



DLMS/COSEM

En kort til introduktion til brug af DLMS/COSEM udlæsning af data fra Kamstrups Omnipower-måler hos Radius.

Indholdsfortegnelse

1.	Indledning.....	3
2.	Rekvirering af nøgler og opdatering af firmware.....	3
3.	Forbindelse.....	3
4.	Udlæste registre.....	4
5.	Eksempel på dekryptering.....	5

1. Indledning

Denne manual, sammen med bilagene, er tænkt som en introduktion til aflæsning af Kamstrups elmålere i Radius' område v. h. a. DLMS/COSEM protokollen. Det er ikke en detaljeret beskrivelse af det nødvendige kodelarbejde, og Radius og Kamstrup kan ikke stille ressourcer til rådighed til en implementering.



You are on your own¹

¹ Det er ikke helt sandt, der er en hel del hjælp at hente på nettet.

Alternativet er at benytte et 'færdigt' modul, som f. eks. fra [SmartMe](#), som f.eks. kan fås [her](#).

2. Rekvirering af nøgler og opdatering af firmware

Nøgler rekvireres ved at sende email til kundesupport@radiusnet.dk med navn, aftagenummer og målnummer.

Måleren skal være opdateret til den nyeste FW version. Hvis det ikke er tilfældet, vil måleren blive opdateret samtidigt med rekvireringen af nøglerne.



3. Forbindelse

Forbindelse til måleren foregår via CCC porten.



Detaljer fremgår af bilaget OMNIPOWER-HAN.

Hvis man forbinder direkte til stikket er det en RS232 forbindelse (2400, none, 8, 1).

Hvis man bruger det norske HAN modul, er det en M-bus.

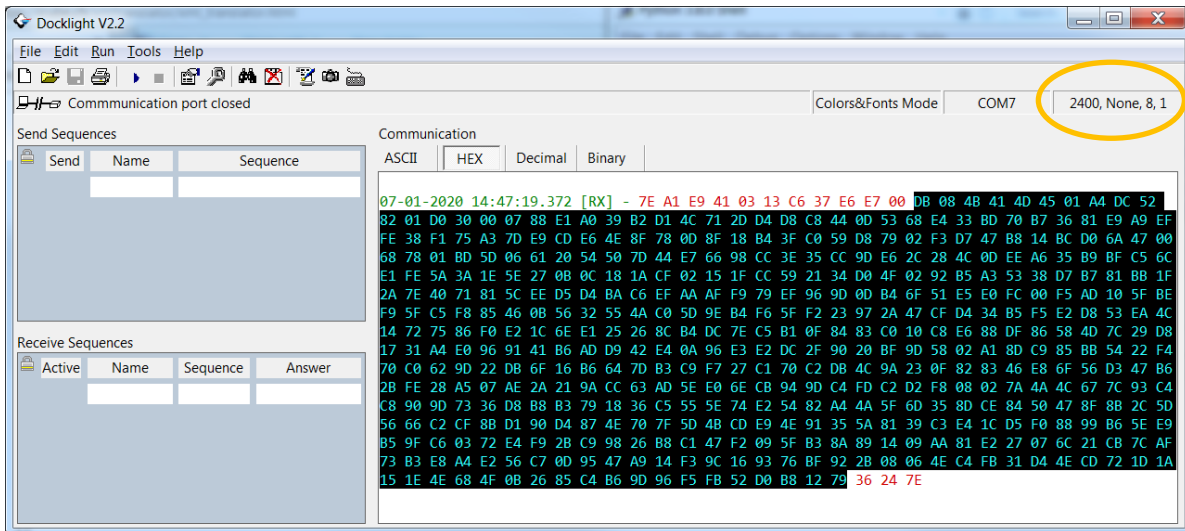
4. Udlæste registre

Måleren pusher hvert 10. sekund disse registre, som også fremgår af afsnit 5.2.1 i bilaget.

Logical name	Object name	Unit	Scaler
1.1.0.2.129.255	OBIS list version identifier -		
1.1.1.8.0.255	Active energy A14	Wh	0
1.1.2.8.0.255	Active energy A23	Wh	0
1.1.3.8.0.255	Reactive energy R12	varh	0
1.1.4.8.0.255	Reactive energy R34	varh	0
1.1.0.0.1.255	Meter number 1 -		
1.1.1.7.0.255	Actual power P14	W	0
1.1.2.7.0.255	Actual power P23	W	0
1.1.3.7.0.255	Actual power Q12	var	0
1.1.4.7.0.255	Actual power Q34	var	0
0.1.1.0.0.255	Real-time clock -		
1.1.32.7.0.255	RMS voltage of phase L1	V	0
1.1.52.7.0.255	RMS voltage of phase L2	V	0
1.1.72.7.0.255	RMS voltage of phase L3	V	0
1.1.31.7.0.255	RMS current of phase L1	A	-2
1.1.51.7.0.255	RMS current of phase L2	A	-2
1.1.71.7.0.255	RMS current of phase L3	A	-2
1.1.21.7.0.255	Actual power P14 of phase L1	W	0
1.1.41.7.0.255	Actual power P14 of phase L2	W	0
1.1.61.7.0.255	Actual power P14 of phase L3	W	0
1.1.33.7.0.255	Power factor of phase L1 -		
1.1.53.7.0.255	Power factor of phase L2 -		
1.1.73.7.0.255	Power factor of phase L3 -		
1.1.13.7.0.255	Power factor Total -		
1.1.22.7.0.255	Active power P23 of phase L1	W	0
1.1.42.7.0.255	Active power P23 of phase L2	W	0
1.1.62.7.0.255	Active power P23 of phase L3	W	0
1.1.22.8.0.255	Active energy A23 of phase L1	Wh	0
1.1.42.8.0.255	Active energy A23 of phase L2	Wh	0
1.1.62.8.0.255	Active energy A23 of phase L3	Wh	0
1.1.21.8.0.255	Active energy A14 of phase L1	Wh	0
1.1.41.8.0.255	Active Energy A14 of phase L2	Wh	0
1.1.61.8.0.255	Active Energy A14 of phase L3	Wh	0

5. Eksempel på dekryptering

Nedenfor er vist et eksempel på håndtering af outputtet fra måleren med de let tilgængelige værktøjer Docklight Python og GuruX.



Eksempel på krypteret output fra måler. Selve cipher tekst er markeret.

Outputtet er krypteret med to nøgler. Et eksempel på dekryptering v. h. a. Python:

```
from Crypto.Cipher import AES # Library: PyCryptodome

encryption_key = bytes.fromhex('5AD84121D9D20B364B7A11F3C1B5827F') # gpk60
authentication_key = bytes.fromhex('AFB3F93E3E7204EDB3C27F96DBD51AE0') # gpk61

header = '7E A1 E9 41 03 13 C6 37 E6 E7 00' # bruges ikke ifm. dekryptering

# +-----+-----+-----+-----+-----+-----+-----+-----+
# | 1 byte | len + 8 bytes | x bytes | 1 or 5 bytes | y bytes | 12 bytes |
# +-----+-----+-----+-----+-----+-----+-----+-----+
# | Tag | System title | Length | Security header | Cipher text | Auth tag |
# +-----+-----+-----+-----+-----+-----+-----+-----+
#
#                               Invokation
#           T l ST                               L           S Counter           Data(Cipher text)
sidste 12 bytes er Auth tag
cipher_text = 'DB 08 4B 41 4D 45 01 A4 DC 52 82 01 D0 30 00 07 88 E1 A0 39 B2 D1 4C 71
2D D4 D8 C8 44 0D 53 68 E4 33 BD 70 B7 36 81 E9 A9 EF FE 38 F1 75 A3 7D E9 CD E6 4E 8F
78 0D 8F 18 B4 3F C0 59 D8 79 02 F3 D7 47 B8 14 BC D0 6A 47 00 68 78 01 BD 5D 06 61 20
54 50 7D 44 E7 66 98 CC 3E 35 CC 9D E6 2C 28 4C 0D EE A6 35 B9 BF C5 6C E1 FE 5A 3A 1E
5E 27 0B 0C 18 1A CF 02 15 1F CC 59 21 34 D0 4F 02 92 B5 A3 53 38 D7 B7 81 BB 1F 2A 7E
40 71 81 5C EE D5 D4 BA C6 EF AA AF F9 79 EF 96 9D 0D B4 6F 51 E5 E0 FC 00 F5 AD 10 5F
```

```

BE F9 5F C5 F8 85 46 0B 56 32 55 4A C0 5D 9E B4 F6 5F F2 23 97 2A 47 CF D4 34 B5 F5 E2
D8 53 EA 4C 14 72 75 86 F0 E2 1C 6E E1 25 26 8C B4 DC 7E C5 B1 0F 84 83 C0 10 C8 E6 88
DF 86 58 4D 7C 29 D8 17 31 A4 E0 96 91 41 B6 AD D9 42 E4 0A 96 E3 E2 DC 2F 90 20 BF 9D
58 02 A1 8D C9 85 BB 54 22 F4 70 C0 62 9D 22 DB 6F 16 B6 64 7D B3 C9 F7 27 C1 70 C2 DB
4C 9A 23 0F 82 83 46 E8 6F 56 D3 47 B6 2B FE 28 A5 07 AE 2A 21 9A CC 63 AD 5E E0 6E CB
94 9D C4 FD C2 D2 F8 08 02 7A 4A 4C 67 7C 93 C4 C8 90 9D 73 36 D8 B8 B3 79 18 36 C5 55
5E 74 E2 54 82 A4 4A 5F 6D 35 8D CE 84 50 47 8F 8B 2C 5D 56 66 C2 CF 8B D1 90 D4 87 4E
70 7F 5D 4B CD E9 4E 91 35 5A 81 39 C3 E4 1C D5 F0 88 99 B6 5E E9 B5 9F C6 03 72 E4 F9
2B C9 98 26 B8 C1 47 F2 09 5F B3 8A 89 14 09 AA 81 E2 27 07 6C 21 CB 7C AF 73 B3 E8 A4
E2 56 C7 0D 95 47 A9 14 F3 9C 16 93 76 BF 92 2B 08 06 4E C4 FB 31 D4 4E CD 72 1D 1A 15
1E 4E 68 4F 0B 26 85 C4 B6 9D 96 F5 FB 52 D0 B8 12 79'
cipher_text = bytes.fromhex(cipher_text.replace(' ', ''))
footer = '36 24 7E' # bruges ikke ifm. dekryptering

system_title = cipher_text[2:2+8]
initialization_vector = system_title + cipher_text[14:14+4]
additional_authenticated_data = cipher_text[13:13+1] + authentication_key
authentication_tag = cipher_text[len(cipher_text)-12:len(cipher_text)]

cipher = AES.new(encryption_key, AES.MODE_GCM, nonce=initialization_vector,
mac_len=len(authentication_tag)) #len(authentication_tag) 4..14
cipher.update(additional_authenticated_data)
plaintext = cipher.decrypt_and_verify(cipher_text[18:len(cipher_text)-12],
authentication_tag)
print(plaintext.hex())

```

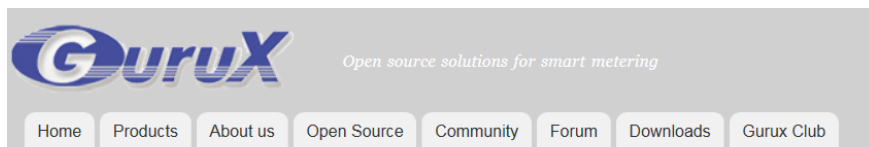
Den dekrypterede streng vil se sådan ud:

```

0f00000000c07e40107020e2f14ff80000002410a0e4b616d73747275705f56303030310906010101
0800ff06000d394c09060101020800ff060000000009060101030800ff060000452c09060101040800
ff060000000009060101000001ff0601a4dc5209060101010700ff060000000009060101020700ff06
0000000009060101030700ff060000000009060101040700ff060000000009060001010000ff090c07
e40107020e2f14ff80000009060101200700ff1200e009060101340700ff1200df09060101480700ff
1200df090601011f0700ff060000000009060101330700ff060000000009060101470700ff06000000
0009060101150700ff060000000009060101290700ff0600000000090601013d0700ff060000000009
060101210700ff12006409060101350700ff12006409060101490700ff120064090601010d0700ff12
006409060101160700ff0600000000090601012a0700ff0600000000090601013e0700ff0600000000
09060101160800ff0600000000090601012a0800ff0600000000090601013e0800ff06000000000906
0101150800ff06000468a409060101290800ff060004678c090601013d0800ff060004691b

```

Oversat til XML OBIS koder v. h. a. GuruX DLMS Translator
<http://www.gurux.fi/GuruxDLMSTranslator>



Home

Manufacturers Ciphering ASCII ASN.1 OBIS code Messages PDU

To XML To PDU Standard XML Hex

```
0f00000000c07e40107020e2f14ff80000002410a0e4b616d73747275705f563030303109060101010800ff0
6000d394c09060101020800ff060000000009060101030800ff060000452c09060101040800ff060000000009
060101000001ff0601a4dc5209060101010700ff060000000009060101020700ff06000000000906010103070
0ff0600000000009060101040700ff060000000009060001010000ff090c07e40107020e2f14ff8000000906010
1200700ff1200e009060101340700ff1200df09060101480700ff1200df090601011f0700ff0600000000090601
01330700ff060000000009060101470700ff060000000009060101150700ff060000000009060101290700ff0
600000000090601013d0700ff060000000009060101210700ff12006409060101350700ff120064090601014
90700ff120064090601010d0700ff12006409060101160700ff0600000000090601012a0700ff060000000009
0601013e0700ff060000000009060101160800ff0600000000090601012a0800ff0600000000090601013e08
00ff060000000009060101150800ff06000468a409060101290800ff060004678c090601013d0800ff06000469
1b
```

```
<DataNotification>
<LongInvokeldAndPriority Value="00000000" />
<!--2020-01-07 14:47:20-->
<DateTime Value="07E40107020E2F14FF800000" />
<NotificationBody>
<DataValue>
<Structure Qty="41" >
<String Value="Kamstrup_V0001" />
<!--1.1.1.8.0.255-->
<OctetString Value="0101010800FF" />
<UInt32 Value="000D394C" />
<!--1.1.2.8.0.255-->
<OctetString Value="0101020800FF" />
<UInt32 Value="00000000" />
<!--1.1.3.8.0.255-->
<OctetString Value="0101030800FF" />
```

Den fulde XML:

```
<DataNotification>
<LongInvokeldAndPriority Value="00000000" />
<!--2020-01-07 14:47:20-->
<DateTime Value="07E40107020E2F14FF800000" />
<NotificationBody>
<DataValue>
<Structure Qty="41" >
<String Value="Kamstrup_V0001" />
<!--1.1.1.8.0.255-->
<OctetString Value="0101010800FF" />
<UInt32 Value="000D394C" />
<!--1.1.2.8.0.255-->
<OctetString Value="0101020800FF" />
<UInt32 Value="00000000" />
<!--1.1.3.8.0.255-->
<OctetString Value="0101030800FF" />
<UInt32 Value="0000452C" />
<!--1.1.4.8.0.255-->
<OctetString Value="0101040800FF" />
<UInt32 Value="00000000" />
```

```
<!--1.1.0.0.1.255-->
<OctetString Value="0101000001FF" />
<UInt32 Value="01A4DC52" />
<!--1.1.1.7.0.255-->
<OctetString Value="0101010700FF" />
<UInt32 Value="00000000" />
<!--1.1.2.7.0.255-->
<OctetString Value="0101020700FF" />
<UInt32 Value="00000000" />
<!--1.1.3.7.0.255-->
<OctetString Value="0101030700FF" />
<UInt32 Value="00000000" />
<!--1.1.4.7.0.255-->
<OctetString Value="0101040700FF" />
<UInt32 Value="00000000" />
<!--0.1.1.0.0.255-->
<OctetString Value="0001010000FF" />
<!--2020-01-07 14:47:20-->
<OctetString Value="07E40107020E2F14FF800000" />
<!--1.1.32.7.0.255-->
<OctetString Value="0101200700FF" />
<UInt16 Value="00E0" />
<!--1.1.52.7.0.255-->
<OctetString Value="0101340700FF" />
<UInt16 Value="00DF" />
<!--1.1.72.7.0.255-->
<OctetString Value="0101480700FF" />
<UInt16 Value="00DF" />
<!--1.1.31.7.0.255-->
<OctetString Value="01011F0700FF" />
<UInt32 Value="00000000" />
<!--1.1.51.7.0.255-->
<OctetString Value="0101330700FF" />
<UInt32 Value="00000000" />
<!--1.1.71.7.0.255-->
<OctetString Value="0101470700FF" />
<UInt32 Value="00000000" />
<!--1.1.21.7.0.255-->
<OctetString Value="0101150700FF" />
<UInt32 Value="00000000" />
<!--1.1.41.7.0.255-->
<OctetString Value="0101290700FF" />
<UInt32 Value="00000000" />
<!--1.1.61.7.0.255-->
<OctetString Value="01013D0700FF" />
<UInt32 Value="00000000" />
<!--1.1.33.7.0.255-->
<OctetString Value="0101210700FF" />
<UInt16 Value="0064" />
<!--1.1.53.7.0.255-->
<OctetString Value="0101350700FF" />
<UInt16 Value="0064" />
<!--1.1.73.7.0.255-->
<OctetString Value="0101490700FF" />
<UInt16 Value="0064" />
<!--1.1.13.7.0.255-->
<OctetString Value="01010D0700FF" />
<UInt16 Value="0064" />
<!--1.1.22.7.0.255-->
<OctetString Value="0101160700FF" />
<UInt32 Value="00000000" />
<!--1.1.42.7.0.255-->
<OctetString Value="01012A0700FF" />
<UInt32 Value="00000000" />
<!--1.1.62.7.0.255-->
<OctetString Value="01013E0700FF" />
<UInt32 Value="00000000" />
<!--1.1.22.8.0.255-->
<OctetString Value="0101160800FF" />
<UInt32 Value="00000000" />
<!--1.1.42.8.0.255-->
<OctetString Value="01012A0800FF" />
<UInt32 Value="00000000" />
```



```
<!--1.1.62.8.0.255-->  
<OctetString Value="01013E0800FF" />  
<UInt32 Value="00000000" />  
<!--1.1.21.8.0.255-->  
<OctetString Value="0101150800FF" />  
<UInt32 Value="000468A4" />  
<!--1.1.41.8.0.255-->  
<OctetString Value="0101290800FF" />  
<UInt32 Value="0004678C" />  
<!--1.1.61.8.0.255-->  
<OctetString Value="01013D0800FF" />  
<UInt32 Value="0004691B" />  
</Structure>  
</DataValue>  
</NotificationBody>  
</DataNotification>
```