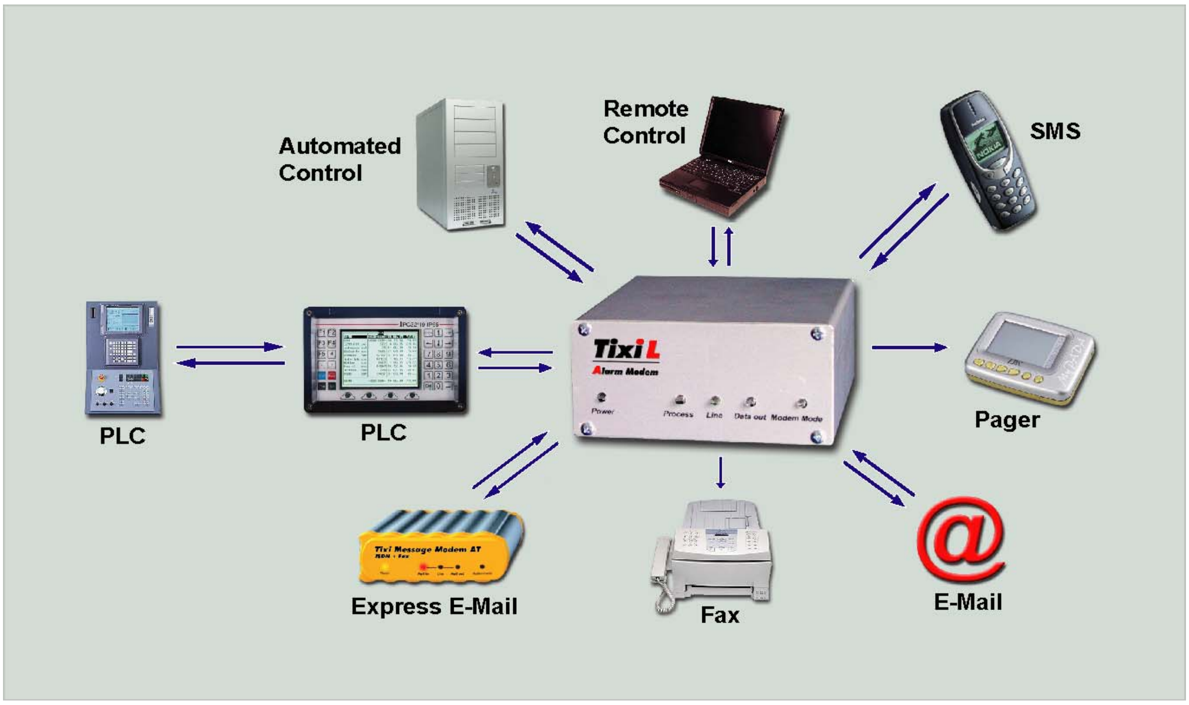
PLC-EN



Tixi PLC TiXML Manual

Tixi Alarm Modem

V 2.5

Publication close: 07. May 2007. Supported Tixi Alarm Modem Firmware: 2.2.12.0 (BETA 3.0.1.276)

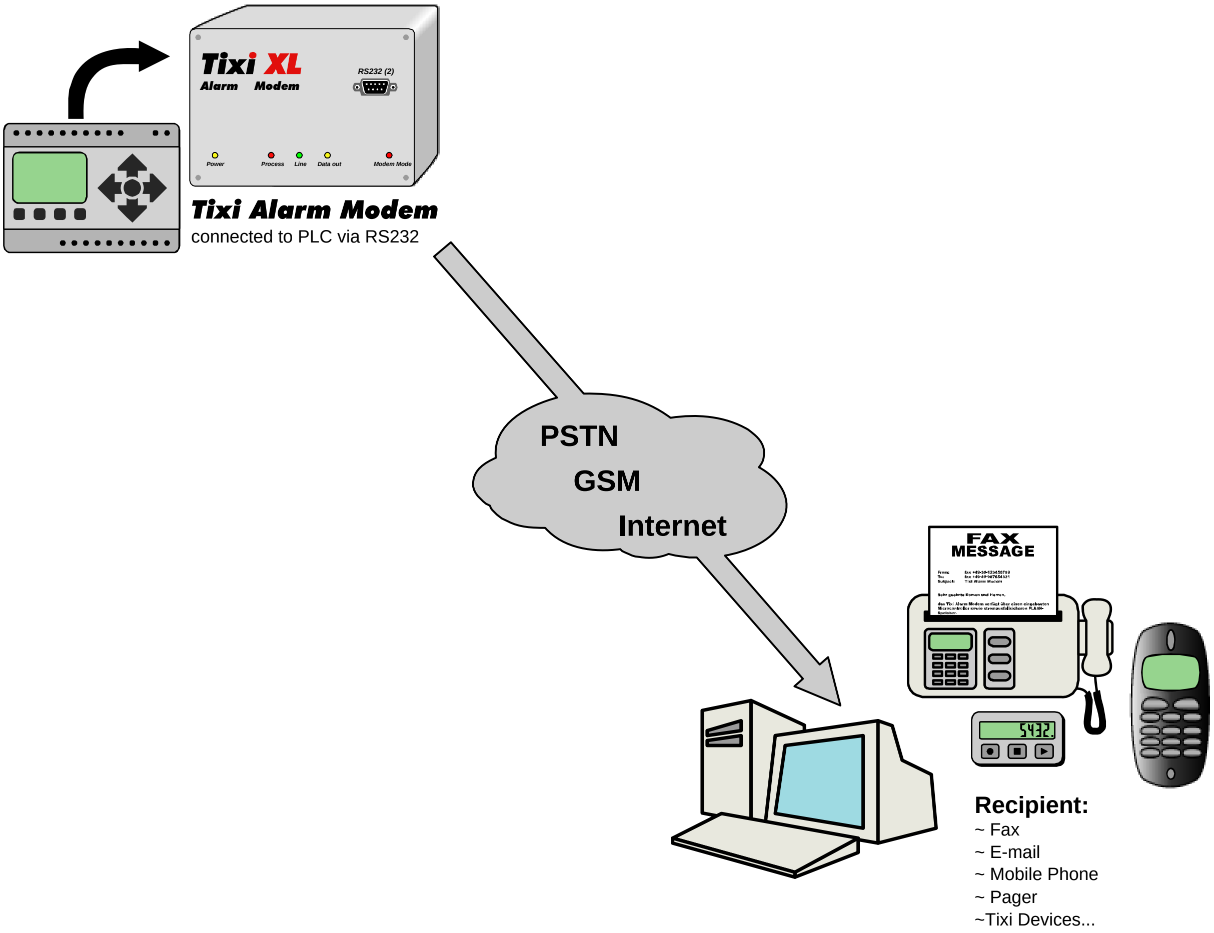
Table Of Contents

1 Overview

The **Tixi Alarm Modem** provides you with a completely new type of communication device, which can be integrated into existing systems with ease.

The communication protocols of common PLCs are already implemented into the Tixi Alarm Modem, so there's no need to change the PLC or it's programming. Other PLCs can control the Tixi Alarm Modem via simple text strings: the **TiXML-commands**.

Imagine this as a simple application of Tixi Alarm Modem:



This manual describes the settings to be configured in order to connect the Tixi Alarm Modem to a PLC. Additionally, the range of supported PLCs along with detailed info on variable range etc is provided, as well as commands for formatting of PLC variable values.

2 Configuring Tixi For Supporting PLCs

2.1 Configure PLC- and Bus system

In order to let the Tixi Alarm Modem communicate with the PLC, the *External* database of the Tixi Alarm Modem is to be configured accordingly. This description assumes that basic knowledge of TiXML and Tixi Alarm Modem configuration are already present.

The database to hold the PLC config data is `PROCCFG/External` and looks like this:

```
<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>
```

Attributes within **{...}** are option.
Attributes within **[...]** are required by some PLCs or Fieldbus only.

These are the legal values to be inserted for the italic tags:

`<Bus>-Parameters`

`Bus` Declares the interface where the PLC is connected to. Use these values:

`MB`	PLC is connected to the mainboard RS232 interface
`C0`	PLC is connected to the C0 extension card interface
`C1`	PLC is connected to the C1 extension card interface
`COM1`	PLC at COM1 (Only Hx-Modems with >=FW 1.80.0.0, requires BusId)
`COM2`	PLC at COM2 (Only Hx-Modems with >=FW 1.80.0.0, requires BusId)

`BusId` Allows to address the PLC-bus independent from the interface address. Must not be used with „Name“ attribute.

`Alias` Option to give a name for the PLC-bus which allows variable addressing independend from the PLC-interface address. Must not be used with „BusId“ attribute. (max. 20 alphanumeric characters, no special characters, must not start with a digit)

Proto Determines which PLC or Fieldbus protocol is to be used, i.e. which PLC is connected. Valid parameters can be found in chapter 3.

BType Sets the role of the TAM in the bus communication to either Master or Slave. Valid parameters can be found in chapter 3.

Speed Sets the baudrate between Tixi Alarm Modem and PLC (e.g. `19200`) or on the bus. Valid parameters can be found in chapter 3.

Dataformat

Defines the data format on the serial interface. By default the PLC specific data format (e.g. 8E1 with S7-200) or 8N1 with PLC independen protocols (Modbus RTU) is choosen. Valid parameters can be found in chapter 3.

Syntax: **DataBitsParityBitsStopBits**

DataBits:

8..8 data bits are used
7..7 data bits are used

ParityBits:

N..no parity bit
E..even parity
O..odd parity

StopBits:

1..one Stopp bit
2..two Stopp bits

Handshake

`None`	communication without handshake
`XONXOFF`	software handshake
`XONXOFFPASS`	software handshake, XONXOFF forwarded to application
`RTSCTS`	hardware handshake with RTS CTS
`DTRDSR`	hardware handshake with DTR DSR
`HALF`	Halfduplex RS 485 communication
`FULL`	Fullduplex RS 485/422
`HALFX`	Halfduplex RS 485 communication with XON XOFF
`FULLX`	Fullduplex RS 485/422 with XON XOFF
`noDTR`	disables DTR

Memory Defines, how much memory (in byte) is reserved for the PLC bus. For both bus definitions there are together 131072 byte memory available

OwnID Sets the station ID of the Tixi Alarm Modem. If this parameter is necessary and which parameters are valid can be found in chapter 3.

Range Address range to be searched fo other Master devices. If this parameter is necessary and which parameters are valid can be found in chapter 3.

Factor Gap update factor to find other slaves. If this parameter is necessary and which parameters are valid can be found in chapter 3.

Retries Amount of retries on communication errors. If this parameter is necessary and which parameters are valid can be found in chapter 3.

Until Tixi Alarm Modem Firmware 1.72.x.x only one BUS definition is possible.
Since Firmware version 1.80.x.x two different PLC systems on two interfaces are possible.
If one of both interfaces is used for MPI communication, the MPI bus has to be the first entry in the external database.

<Device>-Parameters

ID Provides the device ID of the appropriate PLC. This must correspond to the station ID set inside the PLC (e.g. 1). Some PLCs support multiple devices.

Alias Option to give a name for the PLC station which allows variable addressing independend from the station ID.

Rate Frequency (see TUnit, default is 60s) used by the Tixi Alarm Modem to query the PLC for new values in ‘master’ mode or communication timeout in ‘slave’ mode.

CT Timeout (see TUnit) between characters. If this parameter is necessary and which parameters are valid can be found in chapter 3.

Wait Idle time (see TUnit) between messages. If this parameter is necessary and which parameters are valid can be found in chapter 3.

Timeout Timeout (see TUnit) for response. If this parameter is necessary and which parameters are valid can be found in chapter 3.

AI AdressIncrement between two successive DWORDSs (2, Modbus RTU only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

Swap Has to be enabled if low word is send before high word DWORD (0, Modbus RTU only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

Funct Enable, if Funct 0x06 has to be used instead of 0x10 for single WORD writing (0, Modbus RTU only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

PA Primary address of device (M-Bus only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

SA Secondary address of device (M-Bus only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

FA Fabrication address of device (M-Bus only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

MC Device manufactor code (M-Bus only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

Gen Device generation (M-Bus only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

Med Device medium (M-Bus only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

DType PLC CPU Type (Mitsubishi FX only). If this parameter is necessary and which parameters are valid can be found in chapter 3.

Cache If set to 0, the combination of variables to a queried block (caching) will be deactivated. All variables will be queried separately. (Default: 1, only Modbus RTU). If this parameter is necessary and which parameters are valid can be found in chapter 3.

`Elements` Limit number of elements queried by a single modbus telegramm (caching). If this parameter is necessary and which parameters are valid can be found in chapter 3.

`Name` Name of condition to change the PLC pollrate.

`TUnit` Time units (ms, s, m, h)

`<Condition>-Parameters`

Conditions are used to change the PLC pollrate, e.g. to increase logging rate during system failure.

`Name` Name of condition to be referenced inside device section.

`Path` Variable to check the condition (=1). Entered without reference sign.

`Rate` Frequency used by the Tixi Alarm Modem to query the PLC for new values in ‘master’ mode. Communication timeout in ‘slave’ mode. (e.g. `1s,5m,1h`)

`<Variable>-Parameters`

Due to performance reasons we recommend to use not more than 100 variables.

Each variable within Device definition must be unique. Multiple variables with same type AND index are not allowed.

`VName` Name of the appropriate variable to be defined as follows. The name must not be longer than 20 characters.

`VType` Determines the variable type, e.g. a counter or a flag etc. The legal variable types for each PLC can be found in chapter 3 below.

`BasicType` Defines the basic type of the variable, as used by the specific application. The basic type determines the valid format possibilities and the native display within TiXML-protocol. Serveral basic types (Attribute „simplyType“) may be valid for a single variable type (Attribute _=““) which has to be choosen. For some basic types additional attributes have to be specified (see „exp“, „size“). Following values are possibles:

BasicType Value	Description	Add. Attrib.	native Display Examples
Uint8	unsigned 8 Bit value (0…255)	exp	“123” (exp = 0) “12.3” (exp = -1) “12300” (exp = 2) “0.0123” (exp = -4)
Uint16	unsigned 16 Bit value (0…65535)	exp	“12345” (exp = 0) “1234.5” (exp = -1) “1234500” (exp = 2)
Uint32	unsigned 32 Bit value (0…4294967295)	exp	“1234567” (exp = 0) “123456.7” (exp = -1) “123456700” (exp = 2)
Int8	signed 8 Bit value (-128…+127)	exp	„123“ (exp=0) „-123“ (exp=0) “1.23” (exp=-2) “12300” (exp=2)
Int16	signed 16 Bit value (-32768…32767)	exp	“-12345” (exp = 0) “ 12345” (exp = 0) “-1234.5” (exp = -1) “ 1234.5” (exp = -1) “-1234500” (exp = 2) “ 1234500” (exp = 2)

Int32	signed 32 Bit value (-2147483648…2147483647)	exp	“-1234567” (exp = 0) “ 1234567” (exp = 0) “-123456.7” (exp = -1) “ 123456.7” (exp = -1) “-123456700” (exp = 2) “ 123456700” (exp = 2)
String	text (0…size characters)	size	„This is a Text „ (only the first 100 characters)
Blob	binary data Array (0...size Byte) currently not supported	size	“AF037FFF” hex bytes (only the first 100 Bytes)
Bit	digital value (0...1)		„0“ „1“
Float	Flaoting point single precision (±3.402823466*10^{38})	exp	12.34567 -0.001234 -0.123456 E-7 0.123456 E+7 (Displayed with exponent if |Exponent| > 6)
Double	Floating point double precision (±1.7976931348623158*10^{308})	exp	12.345678910 -0.0012345678 -0.1234567890 E-11 0.1234567890 E+11 (Exponential display at |Exponent| > 11)

Rights Sets the acces rights for the specified variable:

R read access only

W write access only

RW read and write access

The valid values are depending on the variable type and PLC and can be found in chapter 3.

Storage Defines additional memory and access rights:

A (RA, WA, RWA) enables active access for S7 master

L (RWL) linked buffer (only available for S-Bus variables with RW access)

C Activates the cached access to variable blocks only if cache was deactivated by ’UseCache’ parameter. If the attribute is set, the Tixi Alarm Modem will check if the following variables may be queried with the same request. If the following variable is a different type, the query is done separately again.

Index The variable index range depends on the PLC and variable type that is used. Legal values are to be found in chapter 3 below.

Index2 The sub variable index range depends on the PLC and variable type that is used. Legal values are to be found in chapter 3 below.

Array Number of elements queried as an array.

Default Start value of the variable. With variable write access, value will be written into PLC during every modem startup. With variable read access, value will be used on modem startup until modem gets the real value from the PLC. The start value has to be given in coherence to the „exp“ and simpleType (see table Chap. 2.3).

ArraySize

1. simpleType = String *(see BasicType)*
The maximum amount of ASCII characters of a text value (only valid for simpleType=”String”, 0…65535). With zero terminated strings the zero has to be included. The value is depending on the PLC or bus (option, depending on basic type, device or bus).

2. simpleType= Blob (see *BasicType*)
The maximum amount of Bytes within a byte array (only valid for simpleType=”Blob”, 0…65535). The value is depending on the PLC or bus (option, depending on basic type, device or bus).

Factor The value received from the PLC will be multiplied by this value before processing:

valueTAM = Factor * valueDevice
valueDevice = 1/Factor * valueTAM

The factor is used a a fraction, e.g.: „1/1000“ or „3600/1“, the denominator and numarator must not be zero.
The usage of this attribute depends on the variable type. Valid values can be found in chapter 3.

Exp Exponent of base 10 to specify fix point precision of **simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (see *BasicType*)**.
The Alarm Modem value will be multiplied by 10 exp(*Exp*) (after appling *Factor*), to get the parameter value.

valueParameter = 10^{Exp} * valueTAM.

The exponent therefore specifies the position of comma within a fix point value

Following values are possible:

Exp value	Description
-6	Precision = 0,000001
-5	Precision = 0,00001
-4	Precision = 0,0001
-3	Precision = 0,001
-2	Precision = 0,01
-1	Precision = 0,1
0	Precision = 1 (default
1	Precision = 10
2	Precision = 100
3	Precision = 1000
4	Precision = 10000
5	Precision = 100000
6	precision = 1000000

Precision

Precision of integer display for a value with **simpleType = Float , Double (see BasicType)**.

The Alarm Modem value will be multiplied by 10 exp(*Exp*), to convert the value into integer. The integer will be used by calculation of ProcessVariables e.g. by using instructions (GT, LT etc.). Therefore it specifies the precision during calculating ProcessVariables and has to be configured depending on the application.

Integer = Value(10^{Exp} * valueParameter).

Following values are possible:

Precision value	Description
-6	Factor = 0,000001
-5	Factor = 0,00001
-4	Factor = 0,0001
-3	Factor = 0,001
-2	Factor = 0,01
-1	Factor = 0,1
0	Factor = 1 (default)
1	Factor = 10
2	Factor = 100
3	Factor = 1000
4	Factor = 10000
5	Factor = 100000
6	Factor = 1000000

Format With the format option PLC variables may be formatted for messages or get commands by e.g. decimal points or units added. (see chapter 5).

wFunct Function code, used to write a value. If this parameter is necessary and which parameters are valid can be found in chapter 3.

rFunct Function code, used to read a value. If this parameter is necessary and which parameters are valid can be found in chapter 3.

2.2 Reading device variables

The value of a variable defined within External database can be queried by TiXML command „Get“ as any other variable value. This command is extended by the attribute **„format“** for variables of the External database.

Command:

[<Get _=“***Vpath***“ format =”***Format***”/>]

Overview of valid parameters (italic written values):

Vpath: Path to paramater, device or bus.
Format Format specification, following values are possible:

Format value	Description
„integer“	**1. simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (see** 2 ***BasicType*).** The value is displayed as integer. The integer calculates itself by using the exponent specified with the variable and its value (see 2.1 *Exp*) : **Value as integer** = 10^{-Exp} * **value**

	2. simpleType = float, double (see 2.1 *BasicType*). The value is displayed as integer. The integer calculates itself by using the precision specified with the variable and its value (see 2.1 *Precision*): **Value as integer** = 10 $^{\textit{-Precisoin}}$ * **value** **3. all other data types (see 2.1 *BasicType*).** The value is displayed native (see 2.1 *BasicType*).
empty(„“) or “native”	The value is displayed native (see 2.1 *BasicType*).
Format string	The value is displayed according to the given format (see chapter 5).

If the „format“ attribute is missing, the value will be displayed using the default format specified within variables definition (see 2.1) . If no default format is defnined, the value will be shown native (see 2.1 *BasicType*).

Response:
[<Get _=“***Value***“ />]

Value: Value using specified format display.

2.3 Writing device variables

The value of a variable defined within `External` database can be set by TiXML command „Set“ as any other variable value. This command is extended by the attribute **„format“** for variables of the `External` database.

Command:
[<Set _=“***Vpath***“ value=”***Value***” format =”***Format***”/>]

Overview of valid parameters (italic written values):
Vpath: Path to paramater, device or bus.
Value: Value of parameter.
Format: Format specification, following values are possible:

Format value	Description
„integer“	**1. simpleType = Uint8, Uint16, Uint32, Int8, Int16, Int32 (siehe 2.1 *BasicType*).** The value has to be entered as integer. The integer is calculated using the exponent specified with the variable and its value (see 2 *Exp*) : **Value as integer** = 10 $^{\textit{-Exp}}$ * value **2. simpleType = float, double (see 2.1 *BasicType*).** The value has to be entered as integer. The integer is calculated using the precision specified with the variable and its value : **Value as integer** = 10 $^{\textit{-Precision}}$ * value **3. all other data types (see 2.1 *BasicType*).** The value has to be entered native (see 2.1, *BasicType*)
empty(„“) or “native”	The value has to be entered native (see 2.1 *BasicType*)
Format string	The value has to be antered according to the given format (**not yet available**).

If the „format“ attribute is missing, the value has to be entered native.

Response:
[<Set/>]

2.4 Arrays

If variablen within the External are defined as arrays (attribute “no“), e.g..

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

single elements of the array can be addressed by a suffix to the variable path:

2.4.1 Reading arrays

The addressed element of the array is attached in square brackets:

[<Get _=“***Vpath[element]***“/>]

Vpath: Path to address the array.
element: Addressed element within the array.

e.g. third value of the array:

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

If no suffix is specified, all array values will be separated by comma:

[<Get _=“***Vpath***“ >]

Response:

[<Get _=“Value1,Value2,Value3,…Value8“ />]

2.4.2 Writing arrays

The addressed element of the array is attached in square brackets:

[<Set _=“***Vpath[element]***“ value=”***Value***”/>]

Vpath: Path to address the array.
element: Addressed element within the Array.
Value: Value of the element.

e.g. the third value of the array:

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

If no suffix is specified, all array values has to be separated by comma. No gaps are allowed.

[<Set _=“***Vpath***“ value=”Value1,Value2,Value3,Value4,Value5,Value6,Value7,Value8”>]

Response:

[<Set/>]

2.5 Processing device error states

Variables defined within `External` database may also store different error states depending on the PLC or bus system. Especially using a plc or bus, communication or protocol errors may occur and leading to invalid values.

The error state of variables will be displayed using the error codes „ErrorClass“ and „ErrorNumber“. Both can be read, processed and used to trigger events.

2.5.1 Reading error states

If the value of a variable is invalid, whoch means an error was detected and saved, the Get command is influenced:

TiXML „Get“ command to query a **Variable group** (e.g. [<Get _=“/Process/Bus1/D2“/>]): Parameter entries with invalid values will be **omitted.**

1. TiXML „Get“ to read **a variable**:
   (e.g. [<Get _=“/Process/Bus1/D2/Language“/>)

   Within the response a TiXML error is displayed (see TiXML Error Frame in TiXML Reference Manual) :
   ErrorText: „Variable exists but does not contain data“
   ErrNo: -2194

Note:
Both commands are displaying the last successfull read value instead of the error if at least one access after uploading the variables definition or system start (reset, power on) was successfull.

For directly reading the error state of a parameter or parameter group the TiXML command „Get“ (see TiXML Reference Manual) will be extended by an attribute that displays the value of an additional information (error state) instead of displaying the parameter value:

Command:

[<Get _=“***VPath***“ AddInfo=“***AddInfo***“ />]

Overview of valid parameters (italic written values):

`Vpath:` Path to address the parameter.

`AddInfo:`
`Error ........` Displays error state of the value

The attribute „AddInfo“ will be ignored by all parameters of non external devices. The variable value will be displayed instead.
Response:

AddInfo = Error
[<Get _=“***ErrorClass,ErrorValue“*** />]
`ErrorClass:` Error class
0... no error
>0... error

`ErrorValue:` error value

ErrorClass	ErrorValue	Discription
0	**0**	No error
1	***n* > 0**	Error within access implementation of the Alarm Modem. Number *n* is > 0 and depending on PLC or bus system.
***c* >1**	***n***	Error message of external device or bus. Number *n* and *c* are depending of PLC on bus system.

Note:

The error state will be deleted by a following succesfull access to the PLC or Bus (set to 0,0).

2.5.2 Processing error states

To use an error state to trigger alarms (or other events) a special load instruction within the instruction list of a ProcessVar „<Value/>“ entry is defined (see TiXML Reference Manual):

Command:
<LDS _=“***Vpath***“ AddInfo=“***AddInfo***“ />

Overview of valid parameters (italic written values):

Description:
Reads the error state of the parameter, referenced by *Vpath* and writes following values to the processing stack:

High-part (Bit 16 - 31)	Low part (Bit 0 - 15)
Value of error class	**Value of error**

`Vpath:` Path to address the parameter.

`AddInfo:`
`ErrorCode ........`Displays error state of the value

3 PLCs supported

3.1 Mitsubishi Alpha XL

The used variables of a connected Mitsubishi Alpha XL have to be registered to the Tixi Alarm Modem variable set.
The Mitsubishi variables are registered in the External section of the **'PROCCFG**' database.

```
<External>           Use COM port on extension card 1
   <Bus _=”C1” protocol=”Mitsubishi,Alpha2” type=”Master”
   baud=”9600”>
                                                     be Alpha Master
     <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" />          set of variables
         <CW2 _="CW" ind="2" />
         <Out2 _="O" ind="2" acc="W"/>
         <ExtOut129 _="EO" ind="129" acc="W"/>
     </Device>

   </Bus>
</External>
```

The BUS-parameter contains the port address of the used extension card, the protocol manufacturer “Mitsubishi”, the kind of attached device “Alpha2”, the operation mode “Master” and the used baud rate.

The Device ID has to match the device ID of the PLC (default 0).

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For this device a set of variables can be registered:

Variable type in Alpha controller (I,O,IA...)

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

Each line declares a logical name (alias, e.g. Alarm11) and the type of the variable in the Mitsubishi PLC (see list of supported variables).
The ‘ind’ attribute defines the index of the variable in the Mitsubishi controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access depending on the type of variable.

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first writing to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported variables for Alpha XL:

Type	index	access	comment
M	1-14	R	System bits
I	1-15	R	digital input
EI	129-132	R	External inputs
O	1-9	RW	outputs
EO	129-132	RW	External outputs
K	1-8	R	Buttons
E	1-4	R	Link inputs
A	1-4	RW	Link Outputs
N	1-4	RW	Control bits
AI	1-8	R	Analog input
CB	1-100	RW	Bit-operands functions
CW	1-100	RW	Word-operands functions

If access type ‚R' is possible only, the parameter ‚acc' can be omitted.
If the PLC is in "RUN" mode, some variables are overwritten by the PLC program. In that case connect coils with the variables and sets the corresponding coil.

Note: If the automatically created DeviceState (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**) is "1" but no variable values are received, some of the defined variables do not exist in the PLC itself or some variables defined in the PLC do not exist in the modem. Each variable defined in the External-Database has to exist in the PLC too.

Connecting a Mitsubishi Alpha XL Hardware
Tixi Alarm Modem has to be connected to a Alpha XL using the Mitsubishi GSM-CAB.

Please pay attention on these hints:

1. There must be a project on the Alpha XL with activated "serial communication" set to 9600/8N1 (see Alpha Programming software online help). After activating this setting, the Alpha XL has to be restarted.
2. You can connect the GSM-CAB directly to the Tixi Alarm Modem mainboard RS232 (MB).
3. If you connect the GSM-CAB to a Tixi Alarm Modem extension card RS232-2 (C1), you have to use a null modem cable between Alarm Modem and GSM-CAB.

Remote Control
For remote access to the PLCs following "TransMode" commands are necessary (see TiXML-Reference manual for more informations)

Alpha XL:

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

(use com="C1" if connected to extension card RS232-2)

3.2 Mitsubishi MELSEC FX

The used variables of a connected Mitsubishi MELSEC FX controller have to be registered to the Tixi Alarm Modem variable set.
The Mitsubishi variables are registered in the External section of the **'PROCCFG**' database.

```
<External>
   <Bus _=”C1” protocol=”Mitsubishi,Format1” type=”Master”
   baud=”9600”>

     <Device _=”0” Pollrate=”1s” devType=”FX1N”>
         <Input1 _="X" ind="1"/>
         <M1 _="M" ind="1" acc="R"/>
         <Timer _="TS" ind="1" acc="R" />
         <Counter _="CS" ind="3" acc="R" />
         <CN2 _="CN" ind="2" />
         <Out2 _="Y" ind="2" acc="W"/>
     </Device>

   </Bus>
</External>
```

Use COM port on extension card 1

be Alpha Master

Set of variables

The BUS-parameter contains the port address of the used extension card, the protocol manufacturer “Mitsubishi”, the kind of attached device “FX”, the operation mode “Master” and the used baud rate.

The Device ID has to match the device ID of the PLC (default 0).

The Format1 protocol supports networks (RS485), therefore it is possible to connect devices with different device IDs.
Parameter “devType” specifies the CPU type: FX1S, FX1N or FX2N.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For this device a set of variables can be registered:

Variable type in FX controller (Y,X,M...)

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

Each line declares a logical name (alias, e.g. Alarm11) and the type of the variable in the Mitsubishi PLC (see list of supported variables).
The ‘ind’ attribute defines the index of the variable in the Mitsubishi controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access depending on the type of variable.

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first writing to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported variables MELSEC FX “Format1 protocol”:

Type	index	access	comment
M	0-8255	RW	Flag
Y	0-317	RW	Output (Octal)
X	0-317	R	Input (Octal)
S	0-999	RW	System Flag
CS	0-255	RW	Counter Switch
TS	0-255	RW	Timer Switch
TN	0-255	RW	Timer Value unsigned
TNI	0-255	RW	Timer Value signed
CN	0-199	RW	Counter Value (16Bit)
CD	200-255	RW	Counter Value (32Bit) unsigned
CDI	200-255	RW	Counter Value (32Bit) signed
D	0-8255	RW	Register (16 Bit) unsigned
DI	0-8255	RW	Register (16 Bit) signed
DW	0-8254	RW	Register (32 Bit) unsigned
DWI	0-8254	RW	Register (32 Bit) signed

If access type ‚R’ is possible only, the parameter ‚acc’ can be omitted.
If the PLC is in “RUN” mode, some variables are overwritten by the PLC program. In that case connect coils with the variables and sets the corresponding coil.

Additionally to the open protocol „Format1“ the FX-internal protocol is available. It may be activated using the bus parameter `protocol=”Mitsubishi,FX”.` This protocol has no network capability but offers additional access to the contacts und resets of timers/counters:

Additionally supported variables MELSEC FX “FX protocol”:

Type	index	access	comment
CC	0-255	R	Counter Coil
TC	0-255	R	Timer Coil
CR	0-255	R	Counter Reset
TR	0-255	R	Timer Reset

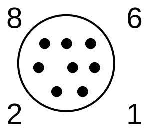
Connecting a Mitsubishi FX Hardware

Tixi Alarm Modem can be connected to a FX internal RS422 interface or via additional interface extension RS232-BD / RS422-BD / RS485-BD card.
If using a BD extension card, this interface must be activated using the GX developer software with parameter 9600/7E1 (parity “even”!).
Both interfaces may be used simultaneously to connect an Alarm Modem and e.g. a touch panel at the same time.

RS422 connection:

Tixi Alarm Modem	MELSEC FX
(RS422 extension card)	(8pin female)

T+ --- 2
T - --- 1
R- --- 4
R+ --- 7



RS485 connection:

```
Tixi Alarm Modem                                        MELSEC FX RS485-BD
(RS485 extension card)                               (5pol screw interface)
          T+  ------------------------------------------------------ RDA
          T -  ------------------------------------------------------ RDB
          R-  ------------------------------------------------------ SDB
          R+  ------------------------------------------------------ SDA
          0V ------------------------------------------------------- SG
```

Remote Control
For remote access to the PLCs following “TransMode” commands are necessary (see TiXML-Reference manual for more informations)

MELSEC FX:

Tixi Alarm Modem RS232 interface:
`[<TransMode baud="9600" format="7E1" com="MB"/>]`
(use `com=”C1”` if connected to extension card RS232-2)

Tixi Alarm Modem RS422 interface:
`[<TransMode baud="9600" format="7E1" handshake=”FULL” com="C0"/>]`

3.3 Siemens Simatic S7-200

The variables of a connected Simatic S7-200 controller have to be registered to the Tixi Alarm Modem variable set.
The variables are registered in the External section of the '**PROCCFG**' database.

```
<External>          Use RS485 port on extension card 0
   <Bus _=”C0” protocol=”Siemens,S7-200” type=”Master”
   baud=”9600” handshake=”HALF” TS=”0” MAXADR=”15” GUF=”1”
   RC=”1”>

      <Device _=”2” Pollrate=”1s”>
         <M30 _="M" ind="30" acc="R" />
      </Device>                                      Station #2
      <Device _=”3” Pollrate=”60s”>                   Station #3
         <V100 _="V" ind="100" acc="R" />
         <VS50 _="VS" size=”4” ind="100" acc="R" />
      </Device>
   </Bus>
</External>
```

The BUS-parameter contains the port address of the used extension card, the protocol manufacturer “Siemens”, the kind of attached device “S7-200”, the operation mode “Master”, the used baud rate and the necessary handshake (if using RS485 extension card). TS is the station ID of the device itself, MAXADR the scanned station ID range, GUF the “gap update factor” to find other slaves and RC the amount of retries on communication errors.

Tixi Alarm Modem can only be master in communication with a S7-200. The default baud rate is 9600.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the PPI communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For each controller a ‘Device’ entry has to be inserted, which contains the station-Id of the controller (‘_’ – attribute) and the poll rate. In the period set with the poll rate attribute the scanning of the S7-200 is restarted.

For each device a set of variables can be registered:

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

Variable type in S7-200 controller

Each line declares a logical name (alias, e.g. AlarmM10), the type of the variable see list below).

The ‘ind’ attribute defines the index of the variable in the S7-200 controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ – read, ‘W’ – write or ‘RW’ for read-write access respectively.
You will have to add an additional special access type ‘A’ which enables active access for the master, e.g. ‘RWA’.

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first write operation to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported Variables for S7-200 CPUs

Type	index	access	comment
V	0-10239.7	R/W	Variable Memory (Bit)
VB	0-10239	R/W	Variable Memory (Byte)
VW	0-10238	R/W	Variable Memory (Word)
VD	0-10236	R/W	Variable Memory (DoubleWord)
VS	0-10239	R/W	Variable Memory (string, requires parameter “size”)
I	0-15.7	R	Input Register
Q	0-15.7	R/W	Output Register
M	0-31.7	R/W	Bit Memory
MB	0-31	RW	Byte Memory
MW	0-30	RW	Word Memory
MD	0-28	RW	DWord Memory
SM	0-549.7	R	Special Memory
S	0-31.7	R/W	Sequential Control Relay
T	0-255	R/W	Timer
C	0-255	R/W	Counter
AI	0-62	R	Analog input
AQ	0-62	R/W	Analog output
HC	0-5	R	High speed counters

Connecting a S7-200 Hardware

The S7-200 may be connected via PPI-Cable (RS232) or via Profibus-Adapter (RS485 extension card).

```
Tixi Alarm Modem                          S7-200
(RS485 extension card)               (Profibus-Adapter)
            T+  ------------------------------------B1
            T-   ------------------------------------A1
```

Remote Control

For remote access to the PLC following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

Tixi Alarm Modem RS232 interface:
`[<TransMode baud="9600" format="8E1" handshake="noDTR" com="MB"/>]`
(use `com=”C1”` if connected to extension card RS232-2)

Tixi Alarm Modem RS485 interface:
`[<TransMode baud="9600" format="8E1" handshake="HALF" com="C0"/>]`

To connect to a S7-200 a 11-Bit modem (supporting 8E1 data format) has to be used on local side. The timing of the MicroWin software has to be modified using a registry patch delivered on the Tixi CD.

3.4 Siemens Simatic S7-300/400

The variables of a connected Simatic S7-300/400 controller have to be registered to the Tixi Alarm Modem variable set.
The variables are registered in the External section of the '**PROCCFG**' database.

```
<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>
```

The BUS-parameter contains the port address of the used extension card, the protocol manufacturer “Siemens”, the kind of attached device “S7-300/400-A” and the operation mode “Master” or “Slave”. TS is the station ID of the device itself, MAXADR the scanned station ID range, GUF the “gap update factor” to find other slaves and RC the amount of retries on communication errors.

Tixi Alarm Modem can be master or slave in communication with a S7-300/400.

For each controller a ‘Device’ entry has to be inserted, which contains the station-Id of the controller (‘_’ – attribute) and the poll rate. In the period set with the poll rate attribute the scanning of the S7-300/400 is restarted (master mode) or a communication timeout is detected (slave mode).

If an external definition is loaded, the TS-Adapter parameters set by Siemens Teleservice software will be disabled.

For each device a set of variables can be registered:

Variable type in S7-300 controller

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

Datablock

Each line declares a logical name (alias, e.g. AlarmM10), the type of the variable see list below).
The 'ind' attribute defines the index of the variable in the S7-200 controller and the 'acc' attribute the access type. The access type can be either 'R' – read, 'W' – write or 'RW' for read-write access respectively.

The "db" parameter defines the data block number (1-65535) of the variable.

The 'def' attribute defines the default value for a variable. A write-access variable is set with this value, until the first write operation to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported Variables for S7-300/400 CPUs

Type	index	access	comment
DBX	0.0-65535.7	R/W	Data block (Bit, requires parameter "db")
DBB	0-65535	R/W	Daten block (Byte, requires parameter "db")
DBW	0-65534	R/W	Data block (Word, requires parameter "db")
DBD	0-65532	R/W	Data block (DoubleWord, requires parameter "db")
DBS	0-65535	R/W	Data block (string, requires parameter "size" and "db")
I	0-16384.7	R	Inputs
Q	0-16384.7	R/W	Outputs
M	0-65535.7	R/W	Marker Bit
MB	0-65535	RW	Marker Byte
MW	0-65534	RW	Marker Word
MD	0-65532	RW	Marker DWord
T	0-2047	R/W	Timer
C	0-2047	R/W	Counter

Connecting a S7-300/400 Hardware
The S7-300/400 has to be connected to a MPI Alarm Modem via Profibus-Adapter.

```
Tixi Alarm Modem MPI              S7-300/400
(RProfibus-Adapter)             (Profibus-Adapter)
          B1  ----------------------------------B1
          A1   ----------------------------------A1
```

Remote Control
The remote access is fully transparent using TS-Adapter functions.

Following conditions are valid for TS-Adapter access protection:

1. The local access to the S7 (through modem COM1->COM2) can not be blocked. (Same as with genuine Siemens-TS-Adapter).

2. The TS users setup via SIMATIC-Software (Menue "TS-Adapter-Parameters") are working similar to the genuine Siemens-TS-Adapter, an ADMIN and 2 User may be defined, even a callback.

3. TS-Adapter settings are deleted on factory reset.

4. As soon as a S7 definition (External) AND (!) AccRights Users (see TiXML-Reference-Manual) are defined, only AccRights users with service "TSAdapter" are valid. With these a callback may be initiated. The users within TS-Adapter are kept but disabled. After deleting the AccRights they become enabled again.
   Additional advantage: The AccRights can use unlimited amount of users, a limitation to 3 users (as with genuine TS-Adapter) does not exist.

Example TS-Adapter AccRights:

```
[<SetConfig _="USER" ver="y">
   <AccRights>
      <Groups>
         <Housekeeper>
            <TSAdapter AccLevel="1"/>
         </Housekeeper>
      </Groups>
      <User>
         <Smith Plain="sunrise" Group="Housekeeper" Callback="123"/>
      </User>
   </AccRights>
</SetConfig>]
```

3.5 VIPA CPU 100/200/300

The variables of a connected VIPA controller have to be registered to the Tixi Alarm Modem variable set.

The VIPA CPUs can also be connected to an Alarm Modem with MPI interface, with full TS-Adapter support. (see chapter 3.4).

The variables are registered in the External section of the **'PROCCFG'** database.

Use COM port on extension card 1

```
<External>
   <Bus _=”C1” protocol=”Vipa,GreenCable” type=”Master”
   baud=”38400” TS=”1”>

      <Device _=”2” Pollrate=”1s”>
         <M30 _="M" ind="30" acc="R" />
      </Device>
   </Bus>
</External>
```

Station #2

The BUS-parameter contains the port address of the used extension card “C1”, the protocol manufacturer “Vipa”, the used protocol “GreenCable”, the operation mode “Master”, the baudrate 38400 and TS which is the station ID of the device itself.

Tixi Alarm Modem is always a master in communication with a VIPA.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For each controller a ‘Device’ entry has to be inserted, which contains the station-Id of the controller (‘_’ – attribute) and the poll rate. In the period set with the pollrate attribute the scanning of the VIPA is restarted (master mode) or a communication timeout is detected (slave mode).

For Aluline Alarm Modemx VIPA is only supported until firmware version 2.0.

A set of variables can be registered for the device:

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

Variable type in VIPA controller

Each line declares a logical name (alias, e.g. AlarmM10), the type of the variable (this can be either M, MB, MW, MD, I, Q, see VIPA controller for details about data types). The ‘ind’ attribute defines the index of the variable in the VIPA controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ – read, ‘W’ – write or ‘RW’ for read-write access respectively.

The ‘def’ attribute defines the default value for a variable.
A write-access variable is set to this value, until the first write operation to it is performed.
A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported Variables for VIPA System 200V CPU

Type	index	access	comment
M	0-2047.7	R/W	Marker (Bit)
MB	0-2047	R/W	Marker (Byte)
MW	0-2046	R/W	Marker (Word)
MD	0-2044	R/W	Marker (DoubleWord)
I	0-15.7	R	Input Register
Q	0-15.7	R/W	Output Register

Tixi Alarm Modem also supports VIPA System 100V CPU and VIPA System 300V CPU.

Remote Control
For remote access to the PLC following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

```
 [<TransMode baud="38400" format="8O1" com="MB"/>]
```

(use `com=”C1”` if connected to extension card RS232-2)

To connect to a VIPA CPU a 11-Bit modem (supporting 8O1 data format) has to be used on local side.

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

The used variables of a connected Moeller controller (e.g. easy 400/500/600/700/800/MFD) have to be registered to the Tixi Alarm Modem variable set.
The Moeller variables are registered in the External section of the **'PROCCFG**' database.

```
<External>                 Use COM port on extension card 2
   <Bus _=”C1” protocol=”Moeller,Easy 800” type=”Master”
   baud=”9600” handshake=”noDTR”>
   <Device _=”0” Pollrate=”1s”>
         <Input1 _="I" ind="1"/>                 Easy 800 Master
         <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" />          set of variables
         <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>
```

The BUS-parameter contains the port address of the used extension card, the protocol manufacturer “Moeller”, the kind of attached device “Easy 400/600”, “Easy 500/700” or “Easy 800”, the operation mode “Master”, the used baud rate and the necessary noDTR handshake.

For Easy 400/600 choose protocol “Moeller, Easy 400/600”, baudrate: 4800.
For Easy 500/700 choose protocol “Moeller, Easy 500/700”, baudrate: 4800.
For Easy 800/MFD choose protocol “Moeller, Easy 800””, baudrate 9600 or 19200.

Easy 400/600: The ‘Device’ entry has to be set to “1” and contains the poll rate.
Easy 500/700/800/MFD: The ‘Device” number of the PLC connected to Tixi Alarm Modem is always “0”. The 800/MFD supports multiple device stations (1-8).
If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.
For this device a set of variables can be registered:

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

Variable type in Moeller controller (IK,I,OU,...)

Each line declares a logical name (alias, e.g. Alarm11) and the type of the variable in the Moeller PLC (see list of supported variables).
The ‘ind’ attribute defines the index of the variable in the Moeller controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access depending on the type of variable.

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first writing to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported variables for Easy 400 / 600:

Type	index	access	comment
IK	1-16	R	extension input
I	1-8	R	digital input
OU	1-4	R	output clock
OC	1-8	R	output counter
OA	1-8	R	output analog compare
OT	1-8	R	output timer
IC	1-8	R	current value counter
D	1-8	RW	Text display
IT	1-8	R	current value timer
Q	1-8	R(W)	output (write only if STOP)
M	1-16	RW	Marker
U	-	RW	Clock
ST	1-8	R	Setpoint Timer
SC	1-8	R	Setpoint Counter
AW	1-8	R	Setpoint analog compare
QK	1-8	R(W)	extension output (write only if STOP)
Key	1-4	R	Button (Key1-Key4)
AI	7,8	R	analog input (index 7,8)

Supported variables for Easy 500 / 700:

Type	index	access	comment
I	1-16	R	Inputs
Q	1-8	R	Outputs
R	1-16	R	Iput EASY-LINK
S	1-8	R	Output EASY-LINK
OU	1-8	R	Output clock, same as OUW (Easy 400/600 compatibilty)
OUW	1-8	R	Output clock week
OUY	1-8	R	Output clock year
OC	1-16	R	Output timer
OA	1-16	R	Output analog compare
OT	1-16	R	Output timer
IC	1-16	RW	Current value counter
D	1-16	R	Text display
IT	1-16	RW	Current value timer
IW	1-4	RW	Current value hours run counter
SW	1-4	RW	Setpoint hours run counter
N	1-16	RW	Marker N
M	1-16	RW	Marker M
U	-	RW	Clock
ST	1-16	RW	Setpoint 1 Timer
ST2	1-16	RW	Setpoint 2 Timer
SC	1-16	RW	Setpoint Counter
AW	1-16	RW	Setpoint analog compare, same as AWI2 (Easy 400/600 compatibility)
AWI1	1-16	RW	Setpoint 1 analog compare
AWI2	1-16	RW	Setpoint 2 analog compare
AWF1	1-16	RW	Gain 1 analog compare
AWF2	1-16	RW	Gain 2 analog compare
AWOS	1-16	RW	Offset analog compare
AWHY	1-16	RW	Hysterese analog compare
Key	1-4	R	Buttons
AI	7,8.11,12	R	Analog input

Supported variables for Easy 800 / MFD:

Type	index	access	comment
I	1-16	R	digital input ports I1-I16
Q	1-8	R(W)	digitaloutput ports Q1-Q8 (write only if STOP)
R	1-16	R	digital input ports R1-R16
S	1-8	R(W)	digital output ports S1-S8 (write only if STOP)
M	1-96	RW	Marker
MB	1-96	RW	Byte Marker
MW	1-96	RW	Word Marker
MD	1-96	RW	Doubleword Marker
Key	1-4	R	image of key pad Key1-Key4
IA	1-4	R	analog inputs IA1-IA4
QA	-	RW	analog output
U	-	RW	clock

If the variable is unique, e.g. “U” ,the parameter ‚ind’ can be omitted.
If access type ‚R’ is possible only, the parameter ‚acc’ can be omitted.

If the PLC is in “RUN“ mode, the output ports are immediately overwritten by the PLC program. In that case connect coils with the output ports and sets the corresponding coil.

For each registered device automatically one special system variables is included:
Type, e.g.
`[<Get _=”/Process/Aux?/D?/Type”/>]`
This variable shows the connected Easy Type, e.g. “412-DC-Rx”.

Remote Control
For remote access to the Easy following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

Easy 400 / 500 / 600 / 700

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

(use `com=”C1”` if connected to extension card RS232-2)

Easy 800/MFD

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

(use `com=”C1”` if connected to extension card RS232-2)

To connect to a Easy via GSM at least Easy-Soft 5.01 is necessary (increaseable device timeout).

3.7 Moeller PS30 & PS4/40

The used variables of a connected Moeller PS30 or PS4/40 PLC have to be registered to the Tixi Alarm Modem variable set.
The PS4 variables are registered in the External section of the '**PROCCFG**' database.

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

Use COM port on extension card 2

SucomA Master

Set of variables

The BUS-parameter contains the port address of the used extension card, the protocol manufacturer “Moeller”, the kind of protocol “SucomA”, the operation mode “Master” and the used baud rate (supported 4800 up to 57600).
The Device ID has to be “7”.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.
For this device a set of variables can be registered:

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

Variable type in Moeller controller (S,F,B,R,D)

Each line declares a logical name (alias, e.g. Alarm11) and the type of the variable in the Moeller PLC (see list of supported variables).

The ‘ind’ attribute defines the index of the variable in the Moeller controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access depending on the type of variable.

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first writing to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported variables for PS30 & PS4/40:

Type	index	access	comment
S	0-65535	R	Status WORD
F	0-14999.7	R	Marker BIT
B	0-14999	RW	Marker BYTE
R	0-14998	RW	Marker WORD
D	0-14996	RW	Marker DWORD

If access type ‚R’ is possible only, the parameter ‚acc’ can be omitted.
If the PLC is in “RUN“ mode, the output ports are immediately overwritten by the PLC program. In that case connect coils with the output ports and sets the corresponding coil.

Remote Control
For remote access to the PS4 following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

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

(use com=”C1” if connected to extension card RS232-2)

3.8 SAIA Burgess S-Bus

The used S-Bus type and variables of a connected SAIA S-Bus controller (e.g. PCD2) have to be registered to the Tixi Alarm Modem variable set.
The S-Bus type and variables are registered in the External section of the **'PROCCFG**' database.

```
<External>
   <Bus _=”C0” protocol=”Saia,SBus-DataMode” type=”Master”
   baud=”19200” handshake=”HALF”>

      <Device _=”1” Pollrate=”1s”>
         <F100 _="F" ind="100" acc="R" />
      </Device>
      <Device _=”3” Pollrate=”60s”>
         <R102 _="R" ind="102" acc="R" />
      </Device>

   </Bus>
</External>
```

Use COM port on extension card 0

be S-Bus Master

enables RS485 mode

S-Bus station #1

S-Bus station #3

Tixi Alarm Modem can be configured as S-Bus master or as slave. All S-Bus baud rates are supported, the default baud rate (no entry) is 19200.
If Tixi Alarm Modem is operating as a S-Bus master on port “MB” (mainboard), the S-Bus communication starts as soon as the PC is disconnected and will be interrupted automatically, of a TiXML command is detected.

The parameter `handshake=”HALF”` enables the RS485 mode on special extansion cards.

For each controller on the S-Bus a ‘Device’ entry has to be inserted, which contains the station-Id of the controller on the S-Bus (‘_’ – attribute) and the poll rate. In the period set with the poll rate attribute the scanning of the S-Bus slaves is restarted (master mode) or a communication timeout is detected (slave mode).

For each device a set of variables can be registered:

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

Variable type in S-Bus controller (F,T,C,I,O,R)

Each line declares a logical name (alias, e.g. Alarm11), the type of the variable on the S-Bus (this can be either ‘F’ - Flag, ‘T’ – Timer, ‘C’ – Counter, ‘I’ – Input, ‘O’ – Output, ‘R’ – Register) see S-Bus controller for details about S-Bus data types). The ‘ind’ attribute defines the index of the variable in the S-Bus controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ – read, ‘W’ – write or ‘RW’ for read-write access respectively. An additional special access type is ‘RWL’ which means read-write access with linked buffers. With ‘RW’ access, Tixi Alarm Modem separates the variable buffer for read and write access. This should be used as “Slave” only.
In this case the PLC writes into Tixi Alarm Modems read buffer for this variable, and Tixi Alarm Modem writes into its write buffer of the variable. It’s therefore possible that the PLC writes a 1 into the variable read buffer, but when the PLC reads the variable, it will get the value of Tixi Alarm Modems write buffer, which still can be set to 0.
With ‘RWL’ Tixi Alarm Modem uses just one buffer for both accesses, which means that the last written value is valid, regardless who has written it.

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first writing to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported Variables for PCD2

Type	index	access	comment
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	Output
I	0-256	R	Input (Master only)

Connecting a PCD2 Hardware

Tixi Alarm Modem can be connected to a PCD2 on all 3 serial ports S0-S2. Only a 3-pin connection (RX,TX,GND) is necessary.

Please pay attention on these hints:

1. If you connect Tixi Alarm Modem to the PGU-Port (S0) of the PCD2, the DSR must not be connected, otherwise the PCD2 disables the S-Bus protocol.
2. If you connect Tixi Alarm Modem from mainboard RS232 (MB) the DTR must not be connected, otherwise the S-Bus protocol will be disabled.

```
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
```

Remote Control

For remote access to the PLC following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

Tixi Alarm Modem RS232 interface:

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

(use `com=”C1”` if connected to extension card RS232-2)

Tixi Alarm Modem RS485 interface:

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

Choose same baudrate as PLC.

3.9 Carel Macroplus

The used variables of a connected carel-controller (e.g. Magcroplus) have to be registered to the Tixi Alarm Modem variable set.

The Carel-bus variables are registered in the External section of the '**PROCCFG**' database.

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

Use COM port on extension card 0

Enables RS422-Mode

CarelBus controller #1

CarelBus controller #3

The parameter `handshake=”FULL”` enables the RS422 mode. Without that entry the communication will be made using the RS232 interface.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For each controller on the carel-bus a ‘Device’ entry has to be inserted, which contains the Id of the controller on the carel-bus (‘_’ – attribute) and the Poll rate. Tixi Alarm Modem queries a controller for changed variables as long as the controller doesn’t answer with a NULL frame, indicating that no more changed variables are to report. In the period set with the Poll rate attribute the scanning of the carel controller is restarted.

For each device a set of variables can be registered:

Variable type in carel controller (D,I,A)

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

Each line declares a logical name (alias, e.g. Alarm11), the type of the variable on the carel-bus (this can be either ‘D’ - boolean, ‘I’ – Integer, or ‘A’ – Analog , see carel-bus controller for details about carel data types). The ‘ind’ attribute defines the index of the variable in the carel-bus controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access respectively.

Supported variables for Macroplus

Type	index	access	comment
D	1-183	R/W	Boolean
I	1-50	R/W	Integer
A	1-50	R/W	Analog

Connecting a Macroplus Hardware
The PC2 may be connected via RS422-RS232 adapter or directly to a RS42 extension card.

```
Tixi Alarm Modem                       PC2
(RS422 extension card)               (9pin RS422 female)
            T+  -----------------------------------R+ (4)
            T-   -----------------------------------R-  (5)
            R-  ------------------------------------T-   (1)
            R+ ------------------------------------T+  (2)
```

Remote Control
For remote access to the PLC following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

Tixi Alarm Modem RS232 interface:

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

(use `com=”C1”` if connected to extension card RS232-2)

Tixi Alarm Modem RS422 interface:

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

3.10 ABB AC010, AC031, CL Series

The AC010 logic control is an OEM-version of Moeller EASY 400/600 products.
See chapter 3.6 for details.

The AC031 is supported via MODBUS RTU protocol.
See chapter 4.2 for details.

The CL Series logic control is an OEM-version of Moeller EASY 500/700 products.
See chapter 3.6 for details.

3.11 Allen Bradley Pico

This series of small intelligent relays is an OEM-version of Moeller EASY products.
Pico Series A = Moeller Easy 400/600
Pico Series B = Moeller Easy 500/700
Pico GFX = Moeller MFD
See chapter 3.6 for details.

3.12 Theben PHARAO II

The PHARAO II logic control is an OEM-version of Mitsubishi Alpha2 (XL).
See chapter 3.1 for details.

4 Fieldbus Support

4.1 Tixi-Bus

Tixi-Bus is a simple field bus protocol for exchanging variables in an efficient way, even with multiple PLCs connected to the Alarm Modem.
(Check Tixi.Com web site for details or send email to Developer@Tixi.Com)

The Tixi-Bus variables are registered in the External section of the **'PROCCFG**' database.

```
                                               Use COM port on extension card 0
   <External>
      <Bus _="C0" protocol="Tixi.Com,Tixi-Bus" type="Master"
      handshake="none">
         <Device _="1" Pollrate="1s">                      PLC #1
            <Alarm11 _="F" ind="22" acc="RW"/>
         </Device>
         <Device _="3" Pollrate="60s">                     PLC #3
            <Alarm31 _="R" ind="243" acc="RW"/>
            <Array _="B" ind="1" no="8" acc="RW"/>
         </Device>
      </Bus>
   </External>
```

The modem can be Tixi-Bus “Master“ only.

The parameter `handshake="mode"` enables different communication modes:
Using s RS232 interface no entry is necessary.
With handshake “HALF” the communication is done via RS 485 and with “FULL” the RS 422 is used. For RS 485/422 a special extension card is available.

TS is the station number of the Alarm Modem.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For each controller on the Tixi-Bus a ‘Device’ entry has to be inserted, which contains the Id of the controller on the Tixi-Bus (‘_’ – attribute) and the poll rate. Tixi Alarm Modem queries a controller for changed variables as long as the controller doesn’t answer with a NULL frame, indicating that no more changed variables are to report. In the period set with the poll rate attribute the scanning of the Tixi-Bus controller is restarted.

For each device a set of variables can be registered:

```
                                   Variable type in controller (F,W,DW,R...)
   <Alarm11 _="F" ind="22" acc="RW"/>
```

Each line declares a logical name (alias, e.g. Alarm11), the type of the variable on the Tixi-Bus (see “Supported variables list” below). The ‘ind’ attribute defines the index of the variable in the controller and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access respectively.

Tixi-Bus variables can be registered as an array of elements using the attribute ”no“. See chapter 2.4 for further information.

Supported variables for Tixi-Bus

Type	index	access	comment
C	0-65535	R/W	Flag (Coil)
W	0-65535	R/W	Word
DW	0-65535	R/W	DWord
F	0-65535	R/W	Float
DF	0-65535	R/W	Double
B	0-65535	R/W	Byte
S	0-65535	R/W	String (“size” has to be specified)

4.2 ASCII Protocol

The ASCII protocol is an easy way to let the Tixi Alarm Modem query values of an attached device using a simple text protocol.

The ASCII protocol is defined in the External section of the **'PROCCFG**' database.

Use COM port on extension card 0

Variables of the attached device.

```
<External>
   <Bus _=”C0” protocol=”Tixi.Com,ASCII” type=”Master”
   handshake=”none” baud=”115200” format=”8N1”>

      <Device _=”1” Pollrate=”1s”>
         <Float  _="DF"  Pos="14" End="24" acc="R" Radix="K"
         Request="&#13;" ResEnd="Ende&#13;" ResTime="10s"/>
         <Flag _="C"  Pos="0" acc="R" />
         <Word _="W"  Pos="3" End="5" acc="R" />
         <BinWord _="W"  Pos="7" Radix="B" acc="R" />
         <String _="S"  Size="30" Pos="3" Width="10" acc="R"/>
      </Device>

   </Bus>
</External>
```

The parameter `handshake=”mode”` enables different communication modes:
Using s RS232 interface no entry is necessary.
With handshake “HALF” the communication is done via RS 485 and with “FULL” the RS 422 is used. For RS 485/422 a special extension card is available.

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

For each device a ‘Device’ entry has to be inserted, which contains the id of the attached device (‘_’ – attribute) and the poll rate. In the period set with the poll rate attribute the queriing of the attached device is restarted.

For each device a set of variables can be registered:

Variable type in controller (F,W,DW,R...)

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

Each line declares a logical name (alias, e.g. Alarm11), the type of the variable at the attached device (see “Supported variables list” below) and the ‘acc’ attribute the access type ‘R” (read). All other parameters are explained in the next chapters.

The ‘def’ attribute defines the default value for a variable. The variable will contain this value until Tixi Alarm Modem gets the first value from the attached device.

4.2.1 Definition of variable queries

The format of the variable query may be defibed for each variable using these parameters:

Request
String that has to be sent to query data.

Wait
Time the Tixi Alarm Modem will wait before sending the request.

ResTime
Time the Tixi Alarm Modem will wait for the first character.

CharTime
Specifies the maximum idle time between two characters. If this time is exceeded, the end of the message will be assumed.
If **ResTime** is used without **CharTime**, ResTime will be the maximum time for receiving the message. After this time the message is assumed as completed, even if other characters are following.

ResEnd
String that specifies the end of the message. If ResEnd is not specified, the timeout (ResTime/CharTime) is valid only. Otherwise both has to be fulfiled.

4.2.2 Strings interpretation

Parameters are valid for all received packets until a new format is specified by another variable query.

Find
Searchs the receive buffer for he specified string.

Pos
The data field is found at the specified position.
The position will bet the beginning of the received string or, if specified, the end of the search-string (Find).

FindPos
Starts the string search at the specified position.

End
The end of the data filed is found at the specified position.
The position will bet the beginning of the received string or, if specified, the end of the search-string (Find).

Width
As alternative to ’End’ attribute the size of the data field may be defined.

4.2.3 Modifiing Values

Data types may be modified using parameter "**Radix**":

"D"	decimal
"H"	hexadecimal
"O"	octal
"B"	binary

Floating point formats:

"K" Left- and right-of-comma position separated by comma. Dots are ignored as thousand delimiter.
'E/e' used as exponent, '+/-' as sign. All other characters except numbers will stop the convertion.

"P" Left- and right-of-comma position separated by dot. Commas are ignored as thousand delimiter.
'E/e' used as exponent, '+/-' as sign. All other characters except numbers will stop the convertion.

The default Radix depends on the data type:

Data type	Radix
B,W,DW,I	D
F,DF	P
C,S	"" (empty Text)

Information about flags:
A flags is read as a single character. Characters "J/j/Y/y/1/0" will be TRUE (1). Characters "N/n/0" will be FALSE (0).

List of supported Variables ASCII Protocol

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

4.3 Modbus RTU

The used variables of a connected PLC with Modbus RTU support have to be registered to the Tixi Alarm Modem variable set.
The Modbus RTU variables are registered in the External section of the **'PROCCFG**' database.

```
<External>
   <Bus _=”C1” protocol=”Modbus,RTU” type=”Master” baud=”19200”>
   <Device _=”1” Pollrate=”1s”>
          <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 >
```

Use COM port on extension card 1

Modbus RTU Master

Set of Variables

The BUS-parameter contains the port address of the used extension card, the protocol “Modbus,RTU”, the operation mode “Master” and the used baudrate.

Supported baudrates: 1200, 4800, 9600, 14400, 19200, 38400

If Tixi Alarm Modem is operating on port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

Special parameters may be antered in the device definition, to adjust the Tixi Alarm Modem PLC communication (Modbus RTU only):

CharTimeout Timeout between characters (50ms)

Pause Idle time between messages (50ms)

Timeout Timeout for response (300ms)

DWordInc AdressIncrement between two successive DWORDSs (2)

DwordSwap Has to be enabled if low word is send before high word DWORD (0)

ForceSingleWordWrite

Enable, if Funct 0x06 has to be used instead of 0x10 for single WORD writing (0)

UseCache If set to 0, the combination of variables to a queried block (caching) will be deactivated. All variables will be queried separately. (Default: 1)

For this device a set of variables can be registered:

Variable type in Modbus RTU protocol (C,I,R,H,D)

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

Each line declares a logical name (alias, e.g. Alarm11) and the type of the Modbus RTU variable (see list of supported variables).
The ‘ind’ attribute defines the index of the Modbus RTU variable as HEX-code and the ‘acc’ attribute the access type. The access type can be either ‘R’ or ‘RW’ for read-only or read-write access depending on the type of variable. You may add a ’C’ to the acc attribute value (e.g. ’RWC’) to reactivate the cached access to the following variables block (if UseCache=0).

The ‘def’ attribute defines the default value for a variable. A write-access variable is set with this value, until the first writing to it. A read-access variable will contain this value until Tixi Alarm Modem gets the first value from the PLC.

Supported variables for Modbus RTU:

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

If access type ‚R’ is possible only, the parameter ‚acc’ can be omitted.

Modbus function codes
The Tixi Alarm Modem uses following Modbus function codes for variable access:

Code (decimal)	**Type of variable**
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 (if ForceSingleWordWrite=1)
15 – Force Multiple Coils	C (if size>1)
16 - Preset Multiple Registers	H, R, D, HI, RI, DI

Using variable attributes “read” and “write” the default function codes can be changed.
For writing registers FC16 will be used by default. FC6 can be enabled by setting the parameter “ForceSingleWordWrite”.

Remote Control
For remote access to a Modbus RTU device following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

```
[<TransMode baud="19200" format="8N1" com="MB"/>]
```
(use `com=”C1”` if connected to extension card RS232-2)

4.4 Modbus ASCII

The Modbus ASCII configuration is similar to Modbus RTU (chapter 4.3).
The only difference is the bus `protocol` parameter which has to be `“Modbus,ASCII”`.
Die speziellen Modbus RTU Device-Parameter (CharTimeout, Pause, Timeout, DWordInc, DwordSwap, ForceSingleWordWrite) sind bei Modbus ASCII nicht gültig.

4.5 M-Bus

M-Bus (Meter-Bus) is a field bus protocol for monitoring energy and climate devices in an efficient way, even with multiple devices connected to the Alarm Modem.

The M-Bus variables are registered in the External section of the **'PROCCFG'** database.

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

Use M-BUS interface on extension card 0

Device 1

Device 2

For best connection to M-Bus devices a special M-Bus extension card is available. The default interface address is “C0”. M-Bus devices may also be connected via external

M-Bus/RS232 converter. If in this case the Tixi Alarm Modem is operating on serial port “MB” (mainboard), the communication starts as soon as the PC is disconnected and will be interrupted automatically, if a TiXML command is detected.

The M-BUS implementation operates as Bus Master with baudrate 2400, 8 data bits, 1 stop bit, even parity and without handshake.
For each device on the M-Bus a ‘Device’ entry has to be inserted, which contains at least the primary (`PrimaryAddr,` decimal), secondary (`SecondaryAddr,` 8 decimals) or fabrication address (`FabricationAddr,` 8 decimals) of the device and the poll rate.

Optional parameters are “`ManufactoryCode`” (3 ASCII characters), “`Generation`” (hex) and “`Medium`” (hex) which are used for further differentiation if devices with same addresses are connected.

For each device a set of variables can be registered:

```
    <Var01 ind=”1”/>
```

Each line declares a logical name (alias, e.g. `Var01`) and the ‘`ind`’ attribute defines the position of the variable inside the M-BUS telegram from the device.
After registering the M-Bus variables to the Tixi Alarm Modem these variables can be accessed as variables in the Process-subtree.

Example:

```
    <L _=”&#xae;/Process/Aux0/D1/Var01”/>
    <L _=”&#xae;/Process/Aux0/D2/Var01”/>
```

Display of device addresses

The device „primary address“, „secondary address“ and the „manufacturer code“ can be displayed and used for data logging and other purposes.
Following entries have to be added to the variable list:

```
<PrimaryAddr _="primary"/>
<SecondaryAddr _="ident"/>
<Manufactor _="manufactor"/>
```

Special initializations

Time:
To transmit the Alarm Modem RTC time to the M-Bus device a special “Time” variable (not readable) has to be defined in the appropriate bus section:

```
<Time _=”DateTime”/>
```

Reset:
At the beginning of the communication a “reset code” may be sent to the device. The necessay variable “ResCode” is not readable and contains a reset code as parameter “def” (value=0-255):

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

Raw data initialisation:
At the beginning of the communication a user defined datagram (parameter `def`) is sent to the appropriate device:

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

This def string has to be entered in hex bytes. The first byte is the CI-field followed by the raw data without checksum.

VIF – Value Information Field: Medium/Einheit:
During reading M-BUS variables the Alarm Modem can show the information of VIF regarding Medium and Unit. Additional information can be found in chapter 5.

Data logging:
To reduce the risk of data loss, the binary logfile memory assigned for M-Bus variables is by default 37Byte per value (32Byte String + 5Byte Data type).
If integer variables are logged, the logfile record attribute „size“ may be used to reduce the memory for each entry. We recommend a size of 9Byte (4Byte value + 5Byte Data type), e.g.:

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

Further information on data logging and calculating logfile size can be found in the TiXML-Recerence-Manual.

Remote Control
For remote access to the M-BUS devices following “TransMode” command is necessary (see TiXML-Reference manual for more informations)

Tixi Alarm Modem M-BUS interface:

```
[<TransMode baud="24200" format="8E1" com="C1"/>]
```

5 Formatting PLC Variable Values

Without any formatting the value of the variables will be shown as retrieved by the PLC. Tixi Alarm Modem is able to reformat the value into a number format or to replace the status of a boolean variable with a string. The reformatted variable will be used for email generation and as result of get commands (without own format option).

Example:

```
<Temperature  _="I" ind="1" acc="R" format="R10F+4,2;°C"/>
```

Temperature is an integer value “1234”, but with the format string the output will look like this:

```
<Get _="/Process/Aux1/D1/Temperature"/>
<Get _="+12,34°C"/>
```

Formatting output of PLC variables

On variable definition:

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

On query of variable value:

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

On writing variable value:

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

Description:

The parameter `format` persists of two parts separated by s**emicolon**:

1.part:

Contains **Format-Elements** to describe the in- and output of values. Except thousand limiter „T“ and number format „F“ the format elements can not be combined. The position of the thousand delimiter within format instruction can be choosen at pleasure. The format depends on the type of variable. Not every format is available for all types of variables. The availability of a format element depends on the attribute „simpleType“ of the variablen definition. Therefore the valid basic types are given within this discription. The first part may be left empty to show the values native.

2.part:

Contains a **Text** to be displayed together with the value. The value may be displayed within this text using ist given format of part 1. The position of the value is defined by %%. For some variable additional values (e.g. physical medium and unit) may be included. Second part may be left empty too. In this case no semicolon is necessary.

Example:

Both parts:	“T’F+9,2 ;Radius %% cm“
Only 1. part:	“R16“
Only 2. part:	“; Text with:%% as value“

Format elements (Part 1):

? - logical alternative **`?string1,string2`**

This command is used to replace the boolean values of a variable by predefined strings. If the variable is not zero string1 is emitted, otherwise string2.

Available for following simpleType values:

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

***Example*:**

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

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

Tixi Alarm Modem answers:

```
<Get _="open"/>  on value 1
```

*** - case alternative** **`*Value1:Text1*Value2:Text2**:Text3`**

This command is used to replace a value of a variable by predefined strings. If the variable is equals Value1 Text1 is emitted, if the variable is equals Value2 Text2 is emitted, on every other value Text3 is emitted.

*****	**separator for values to detect**
******	**separator for all other values**

The number of values is not limited.

Available for following simpleType values:

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

***Example*:**

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

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

Tixi Alarm Modem answers:

```
<Get _="low"/>  on value 0
<Get _="medium"/>  on value 1
<Get _="high"/>  on value 2
<Get _="faulty"/>  on value 7
```

R/r - Basis **`Rn/rn`**

This command defines the basis n of the value.

n = 2	binary output (e.g. 01101010)
n = 8	octal output (e.g. 21057)
n = 10	decimal output (default, e.g. 1234)
n = 16	hexadecimal output (e.g. AE03)

With upper/lower case the display of hex letters (A-F) is specified:

R	Only upper case letters (e.g. AE03)
r	Only lower case letters (e.g. ae03)

Available for following simpleType values:

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

Example:

Value lower case:

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

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

Tixi Alarm Modem answers (variable value=90):

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

Value upper case:

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

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

Tixi Alarm Modem answers (variable value=90):

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

T – thousand delimiter **T*n***

Defines the thousand delimiter.

n= , comma as thousand delimiter (e.g. 12,345,678)
n= . Punkt als Tausendertrennzeichen (e.g. 12.345.678)
**n= `** apostrophe as thousand delimiter (e.g. 12`345`678)
n= empty no thousand delimiter (default)

Note:
Can be combined with format element „F“ or left alone.

Available for following simpleType values:
Uint8, Uint16, Uint32, Int8, Int16, Int32, Float, Double

Example:

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

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

Tixi Alarm Modem answers (variable value=98765):

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

F - number format

`F Sign padding field width decimal point fixed point numbers`

This command defines the format of the number.
It includes several subitems, which have to be in the given order:

sign: defines, if a sign should emitted

+	the sign is emitted, even if the value is positive (e.g. „+12.3“ , „-12.3“)
-	the sign is only emitted, if the value is negative (e.g. „12.3“ , „-12.3“)
empty	the value is unsigned

padding: defines how empty positions in the output field have to be filled (only used, if output field width is given)

0	the field is filled with zeros (e.g. 0066.3)
empty	unfilled fields are cut off (e.g. 66.3)

field width: gives the maximum size of the number output, **including** sign, thousand delimiter, decimal point and the value itself. If omitted, the field width is not limited (and no insertion of padding characters takes place).
Always define enough characters, otherwise the value will be cut off on the left side.

decimal point character: this character is used as decimal separator (option)

,	a comma is used as decimal separator
.	a dot is used as decimal separator (default)

fixed point number: this defines the number of digits behind the decimal separator. Can be omitted, if no decimal point separator is given.

Note:
Can be combined with format element „T“ or left alone.

Available for following simpleType values:
Uint8, Uint16, Uint32, Int8, Int16, Int32, Float, Double

<u>Examples</u>:

The value of the variable is “12345” for all examples:

sign:

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

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

Tixi Alarm Modem answers (Value = 123,45):

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

field width, padding:

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

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

Tixi Alarm Modem answers (Value = 123,456):

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

Fix point value , decimal point, fixed point numbers, padding :

```
<Variable _="R" simpleType="Int32" exp="-3"… format="T’F+9.2"/>
```

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

Tixi Alarm Modem answers (Value = 3123,456):

```
<Get _="+3’123.45"/>
```

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

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

Tixi Alarm Modem answers (Value = -3123456):

```
<Get _="-312’345"/>
```

Floating pouint valuel, decimal point, fixed point numbers, padding :

```
<Variable _="F" simpleType="Float"… format="T’F+9.2"/>
```

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

Tixi Alarm Modem answers (Value = 3123,456):

```
<Get _="+3’123.45"/>
```

<u>Text (part 2):</u>

%%

Defines the position of the value within output.
This part is available for all types of data. It is the only format option for data type „String“.

<u>Example</u>:

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

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

Tixi Alarm Modem answers (Value = 123,45):

```
<Get _=”Temp: +123.45°C”/>
```

%M% – M-BUS Medium (VIF)

%U% – M-BUS Unit (VIF)

These commands will add the M-BUS Value Information Field data to the value.

<u>Example:</u>

```
<Var01 simpleType="Int32" exp="-2"...
       format=";Medium:%M% value=%%   %U%"/>
<Get _="/Process/Aux0/D1/Var01"/>
```

Tixi Alarm Modem answers (Variable value=25,30, Heat counter volume):

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

6 Usage of PLC variables with Tixi Alarm Modem

The variables can be used in the same manner as, for example, the variables from the bit ports for process variables, event states, message text or binary/xml logging.

6.1 Addressing

After registering the PLC variables to the Tixi Alarm Modem these variables can be accessed as variables in the Process-subtree.

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

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

The variables are inserted in the path generated from the number of the extension card (e.g. AuxMB, Aux0 for C0, Aux1 for C1) and the device Id on the PLC (e.g. D0, D1,D2,…) and may be addressed there via their variable name.

6.2 Addressing with Bus name and Station name

With usage of Bus name and Stations name the variables may be addresses independent from the interface and station number. Therefore the PLC interface and/or station number may be changed in the PLC definition without need to change the whole project.
Example:

```
<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>
```

The Input1 may now be addressed like this:
`<L _=”&#xae;/Process/MyPLC/AlphaXL/Alarm11”/>`

This address would be still valid, even if the interface “COM2” and/or the DeviceID “0“ are changed inside the PLC definition.

6.3 Monitoring PLC communication

For each registered device a set of system variables is created automatically.

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

This variable shows the actual connection state of the PLC
online: DeviceState=1
offline: DeviceState=0

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

If the Alarm Modem detects changed variable values on the bus, the ChangeToggle bit changes its state.

Both variables may be used as an alarm or logging trigger within an EventState.

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

This write access variable is used to control the variable polling:

`[<Set _=“/Process/Aux?/D?/Active“` value=”0”`/>]` stops the communication.
`[<Set _=“/Process/Aux?/D?/Active“` value=”1”`/>]` starts the communication (default).

Index