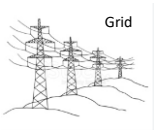
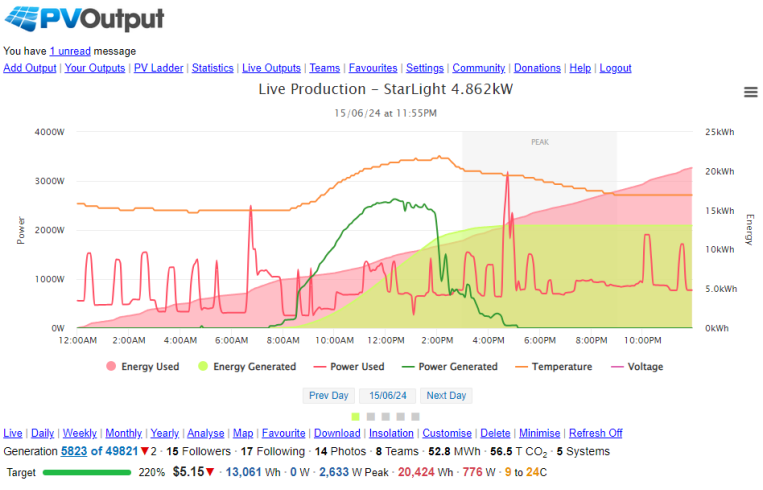
Flukso V2 & V3 Intergation
Into
Custmom Rules

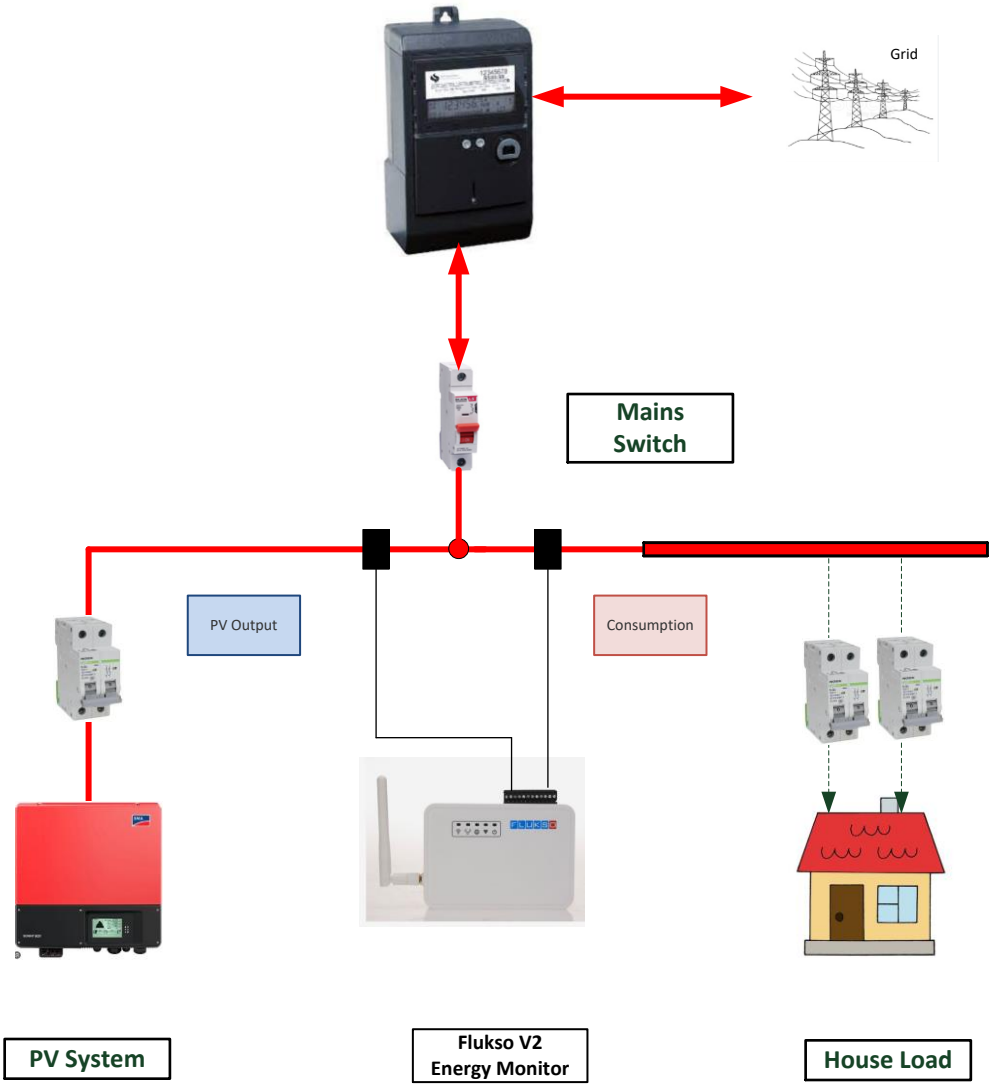


Flukso Portal No More
What Now



Drawn By Chris James

Updated: 27/06/2024
Version 1.1



Summary of Flukso V2 & V3 Diferences

This document highlights the technical differences between the Flukso version 2 and the Flukso version 3 and show's how to accommodate these differences when used in a configuration as shown above to send the data collection directly to pvoutput using the python script.

Its important to use the instructions that you received when you download your python script.
The file (Rules_HowTo_F2VO) which is sourced at "https://github.com/B-Mqn/FluksoToPvoutput/blob/B-Mqn-patch-1/Rules_HowTo_F2PVO", which details how to customize your script.

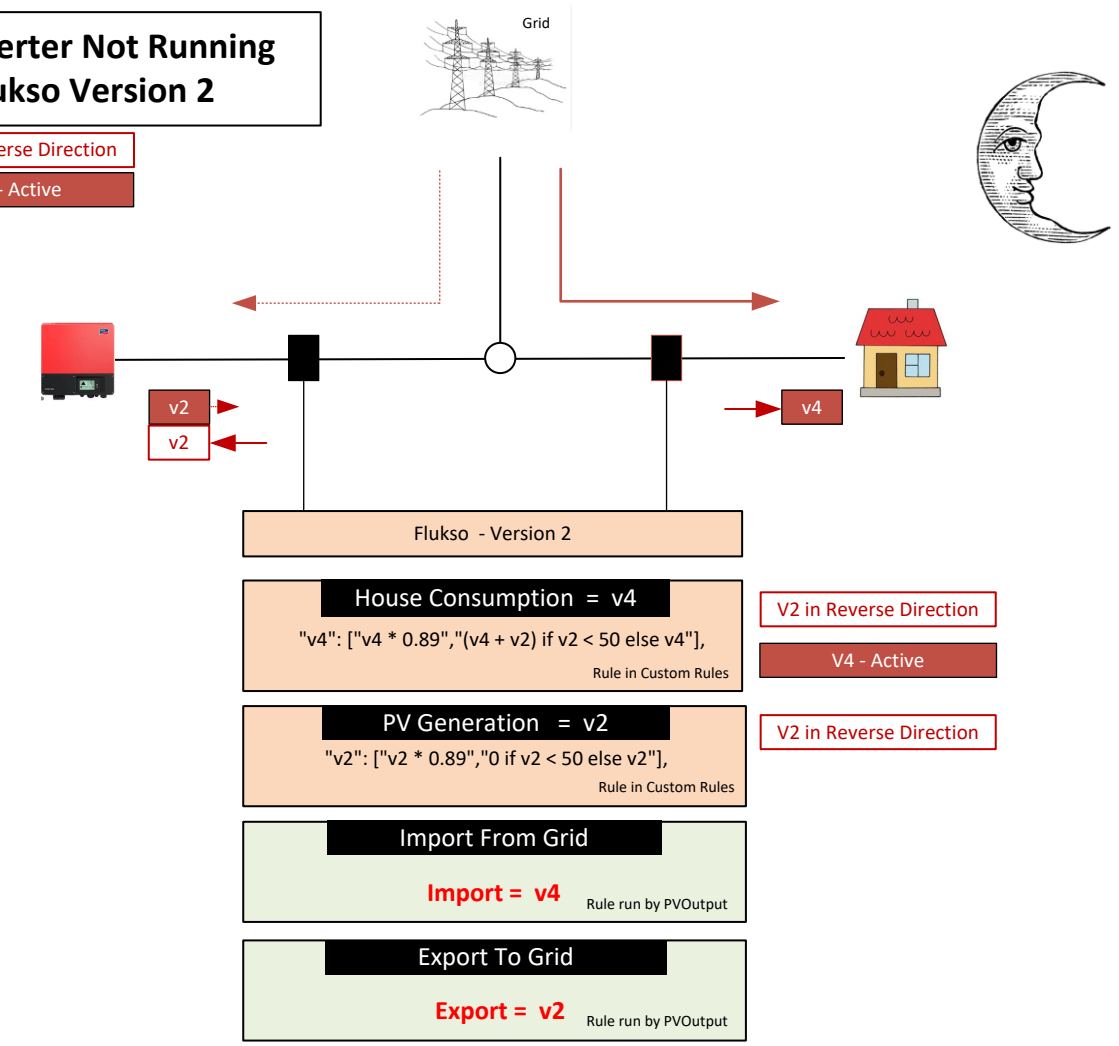
This document focus's on how the Flukso versions work in your enviroment drawing your attention to the fact you may need to create a custom rule to match your enviroment.

Flukso Version 2: The version 2 Flukso is powered by a 12v DC power pack. As a result it's hardware is unable to use AC voltage as a reference to calculate the phase of the current detected by its current clamps nor the voltage of the loads being monitored.

As a result you must configure the version 2 for a nominal line voltage which it uses to make its power calculations (Power = Nominal Voltage * Current measured by current clamps). Live changes in the line voltage are NOT reflected in the power calculations as these power calculations are only accurate when the line voltage is close to the nominal voltage you set in the flukso. Because it has no voltage reference to compare the measured current to. It cannot determine the phase and hence which direction the current is flowing in the line being monitored. So without some corrections, it shows solar generation when the inverter is running or idle. Summary can't determine the direction of current flow and measure live line voltage.

Flukso Version 3: While the power port can still accepts a 9-15V DC voltage, the Fluksometer v3's can be supplied with a 9V AC-AC adapter. The AC adapter acts as a power source to the FLM03 as well as the line voltage sensor. As the hardware now has voltage reference to compare the phase and magnitude of the measure current from the current clamp it can now determine both the current direction and the live line voltage. This makes the scripting to indicate generation and consumption much simpler as no IF conditions are required. In addition its power calculations are more accurate as its tracking the actual line voltage at any point in time. For each port, has a separate senor-id for each direction of current flow and measures live line voltage.

1 - Inverter Not Running Flukso Version 2



Extract From Custom Rules

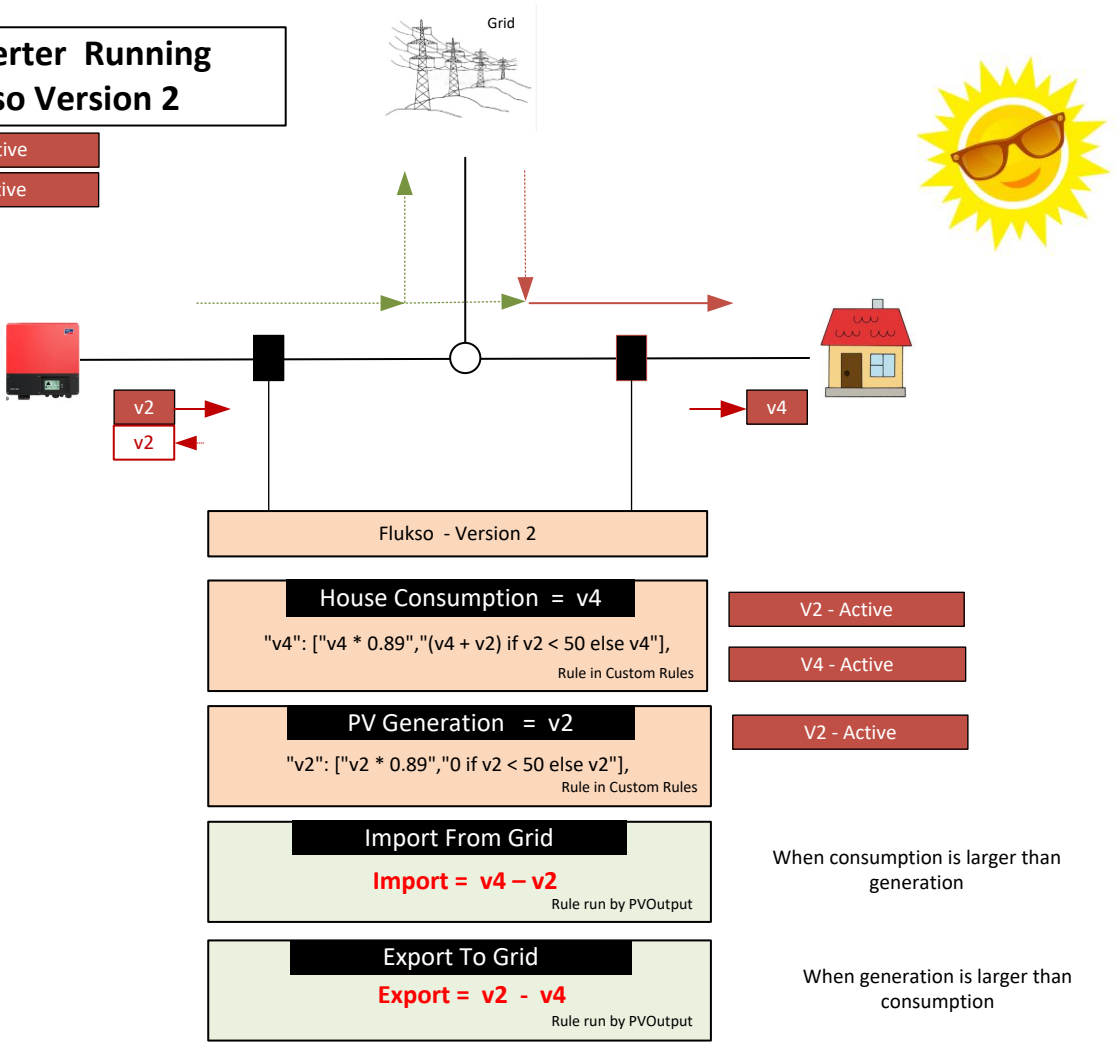
```
CUSTOM_RULES = {
    "v4": ["v4 * 0.89","(v4 + v2) if v2 < 50 else v4"],
    "v2": ["v2 * 0.89","0 if v2 < 50 else v2"],
    "v11": ["v2 - v4 if v2 > v4 else 0.0001"],
    "v12": [ "v4 - v2 if v4 > v2 else 0.0001"],
}
```

Note: Using the two electrical ports on the Flukso version 2 each of which has one sensor-id assigned to it. In order to simplify this description I refer to each sensor-id as v2 and v4 as shown above. When the inverter is not running it draws a standby current from the grid. This causes a current to flow in the reverse direction to when it's operating. As the Flukso cannot detect the current direction this would register as PV generation sent to PV output. In my case, the inverter draws 45 watts when it's not operating. So each 5 minute interval, if this is not corrected, the Flukso via the python script would tell PVOutput the generation is 45 watts when it's actually zero and the consumption value is 45 watts low. So the total error sent to pvoutput is 90 watts, or 0.09Kwh, perhaps not large enougth to effect things too much, but it does look a little strange seeing 45w being generated throughout the evening.

2 - Inverter Running Flukso Version 2

V2 - Active

V4 - Active



Extract From Custom Rules

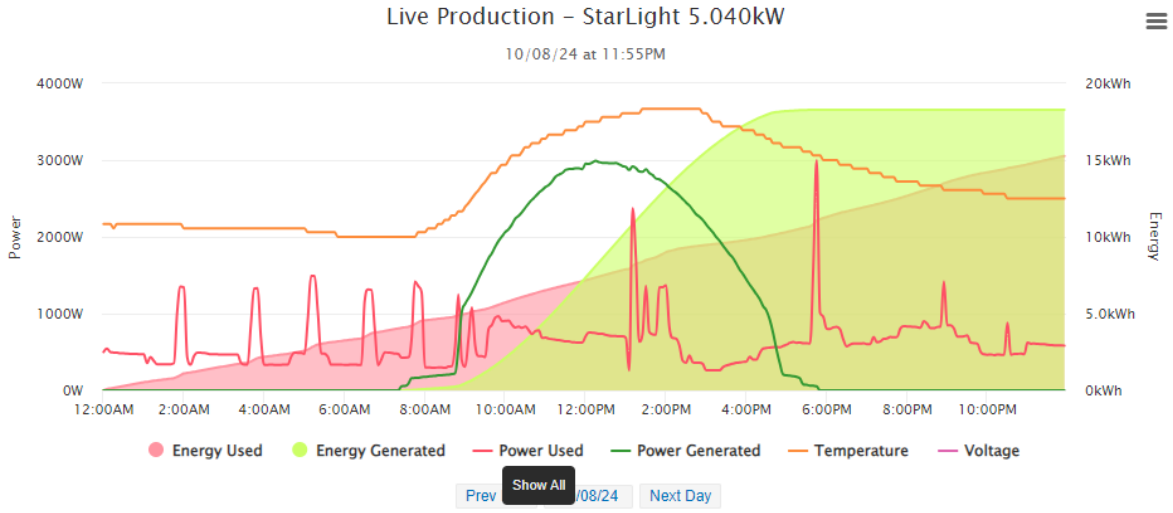
```
CUSTOM_RULES = {
    "v4": ["v4 * 0.89","(v4 + v2) if v2 < 50 else v4"],
    "v2": ["v2 * 0.89","0 if v2 < 50 else v2"],
    "v11": ["v2 - v4 if v2 > v4 else 0.0001"],
    "v12": [ "v4 - v2 if v4 > v2 else 0.0001"],
}
```

Note: Using the two electrical ports on the Flukso version 2 each of which has one sensor-id assigned to it. In order to simplify this description I refer to each sensor-id as v2 and v4 as shown above. When the inverter is running it does not draw any current from the grid. As a result current only flows in one direction when it's operating. As a result generation is always equal to v2, with only a simple modifier applied to it to correct for the nominal voltage setup in the flukso.

PVOutput

You have 1 unread message

Add Output | Your Outputs | PV Ladder | Statistics | Live Outputs | Teams | Favourites | Settings | Community | Donations | Help | Logout



Live | Daily | Weekly | Monthly | Yearly | Analyse | Map | Favourite | Download | Insolation | Customise | Delete | Minimise | Refresh Off

Generation 5894 of 50067▼1 · 15 Followers · 18 Following · 14 Photos · 8 Teams · 53.5 MWh · 57.3 T CO_2 · 4 Systems

Target 238% $3.26▼ · 18,283 Wh · 0 W · 2,990 W Peak · 15,274 Wh · 584 W · 10 to 27C

PVOutput

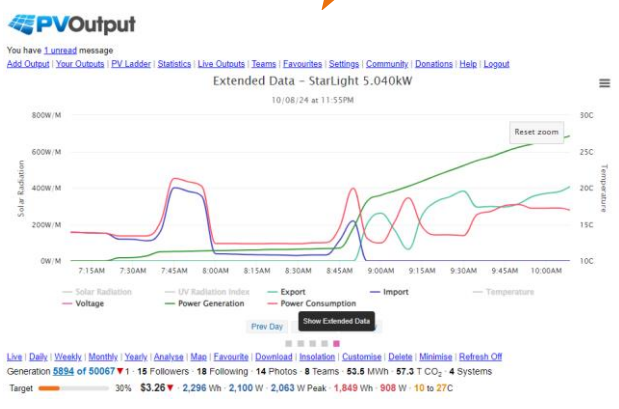
You have 1 unread message

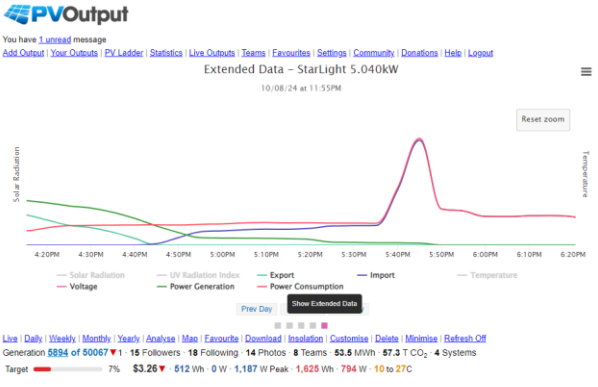
Add Output | Your Outputs | PV Ladder | Statistics | Live Outputs | Teams | Favourites | Settings | Community | Donations | Help | Logout

Live | Daily | Weekly | Monthly | Yearly | Analyse | Map | Favourite | Download | Insolation | Customise | Delete | Minimise | Refresh Off

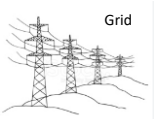
Generation 5894 of 50067▼1 · 15 Followers · 18 Following · 14 Photos · 8 Teams · 53.5 MWh · 57.3 T CO_2 · 4 Systems

Target 238% $3.26▼ · 18,283 Wh · 0 W · 2,990 W Peak · 15,274 Wh · 584 W · 10 to 27C





3 - Inverter Not Running Flukso Version 3



V2 – Not Active
V12 – Active
V4 - Active

v2
v12
v4

Flukso - Version 3

House Consumption = v4

"v12": ["v12 * 0.89"],
"v4": ["v4 * 0.89","v4 + v12"],

Rule in Custom Rules

V12 – Active
V4 - Active
V2 – Not Active

PV Generation = v2

"v2": ["v2 * 0.89"],

Rule in Custom Rules

V2 – Not Active

Import From Grid

Import = v4 - v2

Rule run by PVOutput

Export To Grid

Export = v2 - v4

Rule run by PVOutput

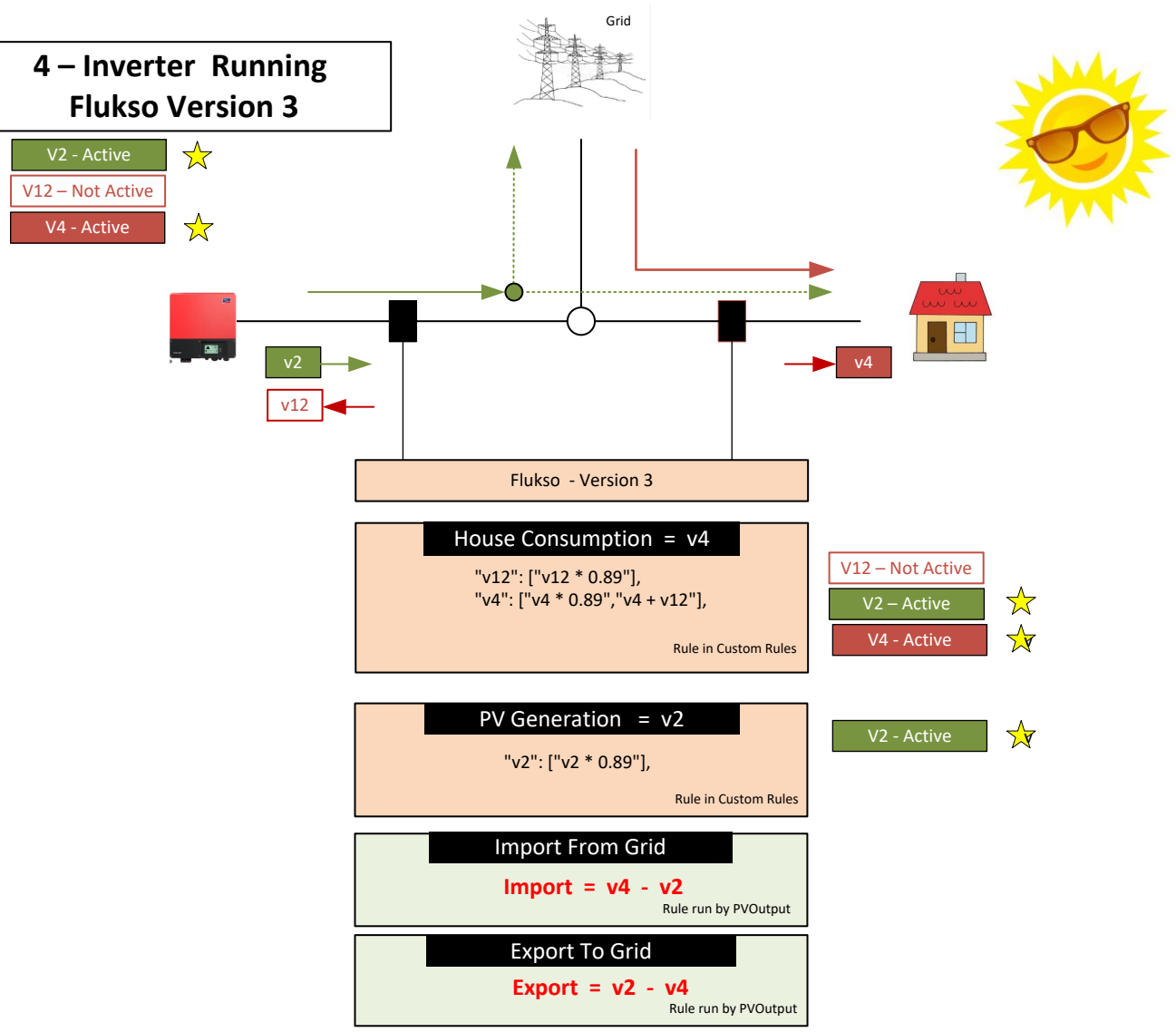
Extract From Custom Rules

```
CUSTOM_RULES = {
                "v6": ["v6 * 1.126"],
                "v2": ["v2 * 0.89"],
                "v12": ["v12 * 0.89"],
                "v4": ["v4 * 0.89","v4 + v12"],
}
```

There are three sensor ID's in use for the two current clamps, v2, v12 & v4.

Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp. In order to simplify this description I refer to each sensor-id as v2, v12 and v4 as shown above and assigned in the python script. When the inverter is not running it draws a standby current from the grid. This causes a current to flow in the reverse direction in the current clamp to when it's operating. As a result the sensor Id assigned against v2 goes to zero and the sensor ID assigned to v12 activates to indicate current in the reverse direction now flowing into the inverter. In my case, the inverter draws 45 watts when it's not operating. So each 5 minute interval, if this is not corrected, the Flukso via the python script would tell PVOutput the consumption is 45 watts low. Not large enough to effect things too much, but the evening consumption will read load and its effect would accumulate over time if not corrected. As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2 and v4 if required.

4 – Inverter Running Flukso Version 3



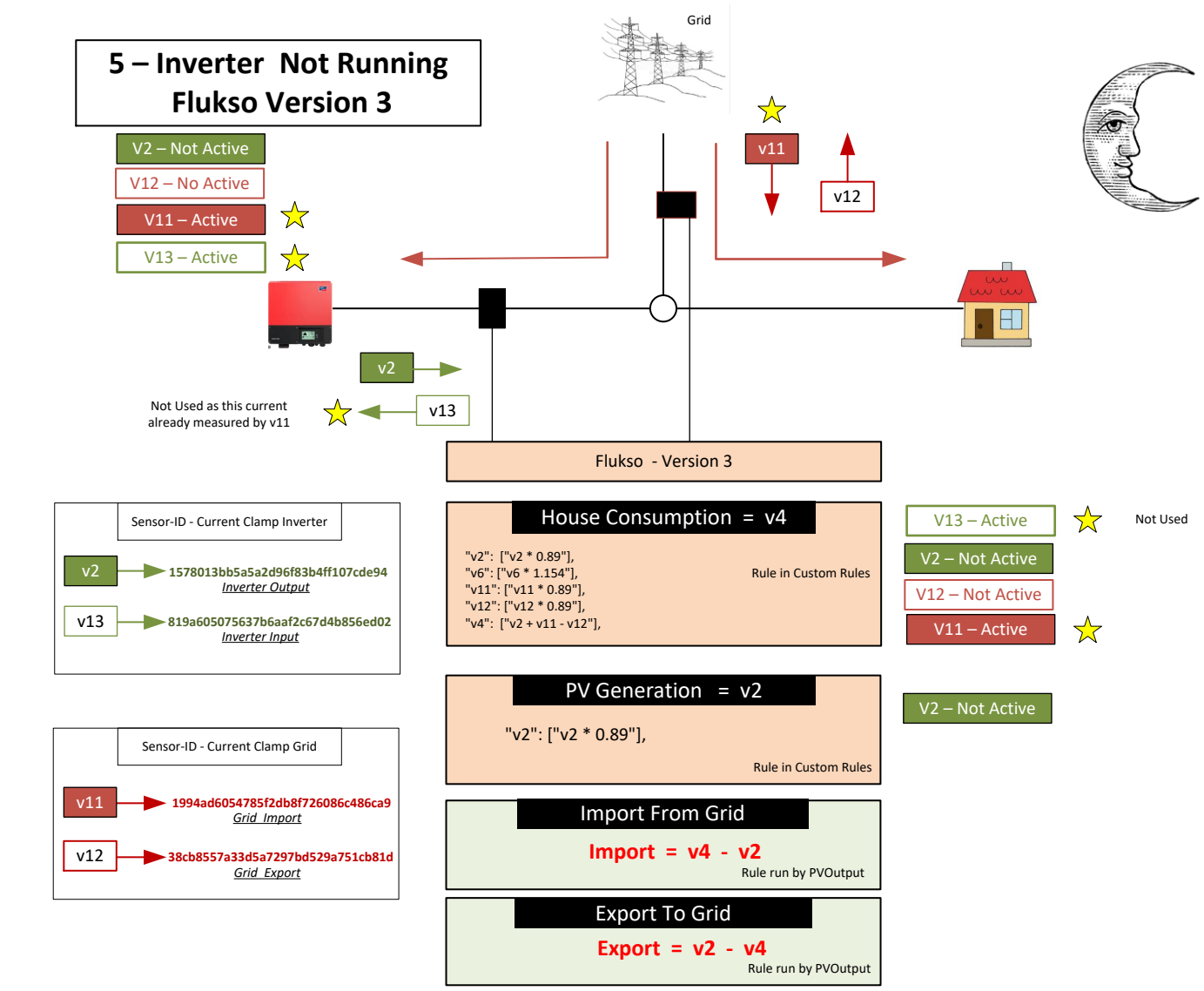
Extract From Custom Rules

```
CUSTOM_RULES = {
                "v6": ["v6 * 1.126"],
                "v2": ["v2 * 0.89"],
                "v12": ["v12 * 0.89"],
                "v4": ["v4 * 0.89","v4 + v12"],
}
```

There are three sensor ID's in use for the two current clamps, v2, v12 & v4.

Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp. In order to simplify this description I refer to each sensor-id as v2, v12 and v4 as shown above and assigned in the python script. When the inverter is not running it draws a standby current from the grid. This causes a current to flow in the reverse direction in the current clamp to when it's operating. As a result the sensor Id assigned against v2 goes to zero and the sensor ID assigned to v12 activates to indicate current in the reverse direction now flowing into the inverter. In my case, the inverter draws 45 watts when it's not operating. So each 5 minute interval, if this is not corrected, the Flukso via the python script would tell PVOutput the consumption is 45 watts low. Not large enough to effect things too much, but the evening consumption will read load and its effect would accumulate over time if not corrected. As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2 and v4 if required.

5 – Inverter Not Running
Flukso Version 3



Sensor-ID - Current Clamp Inverter

v2	1578013bb5a5a2d96f83b4ff107cde94 *Inverter Output*
v13	819a605075637b6aaf2c67d4b856ed02 *Inverter Input*

Sensor-ID - Current Clamp Grid

v11	1994ad6054785f2db8f726086c486ca9 *Grid Import*
v12	38cb8557a33d5a7297bd529a751cb81d *Grid Export*

House Consumption = v4

"v2": ["v2 * 0.89"],
"v6": ["v6 * 1.154"],
"v11": ["v11 * 0.89"],
"v12": ["v12 * 0.89"],
"v4": ["v2 + v11 - v12"],

Rule in Custom Rules

V13 – Active ★ Not Used
V2 – Not Active
V12 – Not Active
V11 – Active ★

PV Generation = v2

"v2": ["v2 * 0.89"],

Rule in Custom Rules

V2 – Not Active

Import From Grid

Import = v4 - v2

Rule run by PVOutput

Export To Grid

Export = v2 - v4

Rule run by PVOutput

Extract From Custom Rules

```
CUSTOM_RULES = {
                "v2":  ["v2 * 0.89"],
                "v6":  ["v6 * 1.154"],
                "v11": ["v12 * 0.89"],
                "v12": ["v12 * 0.89"],
                "v4":  ["v2 + v11 - v12"],
}
```

Note: I have used a Flukso Version 3 setup with one current clamp on the Inverter output with the other on the Mains/Grid connection. Testing has confirmed the Custom Rules used in this example work sending the correct values to Pvoutput.

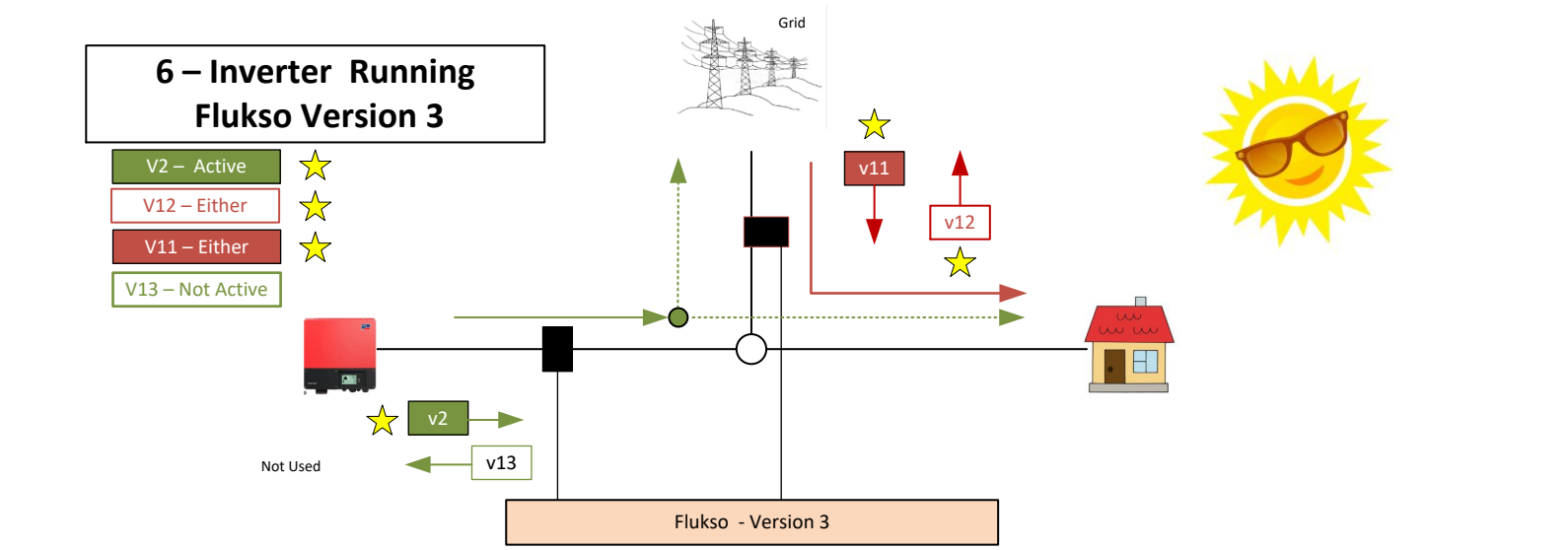
Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp.

In this setup one clamp is placed on the inverter output and the other on the grid connection as shown above. Both clamps will be subject to current flowing in two directions, so there will be four sensor id's generated by the Flukso version 3. I have assigned these for this explanation only as per the diagram above, you may choose different assignment to meet your needs.

In order to send pvoutput the generation and consumption data (v2 & v4) the script needs a Custom Rule to calculate v4 as there is no sensor id directly assigned to it as there is no current clamp on the load circuit.

As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2,v11,v12 and v13 if required.

6 – Inverter Running Flukso Version 3



Flukso - Version 3

Sensor-ID - Current Clamp Inverter

v2 → 1578013bb5a5a2d96f83b4ff107cde94 *Inverter Output*

v13 → 819a605075637b6aaf2c67d4b856ed02 *Inverter Input*

Sensor-ID - Current Clamp Grid

v11 → 1994ad6054785f2db8f726086c486ca9 *Grid Import*

v12 → 38cb8557a33d5a7297bd529a751cb81d *Grid Export*

House Consumption = v4

```
"v2":  ["v2 * 0.89"],
"v6":  ["v6 * 1.154"],
"v11": ["v11 * 0.89"],
"v12": ["v12 * 0.89"],
"v4":  ["v2 + v11 - v12"],
```

Rule in Custom Rules

V2 - Active
V12 – Either
V11 – Either

PV Generation = v2

```
"v2": ["v2 * 0.89"],
```

Rule in Custom Rules

V2 - Active

Import From Grid

Import = v4 - v2

Rule run by PVOutput

Export To Grid

Export = v2 - v4

Rule run by PVOutput

Extract From Custom Rules

```
CUSTOM_RULES = {
                "v2":  ["v2 * 0.89"],
                "v6":  ["v6 * 1.154"],
                "v11": ["v12 * 0.89"],
                "v12": ["v12 * 0.89"],
                "v4":  ["v2 + v11 - v12"],
}
```

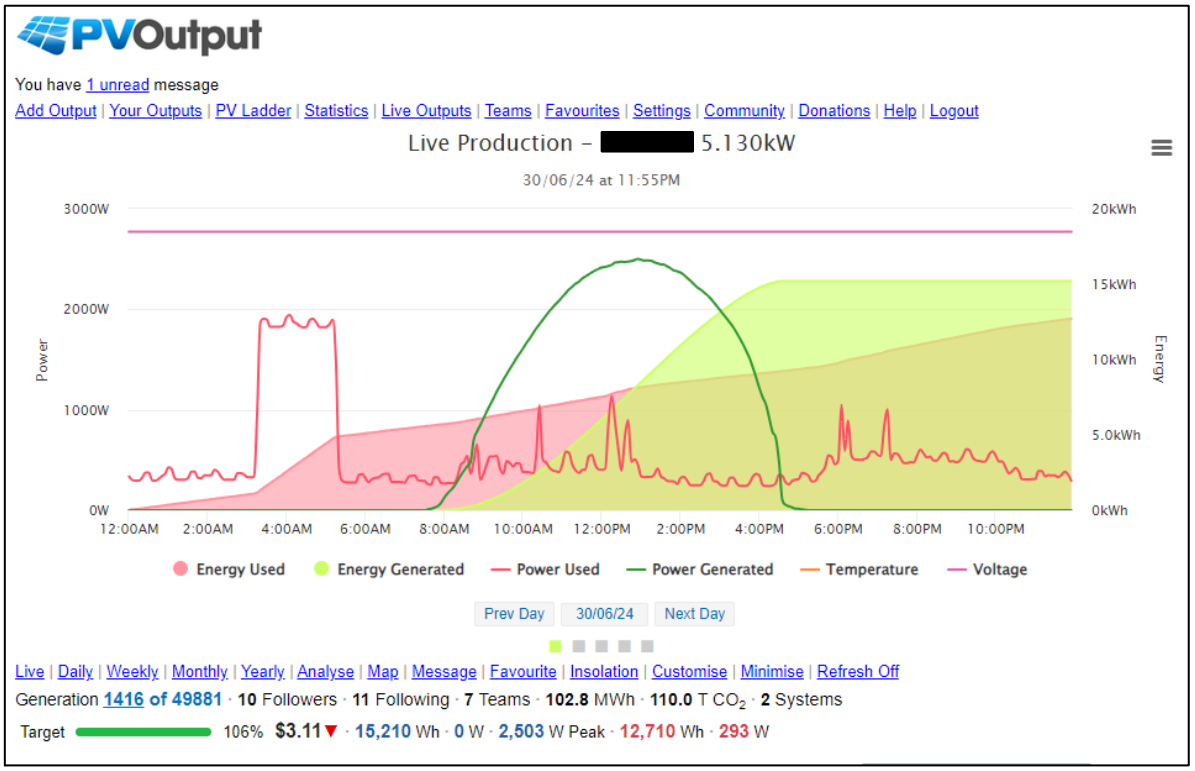
Note: I have used a Flukso Version 3 setup with one current clamp on the Inverter output with the other on the Mains/Grid connection. Testing has confirmed the Custom Rules used in this example work sending the correct values to Pvoutput.

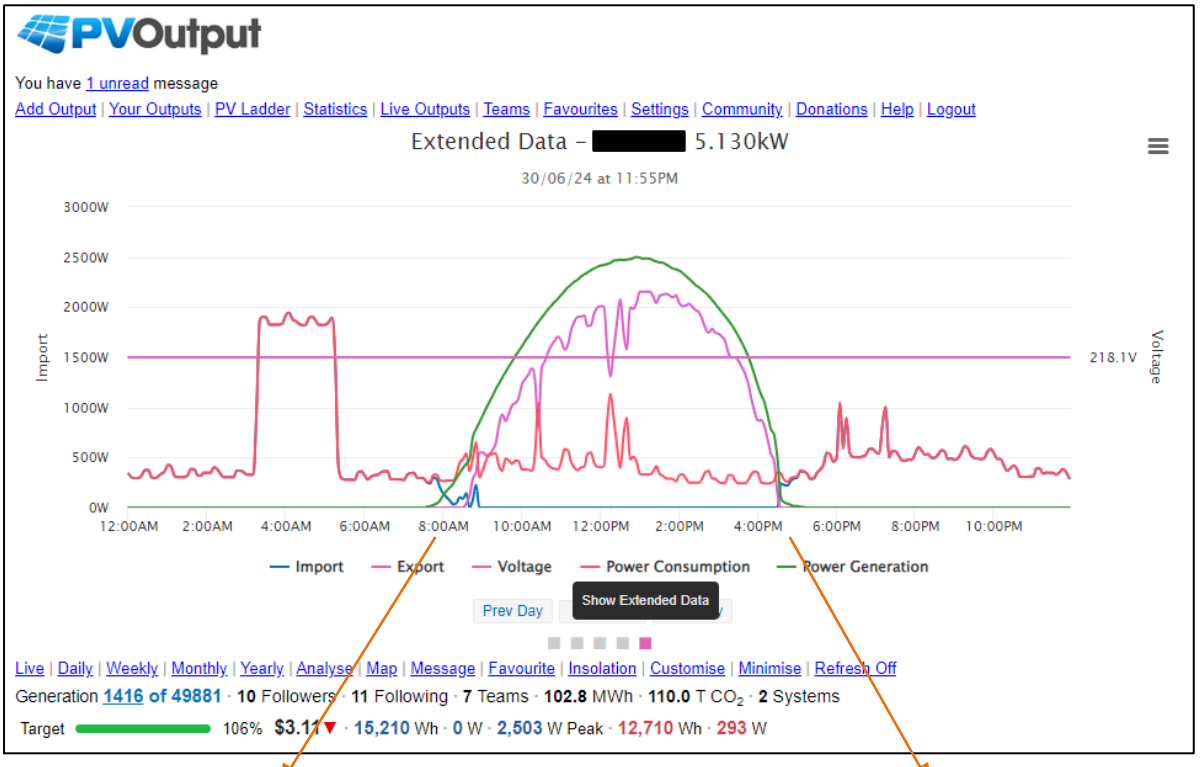
Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp.

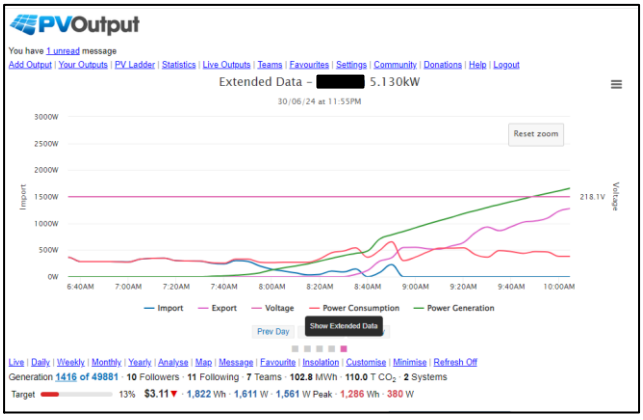
In this setup one clamp is placed on the inverter output and the other on the grid connection as shown above. Both clamps will be subject to current flowing in two directions, so there will be four sensor id's generated by the Flukso version 3. I have assigned these for this explanation only as per the diagram above, you may choose different assignment to meet your needs.

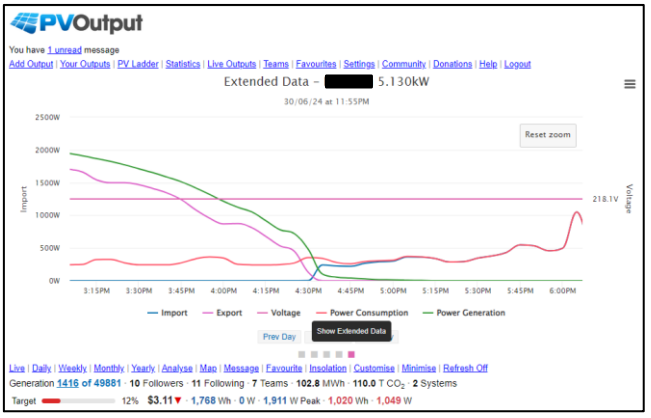
In order to send pvoutput the generation and consumption data (v2 & v4) the script needs a Custom Rule to calculate v4 as there is no sensor id directly assigned to it as there is no current clamp on the load circuit.

As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2,v11,v12 and v13 if required.


PVOutput
You have 1 unread message
Add Output | Your Outputs | PV Ladder | Statistics | Live Outputs | Teams | Favourites | Settings | Community | Donations | Help | Logout
Live Production – [redacted] 5.130kW
30/06/24 at 11:55PM
3000W
2000W
1000W
0W
Power
20kWh
15kWh
10kWh
5.0kWh
0kWh
Energy
12:00AM 2:00AM 4:00AM 6:00AM 8:00AM 10:00AM 12:00PM 2:00PM 4:00PM 6:00PM 8:00PM 10:00PM
Energy Used
Energy Generated
Power Used
Power Generated
Temperature
Voltage
Prev Day
30/06/24
Next Day
Live | Daily | Weekly | Monthly | Yearly | Analyse | Map | Message | Favourite | Insolation | Customise | Minimise | Refresh Off
Generation 1416 of 49881 · 10 Followers · 11 Following · 7 Teams · 102.8 MWh · 110.0 T CO2 · 2 Systems
Target 106% $3.11▼ · 15,210 Wh · 0 W · 2,503 W Peak · 12,710 Wh · 293 W


PVOutput
You have 1 unread message
Add Output | Your Outputs | PV Ladder | Statistics | Live Outputs | Teams | Favourites | Settings | Community | Donations | Help | Logout
Extended Data – [redacted] 5.130kW
30/06/24 at 11:55PM
3000W
2500W
2000W
1500W
1000W
500W
0W
Import
218.1V
Voltage
12:00AM 2:00AM 4:00AM 6:00AM 8:00AM 10:00AM 12:00PM 2:00PM 4:00PM 6:00PM 8:00PM 10:00PM
Import
Export
Voltage
Power Consumption
Power Generation
Prev Day
Show Extended Data
Live | Daily | Weekly | Monthly | Yearly | Analyse | Map | Message | Favourite | Insolation | Customise | Minimise | Refresh Off
Generation 1416 of 49881 · 10 Followers · 11 Following · 7 Teams · 102.8 MWh · 110.0 T CO2 · 2 Systems
Target 106% $3.11▼ · 15,210 Wh · 0 W · 2,503 W Peak · 12,710 Wh · 293 W


PVOutput
Extended Data – [redacted] 5.130kW
Reset zoom
Show Extended Data


PVOutput
Extended Data – [redacted] 5.130kW
Reset zoom
Show Extended Data

PVOutput Get and Post Parameters

Parameters

This service accepts both GET or POST requests with the following parameters -

Parameter	Field	Required	Format	Unit	Example
d	Output Date	Yes	yyyymmdd	date	20210228
t	Time	Yes	hh:mm	time	14:00
v1	Energy Generation	No [1]	number	watt hours	10000
v2	Power Generation	No	number	watts	2000
v3	Energy Consumption	No	number	watt hours	10000
v4	Power Consumption	No	number	watts	2000
v5	Temperature	No	decimal	celsius	23.4
v6	Voltage	No	decimal	volts	239.2
c1	Cumulative Flag	No	number		1
n	Net Flag	No	number		1
v7	Extended Value v7	No	number	User Defined	
v8	Extended Value v8	No	number	User Defined	
v9	Extended Value v9	No	number	User Defined	
v10	Extended Value v10	No	number	User Defined	
v11	Extended Value v11	No	number	User Defined	
v12	Extended Value v12	No	number	User Defined	
m1	Text Message 1	No	text	30 chars max	

[1] At least one of the values v1, v2, v3 or v4 must be present.