DrvMBus

DrvMBus är en drivrutin för MBus till Web Port.

OBS!

Drivrutinen är i BETA-version

För att använda DrvMBus skapas först en IO-enhet som sedan används för att kommunicera via MBus TCP eller UDP.

Installation

Installationen av DrvMBus görs genom att välja DrvMBus vid installationen av Web Port. Se *kapitel 1* för mer information kring installationsprocessen.

IO-enhet

För att använda DrvMBus skapas för en IO-enhet enligt kapitel 3.2.1 i manualen.
Utöver generella inställningar (se kapitel 3.2.4 i manualen) finns följande inställningar för en IO-enhet av typen DrvMBus.

------ Exempel på inställningar för drivrutin -------

Protokoll

Välj TCP eller UDP beroende på inställningarna i MBus-gatewayen.

IP-adress eller enhetsnamn

IP eller hostname

Port

Anger port till MBus-gateway

Max lästid

Maximal lästid innan frågan time:ar ut.

Paketcache

Tid som MBuspaket sparas i Web Port cache. Taggar som hämtar olika värden från samma MBus-meddelande frågar då mot samma sparade paket.

Initieringadress

Om ingen initieringsadress anges görs endast enklare anslutningsvalidering vid TCP (verifierar att port är öppen), och vid UDP ingen alls. Om initieringsadress anges används en fråga till den mätaren för att bekräfta om anslutnings upprättats.

Taggar

För mer information om hur tagglistor och taggar skapas se kapitel 4 i manualen.

” Taggadresser:

Web Port har stöd för adressering via primäradress eller sekundäradress. Primäradress anges som en siffra mellan 0 och 250.

Sekundäraddress på formatet:

NNNNNNNN.AAAA.BB.CC .

Där NNNNNNNN är serinummer, AAAA är producer code (HEX), BB är firmwareversion (HEX), CC är medium (HEX).

Exempel 11034493.4C43.36.02

MBus-mätaren svarar på två olika format, paket med variabel längd eller paket med fast längd. Dess behöver adresseras på olika sätt. För att se innehåll i MBus-paketen, ange endast adress och välja datatyp String. Då visas hela paketet omvandlat till JSON-format och kan användas för att se vad som kan läsas ut.

Önskat värde anges sedan på en ny tagg efter primäradressen som separeras med @-tecken. T.ex. 1@EnergyWh.Instantaneous.Tariff0.SubUnit0

Även information om mätaren anges på samma sätt. T.ex. 1@Manufr

Som standard returneras normaliserade värden för paket med variabel längd. Önskat ett icke normaliserad värde ange !nv efter adressen. T.ex. 1@EnergyWh.Instantaneous.Tariff0.SubUnit0!nv

Vissa mätare kräver att man skicka en ACK (NKE-paket) innan man kan läsa ut värdet. För att välja att ett NKE-paket skall skickas, ange !nke, efter adressen. (går att kombinera med !nv

T.ex. 1@EnergyWh.Instantaneous.Tariff0.SubUnit0!nv!nke

Vissa värden är av samma typ med ligger på olika StorageNumber. Dessa addresserar då genom att ange StorageNumberX (där X är aktuellt StorageNumber) efter adresse.

T.ex. 1@EnergyWh.Instantaneous.Tariff0.SubUnit0.StorageNumber0

Utöver adressering på namn finns även möjlighet att adressera på värdets index (ordning). Det hittas ofta i mätarens manual och kan även läsas ut i JSON-svaret i respektive Record. I exemplet på nästa sida visas att paket med index 0.

Adresseringen blir då t.ex. 01247556@0

” Exempel:

Exempel på hur paketen i JSON-format kan användas:

JSON-svar


```
{
  "IdentificationNo": 1247556,
  "Manufr": 7222,
  "Version": 90,
  "DeviceType": 2,
  "TransmissionCounter": 19,
  "Status": 0,
  "Signature": 0,
   "FRecords": null,
   "PacketType": "Variable",
   "Address": 0
  "VRecords": {
    "0": {
      "RecordType": 4,
      "Function": "Instantaneous",
      "StorageNumber": 0,
      "Tariff": 0,
      "SubUnit": 0,
      "ValueDataType": "_32_Bit_Integer",
      "ValueData": null,
      "Value": 417920,
      "Units": [{
        "Units": "EnergyWh",
        "Unit": "Wh",
        "Magnitude": 2,
        "Quantity": "Energy",
        "VIF_string": "05h"
      }],
      "Magnitude": 2,
      "Offset": 0,
      "Name": "EnergyWh.Instantaneous.Tariff0.SubUnit0",
      "NormalizedValue": {
        "Item1": "EnergyWh.Instantaneous.Tariff0.SubUnit0",
        "Item2": 41792000.0
      }
    }
  }
}
```


Normaliserad värdet väljs genom att ange adress: 0@EnergyWh.Instantaneous.Tariff0.SubUnit0
Icke normaliserat: 0@EnergyWh.Instantaneous.Tariff0.SubUnit0!nv
Mätarid: 0@IdentificationNo

” Import:

MBus-drivern ger även möjlighet att scanna av och importera taggar från mätare.

I en tagglista finns möjlighet att välja importera taggar, välja MBus-enheten och finns möjlighet vilja vilka adresser som skall importeras.

Det går antigen att ange enskilda adresser eller adressområden.

Exempel för primära adresser: 1,20-30,145 försöker läsa in adresserna 1, 20 till 30 och 145.

Exempel sekundäradresser: 1FFFFFFF eller FFFFFFFF.FFFF.FF.FF , där F används som wildcard vid sökning.

Sökningen kan ta en viss tid.

Under /assets/import/drvmbus finns ett script för att läsa in 2 taggar per mätare. Paketen i JSON-format samt mätar-ID.

Fler script där det kommer gå att läsa in taggar för olika värden kommer läggas till vid senare tillfälle.