

Node-RED, som ersättare för Condilink



Node-RED är ett flödesbaserat utvecklingsverktyg för visuell programmering som ursprungligen utvecklats av IBM för att koppla ihop hårdvaruenheter, API: er och onlinetjänster som en del av sakernas internet. 2016 bidrog IBM med Node-RED som ett JS Foundation-projekt med öppen källkod. Node-RED tillhandahåller en webbläsarbaserad flödesredigerare som kan användas för att skapa JavaScript-funktioner. Element av applikationer kan sparas eller delas för återanvändning. Exekvering bygger på Node.js. Flöden som skapas i Node-RED lagras med JSON.

Installation

Just nu har vi ingen specialanpassad installationsfil eller liknande för *Node-RED*. Se istället *Node-RED*:s officiella dokumentation för installationsanvisningar:

<https://nodered.org/docs/getting-started/installation>

Med det sagt så går installationen i enklaste fall till så här:

- Installera NodeJS (JavaScript-tolk) från <https://nodejs.org/>. Används gärna en LTS-utgåva (Long Term Support). När jag skriver detta är den senaste LTS-utgåvan version 8.10.0 och vi har verifierat att den fungerar bra.
- Starta en kommandotolk ("DOS-fönster"). När NodeJS installeras så läggs det till en genväg i Windows startmeny som har Node med i sökvägen.
- Kör följande kommando för att installera Node-RED: `npm install -g node-red`
- Starta Node-RED med kommandot: `node-red`
Node-RED kör nu i bakgrunden. För att göra inställningar, starta webbläsaren och gå till adress <http://localhost:1880>
- Klicka på hamburgermenyn uppe till höger, välj **Manage Palette**
- Klicka på fliken "Install" och sök på "cmteknik". En träff ska hittas (node-red-contrib-cmteknik), välj att installera den.

Autostart

Autostart är en funktion som vi måste använda. Man kan använda t.ex. Windows *Autostart* som vi alltid gjort. Det är dock lite mer komplicerat i *Windows 10*.

Kör som en service i bakgrunden

Ett stort problem är att kommandotolken visas i förgrunden och att den säkerligen kommer att stängas ned av användaren av misstag eller inte misstag

Vi behöver köra *Node-RED* som en service i bakgrunden. Det finns en mjukvara kallad *NSSM* (The Non-Sucking Service Manager) *NSSM* är en generell mjukvara för att köra *Windows*-applikationer som *NT*-tjänster.

Installera nssm

Gå till <https://nssm.cc/download>

Ladda ner *nssm2.24* och packa upp i en lämplig mapp.

Skapa en service

Vi använder *nssm* för att skapa servicen.

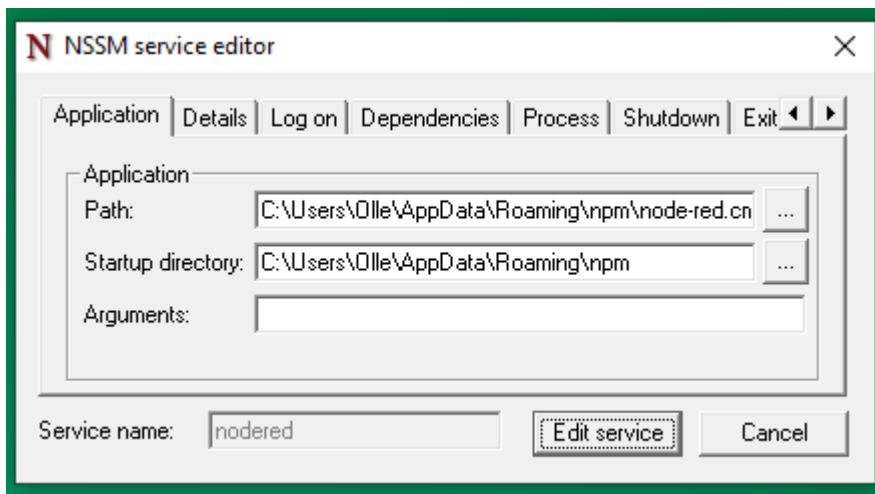
Öppna kommandotolken som administratör (högerklicka på cmd och välj "Starta som administratör").

Gå till mappen där nssm är installerat, t ex C:\Users\Olle\Documnets\Download\nssm-2.24\Win64>

Vi kallar tjänsten för något klatschigt, t ex *nodered*

Kör C:\...\nssm-2.24\Win64>nssm install nodered (t.ex.)

Nu kommer en grafisk dialogruta upp:



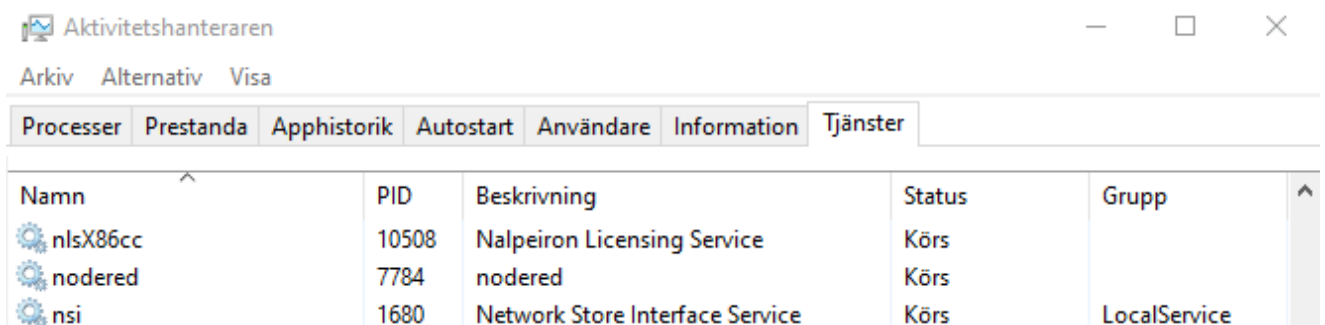
Vid "Path" skriv in sökvägen till node-red, t.ex. C:\...\AppData\Roaming\npm\node-red.cmd. Om osäkerhet råder, skriv "Where node-red" i kommandotolken och kopiera svaret till "Path".

Startup directory är normalt samma som Path.

Nu har vi alltså talat om för servicen, *nodered*, hur den ska hitta applikationen, *node-red*. Som default är nssm inställd på att starta automatiskt. Bra så. Tryck på knappen "Install service" ("Edit service" heter den när man öppnat den igen för att ändra något) för att avsluta. För att editera NSSM skriver man "nssm edit nodered" i kommandotolken..

Vi kontrollerar i aktivitetshanteraren:

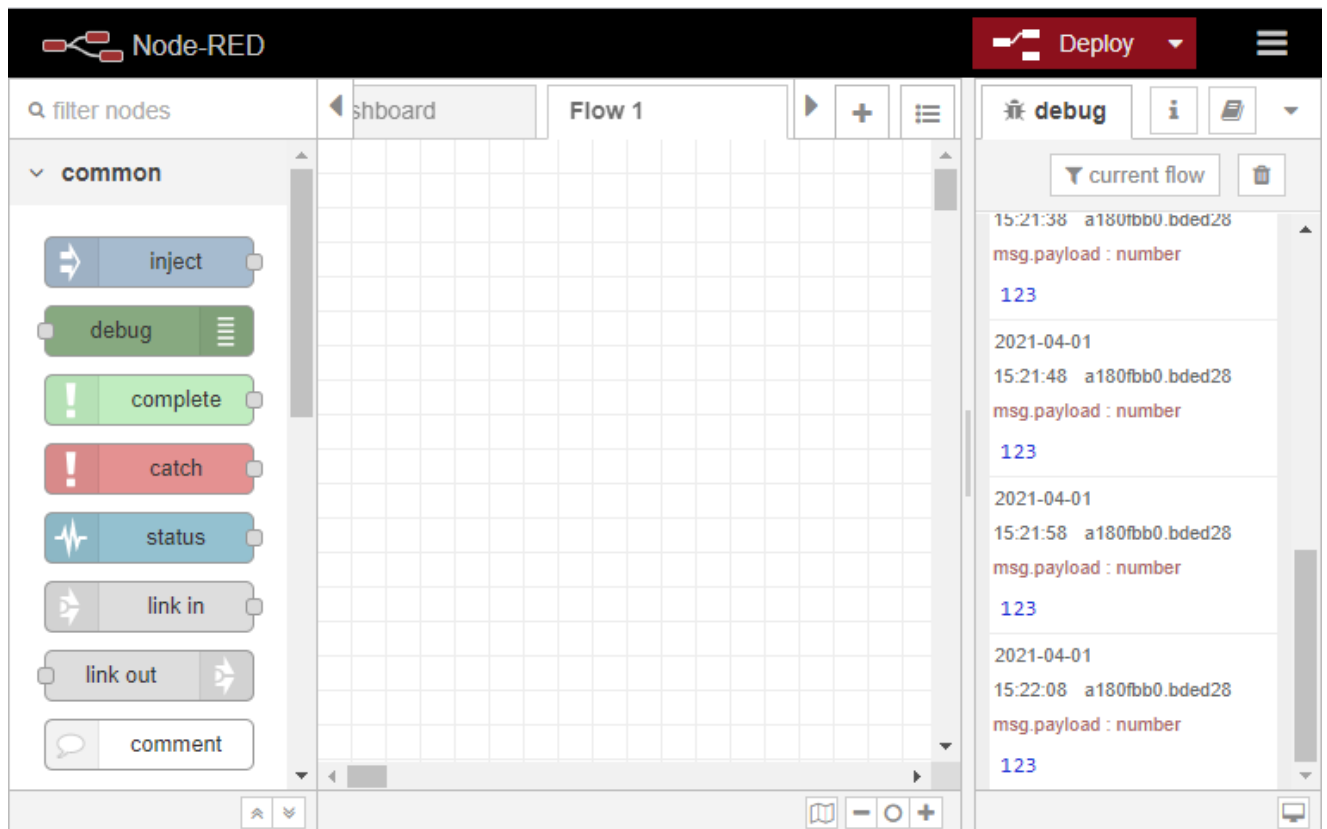
Här ser vi att nodered körs. OK. Nu kan vi starta node-red i webbläsaren (http://localhost:1880) som angivits tidigare.



Det finns en risk att processen hänger sig om man editerar ett befintligt flöde utan att stoppa processen. Detta kan man göra i aktivitetshanteraren men enklare i webgränssnittet för Node-RED. Se [Observera vid editering av ett flöde](#): på sidan 10.

Arbetsytan

Arbetsytan, ser ut så här i förminskad storlek.



Till vänster finns ett fält, paletten, med *noder*. *Noder* är de avrundade rektanglarna i olika färger. Man kan kopiera dem till arbetsytan (det tomma fältet i mitten) genom att klicka-och-dra. Det finns en stor mängd noder med olika funktion.

I mitten hittar man arbetsytan där noderna kan arrangeras enligt egen önskan. Man kan ha flera arbetsytor, var och en med en egen, namngiven, flik. Det som händer på arbetsytan när den driftsatts kallas för ett *flöde* (*Flow*). Flikarnas namn är därför default Flow 1, Flow 2 osv. För att lägga till ett nytt flöde, klickar jag på plustecknet överst till höger. Man kan ändra namn på flödet, fliken, genom att dubbelklicka på den och skriva ett passande namn.

Till höger finns ett fält som kan innehålla olika saker, t ex ett debug-fönster som i exemplet ovan.

Mitt första flöde

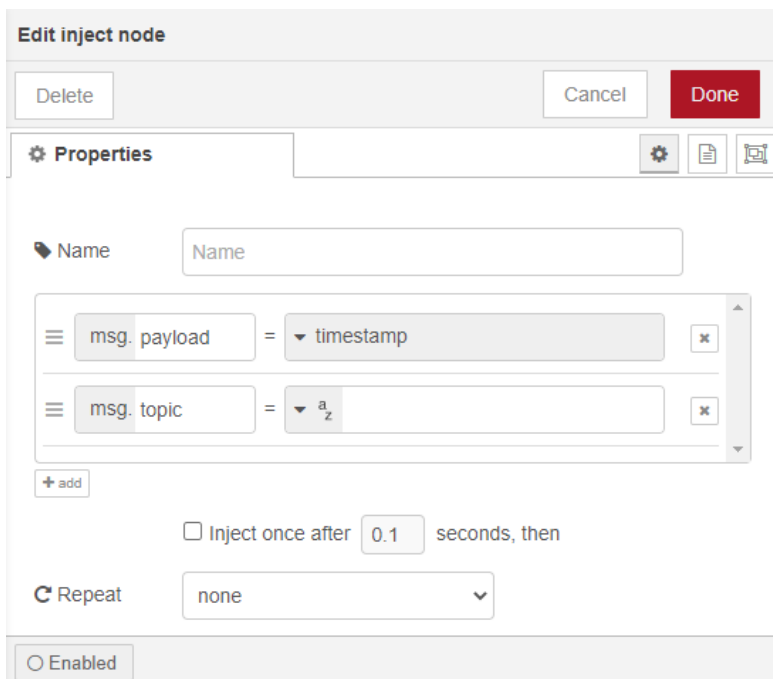
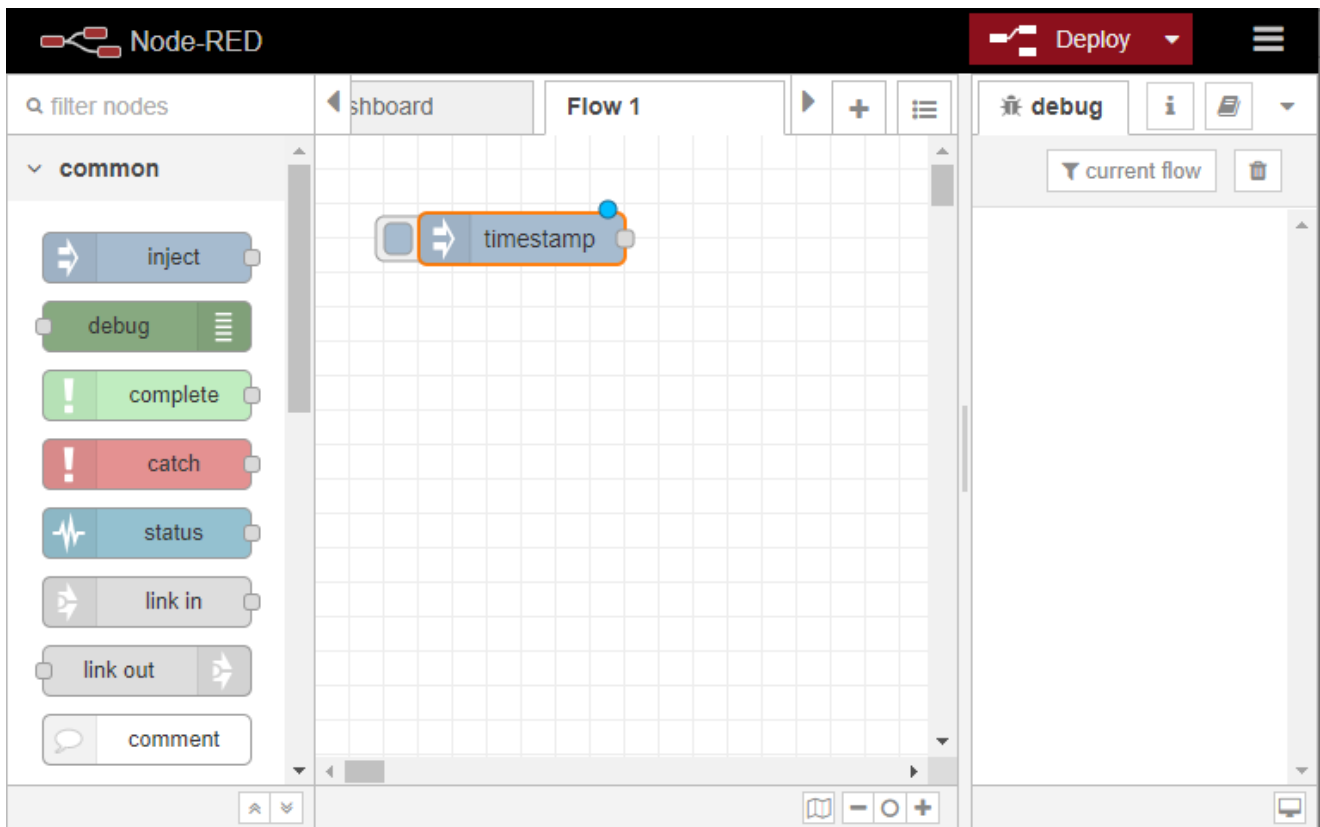
Gör ett mycket enkelt flöde för att bekanta dig med hur det fungerar.

Funktionen ska bli:

Ett värde som angivits manuellt ska skrivas till debug-fönstret varje gång man trycker på en knapp.

Överst bland noderna finn en som heter *inject*. Värdet ska ju injiceras i flödet så välj *inject*-noden.

Klicka på *inject*-noden och flytta den till arbetsytan genom klicka-och-dra. Namnet eller titeln på noden ändrar sig till *timestamp* (tidsangivelse) vilket är defaultfunktionen för *inject*-noden. Detta betyder att varje gång noden aktiveras sänder den sitt *innehåll* (meddelande, payload, nyttolast) till *flödet*. Innehållet är ett uttryck för aktuell tid. Det är inte vad som önskas så jag dubbelklickar på noden för att ändra nodens egenskaper.



Först ändrar man innehållets typ från *timestamp* till siffror (*number*) genom att klicka på den minimala nedåtpilen i fältet efter "msg.payload".
Välj *number* från menyn.

Därefter anger man ett tal, 123, i *payload*-fältet.

Sedan trycker man på knappen *Done*.

Nu har "namnet" på *inject*-noden ändrats till 123.

Den blå punkten upptill till höger på noden indikerar att nodens funktion inte är driftsatt ännu.

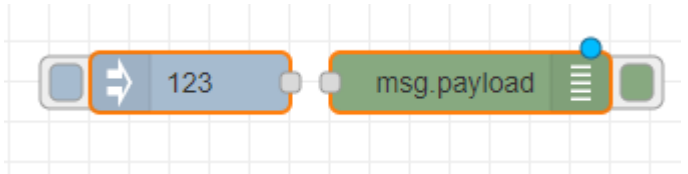
Tryck på *Deploy*-knappen (driftsätt).

Ett meddelande kommer fram med texten "Successfully deployed". Detta betyder att noden är tagen i drift.

Den orangea tråden runt noden indikerar att noden befinner sig i fokus. Klicka någonstans på arbetsytan för att flytta fokus från noden.

Till vänster om noden finns en knappliknande symbol. Varje gång man trycker på denna knapp injicerar noden sin nyttolast till flödet. Men det finns ännu inget flöde att injicera i. Avsikten var ju att skriva nyttolasten, meddelandet, i debug-fönstret. För att göra detta väljer man, naturligt nog, *debug*-noden. Man använder samma procedur som för *inject*-noden.

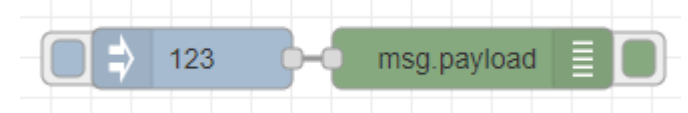
Placera *debug*-noden till höger om *inject*-noden. Standardfunktionen för *debug*-noden är att skicka meddelandet till debugfältet, så man behöver inte ändra något.



För att få till ett flöde saknas en förbindelse mellan *inject*- och *debug*-noderna. Tryck på anslutningspunkten, den grå, rundade lilla fyrkanten och dra från denna till anslutningspunkten på *debug*-noden. En tråd framträder som kommer att distribuera flödet från en nod till en annan.

Man kan ha flera anslutningar till varje anslutningspunkt, men i det här fallet räcker det naturligtvis med en. Om man skulle få lust att flytta runt på noderna kan man bara klicka-och-dra till vilken position som helst. Anslutningstråden kommer att anpassa sin form efter detta.

Nu trycker man på *Deploy*-knappen. Efter detta ser flödet ut såhär:



Ta fram debugfönstret genom att trycka på den lilla "bug"-symbolen till höger och välj "current flow" för att se vad som händer i det aktuella flödet.

Tryck på *inject*-knappen till vänster på *inject*-noden. Nu dyker debug-meddelandet upp i debug-fönstret. Först kommer

en tidsstämpel och ID på noden. Därefter, i rött, slås fast att meddelandet är ett nummer eller siffror, *msg.payload : number*, och slutligen, i blått, visas meddelandet, *123*. Varje gång man trycker på knappen kommer ett nytt meddelande i debugfönstret. Då så.

Mitt andra flöde

Nu har man förstått det mest elementära och det är dags för ett, aningen mer, komplicerat flöde. Den här gången tänker vi kommunicera med en styrdator, en *LCC Completa*. Val av *Completa* beror på att det är ett exempel på styrdatorer för växthus, men de flesta styrdatorerna från *DGT* eller *CM Teknik* fungerar. Kommunikationen ska som vanligt, för *CM Teknik*, ske via *Netcomd*.

En ganska vanlig uppgift för en växthusdator, för t.ex. gurkodling, är att man använder en ganska hög minimum framledningstemperatur på växtrören (de värmerör som är placerade lite längre upp, nära plantan) för att öka värmestrålningen på plantan när det är mulet väder eller tidigt på säsongen när solstrålningen är låg. När solen skiner vill man minska minimum framledningstemperatur i motsvarande grad som solstrålningen ökar.

Alltså: **Minska min framledningstemperatur i omvänt förhållande till solinstrålningen.**

Skapa ett nytt flöde genom att trycka på plustecknet t.h. om de andra flikarna. Dubbelklicka sedan på fliken och skriv namnet "LightDepFlow" t. ex.

Skriv en kort beskrivning i fältet för detta så att man enkelt senare minns vad just detta flöde gör.

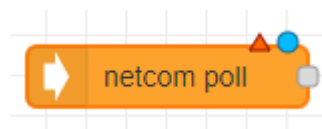
Avsluta med att trycka på *Done*.



Det första som jag behöver göra i flödet är att läsa av aktuell instrålning. För detta finns det en nod som kan läsa från *Netcomd*. Man hittar den långt ner i nod-fältet under *CM Teknik*. Den kallas *netcom poll*.

Denna *nod* kan läsa en enkel parameter från en enhet som är ansluten till *Netcomd* kommunikations-server. Aktuell variant läser en parameter vid en fast intervall. Det finns även en annan variant, *netcom read*, där avläsning kan triggas av en annan nod.

Dra den till arbetsytan och dubbelklicka på den för att ändra nodens egenskaper.



Properties:

Name

Ett läsbart namn på noden. Ej obligatoriskt.

Device

Namn på den enhet som noden ska läsa från.

Parameter

Den parameter som ska avläsas. *Datatyp* (flagga) ska också anges i fältet t.h. om parameter. Om ingen datatyp anges används default-datatypen för enheten och nätverkstypen.

Rate

Intervall mellan avläsningar i sekunder.

Connection

Anslutningen, möjligen delad, till *Netcomd*-servern.

Name

Ändra namnet på noden till "Irradiation".

Device

Ange "Completa" som enhet.

Parameter

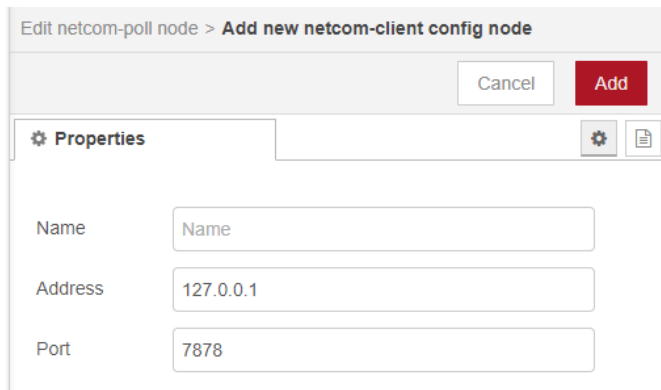
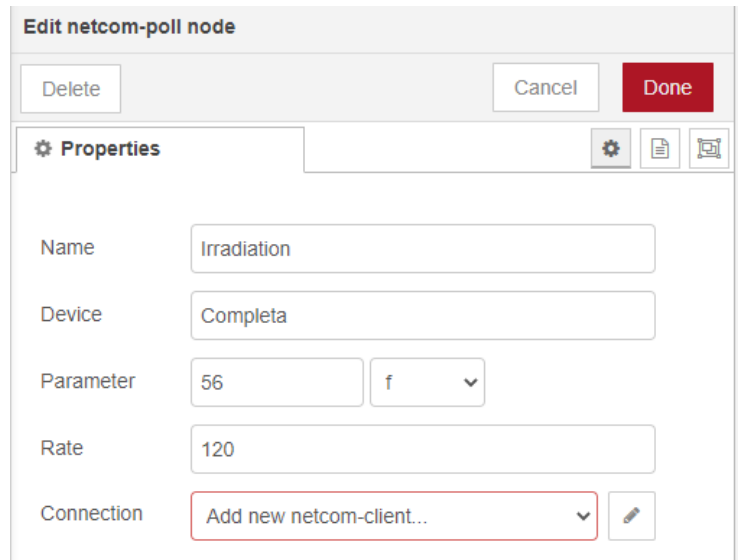
Ange parameter till 56, som är instrålning avläst från väderstationen via LCC Completa i W/m².

Datotyp

Välj "f" som står för flyttal (decimaltal).

Rate

Sätt avläsningsintervallet till 120 sekunder. Därmed kommer noden att läsa av aktuell instrålning varannan minut.



Klicka nu på penn-symbolen nederst till höger. Då kommer "Add new netcom-client"-dialogrutan upp.

Name

Skriv in namnet "Completa".

Address

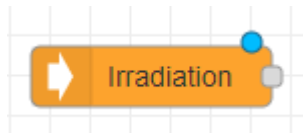
Standardadressen till Netcomd är 127.0.0.1. Ändra inte denna om det inte finns någon speciell anledning till det.

Port

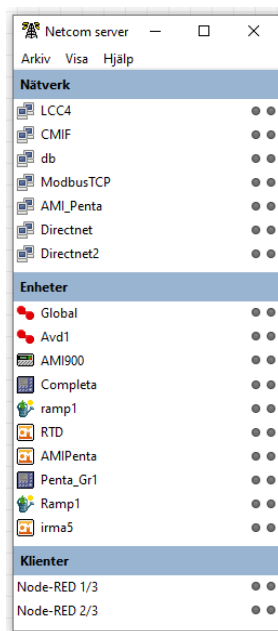
Standardporten är 7878. Behåll även denna.

Tryck på knappen Add.

Då ser netcom poll-noden ut så här:



Den blå punkten betyder att noden ännu inte driftsatts. Man försäkras sig om att Netcomd kör och kommunicerar med styrdatorn (LCC Completa).



Netcomd kan se ut som figuren till vänster.

Tryck på *Deploy*-knappen.

Efter det ser noden ut så här:



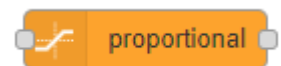
Den blå punkten har försvunnit indikerande att noden är driftsatt (deployed).

Noden har faktisk blivit levande och tickar på som ett ensamt hjärta och hämtar instrålningsdata från väderstationen. T.h. om den gröna punkten står 300, vilket är den senast avlästa instrålningen i W/m².

Det är ju fint, men nu måste något göras som tar hand om datan. Föresatsen var att reducera minimum framledningstemperatur i förhållande till instrålningen.

Lyckligtvis finns det en nod som kan göra just detta!

Inte förvånande kallas denna nod *proportional* och man hittar den bland CM Teknik-noderna.



Noden har en ingång (alltid till vänster eftersom flödesriktningen är från vänster till höger). Den har även en utgång, till höger.

Ingången ska vara instrålning som är meddelandet (nyttolasten) i den redan befintliga noden. Så om man ansluter utgången på den befintliga netcom poll noden (*irradiation*) till ingången på proportional-noden så borde något användbart vara på gång.

Kopiera in *proportional*-noden till arbetsytan.



Det är dags att ta en titt på egenskaperna (*properties*) för *proportional*-noden. Dubbelklicka på den och nedanstående dialogruta kommer fram:

Name: Här kan man välja ett passande namn för noden. Här lämnad tom.

Edit proportional node

Delete Cancel Done

⚙️ Properties 📄 🖨️

Name

Input range to

Output range to

Input range: Här kan man ställa in instrålningsnivåerna mellan vilka värdet på utgången ska vara proportionellt. I det vänstra fältet skriver man 80 (W/m²) och i det högra 400 (W/m²).

Output range: Minimum framledningstemp ska minskas i förhållande till ökad instrålning. Min. framledning uttrycks i °C. Man önskar en lägsta framledningstemperatur på växtrören på 50°C när solens instrålning är svag, t.ex. i

regniga förhållanden eller tidigt på säsongen för att upprätthålla plantans aktivitet, främst gasutbytet. En temperatur på 50°C ger en viss strålningsvärme mot plantorna när solen är borta. När det är soligt däremot ska min. framledning reduceras eftersom plantorna får en större eller mindre andel strålning från solen, och det kan leda till plantstress om man fortsätter brasa på med rörvärme. Och det kostar. Så vid 400 W/m² vill man att framledningstemperaturen ska ligga nära lufttemperaturen, säg 20°C. För att uppnå detta sätter man 50°C i den vänstra boxen och 20°C i den högra. Nu ser det ut såhär:

Edit proportional node

Delete Cancel Done

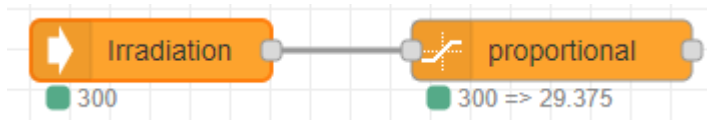
⚙️ Properties 📄 🖨️

Name

Input range to

Output range to

Man drar en tråd mellan de två noderna och trycker på *deploy*. Efter den inställda intervallen, 120 sekunder, mottar den proportionella noden meddelandet från *netcom poll*-noden och skalerar meddelandet enligt inställningarna. Under den proportionella noden t.h. om den gröna punkten kan man läsa ingångsvärdet (300) och utgångsvärdet (29.375). Bra.



Algoritmen i den proportionella noden gör exakt vad den förväntas göra. Detta kan medföra ett stort antal decimaler i utgångsmeddelandet. Det gör i regel inte så mycket skada men ser fult ut. Så man kan fixa detta och få lite mer erfarenhet hur man gör i de fall antalet decimaler kan ha betydelse för funktionen. Tillför lite mer matematik.

Lyckligtvis finns det en matematik-nod. Den är användbar vid olika slag av enklare matematiska uppgifter. En av användarna har gjort detta bidrag till nodbiblioteket precis som CM Teknik gjort med sina bidrag. Den här kallas *calculator* (node-red-contrib-calc). Jag laddar ner den härifrån:

<https://flows.nodered.org/node/node-red-contrib-calc>

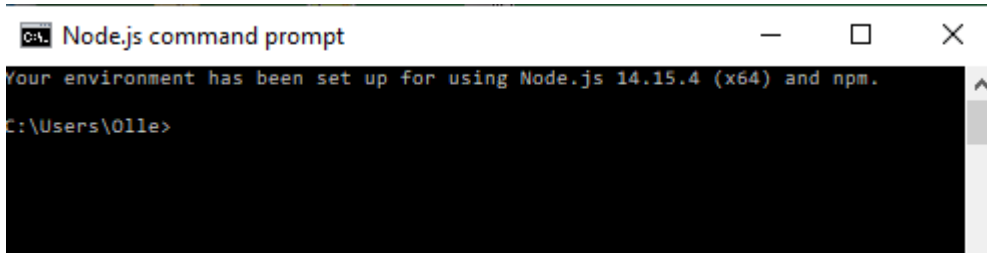
Ladda ner och installera

Kör följande *npm*-kommando i Node-RED-biblioteket, (typiskt `~/node-red`):

```
npm install node-red-contrib-calc
```

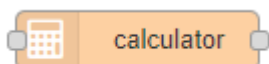
Kopiera texten i det bruna fältet och klistra in det i *Node.js* kommandotolk:

(Man kan hitta *Node.js* kommandotolk genom att ange *cmd* i Windows sökfält).

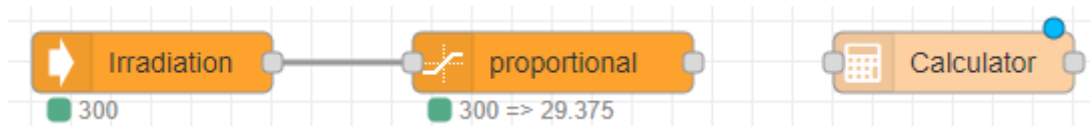


Här är den. Nu klistrar man in texten i *Dos*-fönstret och trycker *Enter*.

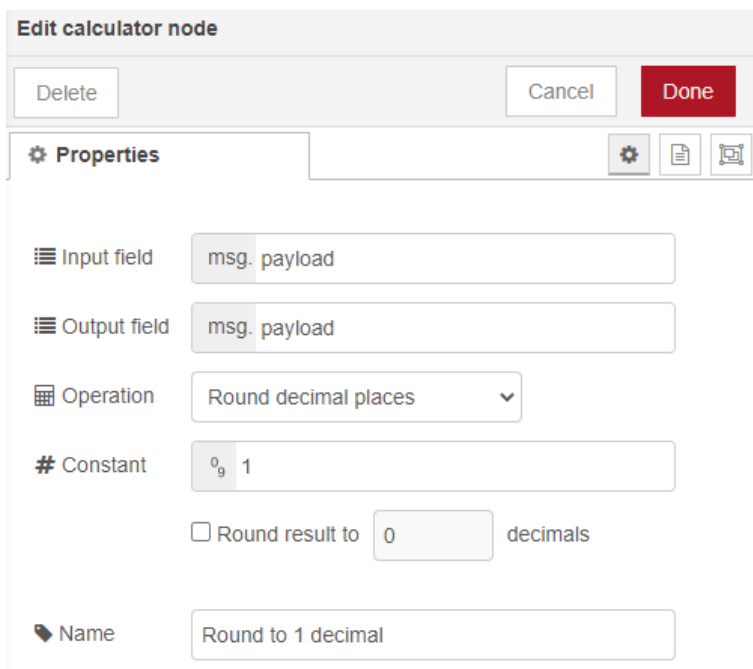
Starta om *Node-RED* (både i *Windows* och *Node-RED*s kommandotolk. Efter omstart hittas den nya noden, kallad *calculator*, i nodfältet till vänster i *paletten* under *function*.



Dra in den och placera den höger om den senaste noden:



Ändra inställningarna för *calculator*-nodens egenskaper genom att dubbelklicka på den.



Under *Operation* finns det många olika funktioner.

Välj "*Round decimal places*" eftersom det är det som önskas.

Ställ in hur många decimaler som önskas, "1", i *Constant*-fältet.

Bocka från i fyrkanten framför "*Round result to*".

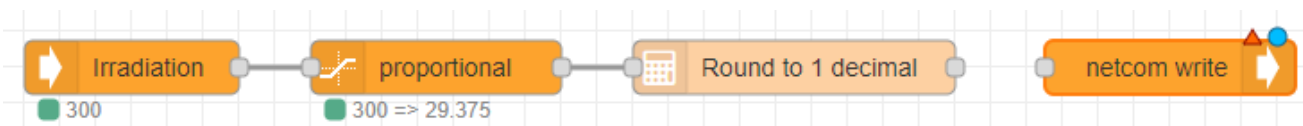
Skriv ett namn på noden, "*Round to 1 decimal*".

Sedan trycker man på *Done*.

Nu ansluter man *proportional*- till *calculator*-noden med en tråd och trycker på *Deploy*. Då ser flödet ut såhär:



Nu återstår det att överföra meddelandet (min. framledningstemperatur på växtrören). För detta använder man naturligt nog noden *netcom write*. Dra den på plats och justera avstånden mellan noderna så allt ser bra ut.



Då ser flödet ut såhär.

Edit netcom-write node

Delete Cancel Done

Properties

Name:

Device:

Parameter:

Connection:

Nu ska egenskaperna editeras i *netcom write*-noden:

Name

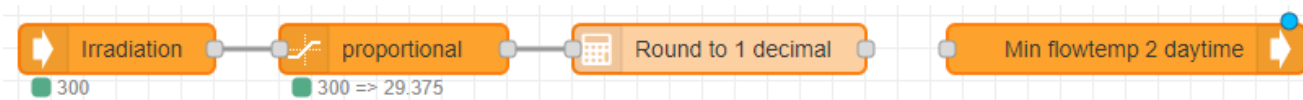
Välj ett förklarande namn, "Min flowtemp 2 daytime"

Device: Completa.

Parameter: 41262, som är parametern för *Minimum framledningstemperatur shunt 2 dagtid*.

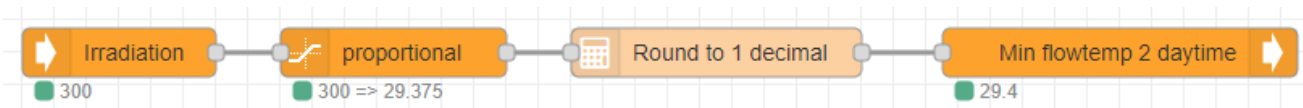
Datotyp/Flag: Ange "f" eftersom datatypen är flyttal.

Connection: Completa.



Tryck *Done*.

Då ser flödet ut såhär.



Slutligen ansluter man *calculator*-noden till *netcom write*-noden och trycker på *Deploy*.

Nu fungerar det. Vid en instrålning på 300 W/m², sätts meddelandet enligt den proportionella funktionen till 29,375°C som därefter avrundas till en decimal, alltså 29.4°C, vilket kan avläsas efter den gröna punkten under *netcom write*-noden. Meddelandet (värdet) sänds till Completan via Netcomd. Klart.

Observera vid editering av ett flöde:

Det finns en risk att processen hänger sig om man editerar ett befintligt flöde utan att stoppa processen. Detta kan man göra i aktivitetshanteraren men enklare i webbgöransnittet för Node-RED.

Välj "Info" i högra fältet. Tryck på ringen t.h. om den aktuella processen. Ett diagonalt streck uppträder i ringen, indikerande att processen är pausad. Gör editeringen. Tryck åter på ringen. Nu kör flödet igen.

info i [document] [trash] [gear] [database]

Search flows

Flows

- > Flow 1 ⊗
- > AMI Penta fast intervall ○
- > AMI34 ○