Modbus RTU Auto Configuration for FP Gateways Quick Guide

Version: 1.0.1

www.inovolabs.com

Publication date: 23/02/2021

Table of contents

1 Introduction

FP gateways as of firmware version 5.1.7.0, support a Modbus RTU scan algorithm.
The Modbus RTU scan works in a similar way to the M-bus scan. The following properties were implemented for maximum flexibility:

a) Two scan modes
1. Only display the scan result
2. Automatic configuration of External / LOG / EVENTS / SCHEDULE

An optional start address and end address can be specified for the scan (Modbus station number)

b) Variable search criteria
The scan is performed with different numbers of search criteria (data points) depending on the device
* When comparing with the search criteria, "Wildcards" should also be allowed.
When scanning the Modbus devices, you read a register in which the serial number is stored for example.
However, only the first 2 characters of the serial number are to be compared for example.
* It should also be possible to specify several values for the comparison for a register (OR link).

c) Selective automatic configuration
Automatic configuration should be selective, i.e. you should be able to specify for each database (External / LOG / EVENTS / SCHEDULE) that, for example, only the external database is written but not the LOG database and no SCHEDULE database

d) Logging with two modes
1. Logging in a global file
2. Logging in individual files per device

e) Displaying the scan activities
The start, end and result of the scan should be displayed using variables in the process branch:
/Process/ScanActive
/Process/LastScanFound

The "DEVICEDB" XML device database contains a list of known device types (device templates) and can be adjusted at any time. One or several Modbus registers are queried per device type and compared with one or several comparison values in order to identify the devices clearly.
In the database, you can specify which Modbus data points are to be read at all and which data points are to be logged.

1.1. Option 1: Only display the scan results

The simplest mode for the Modbus scan is displaying the scan results.

To do so, call the command for scanning the Modbus as follows:

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

The Modbus is scanned and a list of devices found output as the result:

```
<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>
```

The list of devices found is a copy of the device's variable list in the **DEVICEDB**.
="0" is the station ID on the Modbus (i.e. 0 here), **Type** and **Version** are copied from the **ModbusDeviceType** and **ModbusDeviceVersion** <Ident**xx>** entries (see section 1.4.1.2).

Devices that are not recognised are displayed in the "**UnknownDevices**" section along with the corresponding station address. **value** displays the register's value at the time of the scan.
StartAddr / EndAddr = Start address / end address (=Modbus station number) for the scan, optional

1.2. Option 2: Automatic configuration

As soon as the **DEVICEDB** device database is present in the device, completely automatic configuration can be initiated by calling the **<ScanDevices>** command. All internal configuration databases such as **External**, **LogDefinition**, **EventHandler** and **Scheduler** are generated automatically as required. The scan result is output in exactly the same was as option 1.

1.3. Learning mode

When scanning the bus, if a device is found for which no device template exists in the **DEVICEDB**, the FP gateway returns the data as normal but inserts a list of unknown devices at the end of the output. The unknown devices are not configured automatically at this time.

2 Mode of operation

1.4. Basic function

Automatic configuration is started by calling the <ScanDevices> command with the auto="TRUE" parameter.

Example for calling the command:

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

If the "auto" parameter is omitted or auto="FALSE" specified, the devices found are only output and the configuration is not changed.

The following describes the final version. In the first implementation step, only the automatic configuration for the "External" is implemented.

During the scan, the system searches the DEVICEDB for a suitable device.
Key: one or several variables with one or several comparison values

The following XML databases are configured automatically:

- the External (device-specific poll rate)
- the LogDefinition (1 log file and records)
- the event database
- the scheduler database (one log interval for the individual log file)

All meters are entered into one single log file. All variables that are predefined in the DEVICEDB are entered into the External; therefore, not all available Modbus variables for the device have to be used if they are not required.
The configuration follows a pattern. Devices and variables all have the same designator with a numeric suffix that is assigned automatically. Numbering is sequential in the order in which the devices are found.
The designators for events, log files and scheduler entries are specified in the DEVICEDB.

1.4.1. Layout of the **DEVICEDB** device database

The DEVICEDB is a structured XML database. It always starts with the Autoconf element and a version number can be assigned. We recommend increasing this version number each time the database changes.

The "Static configuration head" and the "Device templates" main sections then follow. These are divided into further subsections. Attention: The sections shown below in green and bold are optional.

Basic layout of the DEVICEDB:

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

      <!-- Start of static configuration head -->

          <!-- Bus definition -->
          <Bus ..>
          </Bus>

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

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

            <Schedule>
            ..
            </Schedule>

        <!-- End of static configuration head -->

        <!-- Start of device templates -->

            <DeviceConfig>

                  <!-- Device 1 -->

                  <!-- Device 2 -->
                  ..

                  <!-- Device n -->

            </DeviceConfig>

        <!-- End of device templates -->

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

Please note:

The row length (number of columns) in the DEVICEDB database is limited to a maximum of 260 characters!

1.4.1.1. Static configuration head

The static configuration head is used for global definitions of the bus parameters and data logging. Data logging is divided into `Logdefinition`, `Eventhandler` and `Scheduler`.

> **Please note:**
> The data logging entries apply to all Modbus meters. The log cycles in particular are defined globally for all meters.

Example for the static configuration head:

```
[<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>
```

Only the areas shown in **bold** are permitted to be changed.

The option to fill entries that were not generated automatically in LogDefinition, EventHandler and Schedule from the existing databases was retained. `However, to` do this, the automatically generated entries must receive the "`TiX`" prefix.
The bus definition can also contain further Modbus attributes, e.g. `DWordSwap` etc.

Important: The entries in the `SCHEDULE` database are only transferred correctly if the database is created according to the TICO convention, i.e. separate files are used for "Conditions" and "Schedule". The TICO files are called "`15-ScheduleConditions.txt`" and "`16-Schedule.txt`".

1.4.1.1.1 Bus name

A name can be assigned to the bus:

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

Please note that this name is used in the path specifications for the variables and must therefore be adjusted everywhere in the `DEVICDB` (in the example, `Modbus`).

Optional, the size of the variable memory `Mem="1200000"` must be adjusted.
The maximum usable size is Mem="20000000" (20 million bytes).

1.4.1.1.2 Size of the log files (optional)

The log file size can be changed using the `size=".."` parameter:

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

1.4.1.1.3 Log file name (optional)

The log file name is specified in the DEVICEB.
In the example, the log file is called **TiXDatalogging_0**.

The log cycles are defined in the Scheduler section. In the example, log entries are generated every 5 minutes. Further scheduler options are listed in the TiXML Reference Guide, section 7 Scheduler.

1.4.1.2 Device templates

The device templates are each surrounded with brackets <Ident_**nn**> </Ident_**nn**>. **nn** is a sequence number (01 - 99). This can be used to enter the devices retrospectively in the DEVICEDB without having to re-write the entire database. A **ModbusDeviceType** and the **ModbusDeviceVersion** are defined behind the Ident entry. These values are static and are included when outputting the results in order to be able to differentiate between the meters clearly.

The device entries comprise a device header with information providing a clear designation of the corresponding device, a detect section in which a device's Modbus registers that are to be checked are defined, followed by the variable definitions and the log entries.

The "External" area in the "PROCCFG" database is rewritten completely.
Objects that are already defined are not transferred.
In the "LogDefinition", "EventHandler" and "Schedule" database areas, objects that are already defined are transferred to the newly-generated database version. The indication is the "**TiX**" prefix that must be in front of the automatically generated objects. Objects without this prefix are copied from the old database.

1.4.1.2.1 Device head

```
<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>
```

Only the areas shown in **bold** are permitted to be changed.

```
  XX    = Modbus station number (1 - 247). Replaced automatically.
  Type  = [I,R,H,D,RI,HI,DI]
  Index = 0 - 65535 (Modbus register)
  Size  = (optional, not currently supported); the number of Modbus registers is
   determined implicitly from the type.
  Value = value or value with "Wildcards"; if several values are specified, the
          comparison is TRUE if at least one is valid (OR)
         Wildcards can be specified. These positions are not checked
```

If several "Scan" rows are defined, these are worked through from top to bottom. If all "Scan" rows apply, the meter is considered identified.

Example 1 (meter type Carlo Gavazzi "EM24"):
Index 11 contains the serial number of one “word” (16 bit) length. The first two positions should have the value 71, 72 or 73 (therefore, "x" is used as a wildcard for the remaining positions).

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

Example 2 (meter type Janitza "UMG 96 RM (without supplementary module)"):
Index 754 contains the serial number of two “words” (32 bit) length. If the first 2 positions in the decimal representation have the value 17, the system should check whether the register on index 761 has the value 0.

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

The entries must be made in the decimal representation. Here, “x” indicates positions that are not to be checked.

The device head is always initiated with **<Ident_nn>**, where **nn** is a sequence number from 01 to 99. The device type and the device version then follow.

ModbusDeviceType	Modbus device name assigned by the user
ModbusDeviceVersion	Device version number assigned by the user

The combination of device type and device version must be unique, e.g.

```
<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 Variable definition

The "Variable definition" section lists all variables that are transferred during an automatic Modbus scan exactly as defined here in the External. The variables are defined line by line and can contain valid tags in the External. The variable definition is ended with the </Device> tag.

Example:

```
<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>]
```

The areas shown in **bold** are permitted to be changed by the user.

The device name (Name="Dev_XX") can be defined freely. The name should start with a letter, not contain any special characters (e.g. umlauts, etc.) and not be longer than 20 characters.

The poll rate can be specified individually for each device type.
The "Signifier1" and "Signifier2" tags are optional and contain a static text that can be transferred to the LogDefinition via a reference. Max. length=59 characters per signifier.
The signifiers are used to save the measuring point ID.

The XML variable names comprise the Dev_XX prefix, where XX is the station number in this case and "Dev" can be assigned freely by the user in DEVICEDB. According to the possible station addresses in the Modbus protocol, XX is in the range of 1-247.

If the file name for the log definition (records) contains the text "_XX_", a separate log file is created for each Modbus device. Otherwise, all records are written into a global log file.

Targeted insertion of new entries at a specific point in a group is not possible. New sections are generally inserted at the start of the addressed group. In the aforementioned example, Ident_04 is therefore before Ident_01 in the DeviceConfig group of the DEVICEDB. This has no negative influence on the search algorithms.

The option to fill entries that were not generated automatically in LogDefinition, EventHandler and Schedule from the existing databases is retained. However, to do this, the automatically generated entries must receive the "TiX" prefix.

1.4.1.3 Logging the automatic configuration in the SupportLog

The results of the automatic configuration are logged in the "SupportLog" log file. The user him/herself is responsible for creating a SupportLog file in the LogDefinition.
Example for a SupportLog entry:

```
<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>
```

Note: <StartOfConfiguration/> … <EndOfConfiguration/> is omitted in the first step (only External is configured).

The start time, interface settings, the stop time (the end of the Modbus can) and the number of known and unknown devices are logged.
Furthermore, when creation of the internal configuration databases was started (StartOfConfiguration) and completed (EndOfConfiguration) is also logged.

Any errors that occur are displayed with an error code. Possible error codes are:

- -109: General database error; normally an indication of incorrect syntax in the DEVICEDB
- -104: Lack of memory: insufficient memory reserved for the Modbus in the External
- -117: Lack of memory: insufficient memory reserved for the Modbus in the External
- -214: Lack of memory: insufficient memory reserved for the Modbus in the LOGDefinition

During a scan with auto=TRUE, if the system detects that the bus definition is missing or incorrect, the scan is performed with auto="FALSE" and "<WrongBusDefinition _="CheckDEVICEDB"/> is entered in the support log.

The ScanDevices command can also be used in an EventHandler. In this case, the aforementioned entries are also created in the SupportLog.

1.4.1.4 Practical notes

In order to accelerate automatic configuration if there are many meters with many variables (as of a total of around 5000 variables for all Modbus meters), it can be helpful to use the following options when implementing the basic configuration:

1. Use the OmitInvalidateVars parameter

We recommend importing an empty `External` with a preconfigured Modbus and the `<OmitInvalidateVars _="On"/>` parameter into the device:

```
[<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. Restart the device (activates the Modbus)

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

3. Execute the ScanDevices command

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

Send to the device.

4. The bus starts subsequently, the device is ready for operation.

Depending on the number of meters found and their address distribution (station address), it may take a while until device detection and configuration is complete.
The process LED is illuminated constantly during configuration.

1.5 Learning mode

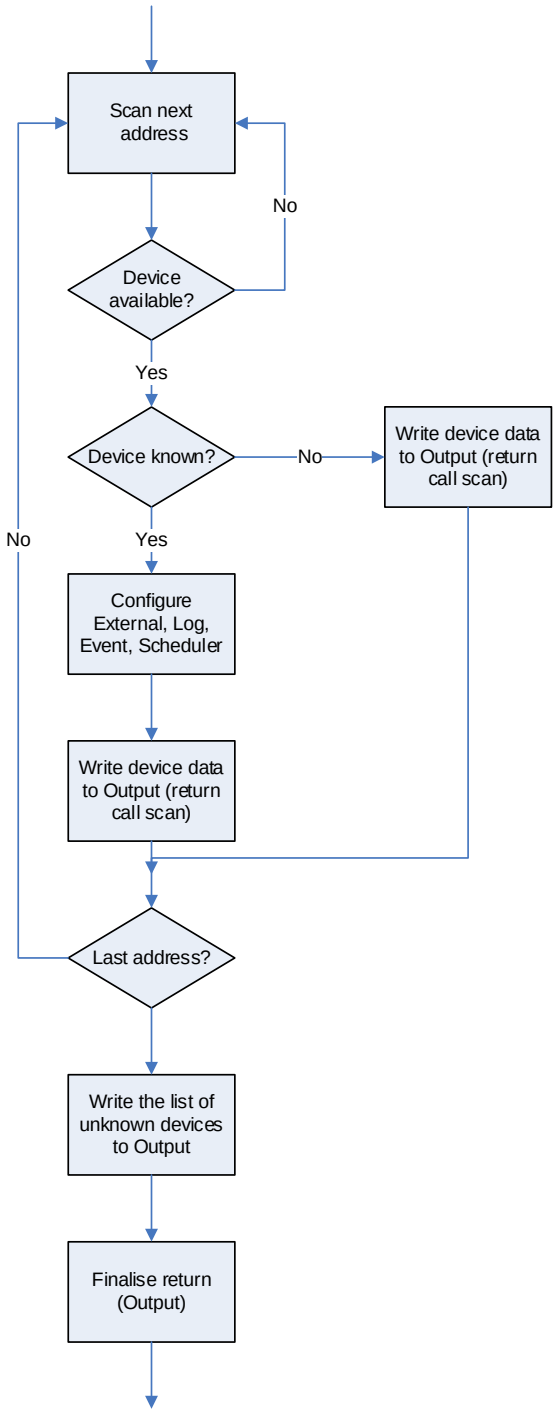

Figure 1: Flow chart for of the firmware

The devices found during the scan are all output.

1.6 Examples

Example 1:

Result of a ScanDevices command

```
<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>
```

The `External`, `LOG`, `EVENTS` and `SCHEDULE` databases are reconfigured (if required) for all devices defined in the `DEVICEDB`.
The `<UnknownDevices>` section is output at the end of the scan for the devices that are not found in the `DEVICEDB`:

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

output (example). These devices are not configured automatically.

Note that not all Modbus devices can be detected on the bus because not all devices respond to general bus queries (e.g. UMG 96 from Janitza). Therefore the list of devices that are not recognised in then `<UnknownDevices>` section may be incomplete.

Example 2: Complete DEVICEDB for Janitza UMG96RM and 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 History

Version	Date	Editor	Changes
1.0.1	23/02/2021	IVH	First English issue
1.0.0	10/11/2020	SH	First issue