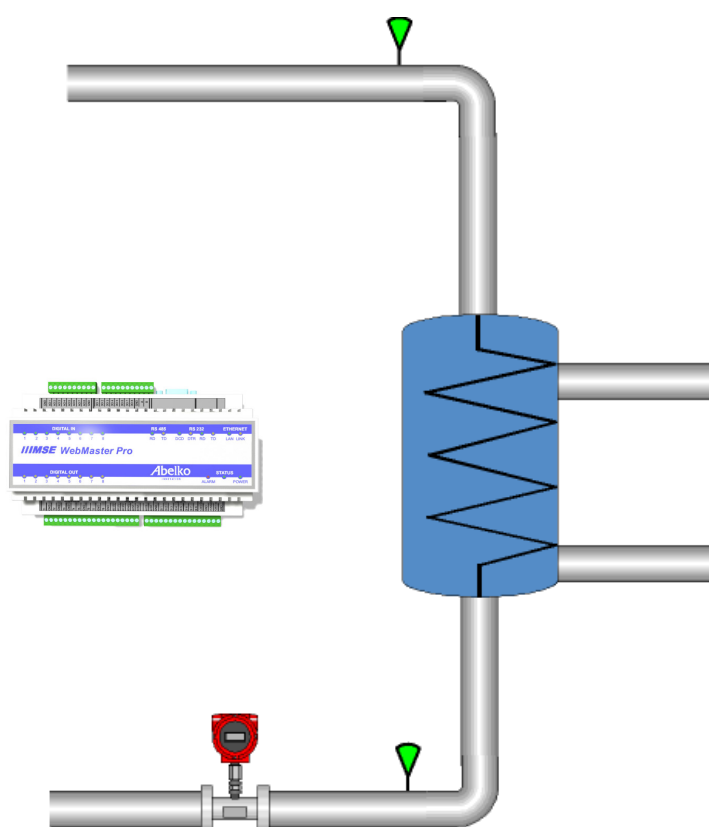


IMSE WebMaster Pro

Effekt och Energi

Implementation av integrationsverk i WMPPro



Document title		
Effekt och Energi - Integrationsverk i WMPPro		
Document Identity		Date
4655-020-00		2006-10-17
Valid for	Firmware version	Webpages version
IMSE WebMaster Pro	2.00	2.00

1. Sammanfattning

Om man mäter temperaturen på vattnet man skickar in i en ”process”, och temperaturen på vattnet som kommer tillbaka, samtidigt som man mäter flödet, så kan man räkna ut den effekt och den energi som avges eller genereras. Det generella ordet process betyder i praktiken oftast en värmeväxlare eller en panna.

Det finns speciella mätare, värmemängdsmätare eller integrationsverk, som sköter om den uppgiften. Om man inte ska använda energimätningen som underlag för debitering så kan man i stället låta en WMPPro göra dessa beräkningar.

I detta applikationsexempel visar vi på hur man dels räknar ut en effekt, och dels hur man mäter energi. Att räkna ut effekten är ganska enkelt, men att få till en väl fungerande energiräknare är lite mer komplicerat. Lösningen implementeras med skript, och detta förklaras i detalj.

Om man bara vill ha funktionen, utan att vara intresserad av detaljerna om hur och varför det fungerar som det gör, så kan man ladda ner och använda skriptet i färdig form från supportsidorna. Avsnitt 5 beskriver hur man får det att fungera.

2. Förutsättningar

För att göra beräkningar av energi och effekt så behöver man veta framledningstemperatur, returtemperatur och flöde. Man behöver också känna till det flödande mediets egenskaper. Vi förutsätter här att det flödande mediet är vatten.

I detta exempel utgår vi från att flödesmätaren lämnar pulser och är kopplat till någon av digitalingångarna DI1 till DI4. Flödesingången ska vara rätt skalad så att flödet anges i liter per sekund. Räknaringången skall också vara konfigurerad för flödesgivaren. Den ska vara skalad så att den räknar liter vatten.

Temperaturkanalerna ska mäta temperaturen i °C (eller möjligen Kelvin).

3. Att mäta effekt

3.1. Den enkla fysiken

Effekt är energi per tidsenhet. En Watt är en Joule per sekund. För vatten vet man att dess specifika värmekapacitet är 4.18 kJ/kgK (kilojoule per kilo och Kelvin). Att värma upp ett kilo vatten en grad kräver 4.18 kJ. För att göra det på en sekund krävs 4.18 kW.

En liter vatten väger i stort sett ett kilo. Visserligen påverkar temperaturen och en del andra faktorer densiteten, men dessa kan ignoreras utan större förlust i precision.

När vi vet flödet i liter per sekund och hur mycket vattnets temperatur har ändrats så är beräkningen enkel.

Avgiven effekt [kW] = (framledningstemperatur [°C] - returtemperatur [°C]) * 4.18 * flödet [l/s]

I skriptform ser beräkningen ut som

```
TempDiff := TempFram - TempRet;  
Power <- TempDiff * 4.18 * Flow;
```

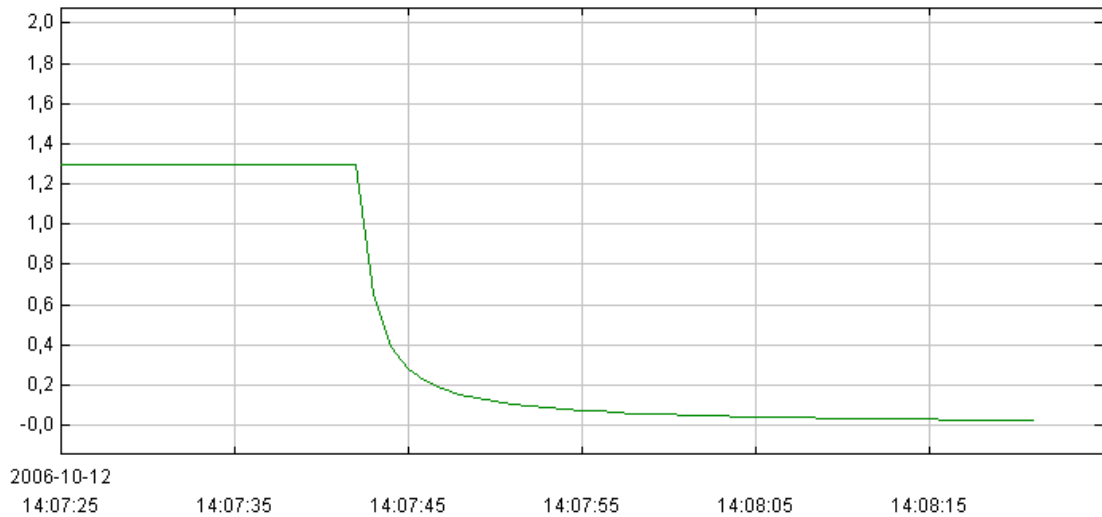
3.2. Verklighetens bieffekter

Det är inget fel på beräkningen ovan, men det finns vissa bieffekter man bör känna till. När man mäter flödet med en flödesgivare med pulsutgång, så känner man inte till flödet i varje given tidpunkt. Det är endast när det kommer en puls som man vet flödet. För att vara exakt så är det det genomsnittliga flödet sedan föregående puls som man får kännedom om.

När flödesgivaren ger pulser ofta, kanske flera gånger per sekund, så är det inget problem. Om flödet sedan plötsligt stängs av, så kommer inga pulser alls. Följaktligen får man heller inte veta vad flödet är. Det en WMPPro gör då, är att den först antar att flödet är konstant. Om flödet var konstant så skulle det komma en puls efter en viss tid. När det inte gör det så antar WMPron varje sekund att den puls den väntar på antagligen kommer alldeles strax. Det flöde den visar blir alltså inte noll på en gång när flödet upphör, utan sjunker först snabbt och sedan allt långsammare ner mot noll. Egentligen skulle det aldrig bli riktigt noll, men av numeriska skäl så blir det noll efter någon timme.

Detta får till följd att det också kommer att se ut som det levereras eller genereras en liten effekt, även när flödet är noll. Man kan säga att den visade effekten är antingen rätt, eller för hög.

Ju högre pulsfrekvens man har från flödesgivaren desto mindre blir denna effekt.



Databasplotten ovan visar hur det ser ut när man plötsligt stänger av ett flöde som genererat pulser med en frekvens strax över en Hertz. En loggad effektkurva skulle följa samma mönster. Den sjunker först fort, men kommer inte riktigt ner till noll.

I system där man har flöden som inte varierar snabbt så är detta inget problem alls, men om man mäter på till exempel tappvarmvatten så blir det tydligt.

4. Att mäta Energi

4.1. Den enkla fysiken, något tillkrånglad

Eftersom effekt är energi per tidsenhet, så är det teoretiskt sett bara att summera energin varje sekund för att få en energiräknare. Om man summerar varje sekunds kW-värde så får man en energiräknare med enheten kJ. För att få det mer bekanta kWh så får man dela med 3600.

Eftersom verkligheten ställde till det lite med effekt mätningen så kan man inte bara summera dessa värden. Det skulle på grund av tidigare nämnda bieffekter ge ett alldeles för högt värde.

För att mäta energin så använder man i stället räknarkanalerna, som ska vara konfigurerade för att räkna vattenmängden som flödar genom flödesmätaren i kubikmeter vatten. Varje gång räknaren ändrar värde räknas motsvarande energiförändring ut, genom att multiplicera med temperaturdifferens och värmekapacitet. Om det är långt mellan pulserna så hanteras en puls i taget. De sekunder då ingen puls har kommit in så har inte räknaren ändrats.

```
IF LastCounter < Counter THEN
  SecInstant <- 1000 *(Counter - LastCounter)
               * TempDiff * 4.18 / 3600;
  LastCounter := Counter;
ENDIF;
```

Det energivärde man får varje sekund sparar man till en kanal, i skriptet kallat SecInstant. Man kopplar sedan en annan kanal till denna kanal, och använder den matematiska funktionen Sum, för att summera alla små energibidrag och få en energimätare.

Om man har en flödesgivare som ger få pulser, och samtidigt har ett system där temperaturdifferensen ändras fort, så kan man få problem med noggrannheten. Algoritmen ovan antar nämligen att temperaturdifferensen varit konstant mellan två pulser från flödesgivaren.

4.2. Numeriska problem och lösningar

I WMPPro används flyttal för att representera numeriska värden. Vid beräkningar används flyttal med dubbel precision, där talet representeras av 8 byte. För att spara plats i minnet använd enkel precision, dvs 4 byte, när talen ska lagras. Det begränsade antalet byte gör att inte alla tal kan representeras exakt. Tal som inte har en exakt representation avrundas till det tal som kan representeras som ligger närmast. Till exempel så kan inte 0.01 representeras exakt, det blir i stället .009999999776482582092285156250.

Detta spelar normalt inte så stor roll. Man måste oftast gå till sjunde eller åttonde värdesiffran för att hitta en avvikelse. När man håller på att summera saker kan det dock bli problem. Om man till ett stort tal adderar ett jämförelsevis mycket litet tal, så kan det bli så att det närmaste representerbara talet är samma tal som man började med. Med 32 bitars (fyra byte) flyttal så blir till exempel $1 + 0.000000001 = 1$.

Tekniken som är beskriven i avsnitt 4.1, där man lägger/adderar ett nytt värde till totalsumman varje sekund skulle fungera bra i något år eller så. Sedan skulle totalen bli så stor att additionen av små energitillskott skulle försvinna. Totalräknaren skulle så småningom sluta öka helt.

För att komma runt detta så kan man dela upp summeringen i tre delar. Den första summeringen som sker varje sekund avbryter man och nollställer en gång i timmen. Vid timskiftet innan summeringen nollställs så förs värdet över till en tim-impulskanal. En dygns-energimätare införs genom en kanal som summerar tim-impulsvärdena. Tim-impulskanalen är noll, utom just vid timsskiftet. Eftersom när den är skild från noll representerar en hel timmes förbrukning, så är det ett mycket större värde än ensekundsimpulserna, och får inte samma numeriska problem. För att ytterligare förbättra situationen så används denna mätare som en dygnsmätare, som nollställs vid varje dygnskifte. En dygns-impuls summeras av den slutliga totalräknaren. Denna extra utväxling gör att man kan ignorera problemet under apparatens livslängd.

```
ROUTINE Integrationsverk
ALIAS
  TempFram = CHANNEL[1]; %Framledningstemperatur [C]
  TempRet  = CHANNEL[2]; %Returledningstemperatur [C]
  Flow     = CHANNEL[17]; %Flöde [l/s]
  Counter  = CHANNEL[41]; %Förbrukningsräknare [m3]

  Power = CHANNEL[140]; %Effekt
  SecInstant = CHANNEL[141]; %SekundImpuls
  HourInstant = CHANNEL[142]; %TimImpuls
  DayInstant = CHANNEL[143]; %DygnsImpuls
  SumHour = CHANNEL[144]; %Tim-mätare
  SumDay = CHANNEL[145]; %Dygns-mätare
VAR
  TempDiff;
  LastCounter;
BEGIN
  TempDiff := TempFram - TempRet;
  SecInstant <- 0;
  HourInstant <- 0;
  DayInstant <- 0;

  Power <- TempDiff * 4.18 * Flow;

  IF LastCounter = 0 THEN
    LastCounter := Counter;
  ELSIF LastCounter < Counter THEN
    SecInstant <- 1000 * (Counter - LastCounter) *
TempDiff * 4.18 / 3600;
    LastCounter := Counter;
  ENDIF;
  IF (TIME_MINUTE = 59) AND (TIME_SEC = 57) THEN
    HourInstant <- SumHour;
    RESET(SumHour);
    IF TIME_HOUR = 23 THEN
      DayInstant <- SumDay;
      RESET(SumDay);
    ENDIF;
  ENDIF;
END;
```

Som bonus med den här metoden får man även timsvisa och dygnsvisa energimätningar. Dessa kan lämpligen sparas direkt i tims- respektive dygnsdatabaserna. För att rätt värde ska lagras i databasen måste dock tidpunkten för återställning av räknarna justeras. Databaserna uppdateras en sekund före tims- respektive dygnskifte. Återställningen bör därför ske tre sekunder före, för att rätt värden ska lagras.

4.3. Förklaring av skriptet

4.3.1. Alias-sektionen

De fyra aliaskanalerna TempFram, TempRet, Flow och Counter är kanalerna för de tre givarna. Counter är räknaringången för flödesgivaren.

Power är kanalen dit den beräknade effekten skrivs. SecInstant, HourInstant och DayInstant är kanaler för impulsvärden. SumHour är en kanal som är kopplad till kanalen SecInstant med matematikfunktionen Sum. SumDay är en kanal som är kopplad till och summerar kanalen HourInstant. Kanalen för totalräknaren anropas aldrig från skriptet, men den ska vara kopplad till och summera DayInstant.

4.3.2. Variablerna

TempDiff är en variabel som används för att hålla reda på den beräknade temperaturdifferensen. LastCounter-variabeln håller reda på föregående värde på räknarkanalen Counter.

4.3.3. BEGIN

De första raderna skriptkod bör inte vara svåra att förstå. Beräkningen av effekt är precis den rad som beskrevs i 3.1.

IF LastCounter = 0 är sann första gången skriptet körs efter en omstart. LastCounter tilldelas då aktuellt värde.

Annars görs en kontroll av att räknaren har ökat. Om den har ökat så beräknas ett nytt SekInstant-värde. Om räknaren inte förändrats så görs ingenting.

Sedan kommer en if-sats som kontrollerar klockan. Om det är exakt tre sekunder till ett tidskifte så uppdateras kanalen HourInstant, och därmed också den dygnsvisa räknaren. Den timsvisa räknaren nollställs. Om tidskiftet också kommer att vara ett dygnskifte så uppdateras även DayInstant och därmed totalräknaren. Den dygnsvisa räknaren nollställs samtidigt.

4.3.4. Vad händer om

Uppdateringen av instantvärdena och dygns och totalräknare sker bara vid bestämda tidpunkter. Vad händer då om apparaten missar en sådan tidpunkt på grund av att den är avslagen eller startar om just då? Jo, den kommer då också att missa att nollställa den räknarkanal den hämtar värdet från. Dessa sparas då till nästa tidpunkt, och den dagen eller den timmen kommer då att få även de värden som inte kunde räknas upp då apparaten inte var i drift. Totalräknaren blir alltså ändå rätt, förutom att apparaten naturligtvis missar alla pulser som kommer medan den är avslagen eller håller på att starta om.

Man måste dock se till att apparaten inte startar om precis vid varje dygnskifte, så att tidpunkten aldrig inträffar medan apparaten är i drift. Det är något att tänka på till exempel om man använder självövervakningsfunktionen i kommunikationsmenyn.

Vid strömavbrott och andra okontrollerade omstarter kan inte WMPron lagra undan räknarvärdena innan de går förlorade. Man kan på det viset förlora information om en hel timmes förbrukning, och till och med ett helt dygn om man har otur.

4.4. Nollställning och ställning av räkneverk

Det som saknas i skriptet ovan är en möjlighet att nollställa totalräknaren, eller att ställa den till något annat specifikt värde. Det går inte att göra detta på något smidigt sätt direkt i kanalen, så därför löser vi det med ytterligare en kanal och lite mer skriptkod.

I skriptspråket kan man använda funktionen RESET för att nollställa en kanal. Den används redan för tim och dygnsräkarna. För att sätta ett specifikt värde till en summerande kanal så kan man först nollställa den, och sedan ge den en impuls med önskat värde.

Vi skapar en kanal med namnet "Ställ Mätare". Vid uppstart ställer vi kanalens värde till -1. -1 används som en markering att mätarvärdet inte ska påverkas. Sedan gör vi en if-sats som kontrollerar om kanalen har något annat värde (är noll eller större). Om den har det så är det en indikering på att mätarvärdet ska ställas. Alla tre mätarkanalerna nollställs med RESET, och sedan sätts DayInstant till kanalens värde. Totalräknaren kommer därmed att få detta värde. För att det inte ska upprepas så ställs kanalen "Ställ Mätare" till -1 igen.

Efter tilläggen ser skriptet ut som nedan:

```
ROUTINE Integrationsverk
ALIAS
  TempFram = CHANNEL[1]; %Framledningstemperatur [C]
  TempRet  = CHANNEL[2]; %Returledningstemperatur [C]
  Flow     = CHANNEL[17]; %Flöde [l/s]
  Counter  = CHANNEL[41]; %Förbrukningsräknare [m3]

  Power = CHANNEL[140]; %Effekt
  SecInstant = CHANNEL[141]; %SekundImpuls
  HourInstant = CHANNEL[142]; %TimImpuls
  DayInstant = CHANNEL[143]; %DygnsImpuls
  SumHour = CHANNEL[144]; %Energimätare Timme
  SumDay = CHANNEL[145]; %Energimätare Dygn
  SumTot = CHANNEL[146]; %Energimätare Total
  SetMeter = CHANNEL[147]; %Ställ Räknare
VAR
  TempDiff;
  LastCounter;
  LastHourCounter;
  LastDayCounter;
BEGIN
  TempDiff := TempFram - TempRet;
  SecInstant <- 0;
  HourInstant <- 0;
  DayInstant <- 0;

  Power <- TempDiff * 4.18 * Flow;

  IF LastCounter = 0 THEN
    LastCounter := Counter;
    LastHourCounter := Counter;
    LastDayCounter := Counter;
```



```
    SetMeter <- -1;
  ELSIF LastCounter < Counter THEN
    SecInstant <- 1000 * (Counter - LastCounter) *
TempDiff * 4.18 / 3600;
    LastCounter := Counter;
  ENDIF;
  IF (TIME_MINUTE = 59) AND (TIME_SEC = 57) THEN
    HourInstant <- SumHour;
    RESET(SumHour);
    IF TIME_HOUR = 23 THEN
      DayInstant <- SumDay;
      RESET(SumDay);
    ENDIF;
  ENDIF;

  IF SetMeter >= 0 THEN
    RESET(SumTot);
    RESET(SumDay);
    RESET(SumHour);
    DayInstant <- SetMeter;
    SetMeter <- -1;
  ENDIF;
END;
```

5. Att använda skriptet

5.1. Konfigurera kanaler

För att skriptet ska fungera så måste också kanalerna vara rätt konfigurerade. Förutom givarkanalerna så behövs det åtta kanaler till skriptet. Här har kanal 140 till 147 använts.

140	Effekt	ingen 1	0.0 kW	ingen
141	SekundImpuls	ingen 1	0.000 kWh	ingen
142	TimImpuls	ingen 1	0.0 kWh	ingen
143	DygnsImpuls	ingen 1	0 kWh	ingen
144	Energimätare Timme	Kanal 141	0.0 kWh	Summa
145	Energimätare Dygn	Kanal 142	0 kWh	Summa
146	Energimätare Total	Kanal 143	0 kWh	Summa
147	Ställ Mätare	ingen 1	-1 kWh	ingen

De tre mätarkanalerna (144-146) är kopplade till var sin impulskanal (141-143). De ska ha matematisk funktionen summa, med skalfaktor ett.

Redigera kanalnummer 146

Kanalnamn	Energimätare Total
Kanalvärde	0
Kanalenhet	kWh
Antal decimaler	0
Faktor	1
Offset	0
Typ av koppling	Kanal ▼
Kopplad till nummer	143
Matematikfunktion	Summa ▼
	0
Faktor	1
Maxgräns	0

Avbryt
Radera
OK

Övriga kanaler behöver man endast ställa namn, enhet och antal decimaler för. De ska inte ha någon koppling eller matematisk funktion.

5.1.1. Ladda upp en färdig konfiguration

Om kanal 140 till 147 är lediga och oanvända i den apparat man ska använda skriptet så kan man ladda in kanalinställningarna från filen kanaler.par.

System

Information
Filhantering

Presentation

Lösenord

Filhantering

Init

Ladda upp fil (från PC)

Parameterbank ▼

Ladda upp fil (från PC)

Ladda ner fil (till PC)

Status

UPLOAD FILE kanaler.par TO putpar.par

DONE

Filen kanaler.par kan hämtas från Abelkos hemsida under fliken support, och den skall laddas upp till Parameterbank med filhanteringsverktyget under systemmenyn. Om kanalerna redan används så skrivs de över.

5.2. Ladda upp och konfigurera snippet-filen

Skriptet i detta applikationsexempel finns i filen integrationsverk.gpss. För att ladda in snippeten i en apparat så bör man först konfigurera givarna och de övriga kanalerna. Sedan går man till menyn Inställningar / Avancerat / Script. Klicka på Snippets och sedan Insert from file. Sök rätt på integrationsverk.gpss och klicka på öppna.

Det dyker upp en ny rad i fönstret till höger med titeln ROUTINE Integrationsverk. Klicka på den raden för att se de kanaler skriptet är kopplat till.

Alias	Type	Description	Connection
TempFram	CHANNEL	%Framledningstemperatur [C]	Temp 1
TempRet	CHANNEL	%Returledningstemperatur [C]	Temp 2
Flow	CHANNEL	%Flöde [l/s]	Digital in 1
Counter	CHANNEL	%Förbrukningsräknare [m3]	Räknare digital in 1
Power	CHANNEL	%Effekt	Effekt
SecInstant	CHANNEL	%Sekundimpuls	Sekundimpuls
HourInstant	CHANNEL	%Timimpuls	Timimpuls
DayInstant	CHANNEL	%Dygnsimpuls	Dygnsimpuls
SumHour	CHANNEL	%Energimätare Timme	Energimätare Timme
SumDay	CHANNEL	%Energimätare Dygn	Energimätare Dygn
SumTot	CHANNEL	%Energimätare Total	Energimätare Total
SetMeter	CHANNEL	%Ställ Räknare	Ställ Mätare

Kolumnen längst till höger visar namnen på kanalerna i apparaten. Genom att klicka på ett namn så fälls en lista med alla kanaler ut. Där kan man ändra vilken kanal som skriptet ska använda. De fyra första kanalerna är det högst troligt att man måste ställa om. De är kanalerna för givarna.

Resterande kanaler är kanalerna 140 till 147 i exemplet. Har man konfigurerat dessa kanaler på andra kanalnummer så måste man ändra även dessa, men har man använt kanaler.par så är de rätt från början.

När alla kanaler är rätt, klicka på Apply. Snippetsfönstret stängs då och skriptrutinen infogas i skriptfilen. Tryck sedan på Spara för att spara skriptet i apparaten. Apparaten startar då om.

5.3. Steg för steg-anvisning

- Hämta den tillhörande Zip filen från supportsidan och packa upp den.
- Konfigurera givaren för framledningstemperatur så att temperaturen visas i °C.
- Konfigurera givaren för returtemperatur så att temperaturen visas i °C.
- Konfigurera flödesgivaren så att flödet visas i liter per sekund.
- Konfigurera flödesmätarens räknaringång så att den räknar i kubikmeter.
- Ladda upp kanaler.par eller konfigurera skriptets kanaler manuellt enligt 5.1.
- Ladda upp och konfigurera integrationsverk.gpss enligt 5.2
- Kontrollera att det fungerar och ger rimliga resultat.

5.4. Tänk på följande

Låg pulsfrekvens från flödesgivaren ger sämre noggrannhet. Skriptet förutsätter att temperaturdifferensen kan betraktas som konstant mellan två pulser.

Om strömmen bryts till apparaten så kan upp till en timmes förbrukningsinformation gå förlorad, eller ett helt dygn om man har otur.

Effektmätningen kommer att visa ett för högt värde när flödet sjunker snabbt. Det är speciellt tydligt om flödet plötsligt blir noll.