



## Tixi SPS TiXML Handbuch

V 3.2

© 2015-2016 Tixi.Com GmbH & Co. KG, Berlin

Redaktionsschluß: 18. Juli 2016.

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

<b>1</b>	<b>ÜBERSICHT .....</b>	<b>4</b>
<b>2</b>	<b>TIXI FÜR SPS-SYSTEME PARAMETRIEREN .....</b>	<b>5</b>
2.1	SPS- UND BUSSYSTEME KONFIGURIEREN .....	5
2.2	GERÄTEVARIABLEN LESEN .....	12
2.3	GERÄTEVARIABLEN SETZEN .....	13
2.4	ARRAYS .....	13
2.4.1	<i>Arrays lesen.....</i>	<i>14</i>
2.4.2	<i>Arrays schreiben .....</i>	<i>14</i>
2.5	FEHLERZUSTÄNDE BEI GERÄTEVARIABLEN VERARBEITEN .....	15
2.5.1	<i>Fehlerzustand auslesen .....</i>	<i>15</i>
2.5.2	<i>Fehlerzustand verarbeiten .....</i>	<i>16</i>
<b>3</b>	<b>UNTERSTÜTZTE SPS-SYSTEME .....</b>	<b>17</b>
3.1	MITSUBISHI ALPHA XL .....	17
3.2	MITSUBISHI MELSEC FX .....	19
3.3	SIEMENS SIMATIC S7-200 ÜBER MPI-SCHNITTSTELLE .....	21
3.4	SIEMENS SIMATIC S7-300/400 ÜBER MPI-SCHNITTSTELLE .....	23
3.5	SIEMENS SIMATIC S7-200/300/400/1200/1500 ÜBER LAN-SCHNITTSTELLE .....	25
3.5.1	<i>Allgemeines und Hinweise .....</i>	<i>25</i>
3.5.2	<i>Busparameter .....</i>	<i>25</i>
3.5.3	<i>Device-Parameter .....</i>	<i>25</i>
3.5.4	<i>Weitere Parameter nur für den internen Gebrauch:.....</i>	<i>25</i>
3.5.5	<i>Variablen-Parameter .....</i>	<i>28</i>
3.5.5.1	<i>S7-300 / 400, S7-1200, S7-1500 .....</i>	<i>28</i>
3.5.5.2	<i>S7-200 .....</i>	<i>28</i>
3.5.6	<i>Beispiele .....</i>	<i>29</i>
3.6	VIPA CPU 100/200/300 .....	29
3.7	MOELLER EASY 400 / 500 / 600 / 700 / 800 / MFD .....	31
3.8	MOELLER PS30 & PS4/40 .....	34
3.9	SAIA BURGESS S-BUS .....	35
3.10	CAREL MACROPLUS .....	37
3.11	ABB AC010, AC031, CL REIHE .....	38
3.12	ALLEN BRADLEY PICO .....	38
3.13	THEBEN PHARAO II .....	38
<b>4</b>	<b>FELDBUS UNTERSTÜTZUNG.....</b>	<b>39</b>
4.1	TIXI-BUS .....	39
4.2	ASCII PROTOCOL.....	40
4.2.1	<i>Definition der Variablenabfrage-Strings .....</i>	<i>41</i>
4.2.2	<i>Auswertung der Strings .....</i>	<i>41</i>
4.2.3	<i>Modifizierung der Werte .....</i>	<i>42</i>
4.3	MODBUS RTU .....	43
4.4	MODBUS ASCII .....	45
4.5	MODBUS TCP.....	45
4.6	M-BUS .....	47
4.6.1	<i>M-Bus Scan .....</i>	<i>49</i>
4.7	CAN-BUS .....	50
4.8	CS-PROTOKOLL (EN 62056-21 MODE C / EN 61107).....	51
4.8.1	<i>Bus-Konfiguration.....</i>	<i>51</i>
4.8.2	<i>Device-Konfiguration.....</i>	<i>53</i>
4.8.3	<i>Konfiguration der Gerätevariablen .....</i>	<i>54</i>
4.8.4	<i>Fehlerwert einer Variablen.....</i>	<i>64</i>
4.9	D0-PROTOKOLL (EN 62056-21 MODE D).....	65
<b>5</b>	<b>FORMATIEREN VON SPS VARIABLENWERTEN .....</b>	<b>66</b>
<b>6</b>	<b>VERWENDEN DER SPS-VARIABLEN IM TIXI-GERÄT .....</b>	<b>71</b>

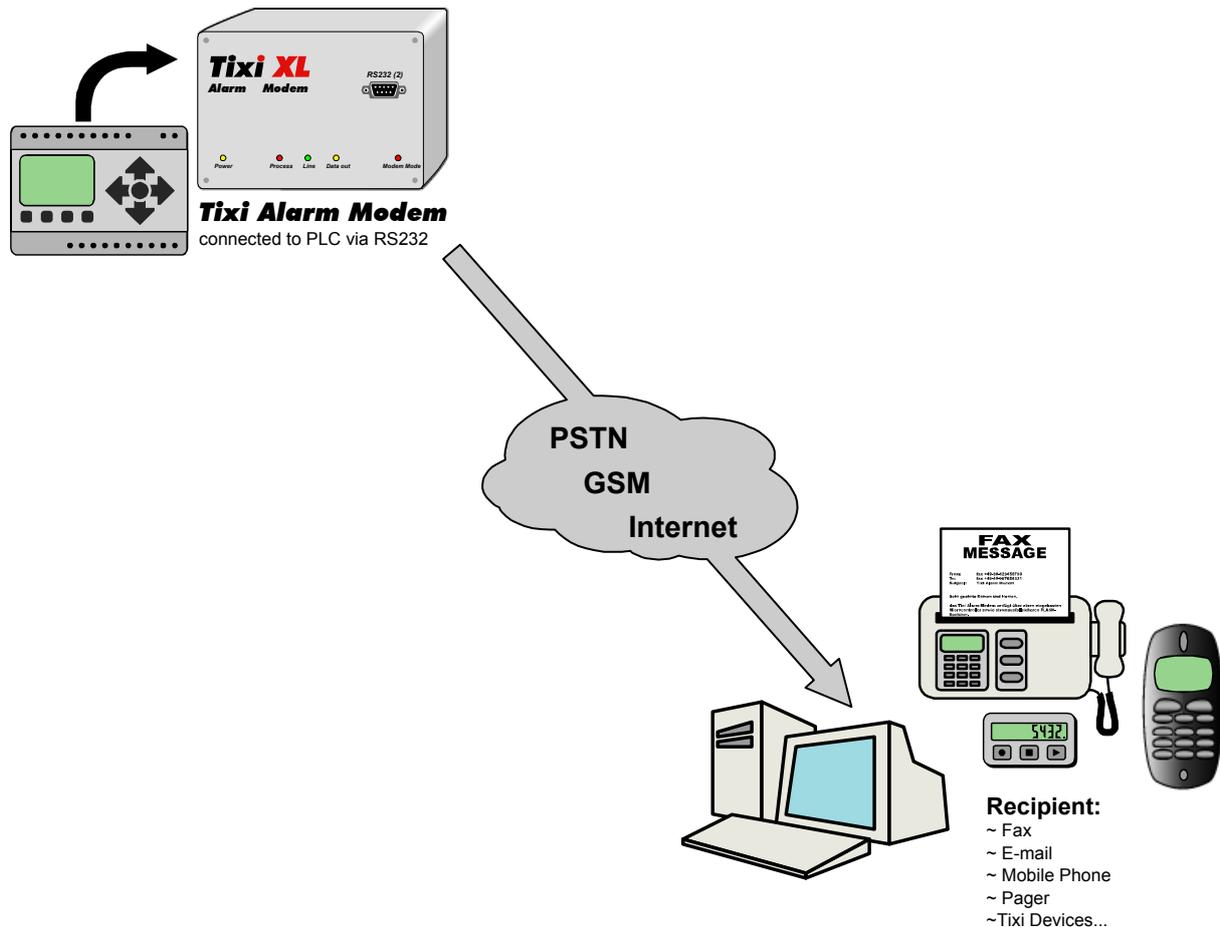
6.1	STANDARDADRESSIERUNG .....	71
6.2	ADRESSIERUNG ÜBER BUSNAME UND STATIONSNAME.....	71
6.3	ÜBERWACHUNG DER SPS-KOMMUNIKATION.....	71

# 1 Übersicht

**Tixi-Geräte** lassen sich ohne großen Aufwand in bestehende Systeme einbinden.

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

Eine einfache Anwendung für das Tixi-Gerät:



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

Alle Informationen über die hier erwähnten Protokolle und Steuerungen gelten für alle Tixi-Geräte der fünften und sechsten Generation einschließlich der Tixi Wand.Box.

## 2 Tixi für SPS-Systeme parametrieren

### 2.1 SPS- und Bussysteme konfigurieren

Um Tixi-Geräte mit der SPS kommunizieren zu lassen, muss die External-Datenbank des Tixi-Gerätes parametrieren werden. Diese Beschreibung setzt die grundlegende Kenntnis der TiXML-Sprache und des Tixi-Gerätes 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:
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)
COM3	SPS an COM3 (normalerweise M-Bus Geräte; nur bestimmte Geräte)
COM4	SPS an COM4 (RS485 Geräte; nur bestimmte Geräte)
COM5	SPS an COM5 (RS485 Geräte; nur bestimmte Geräte)
ETH	SPS an LAN Interface (Modbus TCP, Siemens S7)
<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 SPS-Busses 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.  
Abhängig vom angeschlossenen Gerät, SPS oder Bussystem.  
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-Gerät 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 Software- oder Hardware-Handshake 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, wieviel Speicher (in byte) für den Bus reserviert wird. Insgesamt stehen für alle möglichen Bus-Definitionen bis zu 20.000.000 byte (20 MB, nur bei Linux-Geräten ab FW 5.1.6.8) zur Verfügung.

<i>OwnID</i>	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-Gerätes 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 DWORDS (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-Gerätes 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
<b>Uint8</b>	Vorzeichenloser 8 Bit Wert (0...255)	exp	"123" (exp = 0) "12.3" (exp = -1) "12300" (exp = 2) "0.0123" (exp = -4)
<b>Uint16</b>	Vorzeichenloser 16 Bit Wert (0...65535)	exp	"12345" (exp = 0) "1234.5" (exp = -1) "1234500" (exp = 2)
<b>Uint32</b>	Vorzeichenloser 32 Bit Wert (0...4294967295)	exp	"1234567" (exp = 0) "123456.7" (exp = -1) "123456700" (exp = 2)
<b>Int8</b>	Vorzeichenbehafteter 8 Bit Wert (-128...+127)	exp	„123“ (exp=0) „-123“ (exp=0) "1.23" (exp=-2) "12300" (exp=2)
<b>Int16</b>	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)
<b>Int32</b>	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)
<b>String</b>	Text (0...size Zeichen)	size	„Das ist ein Text „ (nur die ersten 100 Zeichen)
<b>Blob</b>	Binärdaten Array (0...size Byte) currently not supported	size	"AF037FFF" Byteweise hexadezimal (nur die ersten 100 Byte)
<b>Bit</b>	Digitale Wert (0...1)		„0“ „1“
<b>Float</b>	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)
<b>Double</b>	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-Gerätes 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> <ul style="list-style-type: none"> <li><i>L</i> (RWL) gemeinsamer Speicher (nur bei S-Bus Variablen mit Lese-Schreibzugriff)</li> <li><i>A</i> (RA, WA, RWA) Aktiviert aktiven Variablenzugriff (Siemens)</li> <li><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.</li> </ul> <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><b>1. simpleType = String (siehe BasicType)</b> 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><b>2. simpleType= Blob (siehe BasicType)</b> 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. Optional kann noch ein Offset angegeben werden:</p> $\text{valueTAM} = \text{Factor} * \text{valueDevice}$ $\text{valueDevice} = 1/\text{Factor} * \text{valueTAM} + \text{Offset}$ <p>Der Faktor wird durch einen Bruch dargestellt z.B.: „1/1000+20“ oder „3600/1-10“. Der Nenner und der Zähler dürfen nicht Null sein. Die Verwendung dieses Attributs ist vom Variablentyp abhängig. Die gültigen Werte können Kapitel 3 entnommen werden.</p> <p><u>Beispiele:</u> multip="10/32-20" multip="40/100" multip="1/300+50"</p>
<i>Exp</i>	Exponent zur Basis 10 der die Auflösung einer Festkommazahl vom Typ <b>simpleType = UInt8, UInt16, UInt32, Int8, Int16, Int32 (siehe BasicType)</b> beschreibt.

Der im TAM gespeicherte Werte wird mit  $10^{\text{exp}(\text{Exp})}$  multipliziert (nachdem 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.

<i>wFunc</i>	Funktionscode, der beim Schreiben einer Variable verwendet wird. Ob dieses Attribut verwendet wird und welche Werte gültig sind kann aus Kapitel 3 entnommen werden.
<i>rFunc</i>	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" ViewProperties="ViewProperties"/>]
```

### 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><b>1. simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (siehe 2.1 BasicType).</b> 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><b>2. simpleType = float, double (siehe 2.1 BasicType).</b> 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><b>3. alle anderen Datentypen (siehe 2.1 BasicType).</b> 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> )
<i>Formatstring</i>	Der Wert wird entsprechend der angegebenen Formatierung ausgegeben (5, Formatieren von SPS Variablenwerten).
<i>ViewProperties</i>	Wird in der Bus-Definition das Attribute <code>AddProperties="Name,TimeStamp"</code> definiert, kann bei einem Get-Befehl mit Hilfe der <code>ViewProperties</code> der Klarname des Busses (wenn definiert) und ein Zeitstempel ausgegeben werden. Der Zeitstempel liefert die Zeit des letzten erfolgreichen Lesens der Variable.

**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><b>1. simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (siehe 2.1 BasicType).</b>  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>):  <b>Wert in Integer Darstellung = <math>10^{-Exp} * \text{Wert}</math></b></p> <p><b>2. simpleType = float, double (siehe 2.1 BasicType).</b>  Der Wert wird in der Integer-Darstellung eingegeben. Die Integer-Darstellung berechnet sich mit Hilfe der für die Variable definierten Präzision:  <b>Wert in Integer Darstellung = <math>10^{-Precision} * \text{Wert}</math></b></p> <p><b>3. alle anderen Datentypen (siehe 2.1 BasicType).</b>  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> )
<i>Formatstring</i>	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/COM?/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/COM2/D2"/>`]):  
Parametereinträge mit ungültigen Werten werden in der Antwort **nicht aufgeführt**.
2. TiXML „Get“ Befehl zum Auslesen **einer Variablen**:  
(z.B. [`<Get _="/Process/COM2/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*:

**Error** . . . . . Gibt 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
<b>0</b>	<b>0</b>	Kein Fehler
<b>1</b>	<b><i>n</i> &gt; 0</b>	Fehler in der Zugriffsimplementation des TAM. Die Nummer <i>n</i> ist > 0 und abhängig vom jeweiligen Gerät bzw. Bussystem.
<b><i>c</i> &gt;1</b>	<b><i>n</i></b>	Fehlermeldung des angeschlossenen Gerätes bzw. des Busprotokolls. Die Nummern <i>n</i> und <i>c</i> 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*:

**ErrorCode** . . . . . Gibt 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-Gerät definiert sein.

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

```

<External>
  Verwende COM-Port COM2
  <Bus _="COM2" 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>

```

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-Gerät 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
<b>M</b>	1-14	R	Systembits
<b>I</b>	1-15	R	Digitale Eingänge
<b>EI</b>	129- 132	R	Eingänge Erweiterungsmodul
<b>O</b>	1-9	RW	Ausgänge
<b>EO</b>	129- 132	RW	Ausgänge Erweiterungsmodul
<b>K</b>	1-8	R	BTasten
<b>E</b>	1-4	R	Link Eingänge
<b>A</b>	1-4	RW	Link Ausgänge
<b>N</b>	1-4	RW	Kontrollbits
<b>AI</b>	1-8	R	Analogeingang
<b>CB</b>	1-100	RW	Bitoperanden
<b>CW</b>	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-Gerät 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-Gerät 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="COM1"/>]
(verwenden Sie com="COM2", wenn die Alpha XL an der COM2 RS232 angeschlossen ist)
```

### 3.2 Mitsubishi MELSEC FX

Die zu überwachenden Variablen der angeschlossenen Mitsubishi FX Steuerung müssen im Tixi-Gerät definiert sein.

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

Verwende COM-Port COM2

```
<External>
  <Bus _="COM2" 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-Gerät 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
<b>M</b>	0-8255	RW	Merker
<b>Y</b>	0-377	RW	Ausgänge (oktal)
<b>X</b>	0-377	R	Eingänge (oktal)
<b>S</b>	0-999	RW	Schrittstatus
<b>CS</b>	0-255	RW	Zähler Kontakt
<b>TS</b>	0-255	RW	Timer Kontakt
<b>TN</b>	0-255	RW	Timer Istwert vorzeichenlos
<b>TNI</b>	0-255	RW	Timer Istwert vorzeichenbehaftet
<b>CN</b>	0-199	RW	Zähler Istwert (16Bit)
<b>CD</b>	200-255	RW	Zähler Istwert (32Bit) vorzeichenlos
<b>CDI</b>	200-255	RW	Zähler Istwert (32Bit) vorzeichenbehaftet
<b>D</b>	0-8255	RW	Register (16 Bit) vorzeichenlos
<b>DI</b>	0-8255	RW	Register (16 Bit) vorzeichenbehaftet
<b>DW</b>	0-8254	RW	Register (32 Bit) vorzeichenlos
<b>DWI</b>	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
<b>CC</b>	0-255	R	Zähler Spule
<b>TC</b>	0-255	R	Timer Spule
<b>CR</b>	0-255	R	Zähler Reset
<b>TR</b>	0-255	R	Timer Reset

**Eine Mitsubishi FX anschließen**

Das Tixi-Gerät 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-Gerät

(RS422 Erweiterungskarte)

MELSEC FX Port

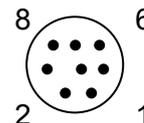
(8pin weiblich)

T+ ----- 2

T - ----- 1

R- ----- 4

R+ ----- 7



RS485 Anschluß:

Tixi-Gerät (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-Gerät RS232 Schnittstelle:

```
[<TransMode baud="9600" format="7E1" com="COM1"/>]
```

(Verwenden Sie com="COM2" wenn die SPS an der COM2 RS232 steckt.)

Tixi-Gerät RS422 Schnittstelle:

```
[<TransMode baud="9600" format="7E1" handshake="FULL" com="COM1"/>]
```

### 3.3 Siemens Simatic S7-200 über MPI-Schnittstelle

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

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

```
<External>
  <Bus _="COM2" 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>
```

RS485 Schnittstelle auf COM2

Station #2

Station #3

Der BUS-Parameter enthält die Adresse des COM-Ports, 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-Gerät kann gegenüber der S7-200 nur Master sein. Die Standardbaudrate ist 9600.

Wenn das Tixi-Gerät 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.

```
Tixi-Gerät                S7-200
(RS485 Erweiterungskarte)  (Profibus-Adapter)
    T+ -----B1
    T- -----A1
```

#### Fernzugriff

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

Tixi-Gerät RS232 Schnittstelle:

```
[<TransMode baud="9600" format="8E1" handshake="noDTR" com="COM1"/>]
```

(Verwenden Sie com="COM1" wenn die SPS an COM1 (RS232) steckt.)

Tixi-Gerät RS485 Schnittstelle:

```
[<TransMode baud="9600" format="8E1" handshake="HALF" com="COM1"/>]
```

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 über MPI-Schnittstelle

Die zu überwachenden Variablen der angeschlossenen Siemens Simatic S7-300/400 müssen im Tixi-Gerät 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 des COM-Ports, 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-Gerät 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
<b>DBX</b>	0.0-65535.7	R/W	Datenbaustein (Bit, erfordert Parameter "db")
<b>DBB</b>	0-65535	R/W	Datenbaustein (Byte, erfordert Parameter "db")
<b>DBW</b>	0-65534	R/W	Datenbaustein (Word, erfordert Parameter "db")
<b>DBD</b>	0-65532	R/W	Datenbaustein (DoubleWord, erfordert Parameter "db")
<b>DBS</b>	0-65535	R/W	Datenbaustein (string, erfordert Parameter "size" und "db")
<b>I</b>	0-16384.7	R	Eingänge
<b>Q</b>	0-16384.7	R/W	Ausgänge
<b>M</b>	0-65535.7	R/W	Merker Bit
<b>MB</b>	0-65535	RW	Merker Byte
<b>MW</b>	0-65534	RW	Merker Wort
<b>MD</b>	0-65532	RW	Merker DWort
<b>T</b>	0-2074	R/W	Timer
<b>C</b>	0-2074	R/W	Zähler

### Eine S7-300/400 anschließen

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

Tixi-Gerät	S7-300/400
(Profibus-Adapter)	(Profibus-Adapter)
B1 -----B1	
A1 -----A1	

### 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 nunmehr 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 parametrierten 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 Siemens Simatic S7-200/300/400/1200/1500 über LAN-Schnittstelle**

Die Kommunikation der S7-Steuerungen über LAN (TCP/IP) erfolgt mit Hilfe des Protokolls "ISO-ON-TCP" nach RFC1006 (manchmal auch als "AGLink" Protokoll bezeichnet). Unterstützt werden derzeit Siemens S7-200, S7-300/400, S7-1200, S7-1500.

Die Konfiguration ist für die einzelnen SPS Typen prinzipiell gleich.

**3.5.1 Allgemeines und Hinweise**

Zur Zeit (Firmware Version 3.04.01.016) muss nach jedem Laden einer neuen Konfiguration (PROCCFG) ein Reset durchgeführt werden. Bei den Bus- und Deviceparametern wird Groß- und Kleinschreibung unterschieden, "PLCC" ist also nicht gleich "plcc" !

**3.5.2 Busparameter**

```
<Bus _="ETH" BusId="1" name="S7_300" protocol="Siemens,AGLink" type="Master">
```

"_="ETH"	Bezeichnet die Ethernetschnittstelle des Tixi-Gerätes (auf dem Mainboard)
"BusId"	Ermöglicht die Adressierung des SPS-Bus unabhängig von der Schnittstelle. Darf nicht mit "Name" vermischt werden.
"Name"	Ermöglicht die Benennung des SPS-Bus und damit eine von der Schnittstelle unabhängige Adressierung. Darf nicht mit "BusId" vermischt werden.
"protocol"	Name des Protokolls, muß auf "Siemens,AGLink" eingestellt sein
"type"	Art des Protokolls, muß immer auf "Master" eingestellt sein

**3.5.3 Device-Parameter**

```
<Device _="2" Name="Dev2" Pollrate="1s" IP="193.101.167.203" GW="193.101.167.193"
  mask="255.255.255.0" rack="0" slot="2" PLCC="0">
```

"2"	Tixi-Gerätenummer; entspricht der PLC-Nummer der angeschlossenen CPU
"IP"	IP-Adresse der Siemens-SPS
"GW"	Zu verwendendes Gateway
"mask"	Netzmaske der IP Adresse (default: 255.255.255.0)
"rack"	Racknummer der Siemens CPU (default: 0)
"slot"	Slotnummer der Siemens CPU (default: 300:2, 400:3, 1200:1)
"PLCC"	PLC Class. Die Art der angeschlossenen Steuerung. 0: 300/400, 1: 200, 2: 1200 (default: 1)

**3.5.4 Weitere Parameter nur für den internen Gebrauch:**

"credit"	Bei einzelnen Fremdfabrikaten (SPS) notwendig, nicht bei Siemens
"ctype"	Typ der angeschlossenen Kommunikationseinheit. 0: PG, 1: OP, 2: Sonstige. (PG = Programmiergerät; OP = Operator Panel)
"port"	Die Portnummer für den TCP/IP-Verkehr ist standardmäßig 102. Soll ausnahmsweise eine andere Portnummer verwendet werden, so muss sie

hier angegeben werden.

Bei Siemens-Steuerungen wird immer Port 102 verwendet.

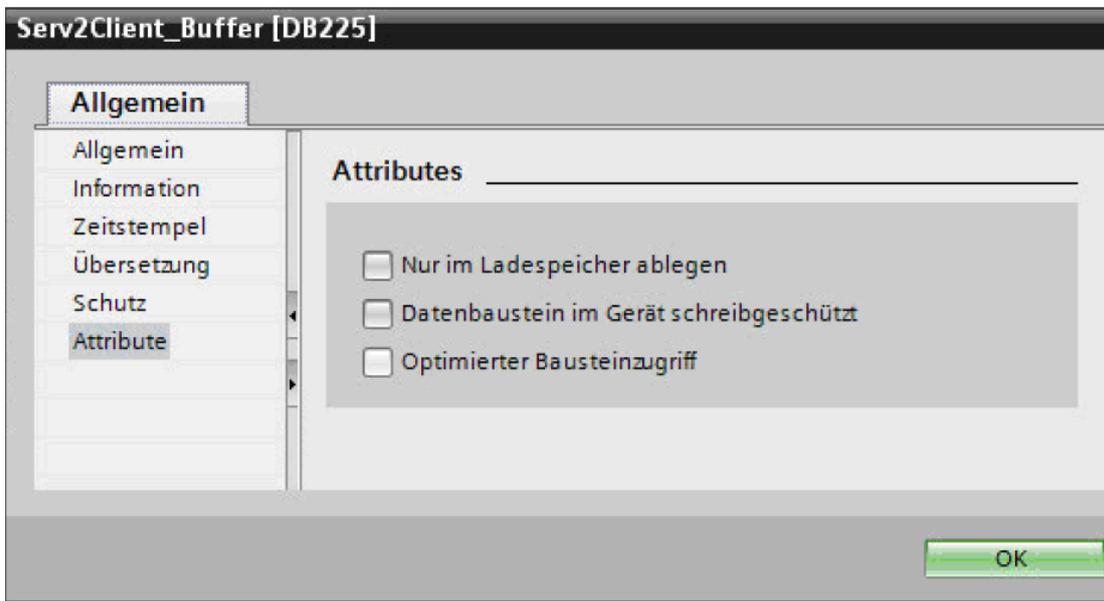
“Timeout“ Standardtimeout der Kommunikation. Ohne Maßeinheit und mit der Maßeinheit „ms“ (Millisekunden) wird der Wert so (als ms) übernommen. Bei anderen gültigen Maßeinheiten wird entsprechend auf ms umgerechnet. Ohne Angabe des Parameters, oder bei Überschreiten des Wertebereichs eines Doppelwortes (32 bit) wird der Defaultwert 6000(ms) eingestellt.

Bei den meisten Anwendungen kann auf die Angabe dieses Parameters verzichtet werden.

### Besonderheiten für S7-1200 und S7-1500

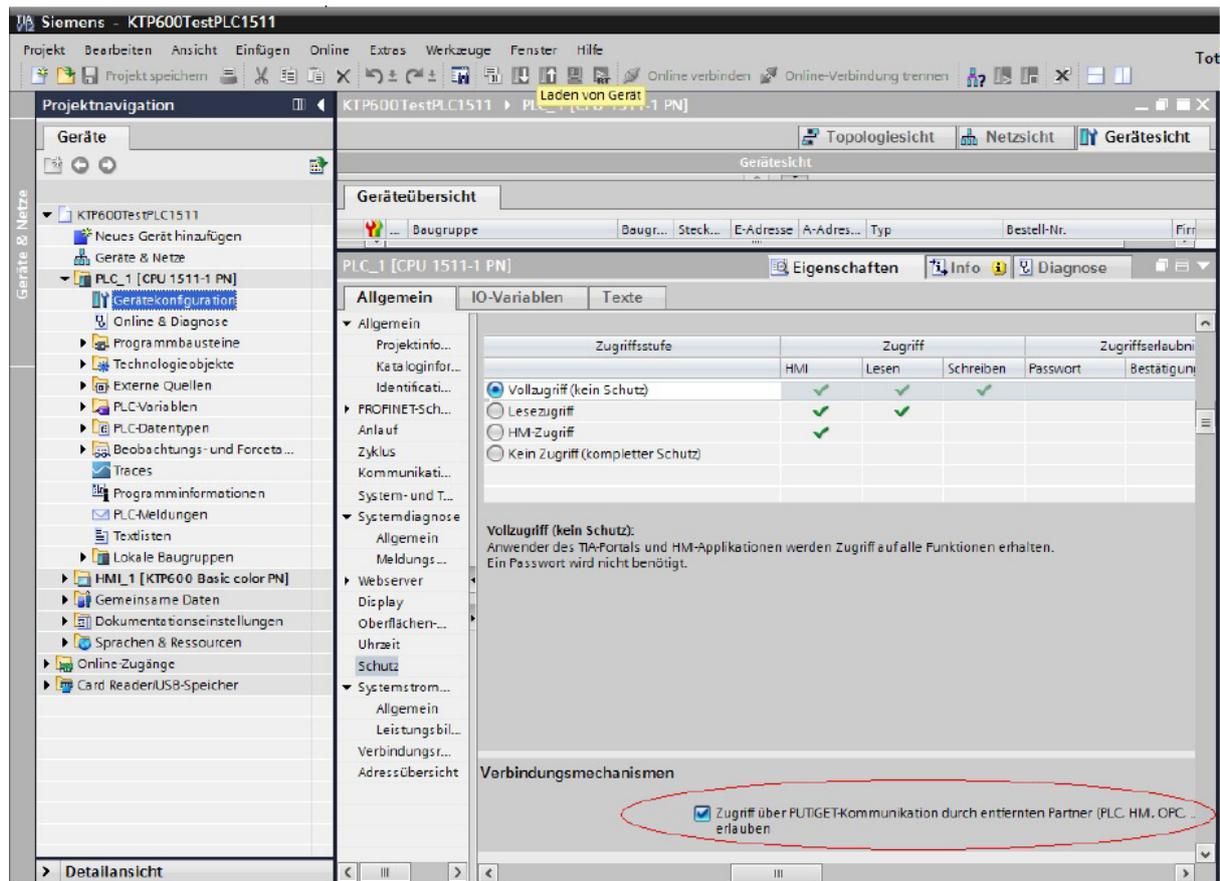
Der Zugriff auf S7-1200/1500 ist grundsätzlich möglich und ähnelt dem Zugriff auf eine S7-300 bzw. S7-400. Jedoch sind die folgenden Punkte dabei zu beachten:

- Der Zugriff auf Datenbausteine ist nur mit absoluter Adressierung möglich. Deshalb muss im Projekt der SPS das Attribut "Optimierter Bausteinzugriff" in den Eigenschaften der Datenbausteine deaktiviert werden.



Der Zugriff auf Peripherieeingangs- und Peripherieausgangsdaten einer S7-1200/1500 ist nicht möglich.

- Zugriff auf Timer (Zeiten), Counter (Zähler) und 64-Bit Datentypen einer S7-1200/1500 ist nicht möglich.
- Bei der S7-1500 und der S7-1200 (ab Firmware Version 4) muss die PUT/GETKommunikation zwingend erlaubt werden. Dies muss in der Gerätekonfiguration im SPS- Projekt aktiviert werden.



- Auf die neuen Datentypen, welche mit der S7-1200/1500 erschienenen sind, ist kein direkter Zugriff möglich.  
Auf die 64-Bit-Datentypen kann zurzeit nicht zugegriffen werden. Eventuell lässt sich der neue Datentyp in einen bekannten 32-Bit Datentyp (REAL oder DWORD) umrechnen und in einem anderen Datenbereich (z. B. als Merker, DB, ...) ausgeben.  
Für den Zugriff auf die restlichen der neuen Datentypen, kann ein vergleichbarer Datentyp einer S7-300/400 verwendet werden.

Es gibt folgende neuen Datentypen bei der S7-1500 mit teilweise vergleichbaren Datentypen der S7-300/400:

Datentyp S7-1500	Vergleichbarer Datentyp S7-300/400
USINT (8 Bit)	BYTE
SINT (8 Bit)	BYTE
UINT (16 Bit)	WORD
UDINT (32 Bit)	DWORD
ULINT (64 Bit)	zur Zeit nicht möglich
LINT (64 Bit)	zur Zeit nicht möglich
REAL (32 Bit)	REAL
LREAL (64 Bit)	zur Zeit nicht möglich

### 3.5.5 Variablen-Parameter

#### 3.5.5.1 S7-300 / 400, S7-1200, S7-1500

Unterstützte Variablen für S7-300 / 400 CPUs über LAN  
(abweichende Werte für S7-1200 / S7-1500 in Klammern)

Typ	index	access	Kommentar
<b>DBX</b>	0.0-65535.7	R/W	Datenbaustein (Bit, erfordert Parameter "db")
<b>DBB</b>	0-65535	R/W	Datenbaustein (Byte, erfordert Parameter "db")
<b>DBW</b>	0-65534	R/W	Datenbaustein (Word, erfordert Parameter "db")
<b>DBD</b>	0-65532	R/W	Datenbaustein (DoubleWord, erfordert Parameter "db")
<b>DBS</b>	0-65535	R/W	Datenbaustein (string, erfordert Parameter "size" und "db") (S7-1200: Bitte DBB verwenden !)
<b>I</b>	0.0-16384.7	R	Eingänge (S7-1200: index 0.0 .. 1023.7)
<b>Q</b>	0.0-16384.7	R/W	Ausgänge (S7-1200: index 0.0 .. 1023.7)
<b>M</b>	0-65535.7	R/W	Merker Bit
<b>MB</b>	0-65535	R/W	Merker Byte
<b>MW</b>	0-65534	R/W	Merker Wort
<b>MD</b>	0-65532	R/W	Merker DWort
<b>T</b>	0-999	R/W	Timer
<b>C</b>	0-999	R/W	Zähler

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 (also `acc="RA"` für Lesen, `acc="RWA"` für Lesen + Schreiben).

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

Beispiel:

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

"DBX" Variablentyp "Datenbaustein"  
 "2" Datenbaustein der Variablen  
 "0.0" Index der Variable innerhalb des Datenbausteins  
 "RA" Zugriffsrecht "Lesen"

#### 3.5.5.2 S7-200

Unterstützte Variablen für S7-200 CPUs über LAN:

Typ	index	access	Kommentar
<b>DBX</b>	0.0-10239.7	R/W	Entspricht Datentyp <b>V</b> (Variablenspeicher Bit). Zugriff über db=1: db="1" <b>DBX</b> ="0.0 – 10239.7"
<b>DBB</b>	0-10239	R/W	Entspricht Datentyp <b>VB</b> (Variablenspeicher Byte). Zugriff über db=1 <b>DBB</b> ="..": db="1" <b>DBB</b> ="0 – 10239"
<b>DBW</b>	0-10238	R/W	Entspricht Datentyp <b>VW</b> (Variablenspeicher Word). Zugriff über db=1 <b>DBW</b> ="..": db="1" <b>DBW</b> ="0 – 10238"
<b>DBD</b>	0-10236	R/W	Entspricht Datentyp <b>VD</b> (Variablenspeicher DWord). Zugriff über db=1 <b>DBD</b> ="..": db="1" <b>DBD</b> ="0 – 10236"
<b>I</b>	0.0-15.7	R	Eingänge
<b>Q</b>	0.0-15.7	R/W	Ausgänge
<b>M</b>	0-31.7	R/W	Merker Bit
<b>MB</b>	0-31	R/W	Merker Byte
<b>MW</b>	0-30	R/W	Merker Wort
<b>MD</b>	0-28	R/W	Merker Dwort
<b>M</b>	256.0 – 805.7	R	Entspricht Datentyp <b>SM</b> (Sondermerker Bit). Zugriff über M mit Offset 256 <b>M</b> ="256.0" entspricht <b>SM</b> ="0.0", <b>M</b> ="805.7" entspricht <b>SM</b> ="549.7"
<b>T</b>	0-255	R/W	Timer
<b>C</b>	0-255	R/W	Zähler

### 3.5.6 Beispiele

#### External-Konfiguration für eine Siemens S7-300

```
[<SetConfig _="PROCCFG" ver="v">
<External>
  <Bus _="ETH" BusId="1" Name="S7_300" protocol="Siemens,AGLink" type="Master">
    <Device _="2" Name="Dev2" Pollrate="1s"
      IP="193.101.167.203" GW="193.101.167.193"
      mask="255.255.255.0" rack="0" slot="2" PLCC="0">
      <MB0 _="MB" acc="RWA" def="" ind="0"/>
      <MB1 _="MB" acc="RWA" def="" ind="1"/>
      <MW2 _="MW" acc="RWA" def="" ind="2"/>
      <M0.0 _="M" acc="RWA" def="" ind="0.0"/>
      <M0.1 _="M" acc="RWA" def="" ind="0.1"/>
      <T0 _="T" acc="RWA" def="" ind="0"/>
      <C0 _="C" acc="RWA" def="" ind="0"/>
      <I0.0 _="I" acc="RA" def="" ind="0.0"/>
      <I0.1 _="I" acc="RA" def="" ind="0.1"/>
    </Device>
  </Bus>
</External>
</SetConfig>]
```

#### External-Konfiguration für eine Siemens S7-1200

```
[<SetConfig _="PROCCFG" ver="v">
<External>
  <Bus _="ETH" BusId="1" Name="S7_1200" protocol="Siemens,AGLink" type="Master">
    <Device _="1" Pollrate="1s" IP="193.101.167.135" GW="193.101.167.193"
      mask="255.255.255.128" rack="0" slot="1" PLCC="2">
      <!--Eingaenge-->
      <I0.0 _="I" acc="RA" def="" ind="0.0"/>
      <!--Ausgaenge-->
      <Q0.0 _="Q" acc="RA" def="" ind="0.0"/>
    </Device>
  </Bus>
</External>
</SetConfig>]
```

### 3.6 VIPA CPU 100/200/300

Die zu überwachenden Variablen der angeschlossenen VIPA CPU müssen im Tixi-Gerät 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 _="COM2" 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 COM2

Station #2

Der BUS-Parameter enthält die Adresse des COM-Ports "COM2", den Protokoll-Hersteller "Vipa", das Protokoll "GreenCable", den Kommunikationsmodus "Master", die Baudrate "38400" und den Parameter TS, welcher die Stationsnummer des Tixi-Gerätes enthält.

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

Wenn das Tixi-Gerät an "COM1" 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
<b>M</b>	0-2047.7	R/W	Merker (Bit)
<b>MB</b>	0-2047	R/W	Merker (Byte)
<b>MW</b>	0-2046	R/W	Merker (Word)
<b>MD</b>	0-2044	R/W	Merker (Doppel-Word)
<b>I</b>	0-15.7	R	Eingänge
<b>Q</b>	0-15.7	R/W	Ausgänge

Das Tixi-Gerät 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="COM1"/>]
```

(Verwenden Sie com="COM2" wenn die VIPA an der COM2 RS232 angeschlossen ist)

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

### 3.7 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-Gerät definiert sein.

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

```

<External>
  <Bus _="COM1" 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>

```

Verwende COM-Port auf Erweiterungskarte C1

Easy 800 Master

Variablenliste

Der BUS-Parameter enthält die Adresse der seriellen Schnittstelle "COM1", 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-Gerät 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
<b>I</b>	1-16	R	Eingänge
<b>Q</b>	1-8	RW	Ausgänge (Schreiben nur im STOP)
<b>R</b>	1-16	R	Eingänge Erweiterung
<b>S</b>	1-8	RW	Ausgänge Erweiterung (Schreiben nur im STOP)
<b>M</b>	1-96	RW	Merker
<b>MB</b>	1-96	RW	Byte Merker
<b>MW</b>	1-96	RW	Word Merker
<b>MD</b>	1-96	RW	Doppelword Merker
<b>Key</b>	1-4	R	Tasten
<b>IA</b>	1-4	R	Analogeingänge
<b>QA</b>	-	RW	Analogausgänge
<b>U</b>	-	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 parametrierte Gerät wird automatisch eine **Type** Variablen eingefügt:

```
[<Get _="/Process/COM?/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="COM2"/>]
```

(Verwenden Sie com="COM1" wenn die SPS an der COM1 RS232 steckt.)

**Easy 800/MFD**

```
[<TransMode baud="9600" format="8N1" handshake="noDTR" com="COM2"/>]
```

(Verwenden Sie com="COM1" wenn die SPS an der COM1 RS232 steckt.)

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

### 3.8 Moeller PS30 & PS4/40

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

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

```
<External>
  Verwende COM-Port COM1
  <Bus _="COM1" 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>
```

SucomA Master  
Variablenliste

Der BUS-Parameter enthält die Adresse des COM-Ports "COM1", den Protokoll-Hersteller "Moeller", das Protokoll "SucomA", den Kommunikationsmodus "Master" und die Baudrate (4800 bis 57600).

Die Stationsnummer muss "7" sein.

Wenn das Tixi-Gerät 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
<b>S</b>	0-65535	R	Status WORD
<b>F</b>	0-14999.7	R	Merker BIT
<b>B</b>	0-14999	RW	Merker BYTE
<b>R</b>	0-14998	RW	Merker WORD
<b>D</b>	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="COM1"/>]
(Verwende com="COM2" wenn die SPS an der COM2 RS232 angeschlossen ist)
```

## 3.9 SAIA Burgess S-Bus

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

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

```
<External>
  Verwende RS485-Port auf COM2
  <Bus _="COM2" 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>
```

aktiviert RS485 Modus

S-Bus Station #1

S-Bus Station #3

Das Tixi-Gerät kann als S-BUS Master oder Slave parametrierbar werden. Alle S-BUS Baudraten werden unterstützt, die Standardbaudrate (kein Eintrag) ist 19200. Wenn das Tixi-Gerät 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-Gerät einen getrennten Speicher für Schreib- und Lesezugriffe. In diesem Fall kann die SPS einen Variablenwert in den Lesespeicher des Tixi-Gerätes schreiben und das Tixi-Gerät 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-Gerät 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-Gerät 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-Gerät 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-Gerät 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-Gerät RS232 Schnittstelle:

```
[<TransMode baud="9600" format="8N1" com="COM1"/>]
```

(Verwenden Sie com="COM2", wenn das Tixi-Gerät an der COM2 RS232 angeschlossen ist)

Tixi-Gerät RS485 Schnittstelle:

```
[<TransMode baud="9600" format="8N1" handshake="HALF" com="COM1"/>]
```

Wählen Sie die gleiche Baudrate wie die SPS.

### 3.10 Carel Macroplus

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

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

```

<External>
  _____ Verwende RS422-Port auf COM2
  <Bus _="COM2" protocol="Carel,PC2" type="Master"
    handshake="FULL"> _____ Aktiviert RS422-Modus
    <Device _="1" Pollrate="1s"> _____ CarelBus Steuerung #1
      <Alarm11 _="D" ind="22" acc="RW"/>
    </Device>
    <Device _="3" Pollrate="60s"> _____ CarelBus Steuerung #3
      <Alarm31 _="D" ind="22" acc="RW"/>
    </Device>
  </Bus>
</ External >

```

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

Wenn das Tixi-Gerät 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-Gerät 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: \_\_\_\_\_ Variablentyp in der Carel SPS (D,I,A)

```
<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
<b>D</b>	1-183	R/W	Bits
<b>I</b>	1-50	R/W	Integer
<b>A</b>	1-50	R/W	Analog

**Macroplus anschließen**

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

Tixi-Gerät (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-Gerät RS232 Schnittstelle:

```
[<TransMode baud="1200" format="8N2" com="COM1"/>]
```

(Verwenden Sie com="COM2", wenn das Tixi-Gerät an der COM2 RS232 angeschlossen ist)

Tixi-Gerät RS422 Schnittstelle:

```
[<TransMode baud="1200" format="8N2" handshake="FULL" com="COM1"/>]
```

**3.11 ABB AC010, AC031, CL Reihe**

Der AC010 Logikcontroller ist eine OEM-Version der Moeller EASY 400/600 Produkte. Siehe Kapitel 3.7 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 3.7 für Details.

**3.12 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 3.7 für Details.

**3.13 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](mailto:Developer@Tixi.Com))

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

```

<External>
    _____ Verwende RS422-Schnittstelle auf COM2
    <Bus _="COM2" 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-Gerät 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:

```

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

```

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
<b>C</b>	0-65535	R/W	Flag (Coil)
<b>W</b>	0-65535	R/W	Word
<b>DW</b>	0-65535	R/W	DWord
<b>F</b>	0-65535	R/W	Float
<b>DF</b>	0-65535	R/W	Double
<b>B</b>	0-65535	R/W	Byte
<b>S</b>	0-65535	R/W	String (erfordert Angabe von "size")

## 4.2 ASCII Protocol

Das ASCII protocol ist eine einfache Möglichkeit mit dem Tixi-Gerät 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 auf COM2
    <Bus _="COM2" protocol="Tixi.Com,ASCII" type="Master"
        handshake="none" baud="115200" format="8N1">
        _____ Variablen des abzufragenden Gerätes
        <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>

```

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-Gerät 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:

```

_____ Variablentyp im ASCII Protokoll (C,W,...)
<Float _="DF" Pos="14" End="24" acc="R" Radix="K"
    Request="#13;" ResEnd="Ende#13;" ResTime="10s"/>

```

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-Gerät wartet, bevor der Requeststring an das Gerät geschickt wird.

##### **ResTime**

Gibt die Zeit an, die das Tixi-Gerät 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" 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
<b>C</b>	0-65535	R/W	Flag
<b>W</b>	0-65535	R/W	Word
<b>DW</b>	0-65535	R/W	DWord
<b>F</b>	0-65535	R/W	Float
<b>DF</b>	0-65535	R/W	Double
<b>B</b>	0-65535	R/W	Byte
<b>S</b>	0-65535	R/W	String
<b>I</b>	0-65535	R/W	Integer

### 4.3 Modbus RTU

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

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

```
<External>
  <Bus _="COM2" 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 COM2

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-Gerät 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-Gerät 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="COM1"/>]
(Verwenden Sie com="COM2" wenn die SPS an der RS232 COM2 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 Modbus TCP

Tixi Geräte der "Generation 6" (H600 und Wand.Box Serie) unterstützen Modbus TCP.

Die Modbus TCP Variablen werden in der External Sektion der 'PROCCFG' Datenbank definiert.

```
<External>
  <Bus _="ETH" protocol="ModbusTCP" type="Master" >
  <Device _="1" IP="IPAddress" Pollrate="1s" Timeout=""
    DWordInc="2" DwordSwap="0" ForceSingleWordWrite="0"
    UseCache="1">
    <Input1 _="I" ind="0x03E0" acc="R"/>
    <Coil1 _="C" ind="0x2000" acc="RW"/>
    <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 LAN interface

Set of Variables

Die BUS-Parameter legen den LAN Port, das Busprotokoll "ModbusTCP" und den Modus "Master" fest.

### Parameter der Device-Sektion:

`IPAddress` IP-Adresse der Steuerung

Weitere Parameter können in der Device-Sektion angegeben werden, um die Kommunikation mit der Steuerung gezielt zu steuern:

<code>CharTimeout</code>	Timeout zwischen den Zeichen (50ms,)
<code>Pause</code>	Pause zwischen den Nachrichten (50ms)
<code>Timeout</code>	Timeout für Antwort (300ms)
<code>DWordInc</code>	AdressIncrement zwischen zwei aufeinanderfolgenden DWORDSs (2)
<code>DwordSwap</code>	Muss gesetzt werde, wenn Low vor High Word in DWORD gesendet wird (0)
<code>ForceSingleWordWrite</code>	Setzen, wenn Funct 0x06 anstelle von 0x10 für einzel WORD Schreiben verwendet werden soll (0)
<code>UseCache</code>	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)

Für jedes Device können Variablen definiert werden: Variable type in Modbus RTU protocol (C,I,R,H,D)

```
<Alarm11 _="C" ind="1" simpleType="Bit" acc="RW" def="0"/>
```

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 TCP Variable 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.

#### Liste der unterstützten Variablen für Modbus TCP:

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-Gerät 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.

#### Beispiel:

```
<Device Name="Device_31" NameUser="Software-Slave-HR" _="1"
  IP="193.101.167.29" port="503" Pollrate="1s" DWordInc="2"
  DwordSwap="0" ForceSingleWordWrite="1" UseCache="1">
  <Variable_3100 Name="H0" _="H" simpleType="Uint16" ind="0" acc="RW"/>
  <Variable_3101 Name="H1" _="H" simpleType="Uint16" ind="1" acc="RW"/>
  <Variable_3102 Name="H2" _="H" simpleType="Uint16" ind="2" acc="RW"/>
  <Variable_3103 Name="H3" _="H" simpleType="Uint16" ind="3" acc="RW"/>
  <Variable_3104 Name="H4" _="H" simpleType="Uint16" ind="4" acc="RW"/>
</Device>
```

## 4.6 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-Gerät angeschlossen sind.

Für den einfachen Anschluß von M-Bus Geräten sind spezielle M-Bus Geräte mit Pegelwandler verfügbar. M-Bus Geräte können ebenso über einen externen M-Bus/RS232 Konverter (Pegelwandler) angeschlossen werden.

Die M-Bus Implementierung arbeitet als Bus-Master mit der Standard-Baudrate 2400 (konfigurierbar), 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) oder die Sekundäradresse (*SecondaryAddr*, 8 dezimals) oder Fabrikationsadresse (*FabricationAddr*, 8 dezimals) des Gerätes sowie den Abfragezyklus enthalten muss.

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

```

<External>
  <Bus _="COM3" protocol="meterbus" type="Master"
    baud="Baudrate" format="Format" Timeout="Timeout">
    <Device _="1" PrimaryAddr="123" SecondaryAddr="12345678"
      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 COM3

Gerät 1

Gerät 2

### Beschreibung der möglichen Attribute (kursiv dargestellt):

<i>baud</i>	Baudrate. Standard=2400 Baud. Mögliche Werte: 300, 2400, 9600
<i>format</i>	Datenformat. Standard=8E1
<i>Timeout</i>	maximale Wartezeit zwischen den Abfragen
<i>PrimaryAddr</i>	Primäradresse
<i>SecondaryAddr</i>	Sekundäradresse (immer 8-stellig)
<i>Pollrate</i>	Abfragerate

Wenn alle Geräte auf dem M-Bus die gleiche Primäradresse haben, können die Geräte nur über die Sekundäradresse adressiert werden. In diesem Fall darf die Primäradresse NICHT in der Device-Definition verwendet werden !

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“, „Herstellercode“ und weitere können bei der Abfrage des Gerätes z.B. für das Logging mit ausgegeben werden. Dazu müssen spezielle Einträge in der Variablenliste vorgenommen werden:

```
<PrimaryAddr _="primary"/>
<SecondaryAddr _="ident"/>
<Manufacturer _="manufacturer"/>
<State _="stat"/>
<Medium _="medium"/>
<Version _="version"/>          (verfügbar ab FW 5.1.6.6)
```

## Spezielle Initialisierungen

### Time:

Um die Tixi-Gerät 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 und weitere Werte ausgeben. Weitere Informationen: siehe Kapitel 5.

## Datenlogging:

Beim Logging von M-Bus Werten werden, um keine M-Bus Daten zu verlieren, im Binärlogfile standardmäßig 37Byte pro Wert beansprucht (32 Byte String + 5 Byte 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 13 Byte (8 Byte Wert + 5 Byte Datentyp), z.B.:

```
<Datalogging_0>
  <Variable_0 _="meterbus" path="/Process/Bus1/Dev_0/Variable_0"/>
  <Variable_1 _="meterbus" size="13" path="/Process/Bus1/Dev_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-Gerät M-BUS Schnittstelle:

```
[<TransMode baud="2400" format="8E1" com="COM2"/>]
```

### 4.6.1 M-Bus Scan

Tixi-Geräte verfügen über einen Mechanismus, um alle am M-Bus angeschlossenen Geräte automatisch zu erkennen (Scan). Folgende Voraussetzungen gelten:

- alle Zähler werden über Sekundäradresse angesprochen
- alle Zähler verwenden eine einheitliche Einstellungen für Baudrate und Datenformat

Folgender Befehl startet einen Scan auf dem M-Bus:

```
[<ScanDevices _="COM3" protocol="meterbus" type="Master" ver="v"/>]
```

Das Ergebnis eines Scans liefert das Tixi-Gerät eine Liste aller gefundenen Zähler und deren Variablen zurück. Je nach Firmwareversion des Tixi-Gerätes liefert der Scan eine unterschiedliche Anzahl von Werten zurück.

#### Geräte-Infos:

Wert	Beschreibung
PrimaryAddr	Primäradresse
SecondaryAddr	Sekundäradresse
Manufacturer	Herstellercode
Version	Version
Medium	Medium
State	Status

#### Variablen-Infos:

Wert	Beschreibung
ind	Index im M-Bus Telegramm
vib	Value information block
value	Wert
storage	storage number
tariff	Tarif-Info
subunit	subunit
Function	Function code

Optional können ein Reset-Code und eine einzelne Sekundäradresse angegeben werden:

```
[<ScanDevices _="COM3" protocol="meterbus" type="Master"
  SecondaryAddr="12345678" Reset="192" ver="v"/>]
```

#### Beispiel:

Qundis WTT16 Knoten (Device 1) mit 1 angeschlossenen Heizkostenverteiler (Device 2):

```
<ScanDevices>
  <Device PrimaryAddr="2" SecondaryAddr="10307448" Manufacturer="LSE"
    Version="30" Medium="Bus/System" State="0">
    <Var01 ind="1" vib="On Time [h]" value="6032"
      storage="0" tariff="0" subunit="0" function="0"/>
    <Var02 ind="2" vib="Time Point [Date+Time]" value="2016/07/12,14:18"
      storage="0" tariff="0" subunit="0" function="0"/>
    <Var03 ind="3" vib="Model / Version" value="3444564688926"
      storage="0" tariff="0" subunit="0" function="0"/>
    <Var04 ind="4" vib="Parameter Set Ident." value="WTT16"
      storage="0" tariff="0" subunit="0" function="0"/>
    <Var05 ind="5" vib="Time Point [Date]" value="19127/15/31"
      storage="0" tariff="0" subunit="0" function="3"/>
    <Var06 ind="6" vib="Bus Address" value="258"
      storage="0" tariff="0" subunit="0" function="0"/>
    <Var07 ind="7" vib="% BATT" value="99"
      storage="0" tariff="0" subunit="0" function="0"/>
  </Device>
```

```
<Device PrimaryAddr="" SecondaryAddr="90510717" Manufacturer="LSE"
  Version="51" Medium="Heat_Cost_Alloc." State="0">
  <Var01 ind="1" vib="Heat Cost Allocator" value="289"
    storage="0" tariff="0" subunit="0" function="0"/>
  <Var02 ind="2" vib="Heat Cost Allocator" value="12"
    storage="1" tariff="0" subunit="0" function="0"/>
  <Var03 ind="3" vib="Time Point [Date]" value="2015/11/30"
    storage="1" tariff="0" subunit="0" function="0"/>
  <Var04 ind="4" vib="Time Point [Date]" value="19127/15/31"
    storage="0" tariff="0" subunit="0" function="3"/>
  <Var05 ind="5" vib="Time Point [Date+Time]" value="2016/07/12,10:36"
    storage="0" tariff="0" subunit="0" function="0"/>
  <Var06 ind="6" vib="Size of Storage" value="8"
    storage="8" tariff="0" subunit="0" function="0"/>
  <Var07 ind="7" vib="Time Point [Date]" value="2016/06/30"
    storage="15" tariff="0" subunit="0" function="0"/>
  <Var08 ind="8" vib="Storage Interval [months]" value="1"
    storage="8" tariff="0" subunit="0" function="0"/>
  <Var09 ind="9" vib="Heat Cost Allocator" value="289"
    storage="15" tariff="0" subunit="0" function="0"/>
  <Var10 ind="10" vib="Heat Cost Allocator" value="289"
    storage="14" tariff="0" subunit="0" function="0"/>
  <Var11 ind="11" vib="Heat Cost Allocator" value="276"
    storage="13" tariff="0" subunit="0" function="0"/>
  <Var12 ind="12" vib="Heat Cost Allocator" value="259"
    storage="12" tariff="0" subunit="0" function="0"/>
  <Var13 ind="13" vib="Heat Cost Allocator" value="226"
    storage="11" tariff="0" subunit="0" function="0"/>
  <Var14 ind="14" vib="Heat Cost Allocator" value="207"
    storage="10" tariff="0" subunit="0" function="0"/>
  <Var15 ind="15" vib="Heat Cost Allocator" value="63"
    storage="9" tariff="0" subunit="0" function="0"/>
  <Var16 ind="16" vib="Heat Cost Allocator" value="12"
    storage="8" tariff="0" subunit="0" function="0"/>
  <Var17 ind="17" vib="Model / Version" value="2645700378675"
    storage="0" tariff="0" subunit="0" function="0"/>
  <Var18 ind="18" vib="Bus Address" value="258"
    storage="0" tariff="0" subunit="0" function="0"/>
</Device>
```

### **Achtung !**

Der Scan kann unter Umständen sehr lange dauern, wenn sehr viele M-Bus Geräte am Bus angeschlossen sind (bis zu mehrere Stunden).

## **4.7 CAN-Bus**

Noch nicht freigegeben.

## 4.8 CS-Protokoll (EN 62056-21 Mode C / EN 61107)

Tixi-Geräte unterstützen das CS-Protokoll nach EN 62056-21 Mode C (nur Datenabfrage). Dieses Protokoll wird häufig bei Stromzählern verwendet.

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

```

<External>
  Verwende Schnittstelle COM2
  <Bus _="COM" Name="BusName" protocol="DIN1107,CSData"
    type="Master"
    baud="Speed" handshake="HALF" format="DataFormat"
    BusPollrate="BusPollrate TUnit">
    <Device _="1" Address="xxxxxx" Pollrate="1s">
      <Manufacturer _="MC" acc="R" simpleType="String"/>
      <Identification _="ID" acc="R" size="16"
        simpleType="String" format=";%% "/>
      <DevID_OBISCode _="OBIS" obis="0.0.0" acc="R" size="5"
        simpleType="String" format=";%% "/>
      <DevID_DevID _="Text" obis="0.0.0" acc="R" size="8"
        simpleType="String" format=";%% "/>
      <DevID_DevIDDW _="DWord" obis="0.0.0" acc="R"
        simpleType="UInt32" format=";%% "/>
      <Cnt_OBISCode _="OBIS" obis="3.8.1" acc="R" size="5"
        simpleType="String" format=";%% "/>
      <Cnt_CounterDbl _="Count" obis="3.8.1" acc="R"
        simpleType="Double" format="F09,3;%% "/>
      <Cnt_Counter _="DWord" obis="3.8.1" acc="R"
        simpleType="UInt32" format="F09,3;%% "/>
      <Cnt_Unit _="Text" obis="3.8.1" acc="R" size="3"
        subind="1" simpleType="String" format=";%% "/>
      <Cnt_Date _="Text" obis="1.6.1" acc="R" size="8"
        simpleType="String" format=";(%%)"/>
      <OBIS_DevIDCode _="OBISLine" ind="1" acc="R" size="15"
        simpleType="String" format=";%% "/>
      <OBIS_Error _="OBISLine" ind="0" acc="R" size="15"
        simpleType="String" format=";%% "/>
    </Device>
  </Bus>
</External>

```

Die Zähler werden an eine RS485-Schnittstelle an das Tixi-Gerät angeschlossen. Alle rot dargestellten XML-Attribute sind erforderliche Attribute mit konstanten Werten.

### 4.8.1 Bus-Konfiguration

Beschreibung der konstanten (rot gekennzeichneten) Attribute:

<i>protocol</i>	Definiert das Busprotokoll: <b>"DIN1107,CSData"</b>
<i>type</i>	Definiert die Rolle des Tixi-Gerätes im Bus als <b>"Master"</b>
<i>handshake</i>	Bestimmt die Art der Flusskontrolle zwischen Tixi-Gerät und Zähler. Bei Nutzung des 2-Draht RS485 Busses muß eine Halbduplex-Verbindung konfiguriert werden, daher <b>"HALF"</b>

**COM** erforderlich  
Schnittstelle des Tixi-Gerätes, an dem der Bus angeschlossen wird.  
Werden 2 Busse definiert, müssen sich die Schnittstellen unterscheiden.

Wert	Beschreibung
COM2	RS 485 Schnittstelle auf COM2
COM <b>X</b>	Weitere RS485-Schnittstelle (falls im Tixi-Gerät vorhanden) <b>X</b> = Nummer der Schnittstelle

### Übersicht der möglichen Parameter (kursiv geschriebene Werte):

**BusName** optional, Standard ist: "" (leer). Maximal 20 alphanumerische Zeichen, darf nicht mit einer Zahl beginnen.

Definiert den Namen der TiXML-Attributgruppe, die den Bus in der "Process"-Attributgruppe repräsentiert. Der Gruppenname ist der erste Teil einer Gerätevariablen-Adresse:

Process/**Bus**/Device/Variable

Die kursiv geschriebenen Namen werden durch die Prozesskonfiguration definiert. Der fett geschriebene Name wird durch dieses Attribut definiert.

Wert	Beschreibung
"" (leer)	(Standardwert) Auto-Name. Die Attributgruppe bekommt einen automatischen Namen zugewiesen: Process/Auxn/... n automatisch generierte Indexnummer
<b>BusName</b> (nicht leer)	Die Attributgruppe erhält den angegebenen Namen: Process/MeinName/... Beispiel: <Bus Name="CS" ...> Process/CS/...

**Speed** optional, Standard: 9600  
Gültige Werte: 300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 115200

**DataFormat** optional, Standard: 7E1  
Bestimmt das Datenformat auf der seriellen Schnittstelle.  
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

**BusPollrate** optional, Standard: "200ms"

Zeit in Zeiteinheiten (siehe TUnit), die das Tixi-Gerät mindestens zwischen zwei Bus-Abfragen auf dem Bus wartet. Damit läßt sich ein Blockieren des Prozessors bei "leerer" Buskonfiguration verhindern.

Wertebereich:  
0ms ... 1073741823ms  
0s ... 1073740s  
0m ... 17894m  
0h ... 297h

TUnit            Zeiteinheit (ms, s, m, h)

#### 4.8.2 Device-Konfiguration

Die Gerätekonfiguration wird durch ein "Device" XML Element innerhalb des Bus Elements definiert und durch folgende XML-Attribute beschrieben:

```
<External>
  <Bus ...>
    <Device _="ID" Name="Alias" Address="Address" PollRate="Rate TUnit">
      <<< Konfiguration der Geräte-Variablen >>>
    </Device>
  </Bus>
</External>
```

Für jeden Zähler, der an der Schnittstelle (COM2 oder COMX) angeschlossen ist und mit dem das Tixi-Gerät kommunizieren soll, muß ein "Device"-Eintrag angelegt werden.

#### **Beschreibung der (*kursiv und fett geschrieben*) Attributwerte:**

- ID*            erforderlich, Zahlen, 0 ..255  
Bus-ID des Zählers. Die Adresse ist im Prinzip nicht erforderlich, da im CS-Protokoll keine laufenden Nummern für das Gerät benötigt werden. Sie dient der Unterscheidung, falls mehrere Devices konfiguriert werden.
- Address*      optional, Standard: "" (leer)  
String alphanumerisch und Leerzeichen, max. 30 Zeichen  
  
Kommunikationsadresse des Zählers (in der Regel die Seriennummer). Die Adresse ist nur erforderlich, wenn mehrere Zähler an einem Bus angeschlossen sind. Ist die Adresse leer (Standard) wird jedes Gerät angesprochen (siehe EN 62056-21:2002).
- Alias*           optional, Standardwert: "" (leer), max. 20 alphanumerische Zeichen, darf nicht mit einer Zahl beginnen.  
  
Definiert den Namen der TiXML Attributgruppe, die das Gerät in der „Bus“ Attributgruppe repräsentiert. Der Gruppenname ist der zweite Teil einer Gerätevariablen- adresse:

*Process/Bus/Device/Variable*

Die kursiv geschriebenen Namen werden durch die Prozesskonfiguration definiert. Der fett geschriebene Name wird durch dieses Attribut definiert.

Wert	Beschreibung
"" (leer)	(Standartwert) Auto-Name. Die Attributgruppe erhält einen automatisch generierten Namen zugewiesen: Process/BusName/DId... <i>Id</i> Wert des Attributs "_"
<i>DeviceName</i> (nicht leer)	Die Attributgruppe erhält den angegebenen Namen: Process/Bus/DeviceName... Beispiel: <Bus Name="CS" ....> <Device Name="Meter1" ....> Process/CS/Meter1/...

<i>Rate</i>	optional, Standard: „15m“, gültige Werte, siehe unten Pollrate in Zeiteinheiten (siehe TUnit), mit der das Tixi Data Gateway das Gerät abfragt. Wertebereiche siehe "BusPollrate"
<i>TUnit</i>	Zeiteinheit (ms, s, m, h)

### 4.8.3 Konfiguration der Gerätevariablen

Die Konfiguration einer Gerätevariablen wird durch ein [XML-Element](#) innerhalb des „Device“ Elements definiert zu dem die Gerätevariable gehört. Folgende [XML-Attribute](#) bestimmen die Konfiguration einer Variablen:

```

<External>
<Bus ...>
<Device ...>

<ValueName _="MC" format="Format"/>

<ValueName _="ID" sizeID="IDSize" format="Format"/>

<ValueName _="OBIS" obis="OBIS" size="Size" indVAL="OBISInd"
format="Format" />

<ValueName _="Text" obis="OBIS" size="Size" indVAL="ValInd"
subindTXT="TxtSubInd" format="Format" />

<ValueName _="Count" obis="OBIS" precision="Precision" indVAL="ValInd"
subindVAL="ValSubInd" format="Format" />

<ValueName _="DWord" obis="OBIS" indVAL="ValInd" subindVAL="ValSubInd"
exp="Exp" multip="Factor" format="Format" />

<ValueName _="Byte" obis="OBIS" indVAL="ValInd" subindVAL="ValSubInd"
exp="Exp" multip="Factor" format="Format" />

<ValueName _="OBISLine" indVAL="LineInd" size="Size" format="Format" />

</Device>
</Bus>
</External>

```

Jedes Element beschreibt einen Geräteparameterwert, der vom Zähler gelesen werden kann, um z.B. Alarmnachrichten zu generieren oder den Gerätestatus zu loggen.

Es gibt acht Klassen von Gerätevariablen. Eine Klasse fasst alle Geräteparameter zusammen, die gleiche Eigenschaften (Speicherung, Kodierung, Lesen oder Schreiben möglich) besitzen.

Die Klassenzugehörigkeit einer Gerätevariablen ist durch den Wert des Typ-Attributs „\_“ (**rot hervorgehoben**) definiert:

Variablenklasse	Beschreibung
Text	<p>Parameterwert dargestellt durch einen ASCII Text String im OBIS-Format.</p> <p>Der Wert einer „Text“ Gerätevariable wird im Tixi Data Gateway als ein C-String (simpleType=„<b>String</b>“) gespeichert. Die „Text“-Parameter besitzen eine maximale Länge, die durch das „size“ Attribut festgelegt wird.</p> <p>Das Attribut „<b>obis</b>“ kennzeichnet die Zeile im OBIS Format aus, der der Wert gelesen wird.</p> <p>Durch das Attribut „<b>indVAL</b>“ wird festgelegt welcher Wert (Klammerausdruck) aus der Zeile gelesen werden soll.</p> <p>Besteht ein Wert aus mehreren Teilen z.B. Zählerwert und Einheit (sind durch *-Zeichen getrennt) wird mit dem Attribut „subind“ der entsprechende Teil des Wertes adressiert. Ist der „<b>subindTXT</b>“ Wert 255 (Standardwert) wird der ganze Klammerausdruck gelesen, ohne Beachtung der Werteteile.</p> <p><b>Beispiel:</b>  <b>Lesen der Geräteidentifikation</b>  <code>&lt;DevID_DevID _="Text" obis="1-0:0.0.9*255" indVAL="0" subindTXT="0" size="16" /&gt;</code></p> <p>Vom Zähler gesendete Nutzdaten: <span style="float: right;">anhand der OBIS Kennzahl selektierte Zeile,</span></p> <pre>1-0:0.0.9*255(221020030000100A) 1-0:1.8.0*255(141002.0000*kWh) 1-0:96.5.5*255(00)</pre> <p style="text-align: right;">Wert der „Text“ Variablen</p> <p><b>Beispiel:</b>  <b>Lesen der Zählwerteinheit</b>  <code>&lt;Cnt_Unit _="Text" obis="1-0:1.8.0*255" size="3" indVAL="0" subindTXT="1"/&gt;</code></p> <p>Vom Zähler gesendete Nutzdaten: <span style="float: right;">anhand der OBIS Kennzahl selektierte Zeile,</span></p> <pre>1-0:0.0.9*255(221020030000100A) 1-0:1.8.0*255(141002.0000*kWh) 1-0:96.5.5*255(00)</pre> <p style="text-align: right;">Wert der „Text“ Variablen</p> <p><b>Beispiel:</b>  <b>Lesen des gesamten Zählwertes mit Einheit als Text</b>  <code>&lt;Cnt_Unit _="Text" obis="1-0:1.8.0*255" size="3" indVAL="0" subindVAL="255"/&gt;</code></p> <p>Vom Zähler gesendete Nutzdaten: <span style="float: right;">anhand der OBIS Kennzahl selektierte Zeile,</span></p> <pre>1-0:0.0.9*255(221020030000100A) 1-0:1.8.0*255(141002.0000*kWh) 1-0:96.5.5*255(00)</pre> <p style="text-align: right;">Wert der „Text“ Variablen</p> <p><b>Beispiel:</b>  <b>Lesen des Zählwerte Zeitstempels als Text</b>  <code>&lt;Cnt_DateRAW _="Text" obis="1-0:1.8.0*255" size="10" indVAL="1"</code></p>

	<p>subind="255"/&gt;</p> <p>Vom Zähler gesendete Nutzdaten: anhand der OBIS Kennzahl selektierte Zeile,</p> <p>1-0:0.0.9*255(221020030000100A)</p> <p>1-0:1.8.0*255(141002.0000*kWh)(0504010000)</p> <p>1-0:96.5.5*255(00)</p> <p>Wert der „Text“ Variablen</p>
<p><b>OBIS</b></p>	<p>OBIS Code dargestellt durch einen ASCII Text String im OBIS-Format. Der Wert einer „OBIS“ Gerätevariable wird im Tixi Data Gateway als ein C-String (simpleType=„String“) gespeichert. Die „OBIS“-Parameter besitzen eine maximale Länge, die durch das „size“ Attribut festgelegt wird.</p> <p>Das Attribut „obis“ kennzeichnet die Zeile im OBIS Format aus, der der Wert gelesen wird.</p> <p>Durch das Attribut „ind“ wird festgelegt welcher Teil des OBIS Codes gelesen werden soll (Standard ist 0 = kompletter OBIS Code).</p> <p><b>Beispiel:</b>  <b>Lesen des ganzen OBIS Codes</b>  &lt;DevID_OBIS _="OBIS" obis="1-0:0.0.9*255" indVAL="0" size="16" /&gt;</p> <p>Vom Zähler gesendete Nutzdaten: anhand der OBIS Kennzahl selektierte Zeile,</p> <p>1-0:0.0.9*255(221020030000100A)</p> <p>1-0:1.8.0*255(141002.0000*kWh)</p> <p>1-0:96.5.5*255(00)</p> <p>Wert der „OBIS“ Variablen</p> <p><b>Beispiel:</b>  <b>Lesen des (optionalen) ersten Teils(Medium)</b>  &lt;Cnt_Medium _="OBIS" obis="1-0:1.8.0*255" size="3" indVAL="1"/&gt;</p> <p>Vom Zähler gesendete Nutzdaten: anhand der OBIS Kennzahl selektierte Zeile.</p> <p>1-0:0.0.9*255(221020030000100A)</p> <p>1-0:1.8.0*255(141002.0000*kWh)</p> <p>1-0:96.5.5*255(00)</p> <p>Wert der „OBIS“ Variablen</p> <p><b>Beispiel:</b>  <b>Lesen des mittleren (obligatorischen) Teils</b>  &lt;Cnt_OBISID _="OBIS" obis="1-0:1.8.0*255" size="5" indVAL="2" /&gt;</p> <p>Vom Zähler gesendete Nutzdaten: anhand der OBIS Kennzahl selektierte Zeile,</p> <p>1-0:0.0.9*255(221020030000100A)</p> <p>1-0:1.8.0*255(141002.0000*kWh)</p> <p>1-0:96.5.5*255(00)</p> <p>Wert der „OBIS“ Variablen</p> <p><b>Beispiel:</b>  <b>Lesen des dritten(optionalen)Teils (Vorwert)</b>  &lt;Cnt_VV _="OBIS" obis="1-0:1.8.0*255" size="3" indVAL="3" /&gt;</p> <p>Vom Zähler gesendete Nutzdaten: anhand der OBIS Kennzahl selektierte Zeile,</p>

	<p>1-0:0.0.9*255(221020030000100A)  1-0:1.8.0*255(141002.0000*kWh)(05040100000)  1-0:96.5.5*255(00) Wert der „OBIS“ Variablen</p>
<b>Count</b>	<p>Zählerwert dargestellt durch eine Festkommazahl im OBIS Format. Der Wert einer „Count“ Gerätevariable wird im Tixi Data Gateway als eine Double Precision Floatingpoint (simpleType=„<b>Double</b>“) Zahl gespeichert.</p> <p><b>Anmerkung:</b>  Count Variablen sollen zur Speicherung von Zählerständen dienen. Die Zählerstände besitzen bis zu 10 Stellen. Eine solche Zahl lässt sich nicht in einem DWORD Integerwert speichern (maximal 9 Stellen). Double Werte erreichen die geforderte Genauigkeit (maximal 16 Stellen).  Da für weitere Berechnungen im Tixi Data Gateway nur eine Integer Arithmetik zur Verfügung steht, muss der Double Wert mit dem „<b>precision</b>“ Attribut in einen Integerwert umgewandelt werden. Dabei können Genauigkeitsverluste auftreten.</p> <p>Das Attribut „<b>obis</b>“ kennzeichnet die Zeile im OBIS Format aus, der der Wert gelesen wird.  Durch das Attribut „<b>indVAL</b>“ wird festgelegt, welcher Wert (Klammerausdruck) aus der Zeile gelesen werden soll (Standard ist 0).  Besteht ein Wert aus mehreren Teilen, z.B. Zählerwert und Einheit, wird mit dem Attribut „<b>subindVAL</b>“ der entsprechende Teil des Wertes adressiert. Der Standardwert ist 0, so dass ohne Angabe von „ind“ und „subind“ immer der Zählerwert gelesen wird.</p> <p><b>Beispiel:</b>  <b>Lesen des Zählerwertes.</b>  &lt;Cnt_Counter _="Count" obis="1-0:1.8.0*255" indVAL="0" subindVAL="0" /&gt;</p> <p>Vom Zähler gesendete Nutzdaten:</p> <p>1-0:0.0.9*255(221020030000100A) anhand der OBIS Kennzahl selektierte Zeile.  1-0:1.8.0*255(141002.0000*kWh) Wert der „Count“-Variablen  1-0:96.5.5*255(00)</p>
<b>Byte</b>	<p>Parameterwert dargestellt durch eine ASCII codierte Integerzahl (maximal 3 Ziffern) im OBIS Format. Der Wert einer „Byte“ Gerätevariable wird im Tixi Data Gateway als eine Bytezahl (simpleType=„<b>UInt8</b>“) gespeichert.</p> <p>Das Attribut „<b>obis</b>“ kennzeichnet die Zeile im OBIS Format aus, der der Wert gelesen wird.  Durch das Attribut „<b>indVAL</b>“ wird festgelegt, welcher Wert (Klammerausdruck) aus der Zeile gelesen werden soll (Standard ist 0).  Besteht ein Wert aus mehreren Teilen, z.B. Zählerwert und Einheit, wird mit dem Attribut „<b>subindVAL</b>“ der entsprechende Teil des Wertes adressiert. Der Standardwert ist 0, so dass ohne Angabe von „indVAL“ und „subindVAL“ immer der Zahlenwert gelesen wird.</p> <p><b>Beispiel:</b>  <b>Lesen des Statuswertes als Bytewert.</b>  &lt;State_State _="Byte" obis="1-0:96.5.5*255" /&gt;</p>

	<p>Vom Zähler gesendete Nutzdaten:</p> <p>1-0:0.0.9*255(221020030000100A) <span style="float:right">anhand der OBIS Kennzahl selektierte Zeile.</span></p> <p>1-0:1.8.0*255(141002.0000*kWh)</p> <p>1-0:96.5.5*255(00) <span style="float:right">Wert der „Byte“ Variablen</span></p>
<b>DWord</b>	<p>Parameterwert dargestellt durch eine ASCII codierte Integerzahl (maximal 10 Ziffern) im OBIS Format.</p> <p>Der Wert einer "Byte" Gerätevariable wird im Tixi Data Gateway als eine Double-Word-Zahl (simpleType="Uint32" siehe Kapitel 2.1 des <a href="#">SPS-TiXML-Handbuch.pdf</a>) <b>gespeichert</b>.</p> <p>Das Attribut „<b>obis</b>“ kennzeichnet die Zeile im OBIS Format aus, der der Wert gelesen wird.</p> <p>Durch das Attribut „<b>indVAL</b>“ wird festgelegt welcher Wert (Klammerausdruck) aus der Zeile gelesen werden soll (Standard ist 0). Besteht ein Wert aus mehreren Teilen, z.B. Zählerwert und Einheit, wird mit dem Attribut „<b>subindVAL</b>“ der entsprechende Teil des Wertes adressiert. Der Standardwert ist 0, so dass ohne Angabe von „indVAL“ und „subindVAL“ immer der Zahlenwert gelesen wird.</p> <p><b>Beispiel:</b>  <b>Lesen des Zählwerte Zeitstempels als Zahlenwert</b>  <code>&lt;Cnt_Date _="Text" obis="1-0:1.8.0*255" indVAL="1" /&gt;</code></p> <p>Vom Zähler gesendete Nutzdaten: <span style="float:right">anhand der OBIS Kennzahl selektierte Zeile</span></p> <p>1-0:0.0.9*255(221020030000100A)</p> <p>1-0:1.8.0*255(141002.0000*kWh)(0504010000) <span style="float:right">Wert der „DWord“ Variablen</span></p> <p>1-0:96.5.5*255(00)</p>
<b>OBISLine</b>	<p>Speichert eine Zeile des OBIS Formats als ASCII Text. Diese Variablen Klasse soll es ermöglichen den vom Zähler gesendeten OBIS-Text im Original an ein Abrechnungssystem weiterzuleiten.</p> <p>Die „OBISLine“-Parameter besitzen eine maximale Länge, die durch das „size“ Attribut festgelegt wird.</p> <p>Durch das Attribut „indVAL“ wird die Nummer der Zeile definiert, die gespeichert werden soll.</p> <p><b>Beispiel:</b>  <b>Lesen der OBIS Zeile mit dem Zählwert</b>  <code>&lt;OBIS_Cnt _="OBISLine" size="40" indVAL="1" /&gt;</code></p> <p>Vom Zähler gesendete Nutzdaten: <span style="float:right">anhand des „ind“ Attributes selektierte Zeile.</span></p> <p>1-0:0.0.9*255(221020030000100A)</p> <p>1-0:1.8.0*255(141002.0000*kWh) <span style="float:right">Wert der „OBISLine“ Variablen</span></p> <p>1-0:96.5.5*255(00)</p>
<b>MC</b>	<p><b>Manufacturer Code</b> dargestellt durch drei ASCII Zeichen im CS-Protokoll. Der Wert einer "MC" Gerätevariable wird im Tixi Data Gateway als ein C-String (simpleType="String") gespeichert. Die Stringlänge ist fix auf <b>3 Zeichen</b> eingestellt.</p> <p>Der Wert einer „MC“ Gerätevariablen wird auf folgende Weise aus dem CS-Protokoll Telegramm ermittelt:</p>

<b>ID</b>	<p><b>Identification</b> dargestellt durch maximal 16 ASCII Zeichen im CS-Protokoll. Der Wert einer „ID“ Gerätevariable wird im Tixi Data Gateway als ein C-String (simpleType=„<b>String</b>“ siehe Kapitel 2.1 des <a href="#">SPS-TiXML-Handbuch.pdf</a>) <b>gespeichert</b>. Die „ID“-Parameter besitzen eine maximale Länge, die durch das „size“ Attribut festgelegt wird.</p> <p>Der Wert einer „ID“ Gerätevariablen wird auf folgende Weise aus dem CS-Protokoll Telegramm ermittelt:</p>

Jede Klasse hat ihr eigenes Set an Attributen.

**Beschreibung der (*kursiv und fett geschriebenen*) Attributwerte:**

*ValueName* Name der Gerätevariablen maximal 21 Zeichen, nur alphanumerische Zeichen, keine Umlaute, darf nicht mit einer Zahl beginnen. Bildet den letzten Teil der Gerätevariablen-Adresse:

*Process/Bus/Device/VariableName*

Die kursiv geschriebenen Namen werden durch die Prozesskonfiguration definiert. Der fett geschriebene Name wird durch diesen Wert definiert.

Beispiel:

```
<Bus Name="CS" ... >
  <Device Name="Meter1" ... >
    <Cnt_Counter...../>
```

Process/ CS /Meter1/Cnt\_Counter

*OBISInd* optional, Standardwert: 0 Dezimalzahl (0...255)  
Kennzeichnet den **Index eines OBIS Kennzahl Teils** in einer OBIS-Format Zeile.

Die Kennzahl hat folgenden Aufbau:

Medium A	Kanal B	Messgröße C	Messart D	Tarif E	Vorwert F	Daten
-------------	------------	----------------	--------------	------------	--------------	-------

Die Felder A+B und F sind optional.  
Damit besteht die Kennzahl aus drei Teilen, die mit diesem Attribut adressiert werden können. Der Standardwert 0 liefert die komplette (empfangene) Kennzahl.

**Beispiel:****Einzelne Teile**

1-0:1.8.0\*255(141002.0000\*kWh)(0504010000)

Wert ind="2"  
Wert ind="1"

Die Zählung beginnt mit dem ersten Teil bei 1.

**ganze Kennzahl**

1-0:1.8.0\*255(141002.0000\*kWh)(0504010000)

Wert ind="0"

*ValInd* optional, Standardwert: 0 Dezimalzahl (0...255)

Kennzeichnet den **Index eines Werts** (Klammerausdruck) in einer OBIS-Format Zeile.

**Beispiel:**

1-0:1.8.0\*255(141002.0000\*kWh)(0504010000)

OBIS  
Kennzahl

Wert ind="0"

Wert ind="1"

Die Zählung beginnt mit dem ersten Klammerausdruck bei 0.

*ValSubInd* optional, Standardwert: 0 Dezimalzahl (0...255)

Kennzeichnet den **Index eines Werteteils** (innerhalb eines Klammersausdrucks) in einer OBIS-Format Zeile. Ein Wert kann aus mehreren Teilen bestehen, die durch das „\*“-Zeichen oder durch ein Leerzeichen getrennt sind. Diese Teile können mit diesem Index einzeln adressiert werden.

**Beispiel:**

1-0:1.8.0\*255(141002.0000\*kWh)(0504010000)

OBIS Kennzahl      Wert subind="0"  
Wert subind="0"      Wert subind="1"

Die Zählung beginnt mit dem ersten Teil eines Wertes (Klammerausdruck) bei 0.

*TxtSubInd* optional, Standardwert: 255 Dezimalzahl (0...255)

Kennzeichnet den **Index eines Werteteils** (innerhalb eines Klammersausdrucks) in einer OBIS-Format Zeile oder den **gesamten Klammersausdruck**. Ein Wert kann aus mehreren Teilen bestehen, die durch das „\*“-Zeichen oder durch ein Leerzeichen getrennt sind. Diese Teile können mit diesem Index einzeln adressiert oder durch den Index **255 zusammengefasst** werden.

**Beispiel:**

**Einzeladressierung**

1-0:1.8.0\*255(141002.0000\*kWh)(0504010000)

OBIS Kennzahl      Wert subind="0"  
Wert subind="0"      Wert subind="1"

**Beispiel:**

**Gesamtadressierung**

1-0:1.8.0\*255(141002.0000\*kWh)(0504010000)

OBIS Kennzahl      Wert subind="255"  
Wert subind="255"

Die Zählung beginnt mit dem ersten Teil eines Wertes (Klammerausdruck) bei 0.

*LineInd* optional, Standardwert: 0, Zahlenwert 0...255

Bedeutung nur für Variablen der Klasse **OBISLine**. Beschreibt den Index der Zeile in den Nutzdaten des CS-Protokolls.

**Beispiel:**

Vom Zähler gesendete Nutzdaten:      Zeile mit ind="0"  
1-0:0.0.9\*255(221020030000100A)      Zeile mit ind="1"  
1-0:1.8.0\*255(141002.0000\*kWh)      Zeile mit ind="2"  
1-0:96.5.5\*255(00)

Die Zählung beginnt mit der ersten Zeile in den Nutzdaten bei 0.

*Size* erforderlich, 0...100  
Bedeutung nur für Variablen der Klasse **Text**. Beschreibt die maximale Anzahl der ASCII Zeichen im OBIS-Format.

*IDSize* Optional, Standardwert ist „16“, 1...16  
Bedeutung nur für Variablen der Klasse **ID**. Beschreibt die maximale Anzahl der ASCII Zeichen im CS-Protokoll.

*OBIS* erforderlich, OBIS-Kennzahl Suchmaske  
OBIS Kennzahl zur Adressierung der Zeile im OBIS-Format.

Die Kennzahl hat folgenden Aufbau:

Medium A	Kanal B	Messgröße C	Messart D	Tarif E	Vorwert F	Daten

Die Felder A+B und F sind optional.

Zur Ermittlung der referenzierten Zeile wird die vom Zähler gesendete Kennzahl mit der im „obis“ Attribut angegebene Kennzahl verglichen. Beide Kennzahlen sind gleich, wenn sie in den Feldern C,D und E übereinstimmen. Es kann weiterhin durch Angabe weiterer Felder die Auswahl präzisiert werden.

Wird kein Vorwert (Sektion F) angegeben, so wird der erste Wert gefunden, der mit dem restlichen Angaben korreliert.

Mit **#0** als Vorwert wird der erste Wert gefunden, der keinen Vorwert besitzt, jedoch bei dem die restlichen Angaben korrelieren.

Mit **#n**, wobei **n** eine Zahl zwischen 1 und 255 ist, findet den Wert mit dem n-ten Vorwert.

Beispiele:

1-1:F.F(00000000)

1-1:0.0.0(00007971)

1-1:0.0.1(93929337)

1-1:0.9.1(024102)

1-1:0.9.2(070906)

1-1:0.1.0(01)

1-1:1.2.1(000.000\*kW)

1-1:1.2.2(000.000\*kW)

1-1:1.5.0(0.000\*kW)

1-1:1.6.1(0.000\*kW)(0000000000)

1-1:1.6.1\*01(0.000\*kW)(0000000000)

1-1:1.6.1\*00(0.000\*kW)(0000000000)

1-1:1.6.1\*00(0.000\*kW)(0000000000)

1-1:1.6.1\*00(0.000\*kW)(0000000000)

1-1:0.0.0

**0.0.0**

1-1:0.0.0\*#0

0.0.0\*#0

1-1:1.6.1

1.6.1

**1-1:1.6.1\*#0**

1.6.1\*#0

1-1:1.6.1\*00

1.6.1\*00

1-1:1.6.1\*#2

**1.6.1\*#2**

Links: Datenausgabe des Zählers. Die gefundenen Zeilen sind umrahmt.

Rechts: OBIS-Kennzahl Suchmaske. Die besten Suchmasken sind fett geschrieben.

*Format* Optional, Standardwert ist „“ (keine Formatierung).  
Die Formatoptionen kennzeichnen die Standardformatierung des Wertes, wie

sie bei Ausgaben z.B. in E-Mails oder beim Get Befehl (siehe "TiXML Reference Manual") erfolgt.

*Factor* optional, Standardwert: 1/1, Format: siehe unten

Der vom Zähler empfangene Wert (valueDevice) wird mit diesem Faktor multipliziert :

$$\begin{aligned} \text{valueTDG} &= \text{Factor} * \text{valueDevice} \\ \text{valueDevice} &= 1/\text{Factor} * \text{valueTDG} \end{aligned}$$

Der Faktor wird als Bruch geschrieben, z.B.: „1/1000“ oder „3600/1“, Zähler und Nenner müssen positive Ganzzahlen sein und dürfen nicht 0 sein.

*Exp* optional, Standardwert: 0, erlaubte Werte: siehe untenstehende Tabelle

Exponent der Basis 10, der die Genauigkeit einer Festkommazahl definiert. Der im Tixi Data Gateway gespeicherte Wert wird mit  $10^{\text{Exp}(\text{Exp})}$  (nach Anwendung des Faktors), zum Variablenwert multipliziert .

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

Der Exponent definiert damit die Position des Kommas in der Festkommazahl.

Folgende Werte sind möglich:

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

*Precision* optional, Standardwert:0, gültige Werte: siehe folgende Tabelle

Der im Tixi Data Gateway 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

#### 4.8.4 Fehlerwert einer Variablen

Jede konfigurierte Gerätevariable hat einen zugehörigen Fehlerzustandswert. Der Fehlerzustandswert enthält einen zweigliedrigen Fehlercode, der den genauen Fehlerzustand und die Fehlerquelle beschreibt. Der Fehlerzustandswert kann verwendet werden, um Konfigurationsfehler aufzudecken oder um Alarme oder Logeinträge zu erzeugen, wenn ein Kommunikationsfehler entsteht.

Sollte eine Variable nicht gelesen werden können, weil Kommunikations- oder Anwendungsfehler auftreten, wird der Fehlerzustandswert mit dem entsprechenden Fehlercode versehen und der Variablenwert auf „**undefiniert**“ gesetzt.

Um den Fehlerzustandswert einer Variablen zu lesen, wird eine Erweiterung des „Get“ Kommandos verwendet:

#### Befehl:

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

#### Übersicht der möglichen Parameter (kursiv geschriebene Werte):

*Vpath*: Pfad zur Adressierung des Parameters.

*AddInfo*:

**Error** . . . . . Gibt Fehlerzustand des Wertes zurück.

AddInfo wird bei Werten, die keine Parameter von externen Geräten sind, ignoriert. Es wird der Wert des Parameters zurückgegeben.

#### Antwort:

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

*ErrorClass*: Fehler Klasse

**0**....Kein Fehler

**1**....von TAM erkannte Fehler

*ErrorValue*: Fehlerwert

ErrorClass	ErrorValue	Bedeutung
0	0	Kein Fehler
1	2	<p>Wertezeile ist nicht im Telegramm enthalten.</p> <p>Das tritt ein, wenn:</p> <p>a) Keine Zeile mit der in der Konfiguration einer Gerätevariablen definierten OBIS - Kennzahl (siehe Attribut „<b>obis</b>“) gefunden wurde.</p> <p>oder</p> <p>b) Wenn keine Zeile in dem in der Konfiguration einer Gerätevariablen definierten Zeilenindex (siehe Attribut „<b>ind</b>“ bei der Variablenklasse „OBISLine“) empfangen wurde</p>
1	3	<p>Wert ist nicht im Telegramm enthalten.</p> <p>Das tritt ein, wenn zwar die Zeile gefunden wurde, jedoch der per Attribut „<b>ind</b>“ und „<b>subind</b>“ adressierte Wert (siehe Konfiguration der Gerätevariablen) nicht gefunden wurde oder nicht gewandelt werden konnte.</p>
1	4	Ungültiges Telegramm. Es wurde ein Telegramm empfangen, das nicht die erwartete Struktur hat oder dessen Block Control Code falsch ist.
1	5	Timeout, es wurde kein vollständiges Telegramm nach der letzten Anfrage empfangen.

#### 4.9 D0-Protokoll (EN 62056-21 Mode D)

Tixi-Geräte unterstützen das CS-Protokoll nach EN 62056-21 Mode D. Dieses Protokoll wird häufig bei Stromzählern verwendet (eHZ = Elektronischer Haushaltszähler).

Die Bus-Definition und die Variablen-Definition sind fast vollständig identisch zum CS-Protokoll (siehe Kapitel 4.8), deshalb werden hier nur die Unterschiede dokumentiert.

External Gruppe der 'PROCCFG' Datenbank:

```
<External>
  <Bus _="COM" Name="BusName" protocol="DIN1107,D0"
    type="Master" baud="Speed" handshake="HALF"
    format="DataFormat" BusPollrate="BusPollrate TUnit">
```

#### Unterschiede

- Protokoll "DIN1107,D0"
- es kann nur jeweils ein Zähler pro serielle Schnittstelle angeschlossen werden (kein Bus!)

## 5 Formatieren von SPS Variablenwerten

Ohne Formatierung werden die Variablen so dargestellt, wie sie von der SPS übermittelt werden. Das Tixi-Gerät 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:           “TF+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/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet:

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

#### **\* - Auswahl Alternative      \*Value1:Text1\*Value2:Text2\*\* :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/COM2/D1/Variable"/>
```

Tixi-Gerät 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/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet (Variablenwert=90):

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

#### **Wert in Kleinbuchstaben:**

```
<Variable _="R" simpleType="UInt8" exp="0" ... format="r16"/>
```

```
<Get _="/Process/COM2/D1/Variable"/>
```

Tixi-Gerät 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/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet (Variablenwert=98765):

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

### **F - Zahlenformat**

**F** Vorzeichen **L** Leerstellen **F** Feldbreite **D** Dezimalzeichen **D** 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/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet (Wert = 123,45):

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

**Feldbreite, Leerstellen:**

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

```
<Get _="/Process/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet (Wert = 123,456):

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

**Feststellenzahl, Dezimalzeichen, Dezimalstellen, Länge :**

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

```
<Get _="/Process/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet (Wert = 3123,456):

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

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

```
<Get _="/Process/COM2/D1/Variable"/>
```

Tixi-Gerät antwortet (Wert = -3123456):

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

**Gleitkommazahl, Dezimalzeichen, Dezimalstellen, Länge:**

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

```
<Get _="/Process/COM2/D1/Variable"/>
```

Tixi-Gerät 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/COM2/D1/Variable"/>
```

Tixi-Gerät 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.

Weitere M-Bus Platzhalter aus den Data Information Fields (DIF) (ab FW 5.1.6.8):

**%N% – Storage number (DIF)**

**%T% – Tarif (DIF)**

**%S% – Subunit (DIF)**

**%F% – Function field (DIF)**

Um diese DIF-Platzhalter nutzen zu können, müssen in der LOG-Definition die betreffenden Variablen mit einer Größe von **size="17"** konfiguriert werden:

```
<D_XX_V_1 _="meterbus" path="/Process/MBus/D_XX/D_XX_V_1" size="17"
  format=";%%,%U%,%N%,%T%,%S%"/>
```

### **Beispiel:**

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

```
<Get _="/Process/COM3/D1/Var01"/>
```

Tixi-Gerät antwortet (Variablenwert=25,30, Wärmehähler Volumen):

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

## 6 Verwenden der SPS-Variablen im Tixi-Gerät

Die SPS-Variablen können im Tixi-Gerät genauso verwendet werden, wie die Tixi-Gerät 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-Gerät können diese über den Prozesspfad angesprochen werden:

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

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

Die Variablen sind im Pfad der Erweiterungskarte (z.B. COM1, COM2, COM3) 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/COM?/D?/DeviceState"/>]
```

Diese Variable zeigt den aktuellen Zustand der Kommunikation an:

SPS-Antwortet: DeviceState=1

keine Antwort: DeviceState=0

**ChangeToggle:**

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

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

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

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

## Index

- ABB series 38
- Active 71
- Adresse 10
- Adressierung 71
- Allen Bradley 38
- Alpha XL 17
- Arrays 10, 13
- ASCII Protokoll 40
- Bus 5
- BusId 5
- Busname 71
- CAN-Bus 50
- Carel 37
- Condition 8
- CS-Protokoll 51
- D0-Protokoll 65
- Definition 5
- Device 5
- DeviceState 71
- DIN 61107 65
- Easy 31
- EN 61107 51, 65
- EN 62056-21 51
- EN 62056-21D 65
- External 5
- Fehlerwerte 64
- Fehlerzustände 15
- Feldbus 39
- Format 66
- Formatierung 12, 66
- FULL 6
- Fullduplex 6
- FX 19
- Get 12
- GUF 7
- HALF 6
- Halfduplex 6
- Identification 59
- Manufacturer Code 58
- MAXADR 7
- M-Bus 47
- M-Bus Einheit (VIF) 70
- M-Bus function code (DIF) 49
- M-Bus Function field (DIF) 70
- M-Bus Hersteller 48, 49
- M-Bus Index 49
- M-Bus Logging 48
- M-Bus Medium (VIF) 48, 49, 70
- M-Bus Primäradresse 48, 49
- M-Bus Scan 49
- M-Bus Sekundäradresse 48, 49
- M-Bus Status 48, 49
- M-Bus Storage Number (DIF) 49, 70
- M-Bus Subunit (DIF) 49, 70
- M-Bus Tarif (DIF) 49, 70
- M-Bus Value 49
- M-Bus Version 48, 49
- MELSEC 19
- Meterbus 47
- Mitsubishi 17, 19
- Modbus ASCII 45
- Modbus Function Codes 44, 46
- MODBUS RTU 43
- MODBUS TCP 45
- Moeller Easy 400/600/800/MFD 31
- Moeller PS30 34
- Moeller PS4/40 34
- noDTR 6
- OBIS Code 51, 54, 55, 56, 58, 62
- OBIS Index 59
- OBIS Subindex 61
- OBIS Zeile 61
- OMRON 38
- Parameternummer 10
- Referenzen 71
- RTSCTS 6
- RTU 45
- SAIA Burgess 35
- S-Bus 35
- Set 13
- Siemens Simatic S7-200 über MPI 21
- Siemens Simatic S7-200/300/400/1200/1500 über LAN 25
- Siemens Simatic S7-300 über MPI 23
- SPS-Anbindung 5
- SPS-Status 71
- SPS-Systeme 17
- Startwert 10
- Stationen 5
- Stationsname 71
- Text Protokoll 40
- Text Subindex 61
- Tixi-Bus 39
- Trigger 71
- TS 7
- TS-Adapter 24
- Übersicht 4
- Unterstützte SPS-Systeme 17
- Variablen 8
- Variablen lesen 12
- Variablen schreiben 13
- VIPA 29
- Werte abfragen 12
- Werte schreiben 13
- XONXOFF 6
- Zugriffsrecht 9