichenelementes im Format-Befehl ist beliebig. Die Formatanwendung hängt vom verwendeten Typ der Variablen ab. Nicht alle Formatierungen sind für alle Typen gleichermaßen geeignet. Für die Eignung eines Formatierungselements ist der im Attribut „simpleType“ der Variablendefinition angegebene Basistyp der Variablen ausschlaggebend. Daher werden hier für jedes Formatelement die dafür geeigneten Basistypen angegeben. Der erste Teil kann auch leer sein. Dann werden der Variablenwerte in in seinem nativen Format ausgegeben.

2. Teil:

Enthält einen **Text** der zusammen mit dem Wert der Variablen ausgegeben wird. Der Variablenwert kann innerhalb dieses Textes in dem durch den ersten Teil definierten Format ausgegeben werden. Dazu wird die Position des Variablenwertes durch einen Platzhalter (%%) dargestellt. Für bestimmte Variablen können noch weitere, zugehörige Variablenwerte (z.B. die physikalische Größe und die Einheit) in den Text durch Platzhalter eingesetzt werden. Der zweite Teil kann auch weggelassen werden. Dann braucht auch kein Semikolon vorangestellt werden.

Beispiel:

Beide Teile: “T’F+9,2 ;Radius %% cm“

Nur 1. Teil: “R16“

Nur 2. Teil: “Text mit:%% als Wert“

Format Elemente (Teil 1):

? – **logische Alternative** ?`string1,string2`

Dieser Befehl ersetzt die beiden Werte boolescher Variablen durch Zeichenketten. Wenn die Variable ungleich 0 ist, wird `string1` ausgegeben, andernfalls `string2`.

Anwendbar für folgende simpleType Werte:

Uint8, Uint16, Uint32, Int8, Int16, Int32 mit `exp= "0"` und Bit

Beispiel:

```
<Variable _="F" simpleType="Uint8" exp="0" ... format="?open,closed"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet:

```
<Get _="open"/> bei Wert 1
```

*** - Auswahl Alternative *Value1:Text1*Value2:**:Text3**

Dieser Befehl wird verwendet um Variablenwerte durch vordefinierte Zeichenketten zu ersetzen. Wenn der Wert gleich Value1 ist wird Text1 ausgegeben, wenn der Wert gleich Value2 ist, wird Text2 ausgegeben, in allen anderen Fällen Text3.

*** Trennzeichen für zu erkennenden Wert**

**** Trennzeichen für alle anderen Werte**

Die Anzahl der Werte ist unbegrenzt.

Anwendbar für folgende simpleType Werte:

Uint8, Uint16, Uint32, Int8, Int16, Int32 und exp= "0"

Beispiel:

```
<Variable _="R" simpleType="Uint8" exp="0" ...  
format="*0:low*1:medium*2:high**:faulty"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet:

```
<Get _="low"/> bei Wert 0
```

```
<Get _="medium"/> bei Wert 1
```

```
<Get _="high"/> bei Wert 2
```

```
<Get _="faulty"/> bei Wert 7
```

R/r - Basis Rn/rn

Der Befehl definiert die Basis n des auszugebenden Wertes.

n = 2 Binärausgabe (z.B. 01101010)

n = 8 Oktalausgabe (z.B. 21057)

n = 10 Dezimalausgabe (Standard, z.B. 1234)

n = 16 Hexadezimalausgabe (z.B. AE03)

Mit der Groß- oder Kleinschreibung wird die Ausgabe von Buchstaben bei der Hexadezimaldarstellung gesteuert:

R Es werden Großbuchstaben versendet (z.B. AE03)

r Es werden Kleinbuchstaben verwendet (z.B. ae03)

Anwendbar für folgende simpleType Werte:

Uint8, Uint16, Uint32, Int8, Int16, Int32 und exp= "0"

Beispiel:

Wert in Großbuchstaben:

```
<Variable _="R" simpleType="Uint8" exp="0" ... format="R16"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Variablenwert=90):

```
<Get _="5A"/>
```

Wert in Kleinbuchstaben:

```
<Variable _="R" simpleType="Uint8" exp="0"... format="r16"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Variablenwert=90):

```
<Get _="5a"/>
```

T - Tausendertrennzeichen **Tn**

Definiert das Trennzeichen, welches an jeder tausender Stelle der Ausgabe erscheint.

n= , Komma als Tausendertrennzeichen (z.B. 12,345,678)

n= . Punkt als Tausendertrennzeichen (z.B. 12.345.678)

n= ` Hochkomma als Tausendertrennzeichen (z.B. 12`345`678)

n= leer Kein Tausendertrennzeichen (Standard)

Hinweis:

Mit Formatelement "F" kombinierbar, kann aber auch einzeln verwendet oder weggelassen werden.

Anwendbar für folgende simpleType Werte:

Uint8, Uint16, Uint32, Int8, Int16, Int32, Float, Double

Beispiel:

```
<Variable _="R" simpleType="Uint32" exp="0"... format="T. "/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Variablenwert=98765):

```
<Get _="98.765"/>
```

F - Zahlenformat

F Vorzeichen Leerstellen Feldbreite Dezimalzeichen Dezimalstellen

Dieser Befehl definiert das Format einer Zahl.

Er enthält verschiedene Elemente, welche in dieser Reihenfolge anzugeben sind:

Vorzeichen: Legt fest, ob ein Vorzeichen ausgegeben werden soll

- +** Das Vorzeichen wird immer ausgegeben (z.B. "+12.3" , "-12.3")
- Es wird nur ein negatives Vorzeichen ausgegeben (z.B. "12.3" , "-12.3")

Leerstellen: Legt fest, wie Leerstellen aufgefüllt werden (nur bei Verwendung von Feldbreite)

- 0** Leere Stellen werden mit 0 aufgefüllt (z.B. 0066.3)
- leer** Leere Stellen werden nicht aufgefüllt (z.B. 66.3)

Feldbreite: Gibt die maximale Feldgröße **inklusive** Vorzeichen und Trennzeichen an. Ohne diese Angabe ist das Feld unbegrenzt, und es werden keine Leerzeichen aufgefüllt.

Geben Sie hier immer ausreichend große Werte an, sonst wird die ausgegebene Zahl links abgeschnitten.

Dezimalzeichen: Dieses Zeichen wird als Dezimaltrenner verwendet (optional)

- ,** ein Komma als Trennzeichen
- .** ein Punkt als Trennzeichen (Default)

Dezimalstellen: Bestimmt die Anzahl der Nachkommastellen.
Kann weggelassen werden, wenn kein Dezimalzeichen angegeben wurde.

Hinweis:

Mit Formatelement "T" kombinierbar, kann aber auch einzeln verwendet oder weggelassen werden.

Anwendbar für folgende simpleType Werte:

Uint8, Uint16, Uint32, Int8, Int16, Int32, Float, Double

Beispiele:

Der Wert der Variable ist in allen Beispielen "12345":

Vorzeichen:

```
<Variable _="F" simpleType="Float"... format="F+"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Wert = 123,45):

```
<Get _="+123.45"/>
```

Feldbreite, Leerstellen:

```
<Variable _="R" simpleType="Uint32" exp="-3"... format="F09"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Wert = 123,456):

```
<Get _="00123.456"/>
```

Feststellenzahl , Dezimalzeichen, Dezimalstellen, Länge :

```
<Variable _="R" simpleType="Int32" exp="-3"... format="T'F+9,2"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Wert = 3123,456):

```
<Get _="+3'123,45"/>
```

```
<Variable _="R" simpleType="Int32" exp="0"... format="T'F+9.2"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Wert = -3123456):

```
<Get _="-312'345"/>
```

Gleitkommazahl, Dezimalzeichen, Dezimalstellen, Länge:

```
<Variable _="F" simpleType="Float"... format="T'F+9.2"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (Wert = 3123,456):

```
<Get _="+3'123.45"/>
```

Text (Teil 2):

%%

Dieser Platzhalter kennzeichnet die Position des Variablenwertes im Ausgabertext. Dieser Teil geht für alle Datentypen. Für „String“ ist er die einzige Formatierungsoption.

Beispiel:

```
<Variable _="R" simpleType="Int32" exp="-2" ...  
      format="F+;Temp: %%°C"/>
```

```
<Get _="/Process/Aux1/D1/Variable"/>
```

Tixi Alarm Modem antwortet (wert ist 123,45):

```
<Get _="Temp: +123.45°°C"/>
```

%M% – M-BUS Medium (VIF)

%U% – M-BUS Einheit (VIF)

Diese Platzhalter geben die im M-BUS Value Information Field übergebenen Daten für Medium und Einheit aus.

Beispiel:

```
<Var01 simpleType="Int32" exp="-2"...  
      format=";Medium:%M% Wert=%% %U%"/>
```

```
<Get _="/Process/Aux0/D1/Var01"/>
```

Tixi Alarm Modem antwortet (Variablenwert=25,30, Wärmezähler Volumen):

```
<Get _=" Medium:Heat 0 Volume Flow Wert=25.30 l/h"/>
```

6 Verwenden der SPS-Variablen im Tixi Alarm Modem

Die SPS-Variablen können im Tixi Alarm Modem genauso verwendet werden, wie die Tixi Alarm Modem eigenen Ein- und Ausgänge, z.B. um Alarme auszulösen, Daten zu loggen oder für Fernwirken.

6.1 Standardadressierung

Nach dem Definieren der SPS-Variablen im Tixi Alarm Modem können diese über den Prozesspfad angesprochen werden:

Beispiel (RS232-2, Station 0, Variable Alarm11):

```
<L _="&#xae;/Process/Aux1/D0/Alarm11"/>
```

Die Variablen sind im Pfad der Erweiterungskarte (z.B. AuxMB, Aux0 für C0, Aux1 für C1) und der Stationsnummer (Device ID) der SPS (z.B. D0, D1, D2,...) aufgeführt und können dort über Ihren Aliasnamen angesprochen werden.

6.2 Adressierung über Busname und Stationsname

Durch die Verwendung von Busnamen und Stationsnamen können die Variablen unabhängig von der Schnittstelle und Stationsnummer im System adressiert werden. Dadurch kann die SPS-Schnittstelle oder Stationsnummer an zentraler Stelle geändert werden, ohne das gesamte Projekt ändern zu müssen.

Der COM-Port kann dabei ab Firmware 1.80 entsprechend der Beschriftung mit COM1 bis COM3 definiert werden.

Beispiel:

```
<External>
  <Bus _="COM2" Name="MyPLC" protocol="Mitsubishi,Alpha2"
    type="Master" baud="9600">
    <Device _="0" Name="AlphaXL" Pollrate="1s">
      <Input1 _="I" ind="1"/>
    </Device>
  </Bus>
</External>
```

Der Input1 kann im System nun wie folgt angesprochen werden:

```
<L _="&#xae;/Process/MyPLC/AlphaXL/Alarm11"/>
```

Diese Adressierung wäre auch dann noch gültig, wenn in der SPS-Definition die Schnittstelle "COM2" und/oder die DeviceID "0" geändert würde.

6.3 Überwachung der SPS-Kommunikation

Für jedes parametrisierte Gerät werden automatisch Systemvariablen eingefügt.

DeviceState:

```
[ <Get _="/Process/Aux?/D?/DeviceState"/> ]
```

Diese Variable zeigt den aktuellen Zustand der Kommunikation an:

SPS-Antwortet: DeviceState=1

keine Antwort: DeviceState=0

ChangeToggle:

```
[<Get _="/Process/Aux?/D?/ChangeToggle"/>]
```

Wenn das Alarm Modem bei einem Pollzyklus Wertänderungen in der SPS erkannt hat, wechselt diese Bitvariable ihren Zustand.

Beide Variablen können somit auch in den EventStates z.B. als Alarm- oder Logtrigger verwendet werden.

Active:

```
[<Get _="/Process/Aux?/D?/Active"/>]
```

Über diese schreibbare Variable läßt sich die Kommunikation zur SPS unterbrechen:

```
[<Set _="/Process/Aux?/D?/Active" value="0"/>] stopt die Kommunikation.
```

```
[<Set _="/Process/Aux?/D?/Active" value="1"/>] startet die Kommunikation (default).
```


Index

ABB series 33
Active 47
Adresse 9
Adressierung 47
Allen Bradley 33
Alpha XL 16
Arrays 9, 13
ASCII Protokoll 35
Bus 4
BusId 4
Busname 47
CAN-Bus 42
Carel 32
Condition 7
Definition 4
Device 4
DeviceState 47
Easy 26
External 4
Fehlerzustände 14
Feldbus 34
Format 42
Formatierung 11, 42
FULL 5
Fullduplex 5
FX 18
Get 11
GUF 6
HALF 5
Halfduplex 5
MAXADR 6
M-Bus 40
M-Bus Logging 41
MELSEC 18
Meterbus 40
Mitsubishi 16, 18
Modbus ASCII 40
Modbus Function Codes 39
MODBUS RTU 38
Moeller Easy 400/600/800/MFD 26
Moeller PS30 29
Moeller PS4/40 29
noDTR 5
OMRON 33
Parameternummer 9
Referenzen 47
RTSCTS 5
SAIA Burgess 30
S-Bus 30
Set 12
Siemens Simatic S7-200 20
Siemens Simatic S7-300 22
SPS-Anbindung 4
SPS-Status 47
SPS-Systeme 16
Startwert 9
Stationen 4
Stationsname 47
Text Protokoll 35
Tixi-Bus 34
Trigger 47
TS 5
TS-Adapter 23
Übersicht 3
Unterstützte SPS-Systeme 16
Variablen 7
Variablen lesen 11
Variablen schreiben 12
VIPA 24
Werte abfragen 11
Werte schreiben 12
XONXOFF 5
Zugriffsrecht 8