Modbus RTU Autokonfiguration für FP Gateways
Kurzdokumentation

Version: 1.0.1

www.inovolabs.com

Redaktionsschluss: 23.02.2021

Inhaltsverzeichnis

1 Einführung

FP Gateways ab Firmware Version 5.1.7.0 unterstützen einen Modbus RTU Scan-Algorithmus.
Der Modbus-RTU-Scan funktioniert ähnlich dem M-Bus Scan. Für eine maximale Flexibilität wurden folgende Eigenschaften implementiert:

a) Zwei Scan-Modi
1. Nur Anzeige des Scan-Ergebnisses
2. Automatische Konfiguration von External / LOG / EVENTS / SCHEDULE

Beim Scan kann optional eine Startadresse und Endadresse angegeben werden (Modbus Stationsnummer)

b) Variable Suchkriterien
Der Scan erfolgt je Device mit unterschiedlichen vielen Suchkriterien (=Datenpunkten)
* Beim Vergleich mit den Suchkriterien sollen auch "Wildcards" erlaubt sein.
Beim Scan der Modbus-Geräte liest man z.B. ein Register, in dem die Seriennummer hinterlegt ist.
Von der Seriennummer sollen aber z.B. nur die ersten 2 Stellen verglichen werden.
* Es soll auch möglich sein, für ein Register mehrere Werte für den Vergleich zu hinterlegen (ODER-Verknüpfung).

c) Selektive automatische Konfiguration
Die automatische Konfiguration soll selektiv erfolgen, d.h. man soll je Datenbank (External / LOG / EVENTS / SCHEDULE) festlegen können, daß z.B. nur die External-Datenbank geschrieben wird, aber nicht die LOG-Datenbank und keine SCHEDULE-Datenbank

d) Logging mit zwei Modi
1. Logging in eine globale Datei
2. Logging in Einzeldateien je Device

e) Anzeige der Scanaktivitäten
Beginn, Ende und Ergebnis des Scans soll über Variablen im Process-Zweig angezeigt werden:
/Process/ScanActive
/Process/LastScanFound

Die XML-Gerätedatenbank "DEVICEDB" enthält eine Liste bekannter Gerätetypen (Gerätevorlagen) und ist jederzeit anpassbar. Zur eindeutigen Erkennung der Geräte werden je Devicetyp eine oder mehrere Modbus Register abgefragt und mit einem oder mehreren Vergleichswerten verglichen.
In der Datenbank kann festgelegt werden, welche Modbus Datenpunkte überhaupt gelesen werden sollen und welche Datenpunkte geloggt werden sollen.

1.1. Option 1: Nur Anzeige der Scan-Ergebnisse

Der einfachste Modus beim Modbus-Scan ist die Anzeige der Scan-Ergebnisse.

Dazu wird der Befehl zum Scannen des Modbus wie folgt aufgerufen:

```
[<ScanDevices _="COM2" protocol="Modbus,RTU" baud="9600" handshake="HALF"
              type="Master" format="8N1" [Start="StartAddr"
[End="EndAddr"]] />]
```

Der Modbus wird gescannt und als Ergebnis eine Liste der gefundenen Geräte ausgegeben:

```
<ScanDevices>
<Device _="0" Type="UMG 96 RM" Version="2.5.0.4">
  <Var_001 _="C" ind="0x200" simpleType="float" value="0"/>
  <Var_002 _="C" ind="0x201" simpleType="float" value="1"/>
  <Var_003 _="I" ind="0x210" simpleType="float" value="10"/>
  <Var_004 _="R" ind="0x212" simpleType="float" value="50"/>
  <Var_005 _="H" ind="0x220" simpleType="float" value="5500"/>
  <Var_006 _="D" ind="0x230" simpleType="float" value="555500"/>
</Device>
... (weitere devices)

<UnknownDevices>
  <StationID _="100"/>
  <StationID _="120"/>
  <StationID _="121"/>
</UnknownDevices >

</ScanDevices>
```

Die Liste der gefundenen Geräte ist eine Kopie der Variablenliste des Gerätes in der **DEVICEDB**. **_="0"** ist die Stations-ID auf dem Modbus (hier also 0), **Type** und **Version** werden aus den **<Ident_xx>**-Einträgen **ModbusDeviceType** und **ModbusDeviceVersion** kopiert (siehe Kapitel 1.4.1.2).

Nicht erkannte Geräte werden in der Sektion "**UnknownDevices**" mit der dazugehörigen Stationsadresse angezeigt. **value** zeigt den Wert des Registers zum Zeitpunkt des Scans an. **StartAddr / EndAddr** = Startadresse / Endadresse (=Modbus Stationsnummer) für den Scan, optional

1.2. Option 2: Automatische Konfiguration

Sofern die Gerätedatenbank **DEVICEDB** im Gerät vorhanden ist, kann die komplette automatische Konfiguration durch den Aufruf des Befehls **<ScanDevices>** initiiert werden. Dabei werden alle internen Konfigurationsdatenbanken wie die **External**, **LogDefinition**, **EventHandler** und **Scheduler** bei Bedarf automatisch erzeugt. Die Ausgabe des Scan-Ergebnisses erfolgt genau wie bei Option 1.

1.3. Lernmodus

Wird beim Scannen des Busses ein Gerät gefunden, für das keine Gerätevorlage in der **DEVICEDB** existiert, gibt das FP Gateway die Daten wie gewohnt zurück, fügt aber am Ende der Rückgabe eine Liste der unbekannten Geräte ein. Die unbekannten Geräte werden zu diesem Zeitpunkt nicht automatisch konfiguriert.

2 Funktionsweise

1.4. Grundfunktion

Die automatische Konfiguration wird durch einen Aufruf des Kommandos <ScanDevices> mit dem Parameter auto="TRUE" gestartet.

Beispiel für einen Aufruf des Kommandos:

```
[<ScanDevices _="COM2" Timeout="2s" protocol="Modbus,RTU" type="Master"
  baud="9600" handshake="HALF" format="8N1" auto="TRUE" ver="v"/>]
```

Wird der Parameter "auto" weggelassen, oder auto="FALSE" angegeben, werden gefundene Geräte nur ausgegeben und die Konfiguration wird nicht verändert.

Im Folgenden wird der Endausbau beschrieben. Im 1. Implementierungsschritt wird davon nur die automatische Konfiguration der „External“ umgesetzt.

Während des Scans wird in der DEVICEDB nach einem passenden Gerät gesucht.
Schlüssel: eine oder mehrere Variablen mit einem oder mehreren Vergleichswerten

Automatisch konfiguriert werden folgende XML-Datenbanken:

- die External (Pollrate gerätespezifisch)
- die LogDefinition (1 Logfile und Records)
- die Event - Datenbank
- die Scheduler - Datenbank (ein Log-Intervall für das einzige Logfile)

Alle Zähler werden in eine einzige Logdatei eingetragen. In die External werden alle Variablen eingetragen, die in der DEVICEDB vordefiniert sind; es müssen also nicht alle verfügbaren Modbus Variablen des Gerätes verwendet werden, wenn diese nicht gebraucht werden.
Die Konfiguration folgt einem Schema. Geräte und Variablen haben alle den gleichen Bezeichner mit einem Nummernzusatz, der automatisch vergeben wird. Die Nummerierung erfolgt fortlaufend in der Reihenfolge, in der die Geräte gefunden werden.
Die Bezeichner für Events, Logfiles und Scheduler-Einträge sind fest in der DEVICEDB vorgegeben.

1.4.1. Aufbau der Gerätedatenbank **DEVICEDB**

Die DEVICEDB ist eine strukturierte XML-Datenbank. Sie beginnt immer mit dem Element Autoconf, es kann eine Versionsnummer vergeben werden. Es ist ratsam, bei jeder Änderung der Datenbank diese Versionsnummer zu erhöhen.

Dann folgen die Hauptsektionen "Statischer Konfigurationskopf" und die "Geraetevorlagen", die jeweils in weitere Untersektionen unterteilt sind. Achtung: Die weiter unten grün fett dargestellten Abschnitte sind optional.

Prinzipieller Aufbau der DEVICEDB:

```
[<SetConfig _="DEVICEDB" ver="v">
<AutoConf version="1">

      <!-- Beginn statischer Konfigurationskopf -->

          <!-- Busdefinition -->
          <Bus ..>
          </Bus>

          <!-- Logdefinition, Eventhandler, Scheduler -->
```

```
            <LogFiles>
            ..
            </LogFiles>
            <EventHandler>
            ..
            <EventHandler>

            <Schedule>
            ..
            </Schedule>

        <!-- Ende statischer Konfigurationskopf -->

        <!-- Beginn Geraetevorlagen -->

            <DeviceConfig>

                  <!-- Geraet 1 -->

                  <!-- Geraet 2 -->
                  ..

                  <!-- Geraet n -->

            </DeviceConfig>

        <!-- Ende Geraetevorlagen -->

</AutoConf>
</SetConfig>]
```

Bitte beachten:

Die Zeilenlänge (Anzahl Spalten) in der DEVICEDB-Datenbank ist auf maximal 260 Zeichen begrenzt !

1.4.1.1. Statischer Konfigurationskopf

Der statische Konfigurationskopf dient zur globalen Definition der Busparameter und des Datenloggings. Das Datenlogging wird unterteilt in `Logdefinition`, `Eventhandler` und `Scheduler`.

> **Bitte beachten Sie:**
> Die Einträge zum Datenlogging gelten für alle Modbus Zähler gemeinsam. Insbesondere die Logzyklen werden für alle Zähler global definiert.

Beispiel für den statischen Konfigurationskopf:

```
[<SetConfig _="DEVICEDB" ver="v">
<AutoConf version="1">
<Bus Name="Modbus" _="COM2" Timeout="1s" protocol="Modbus,RTU" type="Master"
     Mem="1200000" handshake="HALF" format="8N1" baud="9600">
</Bus>

<LogFiles>
     <TiXDatalogging_0 record="TiXDatalogging_0_Rec" size="1200000"
                       contenttype="binary"/>
</LogFiles>

<EventHandler>
       <TiXDatalogging_0_Log>
          <BinLog _="TiXDatalogging_0"/>
       </TiXDatalogging_0_Log>
</EventHandler>

<Schedule>
       <TiXDatalogging_0_Log _="TiXDatalogging_0_Log">
              <Minute _="0,5,10,15,20,25,30,35,40,45,50,55"/>
       </TiXDatalogging_0_Log>
</Schedule>
```

Nur die **fett** dargestellten Bereiche dürfen geändert werden.

Die Möglichkeit nicht automatisch generierte Einträge in LogDefinition, EventHandler und Schedule aus den bestehenden Datenbanken mit zu übernehmen wurde beibehalten. Dazu müssen die automatisch generierten Einträge aber den Präfix "**TiX**" erhalten.
Die Bus-Definition kann auch weitere Modbus-Attribute enthalten, z.B. `DWordSwap` usw.

Wichtig: Die Einträge in der `SCHEDULE`-Datenbank werden nur korrekt übernommen, wenn die Datenbank nach der TICO-Konvention erstellt wird, d.h. getrennte Dateien für "Conditions" und "Schedule" verwendet werden. Die TICO-Dateien heißen "`15-ScheduleConditions.txt`" und "`16-Schedule.txt`".

1.4.1.1.1 Busname

Es kann ein Name für den Bus vergeben werden:

```
<Bus Name="Modbus" _="COM2" Timeout="1s" protocol="Modbus,RTU" type="Master"
Mem="1200000"
  handshake="HALF" format="8N1" baud="9600">
</Bus>
```

Bitte beachten Sie, dass dieser Name in den Pfadangaben der Variablen verwendet wird und daher überall in der `DEVICDB` angepasst werden muß (im Beispiel `Modbus`).

Optional kann die Größe des Variablenspeichers `Mem="1200000"` angepasst werden.
Die maximal nutzbare Größe ist Mem="20000000" (20 Millionen Byte).

1.4.1.1.2 Größe der Logdateien (optional)

Die Logdateigröße lässt sich über den Parameter `size=".."` ändern:

```
  <TiXDatalogging_0 record=" Datalogging_0_Rec" size="1200000"
                    contenttype="binary"/>
```

1.4.1.1.3 Logdateiname (optional)

Der Name der Logdateien ist fest in der **DEVICEB** vorgegeben.
Im Beispiel heißt die Logdatei **TiXDatalogging_0**.

Die Logzyklen werden in der Sektion **Scheduler** definiert. Im Beispiel werden alle 5 Minuten Logeinträge generiert. Weitere Scheduler-Optionen finden Sie im TiXML-Referenzhandbuch, Kapitel 7 Scheduler.

1.4.1.2 Gerätevorlagen (device templates)

Die Gerätevorlagen werden jeweils mit **<Ident_nn> </Ident_nn>** geklammert. **nn** ist dabei eine fortlaufende Nummer (01 .. 99). Damit lassen sich einzelne Geräte nachträglich in die **DEVICEDB** eintragen, ohne die komplette Datenbank neu schreiben zu müssen. Hinter dem Ident-Eintrag wird ein **ModbusDeviceType** und die **ModbusDeviceVersion** definiert. Diese Werte sind statisch und werden bei der Ausgabe der Ergebnisse mitgeführt, um die Zähler eindeutig unterscheiden zu können.

Die Geräteeinträge bestehen aus einem Gerätekopf mit Informationen zur eindeutigen Bezeichnung des jeweiligen Gerätes, einer Detect-Sektion, in der die zu prüfenden Modbus-Register eines Gerätes definiert sind, gefolgt von den Variablendefinitionen und den Logeinträgen.

Der „**External**“-Bereich in der „**PROCCFG**“-Datenbank wird jeweils komplett neu geschrieben.
Bereits definierte Objekte werden nicht übernommen.
Bei den Datenbankbereichen „**LogDefinition**“, „**EventHandler**“ und „**Schedule**“ werden bereits definierte Objekte in die neu erzeugte Datenbankversion übernommen. Kennzeichen ist der Präfix „**TiX**“, der den automatisch generierten Objekten vorgestellt werden muss. Objekte ohne diesen Präfix werden aus der alten Datenbankversion kopiert.

1.4.1.2.1 Gerätekopf

```
<Ident_nn _="ModbusDeviceType" Version="ModbusDeviceVersion">
<Detect>
   <Scan_1 _="Typ" ind="Index" [size="Size"] value1="Value" [value2="Value"]
                  [value3="Value"] [value4="Value"] [value5="Value"]/>
   <Scan_2 _="Typ" ind="Index" [size="Size"] value1="Value" [value2="Value"]
                  [value3="Value"] [value4="Value"] [value5="Value"]/>
   <Scan_3 _="Typ" ind="Index" [size="Size] value1="Value" [value2="Value"]
                  [value3="Value"] [value4="Value"] [value5="Value"]/>
   ..
   <Scan_n _="Typ" ind="Index" [size="Size"] value1="Value" [value2="Value"]
                  [value3="Value"] [value4="Value"] [value5="Value"]/>
</Detect>

<Device _="XX" Name="Dev_XX" Pollrate="5s" Signifier1="Sig1" Signifier2="Sig2">
     …
</Device>
```

Nur die **fett** dargestellten Bereiche dürfen geändert werden.

```
  XX    = Modbus Stationsnummer (1 .. 247). Wird automatisch ersetzt.
  Typ   = [I,R,H,D,RI,HI,DI]
  Index = 0 .. 65535 (Modbus-Register)
  Size  = (optional, wird zur Zeit nicht unterstützt); die Anzahl der Modbus-
          Register wird implizit aus dem Typ ermittelt.
  Value = Wert oder Wert mit "Wildcards"; werden mehrere Values angegeben, gilt der
          Vergleich als WAHR, wenn mind. einer gültig ist (ODER)
          Es können wildcards angegeben werden. Diese Stellen werden nicht geprüft
```

Sind mehrere "Scan"-Zeilen definiert, werden diese von oben nach unten abgearbeitet. Wenn alle "Scan"-Zeilen zutreffen, gilt der Zähler als identifiziert.

Beispiel 1 (Zählertyp Carlo Gavazzi "EM24"):
Auf Index 11 liegt die Seriennummer von einem „word“ (16 bit) Länge. Die ersten beiden Stellen sollen den Wert 71, 72 oder 73 haben (deshalb werden hier "x" als wildcards für die restlichen Stellen verwendet).

```
  <Scan_1 _="H" ind="11" value1="71xxxxxx" value2="72xxxxxx" value3="73xxxxxx" />
```

Beispiel 2 (Zählertyp Janitza "UMG 96 RM (ohne Zusatzmodul)"):
Auf Index 754 liegt die Seriennummer von zwei „word“ Länge (32 bit). Wenn die ersten 2 Stellen in Dezimaldarstellung den Wert 17 haben, soll geprüft werden, ob das Register auf Index 761 den Wert 0 hat..

```
   <Scan_1 _="D" ind="754" value1="17xxxxxx"/>
   <Scan_2 _="H" ind="761" value1="0"/>
```

Die Angaben sind jeweils in Dezimaldarstellung zu machen. Dabei kennzeichnet „x” Stellen, die nicht überprüft werden.

Eingeleitet wird der Gerätekopf immer mit **<Ident_nn>**, wobei **nn** eine fortlaufende Zahl von 01 bis 99 ist.
Dann folgen der Gerätetyp und die Version des Gerätes.

`ModbusDeviceType`	Vom Nutzer vergebener Name des Modbus-Gerätes
`ModbusDeviceVersion`	Vom Nutzer vergebene Versionsnummer des Gerätes

Gerätetyp und Geräteversion müssen in ihrer Kombination eindeutig sein, z.B.

```
<Ident_01 _="UMG 96 RM" Version="1.0.3">
<Ident_02 _="UMG 96 RM" Version="1.0.43">
<Ident_03 _="EM24-DIN" Version="2.0.1">
<Ident_04 _="EM24-DIN" Version="4.2">
```

1.4.1.2.2 Variablendefinition

In der Sektion "Variablendefinition" sind alle Variablen aufgeführt, die bei einem automatischen Modbus Scan exakt wie hier definiert in die `External` übernommen werden. Die Variablen werden zeilenweise definiert und können alle in der `External` gültigen Tags enthalten. Die Variablendefinition wird mit dem Tag `</Device>` abgeschlossen.

Beispiel:

```
<Device _="XX" Name="Dev_XX" Pollrate="5s" Signifier1="Sig1" Signifier2="Sig2">
  <Dev_XX_FWVer  _="H" ind="1" simpleType="float" Name="FWVersion" acc="R" />
  <Dev_XX_ModType  _="C" ind="10" simpleType="Uint16" Name="Module Type" acc="R" />
  <Dev_XX_VoltL1  _="H" ind="20" simpleType="float" Name="Voltage L1" acc="R" />
</Device>

<Records>
    <TiXDatalogging_0_Rec>
      <Dev_XX_Sig1 _="Signifier" path="/Process/Modbus1/Dev_XX/DeviceSignifier1"
        size="60"/>
      <Dev_XX_Sig2 _="Signifier" path="/Process/Modbus1/Dev_XX/DeviceSignifier2"
        size="60"/>
      <Dev_XX_FWVer _="Uint16" path="/Process/Modbus1/Dev_XX_FWVer"/>
      <Dev_XX_ModType _="Bit" path="/Process/Modbus1/Dev_XX_ModType"/>
      <Dev_XX_VoltL1 _="float" path="/Process/Modbus1/Dev_XX_VoltL1"/>
    </TiXDatalogging_0_Rec>
</Records>
</Ident_04>
</SetConfig>]
```

Die **fett** dargestellten Bereiche dürfen vom Anwender geändert werden.

Der Gerätename (`Name="Dev_XX"`) kann frei definiert werden. Der Name sollte mit einem Buchstaben beginnen, keine Sonderzeichen (z.B. Umlaute etc.) enthalten und maximal 20 Zeichen lang sein.

Die Pollrate ist für jeden Gerätetyp einzeln spezifizierbar.
Die Tags "Signifier1" und "Signifier2" sind optional und enthalten einen statischen Text, der dann über eine Referenz in die LogDefinition übernommen werden kann. Max. Länge=59 Zeichen je Signifier. Die Signifier werden verwendet, um z.B. die Messstellen-ID zu speichern.

Die XML-Variablennamen bestehen aus einem Präfix Dev_XX, wobei XX in diesem Fall die Stationsnummer ist und "Dev" frei vom Anwender in der DEVICEDB vergeben werden kann. Entsprechend der möglichen Stationsaddressen im Modbus-Protokoll liegt XX im Bereich von 1...247.

Enthält der Dateiname für die Logdefinition (Records) den Text "_XX_", dann wird für jedes Modbus-Gerät eine separate Logdatei erstellt. Ansonsten werden alle Records in eine globale Logdatei geschrieben.

Ein gezieltes Einfügen neuer Einträge an einer bestimmten Stelle einer Gruppe ist nicht möglich. In der Regel werden neue Sektionen am Anfang der adressierten Gruppe eingefügt. Im o. a. Beispiel steht somit Ident_04 vor Ident_01 in der Gruppe DeviceConfig der DEVICEDB. Auf die Suchalgorithmen hat das keinen negativen Einfluss.

Die Möglichkeit nicht automatisch generierte Einträge in LogDefinition, EventHandler und Schedule aus den bestehenden Datenbanken mit zu übernehmen wird beibehalten. Dazu müssen die automatisch generierten Einträge aber den Präfix "TiX" erhalten.

1.4.1.3 Protokollierung der automatischen Konfiguration im SupportLog

Die Ergebnisse der automatischen Konfiguration werden im Logfile "SupportLog" protokolliert. Der Nutzer ist selbst dafür verantwortlich, eine SupportLog-Datei in der LogDefinition anzulegen.
Beispiel für einen SupportLog-Eintrag:

```
<ID_29 _="2020/11/10,16:05:09">
<ModbusScan>
  <ScanStart _="16:05:09, COM2, baud=9600, Start=1, End=247, auto=TRUE" />
  <ScanStop _="16:07:39, Known: 5, Unknown: 0 devices" />
  <StartOfConfiguration _="16:07:39" />
  <EndOfConfiguration _="16:07:50" />
</ModbusScan>
</ID_29>
```

Hinweis: <StartOfConfiguration/> … <EndOfConfiguration/> entfällt in der 1. Stufe (nur External wird konfiguriert).

Protokolliert werden die Startzeit, Einstellungen der Schnittstelle, die Stopzeit (das ist das Ende des Modbus Scans) sowie die Anzahl der bekannten und unbekannten Geräte.
Weiterhin wird protokolliert, wann die Erstellung der internen Konfigurationsdatenbanken gestartet wurde (StartOfConfiguration) und beendet wurde (EndOfConfiguration).

Eventuell aufgetretene Fehler werden mit einem Fehlercode angezeigt. Mögliche Fehlercodes sind:

- -109: Allgemeiner Datenbankfehler; meist ein Hinweis auf fehlerhafte Syntax in der DEVICEDB
- -104: Speichermangel: zu wenig Speicher für den Modbus in der External reserviert
- -117: Speichermangel; zu wenig Speicher für den Modbus in der External reserviert
- -214: Speichermangel; zu wenig Speicher für den Modbus in der LOGDefinition reserviert

Wird bei einem Scan mit auto=TRUE erkannt, dass die Busdefinition fehlt oder fehlerhaft ist, wird der Scan mit auto="FALSE" durchgeführt und im Supportlog „<WrongBusDefinition _="CheckDEVICEDB"/> eingetragen.

Das Kommando ScanDevices kann auch in einem EventHandler verwendet werden. Auch in diesem Fall werden die oben beschriebenen Einträge im SupportLog erstellt.

1.4.1.4 Praktische Hinweise

Zur Beschleunigung der automatischen Konfiguration bei sehr vielen Zählern mit sehr vielen Variablen (ab einer Gesamtsumme von ca. 5000 Variablen für alle Modbus-Zähler) kann es hilfreich sein, folgende Optionen beim Einspielen der Grundkonfiguration zu verwenden:

1. Nutzung des Parameter OmitInvalidateVars

Es wird empfohlen, eine leere `External` mit einem vorkonfigurierten Modbus und dem Parameter `<OmitInvalidateVars _="On"/>` in das Gerät einzuspielen:

```
[<SetConfig _="PROCCFG" ver="y">
<OmitInvalidateVars _="On"/>
<External>
  <Bus Name="Modbus" _="COM2" protocol="Modbus,RTU" baud="19200" Timeout="1s"
       Mem="10000000" handshake="HALF" type="Master" AddProperties="Name,TimeStamp"
       format="8N1">
</Bus>
</External>
</SetConfig>]
```

2. Gerät neu starten (aktiviert den Modbus)

```
 [<Reset _="Keep" ver="y">]
```

3. Befehl ScanDevices ausführen

```
[<ScanDevices _="COM2" Timeout="1s" protocol="Modbus,RTU" type="Master"
              handshake="HALF" baud="19200" auto="TRUE" ver="v"/>]
```

an das Gerät schicken.

4. Bus läuft anschließend an, Gerät ist betriebsbereit.

Je nach Anzahl der gefundenen Zähler und deren Adreß-Verteilung (Stationsadresse) kann es eine Weile dauern, bis die Erkennung und Konfiguration der Geräte abgeschlossen ist.
Während der Konfiguration leuchtet die Process-LED dauerhaft.

1.5 Lernmodus

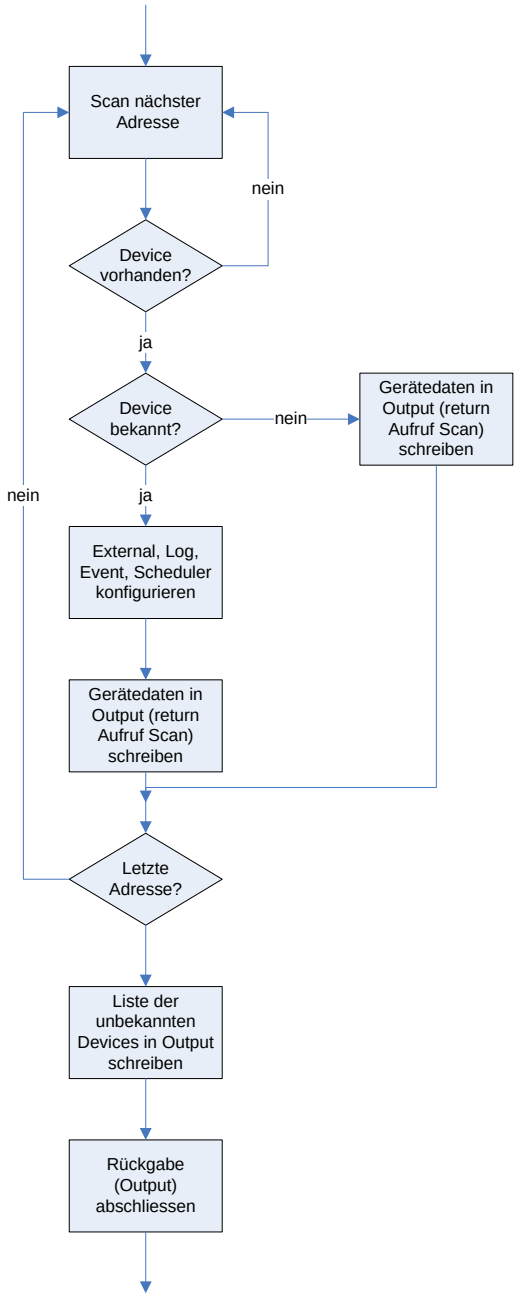

Bild 1: Ablaufdiagramm der Firmware

Die während des Scans gefundenen Geräte werden alle ausgegeben.

1.6 Beispiele

Beispiel 1:

Ergebnis eines ScanDevices-Befehls

```
<ScanDevices>
<Device _="0" Type="UMG 96 RM" Version="2.5.0.4">
  <Var_001 _="C" ind="0x200" simpleType="float" value="0"/>
  <Var_002 _="C" ind="0x201" simpleType="float" value="1"/>
  <Var_003 _="I" ind="0x210" simpleType="float" value="10"/>
  <Var_004 _="R" ind="0x212" simpleType="float" value="50"/>
  <Var_005 _="H" ind="0x220" simpleType="float" value="5500"/>
  <Var_006 _="D" ind="0x230" simpleType="float" value="555500"/>
</Device>

<Device _="2" Type="EM24-DIN " Version="1.2.0">
  <Var_001 _="I" ind="0x2140" simpleType="float" exp="-1" value="20"/>
  <Var_002 _="H" ind="0x3201" simpleType="float" exp="-1" value="145"/>
  <Var_003 _="H" ind="0x4210" simpleType="float" exp="-1" value="1310"/>
  <Var_004 _="R" ind="0x4216" simpleType="float" exp="-1" value="50"/>
  <Var_005 _="H" ind="0x5220" simpleType="float" exp="-1" value="4560"/>
  <Var_006 _="D" ind="0x5230" simpleType="float" exp="-1" value="185500"/>
</Device>

<UnknownDevices>
  <StationID _="100"/>
  <StationID _="120"/>
  <StationID _="121"/>
</UnknownDevices >

</ScanDevices>
```

Für die in der DEVICEDB definierten Geräte werden die Datenbanken External, LOG, EVENTS und SCHEDULE neu konfiguriert (wenn angefordert).
Für die nicht in der DEVICEDB gefundenen Geräte wird zum Ende des Scans die Sektion <UnknownDevices> ausgegeben:

```
<UnknownDevices>
  <StationID _="100"/>
  <StationID _="120"/>
  <StationID _="121"/>
</UnknownDevices >
```

ausgegeben (Beispiel). Konfiguriert werden diese Geräte nicht automatisch.

Zu beachten ist, daß nicht alle Modbus-Geräte am Bus erkennbar sind, weil nicht alle Geräte auf eine allgemeine Busanfrage antworten (z.B. UMG 96 von Janitza). Daher kann die Liste der nicht erkannten Geräte in der Sektion <UnknownDevices> evtl. unvollständig sein.

Beispiel 2: Vollständige DEVICEDB für Janitza UMG96RM und CarloGavazzi EM24-DIN:

```
[<SetConfig _="DEVICEDB" ver="v">

<AutoConf version="1">

<Bus Name="Modbus" _="COM2" Timeout="1s" protocol="Modbus,RTU" type="Master" handshake="HALF" format="8N1" baud="9600"
        AddProperties="Name,TimeStamp">
</Bus>

<DeviceConfig>

<Ident_01 _="EM24-DIN AV2_AV9" Version="1.0.2">
 <Detect>
  <Scan_1 _="H" ind="11" val1="71"/>
 </Detect>

 <Device _="XX" NameUser="EM24-XX" Pollrate="1s" CharTimeout="50ms" Timeout="300ms" Pause="50ms" DWordInc="2"
        DwordSwap="0" ForceSingleWordWrite="1" UseCache="1">
  <SerialNo Name="EM24-XX-SerialNo" _="H" simpleType="Uint16" ind="11" acc="R"/>
  <Voltage_L1_N Name="EM24-XX-Voltage L1-N" _="D" simpleType="Uint32" ind="0" acc="R" exp="-1" format="F.1; V"/>
  <Voltage_L2_N Name="EM24-XX-Voltage L2-N" _="D" simpleType="Uint32" ind="2" acc="R" exp="-1" format="F.1; V"/>
  <Voltage_L3_N Name="EM24-XX-Voltage L3-N" _="D" simpleType="Uint32" ind="4" acc="R" exp="-1" format="F.1; V"/>
  <RealEnergy_L1L2L3Con Name="EM24-XX-Real energy L1..L3, consumed" _="D" simpleType="Uint32" ind="62" acc="R" exp="2"
        format="F.1; Wh"/>
  <Voltage_L1_L2 Name="EM24-XX-Voltage L1-L2" _="D" simpleType="Uint32" ind="6" acc="R" exp="-1" format="F.1; V"/>
  <Voltage_L2_L3 Name="EM24-XX-Voltage L2-L3" _="D" simpleType="Uint32" ind="8" acc="R" exp="-1" format="F.1; V"/>
  <Voltage_L3_L1 Name="EM24-XX-Voltage L3-L1" _="D" simpleType="Uint32" ind="10" acc="R" exp="-1" format="F.1; V"/>
  <ApparentCurrentL2 Name="EM24-XX-Apparent current, L2" _="D" simpleType="Uint32" ind="14" acc="R" exp="-3" format="F.2; A"/>
  <ApparentCurrentL3 Name="EM24-XX-Apparent current, L3" _="D" simpleType="Uint32" ind="16" acc="R" exp="-3" format="F.2; A"/>
  <RealPower_L1_N Name="EM24-XX-Real power L1-N" _="D" simpleType="Uint32" ind="18" acc="R" exp="-1" format="F.1; W"/>
  <RealPower_L2_N Name="EM24-XX-Real power L2-N" _="D" simpleType="Uint32" ind="20" acc="R" exp="-1" format="F.1; W"/>
  <RealPower_L3_N Name="EM24-XX-Real power L3-N" _="D" simpleType="Uint32" ind="22" acc="R" exp="-1" format="F.1; W"/>
  <ReactiveEnergy_L1L3c Name="EM24-XX-Reactive energy L1..L3, inductive" _="D" simpleType="Uint32" ind="64" acc="R"
        format="F.1; varh"/>
  <ApparentPower_L1_N Name="EM24-XX-Apparent power L1-N" _="D" simpleType="Uint32" ind="24" acc="R" exp="-1"
        format="F.2; VA"/>
  <ApparentPower_L2_N Name="EM24-XX-Apparent power L2-N" _="D" simpleType="Uint32" ind="26" acc="R" exp="-1"
        format="F.2; VA"/>
  <ApparentPower_L3_N Name="EM24-XX-Apparent power L3-N" _="D" simpleType="Uint32" ind="28" acc="R" exp="-1"
        format="F.2; VA"/>
  <ReactivePower_L1 Name="EM24-XX-Reactive power L1" _="H" simpleType="Uint16" ind="30" acc="R" exp="-1" format="F.1; var"/>
  <ReactivePower_L2 Name="EM24-XX-Reactive power L2" _="H" simpleType="Uint16" ind="32" acc="R" exp="-1" format="F.1; var"/>
  <Frequency Name="EM24-XX-Measured Frequency" _="H" simpleType="Uint16" ind="55" exp="-1" acc="R" format="F.1; Hz"/>
  <ReactivePower_L3 Name="EM24-XX-Reactive power L3" _="H" simpleType="Uint16" ind="34" acc="R" exp="-1" format="F.1; var"/>
  <RealPowerPSum_P1P2P3 Name="EM24-XX-Real power sum; Psum=P1+P2+P3" _="D" simpleType="Uint32" ind="40" acc="R"
        exp="1" format="F.1; W"/>
  <ApparentPwSSm_P1P2P3 Name="EM24-XX-Apparent power; Ssum=S1+S2+S3" _="D" simpleType="Uint32" ind="42" acc="R"
        exp="-1" format="F.1; VA"/>
  <ReactivePwrQs1_2_3 Name="EM24-XX-Reactive power; Qsum=Q1+Q2+Q3" _="H" simpleType="Uint16" ind="44" acc="R"
        exp="-1" format="F.1; var"/>
  <CosPhi_UL1_IL1 Name="EM24-XX-CosPhi; UL1 IL1" _="H" simpleType="Uint16" ind="50" acc="R" exp="-3" format="F.2; "/>
  <CosPhi_UL2_IL2 Name="EM24-XX-CosPhi; UL2 IL2" _="H" simpleType="Uint16" ind="51" acc="R" exp="-3" format="F.2; "/>
  <CosPhi_UL3_IL3 Name="EM24-XX-CosPhi; UL3 IL3" _="H" simpleType="Uint16" ind="52" acc="R" exp="-3" format="F.2; "/>
  <RealEnergy_L1L2L3Del Name="EM24-XX-Real energy L1..L3, delivered" _="H" simpleType="Uint16" ind="92" acc="R" exp="2"
        format="F.1; Wh"/>
  <ReactiveEnergy_L1L3c Name="EM24-XX-Reactive energy L1..L3, capacitive" _="D" simpleType="Uint32" ind="94" acc="R"
        format="F.1; varh"/>
 </Device>
</Ident_01>

<Ident_02 _="UMG96RM" Version="2.0.5.4">
 <Detect>
  <Scan_1 _="D" ind="754" val1="17"/>
  <Scan_2 _="H" ind="761" size="1" val1="0"/>
 </Detect>
```

```
  <Device _="XX" NameUser="UMG96RM-XX" Pollrate="1s" CharTimeout="50ms" Timeout="300ms" Pause="50ms" DWordInc="2"
        DwordSwap="1" ForceSingleWordWrite="1" UseCache="1">
   <SerialNo Name="UMG96RM-XX-SerialNo" _="D" simpleType="Uint32" ind="754" acc="R"/>
   <Module Name="UMG96RM-XX-Module" _="H" simpleType="Uint16" ind="761" acc="R"/>
   <Voltage_L1_N Name="UMG96RM-XX-Voltage L1-N" _="D" simpleType="float" ind="19000" acc="R" format="F.1; V"/>
   <Voltage_L2_N Name="UMG96RM-XX-Voltage L2-N" _="D" simpleType="float" ind="19002" acc="R" format="F.1; V"/>
   <Voltage_L3_N Name="UMG96RM-XX-Voltage L3-N" _="D" simpleType="float" ind="19004" acc="R" format="F.1; V"/>
   <Voltage_L1_L2 Name="UMG96RM-XX-Voltage L1-L2" _="D" simpleType="float" ind="19006" acc="R" format="F.1; V"/>
   <Voltage_L2_L3 Name="UMG96RM-XX-Voltage L2-L3" _="D" simpleType="float" ind="19008" acc="R" format="F.1; V"/>
   <Voltage_L3_L1 Name="UMG96RM-XX-Voltage L3-L1" _="D" simpleType="float" ind="19010" acc="R" format="F.1; V"/>
   <ApparentCurrentL1 Name="UMG96RM-XX-Apparent current, L1" _="D" simpleType="float" ind="19012" acc="R" format="F.2; A"/>
   <ApparentCurrentL2 Name="UMG96RM-XX-Apparent current, L2" _="D" simpleType="float" ind="19014" acc="R" format="F.2; A"/>
   <ApparentCurrentL3 Name="UMG96RM-XX-Apparent current, L3" _="D" simpleType="float" ind="19016" acc="R" format="F.2; A"/>
   <VectorSum_IN_I1I2I3 Name="UMG96RM-XX-Vector sum; IN=I1+I2+I3" _="D" simpleType="float" ind="19018" acc="R"
        format="F.2; A"/>
   <RealPower_L1_N Name="UMG96RM-XX-Real power L1-N" _="D" simpleType="float" ind="19020" acc="R" format="F.1; W"/>
   <RealPower_L2_N Name="UMG96RM-XX-Real power L2-N" _="D" simpleType="float" ind="19022" acc="R" format="F.1; W"/>
   <RealPower_L3_N Name="UMG96RM-XX-Real power L3-N" _="D" simpleType="float" ind="19024" acc="R" format="F.1; W"/>
   <RealPowerPSum_P1P2P3 Name="UMG96RM-XX-Real power sum; Psum=P1+P2+P3" _="D" simpleType="float" ind="19026"
acc="R"
        format="F.1; W"/>
   <ApparentPower_L1_N Name="UMG96RM-XX-Apparent power L1-N" _="D" simpleType="float" ind="19028" acc="R"
        format="F.2; VA"/>
   <ApparentPower_L2_N Name="UMG96RM-XX-Apparent power L2-N" _="D" simpleType="float" ind="19030" acc="R"
        format="F.2; VA"/>
   <ApparentPower_L3_N Name="UMG96RM-XX-Apparent power L3-N" _="D" simpleType="float" ind="19032" acc="R"
        format="F.2; VA"/>
   <ApparentPwSSm_P1P2P3 Name="UMG96RM-XX-Apparent power; Ssum=S1+S2+S3" _="D" simpleType="float" ind="19034" acc="R"
        format="F.1; VA"/>
   <ReactivePower_L1 Name="UMG96RM-XX-Reactive power L1" _="D" simpleType="float" ind="19036" acc="R" format="F.1; var"/>
   <ReactivePower_L2 Name="UMG96RM-XX-Reactive power L2" _="D" simpleType="float" ind="19038" acc="R" format="F.1; var"/>
   <ReactivePower_L3 Name="UMG96RM-XX-Reactive power L3" _="D" simpleType="float" ind="19040" acc="R" format="F.1; var"/>
   <ReactivePwrQs1_2_3 Name="UMG96RM-XX-Reactive power; Qsum=Q1+Q2+Q3" _="D" simpleType="float" ind="19042" acc="R"
        format="F.0; var"/>
   <CosPhi_UL1_IL1 Name="UMG96RM-XX-CosPhi; UL1 IL1" _="D" simpleType="float" ind="19044" acc="R" format="F.2; "/>
   <CosPhi_UL2_IL2 Name="UMG96RM-XX-CosPhi; UL2 IL2" _="D" simpleType="float" ind="19046" acc="R" format="F.2; "/>
   <CosPhi_UL3_IL3 Name="UMG96RM-XX-CosPhi; UL3 IL3" _="D" simpleType="float" ind="19048" acc="R" format="F.2; "/>
   <Frequency Name="UMG96RM-XX-Measured Frequency" _="D" simpleType="float" ind="19050" acc="R" format="F.1; Hz"/>
   <RealEnergy_L1L2L3 Name="UMG96RM-XX-Real energy L1..L3" _="D" simpleType="float" ind="19060" acc="R" format="F.1; Wh"/>
   <RealEnergy_L1L2L3Con Name="UMG96RM-XX-Real energy L1..L3, consumed" _="D" simpleType="float" ind="19068" acc="R"
        format="F.1; Wh"/>
   <RealEnergy_L1L2L3Del Name="UMG96RM-XX-Real energy L1..L3, delivered" _="D" simpleType="float" ind="19076" acc="R"
        format="F.1; Wh"/>
   <ApparentEnergy_L1L3 Name="UMG96RM-XX-Apparent energy L1..L3" _="D" simpleType="float" ind="19084" acc="R"
        format="F.1; Vah"/>
   <ReactiveEnergy_L1L3 Name="UMG96RM-XX-Reactive energy L1..L3" _="D" simpleType="float" ind="19092" acc="R"
        format="F.1; varh"/>
   <ReactiveEnergy_L1L3i Name="UMG96RM-XX-Reactive energy L1..L3, inductive" _="D" simpleType="float" ind="19100" acc="R"
        format="F.1; varh"/>
   <ReactiveEnergy_L1L3c Name="UMG96RM-XX-Reactive energy L1..L3, capacitive" _="D" simpleType="float" ind="19108" acc="R"
        format="F.1; varh"/>
   <HarmonicTHD_UL1_N Name="UMG96RM-XX-Harmonic, THD, U L1-N" _="D" simpleType="float" ind="19110" acc="R"
        format="F.1; %"/>
   <HarmonicTHD_UL2_N Name="UMG96RM-XX-Harmonic, THD, U L2-N" _="D" simpleType="float" ind="19112" acc="R"
        format="F.1; %"/>
   <HarmonicTHD_UL3_N Name="UMG96RM-XX-Harmonic, THD, U L3-N" _="D" simpleType="float" ind="19114" acc="R"
        format="F.1; %"/>
   <HarmonicTHD_IL1_N Name="UMG96RM-XX-Harmonic, THD, I L1-N" _="D" simpleType="float" ind="19116" acc="R"
        format="F.1; %"/>
   <HarmonicTHD_IL2_N Name="UMG96RM-XX-Harmonic, THD, I L2-N" _="D" simpleType="float" ind="19118" acc="R"
        format="F.1; %"/>
   <HarmonicTHD_IL3_N Name="UMG96RM-XX-Harmonic, THD, I L3-N" _="D" simpleType="float" ind="19120" acc="R"
format="F.1; %"/>
  </Device>
</Ident_02>

</DeviceConfig>
</AutoConf>
</SetConfig>]
```

3 Historie

Version	Datum	Bearbeiter	Änderungen
1.0.1	23.02.2021	IVH	Redaktionelle Bearbeitung im Kontext der ersten englischen Ausgabe
1.0.0	10.11.2020	SH	Erstausgabe