

Shelly Pro 4PM MQTT setup Remote control and data to Grafana.

My test setup Node-red version 3.0.2 running on a windows 10 64bit PC.



Problem: We use 4 units of Shelly 1PM to send power usage of a small Refrigeration plant. We have 8 plants so due to the massive wifi traffic we often loose a unit or two. So we decided to switch to the Shelly Pro 4PM. That way we can reduce the amount of units from 32 to 8.

Due to massive increase in energy costs we are only allowed to run the plant during school hours so a scheduler was needed. That did complicate the code.

The units are name classroom+unit number+ output:

h106_1sum , h106_1Compressor, h106_1defrost, h106_1evaporator fan.

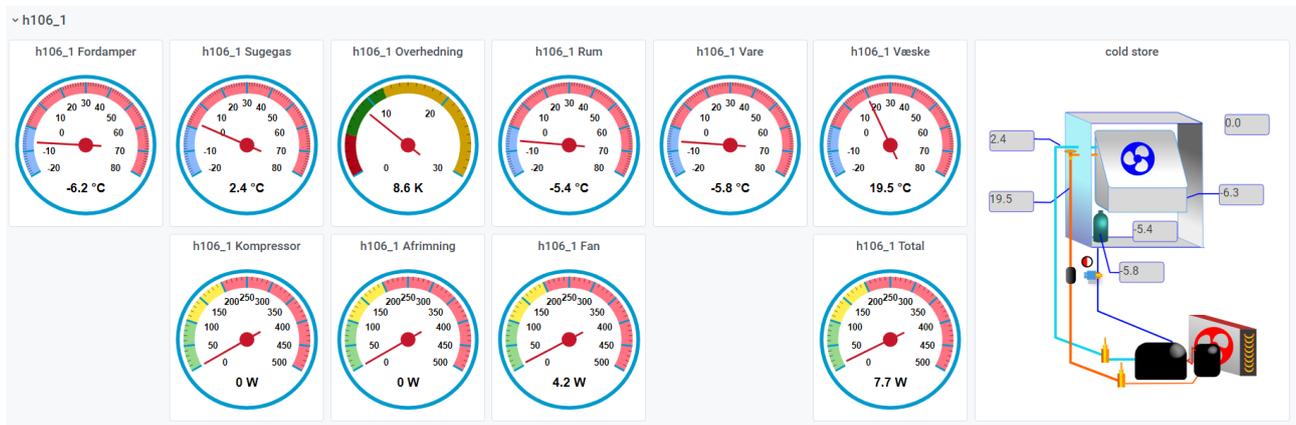
The plant is controlled by a Danfoss ERC 213 Room Controller.



I made this controlbox with 3 socket outlets for Compressor, defrost and evaporator fan.



All the data is sent to Grafana



Note: I am A Danish teacher at a Danish School for Electricians and Refrigeration engineer so many of the titles are in Danish.

The node-red dashboard looks like this

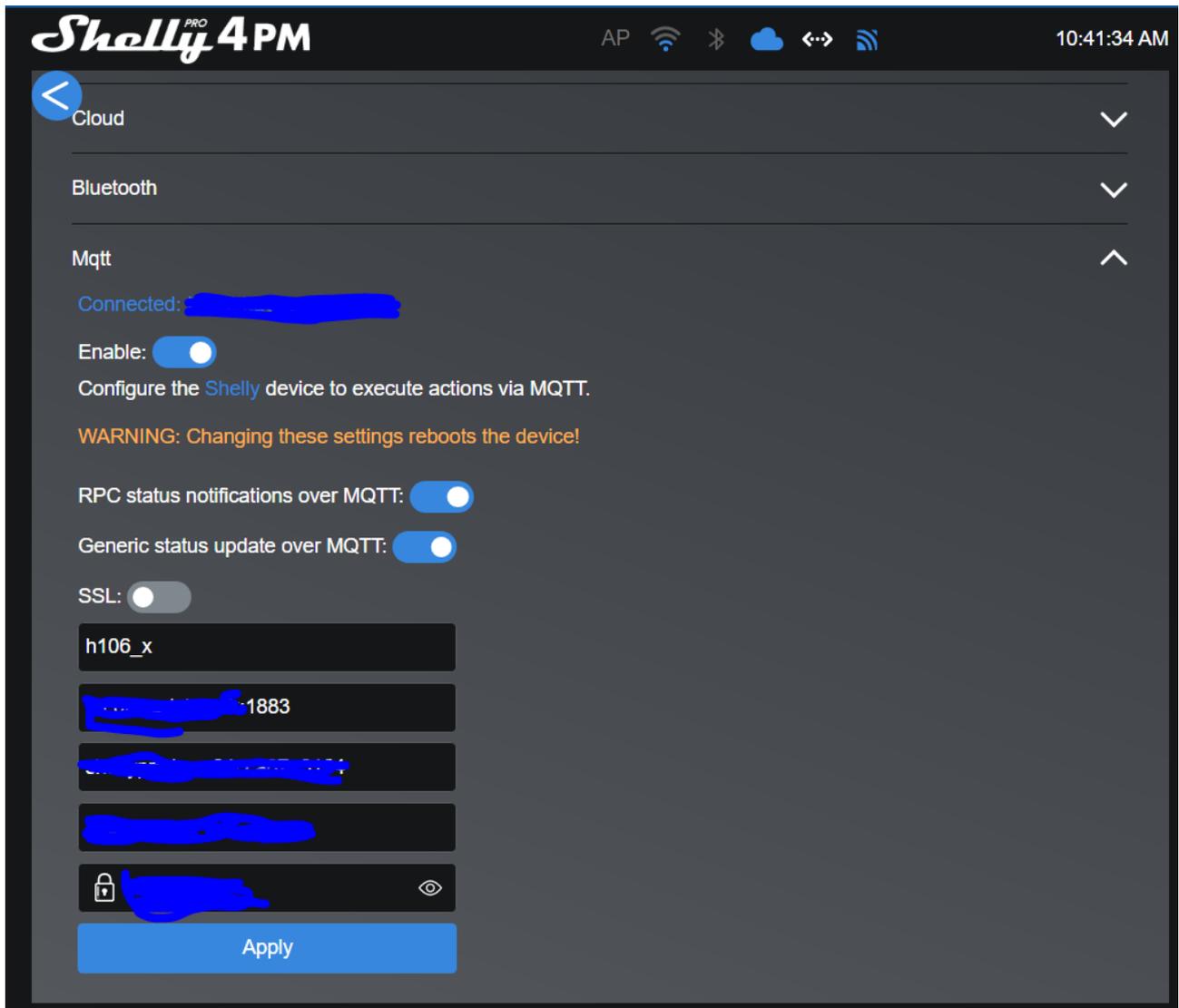


I can set a schedule, Manually turn on/off the plants an see the status of the outputs.

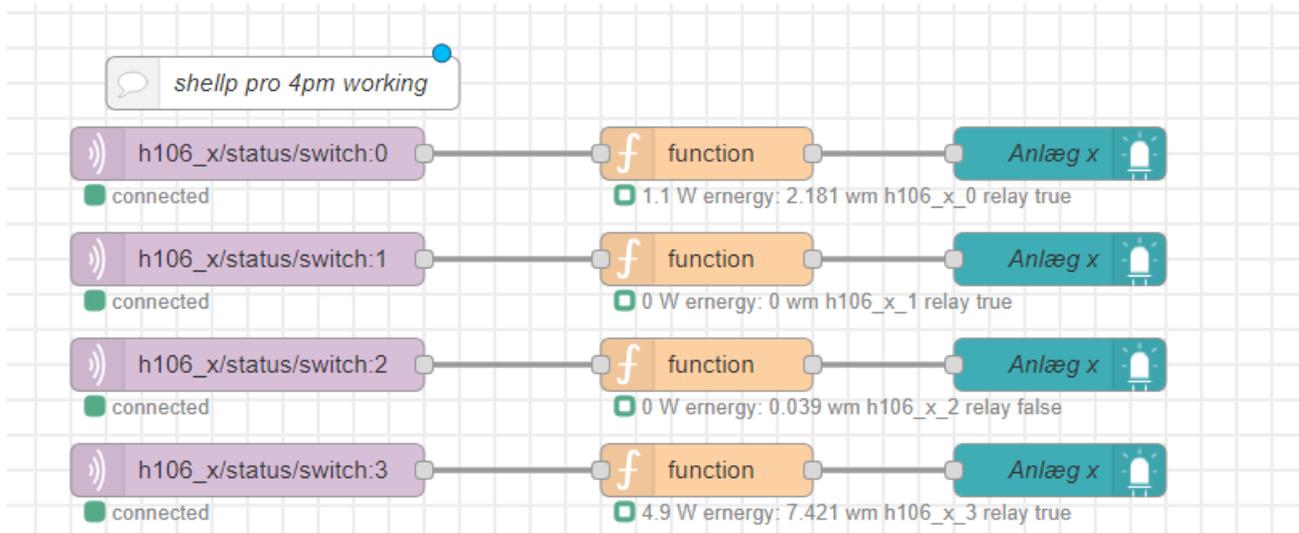
H106_6 is not running,

H106_x is my test unit.

First you must setup the Shelly Pro 4PM unit.



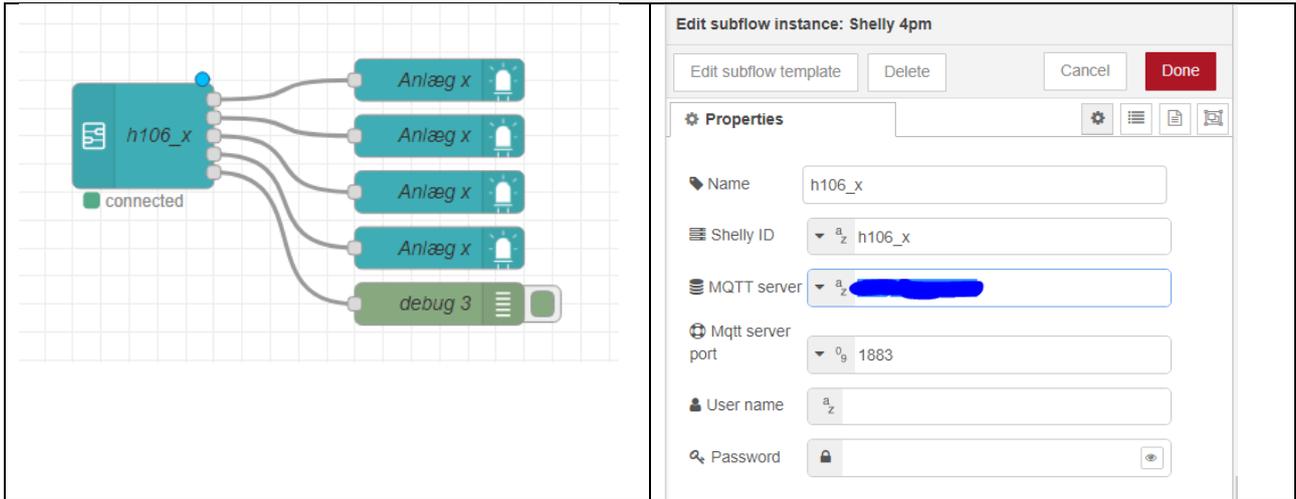
First I had to get the data from the unit.



The main goal is to make the code as dynamic as possible, so I didn't have to change to much code every time I implement a new unit, so alle the function nodes are all identical.

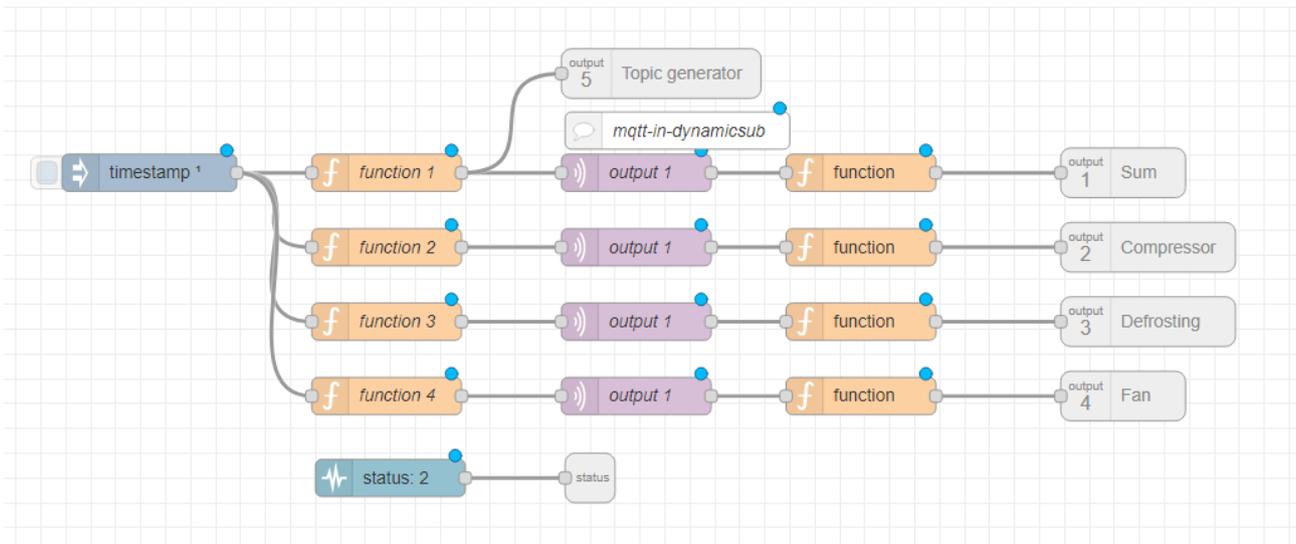
```
1 unit=msg.topic.split("/")[0];
2 relayno=msg.topic.split(":")[1];
3 var idf=[];
4 if (relayno==0){idf="sum"}
5 if (relayno==1){idf="a"}
6 if (relayno==2){idf="b"}
7 if (relayno==3){idf="c"}
8
9 ID=unit+"_"+relayno;
10 var power = msg.payload.apower;
11 var energy = msg.payload.aenergy.total;
12 var relayOn= msg.payload.output;
13
14
15
16 flow.set(unit+idf+"_power",power);
17 flow.set(unit+idf+"_total",energy);
18
19 msg.payload=relayOn;
20 msg.topic =power+" W";
21 msg.unit = unit;
22
23 node.status({fill:"green",shape:"ring",text: power+" W energy: "+energy+ " wm "+ID +" relay "+relayOn });
24 return msg;
```

It is working but not very elegant. Next up was to make it into a subflow.

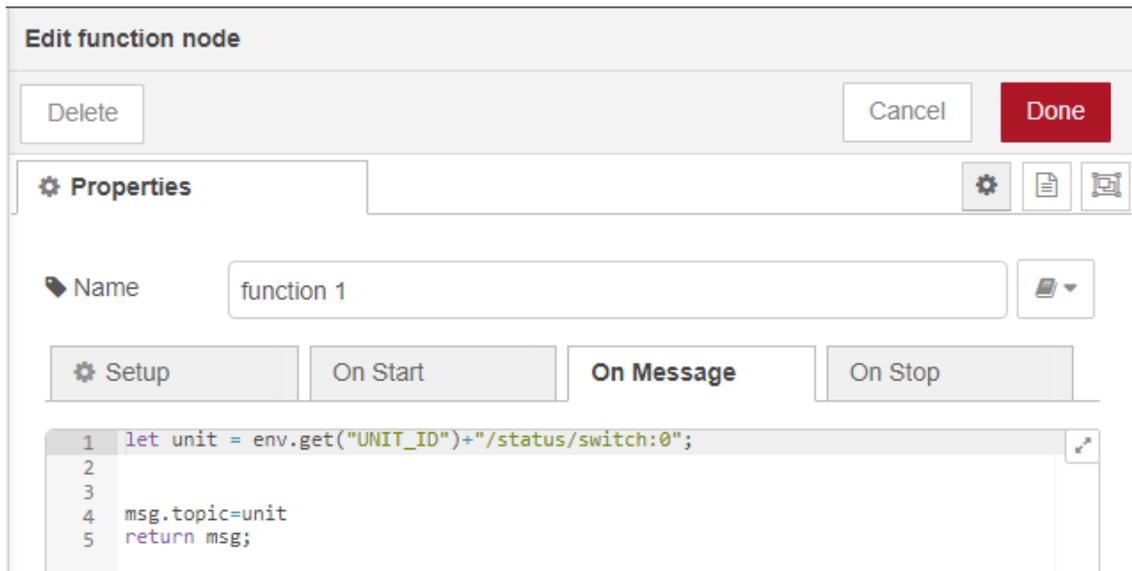


Now the setup is much easier.

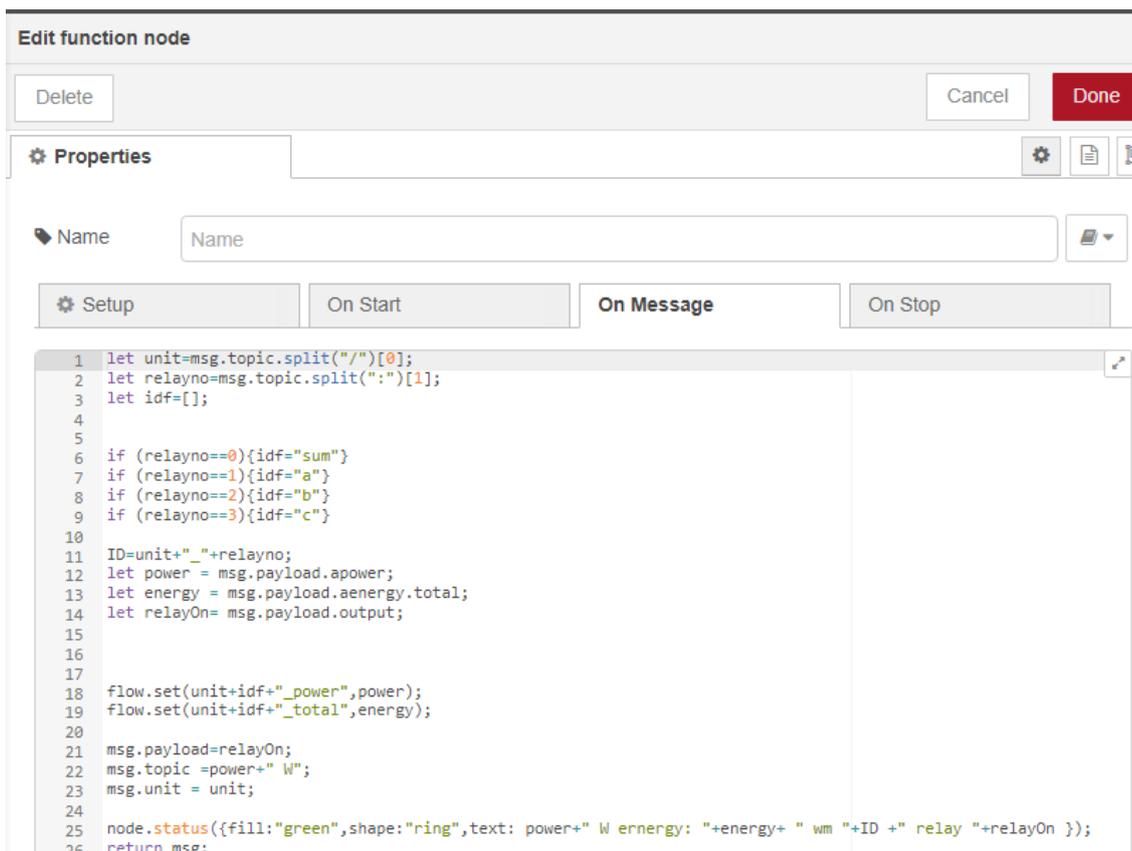
The Sub flow looks like this:



You need to get the **node-red-contrib-mqtt-dynamicsub version 0.0.9** so you can make a dynamic topic for the mqtt subscription.



The output function look like this:

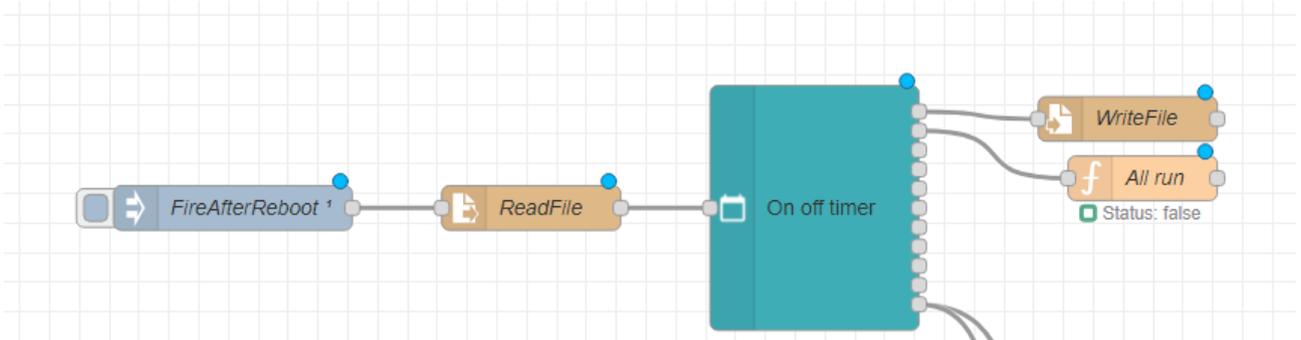


The Power and energy consumption is set to a flow.set because it is needed in another unit that measures temperature at the plant too.

Next up is to remote control the Shelly Pro 4PM:

You need to get the UI schedule node. *node-red-contrib-ui-time-scheduler* version 1.17.2

To store the schedule you need a setup like this:



The output is a Boolean (true or false)

The Shelly Pro 4PM is wired in a way that cuts all the power to the Danfoss Controller on output 0

```
1 var mannum=msg.topic.split(" ")[1]; // the number from the scheduler send as a topic from the UI scheduler
2 var runCommand=msg.payload;
3
4 if (runCommand==false){
5   msg.payload={
6     "id": 1,
7     "src": "h106_"+mannum,
8     "method": "Switch.Set",
9     "params": {
10      "id": 0,
11      "on": false
12    }
13 }
14 }
15 else if (runCommand==true){
16   msg.payload={
17     "id": 1,
18     "src": "h106_"+mannum,
19     "method": "Switch.Set",
20     "params": {
21       "id": 0,
22       "on": true
23     }
24 }
25 }
26 msg.topic="h106_"+mannum+"/rpc";
27 node.status({fill:"green",shape:"ring",text:"Schedule: "+runCommand+" ManRun: "+manrun});
28 return msg;
```

Perhaps a Youtube video would make it more understandable. For now it is working, but there are still some issues that need to be fixed before I would go on that journey.

Best regards

Henrik Boye Østergaard