

SCADA / GLT Anbindung für FP Gateways Handbuch



Version: 1.6.1

© 2018 -2021 FP InovoLabs GmbH

www.inovolabs.com

Redaktionsschluss: 23.02.2021

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.

Inhaltsverzeichnis

1	EINLEITUNG	3
1.1	Kurzbeschreibung FP Gateway	3
1.1.1	Schnittstellen und Feldbusse	3
1.1.2	Überblick Hauptfunktionen.....	4
1.2	Kurzbeschreibung GLT / SCADA	4
2	KOMMUNIKATIONSMÖGLICHKEITEN	5
2.1	Pull – Abholung und Schreiben von Daten	5
2.1.1	via TiXML.....	5
2.1.2	via HTTP	8
2.1.2.1	Lesen von Werten	9
2.1.2.2	Lesen der Konfiguration	10
2.1.2.3	Lesen von Logfiles	11
2.1.2.4	Schreiben einzelner Werte.....	12
2.1.2.5	Schreiben mehrerer Datenbankeinträge	12
2.1.2.6	Aufruf von Funktionen	12
2.2	Push – Empfang von Daten	13
2.2.1	via E-Mail.....	13
2.2.2	via HTTP	13
2.3	TiXML-Schnittstelle über http POST nutzen (TiXML.cgi)	14
3	ANHANG	16
3.1	Beispiel URLs	16
3.2	TDG Systemdaten	16
3.3	M-Bus Konfiguration	19

1 Einleitung

Dieses Handbuch beschreibt die Kommunikationsschnittstellen zwischen einem FP Gateway und einer Gebäudeleittechnik (GLT) bzw. einem SCADA System zur Datenhaltung, -aufbereitung und -visualisierung.

Auf folgende Daten kann die GLT lesend zugreifen:

- aktuelle SPS-, Zähler- oder I/O-Datenpunkte,
- historische SPS-, Zähler- oder I/O-Datenpunkte aus dem TDG Logspeicher,
- aktuelle Systemdaten des TDGs,
- aktuelle Konfigurationsdaten des TDGs.

Auf folgende Daten kann die GLT schreibend zugreifen:

- aktuelle SPS-, Zähler- oder I/O-Datenpunkte,
- einzelne Konfigurationseinträge,
- komplette Konfigurationen (via TiXML).

1.1 Kurzbeschreibung FP Gateway

Die direkte Anschlussmöglichkeit der FP Gateways an fast jede kleine oder große Steuerung sowie sehr viel Zählertypen läutet eine neue Ära einfacher und kostengünstiger Fernwartungslösungen für die Industrie ein. Der ca. 80 MB große Datenspeicher ist stromausfallsicher (Flash-Technologie).

Das FP Gateway wird über TiXML (optimiertes XML) konfiguriert und nutzt TiXML auch zur Datenhaltung und zum Datenaustausch.

1.1.1 Schnittstellen und Feldbusse

An FP-Gateways können bis zu 4 Feldbusse angeschlossen werden, wobei jeder Feldbus ein anderes Protokoll nutzen kann, zum Beispiel:


Bus1	auf COM1	RS232	Moeller EASY Klein-SPS
Bus2	auf COM2	RS485	MODBUS mit mehreren SPSen, z.B. ABB, Mitsubishi
Bus3	auf COM3	M-Bus	10 verschiedene Zähler für Strom, Gas, Wasser, Wärme
Bus4	auf COM4	RS485	Funk-Temperatur-Sensorsystem

An jedem Bus können beliebig viele Stationen oder Geräte angeschlossen sein – je nachdem, was das Bus-Protokoll zulässt.

Jedes Gerät am Bus kann Parameter oder Variablen besitzen.

Die unterschiedlichen Bussysteme, Geräte und Variablen werden vom FP-Gateway in einem hierarchischen Datenmodell aufgebaut mit den Hierarchiestufen

`/Process/BusX/Device_X/Variable_X`

Beispiel 

```

Process
  Bus1
    Device_1
      Variable_1
      Variable_2
      .....
    Device_2
      .....
  Bus2
    Device_1
    .....
  
```

Zugriffe auf den ganzen Bus, ein Gerät oder eine Variable erfolgen immer über diese Pfadangabe.

1.1.2 Überblick Hauptfunktionen

- FP Gateways sind als LAN- und Mobilfunk-Varianten verfügbar
- SPS-Protokolle führender SPS-Hersteller sind bereits integriert – Adaption ohne Umprogrammierung der SPS
- Alarmierung mit Quittierung und Alarmkaskaden via E-Mail
- Fernwirken per E-Mail und integriertem Web Server und über die Cloud
- Kalender z.B. für Nutzung von Schichtplänen bei Alarmierung
- Daten-Logging inklusive automatisierbaren Versand von Daten per E-Mail
- Daten-Sequencer für zeitabhängiges Regeln und Steuern indem z.B. eine Wertekurve für 1 Jahr mit 15 Minuten-Werten vorgegeben wird und Tixi aktuelle Werte mit dieser Kurve vergleicht
- Event-basiertes System
- Embedded Webserver zur Datenpräsentation und zum Steuern einzelner Parameter

1.2 Kurzbeschreibung GLT / SCADA

Eine GLT dient der Anlagen- und Datenverwaltung sowie Visualisierung der über das FP Gateway nutzbaren Daten von SPSen, Steuerungssystemen, Energiezählern und den intelligentem FP Gateways.

Neben dem Lesen und Schreiben einzelner Felddaten, sind Logfiles und Daten zur Pflege eine Maschinenakte möglich. Auch die Konfigurationsdaten aller FP Gateways sollten in der GLT verwaltet werden.

Damit eine GLT Zugriff auf diese Daten hat, müssen die Zugriffsmöglichkeiten zu den Endgeräten definiert sein und umgesetzt werden. Die dafür nutzbaren Schnittstellen werden im Folgenden erläutert.

Die GLT bzw. das SCADA-System werden im Folgenden als „Server“ bezeichnet.

2 Kommunikationsmöglichkeiten

Die Kommunikation zwischen dem FP Gateway und dem Server kann von beiden Seiten aus initiiert werden. Wenn der Server Daten anfordert, spricht man von „Pull“, wenn das Gateway von sich aus Daten sendet, spricht man von „Push“.

2.1 Pull – Abholung und Schreiben von Daten

2.1.1 via TiXML

Die TiXML-Kommunikation wird standardmäßig auf TCP Port 8300 ausgeführt.

Der Server sendet einen TiXML-Frame mit einem TiXML-Befehl, worauf das FP Gateway in einem TiXML-Frame mit den angeforderten Daten antwortet. Ein TiXML-Frame wird von eckigen Klammern umschlossen.

Auf diese Art können **Datenpunkte** einzeln oder in Gruppen gelesen werden, die **Konfiguration** kann gelesen und verändert werden sowie **Logdaten** in verschiedenen Formaten zeit- oder ID-basiert ausgelesen werden.

Beispiele

Abfrage eines Zählerstandes / SPS Datenpunktes:

```
[<Get _="/Process/Bus1/Device_0/Variable_0" ver="v"/>]
```

Antwort:

```
[<Get _="200"/>]
```

Abfrage eines Zählerstandes / SPS Datenpunktes mit Ausgabe des Tagname (ab FW 3.0.6.84):

```
[<Get _="/Process/Bus1/Device_0/Variable_0" ViewProperties="" ver="v"/>]
```

Antwort:

```
[<Get>  
<Variable_0 _="200"/>  
</Get>]
```

Abfrage aller Datenpunkte eines Zählers / einer SPS:

```
[<Get _="/Process/Bus1/Device_0/" ver="v"/>]
```

Antwort:

```
[<Get>  
<Device_0>  
<DeviceState _="1"/>  
<ChangeToggle _="0"/>  
<Variable_0 _="95"/>  
<Variable_1 _="0"/>  
<Variable_2 _="1"/>  
<Type _="412-DC-Rx"/>  
</Device_0>  
</Get>]
```

Abfrage aller Datenpunkte aller Zähler / SPSen:

```
[<Get _="/Process/Bus1/" ver="v"/>]
```

Die Bezeichnungen der Zweige "BusX", "Device_X" sowie der Variablen "Variable_X" können in TICO unter Berücksichtigung der TiXML-Regeln frei gewählt werden, wodurch eine Zuordnung der Daten über eindeutige Zweige/Namen vereinfacht wird.

Bei TILA-Projekten werden die Bezeichner hingegen intern vergeben, was eine Zuordnung erschwert. Ab Firmware 3.0.6.84 und TILA 2.5.0.330 können daher die in TILA vergebenen Variablennamen auf Variablen-, Device- und Busebene mit ausgegeben werden:

Abfrage eines Zählerstandes / SPS Datenpunktes mit Aliasname:

```
[<Get _="/Process/Bus1/Device_0/Variable_0" ViewProperties="Name"
ver="v"/>]
```

Antwort:

```
[<Get>
<Variable_0 _="3.055" Name="Energie"/>
</Get>]
```

Abfrage aller Datenpunkte eines Zählers / einer SPS:

```
[<Get _="/Process/Bus1/Device_0/" ViewProperties="Name" ver="v"/>]
```

Antwort:

```
[<Get>
<Device_0>
<DeviceState _="1"/>
<ChangeToggle _="0"/>
<Variable_0 _="95" Name="Energie"/>
<Variable_1 _="0" Name="Ausgang 1"/>
<Variable_2 _="1" Name="Eingang 3"/>
<Type _="412-DC-Rx"/>
</Device_0>
</Get>]
```

Beim Abruf der Datenpunkte handelt es sich nicht um aktuell vom Zähler gelesene, sondern um zwischengespeicherte Werte vom letzten Pollzyklus. Der Zeitpunkt des letzten Pollings kann daher zur Orientierung optional mit ausgegeben werden (ab FW 3.0.6.84 und TILA 2.5.0.330):

Abfrage eines Zählerstandes / SPS Datenpunktes mit Aliasname und Lesezeitpunkt:

```
[<Get _="/Process/Bus1/Device_0/Variable_0" ViewProperties="Name,TimeStamp" ver="v"/>]
```

Antwort:

```
[<Get>
<Variable_0 _="3.055" Name="Energie" TimeStamp="2009/01/19,11:06:48"/>
</Get>]
```

Abfrage aller Prozessdaten inkl. I/Os und allen angeschlossenen Bussystemen:

```
[<Get _="/Process/ " ver="v"/>]
```

Auslesen der vollständigen Systemdaten:

```
[<Get ver="v"/>]
```

Ändern eines Datenpunktwertes:

```
[<Set _="/Process/Bus1/Device_0/Variable_0" value="23" ver="v"/>]
```

Auslesen historischer Werte (vergangene 7 Tage) als XML-Struktur:

```
[<ReadLog _="Datalogging_0" range="previous 7 days" ver="v"/>]
```

Antwort:

```
[<ReadLog>
<ID_171361 _="2008/12/03,16:26:00">
<Variable_0 _="2"/>
<Variable_2 _="1"/>
<Variable_3 _="1"/>
<Variable_4 _="1"/>
<Variable_5 _="1"/>
</ID_171361>
<ID_171362 _="2008/12/03,16:27:00">
<Variable_0 _="32"/>
<Variable_2 _="1"/>
<Variable_3 _="1"/>
<Variable_4 _="1"/>
<Variable_5 _="1"/>
</ID_171362>
...
<ReadLog> ]
```

Ab FW 3.0.6.84 und TILA 2.5.0.330 können die Aliasnamen optional mit ausgegeben werden:

Auslesen historischer Werte (vergangene 7 Tage) als XML-Struktur mit Aliasnamen:

```
[<ReadLog _="Datalogging_0" range="previous 7 days" flags="UseAlias" ver="v"/>]
```

Antwort:

```
[<ReadLog>
<ID_171361 _="2008/12/03,16:26:00">
<Variable_0 _="2" Name="Energie"/>
<Variable_2 _="1" Name="Ausgang 1"/>
<Variable_3 _="1" Name="Ausgang 2"/>
<Variable_4 _="1" Name="Eingang 1"/>
<Variable_5 _="1" Name="Eingang 2"/>
</ID_171361>
...
<ReadLog> ]
```

Auslesen historischer Werte (letzte Stunde) als CSV-Daten:

```
[<ReadLog _="Datalogging_0" range="last 1 hours" type="CSV" ver="v"/>]
```

Antwort:

```
[<ReadLog>
<LogData>
ID;Date;Time;Variable_0;Variable_2;Variable_3;Variable_4;Variable_5
171369;2008/12/03;16:34:00;46;1;1;1;1
171370;2008/12/03;16:35:00;76;1;1;1;1
171371;2008/12/03;16:36:00;7;1;1;1;1
</LogData>
</ReadLog> ]
```

ID, Datum, Uhrzeit und die Kopfzeile können bei Bedarf über das Attribut "flags" ausgeblendet werden.

Bei der CSV-Ausgabe werden die Variablenbezeichner der Kopfzeile ab FW 3.0.6.84 und TILA 2.5.0.330 durch die in TILA vergebenen Aliasnamen ersetzt.

Auslesen historischer Werte (letzte Stunde) als CSV-Daten mit Aliasnamen:

```
[<ReadLog _="Datalogging_0" range="last 1 hours" type="CSV"
flags="NoID,NoDate" ver="v"/>]
```

Antwort:

```
[<ReadLog>
<LogData>
Date;Time;Energie;Ausgang 1;Ausgang 2;Eingang 1;Eingang 2
16:34:00;46;1;1;1;1
16:35:00;76;1;1;1;1
16:36:00;7;1;1;1;1
</LogData>
</ReadLog>]
```

Abfragen der Konfiguration:

```
[<GetConfig _="DATENBANK/Gruppe" ver="v"/>]
```

Ändern der Konfiguration:

```
[<SetConfig _="DATENBANK" ver="v">
<Gruppe>
...Daten...
</Gruppe>
</SetConfig>]
```

Antwort:

```
[<SetConfig/>]
```

Fehlermeldungen können über das Attribut „ver“ gesteuert werden (numerisch oder Klartext).

Der TiXML-Zugriff kann durch Benutzerdaten geschützt werden.

Weitere umfassende Informationen finden sich im

- **TiXML Reference Manual**
(ca. 200 Seiten, A5, Ringbindung – oder digital als Adobe Acrobat –PDF Dokument)
- **TiXML Tutorial**
(ca. 40 Seiten, A5, Heftbindung – oder digital als Adobe Acrobat PDF Dokument)

2.1.2 via HTTP

Die Kommunikation basiert auf HTTP v1.0 / 1.1. Der Kommunikationsport ist standardmäßig TCP 80.

Es existieren fünf Zugriffsmöglichkeiten:

- Lesen von Werten
- Lesen der Konfiguration
- Lesen von Logfiles
- Schreiben einzelner Werte
- Schreiben mehrerer Datenbankeinträge
- Aufruf von Funktionen

Fehlermeldungen werden als XML-Webinhalte ausgegeben.

Der HTTP-Zugriff kann durch Benutzerdaten geschützt werden.

Weitere umfassende Informationen finden sich im

- **Handbuch „Embedded Webserver für FP Gateways“**
(ca. 20 Seiten, A5, Heftbindung – oder digital als Adobe Acrobat PDF Dokument).

2.1.2.1 Lesen von Werten

Auf die Systemdaten kann man wie folgt zugreifen:

`http://<TDG-IP>/System/Properties/<PropertyPath>`

- <TDG-IP> ist die IP-Adresse des TDG;
- <PropertyPath> ist der Pfad einer Systemeigenschaft



Hinweis:

Zwischen Groß- und Kleinschreibung wird bei Pfadangaben unterschieden.

Die Antwort wird als XML-Webinhalt mit der XML-Kennung `<?xml version="1.0"?>` ausgegeben.



Beispiele

Abfrage eines Zählerstandes / SPS Datenpunktes:

`http://<TDG-IP>/System/Properties/Process/Bus1/Device_0/Variable_0`

Antwort:

```
<?xml version="1.0"?>
<Variable_0 _="200"/>
```

Abfrage aller Datenpunkte eines Zählers / einer SPS:

`http://<TDG-IP>/System/Properties/Process/Bus1/Device_0/`

Antwort:

```
<?xml version="1.0"?>
<Device_0>
<DeviceState _="1"/>
<ChangeToggle _="0"/>
<Variable_0 _="95"/>
<Variable_1 _="0"/>
<Variable_2 _="1"/>
<Type _="412-DC-Rx"/>
</Device_0>
```

Abfrage aller Datenpunkte aller Zähler / SPSen an einem Bus:

`http://<TDG-IP>/System/Properties/Process/Bus1/`

Die Bezeichnungen der Zweige "BusX", "Device_X" sowie der Variablen "Variable_X" können in TICO unter Berücksichtigung der TiXML-Regeln frei gewählt werden, wodurch eine Zuordnung der Daten über eindeutige Zweige/Namen vereinfacht wird.

Bei TILA-Projekten werden die Bezeichner hingegen intern vergeben, was eine Zuordnung erschwert. Ab Firmware 3.0.6.84 und TILA 2.5.0.330 können daher die in TILA vergebenen Variablennamen auf Variablen-, Device- und Busebene mit ausgegeben werden:

Abfrage eines Zählerstandes / SPS Datenpunktes mit Aliasname:

`http://<TDG-IP>/System/Properties/Process/Bus1/Device_0/Variable_0?ViewProperties=Name`

Antwort:

```
<?xml version="1.0"?>
<Variable_0 _="200" Name="Energie"/>
```

Abfrage aller Datenpunkte eines Zählers / einer SPS mit Aliasnamen:

```
http://<TDG-IP>/System/Properties/Process/Bus1/Device_0/?ViewProperties=Name
```

Antwort:

```
<?xml version="1.0"?>
<Device_0>
<DeviceState _="1" />
<ChangeToggle _="0" />
<Variable_0 _="95" Name="Energie" />
<Variable_1 _="0" Name="Ausgang 1" />
<Variable_2 _="1" Name="Eingang 1" />
<Type _="412-DC-Rx" />
</Device_0>
```

Beim Abruf der Datenpunkte handelt es sich nicht um aktuell vom Zähler gelesene, sondern um zwischengespeicherte Werte vom letzten Pollzyklus. Der Zeitpunkt des letzten Pollings kann daher zur Orientierung optional mit ausgegeben werden (ab FW 3.0.6.84 und TILA 2.5.0.330):

Abfrage eines Zählerstandes / SPS Datenpunktes mit Aliasname und Lesezeitpunkt:

```
http://<TDG-IP>/System/Properties/Process/Bus1/Device_0/Variable_0?ViewProperties=Name,TimeStamp
```

Antwort:

```
<?xml version="1.0"?>
<Variable_0 _="200" Name="Energie" TimeStamp="2009/01/19,11:06:48" />
```

Abfrage aller Prozessdaten inkl. I/Os und allen angeschlossenen Bussystemen:

```
http://<TDG-IP>/System/Properties/Process/
```

Auslesen der vollständigen Systemdaten:

```
http://<TDG-IP>/System/Properties/
```

Antwort:

siehe Kapitel 3.2

Möchte man einzelne Datenpunkte ohne XML-Struktur abfragen, so ist dies über ein CGI möglich, z.B.:

```
http://<TDG-IP>/cgi-bin/readVal.exe?/Process/Bus1/Device_0/Variable_0
```

Der Server gibt den Rohwert als MIME-Typ text/plain aus, z.B. „200“.

2.1.2.2 Lesen der Konfiguration

In Analogie zum Lesen von System-Daten kann auf die Konfigurations-Daten für die Pflege der Maschinenakte zugegriffen werden:

```
http://<TDG-IP>/System/Config/<PropertyPath>
```

- <TDG-IP> ist die IP-Adresse des TDG;
- <PropertyPath> ist der Pfad einer Konfigurationseigenschaft.



Beispiel

```
http://<TDG-IP>/System/Config/PROCCFG/External
```

Es wird ein XML-Webinhalt mit allen Konfigurations-Daten der SPS- / Zähleranbindung ausgegeben.

2.1.2.3 Lesen von Logfiles

Durch das TDG gesammelte historische Daten können über ein CGI abgerufen werden, wobei entweder auf vordefinierten Formatvorlagen (XML,CSV) zurückgegriffen, oder die Struktur in gewissem Rahmen selbst gestaltet werden kann.

XML-Daten werden direkt über den Logfilenamen abgerufen:

```
http://<TDG-IP>/System/Config/PROCCFG/External/
```

CSV und benutzerdefinierte Formate werden über eine Aliasformatierung abgerufen:

```
http://<TDG-IP>/cgi-bin/readLog.exe?<Logfile-Aliasname>+<Range>
```

Hierbei können optional ab Firmware 3.0.6.84 und TILA 2.5.0.330 die Aliasnamen der Variablen ausgegeben werden.

Die Angabe von <Range> ist zwingend und wie im **TiXML Reference Manual** beschrieben anzugeben.



Beispiele

Auslesen des vollständigen Logfiles „Datalogging_0“:

```
http://<TDG-IP>/cgi-bin/readLog.exe?Datalogging_0+all
```

Antwort:

```
<?xml version="1.0"?>
<Datalogging_0>
<ID_171361 _="2008/12/03,16:26:00">
<Variable_0 _="2"/>
<Variable_2 _="1"/>
<Variable_3 _="1"/>
<Variable_4 _="1"/>
<Variable_5 _="1"/>
</ID_171361>
<ID_171362 _="2008/12/03,16:27:00">
<Variable_0 _="32"/>
<Variable_2 _="1"/>
<Variable_3 _="1"/>
<Variable_4 _="1"/>
<Variable_5 _="1"/>
</ID_171362>
<ID_171363 _="2008/12/03,16:28:00">
<Variable_0 _="63"/>
<Variable_2 _="1"/>
<Variable_3 _="1"/>
<Variable_4 _="1"/>
<Variable_5 _="1"/>
</ID_171363>
</Datalogging_0>
```

Auslesen historischer Werte (letzte Stunde) als CSV-Daten:

```
http://<TDG-IP>/cgi-bin/readLog.exe?Datalogging_0_CSV+last%20%20hours
(letzte 1 Stunde, %20 = Leerzeichen)
```

Der Server gibt die Daten als MIME-Typ text/plain aus:

```
ID;Date;Time;Variable_0;Variable_2;Variable_3;Variable_4;Variable_5
171369;2008/12/03;16:34:00;46;1;1;1;1
171370;2008/12/03;16:35:00;76;1;1;1;1
171371;2008/12/03;16:36:00;7;1;1;1;1
```

Bei der CSV-Ausgabe werden die Variablenbezeichner der Kopfzeile ab FW 3.0.6.84 und TILA 2.5.0.330 durch die in TILA vergebenen Aliasnamen ersetzt.

2.1.2.4 Schreiben einzelner Werte

Datenpunkte können einzeln über ein CGI geschrieben werden:

`http://<TDG-IP>/cgi-bin/writeVal.exe?<Path>+<value>`



Beispiel

`http://<TDG-IP>/cgi-bin/writeVal.exe?/Process/Bus1/Device_0/Variable_0+23`

2.1.2.5 Schreiben mehrerer Datenbankeinträge

Möchte man mehrere Datenpunkte gleichzeitig schreiben, so kann der Schreibzugriff über ein http POST erfolgen, wie es z.B. ein Formular erzeugt:

```
<FORM method="POST" target="NullDevice">
  <INPUT type="hidden" name="section" value="Form1">
  <INPUT type="text" name="Value1" value="" />
  <INPUT type="text" name="Value2" value="" />
  <INPUT type="text" name="Value3" value="" />
  <INPUT class="button" type="submit" value="submit" />
</FORM>
```

Das hidden-Field "section" erzeugt in der PARAM Datenbank eine Sektion "value" (im Beispiel "Form1").

Dieses Feld muss als erstes im Formular bzw. POST stehen!

Die Formularfelder werden als einzelne Datenelemente in die Sektion übernommen.

Das Ergebnis des Beispiels käme folgendem TiXML-Befehl gleich:

```
[ <SetConfig _="PARAM" ver="v">
<WebForms>
<Form1>
  <Value1 _="xzy" />
  <Value2 _="xzy" />
  <Value3 _="xzy" />
</Form1>
</WebForms>
</SetConfig> ]
```

2.1.2.6 Aufruf von Funktionen

Das FP Gateway bietet ein CGI zum Aufruf von Funktionen (Event Handlern). Dadurch kann beispielsweise der Datenversand per E-Mail angestoßen werden, oder bis zu 10 Datenpunkte per http GET geschrieben werden.

`http://<TDG-IP>/cgi-bin/DoOn.exe?Eventhandler+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10`

Rückmeldungen:

2 --> wartet auf Ausführung

1 --> Ausführung läuft

0 --> Ausführung OK

Fehlernummer --> siehe **TiXML Reference Manual**

Die Rückgabe erfolgt bei OK/Fehler sofort oder nach 4s, wenn die Abarbeitung nicht sofort startet (--> 2).

2.2 Push – Empfang von Daten

2.2.1 via E-Mail

Aktuelle sowie historische Daten können vom FP Gateway in Form von E-Mails versendet werden. Der Nachrichtentext kann in jedem Fall frei gestaltet werden und aktuelle Variablenwerte über Referenzen enthalten.

Historische Daten werden in Form von E-Mail-Anhängen verschickt, wobei entweder auf vordefinierten Formatvorlagen (XML, CSV) zurückgegriffen, oder die Struktur in gewissem Rahmen selbst gestaltet werden kann.

 **Beispiel** (CSV-Attachments parametrisiert über TILA)

```
Datum;Zeit;Energie;Volumen;Vorlauftemp;Rücklauftemp;Differenztemp;Durchfluss
27.09.2008;11:00:00;0;0.0191;0.099;0.1;0.1;0
27.09.2008;12:00:00;0;0.0191;0.099;0.1;0.1;0
27.09.2008;13:00:00;0;0.019;0.099;0.1;0.1;0
27.09.2008;14:00:00;0;0.0189;0.099;0.1;0.1;0
```

2.2.2 via HTTP

Das FP Gateway ist in der Lage aktuelle Werte als http-Request mit der Methode „GET“ zu senden. Diese können dort zentralisiert verwaltet und weitergeleitet werden.

Dieses Kapitel beschreibt Anhand eines Beispiels die Konfiguration einer solchen „HTTP Notification“ über die Software TICO. Die Kenntnis der internen Prozessabläufe des FP Gateways wird vorausgesetzt.

Anlegen eines Event Handlers für die Notification:

```
[<SetConfig _="EVENTS" ver="y">
<EventHandler>
  <HTTP>
    <SendMail _="MessageJobTemplates/HTTP" />
  </HTTP>
</EventHandler>
</SetConfig>]
```

Anlegen einer Nachrichten Templates:

```
[<SetConfig _="TEMPLATE" ver="y">
  <MessageJobTemplates>
    <HTTP _="URLSend">
      <Recipient _="/D/AddressBook/Contact_0" />
    </HTTP>
  </MessageJobTemplates>
</SetConfig>]
```

Anlegen der URL im Adressbuch

Die URL kann eine oder mehrere Variablen enthalten, welche durch Referenzen (®/Pfad) gebildet werden.

Optional können der http-Port des Servers sowie eine Authentifizierung angegeben werden.

```
[<SetConfig _="TEMPLATE" ver="y">
  <AddressBook>
    <Contact_0>
      <URL _="http://glt.tixi.com/cgi-bin/writeData.cgi
        ?Variable_0=&#xae;/Process/Bus1/Device_0/Variable_0" />
    </Contact_0>
  </AddressBook>
</SetConfig>]
```

```

    <URLPort _="80" />
    <User _="DH" />
    <Password _="pwd" />
  </Contact_0>
</AddressBook>
</SetConfig>]

```

Dieses Beispiel ruft folgende HTTP Notification URL auf (wenn Variable_0=20):
 http://glt.tixi.com/cgi-bin/writeData.cgi?Variable_0=20

Beispiel (HTTP-Mitschnitt)
 GET /cgi-bin/writeData.cgi?Variable_0=20 HTTP/1.1
 Authorization: Basic VG14aW5ldGdlYXI6VG14aU1haWxCb3g=

2.3 TiXML-Schnittstelle über http POST nutzen (TiXML.cgi)

Die TiXML-Schnittstelle, die über die Software TICO und TILA zur Konfiguration der FP Gateways verwendet wird, ist auch über http POST verwendbar. Dazu ist in den FP Gateways ein cgi-Kommando `TiXML.cgi` implementiert. Über dieses Interface können alle TiXML-Befehle an das FP Gateway geschickt werden, zum Beispiel:

- Neue / geänderte Konfiguration hochladen
- Konfiguration abfragen
- Logdateien lesen
- Events auslösen

Die folgenden http POSTs werden hier am Beispiel des Posters von Firefox gezeigt.


Beispiel 1 (Abfrage der Gerätedaten über das TiXML-Kommando [`<Get />`])

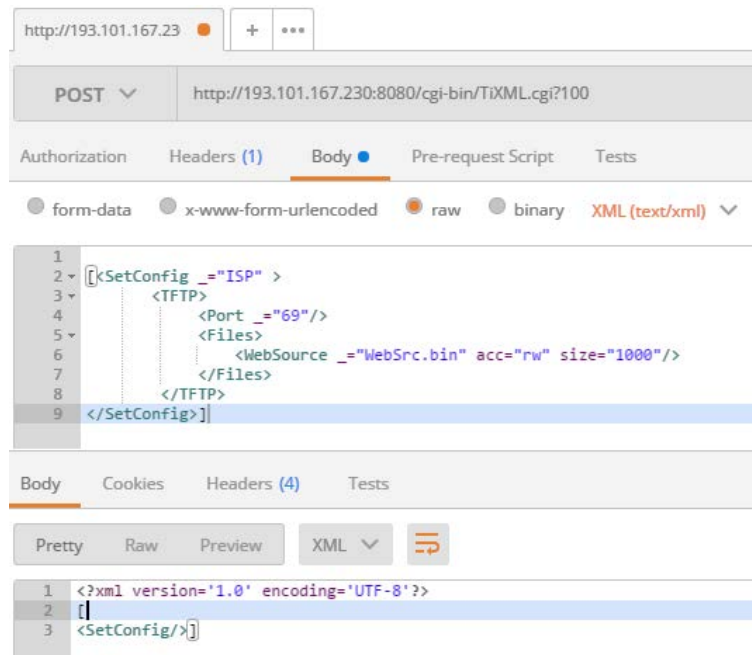
The screenshot shows a Firefox HTTP POST request to `http://193.101.167.68:8080/cgi-bin/TiXML.cgi?100`. The request body is `[<Get />]`. The response body is an XML document with the following structure:

```

1 <?xml version='1.0' encoding='UTF-8'?>
2 [
3 <Get>
4   <SYSTEM>
5     <Hardware>
6       <Modules>
7         <RTC _="RTC8564"/>
8         <Modem0 _="HE623-M25"/>
9         <FlashOnboard _="128MB"/>
10        <COM1 _="RS232"/>
11        <COM2 _="RS232"/>
12        <COM3 _="MBus"/>
13        <ETH1 _="ETH1"/>
14        <MainBoardDigital _="GP21D-I/0"/>
15      </Modules>
16      <RAM>
17        <Size _="128299008"/>
18        <Attributes _="0"/>
19      </RAM>
20      <ROM>
21        <Size _="134217728"/>
22        <Attributes _="0"/>
23      </ROM>


```

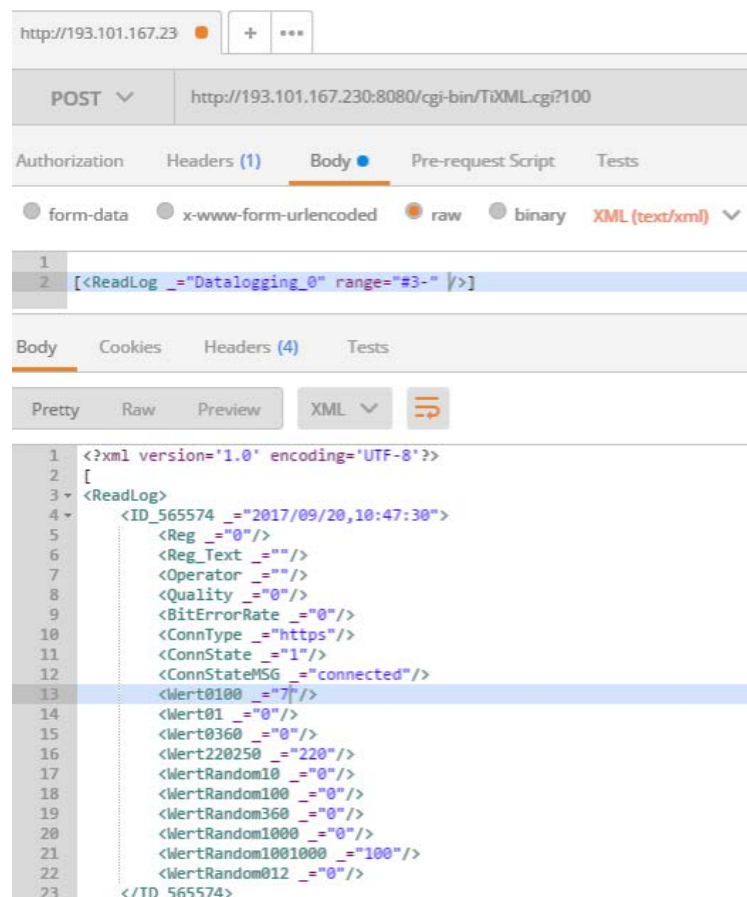
Beispiel 2 
 (Neue TFTP-Datenbank
 in das Gerät einspielen)



```

http://193.101.167.230
POST http://193.101.167.230:8080/cgi-bin/TiXML.cgi?100
Authorization
Headers (1)
Body
Pre-request Script
Tests
form-data x-www-form-urlencoded raw binary XML (text/xml)
1
2 <SetConfig _="ISP" >
3   <TFTP>
4     <Port _="69"/>
5     <Files>
6       <WebSource _="WebSrc.bin" acc="rw" size="1000"/>
7     </Files>
8   </TFTP>
9 </SetConfig>
Body Cookies Headers (4) Tests
Pretty Raw Preview XML
1 <?xml version='1.0' encoding='UTF-8'?>
2 [
3 <SetConfig/>]
  
```

Beispiel 3 
 (Abruf der letzten 3 Einträge
 des Logfiles "Datalogging_0")



```

http://193.101.167.230
POST http://193.101.167.230:8080/cgi-bin/TiXML.cgi?100
Authorization
Headers (1)
Body
Pre-request Script
Tests
form-data x-www-form-urlencoded raw binary XML (text/xml)
1
2 [<ReadLog _="Datalogging_0" range="#3-" />]
Body Cookies Headers (4) Tests
Pretty Raw Preview XML
1 <?xml version='1.0' encoding='UTF-8'?>
2 [
3 <ReadLog>
4   <ID_565574 _="2017/09/20,10:47:30">
5     <Reg _="0"/>
6     <Reg_Text _=""/>
7     <Operator _=""/>
8     <Quality _="0"/>
9     <BitErrorRate _="0"/>
10    <ConnType _="https"/>
11    <ConnState _="1"/>
12    <ConnStateMSG _="connected"/>
13    <Wert0100 _="7"/>
14    <Wert01 _="0"/>
15    <Wert0360 _="0"/>
16    <Wert220250 _="220"/>
17    <WertRandom10 _="0"/>
18    <WertRandom100 _="0"/>
19    <WertRandom360 _="0"/>
20    <WertRandom1000 _="0"/>
21    <WertRandom1001000 _="100"/>
22    <WertRandom012 _="0"/>
23  </ID_565574>
  
```

Anstelle `tixml.cgi` kann als Synonym auch `tixml.exe` verwendet werden.

3 Anhang

3.1 Beispiel URLs

Folgend einige HTTP-Links auf TiXML Datenstrukturen zum direkten Ausprobieren.

Hardware- und Systemdaten

<http://193.101.167.68:8080/System/Properties/>

Datum und Uhrzeit

<http://193.101.167.68:8080/System/Properties/TIMES/RFC822Date>

Filesystem-Speicher

<http://193.101.167.68:8080/System/Properties/Hardware/FileSystem/Size>

SPS-Daten

http://193.101.167.68:8080/System/Properties/Process/Bus1/Device_0/

TDG-Konfiguration komplett

<http://193.101.167.68:8080/System/Config/>

Data Logging Konfiguration

<http://193.101.167.68:8080/System/Config/LOG/LogDefinition/>

SPS Konfiguration

<http://193.101.167.68:8080/System/Config/PROCCFG/External/>

3.2 TDG Systemdaten

Hier der komplette Systemdatenweig eines HE423-M25 mit angeschlossenen M-Bus Zähler:

```
<SYSTEM>
  <Hardware>
    <Modules>
      <RTC _="RTC8564" />
      <Modem0 _="HE423-M25" />
      <FlashOnboard _="6MB" />
      <PowerSupply _="2.0A" />
      <COM1 _="RS232 (Host)" />
      <COM2 _="RS232" />
      <COM3 _="MBUS" />
      <ETH1 _="DM9000 (Rev. 19)" />
      <MBDIO _="GP21D-I/O" />
    </Modules>
  <RAM>
    <Size _="8388608" />
    <Attributes _="0" />
  </RAM>
  <ROM>
    <Size _="8388608" />
    <Attributes _="-939524096" />
  </ROM>
  <FileSystem>
    <Size _="6815744" />
```



```

    <Type _="2" />
    <Attributes _="0" />
  </FileSystem>
</Hardware>
<Firmware>
  <Version _="3.0.6.70" />
  <Date _="2008-11-17 14:36:23" />
</Firmware>
<LicenseRef>
  <UDID _="Tixi.com GM20-S1F2K-120 70426-02489601" />
  <LicenseID _="000100" />
  <ProductID _="1330" />
  <Version _="5.00.12" />
  <OName _="HE423-M25" />
  <ProdName _="HE423-M25" />
  <Oem _="FP InovoLabs GmbH " />
  <PClass _="Tixi Alarm CPU Ethernet G5" />
  <QS _="78001R283" />
  <HW _="970220021384" />
</LicenseRef>
<EEProm>
  <LED0 _="0" />
</EEProm>
<Process>
  <Bus1>
    <Device_0>
      <DeviceState _="1" />
      <ChangeToggle _="0" />
      <Variable_01 _="0.000" />
      <Variable_02 _="0.000" />
      <Variable_03 _="0.000000" />
      <Variable_04 _="-7755.700" />
      <Variable_05 _="25.200" />
      <Variable_06 _="25.800" />
      <Variable_07 _="-0.600" />
      <Variable_08 _="29911" />
      <Variable_09 _="2008/12/03,14:27" />
    </Device_0>
    <Active _="1" />
    <FreeMem _="164766" />
  </Bus1>
  <PV>
    <Alarm_0_ProcVar _="0" />
    <Alarm_1_ProcVar _="0" />
    <SummerFlag _="1" />
  </PV>
  <Program>
    <Mode _="Run" />
  </Program>
  <MB>
    <IO>
      <I>
        <P0 _="1" />
        <P1 _="1" />
      </I>
      <IB>
        <P0 _="3" />
      </IB>
    </IO>
  </MB>
</Process>

```

```

</IB>
<IW>
  <P0 _="3" />
</IW>
<ID>
  <P0 _="3" />
</ID>
<Q>
  <P0 _="0" />
</Q>
<QB>
  <P0 _="0" />
</QB>
<QW>
  <P0 _="0" />
</QW>
<QD>
  <P0 _="0" />
</QD>
</IO>
<FirstCycle _="0" />
<PollButton _="0" />
<ModemOffHook _="0" />
<TransMode _="0" />
<MaxCycleTime _="10" />
<CycleTime _="10" />
<SignalLED _="0" />
<SignalLED2 _="0" />
</MB>
</Process>
<LogCounter>
  <JobReport _="49" />
  <Event _="842" />
  <Login _="0" />
  <IncomingMessage _="0" />
  <FailedIncomingCall _="0" />
  <SupportLog _="595" />
  <Datalogging_0 _="198" />
  <FatalSystemError _="0" />
</LogCounter>
<GSM>
  <SM />
  <FD />
</GSM>
<TIMES>
  <TIME _="14:55:51" />
  <DATE _="2008/12/03" />
  <RFC822DATE _="Wed, 03 Dec 08 14:55:51 +0100" />
  <PowerOffTime _="2008/11/20,13:30:00" />
  <PowerOnTime _="2008/12/02,15:11:19" />
  <DAYOFWEEK _="Wed" />
  <DAYOFWEEKNO _="3" />
  <YYYY_MM_DD _="2008_12_03" />
  <HH_MM_SS _="14_55_51" />
  <HEXDATE _="49369DF7" />
</TIMES>
<Ethernet>

```

```

<Link _="100" />
<LinkState _="1" />
<AssignedIP _="193.101.167.187" />
<SubnetMask _="255.255.255.192" />
<MAC _="00:11:E8:03:cc:84" />
<Gateway _="193.101.167.129" />
<DNS_1 _="193.101.167.2" />
</Ethernet>
<WLAN>
  <LinkState _="0" />
</WLAN>
<FreeFileSize _="6082560" />
<PNP_String _="TIX2027\02489601\MODEM\AMB3100\Tixi Alarm CPU Ethernet
G5" />
<FeatureList _="Debug, Modem Mode, Default, TSAdapter, POP3 Client,
HTTP Server In, CGI DoOn, HTTP Server Out, Time Client, URL Send, SMTP
Client, Script Send, Print Jobs, Job Result Processor, Remote ModemMode,
TSAdapterCallback" />
<SerialNo _="02489601" />
<HardwareID _="GM20-S1F2K-120" />
<Components _="RTC=RTC8564;Modem0=HE423-
M25;FlashOnboard=6MB;PowerSupply=2.0A;COM1=RS232
(Host);COM2=RS232;COM3=MBUS;ETH1=DM9000 (Rev. 19);MBDIO=GP21D-I/O" />
<BoxMode _="TiXML" />
</SYSTEM>

```

3.3 M-Bus Konfiguration

Konfiguration zweier M-Bus-Zähler:

```
<External>
```

```
<Bus Name="Bus1" _="COM3" family="M-Bus" Product="M-Bus Standard" proto-
col="Meterbus" baud="2400" handshake="none" type="Master">
```

```
<Device Name="Device_0" NameUser="EG1" Producer="TCH - Techem AG" dev-
Type="m-bus S / III Warmwasser" _="0" Pollrate="10s" Secondar-
yAddr="26907819">
```

```
<Variable_0 Name="EG1-Sekundaeradresse" _="ident" simpleType="meterbus"
ind="0" size="8" acc="R"/>
```

```
<Variable_10 Name="EG1-Hersteller" _="manufacturer" simple-
Type="meterbus"
ind="0" size="5" acc="R"/>
```

```
<Variable_17 Name="EG1-02-Volumen" _="value" sub="2" simple-
Type="meterbus"
ind="1" exp="-3" size="9" acc="R"/>
```

```
<Variable_21 Name="EG1-06-Durchfluss" _="value" sub="6" simple
Type="meterbus" ind="3" exp="-6" size="9" acc="R"/>
```

```
<Variable_2 Name="EG1-09-Stichdatum" _="value" sub="9" simple-
Type="meterbus"
ind="9" size="9" acc="R"/>
```

```
<Variable_1 Name="EG1-08-Betriebsstunden" _="value" sub="8" simple
Type="meterbus" ind="99" exp="0" size="9" acc="R"/>
```

```
</Device>
```

```
<Device Name="Device_1" NameUser="EG2" Producer="HYD - Hydrometer" dev-
Type="SCYLAR II Waerme (Vorlauf)" _="0" Pollrate="5s" Secondar-
yAddr="31732144">

  <Variable_3 Name="EG2-ResCode" _="Reset" ind="0" acc="W" def="00"/>

  <Variable_4 Name="EG2-Sekundaeradresse" _="ident" simpleType="meterbus"
  ind="0" size="8" acc="R"/>

  <Variable_5 Name="EG2-Hersteller" _="manufacturer" simple-
Type="meterbus"
  ind="0" size="5" acc="R"/>

  <Variable_8 Name="EG2-03-Vorlauftemperatur" _="value" sub="3" simple
  Type="meterbus" ind="2" exp="-3" size="9" acc="R"/>

  <Variable_9 Name="EG2-04-Ruecklauftemperatur" _="value" sub="4" simple
  Type="meterbus" ind="3" exp="-3" size="9" acc="R"/>

  <Variable_6 Name="EG2-01-Energie" _="value" sub="1" simple-
Type="meterbus"
  ind="4" exp="0" size="9" acc="R"/>

  <Variable_7 Name="EG2-02-Volumen" _="value" sub="2" simple-
Type="meterbus"
  ind="5" exp="0" size="9" acc="R"/>

  <Variable_15 Name="EG2-09-Stichdatum" _="value" sub="9" simple
  Type="meterbus" ind="8" size="9" acc="R"/>
</Device>

</Bus>

</External>
```