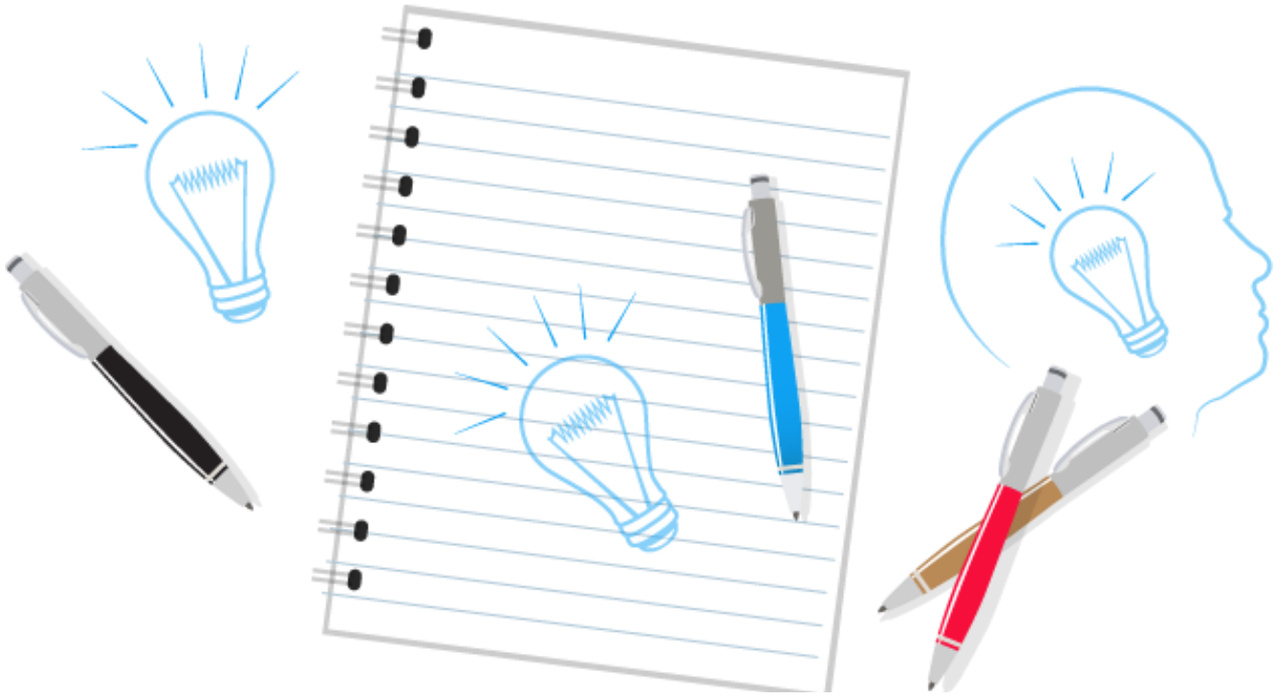
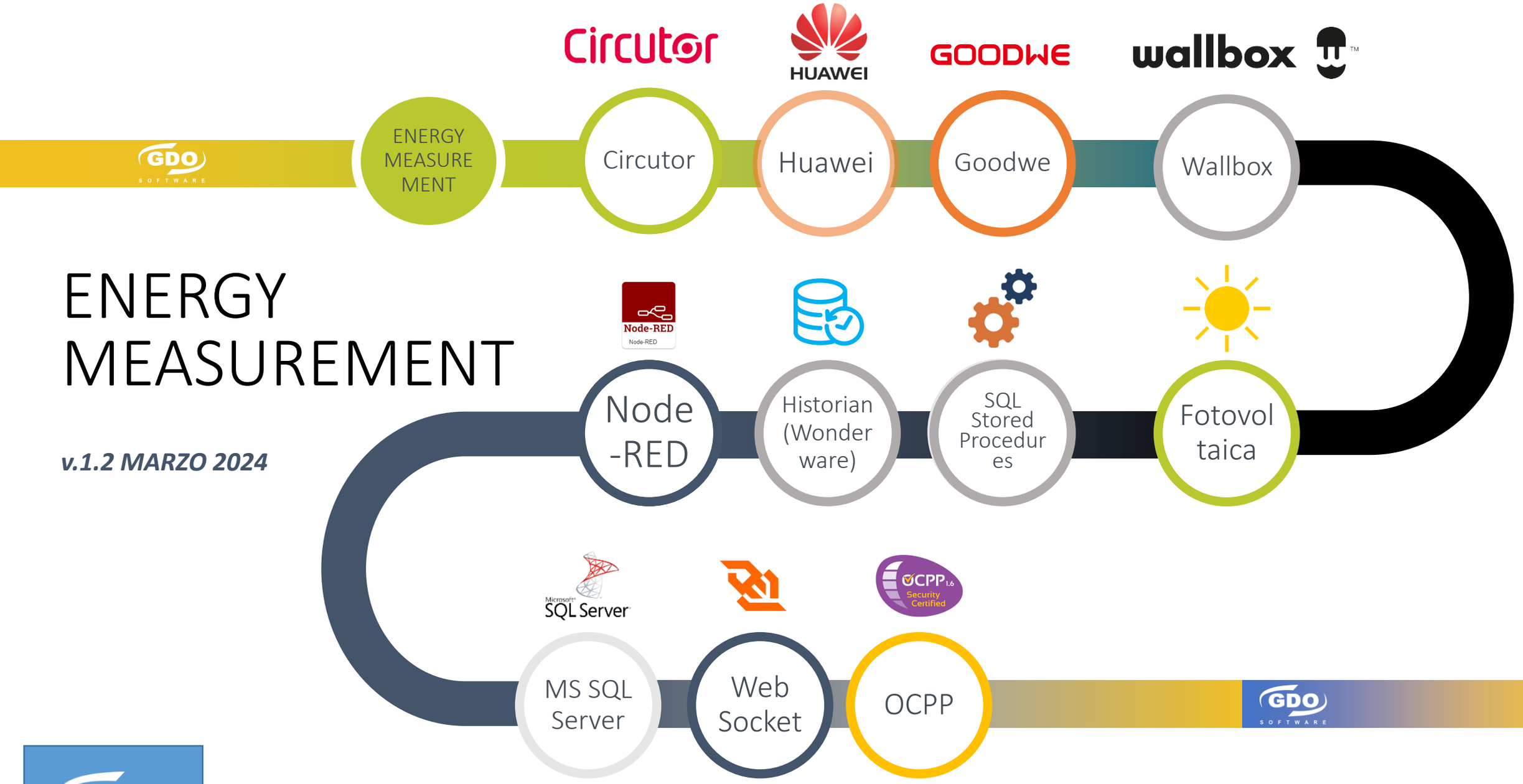
CAPÍTULO 17. Energy Measurement

v.1.2 MARZO 2024

Ricardo Moraleda Gareta

[Director departamento de software de GDO Software]





ENERGY MEASUREMENT

v.1.2 MARZO 2024

ENERGY MEASUREMENT

Energy

En este capítulo veremos 3 sistemas a medir la energía consumida / generada:

- Energía consumida en fábrica medida en Transformadores.

- Energía generada por el sistema de Fotovoltaica y auto consumida en fábrica (sin volcar a red compañía)

- Energía consumida por carga de vehículos eléctricos

Sistemas a medir

Energía consumida medida con dispositivos de Circutor por el protocolo MODBUS TCP.

Circutor

Energía generada leída de Smartlogger 3000 de Huawei o SEC1000 de Goodwe por el protocolo MODBUS TCP.

HUAWEI GOODWE

Energía consumida leída de los cargadores Wallbox por el protocolo OCPP.

wallbox

Circutor CIRCUTOR Circutor

Circutor

Para poder medir la energía consumida en diferentes puntos estratégicos de la planta empleamos medidores de la marca Circutor.

Circutor CVM K2

Circutor CVM C10

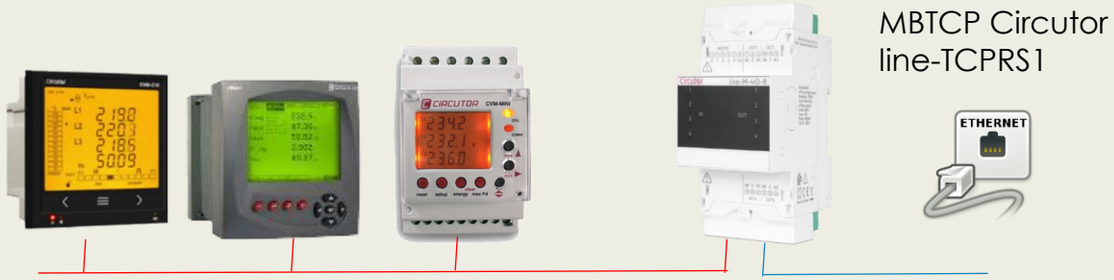
Circutor CVM Mini

Comunicaciones Modbus

Estos dispositivos comunican por Modbus RTU (bus serie RS-485) cosidos uno a uno hasta una pasarela Modbus TCP (para conectarlos a la red Ethernet).

Al ser protocolo Modbus RTU tienen un UID de esclavo de manera que son accesible a través de la dirección IP de la pasarela y el esclavo pertinente.

La pasarela, también de Circutor line-TCPRS1. Dispone interfaz web para su configuración.

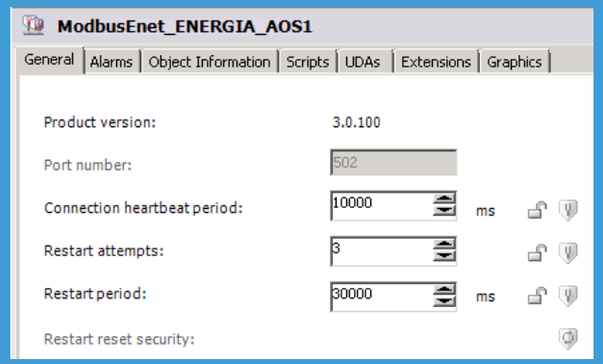


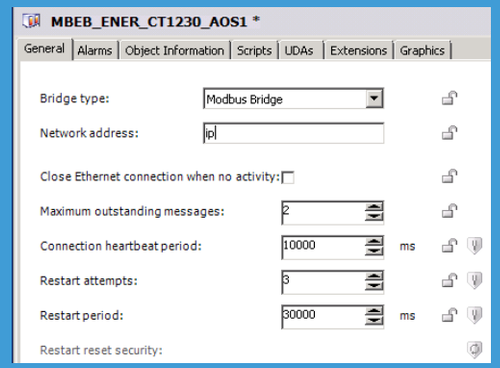
Get & Save

Adquisición

Para la adquisición de los datos nos basamos en una plataforma estándar llamada System Platform y para su historización en Historian de Wonderware.

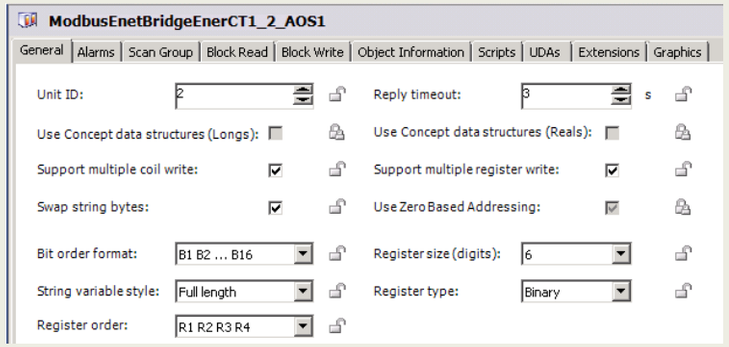
En esta plataforma se programan los objetos y los drivers de comunicaciones para acceder a los medidores a través de las pasarelas MBTCP de Circutor.



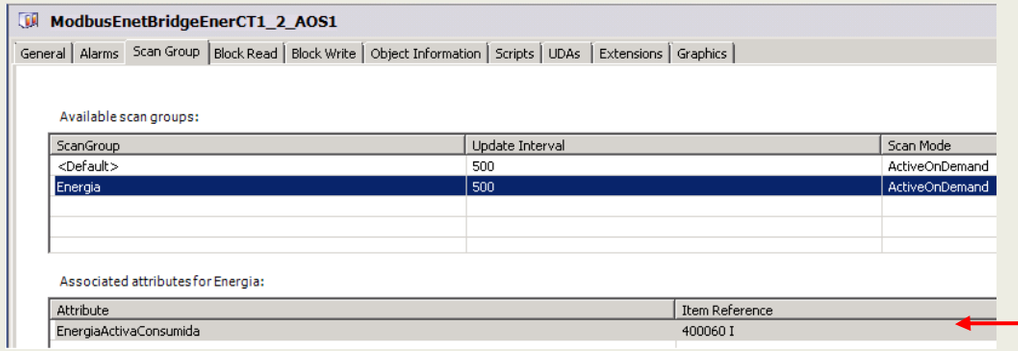


Drivers de Comunicaciones

Driver con la IP de la pasarela, puerto 502 y UID del esclavo del medidor.



Energía activa positiva en el holding register 400.060 I
(2 Words = 32 bits)



Get & Save Modbus

Mapa de memoria CVM Mini

Las direcciones de cada magnitud están en los manuales de cada dispositivo en el apartado Mapa de memoria Modbus. Cada uno es diferente.

Para el caso de **CVM Mini**, la Energía (Wh) instantánea está en las posiciones 3C-3D (hexadecimal). En decimal son 60-61, por eso 400.060 I ya que es la posición 60 con 2 Words de longitud dentro de los 400.000 (holding registers). Luego se debe dividir por 1.000 si se quiere en kWh.

MAGNITUD	SIMBOLO	Instantáneo	Uds.
Energía Activa	*kW·h III*	3C-3D	*w·h*

Attribute	Item Reference
EnergiaActivaConsumida	400060 I

Mapa de memoria CVM K2/C10

CVM K2 → 5D8-5D9 (HEX) = 1496-1497 (DEC)

TOTAL TARIFF			
Active energy	kW·h III	201	5D8-5D9

Attribute	Item Reference
EnergiaActivaConsumida	401496 I

CVM C10 → DC-DD (HEX) = 220-221 (DEC)

Parámetro	Símbolo	Total	Unidades
Energía activa consumida (kW)	kWh III	DC-DD	kWh

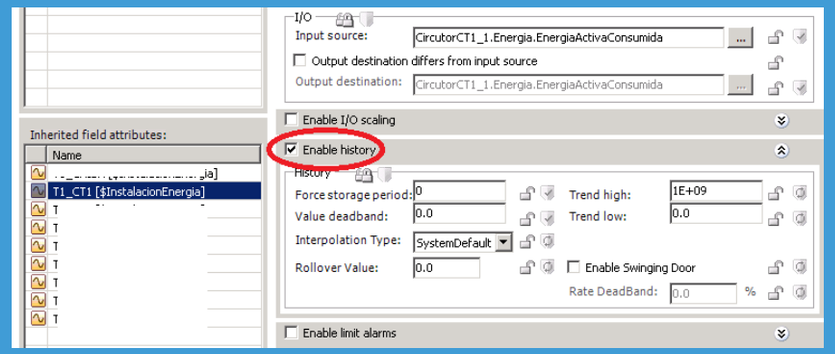
Attribute	Item Reference
EnergiaActivaConsumida	400220 I

Get & Save

Historización

Para historizar los datos en Historian basta con marcar "Enable history" en las señales adquiridas anteriores.

Esta configuración irá registrando el valor en Historian. Para poder obtener los datos de Historian se accederá mediante SQL Server. Para ello el sistema crea una BD llamada **Runtime**, varias tablas y vistas. En este caso accederemos a una vista llamada **Runtime.dbo.AnalogHistory** a través de su Tagname.



HISTORIAN-SQL

Con la aplicación QUERY de Historian podemos ver y filtrar estos datos. Al ser un SQL Server los datos son accesibles desde aplicaciones de terceros con consultas SQL → Node-RED.

Results

SQL | Data

TagName	DateTime	Value
InstalacionEnergia.T1_CT1	2023-03-13 04:41:49.09000	280605
InstalacionEnergia.T1_CT1	2023-03-13 06:23:38.18100	303742
InstalacionEnergia.T1_CT1	2023-03-13 08:05:27.27200	318544
InstalacionEnergia.T1_CT1	2023-03-13 09:47:16.36300	332497
InstalacionEnergia.T1_CT1	2023-03-13 11:29:05.45400	284640
InstalacionEnergia.T1_CT1	2023-03-13 13:10:54.54500	209043
InstalacionEnergia.T1_CT1	2023-03-13 14:52:43.63600	176233
InstalacionEnergia.T1_CT1	2023-03-13 16:34:32.72700	189726
InstalacionEnergia.T1_CT1	2023-03-13 18:16:21.81800	237450
InstalacionEnergia.T1_CT1	2023-03-13 19:58:10.90900	325591
InstalacionEnergia.T1_CT1	2023-03-13 21:40:00.00000	359749
InstalacionEnergia.T1_CT1	2023-03-13 23:21:49.09000	354928
InstalacionEnergia.T1_CT1	2023-03-14 01:03:38.18100	362161
InstalacionEnergia.T1_CT1	2023-03-14 02:45:27.27200	364592
InstalacionEnergia.T1_CT1	2023-03-14 04:27:16.36300	385891
InstalacionEnergia.T1_CT1	2023-03-14 06:09:05.45400	393426
InstalacionEnergia.T1_CT1	2023-03-14 07:50:54.54500	397407
InstalacionEnergia.T1_CT1	2023-03-14 09:32:43.63600	392564
InstalacionEnergia.T1_CT1	2023-03-14 11:14:32.72700	308131
InstalacionEnergia.T1_CT1	2023-03-14 12:56:21.81800	188020
InstalacionEnergia.T1_CT1	2023-03-14 14:38:10.90900	136803
InstalacionEnergia.T1_CT1	2023-03-14 16:20:00.00000	195024

Node-RED

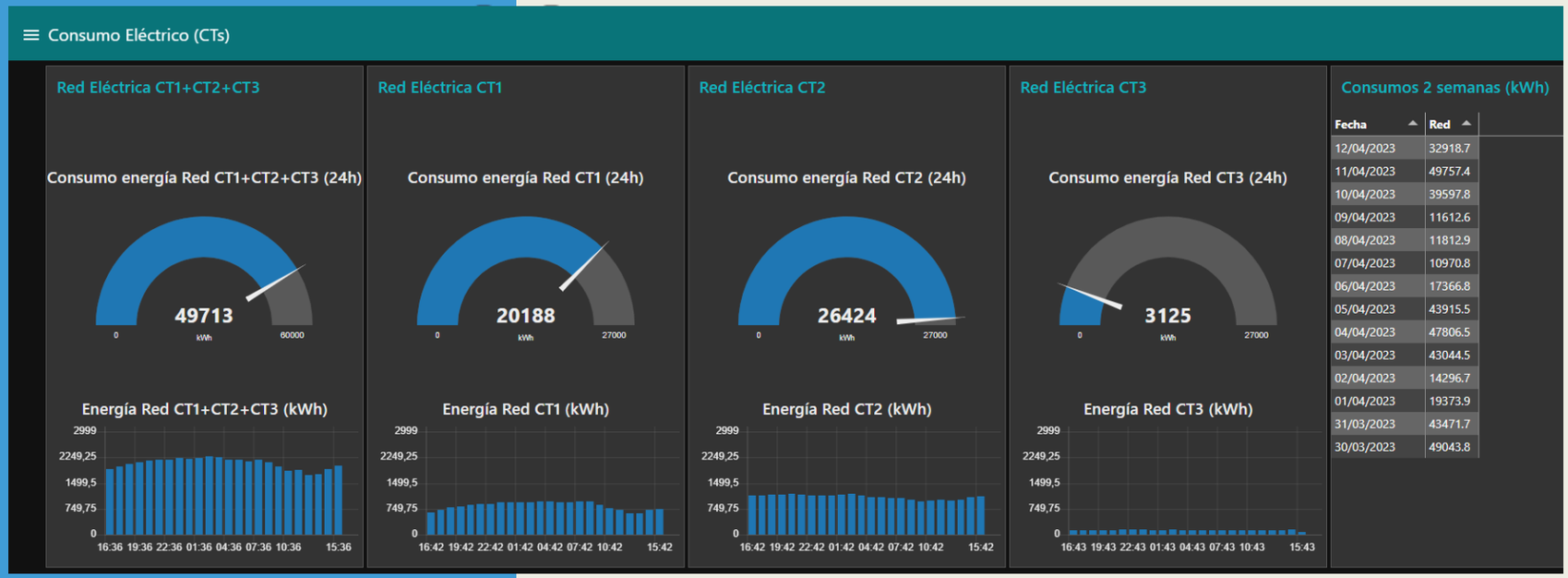
Node-RED

Para la representación gráfica utilizaremos Node-RED. Esta herramienta también es capaz de capturar el dato por MBTCP, incluso de historizarlo, por ejemplo, en una BD InfluxDB, MySQL o cualquier otra para series temporales.

En este caso es sólo UI para consultas en “tiempo real”.

Dashboard

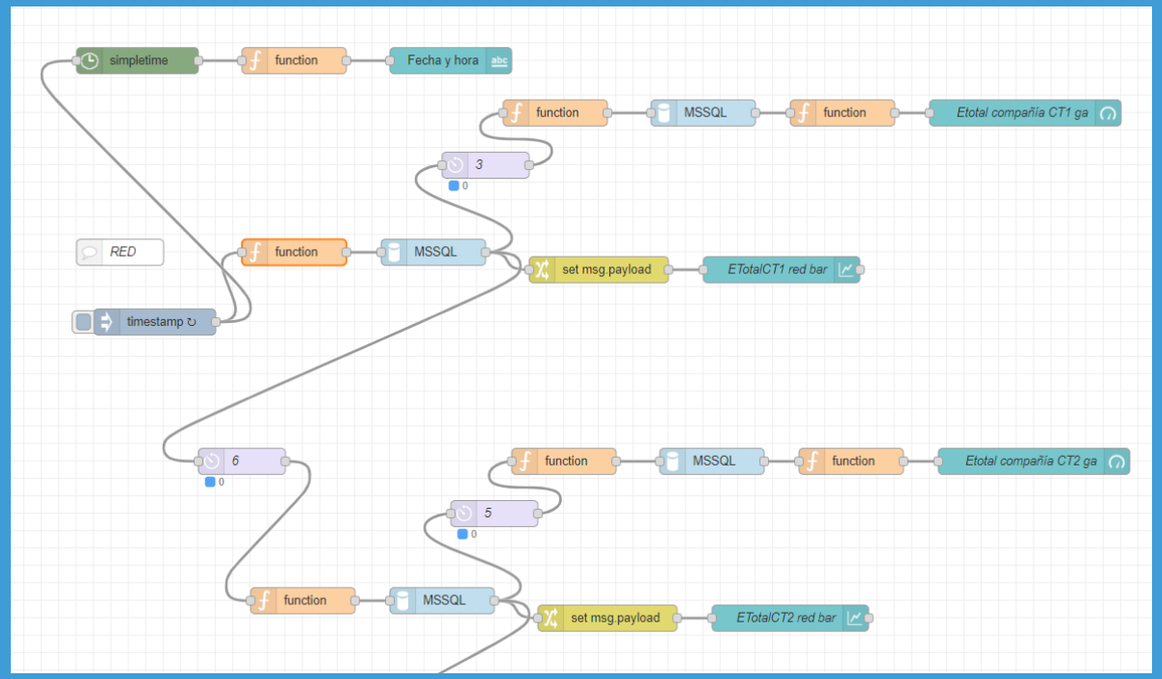
Se representan Gauges por cada Centro de Transformación y la suma de los 3. Valores de energía consumida (kWh) por hora en el último día y tabla de totales por día.



Node-RED

Node-RED

Se representa parte del flujo. En este caso 2 gauges y 2 gráficas de barras por hora.



HISTORIAN-SQL

Para ello se utilizan nodos Function para hacer las queries SQL y pasarlas al nodo MSSQL.

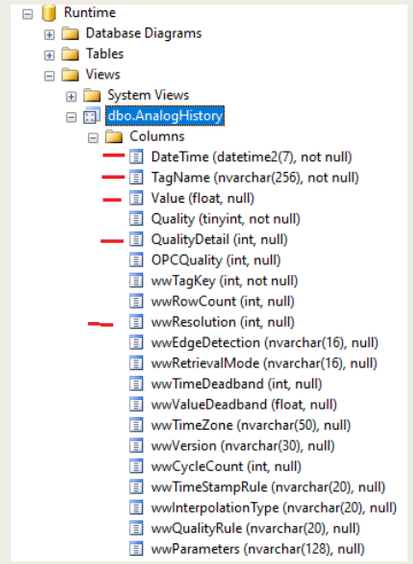
Los resultados se filtran para sacar el dato en cuestión, es decir, la suma total o los datos por hora en un periodo dado (las últimas 24h).

Las queries están basadas

en el origen de datos,

como se ha dicho antes,

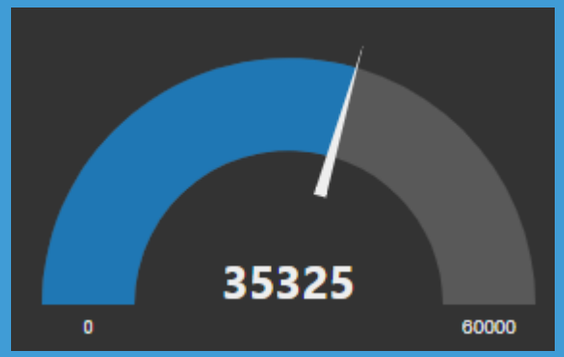
Runtime.dbo.AnalogHistory.

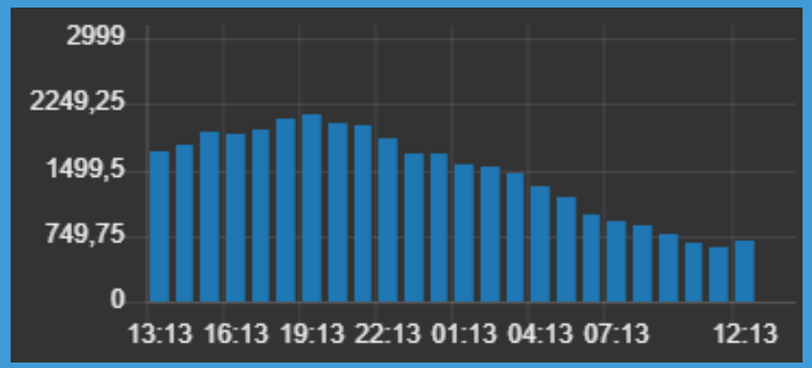


Node-RED

Node-RED

Un detalle clave de una query para dar el consumo de un día por horas y graficarlo.





HISTORIAN-SQL

Query y las claves para entenderla:

```
select t.fechaHora as x, round(t.kWh * 0.001, 2) as kWh, round(t.SaltokWh * 0.001, 2) as y from (SELECT DateTime as fechaHora, ROUND(Value, 2) as kWh, ROUND(Value, 2) - LAG(ROUND(Value, 2)) over(order by Datetime) as SaltokWh FROM Runtime.dbo.AnalogHistory where Tagname = 'InstalacionEnergia.T1_CT1' and DateTime >= DateAdd(hh, -24, GetDate()) and DateTime <= GetDate() and wwresolution = 3600000 and QualityDetail = 192) as t where t.SaltokWh is not null and t.SaltokWh > 0
```

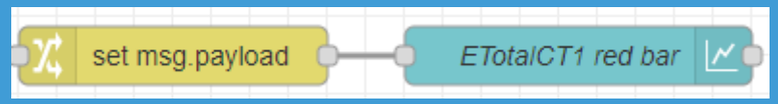
- Se obtienen datos de cada hora con **wwresolution** = 3600000
- Se calcula el incremento de la hora actual respecto la anterior con **LAG()**.

	x	kWh	y
1	2023-04-28 13:17:12.1400000	634944,91	138,17
2	2023-04-28 14:17:12.1400000	635097,15	152,23
3	2023-04-28 15:17:12.1400000	635255,1	157,96
4	2023-04-28 16:17:12.1400000	635414,92	159,82
5	2023-04-28 17:17:12.1400000	635588,82	173,89
6	2023-04-28 18:17:12.1400000	635801,72	212,91
7	2023-04-28 19:17:12.1400000	636042,61	240,88
8	2023-04-28 20:17:12.1400000	636282,48	239,88
9	2023-04-28 22:17:12.1400000	636755,61	473,12
10	2023-04-28 23:17:12.1400000	636964,23	208,62
11	2023-04-29 00:17:12.1400000	637176,68	212,46
12	2023-04-29 01:17:12.1400000	637379,49	202,81
13	2023-04-29 02:17:12.1400000	637581,91	202,41
14	2023-04-29 03:17:12.1400000	637776,43	194,53
15	2023-04-29 04:17:12.1400000	637966,61	190,18
16	2023-04-29 05:17:12.1400000	638145,12	178,51
17	2023-04-29 06:17:12.1400000	638295,75	150,63
18	2023-04-29 07:17:12.1400000	638445,7	149,95
19	2023-04-29 08:17:12.1400000	638590,13	144,43
20	2023-04-29 09:17:12.1400000	638694,52	104,39
21	2023-04-29 10:17:12.1400000	638764,69	70,17
22	2023-04-29 11:17:12.1400000	638821,74	57,05
23	2023-04-29 12:17:12.1400000	638896,89	75,15

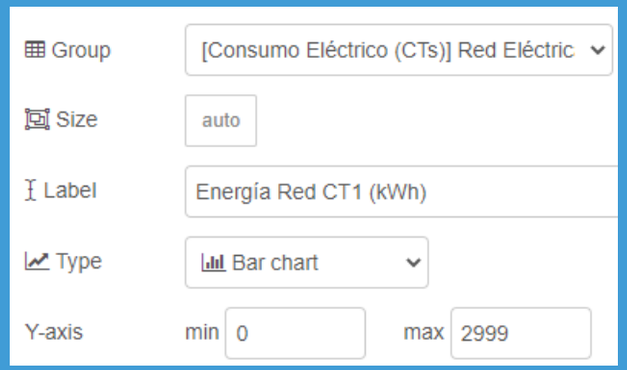
Node-RED

Node-RED

Después de la query anterior se quiere graficar el consumo del día anterior por horas.



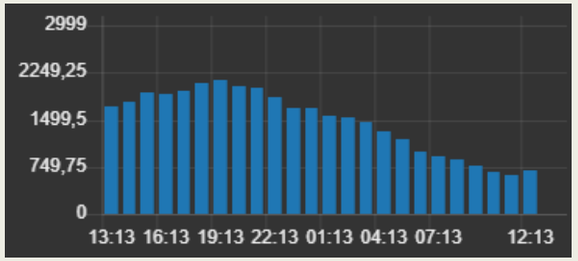

```
[
    {
        "series": ["T1234_CT1"],
        "data": [[msg.payload.SaltokWh]],
        "labels": [msg.payload.fechaHora]
    }
]
```



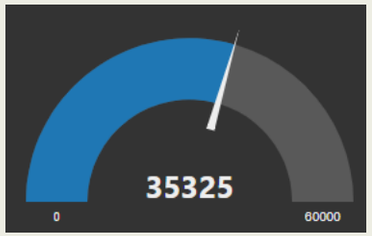
HISTORIAN-SQL

Resultando una Bar Chart como la siguiente.

Es realmente interesante saber el consumo por hora y ver la tendencia de consumo durante el día.



La suma de cada barra de la gráfica daría el área completa del consumo total.

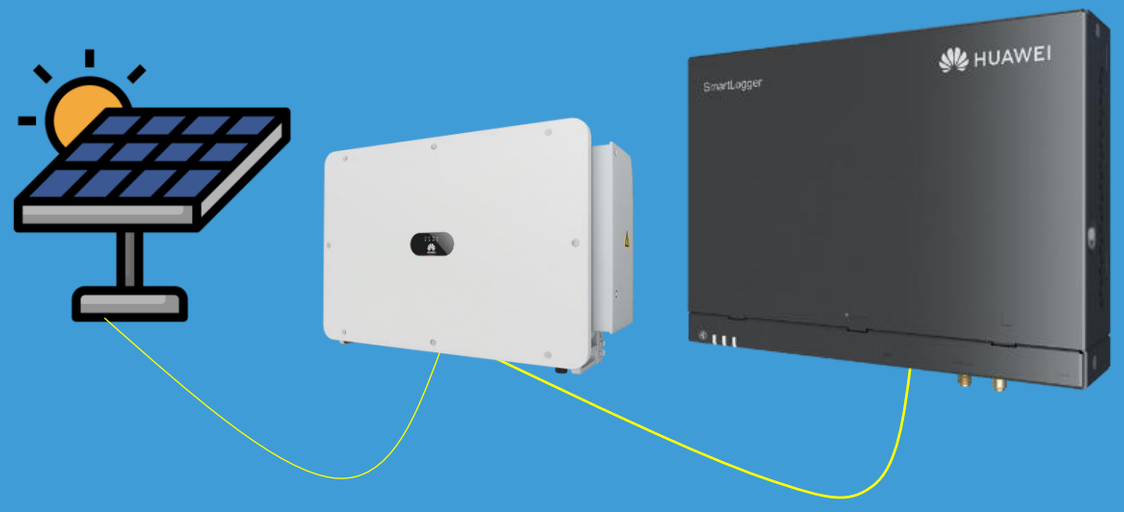


FOTOVOLTAICA

HUAWEI

Para poder generar energía para auto consumir se utilizan paneles solares, inversores (SUN2000-100KTL) y un sistema de control.

El sistema de control de la marca Huawei y modelo Smartlogger 3000A.



Comunicaciones Modbus

El inversor es el equipo encargado de transformar la corriente continua (DC) procedente de las baterías o de los paneles solares en corriente alterna (AC).

Se establece un bus Modbus RTU (bus serie RS-485) entre inversores cosidos uno a uno hasta el SmartLogger que comunica por Modbus TCP (para conectarlos a la red Ethernet).

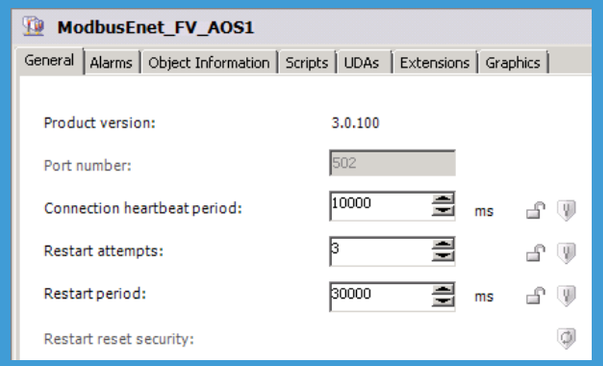
Al ser protocolo Modbus RTU cada inversor tiene un UID de esclavo de manera que son accesible a través de la dirección IP del controlador.

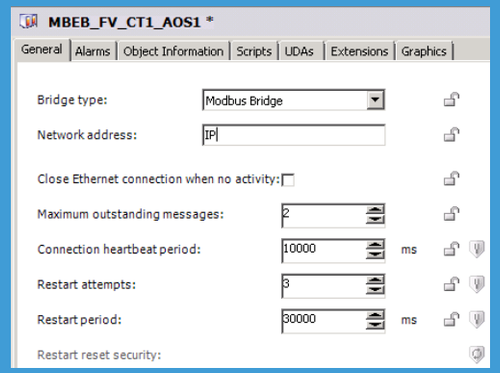
Get & Save

Adquisición

Para la adquisición de los datos nos basamos en una plataforma estándar llamada System Platform y para su historización en Historian de Wonderware.

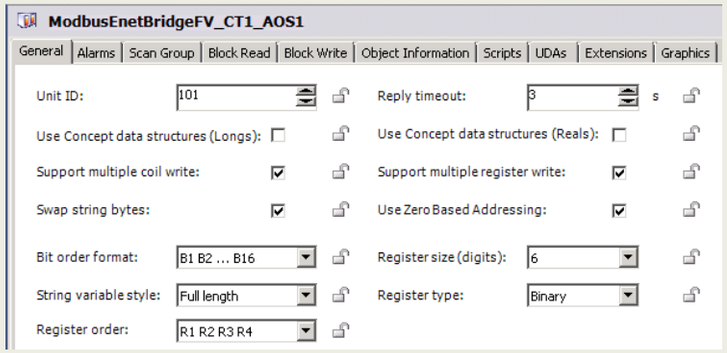
En esta plataforma se programan los objetos y los drivers de comunicaciones para acceder a los inversores a través del master de la instalación.

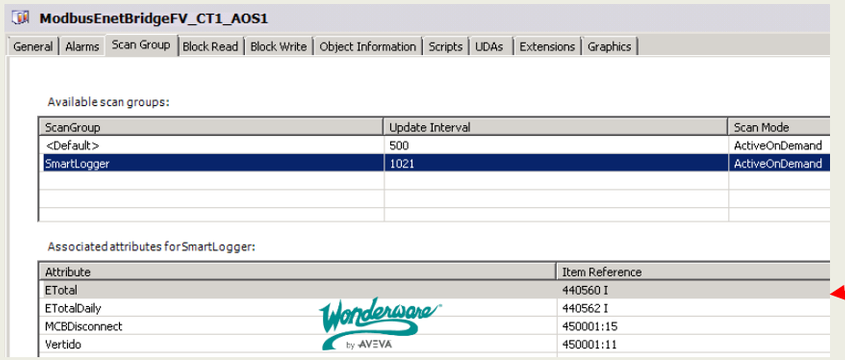




Drivers de Comunicaciones

Driver con la IP de la pasarela, puerto 502 y UID del esclavo del controlador Smartlogger o Inversor.





Energía activa total generada en el holding register 440.560 I
(2 Words = 32 bits)

Get & Save

Mapa de memoria SmartLogger 3000A

Las direcciones de cada magnitud están en los manuales de cada dispositivo en el apartado Mapa de memoria Modbus. Cada uno es diferente.

Para el caso del **Smartlogger 3000A de Huawei**

29	E-Total	RO	U32	kWh	10	40560	2	Equals the total energy yield generatedby all inverters.
30	E-Daily	RO	U32	kWh	10	40562	2	Equals daily energy yield generated byall inverters.
79	Alarm Info 2	RO	U16	N/A	1	50001	1	N/A

1103	MCB Disconnect	1	The general AC circuit breaker at the grid-tied point is OFF.	Major	50001	1
1107	DI1 custom alarm	1	The dry contact signal from the peripheral to the corresponding DI port on the SmartLogger is abnormal	Adaptable	50001	5

En las alarmas el bit lo leo invertido. Bit 1 será mi 15 y el bit 5 será mi 11 (electrical discharge)

Mapa de memoria Inversor 100 KTL

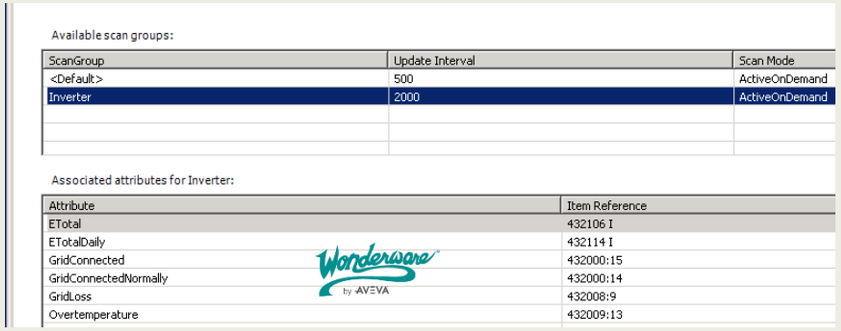
Para el caso del inversor **100 KTL**

48	Accumulated energy yield	RO	U32	kWh	100	32106	2	N/A
49	Daily energy yield	RO	U32	kWh	100	32114	2	N/A

12	State 1	RO	Bitfield16	N/A	1	32000	1

Bit 1: grid-connected
Bit 2: grid-connected normally

15	Alarm 1	RO	Bitfield16	N/A	1	32008	1	Alarm 1	7	Grid Loss
16	Alarm 2	RO	Bitfield16	N/A	1	32009	1	Alarm 2	3	Overtemperature


Available scan groups:

ScanGroup	Update Interval	Scan Mode
<Default>	500	ActiveOnDemand
Inverter	2000	ActiveOnDemand

Associated attributes for Inverter:

Attribute	Item Reference
ETotal	432106 I
ETotalDaily	432114 I
GridConnected	432000:15
GridConnectedNormally	432000:14
GridLoss	432008:9
Overtemperature	432009:13

invirtiendo bits:

bit 1 > 15

bit 2 > 14

bit 7 > 9

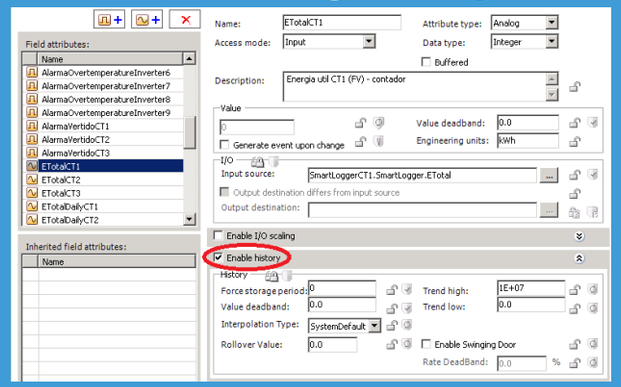
bit 3 > 13

Get & Save

Historización

Para historizar los datos en Historian basta con marcar "Enable history" en las señales adquiridas anteriores.

Esta configuración irá registrando el valor en Historian. Para poder obtener los datos de Historian se accederá mediante SQL Server. Para ello el sistema crea una BD llamada **Runtime**, varias tablas y vistas. En este caso accederemos a una vista llamada **Runtime.dbo.AnalogHistory** a través de su Tagname.



HISTORIAN-SQL

Con la aplicación QUERY de Historian podemos ver y filtrar estos datos. Al ser un SQL Server los datos son accesibles desde aplicaciones de terceros con consultas SQL → Node-RED.

Results

SQL | Data

	TagName	DateTime	Value
▶	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:05.62700	2602303
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:08.65700	2602303
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:11.68700	2602305
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:14.71800	2602305
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:17.74800	2602308
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:20.77800	2602308
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:23.80900	2602309
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:26.83900	2602309
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:29.86900	2602311
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:32.90000	2602311
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:35.93000	2602314
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:38.96000	2602314
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:41.99000	2602315
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:45.02100	2602318
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:48.05100	2602318
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:51.08100	2602319
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:54.11200	2602319
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:48:57.14200	2602322
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:49:00.17200	2602322
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:49:03.20300	2602323
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:49:06.23300	2602323
	InstalacionFotovoltaica.ETotalCT1	2023-04-29 13:49:09.26300	2602326

Node-RED

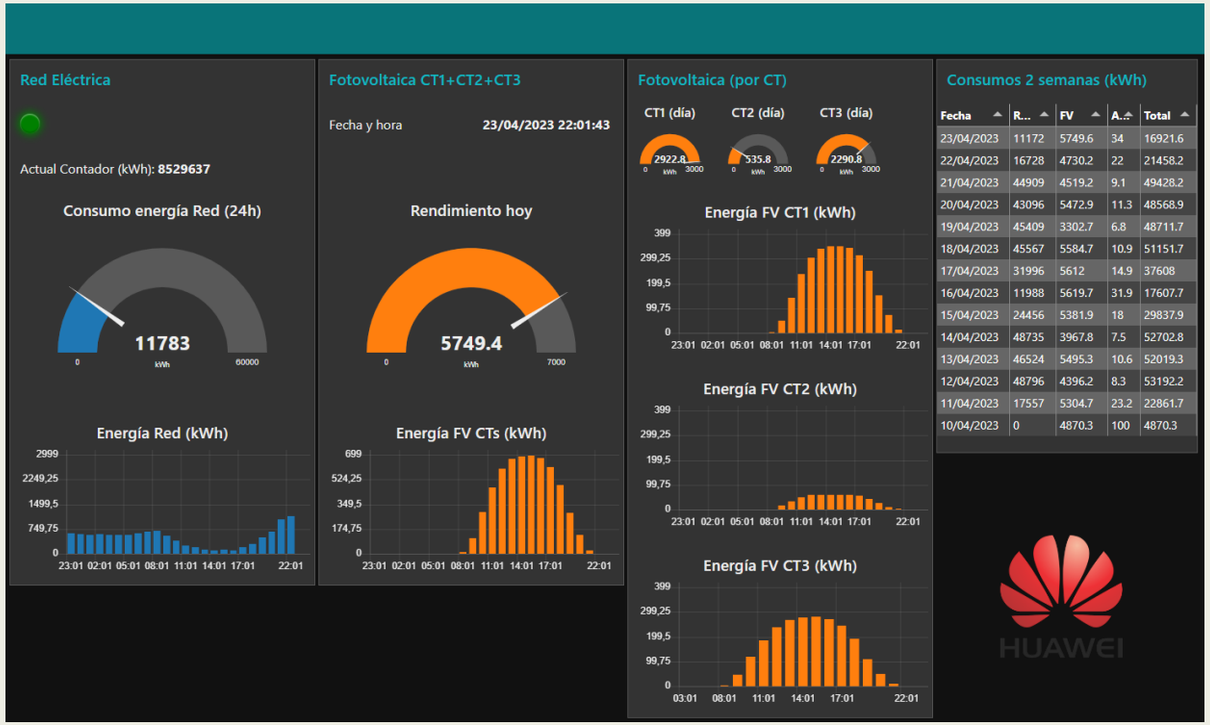
Node-RED

Para la representación gráfica utilizaremos Node-RED. Esta herramienta también es capaz de capturar el dato por MBTCP, incluso de historizarlo, por ejemplo, en una BD InfluxDB, MySQL o cualquier otra para series temporales.

En este caso es sólo UI para consultas en "tiempo real".

Se representan Gauges (color naranja) con la energía Producida por cada Centro de Transformación y la suma total de los 3. Valores de energía generada (kWh) por hora en el último día y tabla de totales por día.

Dashboard



GOODWE FOTOVOLTAICA GOODWE

GOODWE

Para poder generar energía para auto consumir se utilizan paneles solares, inversores (120K HT o 50K MT) y un sistema de control.

El sistema de control de la marca Goodwe y modelo SEC1000.

Comunicaciones Modbus

El inversor es el equipo encargado de transformar la corriente continua (DC) procedente de las baterías o de los paneles solares en corriente alterna (AC).

Se establece un bus Modbus RTU (bus serie RS-485) entre inversores cosidos uno a uno hasta el SEC1000 que comunica por Modbus TCP (para conectarlos a la red Ethernet).

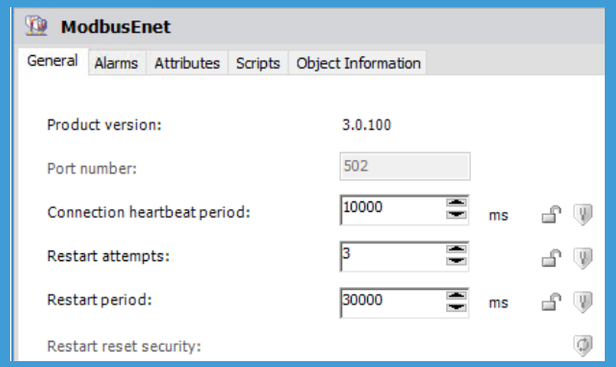
Al ser protocolo Modbus RTU cada inversor tiene un UID de esclavo de manera que son accesible a través de la dirección IP del controlador.

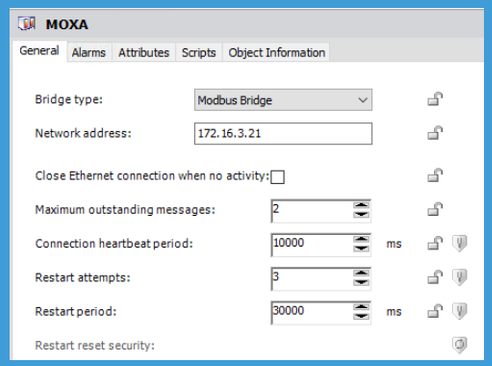
Get & Save

Adquisición

Para la adquisición de los datos nos basamos en una plataforma estándar llamada System Platform y para su historización en Historian de Wonderware.

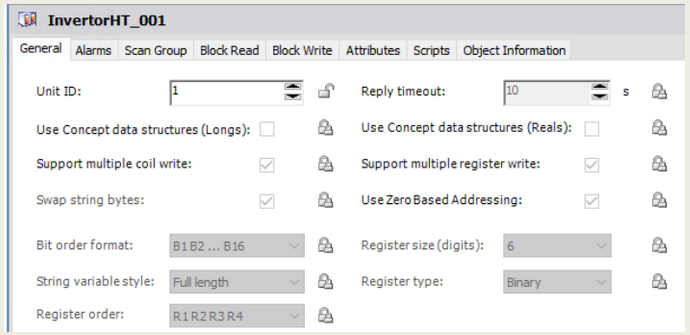
En esta plataforma se programan los objetos y los drivers de comunicaciones para acceder a los inversores a través del master de la instalación.



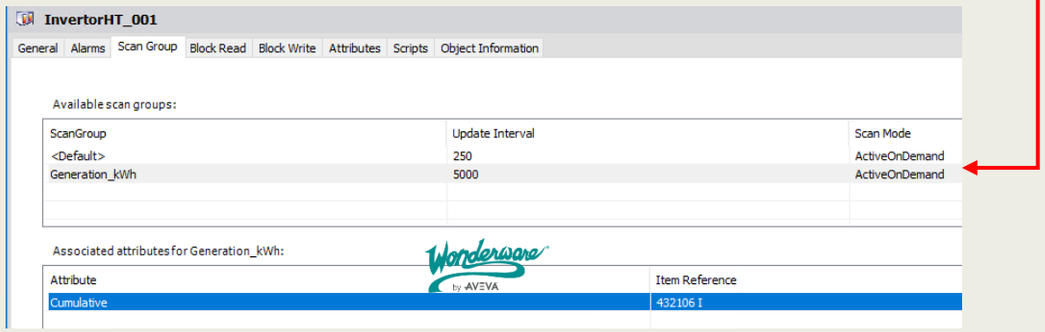


Drivers de Comunicaciones

Driver con la IP de la pasarela, puerto 502 y UID del esclavo del Inversor.



Energía activa total generada en el holding register 432.106 I
(2 Words = 32 bits)



Get & Save

Mapa de memoria Inverter 120 HT

Las direcciones de cada magnitud están en los manuales de cada dispositivo en el apartado Mapa de memoria Modbus. Cada uno es diferente.

Para el caso del **120 HT de Goodwe**

Address	Name	Read write	Type	Unit	Gain	Numb
32106	Cumulative Generation	RO	U32	kWh	100	2

Cumulative	432106 I

En el mapa de memoria la dirección está en decimal (directa a Wonderware)

Mapa de memoria Inverter 50 MT

Las direcciones de cada magnitud están en los manuales de cada dispositivo en el apartado Mapa de memoria Modbus. Cada uno es diferente.

Para el caso del **50 MT de Goodwe**

0312	*E-Total H*		*0.1KW.Hr*	*INT16U*	*R*		*Total Feed Energy to grid*
0313	*E-Total L*		*0.1KW.Hr*	*INT16U*	*R*		*Total Feed Energy to grid*

Cumulative	400786 I

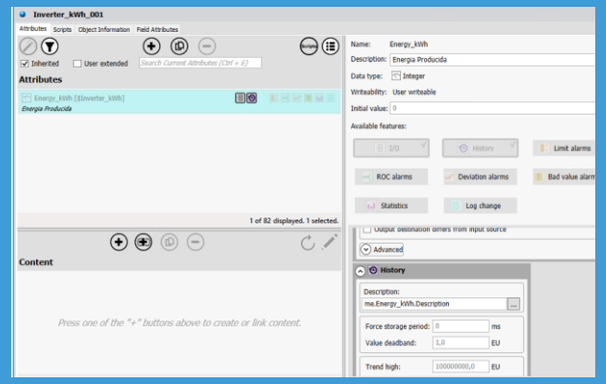
En el mapa de memoria la dirección está en hexadecimal (pasar a decimal para Wonderware)

Get & Save

Historización

Para historizar los datos en Historian basta con marcar "Enable history" en las señales adquiridas anteriores.

Esta configuración irá registrando el valor en Historian. Para poder obtener los datos de Historian se accederá mediante SQL Server. Para ello el sistema crea una BD llamada **Runtime**, varias tablas y vistas. En este caso accederemos a una vista llamada **Runtime.dbo.AnalogHistory** a través de su Tagname.



HISTORIAN-SQL

Con la aplicación QUERY de Historian podemos ver y filtrar estos datos. Al ser un SQL Server los datos son accesibles desde aplicaciones de terceros con consultas SQL → Node-RED.

Results

SQL | Data

TagName	DateTime	Value
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:06.11700	26082355
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:09.14700	26082355
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:12.17700	26082355
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:15.20700	26082357
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:18.23700	26082357
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:21.26700	26082357
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:24.29700	26082357
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:27.32700	26082357
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:30.35700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:33.38700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:36.41700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:39.44700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:42.47700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:45.50700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:48.53700	26082359
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:51.56700	26082361
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:54.59700	26082361
Inverter_kWh_001.Energy_kWh	2023-11-15 15:41:57.62700	26082361
Inverter_kWh_001.Energy_kWh	2023-11-15 15:42:00.65700	26082361
Inverter_kWh_001.Energy_kWh	2023-11-15 15:42:03.68700	26082361
Inverter_kWh_001.Energy_kWh	2023-11-15 15:42:06.71700	26082363
Inverter_kWh_001.Energy_kWh	2023-11-15 15:42:09.74700	26082363
Inverter_kWh_001.Energy_kWh	2023-11-15 15:42:12.77700	26082363

Node-RED

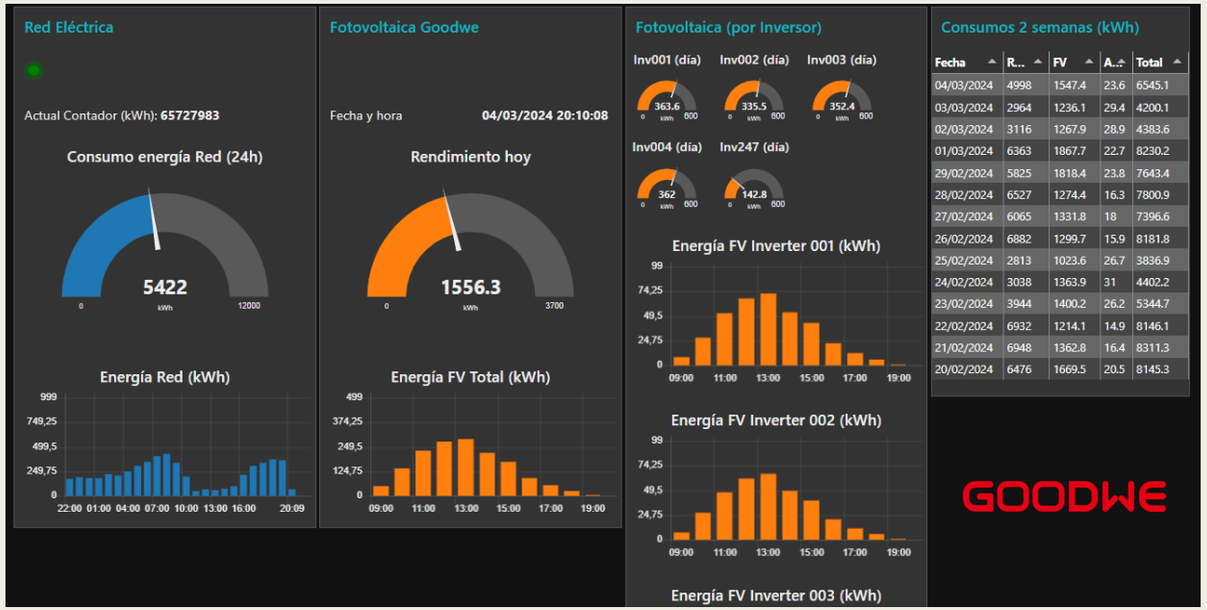
Node-RED

Para la representación gráfica utilizaremos Node-RED. Esta herramienta también es capaz de capturar el dato por MBTCP, incluso de historizarlo, por ejemplo, en una BD InfluxDB, MySQL o cualquier otra para series temporales.

En este caso es sólo UI para consultas en "tiempo real".

Se representan Gauges (color naranja) con la energía Producida por cada INVERTER de y la suma total de los 5. Valores de energía generada (kWh) por hora en el último día y tabla de totales por día.

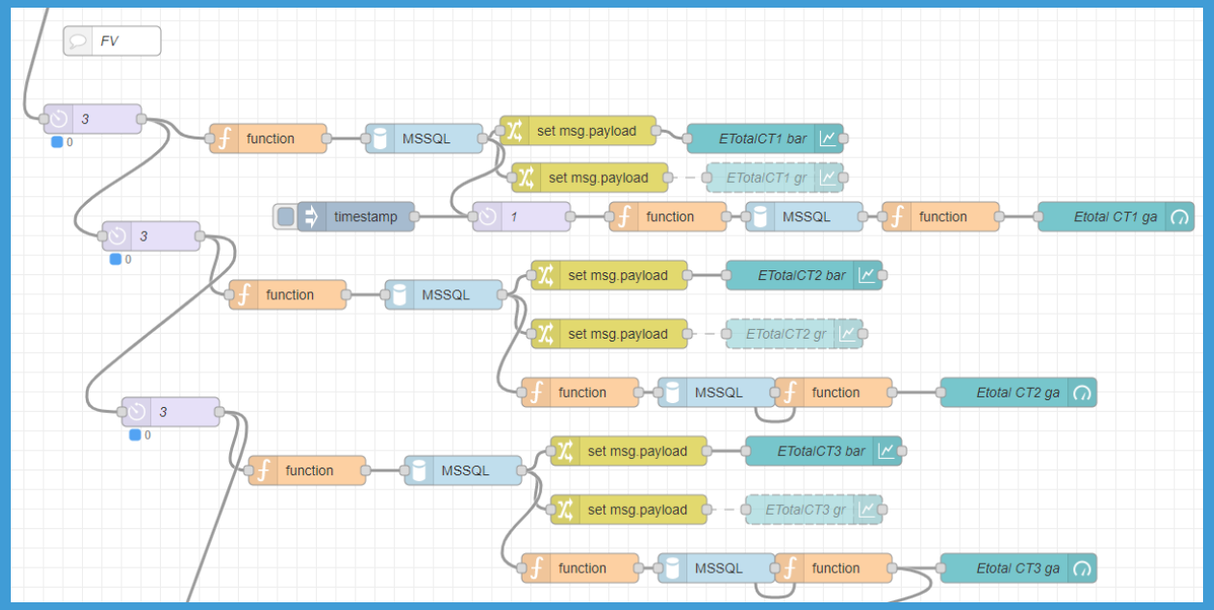
Dashboard



Node-RED

Node-RED

Se representa parte del flujo. En este caso 3 gauges y 3 gráficas de barras por hora.

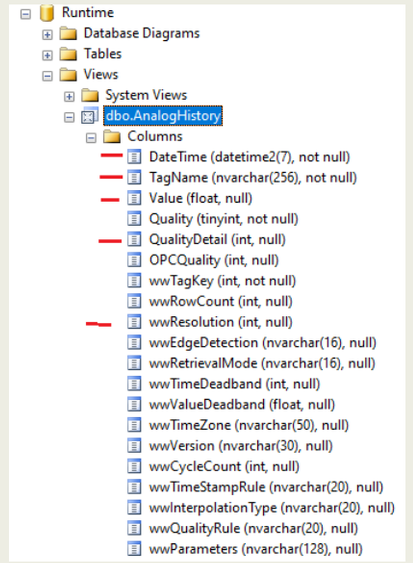


HISTORIAN-SQL

Para ello se utilizan nodos Function para hacer las queries SQL y pasarlas al nodo MSSQL.

Los resultados se filtran para sacar el dato en cuestión, es decir, la suma total o los datos por hora en un periodo dado (las últimas 24h).

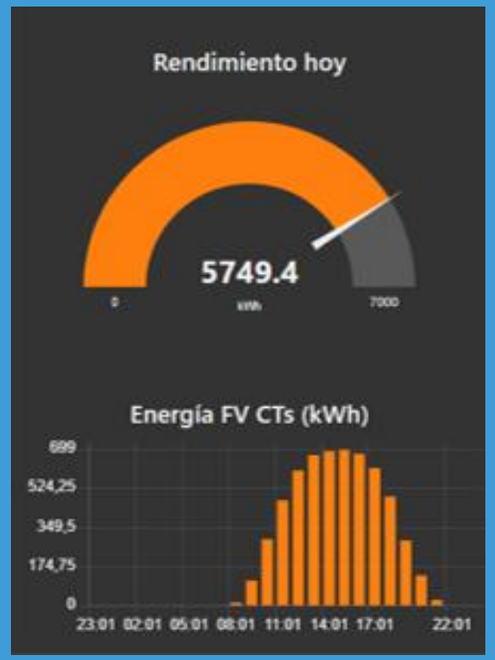
Las queries están basadas en el origen de datos, como se ha dicho antes, **Runtime.dbo.AnalogHistory**.



Node-RED

Node-RED

Un detalle clave de una query para dar el rendimiento (producción) de un día por horas y graficarlo.



HISTORIAN-SQL

Query y las claves para entenderla:

```
select t.fechaHora as x, round(t.kWh*0.1, 2) as kWh, round(t.SaltokWh*0.1, 2) as y from (SELECT DateTime as fechaHora, ROUND(Value, 2) as kWh, ROUND(Value, 2) - LAG(ROUND(Value, 2)) over(order by Datetime) as SaltokWh FROM Runtime.dbo.AnalogHistory where Tagname = 'InstalacionFotovoltaica.ETotalCT1' and DateTime >= DateAdd(hh, -24, GetDate()) and DateTime <= GetDate() and wwresolution = 3600000 and QualityDetail = 192) as t where t.SaltokWh is not null
```

- Se obtienen datos de cada hora con **wwresolution** = 3600000
- Se calcula el incremento de la hora actual respecto la anterior con **LAG()**.

	x	kWh	y
1	2023-04-28 16:05:52.8630000	258745,5	318
2	2023-04-28 17:05:52.8630000	259038,5	293
3	2023-04-28 18:05:52.8630000	259221,2	182,7
4	2023-04-28 19:05:52.8630000	259314,8	93,6
5	2023-04-28 20:05:52.8630000	259350,7	35,9
6	2023-04-28 21:05:52.8630000	259358,9	8,2
7	2023-04-28 22:05:52.8630000	259358,9	0
8	2023-04-28 23:05:52.8630000	259358,9	0
9	2023-04-29 00:05:52.8630000	259358,9	0
10	2023-04-29 01:05:52.8630000	259358,9	0
11	2023-04-29 02:05:52.8630000	259358,9	0
12	2023-04-29 03:05:52.8630000	259358,9	0
13	2023-04-29 04:05:52.8630000	259358,9	0
14	2023-04-29 05:05:52.8630000	259358,9	0
15	2023-04-29 06:05:52.8630000	259358,9	0
16	2023-04-29 07:05:52.8630000	259358,9	0
17	2023-04-29 08:05:52.8630000	259372,7	13,8
18	2023-04-29 09:05:52.8630000	259431,8	59,1
19	2023-04-29 10:05:52.8630000	259582,3	150,5
20	2023-04-29 11:05:52.8630000	259803,5	221,2
21	2023-04-29 12:05:52.8630000	259950,2	146,7
22	2023-04-29 13:05:52.8630000	260126,9	176,7
23	2023-04-29 14:05:52.8630000	260259,7	132,8
24	2023-04-29 15:05:52.8630000	260421,6	161,9

Node-RED

Node-RED

Después de la query anterior se quiere graficar el consumo del día anterior por horas.

Rules

Set msg. payload

to the value JSON [{ "series": ["ETotalCT1"], "data": [[msg.payload.SaltokWh]], "labels": [msg.payload.fechaHora] }]

```
[
  {
    "series": ["ETotalCT1"],
    "data": [[msg.payload.SaltokWh]],
    "labels": [msg.payload.fechaHora]
  }
]
```

Group [Fotovoltaica] Fotovoltaica (por CT)

Size auto

Label Energía FV CT1 (kWh)

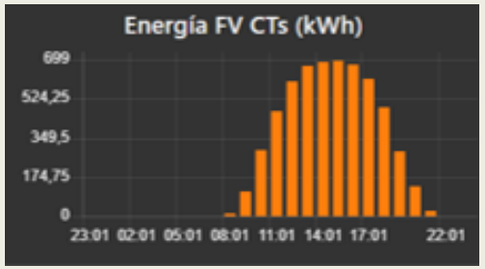
Type Bar chart

Y-axis min 0 max 399

HISTORIAN-SQL

Resultando una Bar Chart como la siguiente.

Es realmente interesante saber el consumo por hora y ver la tendencia de consumo durante el día.



La suma de cada barra de la gráfica daría el área completa del consumo total.

WALLBOX

wallbox

Podemos leer datos de los cargadores de vehículos eléctricos (EV) Cooper SB OCPP Socket Tipo 2 22 kW (400V-32A) + poste Eiffel de Wallbox

Plataforma myWallbox

Dispone de una plataforma web donde puedes gestionar los cargadores, monitorización en tiempo real, pagos, reporting, etc.

OCPP

OCPP 1.6 (Open Charge Point Protocol)

En este capítulo daremos unas pinceladas sobre el protocolo OCPP.

https://www.oasis-open.org/committees/download.php/58944/ocpp-1.6.pdf

¿Qué es **OCPP**? Es un protocolo estándar de aplicación abierto el cual permite a las estaciones de carga de vehículos eléctricos y los sistemas centrales de gestión de distintos fabricantes comunicar unos con otros.

Las versiones de este protocolo son:

- 1.5 → sólo soporta SOAP (WebServices)
- 1.6 → soporta SOAP y JSON (formato mensajes)
- 2.01 → mejoras respecto a la anterior versión

Partes del sistema

Las partes que intervienen en este sistema son:

- Charge Point (CP)
- Central System (CS)
- Vehículo (EV)

Entre vehículo y estación de carga (conexión eléctrica). Entre estación de carga y sistema central (conexión de comunicaciones de red) → OCPP

OCPP

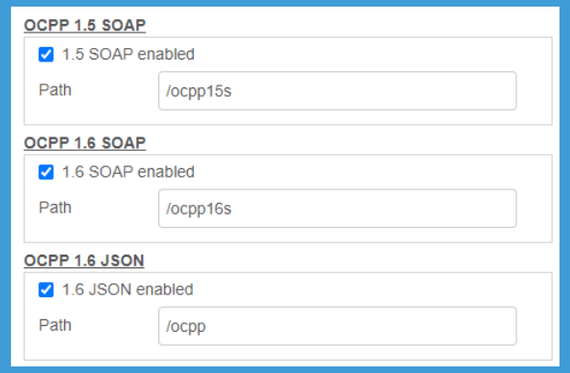
OCPP → Node-RED

Veremos detalles de este protocolo a través de la herramienta multiusos Node-RED.

Para ello he seguido el ejemplo que viene con la librería **node-red-contrib-ocpp**.

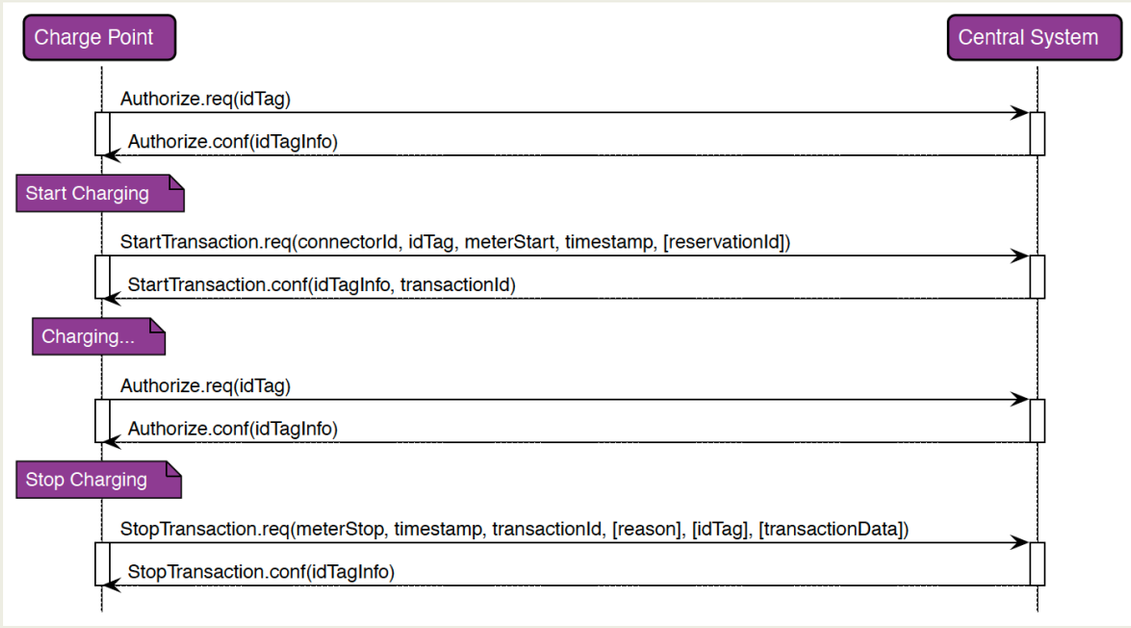
https://flows.nodered.org/node/node-red-contrib-ocpp

Soporta OCPP 1.5 y 1.6 en SOAP y 1.6 en JSON



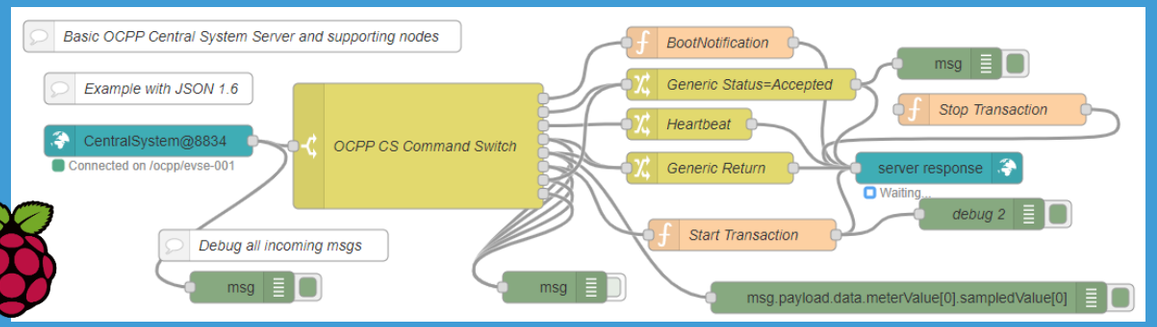
Secuencia básica

Diagrama de secuencia de un ejemplo simple de inicio y fin de una transacción entre CP y CS.

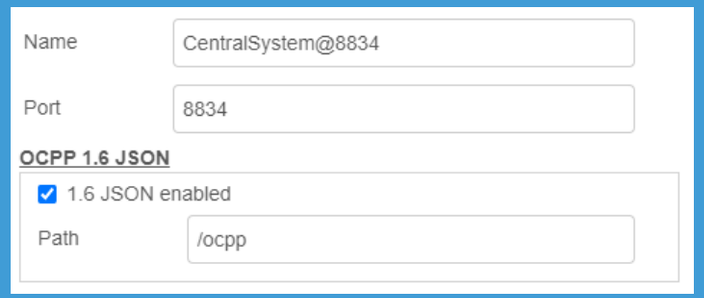


OCPP

Central System (Server)

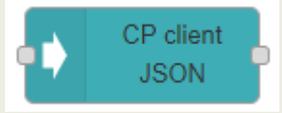


Consta del nodo CS Server, escuchando por el puerto TCP **8834** en la URL **/ocpp**

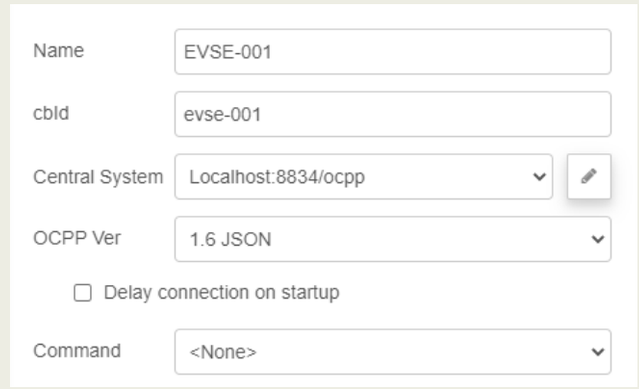


EVSE (CP) Client

Consta del nodo CP client JSON apuntando al CS. El comando y valores se pasarán por nodo Function.



Web socket ws://



OCPP

Meter Values command

De todas las operaciones que puede realizar un Punto de Carga nos centraremos en el envío de métricas al Sistema Central.

El estándar dice que cada elemento **MeterValue** contiene un **timestamp** y un conjunto de 1 o varios **sampledvalue.** A su vez, cada sampledvalue contiene un **value** y como opcional (**measurand**, context, location, **unit**, phase, format).

Usaremos notificación JSON en versión 1.6 de OCPP.

Para generar el mensaje he utilizado un nodo Function de Node-RED (timestamp y value dinámicos).

JSON CP→CS

JSON MeterValues

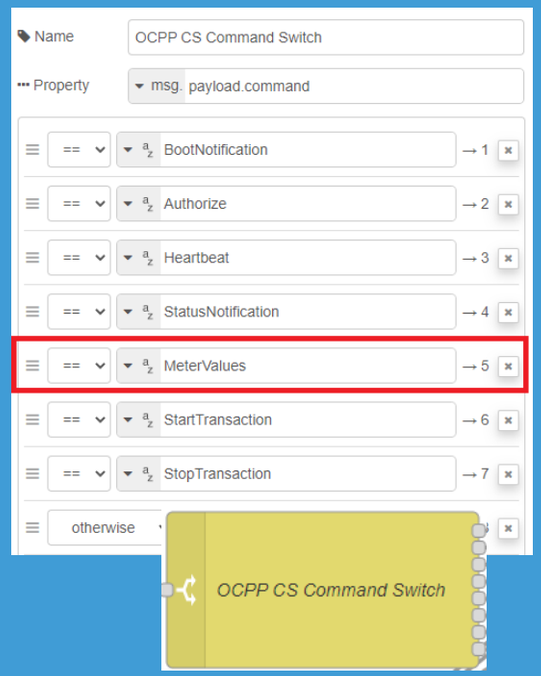
```
msg.payload = {
    "command": "MeterValues",
        "data": {
        "connectorId": 1,
            "meterValue": [
                {
                    "timestamp": new Date().toISOString(),
                    "sampledValue": [
                        {
                            "value": getRndInteger(0, 200),
                            "unit": "kWh",
                            "measurand": "Energy.Active.Import.Register"
                        }
                    ]
                }
            ]
    }
}
return msg;


function getRndInteger(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

OCPP

Datos recibidos en CS

La estación central recibe el JSON en su URL en el puerto 8834 y decodifica el mensaje.

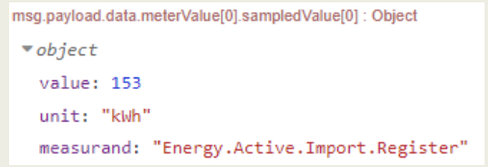
?Qué llega?



JSON CP→CS

Llega un msg con 2 campos principalmente, ocpp y payload acorde al protocolo OCPP 1.6.

- **ocpp** → define la identidad del chargeBox (**cbId**) y el **command**.
- **payload** → contiene el campo data. Este contiene el **connectorId** puede haber varios para un mismo CP y el campo **meterValue**.
  - Dentro del array meterValue:
    - **timestamp** (fecha del dato)
    - Array **sampledValue** con las métricas. En este caso 1 con **value**, **unit**, **measurand** (Energy.Active.Import.Register, 153 kWh)

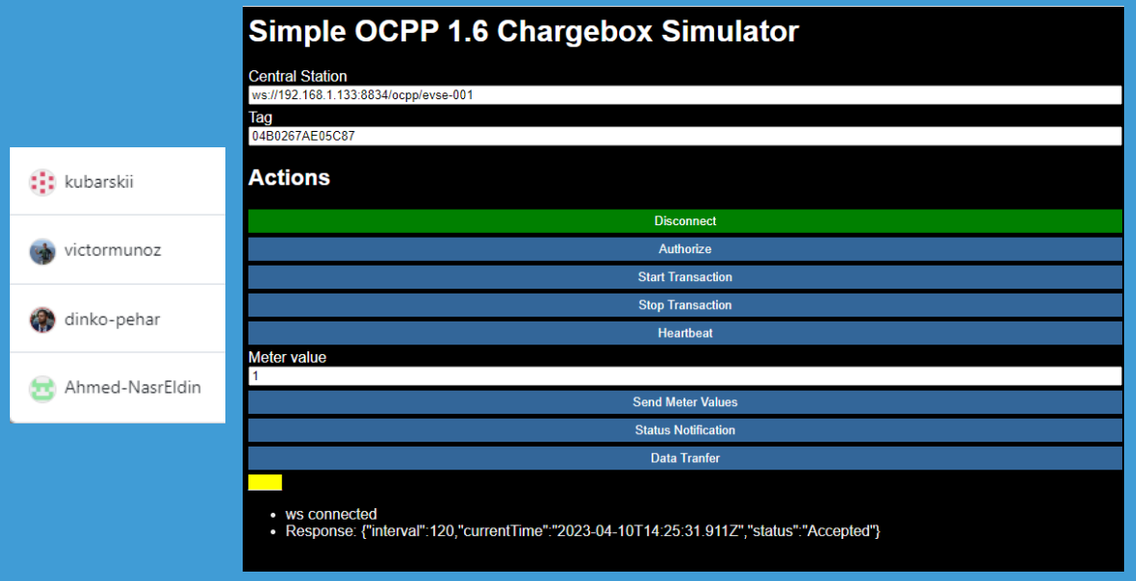


OCPP

OCPP-1.6 Chargebox Simulator (cliente)

Para no partir de 0, utilizo un simulador cliente creado por @Kubarskii, mejorado por @Víctor Muñoz y otros. Lo puedes descargar en la siguiente URL de github: https://github.com/victormunoz/OCPP-1.6-Chargebox-Simulator/

Es un HTML con los comandos del protocolo OCPP.



BootNotification command

Para conectarnos por WebSocket ponemos la siguiente URL donde tengamos la CS.

ws://192.168.1.133:8834/ocpp/evse-001

Para conectar el CP (cliente) envía al CS (servidor en Node-RED) un comando llamado **BootNotification** y el servidor responde con el intérvalo de heartbeats, fecha y un Accepted como status.

Lado servidor

Lado cliente

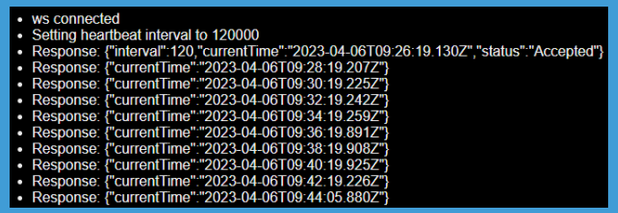
OCPP

Heartbeat command

Una vez conectado el CP (cliente) al CS (servidor) va enviado **Heartbeats** y el servidor va respondiendo esos heartbeats cada 2 minutos.

Lado servidor

Lado cliente

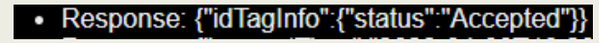


Authorize command

Para petición de autorización se envía:

Lado servidor

Lado cliente



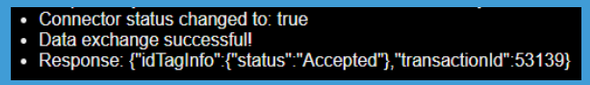
OCPP

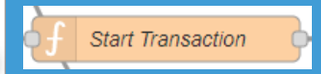
StartTransaction command

Una vez autorizado, en cliente empieza una transacción con StartTransaction. El CS asigna una transactionId.

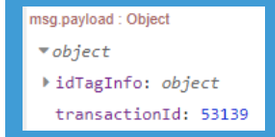
Lado servidor

Lado cliente






```
msg.payload = {

    idTagInfo: {
        status: "Accepted"
    },
    transactionId: getRndInteger(1, 100000)
}
return msg;

function getRndInteger(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

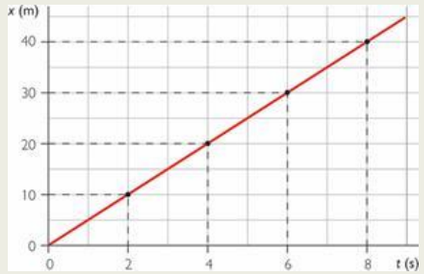


MeterValues command

El CP va enviando la lectura al CS. A medida que cargue los datos se van enviando en tiempo real.

Lado servidor

Lado cliente



OCPP

Node-RED

StopTransaction command

Una vez acabada la carga del vehículo, en cliente acaba la transacción con StopTransaction donde se indica el último valor de la medida. En este caso 20.

Lado servidor

msg : Object
object
ocpp: object
chargeBoxIdentity: "evse-001"
MessageId: "kgOwBy8O1neVLjxgeYRkYsfA8NAs6PYz3J4t"
msgType: 2
command: "StopTransaction"
payload: object
command: "StopTransaction"
data: object
transactionId: "[object Object]"
idTag: "04B0267AE05C87"
timestamp: "2023-04-10T16:38:45Z"
meterStop: 20
msgId: "52f77a4b-98d1-49b7-96a1-97aeba933f6b"
_msgid: "ae4efa87eec41445"

Lado cliente

- Connector status changed to: false
- Response: {"idTagInfo":{"status":"Accepted"}}

Status Notification / Data Transfer commands

En Status Notification el cliente (CP) envía el status (Available, Charging, etc.). En DataTransfer el cliente (CP) envía datos del sistema (VendorId, MessageId, etc.)

Lado servidor

msg : Object
object
ocpp: object
chargeBoxIdentity: "evse-001"
MessageId: "kgOwBy8O1neVLjxgeYRkYsfA8NAs6PYz3J4t"
msgType: 2
command: "StatusNotification"
payload: object
command: "StatusNotification"
data: object
connectorId: 2
status: "Available"
errorCode: "NoError"
info: ""
timestamp: "2023-04-10T16:42:58Z"
vendorId: ""
vendorErrorCode: ""

msg : Object
object
ocpp: object
chargeBoxIdentity: "evse-001"
MessageId: "kgOwBy8O1neVLjxgeYRkYsfA8NAs6PYz3J4t"
msgType: 2
command: "DataTransfer"
payload: object
command: "DataTransfer"
data: object
vendorId: "rus.avt.cp"
messageId: "GetChargeInstruction"
data: ""
msgId: "901b780a-866b-4007-8383-eb435e5977c5"
_msgid: "1dc42ba6208e792a"

response

Value
Accepted
Rejected
UnknownMessageId
UnknownVendorId

States considered Operative are: *Available, Preparing, Charging, SuspendedEVSE, SuspendedEV, Finishing, Reserved.* States considered Inoperative are: *Unavailable, Faulted.*

Enlaces recomendados

Enlaces recomendados

Circutor

https://docs.circutor.com/docs/M001B01-01.pdf

https://docs.circutor.com/docs/M98206501-03.pdf

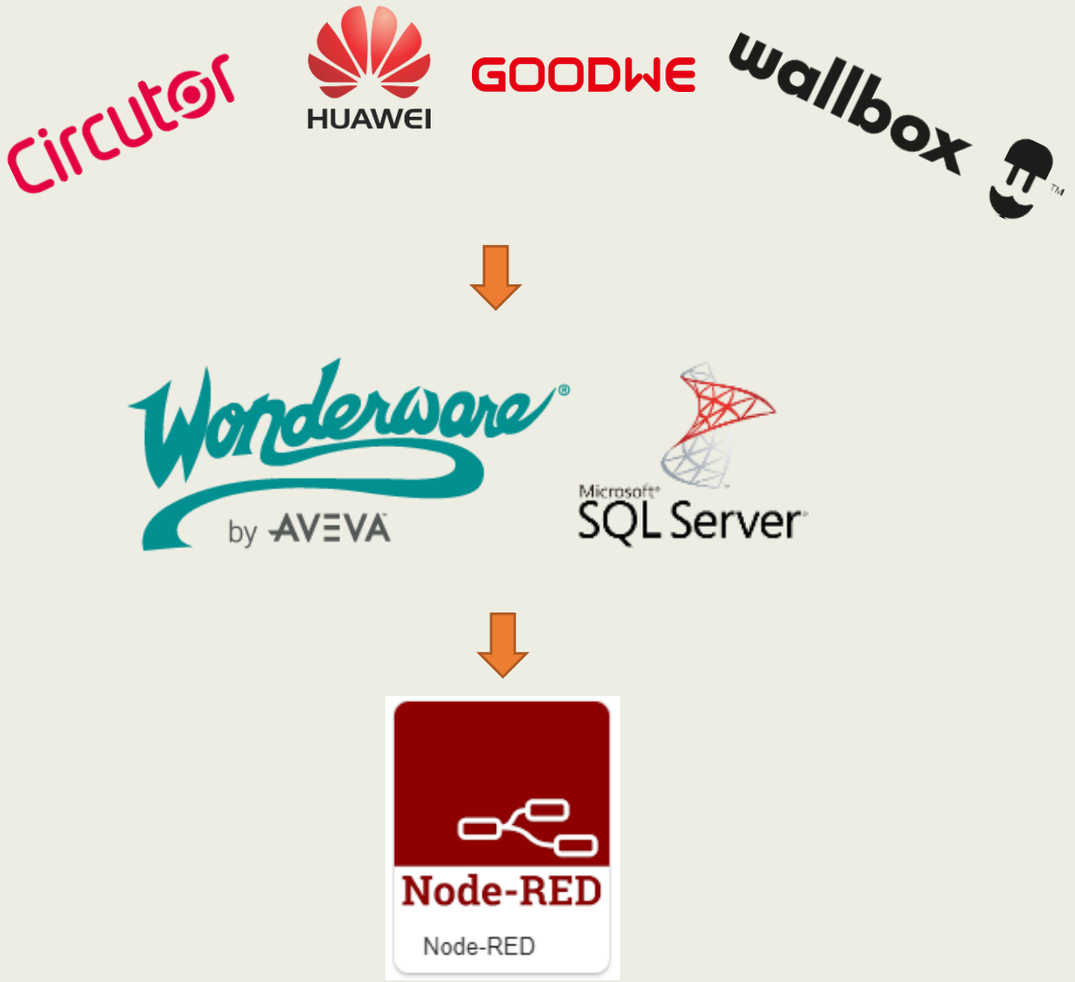
https://docs.circutor.com/docs/M98174001-03.pdf

Huawei

https://support.huawei.com/enterprise/en/doc/EDOC1100050690

OCPP 1.6 (protocolo)

https://www.oasis-open.org/committees/download.php/58944/ocpp-1.6.pdf



ENERGY MEASUREMENT
v.1.2 MARZO 2024

https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099

https://www.linkedin.com/company/gdo-electric1996/

RICARDO MORALEDA GARETA

GDO SOFTWARE