

# **CAPÍTULO 17. Energy Measurement** v.1.2 MARZO 2024

### Ricardo Moraleda Gareta

[Director departamento de software de GDO Software]





<u>Píldoras Tecnológicas © 2024 by Ricardo Moraleda Gareta</u> is licensed under <u>CC BY-NC-ND 4.0</u>



# ENERGY MEASUREMENT 7

Energy	Sistemas a medir
En este capítulo veremos 3 sistemas a medir la energía consumida / generada:	Energía consumida medida con dispositivos de Circutor por el protocolo MODBUS TCP.
<ul> <li>Energía consumida en fábrica medida en Transformadores.</li> </ul>	Circutor
<ul> <li>Energía generada por el sistema de Fotovoltaica y auto consumida en fábrica (sin volcar a red compañía)</li> </ul>	<ul> <li>Energia generada leida de Smarflogger 3000 de Huawei o SEC1000 de Goodwe por el protocolo MODBUS TCP.</li> </ul>
	HUAWEI
<ul> <li>Energía consumida por carga de vehículos eléctricos</li> </ul>	<ul> <li>Energía consumida leída de los cargadores Wallbox por el protocolo OCPP.</li> <li>wallbox T</li> </ul>



# 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 C10

Circutor CVM K2



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.





**MBTCP** Circutor line-TCPRS1

Modbus RTU

Modbus TCP







# Adquisición

# Drivers de Comunicaciones

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.

ModbusEnet_ENERGIA_AOS1	MBEB_EN	NEI	
General Alarms Object Information Scri	ots UDAs Extensions Graphics	General Alarms	0
		Bridge type:	
Product version:	3.0.100	Network addre	ress
Port number:	502		
, ore name en		Close Etherne	et co
Connection heartbeat period:	10000 🖻 ms 🗗 💚	Maximum outs	sta
Restart attempts:	3 🚊 🗗 🕅	Connection he	eart
Destart a srie de		Restart attemp	ipts
Restart penou:	50000 🖃 ms 🖻 🕅	Restart period	d:
Restart reset security:	(J)	Restart reset e	sec
•			

MBEB_ENER_CT123	0_AOS1 *				
ieneral Alarms Object Infor	mation Scripts	UDAs Extensio	ns   Graph	nics	
Bridge type:	Modbus Brid	ge 💽	·	£	
Network address:					
Close Ethernet connection when no activity:					
Maximum outstanding mes	sages:	2		6	
Connection heartbeat perio	d:	10000	ms	ъ.	
Restart attempts:		3		4 V	
Restart period: 30000 🚍 ms 🔐 🖤					
Restart reset security:				Q	

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

ModbusEnetBridgeEnerCT1_2_AOS1						
General Alarms Scan Gr	roup 🛛 Block Read 🗍 Block '	Write 🛛 🕻	Object Information Script	s UDAs Ext	ensions 🗍	Graphics
Unit ID:	2	ſ	Reply timeout:	3	S S	ſ
Use Concept data stru	ctures (Longs): 🔲	84	Use Concept data struc	tures (Reals):		84
Support multiple coil w	vrite: 🔽	ſ	Support multiple regist	er write:	•	£
Swap string bytes:		ſ	Use Zero Based Addres	sing:	×	8
Bit order format:	B1 B2 B16 💌	ſ	Register size (digits):	6	•	ſ
String variable style:	Full length	ſ	Register type:	Binary	•	£
Register order:	D1 D2 D2 D4	-0				

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

#### ModbusEnetBridgeEnerCT1\_2\_AOS1

General Alarms Scan Group Block Read Block Write Object Information Scripts UDAs Extensions Graphics

ScanGroup	Update Interval	Scan Mode
<default></default>	500	ActiveOnDemand
Energia	500	ActiveOnDemand
Associated attributes for Energia:		
Attribute	Ite	m Reference
Epergia@ctivaConcumida	40	D060 T







### Mapa de memoria CVM Mini

# Mapa de memoria CVM K2/C10

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 l 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.

IAGNITUD	SIMBOLO	Instantáneo	Uds.
nergía Activa	kW∙h III	3C-3D	w∙h

Attribute	1 underware	Item Reference
EnergiaActivaConsumida	by AVEVA	400060 I

### $CVM K2 \rightarrow 5D8-5D9 (HEX) = 1496-1497 (DEC)$

1 10 10 10 10 10 10 10 10 10 10 10 10 10	TOTAL TARIFF			
40.00	Active energy	kW·h III	201	5D8-5D9
00000				

Attribute	1 Jonderware	Item Reference
EnergiaActivaConsumida	ty AVEVA	401496 I

### **CVM C10** → DC-DD (HEX) = 220-221 (DEC)

		1	Disco	
Ĩ	- C C C C	215	18 13 15	
4		=	>	H
- 4			5	

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

Attribute	Worderware	Item Reference
EnergiaActivaConsumida	by AVEVA	400220 I







# Historización

# **HISTORIAN-SQL**

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.



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  $\rightarrow$  Node-RED.



Res	ults		
QL	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









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

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.

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











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



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**.











B HISTORIAN-SQL

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



### 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	у
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









HISTORIAN-SQL

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

set msg.payload ETotalCT1 red bar		_			
	set msg.payload	<b>}</b> -	-0	ETotalCT1 red bar	<b>_</b>

<b>≣</b> Rules	
E Set  to the value	msg. payload     J; [ { "series": ["T1234_CT1"], "data": [[msg.payload.SaltokWh]], "labels": [msg.payload.fechaHora] } ]
I Group	[Consumo Eléctrico (CTs)] Red Eléctric V "series": ["T1234_CT1"],
ច្រាំ Size	auto
£ Label	Energía Red CT1 (kWh)
🛃 Туре	<u>Iul</u> Bar chart ✓
Y-axis	min 0 max 2999

Resultando una Bar Chart como la siguiente.

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

2999
2249,25
1499,5
749,75
13:13 16:13 19:13 22:13 01:13 04:13 07:13 12:13

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









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.



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).

Comunicaciones **Modbus** 

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.







Modbus RTU

Modbus TCP







**Adquisición** 

# Drivers de Comunicaciones

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.

ModbusEnet FV AOS1	🚺 MBEI
constitute for a second for a first to the	General A
General Alarms   Object Information   Scripts   UDAs   Extensions   Graphics	
	Bridge t
Product version: 3.0.100	Network
502	
Port number:	Close Et
Connection heartbeat period:	Maximur
Restart attempts: 3 🚍 🔐 🕡	Connect
	Restart
Restart period: BUUUU 🚍 ms 🗗 🔰	
Bestart coast accurity	Restart
Restart reset security:	Restart

MBEB_FV_CT1_A0S1 *										
General Alarms Object Info	Graphics									
Bridge type:	Modbus Bridge	6								
Network address:	IP	6								
Close Ethernet connection	6									
Maximum outstanding mes	sages: 2	6								
Connection heartbeat peri	od: 10000 🚍	ms 占 🔍								
Restart attempts:	3	£ ())								
Restart period:	30000	ms 占 🔍								
Restart reset security:		Ģ								

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

General       Alarms       Scan Group       Block Read       Block Write       Object Information         Unit ID:       101       Image: Concept data structures (Longs):       Image: Concept data structures (Lon	n Scripts UDAs Extensions Graphics t: 8 2 3 6 data structures (Reals): 6 6 ple register write: 7 6 cd Addressing: 7 6 (digits): 6 7
Unit ID:       101       Image: Concept data structures (Longs):	t: β 🚍 s 🗗 data structures (Reals): □ 🗗 ple register write: 🔽 🗗 ed Addressing: ☑ 🗗 (digits): 6 ☑ 🗗
Use Concept data structures (Longs):  Use Concept Use	data structures (Reals):  ple register write:  d Addressing:  (digits):  6
Support multiple coil write:       Image: Constraint of the second	ple register write:  C
Swap string bytes:     Image: Constraint of the string strin	ed Addressing: 🔽 🗗 (digits): 6 🔽 🗗
Bit order format: B1 B2 B16  ☑	(digits): 6
String variable style: Full length   Register type	
	: Binary 🔽 🗗
Register order: R1 R2 R3 R4 🗾 🖆	
ModbusEnetBridgeFV_CT1_AOS1	
eneral Alarms Scan Group Block Read Block Write Object Information Script	s UDAs Extensions Graphics
Available scan groups:	
ScanGroup Update Interv	/al
<default> 500</default>	
SmartLogger 1021	

Energía activa total generada en el holding register 440.5601 (2 Words = 32 bits)

Modi	busEn	etBridgeF	V_CT1_A	051										
ral A	larms	Scan Group	Block Read	Block Write	Object Information	n Scripts	UDAs	Extensions	Graphics					
Avail	lable s	can groups:												
5can(	Group				Upda	ate Interva	al					Scan Mode		
<def< td=""><td>ault&gt;</td><td></td><td></td><td></td><td>500</td><td></td><td></td><td></td><td></td><td></td><td></td><td>ActiveOnDemand</td></def<>	ault>				500							ActiveOnDemand		
SmartLogger					1021	1021						ActiveOnDemand		
Asso	ociated	attributes fo	rSmartLogge	er:										
Attribute								Ite	m Referenc	е				
ETotal								44	0560 I					
ETotalDaily 1.Jondon				1 Janderwa	440562 I									
MCBD	isconn	ect			No. Com			45	0001:15					
Vertido					by AVEVA			45	450001-11					









## 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

9	E-7	Total	RO	U32	kWh	10	40560	2	Equals the tota yield generate inverters.	uals the total energy ld generatedby all rerters.	
0	E-Daily		RO	U32	kWh	10	40562	2	Equals daily energy yield generated byall inverters.		
9	Ala	arm Info 2	RO	U16	N/A	1	50001	1	N/A		]
1103			MCB Disconne	ect	1	The ger grid-tie	neral AC circ ed point is OI	cuit breaker at t FF.	the Major	50001	1
		1107 DII custom 1 alarm			The dry periphe port on	y contact sign eral to the con the SmartLo	nal from the responding DI ogger is	Adaptabl e	50001	5	
						abnorm	nal				

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



Available scan groups:				invirtiendo bit	
ScanGroup	Update Interval		Scan Mode		
<default></default>	500		ActiveOnDemand	bit 1 > 15	
Inverter	2000		ActiveOnDemand		
Associated attributes for Inverter:				bit 2 > 14 bit 7 > 9	
Attribute		Item Reference			
ETotalDaily		432106 I 432114 I		トは 2 、 1 2	
GridConnected	1. Inderware	432000:15		DII 3 ~ 13	
GridConnectedNormally		432000:14			
GridLoss	by AVEVA	432008:9			
Overtemperature		432009:13			







# Historización

# HISTORIAN-SQL

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.

	💶 + 💽 + 🗙		Name:	ETot	alCT1			Attribute type:	Analog	-		
Fie	ld attributes:		Access mode:	Inpu	t	•		Data type:	Integer	•		
	Name							Buffered				
л	AlarmaOvertemperatureInverter6		Description	Foet	cia util CT1 (	Elú - co	otador					
п	AlarmaOvertemperatureInverter7		beschption.			,					ി	
л	AlarmaOvertemperatureInverter8		- Value	I						×		
л	AlarmaOvertemperatureInverter9		value	_		0.23	a	Male and a discord.	0.0			12
п	AlarmaVertidoCT1		0			<b>□</b> . 3	P	value deadband:	0.0		Ξ.	3
л	AlarmaVertidoCT2		Generate ev	ent up	on change	£ ()	1	Engineering units:	k₩h		d'	
<u>n</u>	AlarmaVertidoCT3		-1/0									
2	ETotalCT1		Input source:		Smartl occu	arCT1 S	markl or	ner ETotal			-0	3
2	ETotalCT2		_		pinarcogg	01011.0	marceogy	pricitocal				
2	ETotalCT3		Cutput dest	ination	differs from	input so	urce			_	d'	
2	ETotalDailyCT1		Output destination	ation:							63	R
2	ETotalDailyCT2									_		
Inh	erited field attributes:	1	Enable I/O sc	aling							۷	
	Name	S	<ul> <li>Enable history</li> </ul>	0							\$	
			History 60	1								
			Force storage	period	0		af 🐨	Trend high:	1E+07		8	٩
			Value deadbar	nd-	0.0		0.00	Trend low:	0.0		1	100
				_								
			Interpolation	lype:	SystemDef	ault 💌	9.0					
			Rollover Value		0.0		£ 0	🗌 Enable Swingin	g Door		ď	đ
								Rate DeadBand:	0.0	- %	ŝ	đ
									P		-	

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  $\rightarrow$  Node-RED.



es	ults			
L	Data			
	TagName		DateTime	Value
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:05.62700	2602303
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:08.65700	2602303
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:11.68700	2602305
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:14.71800	2602305
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:17.74800	2602308
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:20.77800	2602308
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:23.80900	2602309
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:26.83900	2602309
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:29.86900	2602311
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:32.90000	2602311
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:35.93000	2602314
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:38.96000	2602314
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:41.99000	2602315
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:45.02100	2602318
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:48.05100	2602318
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:51.08100	2602319
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:54.11200	2602319
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:48:57.14200	2602322
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:49:00.17200	2602322
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:49:03.20300	2602323
	Instalacior	nFotovoltaica.ETotalCT1	2023-04-29 13:49:06.23300	2602323
	Instalacior	nFotovoltaica.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, **e** 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 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 FOTOVOLTA CA 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.





Modbus RTU

Modbus TCP



顶 Moo General

> Product Port nu Connec Restart Restart Restart





# **Adquisición**

# Drivers de Comunicaciones

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.

busEnet				🕼 моха
Alarms Attributes Scripts (	Object Information			General Alarms Attributes Scripts Object Information
				Bridge type: Modbus Bridge 🗸 🚽
version:	3.0.100			Network address: 172.16.3.21
mber:	502			Close Ethernet connection when no activity:
tion heartbeat period:	10000	= ms	6 D	Maximum outstanding messages: 2 😇 🗅
attempts:	3	-	- C (1)	Connection heartbeat period: 10000 📻 ms 🖆 💚
		_		Restart attempts: 3 💽 🗳 🕅
period:	30000	ms	- G. M	Restart period: 30000 🚍 ms 🖆 🐺
reset security:			Ø	Restart reset security:

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

InvertorHT_0	01									
General Alarms Scan	Group Block R	ead Block W	rite	Attributes	Scripts	Object	Information			
Unit ID:	1	×	£	Reply ti	meout:		10	÷	s	8
Use Concept data si	tructures (Longs	s):	8	Use Cor	ncept da	ta structi	ures (Reals):			8
Support multiple co	il write:	$\checkmark$	2	Support	t multiple	e registe	r write:	$\checkmark$		8
Swap string bytes:		$\checkmark$	8	Use Zer	o Based /	Addressi	ing:	$\checkmark$		8
Bit order format:	B1 B2 B1	6 V	8	Registe	r size (di	gits):	6	$\sim$		8
String variable style	: Full length	$\sim$	8	Registe	r type:		Binary	$\sim$		8
Register order:	R1R2R3R4	4 V	8							
_										
InvertorHT_00	1									
General Alarms Scan G	Froup Block Rea	d Block Write	At	tributes Sci	ripts Ob	ject Infor	mation			
Available scan gro	ups:									
ScanGroup					ι	Update In	terval			
<default></default>					:	250				
Generation_kWh					1	5000				

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

F	legister ord	er: R	1 R2 R3 R4	$\sim$	64						
	Invertori	IT_001									
Gene	ral Alarms	Scan Group	Block Read	Block Write	Attributes	Scripts	Object Information				
	Available	can groups:									
	Available	scangroups.					1				-
	ScanGroup						Update Interval			Scan Mode	
	<default></default>						250			ActiveOnDemand	
	Generation	kWh					5000			ActiveOnDemand	-
							1 1				
	Associate	d attributes fo	r Generation	_kWh:		1	londerware				
	Attribute						by AVEVA	Item Re	ference		
	Cumulative							432106	I		









# 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	Туре	Unit	Gain	Numb
32106	Cumulative Generation	RO	U32	kWh	100	2
1						

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	INT1611	D	Total Feed Energy
			1111100	ň	to grid
0313	E-Total L	0.1KW.Hr	INT1611	D	Total Feed Energy
			1111100	n n	to grid
Cumulative					400786 I

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



Tagname.





# Historización

# HISTORIAN-SQL

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



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  $\rightarrow$  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					
	Towarter, JWb, 001 Epergy, JWb	2023-11-15 15:42:12 77700	26082363					





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 entrotal de los 5. Valores de energía generada (kWh) entrota en el último día y tabla de totales por día.



Dashboard









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



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**.





23 01 02 01 05 01 02 01 11 01 14 01 17 01





259582.3

259950.2

260126.9

2023-04-29 10:05:52.8630000 2023-04-29 11:05:52 8630000

2023-04-29 12:05:52.8630000

2023-04-29 13:05:52.8630000

2023-04-29 14:05:52.8630000 260259.7 132,8 2023-04-29 15:05:52.8630000 260421.6 161.9

150.5

221.2

146.7

176.7

Node-RED Node-RED **HISTORIAN-SQL** Un detalle clave de una query para dar el 💻 💻 Query y las claves para entenderla: rendimiento (producción) de un día por horas y 🗖 🗖 select t.fechaHora as x, round(t.kWh\*0.1, 2) as kWh, round(t.SaltokWh\*0.1, graficarlo. 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 Runtime.dbo.AnalogHistory FROM where Tagname **Rendimiento** hoy '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 kWh 2023-04-28 16:05:52 8630000 258745.5 318 293 023-04-28 18:05:52 8630000 182 5749.4 Se obtienen datos de cada hora con wwresolution = 3600000 2023-04-28 23:05:52 8630000 259358.9 2023-04-29 00:05:52 8630000 Energía FV CTs (kWh) Se calcula el incremento de la 2023-04-29 02:05:52 8630000 259358.9 699 2023-04-29 03:05:52 8630000 259358.9 2023-04-29 04:05:52 8630000 524.25 hora actual respecto la anterior 2023-04-29 05:05:52 8630000 259358.9 349.5 174.75 con LAG(). 2023-04-29 09:05:52 8630000 259431.8 59









HISTORIAN-SQL

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

X,	set msg.payload	-	ETotalCT1 bar	

I≣ R	ules				
	Set	~	🕶 msg	J. payload	
Ξ		to the value	▪ J:	[ { "series": ["ETotalCT1"], "data": [[msg.payload.SaltokWh]], "labels": [msg.payload.fechaHora] } ]	
				{     [         [         [         [	
		I Group		[Fotovoltaica] Fotovoltaica (por CT)   "data": [[msg.payload.SaltokWh]],  "labels": [msg.payload.fechaHora]	
		ច្រាំ Size		auto   }	
		£ Label		Energía FV CT1 (kWh)	
		🛃 Туре		Lill Bar chart	
		Y-axis		min 0 max 399	

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 WALBOX T



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 1.6

# (Open Charge Point Protocol)

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  $\rightarrow$  soporta SOAP y JSON (formato mensajes)
- 2.01  $\rightarrow$  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

CS request

**CP Reque** 

CS server

CP serve



- 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

OCPP 1.5 SOA	OCPP 1.5 SOAP						
✓ 1.5 SOAP	enabled						
Path	/ocpp15s						
OCPP 1.6 SOAP							
1.6 SOAP enabled							
Path	/ocpp16s						
OCPP 1.6 JSON							
1.6 JSON enabled							
Path	/осрр						

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

Secuencia básica

Charge Point	Central System
Authorize.req(idTag)	<b></b>
Start Charging	
StartTransaction.conf(idTagInfo, transactionId)	<b></b>
Charging	
Authorize.conf(idTagInfo)	
Stop Charging	
Stop Transaction.req(meterStop, timestamp, transactionId, [reason], [id Tag], [transactionData])	









EVSE (CP) Client



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

	Name	EVSE-001
CP client	cbld	evse-001
JSON	Central System	Localhost:8834/ocpp
_	OCPP Ver	1.6 JSON 🗸
<u> </u>	Delay co	onnection on startup
Web socket ws://	Command	<none></none>

### **Central System (Server)**



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



Name	CentralSystem@8834						
Port	8834						
OCPP 1.6 JSON							
1.6 JSON enabled							
Path	/осрр						





24



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 $\rightarrow$ CS JSON MeterValues msg.payload = { "command": "MeterValues", 2 З "data": { 4 "connectorId": 1, 5 "meterValue": [ 6 7 "timestamp": new Date().toISOString(), "sampledValue": [ 8 9 10 "value": getRndInteger(0, 200), "unit": "kWh", 11 12 "measurand": "Energy.Active.Import.Register" 13 14 15 16 17 18 19 return msg; 20 21 22 function getRndInteger(min, max) {

return Math.floor(Math.random() \* (max - min + 1)) + min;









# Datos recibidos en CS

### 

### ?Qué llega?

Name OCPP CS Command Switch				
••• Pr	operty			
≡	== ¥	• <sup>a</sup> <sub>z</sub> BootNotification	→ 1 ×	
=	== •	▼ <sup>a</sup> <sub>z</sub> Authorize	→ 2 💌	
=	== •	▼ <sup>a</sup> <sub>z</sub> Heartbeat	→ 3 ×	
≡	== •	▼ <sup>a</sup> <sub>z</sub> StatusNotification	→ 4 💌	
≡	== ¥	✓ <sup>a</sup> <sub>z</sub> MeterValues	→ 5 ×	
≡	== ¥	▼ <sup>a</sup> <sub>z</sub> StartTransaction	→ 6 ×	
≡	== ¥	▼ <sup>a</sup> <sub>z</sub> StopTransaction	→ 7 ×	
≡	otherw	ise y	e ×	
		• OCPP CS Command Switch	000	

#### msg∶Object ▼object

▼ocpp: object

chargeBoxIdentity: "evse-001"
MessageId: "f9889a70-c143-497d-9cb2-0aa557a5b3a6"

msgType: 2

command: "MeterValues"

▼payload: *object* 

command: "MeterValues"

▼data: object

connectorId: 1

meterValue: array[1]

\*0: object
timestamp: "2023-04-06T06:50:44.786Z"

sampledValue: array[1]

▼0: object

value: 130

unit: "kWh"

measurand: "Energy.Active.Import.Register"
msgId: "c7ba6fb4-0242-4f65-bb10-9866ac97bbec"

msgid: "5606bdb2dd7c9d53"

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

JSON CP $\rightarrow$ CS

- ocpp → define la identidad del chargeBox (cbld) y el command.
- payload → contiene el campo data. Este contiene el connectorld 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)

msg.payload.data.meterValue[0].sampledValue[0] : Object
▼object
value: 153
unit: "kWh"
measurand: "Energy.Active.Import.Register









# 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://gilhub.com/victomunoz/OCPP-1.6-Chargebox-Simulator/

### Es un HTML con los comandos del protocolo OCPP.

	Simple OCPP 1.6 Chargebox Simulator
	Central Station ws://192.168.1.133.8834/ocpp/evse-001 Tag 0480267AE05C87
🔃 kubarskii	Actions
	Disconnect
🍓 victormunoz	Authorize
•	Start Transaction
🖲 dinko nobar	Stop Transaction
alliko-pellal	Heartbeat
	Meter value
📑 Ahmed-NasrEldin	Send Meter Values
	Status Notification
	Data Tranfer
	<ul> <li>ws connected</li> <li>Response: {"interval":120, "currentTime": "2023-04-10T14:25:31.911Z", "status": "Accepted"}</li> </ul>

# **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 servidor

chargeBoxIdentity: "evse-001"

"JNlwGd7L3V4nJswYlqO5gj4H1a5Sw2Hd

msg : Object

▼object

▼ocpp: object

kvsS"

MessageId:

msgType: 2 command: "Heartbeat"

> payload: object

▼data: object

empty

0a8a689d12a0"

command: "Heartbeat"

msgId: "ef7f96a5-f4a3-4ac9-8df4-

msgid: "b9935da3e0981695"













### StartTransaction command

MeterValues command

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

### Lado servidor



Lado cliente

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

#### Lado servidor



ec5983e847da"

msgid: "ae4efa87eec41445"

#### Lado cliente











response

Value

Accepted

Rejected

UnknownMessageId

UnknownVendorId

## **StopTransaction command**

# 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 (Vendorld, Messageld, etc.)

Lado servidor

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 cliente

### Lado servidor

			msg : Object	mog : objekt
msg : Object	<ul> <li>Connector status changed to: false</li> </ul>		▼ object	▼object
▼ object	<ul> <li>Response: {"idTagInfo"; {"status": "Accepted"}}</li> </ul>		▼ocpp: object	▼ocpp: object
▼ocpp: object			chargeBoxIdentity: "evse-001"	chargeBoxIdentity: "evse-001"
chargeBoxIdentity: "evse-001"			MessageId:	MessageId:
MessageId: "kgOwBy801neVLjxgeYRkYsfA8NAs6PYz			"kgOwBy801neVLjxgeYRkYsfA8NAs6PYz3J 4t"	"kgOwBy801neVLjxgeYRkYsfA8NAs6PYz 3J4t"
3J4t"			msgType: 2	msgType: 2
msgType: 2			command: "StatusNotification"	command: "DataTransfer"
command: "StopTransaction"			<pre>▼payload: object</pre>	<pre>▼payload: object</pre>
▼payload: object			command: "StatusNotification"	command: "DataTransfer"
command: "StopTransaction"			▼data: object	▼data: object
▼data: object			connectorId: 2	vendorId: "rus.avt.cp"
transactionId: "[object			status: "Available"	messageId:
Object]			errorCode: "NoError"	"GetChargeInstruction"
1d1ag: "0480267AE05C87"			info: ""	data: ""
10T16:38:45Z"			timestamp: "2023-04-10T16:42:58Z"	msgId: "901b780a-866b-4007-8383-
meterStop: 20			vendorId: ""	eb435e5977c5"
msgId: "52f77a4b-98d1-49b7-96a1- 97aeba933f6b"			vendorErrorCode: ""	_msgid: "1dc42ba6208e792a"
_msgid: "ae4efa87eec41445"		States considered Operative are: Available, Preparing, Charging, SuspendedEVSE, SuspendedEV,		
			r misning, Reserved. States considered moperative are. Ondvallable	e, ruuteu.



# **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/

<u>Píldoras Tecnológicas © 2024 by Ricardo Moraleda Gareta is licensed under CC BY-NC-ND 4.0</u>

