M-BUS to MQTT

Funktion

Mit dem "Plugin" ist es möglich Daten von einem M-Bus Gerät (Gaszähler, Stromzähler, Wärmezähler, Wasserzähler, ...) auszulesen und die Informationen per MQTT zu veröffentlichen. Mit dem MQTT Plugin / Loxberry 3.0 ist es anschliessend möglich diese Informationen auf den Miniserver zu bringen.

Weitere Infos unter: https://de.wikipedia.org/wiki/M-Bus_(Feldbus)

Das ganze basiert auf der tollen Arbeit von themole
https://the78mole.de/taking-your-m-bus-online-with-mqtt/

Voraussetzungen

- Zähler mit MBUS Interface.
  - Ich habe den Gaszähler GWF BK-G4A mit Anschlussmodul ACM 5.2 M-BUS für BGZcode nachgerüstet: https://docuthek.kromschroeder.com/documents/download.php?lang=en&doc=18981 ~30 Euro
- MBUS Master
  - USB: https://www.aliexpress.com/item/1005004863868466.html?spm=a2g0o.order_list.order_list_main.5.1f191802p7BtNX ~25 Euro
  - oder
  - TTL: https://www.aliexpress.com/item/4000020182211.html?spm=a2g0o.order_list.order_list_main.21.1f191802p7BtNX ~18 Euro

Download

→ Das "Plugin" ist noch nicht wirklich ein Plugin. Und ich weiss auch nicht, ob es zeitnahe zu einem Plugin wird. Dennoch möchte ich die Infos euch hier nicht vorenthalten. Falls jemand den Drang verspührt dies in ein sauberes Plugin umzuwandeln, darf er gerne übernehmen.

Installation

Da es kein einfaches Plugin ist, musst du dich per SSH auf den Loxberry verbinden.

M-Bus Reader Software (libmbus)

Tool installieren

Zum compilieren brauchst du libtool und dies holst du dir mit root Rechten

```
sudo apt-get install libtool
```

Gehe ins Pluginverzeichnis und hol dir libmbus:

```
cd /opt/loxberry/bin/plugins/mbus
git clone https://github.com/rscada/libmbus.git
```

Anschliessend gehst du in den libmbus ordner und führst nacheinander folgende Befehle aus

```
cd libmbus
./build.sh
./configure
make -j4
sudo make install
sudo ldconfig
```

Verbindungs Test

Anschliessend kannst du den ersten Test mit dem loxberry user machen. Hier im Beispiel wird mit dem per USB angeschlossenen M-Bus Master mit 2400 baude gescannt. Falls nichts gefunden wird. Versucht es mit 300 oder 9600 nochmals.

```
loxberry@loxberry:~ $ mbus-serial-scan -b 2400 /dev/ttyUSB0
Found a M-Bus device at address 3
```

Wenn du mehr als ein Gerät am MBUS hast kann es sein, dass die Primäreadresse Konflikte hat. Es lohnt sich daher nacher die Sekundäre Adresse zu verwenden. Diese kannst du wie folgt ermitteln:

```
loxberry@loxberry:~ $ mbus-serial-scan-secondary -b 2400 /dev/ttyUSB0

Found a device on secondary address 21013154E22E3103 [using address mask
2FFFFFFFFFFFFFF]
```

Daten auslesen

Anschliessend kannst du die Daten von deinem Meter auslesen. Modifizier den Befehl mbus-serial-request-data mit deiner Bauderate und deiner Sekundäradresse:

```
loxberry@loxberry:~ $ mbus-serial-request-data -b 2400 /dev/ttyUSB0 21013154E22E3103

<?xml version="1.0" encoding="ISO-8859-1"?>

<MBusData>

  <SlaveInformation>
      <Id>21013195</Id>
      <Manufacturer>GWF</Manufacturer>
      <Version>54</Version>
      <ProductName></ProductName>
      <Medium>Gas</Medium>
      <AccessNumber>230</AccessNumber>
      <Status>00</Status>
      <Signature>0000</Signature>
  </SlaveInformation>

  <DataRecord id="0">
      <Function>Instantaneous value</Function>
      <StorageNumber>0</StorageNumber>
      <Unit>Fabrication number</Unit>
      <Value>21013195</Value>
      <Timestamp>2023-03-25T16:51:43Z</Timestamp>
  </DataRecord>

  <DataRecord id="1">
      <Function>Instantaneous value</Function>
      <StorageNumber>0</StorageNumber>
      <Unit>Volume (1e-2  m^3)</Unit>
      <Value>3146588</Value>
      <Timestamp>2023-03-25T16:51:43Z</Timestamp>
  </DataRecord>

</MBusData>
```

```
loxberry@loxberry:~ $
```

XML Daten in JSON umwandeln und veröffentliche

Dazu solltest du den MQTT client mosquitto-client, den JSON Prozessor ./jq und den python installer pip installieren.

```
sudo apt install mosquitto-clients jq python3-pip
```

Anschliessend kann der YAML/XML Prozessor yq installiert werden.

```
sudo pip3 install yq
```

Dann kannst mit dem pipe command | und xq das ganze von einem XML in ein JSON umwandeln:

```
loxberry@loxberry:~/bin/plugins/mbus/bin $ mbus-serial-request-data -b 2400 /dev/ttyUSB0 21013154E22E3103 | xq .

{
"MBusData": {
  "SlaveInformation": {
    "Id": "21013154",
    "Manufacturer": "GWF",
    "Version": "54",
    "ProductName": null,
    "Medium": "Gas",
    "AccessNumber": "11",
    "Status": "00",
    "Signature": "0000"
  },
  "DataRecord": [
    {
      "@id": "0",
      "Function": "Instantaneous value",
      "StorageNumber": "0",
      "Unit": "Fabrication number",
      "Value": "21013154",
      "Timestamp": "2023-03-25T19:35:15Z"
    },
    {
```

```
      "@id": "1",
      "Function": "Instantaneous value",
      "StorageNumber": "0",
      "Unit": "Volume (1e-2  m^3)",
      "Value": "3146777",
      "Timestamp": "2023-03-25T19:35:15Z"
    }
  ]
}
```

}

loxberry@loxberry:~/bin/plugins/mbus/bin $

Auslese Script erstellen

Jetzt bist du schon fast durch und es wird Zeit das ganze in eine ausführbare Datei zu packen "read_send_meters_mqtt.sh". Geh dazu in den mbus pfad, erstelle eine leere Datei namens read_send_meters_mqtt.sh und gib der datei die ausführungs rechte:

```
cd /opt/loxberry/bin/plugins/mbus

touch read_send_meters_mqtt.sh

chmod u+x read_send_meters_mqtt.sh
```

Anschliessend kannst du die Datei mit dem Texteditor nano öffnen und den Inhalt aus dieser Datei kopieren:

read_send_meters_mqtt.zip

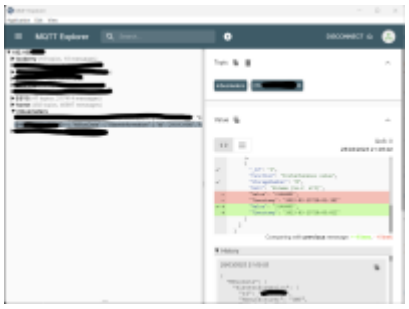
```
nano read_send_meters_mqtt.sh
```

Anschliessend sollst du den Text natürlich noch deinem System anpassen. Editiere dazu mindestens BAUDRATE, MQTT_HOST und MQTT_PASS. Nach dem Speichern der Datei kannst du diese kurz testet:

```
loxberry@loxberry:~/bin/plugins/mbus/bin $ ./read_send_meters_mqtt.sh

Sat Mar 25 20:01:17 GMT 2023 Sending data to host 192.168.1.10 as user 'loxberry' using topic 'mbusmeters/'. Getting data from 21013154E22E3103... 725 bytes sent {

"id": "21013154",
"manufacturer": "GWF",
```

```
"medium": "Gas",
"records": 2

}

loxberry@loxberry:~/bin/plugins/mbus/bin $
```

Am besten vorher noch kurz den MQTT Explorer starten, damit ihr seht, ob da was daher kommt:



Die Daten kann man wie üblich mit dem Loxberry MQTT Plugin und einem virtualen TCP oder UDP eingang nochher ins loxone übernehmen.

Das Script "read_send_meters_mqtt.sh" sucht beim ersten Start den Bus ab und fügt die gefundenen Geräte der Datei "~/config/plugins/mbus/addresses.txt" hinzu. Wenn das Script startet und diese Datei bereits existiert, wird nicht erneut gescannt und lediglich die Adressen in der Datei abgefragt. Wenn ihr also weitere Geräte abfragen möchtet, dann müsst ihr die Datei einfach vor dem nächsten ausführen löschen:

```
rm ~/config/plugins/mbus/addresses.txt
```

Automatisierte Abfrage erstellen

So wenn alles funktioniert und man mit dem ganzen zufrieden ist, wird es an der Zeit diesen Script automatisch zu starten. Dies kann man mit einem crontab machen:

```
crontab -e
```

Und dann dort folgendes anfügen:

```
PATH=$PATH:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib */5 * * * * cd /opt/loxberry/bin/plugins/mbus && ./read_send_meters_mqtt.sh » /opt/loxberry/log/mbus_mqtt.log
```

Dann wird ins verzeichnis gewechselt und alle 5min das script ausgeführt. Der Autput wird ins /opt/loxberry/log/mbus_mqtt.log geschrieben.

From:
https://wiki.loxberry.de/ - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:
https://wiki.loxberry.de/modifikationen_hacks/mbus2mqtt

Last update: **2023/03/30 22:44**