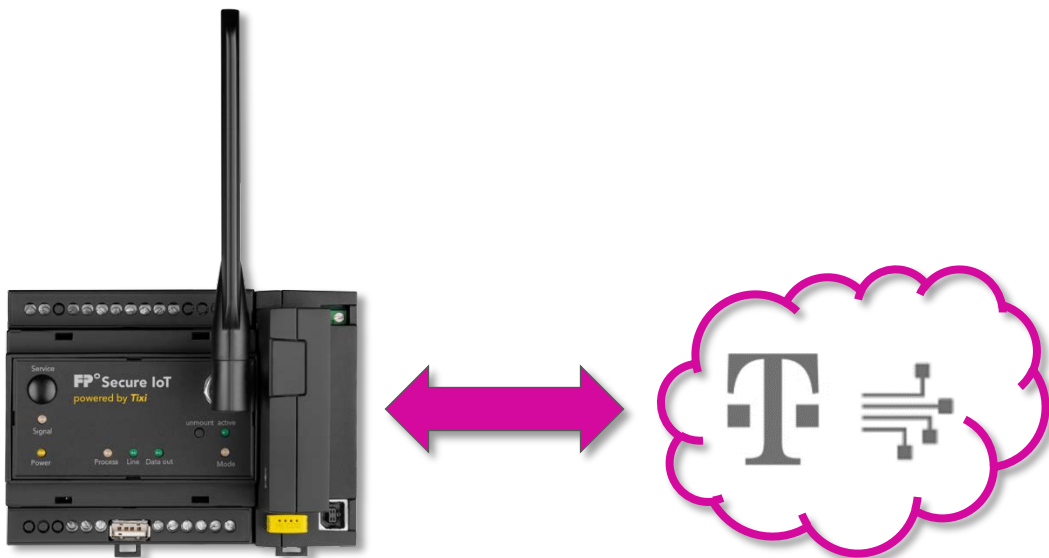


Telekom CoT-Client für FP Gateways Handbuch



Version: 1.0.2

© 2019 -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	EINFÜHRUNG	3
1.1	TLS-Verschlüsselung	3
1.2	Sendeintervall	3
2	TIXML-KONFIGURATION DES COT-CLIENTS	3
2.1	Realtime-Sektionen	5
2.2	Senden von Daten über EventHandler oder Kommando	6
2.3	Hysterese	7
2.4	Optische Signalisierung der MQTT-Verbindung	8
2.5	Service-Routing	8
3	REGISTRIERUNG DES FP-GATEWAYS IN DER CLOUD DER DINGE (COT)	9
4	RÜCKKANAL ZUM FP-GATEWAY (SHELL)	13
4.1	Fehlermeldungen des Gateways	15
5	ALARMIERUNGSFUNKTION	16
5.1	Überblick	16
5.2	TiXML-Parametrierung	17

1 Einführung

Ab Firmware-Version 5.2.6.36 können FP-Gateways sich mit der Cloud der Dinge (CoT) der Deutschen Telekom verbinden. Nähere Informationen zur Cloud der Dinge erhalten Sie hier:

<https://iot.telekom.com/iot-de/plattformen/cloud-der-dinge>

Einen kostenlosen Testzugang zur Cloud der Dinge können Sie über folgenden Link erhalten:

<https://support.ram.m2m.telekom.com/apps/demotenant/index.html#/>

1.1 TLS-Verschlüsselung

Die Verbindung zum Broker erfolgt grundsätzlich verschlüsselt (Port 8883). Unverschlüsselte Verbindungen sind im Netz der Deutschen Telekom aus Sicherheitsgründen nicht möglich.

1.2 Sendeintervall

Sobald eine MQTT-Verbindung aufgebaut wurde, sendet der Client regelmäßig Prozessdaten zum Telekom-Broker. Das Sende-Intervall kann über den Parameter (RTSendTimeX) eingestellt werden.

2 TiXML-Konfiguration des CoT-Clients

Der CoT-Client wird über eine TiXML-Datenbank konfiguriert.

```
[<SetConfig _="ISP" ver="y">
  <CloudConn>
    <!-- connection with tls/ssl -->
    <!-- replace nb-iot.ram.m2m.telekom.com with the URL given
         from Deutsche Telekom for your CoT tenant -->
    <CloudBaseUrl _="ssl://nb-iot.ram.m2m.telekom.com:8883"/>
    <!-- if set to 1 the MQTT connection will be started at device
         startup; otherwise MQTT connection will not be established -->
    <CloudConnStart _="1"/>
    <!-- fixed setting (mandatory) -->
    <CloudConnType _="CoT_MQTTConnector_1"/>
    <!-- RTSendTimeX and UpdateRate1 controls the send interval
         X = 1 .. 5
         RTSendTimeX ranges:
           RTSendTimeX = -1: do not send data using a time
                           or activate hysteresis (RTSendTime1 only)
           RTSendTimeX = 1 .. 2^30: send interval in seconds
           RTSendTimeX = empty: send interval = 3 seconds (default)

         UpdateRate1 ranges (only used when RTSendTime1 = -1)
           UpdateRate1 = 1 .. 3600 = update interval of data
                           points in seconds

         Hysteresis is defined within the RealTime1 section (below).
         In hysteresis mode the variables will only be sent to the
         broker if a pre-defined treshold is reached
         After a connection to the broker has been established all
         variables of a hysteresis RealTime1-section will be sent once.
```

```

        hysteresis ranges: floating point number, max. two digits
        after the decimal point. Examples: 0.1 / 2 / 2.68
-->

<!-- specifies the first send timer, operating mode = hysteresis -->
<RTSendTime1 _="-1" />

<!-- update rate for variables in hysteresis mode in seconds -->
<UpdateRate1 _="20" />

<!-- list of variables for first send timer to be sent to broker
The gateway reads out the current values from the Process
tree using the UpdateRateX. If a data point has changed and
the changes are bigger than the specified hysteresis the data
point will be sent to the broker.
-->
<Realtime1>
  <Datapoint1 _="/Process/Bus1/Device1/Datapoint1" hysteresis="2"/>
  <Datapoint2 _="/Process/Bus1/Device1/Datapoint2" hysteresis="1"/>
  <Datapoint3 _="/Process/Bus1/Device2/Datapoint3" hysteresis="5"/>
  <Datapoint4 _="/Process/PV/ProcVar1" hysteresis="0.4"/>
  <Datapoint5 _="/Process/PV/ProcVar2" hysteresis="1.7"/>
</Realtime1>

<!-- specifies the second send timer, send interval = 10 seconds -->
<RTSendTime2 _="10" />
<Realtime2>
  <Obis_180 _="/Process/Meter/MT174/Obis_180"/>
</Realtime2>

<!-- specifies the third send timer, no send interval -->
<!-- data can be sent event driven using CloudSendRealtimeData -->
<RTSendTime3 _="-1" />
<Realtime3>
  <Obis_181 _="/Process/Meter/MT174/Obis_181"/>
</Realtime3>

<!-- specifies the forth send timer, send interval = 60 seconds -->
<RTSendTime4 _="60" />
<Realtime4>
  <Obis_182 _="/Process/Meter/MT174/Obis_182"/>
</Realtime4>

<!-- specifies the fifth send timer, send interval = 100 seconds -->
<RTSendTime5 _="100" />
<Realtime5>
  <Temp_Board _="/Process/OneWire/Mainboard/TEMP_Board"/>
</Realtime5>
.
.
<!-- specifies the fifth send timer, send interval = 100 seconds -->
<RTSendTime5 _="100" />
<Realtime5>
  <Temp_Board _="/Process/OneWire/Mainboard/TEMP_Board"/>
</Realtime5>

</CloudConn>

</SetConfig>]

```

Bitte beachten Sie folgende Hinweise:

- Die fett gedruckten Konfigurationsoptionen müssen konfiguriert werden (mandatory).
- Es muss eine CloudBaseUrl konfiguriert werden.
- Der Pfad für die Url lautet:
ssl://nb-iot.server.m2m.telekom.com:8883
wobei **nb-iot.server.m2m.telekom.com** die URL Ihres von der Telekom zugeordneten CoT Servers entspricht.
Im Zweifelsfall erfragen Sie die komplette URL von der Deutschen Telekom.
- CloudConnStart und CloudConnType müssen wie oben beschrieben konfiguriert werden.
- Die Daten im Zweig Realtime1 .. RealTime5 können vom Anwender frei konfiguriert werden.
- Die Zeitzone des FP-Gateways muss auf +0000 eingestellt werden. Die Konfiguration erfolgt über den Parameter TimeZone in der USER-Datenbank.
- Die Uhrzeit des FP-Gateways muss auf UTC eingestellt sein, da das Portal der Deutschen Telekom UTC-Zeitstempel erwartet. Anderenfalls werden sind Ihre Datenreihen zeitversetzt angezeigt.

Nach dem Einspielen der MQTT-Client-Konfiguration sollte das FP-Gateway neu gestartet werden. Den Verbindungsstatus des MQTT-Client kann man im Process-Zweig überwachen:

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

Ergebnis (Beispiel):

```
<Get>
  <CloudConn>
    <ConnectionType _="mqtt+ssl:nb-iot.ram.m2m.telekom.com:8883" />
    <ConnectionState _="1" />
    <ConnectionStateMsg _="connected" />
    <LastTimeStamp _="Zeitstempel" />
    <ChangeToggle _="Togglebit" />
  </CloudConn>
</Get>
```

Der ConnectionType zeigt die Art der Verbindung (mqtt+ssl), die URL / IP-Adresse des gerade aktiven Brokers und den Verbindungsstatus an.

ConnectionState = 0 bedeutet: keine Verbindung

ConnectionState = 1 Verbindung ist aufgebaut

ConnectionStateMsg = (not connected, connecting, connected)

LastTimeStamp = Der Zeitstempel zeigt an, wann die letzte MQTT Message gesendet wurde.

Format: DD.MM.YYYY HH:MM:SS Timezone

ChangeToggle = nur bei jedem neuen Zeitstempel (LastTimeStamp) ändert sich das ToggleBit

2.1 Realtime-Sektionen

In bis zu 5 Realtime Sektionen kann festgelegt werden, welche Datenpunkte zu übertragen sind.

Jeder Realtime Sektion muss ihr Zyklus vorangestellt werden. Über den Zykluswert wird auch festgelegt, ob die Daten zyklisch oder über eine Hystereseffunktion übertragen werden.

z.B. RTSendTime1, ..., RTSendTime5.

Dazu gehören dann die Daten aus den entsprechenden Sektionen <RealTime1>..</RealTime1> usw.

Beispiel:

```
<CloudConn>
  <RTSendTime1 _="-1" />
  <UpdateRate1 _=20 />
  <Realtime1>
    <Volt_L1_N _="/Process/Modbus/D1/Volt_L1_N" hysteresis="2" />
  </Realtime1>
</CloudConn>
```

```

    .
  </Realtime1>
  <RTSendTime2 _="10" />
  <Realtime2>
    <Obis_180 _="/Process/Meter/MT174/Obis_180"/>
  </Realtime2>
  .
  <RTSendTime5 _="100" />
  <Realtime5>
    <Temp_Board _="/Process/OneWire/Mainboard/TEMP_Board"/>
  </Realtime5>
</CloudConn>

```

2.2 Senden von Daten über EventHandler oder Kommando

Datenpunkte können per EventHandler oder auch per Kommando verschickt werden:

Kommando (TICO, ...):

```
<CloudSendRealtimeData [_="CloudConn[,Realtime_Specifier"]] />
```

bzw.

```
[<CloudSendRealtimeData />]
```

CloudConn (optional) = Name des CloudConnectors. Werte: CloudConn, CloudConn2, CloudConn3.
Default: CloudConn

Realtime_Specifier (optional) = Name der Realtime-Sektion, z.B. Realtime1. Default=Realtime1
Wird der optionale Parameter weggelassen, werden alle definierten Sektionen versendet.

Beispiel 1: Realtime1-Sektionen von CloudConn versenden

```
<CloudSendRealtimeData />
```

Beispiel 2: Realtime-Sektion "Realtime2" von CloudConn2 versenden

```
<CloudSendRealtimeData _="CloudConn2,Realtime2" />
```

Beispiel 3: Realtime-Sektion "Realtime3" von CloudConn versenden

```
<CloudSendRealtimeData _="CloudConn,Realtime3" />
```

Beispiel 4: Realtime-Sektion "Realtime2" von CloudConn2 über einen EventHandler versenden

```

<CloudSendRealTimeDate_2_2>
  <CloudSendRealtimeData _="CloudConn2,Realtime2" />
</CloudSendRealTimeDate_2_2>

```

Minütliche Ausführung des EventHandlers per Scheduler:

```

<CloudSendRealTimeDate_2_2_SCH _="CloudSendRealTimeDate_2_2">
  <Minute _="0-59" />
</CloudSendRealTimeDate_2_2_SCH>

```

Test per TiXML-Kommando mit TICO

```
[<DoOn _="CloudSendRealTimeDate_2_2" ver="v" />]
```

2.3 Hysterese

Für den Timer 1 (RTsendTime1) ist optional für jeden Datenpunkt einstellbar, wann dieser abhängig von einer Hystereseeinstellung übertragen werden soll.

Beispiel: Datenpunkt „Temperatur“

Startwert = 20,5K

Hysterese absolut = 0,5

Ergebnis: Wenn der Wert des Datenpunktes auf 20 oder 21 wechselt, wird dieser übertragen.

Festlegungen:

- Nach dem Neustart des Gerätes werden alle Werte einmalig übertragen.
- Wenn sich der zuletzt übertragene Wert um den Hysterese-Wert ändert, wird der Wert wieder einmalig übertragen.

Die benötigten Parameter:

```
<RTsendTime1 _="-1" />
```

Der Wert -1 bedeutet, dass Daten nicht mehr zyklisch gesendet werden, sondern anhand der Hysterese-Parametrierung.

```
<UpdateRate1 _="Pollzeit" />
```

Pollzeit spezifiziert, wie häufig die Werte aus dem Process-Zweig abgefragt werden sollen. Die Pollzeit wird in Sekunden angegeben. Wertebereich: 1 .. 3600

Achtung:

Die Hysteresefunktion ist derzeit nur für den ersten Timer verfügbar (RTsendTime1 und UpdateRate1). Die Hysterese ist nur für numerische Werte nutzbar.

Für jeden Realtime-Wert wird in der Konfiguration die Hysterese als Fließkommazahl festgelegt.

Beispiel:

```
hysteresis="1.5"
```

Definition:

- Komma / Dezimalpunkt wird mit "." dargestellt
- Max. 2 Nachkommastellen
- keine negativen Zahlen

Beispiel:

```
<RTsendTime1 _="-1" />
```

```
<UpdateRate1 _="20" />
```

```
<Realtime1>
```

```
<Voltage_L1_N _="/Process/Modbus/D1/Voltage_L1_N" hysteresis="2" />
```

```
<CPULoad _="/Process/MB/CPULoad" hysteresis="8.5" />
```

```
<Vorlauftemp _="/Process/PV/Vorlauftemp" hysteresis="0.5" />
```

```
<Ruecklauftemp _="/Process/PV/Ruecklauftemp" hysteresis="0.8" />
```

```
</Realtime1>
```

Im oben gezeigten Beispiel wird die Hysterese der einzelnen Datenpunkte festgelegt.

Wenn sich der Datenpunkt Voltage_L1_N um +/- 2.0 zum zuvor übertragenen Wert ändert, wird dieser Wert übertragen. Wenn sich der Datenpunkt CPULoad um +/- 8.5 zum zuvor übertragenen Wert ändert, wird dieser Wert übertragen usw.

2.4 Optische Signalisierung der MQTT-Verbindung

Die „Signal“-LED signalisiert den Zustand der MQTT-Verbindung.

Aus = keine MQTT-Verbindung aktiv

Rot blinkend = MQTT-Verbindung wird aufgebaut

Grün leuchtend = MQTT-Verbindung ist aufgebaut

2.5 Service-Routing

Normalerweise wird die Verbindung zur Telekom-Cloud über das aktive LAN-Interface aufgebaut.

Über die TiXML-Datenbank ISP/ISP/OUT kann die MQTT-Verbindung auch über eine

Mobilfunkverbindung oder einen VPN-Tunnel aufgebaut werden:

```
[<SetConfig _="ISP/ISP" ver="y">
  <!-- Kommunikationsinterface fuer Dienste definieren -->
  <OUT>
    <SMTP _="MODEM" />
    <CBIS _="MODEM" />
    <POP3 _="MODEM" />
    <URLSend _="MODEM" />
    <INetTime _="MODEM" />
    <HTTPConn _="MODEM" />
    <CloudConn _="Ethernet" />
    <IBMConn _="MODEM" />
    <FTPPut _="MODEM" />
    <SFTPPut _="MODEM" />
    <VPN _="MODEM" />
  </OUT>
</SetConfig>]
```

Die folgenden Optionen sind verwendbar für den MQTT-Client:

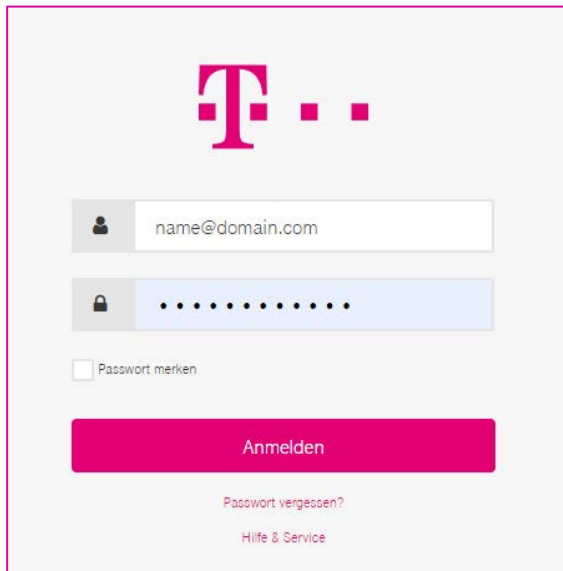
Ethernet

MODEM

VPN

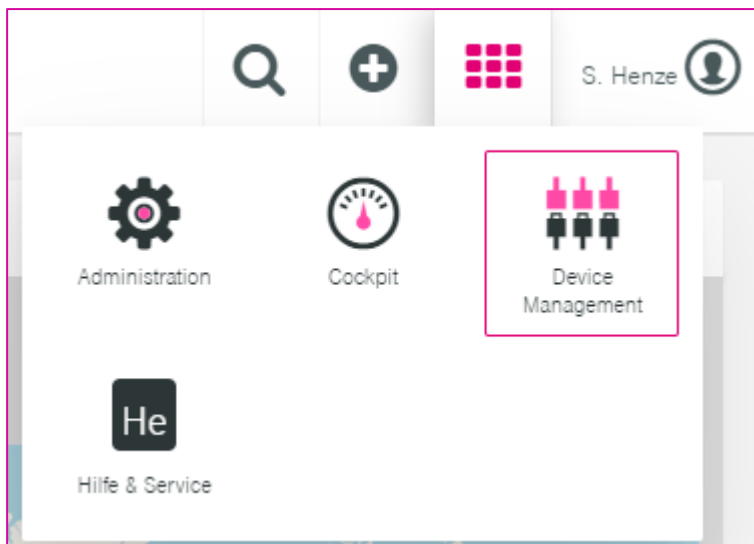
3 Registrierung des FP-Gateways in der Cloud der Dinge (CoT)

Nachdem die Konfiguration in das Gerät überspielt wurde, muss das FP-Gateway in der Cloud der Dinge registriert werden. Loggen Sie sich dazu mit Ihren Zugangsdaten in das Portal der Cloud der Dinge ein.

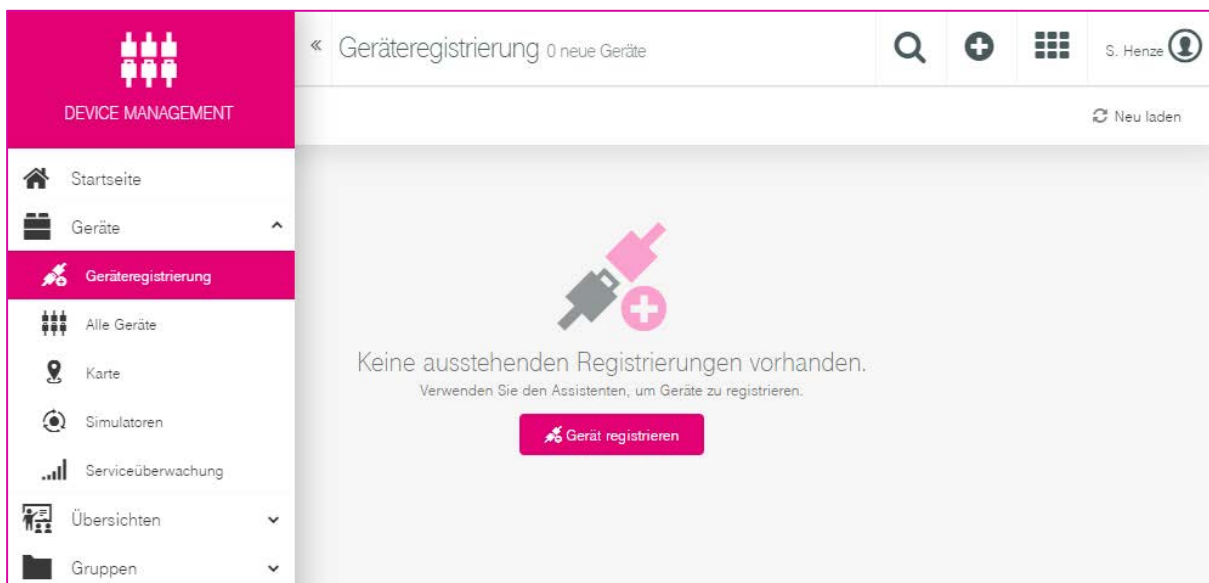


The image shows a login form for the CoT portal. At the top center is the Telekom logo, a stylized 'T' with three dots to its right. Below the logo is a text input field containing 'name@domain.com'. Underneath is a password input field with ten dots. A checkbox labeled 'Passwort merken' is located below the password field. A large red button labeled 'Anmelden' is positioned below the checkbox. At the bottom of the form, there are two links: 'Passwort vergessen?' and 'Hilfe & Service'.

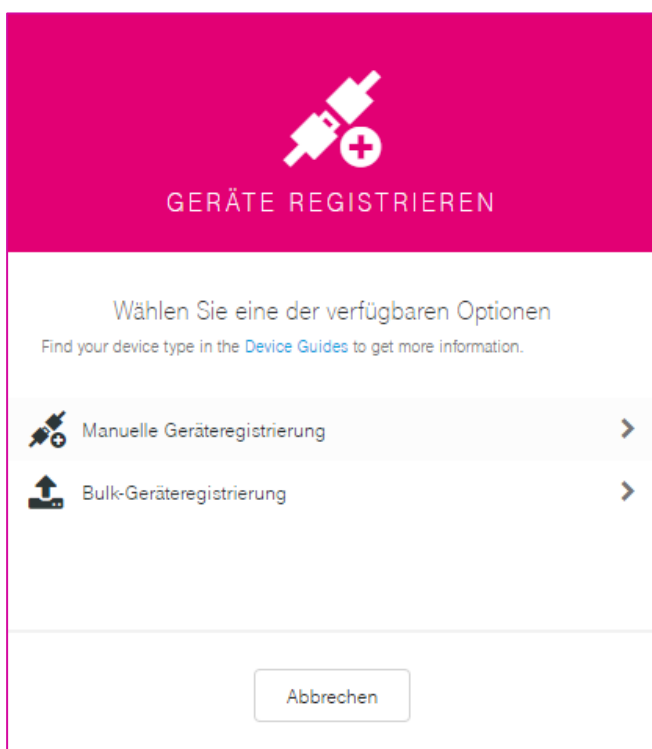
Rufen Sie dann das Devicemanagement auf, indem Sie auf das Menü-Symbol oben rechts klicken und dann den Menüeintrag „Device Management“ anklicken:



Klicken Sie auf der linken Seite auf „Geräte -> Geräteregistrierung“.



Klicken Sie auf „Manuelle Geräteregistrierung“.



Geben Sie dann in das Feld „GERÄTE-ID“ die Seriennummer Ihres FP- Gateways ein. Die 8-stellige Seriennummer finden Sie auf dem Prüfaufkleber Ihres Gateways. Die Nummer startet in der Regel mit „04“.

GERÄTE REGISTRIEREN

GERÄTEINFORMATION GERÄTEREGISTRIERUNG

GERÄTE-ID

ZU GRUPPE HINZUFÜGEN

Gruppe auswählen

+ Weiteres Gerät hinzufügen

Zurück Weiter

Klicken Sie dann auf „Weiter“.

Sie erhalten die Meldung, dass das Gerät erfolgreich registriert wurde:

GERÄTE REGISTRIEREN

GERÄTEINFORMATION GERÄTEREGISTRIERUNG

Gerät erfolgreich registriert.

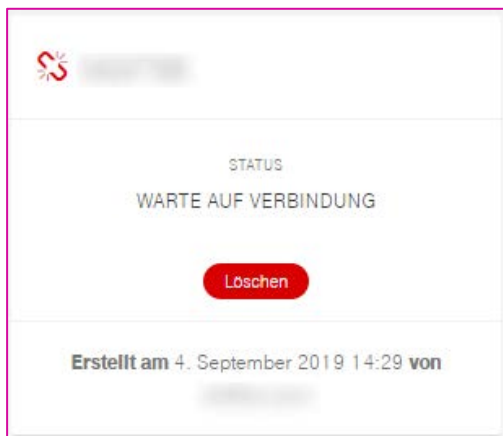
Schalten Sie das/die registrierte/n Gerät(e) ein und warten Sie, bis die Verbindung(en) hergestellt ist/sind. Sobald ein Gerät verbunden ist, ändert sich der Status auf "Bitte akzeptieren". Klicken Sie zum Bestätigen auf die Schaltfläche "Akzeptieren".

✓

Beenden

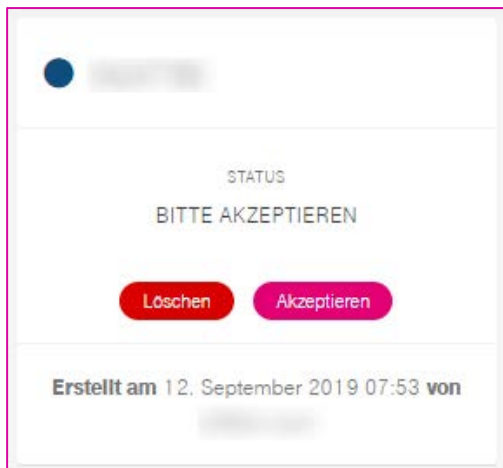
Klicken Sie auf „Beenden“.

Im Telekom-Portal sehen Sie dann folgende Anzeige:



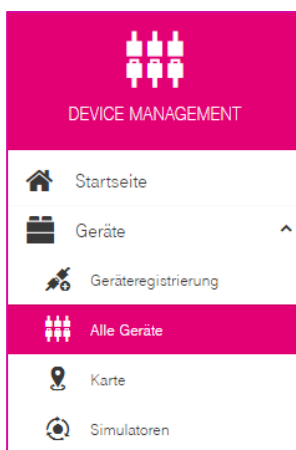
Nach ca. 30 .. 120 Sekunden sollte am Gateway die LED „Signal“ rot blinken.

Anschließend sehen Sie im Portal eine zusätzliche Schaltfläche „Akzeptieren“:

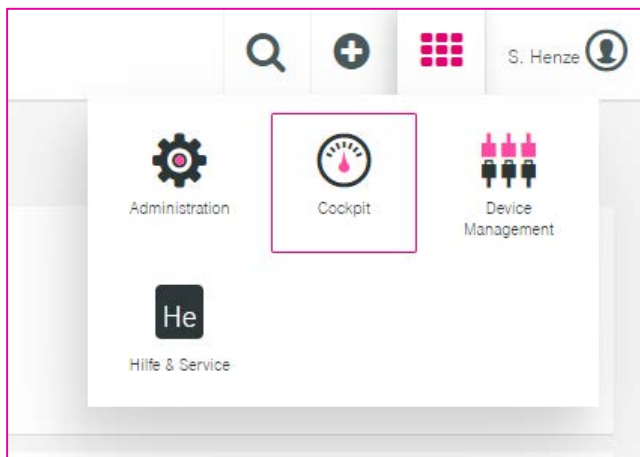


Klicken Sie nun auf die Schaltfläche „Akzeptieren“.
Damit ist die Geräteregistrierung abgeschlossen.

Klicken Sie auf den Menüeintrag „Alle Geräte“ auf der linken Seite und kontrollieren Sie, ob Sie Ihr Gerät in der Geräteliste finden.



Wechseln Sie nun in die Cockpit-Ansicht, indem Sie auf den Menüeintrag oben rechts klicken:



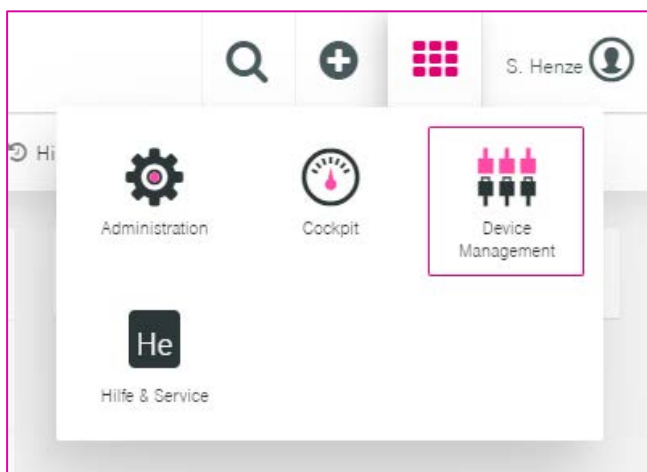
In der Cockpit-Ansicht rufen Sie den Daten-Explorer auf und fügen alle Datenpunkte hinzu, die für Sie relevant sind.

Anschließend können Sie eigene Widgets erstellen, um die Daten zu visualisieren.

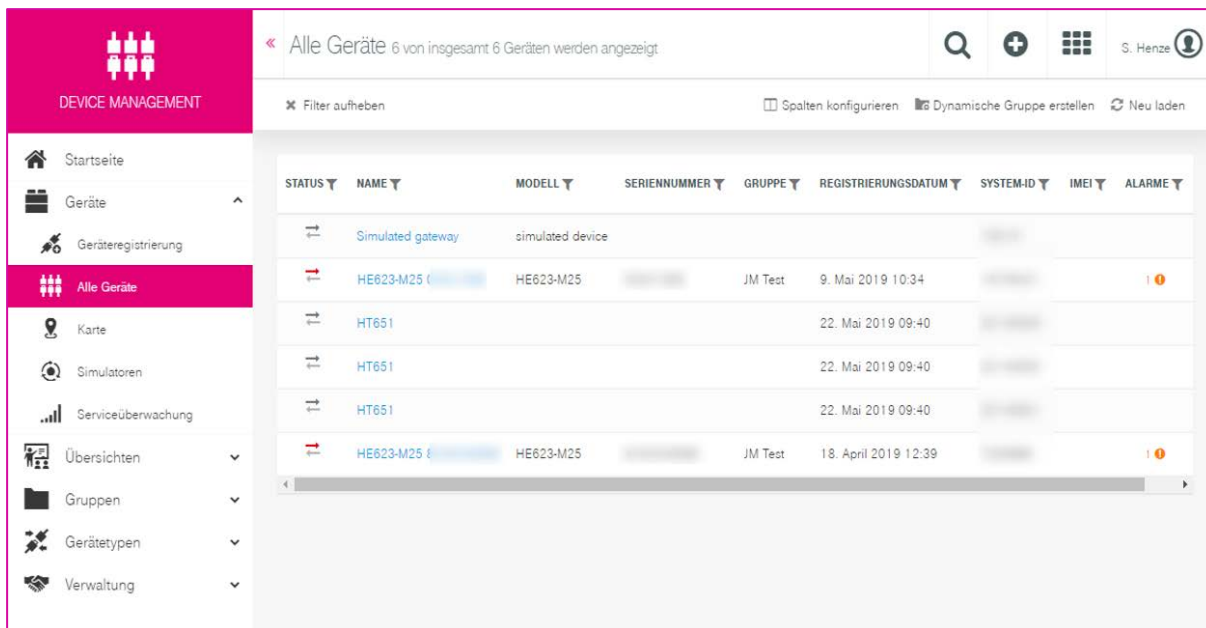
4 Rückkanal zum FP-Gateway (Shell)

Über einen Rückkanal kann der Telekom-Broker TiXML-Befehle an das FP-Gateway senden, um so zum Beispiel eine neue Konfiguration in das Gerät zu speichern, das Gerät neu zu starten oder einen Ausgang zu schalten. Der Rückkanal wird automatisch eingerichtet und kann über die „Shell“ in der Telekom CoT Oberfläche aufgerufen werden.

Um den Rückkanal zu nutzen, wechseln Sie zunächst in das „Device Management“.

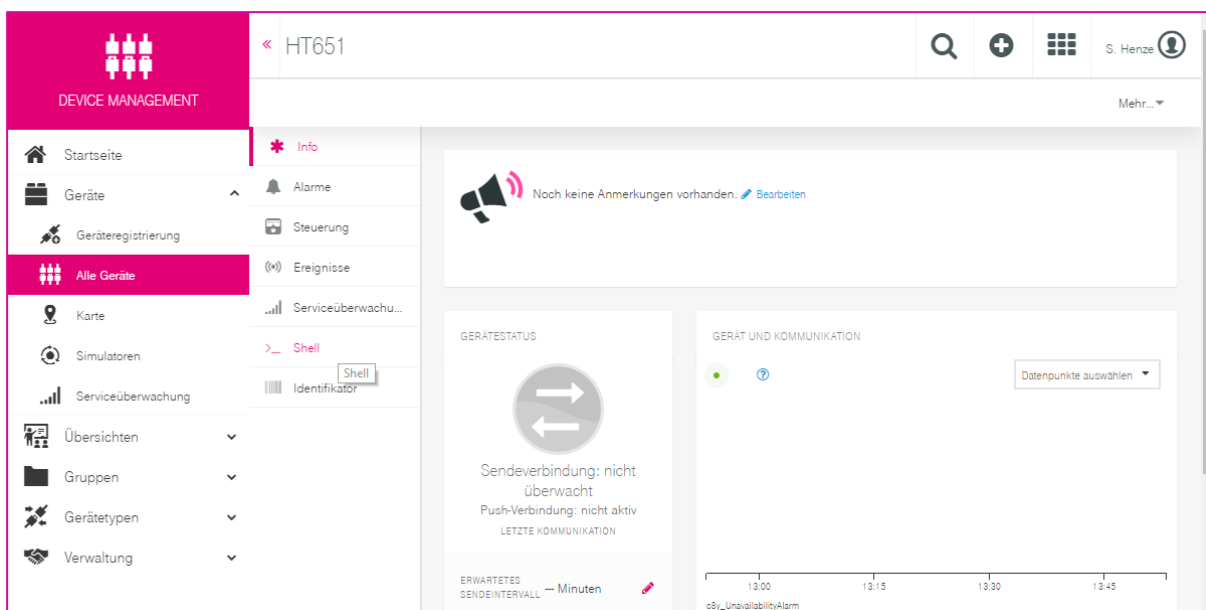


Klicken Sie dann im Menü auf der linken Seite den Eintrag „Geräte“ an.

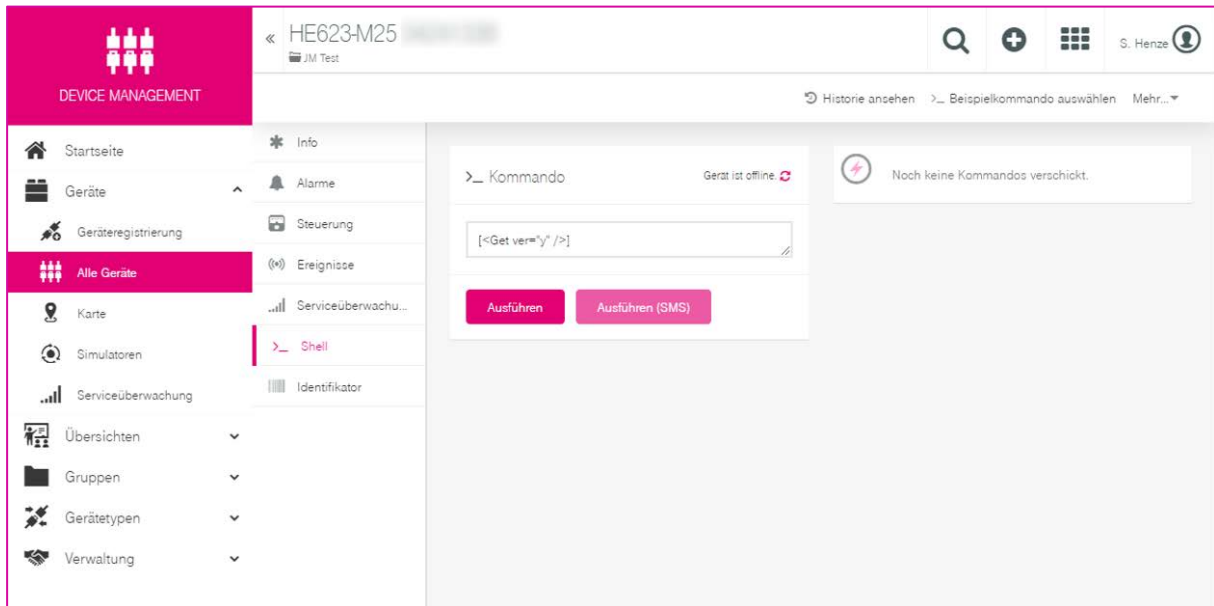


Wählen Sie anschließend aus der Liste auf der rechten Seite das gewünschte Gerät aus, in dem Sie auf den Eintrag in der Spalte „Name“ klicken.

Klicken Sie dann auf den Eintrag „>_ Shell“ auf der linken Seite.



Im Feld „>_“ können Sie nun ein gültiges TiXML-Kommando eintragen oder durch Anklicken der Option „>_ Beispielkommando ...“ ein vorher definiertes Kommando auswählen.



Durch Klick auf die Schaltfläche „Ausführen“ wird das Kommando zum Gateway geschickt. Nach wenigen Sekunden sollte die Antwort des Gateways erfolgen. Die Antwort wird oben rechts angezeigt und kann expandiert werden.

Wichtig:

Vor jedem Konfigurationsbefehl (also alle `SetConfig`-Befehle) sollte immer die Prozessbearbeitung mit folgendem Befehl gestoppt werden:

```
[ <Set _="/Process/Program/Mode" value="Stop" ver="v" /> ]
```

Das Ergebnis des oben genannten Befehls sollte abgewartet werden. Nur wenn der o.g. Stop-Befehl erfolgreich ausgeführt wurde, sollte anschließend ein `SetConfig`-Befehl geschickt werden.

Wenn innerhalb von max. 15 Sekunden keine Antwort auf den Stop-Befehl empfangen wird, muss der Stop-Befehl erneut gesendet werden.

Nachdem die Konfigurationsdatenbank(en) an das Gerät geschickt wurden, kann dann die Prozessbearbeitung mit folgendem Befehl wieder gestartet werden:

```
[ <Set _="/Process/Program/Mode" value="Run" ver="v" /> ]
```

4.1 Fehlermeldungen des Gateways

Wenn das Gateway Kommandos vom Broker empfängt und diese Kommandos ausführt, werden unter Umständen Fehlermeldungen generiert, wenn das Kommando z.B. falsche Parameter enthält oder aus anderen Gründen nicht ausführbar ist.

Über einen Parameter beim Senden des Kommandos kann dabei spezifiziert werden, wie ausführlich die Fehlermeldungen vom Gateway beantwortet werden.

Es gibt drei mögliche Formate, mit denen diese Fehler mitgeteilt werden. Abhängig von der Vorgabe durch den Parameter „ver“, erfolgt eine kurze oder lange Fehlermeldung (siehe nachfolgende Tabelle).

Beispielkommando:

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

vmode	Beschreibung	Beispiel
n	Fehlermeldung als numerischer Wert	<code><Error _="-2196" /></code>
y	Fehlermeldung, kurze Beschreibung	<pre> <Error> <ErrNo _="-2196" /> <ErrMsg _="path to key not found" /> <ErrorCause> <ErrNo _="-2196" /> <ErrMsg _="path to key not found" /> <Class _="TXSTCPGetSetValueCmd" /> </ErrorCause> </Error> </pre>
v	Fehlermeldung, lange Beschreibung	<pre> <Error> <ErrNo _="-2196" /> <ErrMsg _="path to key not found" /> <ErrorCause> <ErrNo _="-2196" /> <ErrMsg _="path to key not found" /> <Line _="127" /> <Module _="SSet.cpp" /> <Class _="TXSTCPGetSetValueCmd" /> </ErrorCause> </Error> </pre>

5 Alarmierungsfunktion

5.1 Überblick

Die FP Gateways lassen sich über TiXML so konfigurieren, dass Alarme direkt in der Cloud ausgelöst werden können. Diese Alarme können dann in der Cloud z.B. mit Hilfe einer Smart Rule weiterverarbeitet werden.

Folgende Daten werden bei einem Alarm an die Cloud übermittelt:

type: Alarmtyp als Text (frei konfigurierbar im Gateway); max. 50 Zeichen
datetime: Datum und Uhrzeit des Alarms (Format: ISO 8601) und Zeitzone
text: Beschreibender Text (frei konfigurierbar); max. 50 Zeichen
status: Alarmstatus. Werte: 1 (=ACTIVE) oder 0 (=CLEARED); 1 Zeichen
severity: Schweregrad. CRITICAL, MAJOR, MINOR, WARNING

Über die Konfiguration können Alarme für eine ganze Gruppe von Datenpunkten oder anderen System Properties bis hin zu jedem einzelnen Datenpunkt erzeugt werden. Die Erkennung und Triggerung der Alarme erfolgt dabei genau wie bei den lokalen Alarmen (SMS, E-Mail) des Gateways. Der Alarm selbst wird dann über den EventHandler-Befehl `SendAlarmECN` ausgelöst. Zusätzlich sendet der Alarmhandler des FP Gateways für jeden Alarm eine fortlaufende ID.

Die gesendeten Alarme werden auch in der Systemdatenbank „Event“ protokolliert. Beispiel:

```

<ID_31 _="2020/08/14,09:18:58">
  <Event>
    <Event _="ECN_Alarm" Origin="EventState">
      <AlarmCode _="ServiceButton" />
      <AlarmText _="Service button alarm triggered" />
      <State _="1" />
      <TimeStamp _="2020/08/14 09:18:58 +0000" />
      <Severity _="WARNING" />
    </Event>
  </Event>
</ID_31>

```


5.2 TiXML-Parametrierung

Grundprinzip für die TiXML-Parametrierung

EventState -> triggert EventHandler -> ruft SendAlarmECN auf

SendAlarmECN verwendet MessageJobTemplates und MessageText Definitionen und nutzt ggfs. ProcessVars und verschiedene SystemProperties

Die grundlegenden Mechanismen zur Konfiguration der oben genannten Datenbanken sind im TiXML Reference Manual zu finden.

Beispielkonfiguration

Im folgenden Beispiel wird der Serviceknopf zur Triggerung eines Cloud Alarms verwendet. Über den EventState „PollButton_Alarm“ wird jeweils der EventHandler „ECN_Alarm“ getriggert, wenn der Serviceknopf gedrückt oder wieder losgelassen wird. Dabei werden dem EventHandler fünf zusätzliche Parameter übergeben.

14-EventStates.txt

Ein EventState wird getriggert, sobald der Nutzer den Serviceknopf drückt.

In diesem Fall wird der EventHandler "ECN_Alarm" mit zusätzlichen Parametern aufgerufen with (AlarmCode, AlarmText, State, TimeStamp, und Severity).

Die Parameter werden zum Teil über Referenzen ermittelt, z.B. wird als AlarmState der Zustand des Serviceknopfes (1=gedrückt; 0=losgelassen) übertragen. Die Uhrzeit wird aus der SystemProperty TIMES/DATE und TIMES/TIME ausgelesen.

```
[<SetConfig _="PROCCFG" ver="y">
<EventStates>
<!-- ALARM Emergency-Off -->
<PollButton_Alarm>
<Enabled _="TRUE" />
<ProcessVar _="/Process/MB/PollButton" flank="both" />
<Event _="ECN_Alarm" >
<AlarmCode _="ServiceButton" />
<AlarmText _="Service button alarm triggered" />
<State _="#xae;/Process/MB/PollButton;" />
<TimeStamp _="#xae;/TIMES/DATE; &#xae;/TIMES/TIME; &#xae;/USER/USER/TimeZone;" />
<Severity _="WARNING" />
</Event>
</PollButton_Alarm>
</EventStates>
</SetConfig>]
```

Es lassen sich insgesamt maximal 100 unterschiedliche EventStates konfigurieren.

10-EventHandler.txt

Die EventHandler Konfiguration (generisch, kann für alle Cloud-Alarme verwendet werden) wird wie folgt definiert:

```
[<SetConfig _="TEMPLATE" ver="y">
<EventHandler>
<ECN_Alarm>
<!-- compute new Alarm_Id -->
<Process _="Counter"/>
<!-- now send alarm using cloudConnector to cloud -->
<SendAlarmECN _="MessageJobTemplates/Alarm_ECEN" />
<!-- compute new Alarm_Id -->
<Process _="Counter"/>
</ECN_Alarm>
</EventHandler>
</SetConfig>]
```

11-MessageJobTemplates.txt

Die MessageJobTemplate Konfiguration (generisch, kann für alle Cloud-Alarme verwendet werden) wird wie folgt definiert:

```
[<SetConfig _="TEMPLATE" ver="y">
<MessageJobTemplates>
  <!-- send alarm to cloud -->
  <Alarm_ECN _="WriteFile">
    <Body _="/D/UserTemplates/ECN_AlarmText" />
  </Alarm_ECN>
</MessageJobTemplates>
</SetConfig>]
```

12-MessageText.txt

Die MessageText Konfiguration (generisch, kann für alle Cloud-Alarme verwendet werden) wird wie folgt definiert:

```
[<SetConfig _="TEMPLATE" ver="y">
<!-- Definition of UserTemplates database -->
<UserTemplates>
  <LocationText hidden="1">
    <Email>
      <C _="" />
    </Email>
    <SMS _="" />
  </LocationText>
  <Attachments hidden="1" />
  <!-- alarm text for cloud connector in xml format -->
  <ECN_AlarmText>
    <S _="&lt;alarm id=&quot;&#xae;/Process/PV/Alarm_Id&quot; " />
    <S _="type=&quot;&#xae;~/AlarmCode&quot; " />
    <S _="datetime=&quot;&#xae;~/TimeStamp&quot; " />
    <S _="errorcode=&quot;&#xae;~/AlarmText&quot; " />
    <S _="state=&quot;&#xae;~/State&quot; " />
    <S _="severity=&quot;&#xae;~/Severity&quot; " />
    <S _="&gt;" />
  </ECN_AlarmText>
</UserTemplates>
</SetConfig>]
```

13-ProcessVars

In der ProcessVars Konfiguration müssen mindestens 2 Process Variables (PV) definiert werden: Alarm_Id und Counter:

```
[<SetConfig _="PROCCFG" ver="y">
<ProcessVars>
<!-- Alarm ID -->
<Alarm_Id def="0" />
<!-- Alarm_Id will be increased automatically via event handler -->
<Counter>
  <Process>
    <LD _="/Process/PV/Alarm_Id" />
    <ADD _="1" />
    <ST _="/Process/PV/Alarm_Id" />
  </Process>
</Counter>
</ProcessVars>
</SetConfig>]
```

Bitte beachten:

Es kann jeder beliebige digitale Eingang, eine SPS-Variable (External) oder Process Variable (PV) verwendet werden, um den Alarm zu triggern.