**Fjärravläsnng av Kamstrup Omnipower elmätare (Umeå Energi m.fl.) av SM2GCT, del 3**

Om allt har fungerat bra hittills, eller om det krånglar kan jag ge några tips som är specifika för Linux och några allmänna tips och informationer.

Tyvärr är jag okunnig på vad Win kan hitta på för att krångla till det, men en hel del info finns säkert på nätet, kanske är säkerhetsinställningarna lite mer förlåtande än på Linux.

Ett par saker som kan hända är att USB-serieporten krånglar på något vis eller att programmet klagar på saknade moduler (de som syns som ‘import xxx’ i programkoden).

Tips 0: Kör du Linux bör du alltid se till att OS är uppdaterat.

Tips1: öppna terminalfönstret och kör ‘pip3 install pyserial’ om ‘serial’ inte hittas.

Tips2: Prova att öppna terminalfönstret och kör ‘sudo chown dittusernamn /dev/ttyUSBx' om du nekas tillgång till USB-serieporten. (Det här sägs vara en bugg i Linux.. :( )

Ändringen varar tills du loggar ut eller kopplar ur USB-serieporten, sedan måste du göra om.

Tips3: För att göra ändringen i tips2 permanent måste du ändra i udev-reglerna:

Öppna terminalfönster och kör 'cd /etc/udev/rules.d' och behåll fönstret öppet.

Kör 'sudo nano 60-extra-acl.rules' och skriv som första rad i editorn som öppnas:

'KERNEL=="ttyUSB[0-9]\*", OWNER="dittusernamn" . Tryck 'Enter'.

Tryck ctrl-x för att avsluta, bekräfta skrivning till filen med den bokstav som motsvarar 'ja'.

När den nu skapade filen sparats kan du stänga terminalfönstret.

Logga ut och in igen för att ändringen ska verka. Ta ut och sätt in USB-serieporten igen.

Information 1: Här [kamread-exempel.py] finns det fungerande programexemplet.

Information 2: Här [kamread-exempel-noreadser.py] finns programexemplet med simulerad serieport.

Information 3: Här [kamread-exempel-loopreadser.py] finns programexemplet med loopad seieport (verkliga data sänds på TXD).

Koppla helt enkelt ihop TXD och RXD på USB-serieporten så visar ev. indikerings-LED att data sänds och tas emot. Hastigheten är satt låg för att man ska hinna se trafiken.

Information 4: Här [3,3V-inverter.png] finns alternativ inverterar-krets som troligtvis fungerar bra mot 3,3V ingång på Raspberry som har egen serieport. På Rpi fungerar även USB- serieporten som beskrivits, möjligtvis ser udev-reglerna annorlunda ut, har inte kollat.

Information 5: Här[kamread-rxdata.txt] finns en text som visar hur data som P1 sänder ser ut.

Här[push-list-p1.txt] finns en lista över förekommande datarader, mer lättläst.

Här [push-list-p1-ordered.txt] finns samma lista fast ordnad på grupperat sätt.

Innehållet i listorna har tagits ur ett officiellt dokument och arrangrerats så att man lättare kan finna vad man söker.

Oavsett du gör egna interface, använder annat materiel du har eller skriver helt egen programkod hoppas jag att den här artikelserien har både väckt intresse för ämnet och kanske till och med gjort att du hittat på egna lösningar som passar dig bättre. Det var min ursprungliga idé med det hela.

Mina kontaktuppgifter finns på både SSA och FURA om du vill fråga, diskutera eller anmärka på någon tveksamhet eller felaktighet. Dessutom lyssnar jag på repeater R5 större delen av dagarna.

Till sist: "If anything can go wrong it will" fortsättningen på samma citat är " , in a way that most possible damage will be the result". ..

Men oftast är det enkla saker som löses med lite grubblande, om det går trögt kan man finna ändlösa förslag till lösningar på nätet, sålla hårt och prova det som verkar ha trovärdig källa och verkar enkelt. Behåll möjligheten att backa ett steg och försöka med ny idé.

Jag kan ha gett otydlig info, då tar jag gärna skäll, men allt är på din egen risk (även skället ;) ).

Bästa hälsningar,

Alf, SM2GCT