

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_\(Feldbus\)](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
 - 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 verspürt 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 baudrate 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  
2FFFFFFFFFFFFFFF]
```

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 $

```

Jetzt bist du schon fast durch und es wird Zeit das ganze in eine ausführbare Datei zu packen
 "read_send_meters_mqtt.sh"

```

#!/bin/bash

ADDRESS_FILE=~/.addresses.txt BAUDRATE=2400 DEVICE=/dev/ttyUSB0
MQTT_HOST=192.168.1.10 MQTT_USER=loxberry MQTT_PASS=your_mqtt_passowrd
MQTT_TOPIC=mbusmeters

if [ ! -f $ADDRESS_FILE ]; then

    mbus-serial-scan-secondary -b $BAUDRATE $DEVICE \
        | sed -e 's/^. *y address \([0-9A-Fa-f]\+\) .*$/\1/' >
    $ADDRESS_FILE

fi

echo -e "\n $(date)" echo "Sending data to host $MQTT_HOST as user '$MQTT_USER' using topic
'$MQTT_TOPIC/'."

while read ameter do

    echo -n "Getting data from $ameter..."
    # The sed is for replacing the @ with _ to be able to match on it in
    HASS templates
    METER_DATA=$(mbus-serial-request-data-multi-reply -b $BAUDRATE $DEVICE
    $ameter | xq . | sed -e "s/@/_/")
    # echo -e "\n$METER_DATA"
    /usr/bin/mosquitto_pub -h $MQTT_HOST -u $MQTT_USER -P $MQTT_PASS \
        -t $MQTT_TOPIC/$ameter -m "${METER_DATA}"
    BYTCNT=$(echo "$METER_DATA" | wc -c)
    echo "  $BYTCNT bytes sent"
    echo "$METER_DATA" | jq '{
        id          : .MBusData.SlaveInformation.Id,
        manufacturer: .MBusData.SlaveInformation.Manufacturer,

```

```
medium      : .MBusData.SlaveInformation.Medium,  
records     : .MBusData.DataRecord | length  }'
```

```
done < <(cat $ADDRESS_FILE)
```

From:

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

Permanent link:

https://wiki.loxberry.de/modifikationen_hacks/mbus2mqtt?rev=1679773621

Last update: **2023/03/25 20:47**