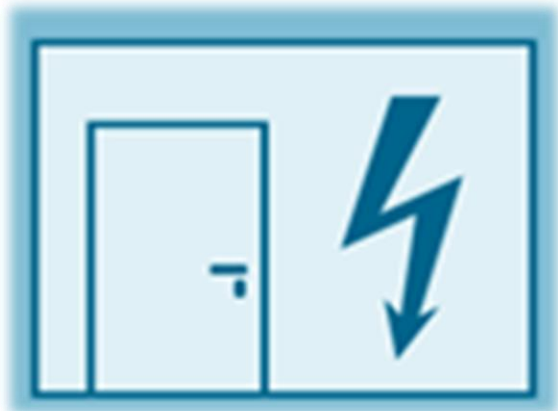




UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño



SISTEMA SCADA PARA EL CONTROL DE TRES CENTROS DE TRANSFORMACIÓN

Grado en Ingeniería Electrónica
Industrial y Automática

Autor: Álvaro Ferrández Salinas
Tutor: Ranko Zotovic Stanisic

Trabajo de fin de grado

AGRADECIMIENTOS

Agradecer a toda mi familia que siempre ha estado apoyándome en mi trayectoria estudiantil y que lo han dado todo por mí. También a mis amigos y a mi novia que me han hecho ser exactamente quien soy ahora mismo.

Me gustaría dar las gracias también a mi tutor Ranko Zotovic, así como al resto de profesores que se han preocupado por mi educación.

Por último, agradecer a toda la comunidad científica que, compartiendo cada descubrimiento con el resto de compañeros, consigue que la sociedad continúe en constante desarrollo.

RESUMEN

El presente trabajo de fin de grado tiene como finalidad la aplicación de un sistema SCADA (*Supervisory Control And Data Adquisition*) para facilitar la monitorización y el mantenimiento de tres centros de transformación desde un puesto de control. Por medio de dispositivos pasivos se obtienen todos los datos posibles sobre los centros de transformación para, con PLCs (Programmable Logic Controller) y dispositivos de periferias descentralizadas, enviar los datos al puesto de control. También se va a hacer uso del estándar de comunicación Modbus para comunicar los valores que almacenan varios analizadores de consumo en las cabinas de los centros de transformación. Finalmente, para facilitar la lectura de los datos, en el puesto de control hay un ordenador que hace de HMI (*Human Machine Interface*) y una pantalla táctil en cada centro de transformación, también como HMI. Gracias a esta aplicación, la cual es escalable al número de centros de transformación deseados, un equipo de mantenimiento eléctrico es capaz de monitorizar estos centros de manera sencilla y unificada en un único dispositivo, para actuar de manera más rápida y eficaz.

Palabras clave:

centro, transformación, automatización, SCADA, PLC.

ABSTRACT

The purpose of the following degree final project is to apply a SCADA (Supervisory Control And Data Acquisition) system in order to help with the monitoring and maintenance of three transformation centres from a single control workstation. By means of passive components all the data possible about the transformation centres is collected and sent to the control workstation using some PLCs (Programmable Logic Controller) and remote controllers. The Modbus standard of communication is also going to be used to read the values stored in some consumption analyzers placed at the transformation centres. Finally, for making easier the reading of the data, there is a computer at the control workstation and a touchscreen per transformation centre as a HMI (Human Machine Interface). Thanks to this application, which is adjustable to the number of transformation centres desired, an electrical maintenance work team is able to read these centres easily and unified in a single device to react faster and efficiently.

Keywords:

Transformation, centre, automatization, SCADA, PLC.

ÍNDICE

1. Introducción	1
2. Estado del arte	2
2.1. Primeros PLC	2
2.2. Actualidad	2
2.3. Justificación	2
3. Sistema SCADA	3
3.1. PLC	3
3.2. Módulos de entradas adicionales	4
3.3. Periferias descentralizadas	5
3.4. Analizadores de consumo	6
3.5. Cable Ethernet	7
3.6. Switch	7
3.7. Software y ordenador	9
3.8. HMI	10
4. Conexión y comunicación entre componentes	11
4.1. Ethernet/IP	12
4.2. Modbus	13
4.2.1. Modbus TCP/IP	13
5. Programación en TIA Portal	13
5.1. Creación de un nuevo proyecto	14
5.2. Identificación del hardware	15
5.3. Asignación de direcciones IP	21
5.4. Tablas de variables	23
5.5. Bloques de programa	24
5.6. Diseño del HMI	36
5.6.1. Diseño de la plantilla	37
5.6.2. Diseño del WinCC Runtime Advanced	40
5.6.3. Diseño de las pantallas KTP700	45
6. Conclusiones	45
7. Bibliografía	45
8. Anexos	46
8.1. Código Sequencer	46
8.2. Código SetCom	78
8.3. Registros Modbus	81

ÍNDICE DE ILUSTRACIONES

FIGURA 1. VISTA GENERAL DE LA TOPOLOGÍA	1
FIGURA 2. PLC	3
FIGURA 3. MÓDULO DE ENTRADAS	4
FIGURA 4. BASE DE ENTRADAS BLANCA	5
FIGURA 5. BASE DE ENTRADAS GRIS	5
FIGURA 6. PERIFERIA DESCENTRALIZADA	6
FIGURA 7. SENTRON PAC4200	6
FIGURA 8. VISTA EN DETALLE DE LA TOPOLOGÍA.....	8
FIGURA 9. SWITCH	8
FIGURA 10. CABLE ETHERNET RJ45.....	7
FIGURA 11. SERVICIOS TIA PORTAL	9
FIGURA 12. ORDENADOR PUESTO DE CONTROL.....	10
FIGURA 13. SIMATIC HMI KTP700	11
FIGURA 14. PUERTOS DE EXPANSIÓN DE MÓDULOS.....	12
FIGURA 15. MONTAJE SOBRE RAIL.....	12
FIGURA 16. NUEVO PROYECTO TIA PORTAL	14
FIGURA 17. PRIMEROS PASOS TIA PORTAL.....	14
FIGURA 18. INICIO DEL PROYECTO EN TIA PORTAL.....	15
FIGURA 19. IDENTIFICACIÓN DEL PLC.....	15
FIGURA 20. DETECCIÓN DE LA CPU DEL PLC.....	16
FIGURA 21. CATÁLOGO DE HARDWARE	17
FIGURA 22. IDENTIFICACIÓN PERIFERIA DESCENTRALIZADA.....	17
FIGURA 23. CONFIGURACIÓN MÓDULOS DE ENTRADAS EN EL PLC.....	17
FIGURA 24. CONFIGURACIÓN MÓDULOS DE ENTRADAS EN LA PERIFERIA	18
FIGURA 25. REDIRECCIONAMIENTO DE LAS ENTRADAS	18
FIGURA 26. ASISTENTE DE CONFIGURACIÓN DE HMI.....	19
FIGURA 27. CREACIÓN VENTANA DE DIAGNÓSTICO.....	19
FIGURA 28. ADICIÓN DEL MÓDULO DE INTERNET	20
FIGURA 29. IDENTIFICACIÓN WINCC RT ADVANCED.....	20
FIGURA 30. COMPONENTES AÑADIDOS	21
FIGURA 31. CONEXIÓN ENTRE COMPONENTES EN TIA PORTAL.....	21
FIGURA 32. ASIGNACIÓN DIRECCIONES IP Y MÁSCARAS SUBRED	22
FIGURA 33. ESTADO DE LA CONEXIÓN EN USO.....	23
FIGURA 34. DETALLES DE LA CONEXIÓN EN USO.....	23
FIGURA 35. PROPIEDADES PROTOCOLO TCP/IPv4	23
FIGURA 36. TABLA DE VARIABLES "CT_1"	23
FIGURA 37. VARIABLES DE ENTRADA BLOQUE DATA_SETCOM	26
FIGURA 38. VARIABLES DE ENTRADA/SALIDA BLOQUE DATA_SETCOM.....	26
FIGURA 39. VARIABLES ESTÁTICAS BLOQUE DATA_SETCOM	26
FIGURA 40. BLOQUE TAGS.....	27
FIGURA 41. ABRIR LA INTERFAZ DE BLOQUE.....	28
FIGURA 42. ENTRADAS INTERFAZ DE BLOQUE SETCOM	28
FIGURA 43. ENTRADAS/SALIDAS INTERFAZ DE BLOQUE SETCOM.....	28
FIGURA 44. CONSTANTE INTERFAZ DE BLOQUE SETCOM	28
FIGURA 45. REGIÓN 1 DEL CÓDIGO SETCOM	29
FIGURA 46. REGIÓN 2 DEL CÓDIGO SETCOM	29
FIGURA 47. REGIÓN 3 DEL CÓDIGO SETCOM	29
FIGURA 48. REGIÓN 4 DEL CÓDIGO SETCOM	29
FIGURA 49. REGIÓN 5 DEL CÓDIGO SETCOM	29
FIGURA 50. REGIÓN 6 DEL CÓDIGO SETCOM	30

Sistema SCADA para el control de tres centros de transformación

FIGURA 51. REGISTROS MODBUS USADOS	31
FIGURA 52. ESTRUCTURA COMÚN CÓDIGO SEQUENCER	32
FIGURA 53. ESTADO 0 DEL CÓDIGO SEQUENCER.....	32
FIGURA 54. ESTADO 10 DEL CÓDIGO SEQUENCER.....	33
FIGURA 55. ESTADO 11 DEL CÓDIGO SEQUENCER.....	33
FIGURA 56. ESTADO 70 DEL CÓDIGO SEQUENCER.....	34
FIGURA 57. ESTADO 80 DEL CÓDIGO SEQUENCER.....	34
FIGURA 58. ESTADO PROCESS_COMM_ERROR DEL CÓDIGO SEQUENCER	35
FIGURA 59. BLOQUE MAIN	36
FIGURA 60. VENTANA DE DIAGNÓSTICO DEL ASISTENTE.....	37
FIGURA 61. DISEÑO DE LA PLANTILLA	37
FIGURA 62. SECCIÓN DE HERRAMIENTAS.....	38
FIGURA 63. CONFIGURACIÓN DE LA VISIBILIDAD DE UN BOTÓN	38
FIGURA 64. CONFIGURACIÓN DE EVENTOS DE BOTÓN	39
FIGURA 65. CONFIGURACIÓN CAMPO DE FECHA/HORA	39
FIGURA 66. CONFIGURACIÓN DE LA ANIMACIÓN DE APARIENCIA DE UN OBJETO	40
FIGURA 67. EVENTO "ACTIVATEPREVIOUSSCREEN"	40
FIGURA 68. CREACIÓN DE LAS PANTALLAS PARA WINCC.....	41
FIGURA 69. PANTALLA PRINCIPAL DE WINCC	41
FIGURA 70. PANTALLA DE DIAGNÓSTICO EN WINCC	42
FIGURA 71. IDENTIFICACIÓN PANTALLAS CT	42
FIGURA 72. PANTALLA CT1 EN WINCC.....	43
FIGURA 73. PANTALLA SENTRON_1.....	44
FIGURA 74. IDENTIFICACIÓN PANTALLAS SENTRON	44

ÍNDICE DE TABLAS

TABLA 1. COMPARACIÓN CATEGORÍAS CABLE ETHERNET	7
TABLA 2. DIRECCIONES IP Y MÁSCARAS SUBRED	22
TABLA 3. ESTRUCTURA "TYPEMBCLIENT"	25
TABLA 4. ESTRUCTURA "TYPEDEVICE"	26
TABLA 5. ESTADOS DEL CÓDIGO SEQUENCER.....	32
TABLA 6. CAMPOS DE VARIABLE DE LAS PANTALLAS SENTRON.....	45

1. Introducción

El siguiente trabajo de fin de grado trata de diseñar un sistema SCADA (*Supervisory Control And Data Acquisition*) capaz de controlar varios centros de transformación de media tensión desde un solo puesto de trabajo, facilitando así la acción del equipo de mantenimiento a cargo de ellos.

El supuesto que se trata es el de una red con una topología de estrella con un *switch* en el centro rodeado de tres centros de transformación (C.T.); uno de ellos dispone de un PLC Siemens S7-1500 y los otros dos tienen módulos de periferia descentralizada; y un puesto de control. Además, se pretende implementar una pantalla táctil en cada CT por si se requiere la supervisión *in situ* del sistema. Se adjunta imagen descriptiva de la topología propuesta:

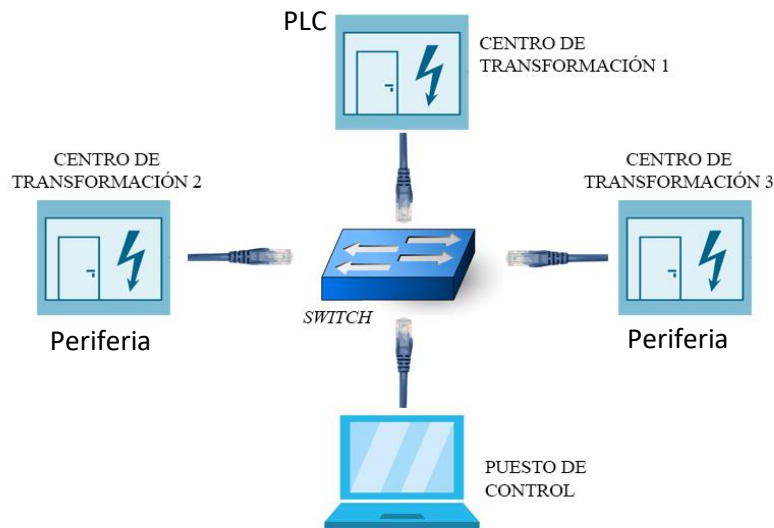


Figura 1. Vista general de la topología

Esta aplicación bien podría darse en un centro comercial, un campus universitario, una zona residencial privada y muchos más casos.

2. Estado del arte

2.1. Primeros PLC

A finales de la década de 1960, la empresa multinacional General Motors comenzó a usar una solución electrónica, propuesta por Bedford Associates, para reemplazar sus sistemas cableados de relés. Esta solución era denominada “084”, al ser el octogésimo cuarto proyecto de Bedford Associates.

Hoy en día se podría decir que ya no existen los “084”, sin embargo, el mundo de la industria tiene muy presentes a sus descendientes, los controladores lógicos programables, más conocidos como PLC, por sus siglas en inglés: *Programmable Logic Controller*. Estos controladores manejan la inmensa mayoría de procesos de las industrias ya que facilitan en gran medida la supervisión y actuación sobre estos mismos.

Pero no son sólo las industrias quienes se nutren de ellos, la potencia, versatilidad y variedad de los PLC los convierte en la opción ideal para controlar desde el clima, las pantallas o las luces de algún edificio público, como un museo, hasta la supervisión y programación de un campo eólico o solar.

2.2. Actualidad

En el sector de la automatización, siempre se busca que cualquier sistema o proceso sea lo más eficiente, seguro y rápido posible. Estos aspectos están cada vez más cubiertos con la gran variedad de herramientas que se ofrecen hoy en día a la hora de supervisar y controlar un proceso.

Marcel Nagel, director de productos de Schunk Gmhb & Co., hablando sobre TIA Portal, sus bloques de funciones, PLC SIM Advanced y MindSphere, se muestra entusiasmado por el futuro de la automatización, cuenta que aún no somos capaces de calcular el tremendo potencial de la digitalización, la cual cubre todas las fases del ciclo de vida de un producto, sin excepción alguna, desde la creación hasta las tareas de servicio y mantenimiento. (Nagel, 2019)

El estudio que se va a realizar trata de acercar los diferentes proyectos realizados anteriormente sobre maquetas, a una aplicación real. El trabajo “Proyecto de automatización de planta multiproceso con horno mediante autómatas Siemens S7-1214C y SCADA en WinCC” del alumno Alfonso Serrano Gallego (Serrano Gallego, 2017), utiliza TIA Portal y un PLC para crear un SCADA que controle una maqueta. Este proyecto podría ser una buena base para nuestro estudio.

2.3. Justificación

En cuanto al aspecto académico, este proyecto es el trabajo de fin de grado, lo que permite al alumno acabar sus estudios académicos y obtener el título del grado en Ingeniería Electrónica Industrial y Automática de la Escuela Técnica Superior de Ingeniería del Diseño en la Universidad Politécnica de Valencia.

Referente al apartado personal y profesional, el proyecto afianza algunos conocimientos obtenidos a lo largo de los estudios anteriormente mencionados, pero necesita de una labor importante de desarrollo más allá de lo aprendido en la escuela, lo cual supone un reto a superar que sirve de aproximación al mundo laboral, así como la satisfacción personal.

3. Sistema SCADA

SCADA es un concepto que engloba los diferentes dispositivos de un sistema que permite la monitorización, control y adquisición de datos de un proceso, generalmente industrial, usando diferentes tipos de dispositivos como un PLC, una pantalla o un sensor. A continuación, se van a detallar los diferentes componentes del sistema SCADA que se pretende desarrollar:

3.1. PLC

Un PLC, o autómatas, es un dispositivo electrónico programable diseñado principalmente para la industria. Normalmente están equipados con entradas y salidas digitales y/o analógicas y una gran capacidad de cálculo y almacenamiento que permiten la automatización de procesos.

El PLC es la parte más importante del SCADA, se encarga de recibir todos los datos del proceso y manejarlos de la manera adecuada para responder como este requiere. Tiene que ser un PLC fiable, preciso y rápido, ya que todo nuestro sistema depende de él, además de tener la capacidad de ser ampliado con diferentes módulos compatibles.

Las empresas más influyentes productoras de PLC, y otros componentes relativos a la automatización, son Siemens, Allen-Bradley y Omron. Todas ellas presentan soluciones aceptables a la hora de escoger el PLC. No obstante, Siemens ofrece productos más compactos y con más soluciones, los cuales disponen de entradas de por sí mismos, sin necesidad de añadir módulos de ampliación.

Para este estudio se ha seleccionado el modelo de Siemens ET 200SP CPU 1512SP-1 PN, número de artículo: 6ES7 512-1DK01-0AB0. Se trata de una periferia descentralizada con la CPU de un PLC S7-1500, esto lo convierte en una opción económica y muy completa que gestionará los datos de manera rápida y eficaz. Sus características que destacar son las siguientes:



Figura 2. PLC

- Memoria de trabajo de 200 KB para programas y 1 Mbyte para datos.

Sistema SCADA para el control de tres centros de transformación

- Interfaz PROFINET IRT con switch de 3 puertos.
- 48 ns rendimiento de bits.
- Tensión de alimentación: 24V DC.
- 1 *slot* para tarjeta SIMATIC Memory Card de hasta 32 Gbyte.
- Remanencia configurable para los datos ante caídas de tensión.
- Capacidad de expansión de hasta 2048 módulos de entradas/salidas.
- Lenguajes de programación: KOP, FUP, AWL, SCL, GRAPH

Es importante también que el PLC sea capaz de comprender protocolos de comunicación como Modbus, para facilitar la programación a la hora de trabajar con varios componentes en la red, como los analizadores de consumo en nuestro caso.

3.2. Módulos de entradas adicionales

Las entradas que nos proporciona el PLC no son suficientes para recoger todos los datos del sistema, por lo que se precisa de módulos de ampliación de entradas, principalmente digitales, ya que los datos analógicos serán adquiridos por los analizadores de consumo.

Con el fin de conseguir la máxima homogeneidad de componentes y evitar posibles incompatibilidades, se van a escoger todos ellos del mismo fabricante que el PLC, a ser posible.

Se necesitan 2 módulos SIMATIC ET 200SP DI 8x 24V DC ST por cada CT, que tienen las siguientes especificaciones:



Figura 3. Módulo de entradas

- SIMATIC ET 200SP DI 8x 24V DC ST (6ES7131-6BH01-0BA0).
- Tensión de alimentación: 24V DC.
- Retardo de la entrada: 0.05-20 ms parametrizable.
- Espacio de 2 bytes por módulo.
- Característica de entrada tipo 3 según IEC 61131.
- Tensión de las entradas: para señal "1", entre 30V y 11V; para señal "0", entre 5V y -30V.

El módulo anteriormente descrito necesita una base con puertos físicos para poder cablear las entradas. Las bases seleccionadas, una para cada módulo, son las SIMATIC ET 200SP BaseUnit

type A0, las cuales disponen de 8 puertos para entradas o salidas y 2 más para la alimentación de la propia base.

Cada CT necesita 2 módulos de entradas, de los cuales, 1 módulo lleva la base de color blanco, código de artículo: 6ES7193-6BP20-0DA0, cuya función es crear un nuevo grupo de potencial y el otro módulo usa la base de color gris, referencia: 6ES7193-6BP20-0BA0, que se une al grupo de potencial al que se conecta. De esta manera se configuran los 2 módulos a un solo grupo de potencial.



Figura 4. Base de entradas blanca



Figura 5. Base de entradas gris

Cómo se ha mencionado, las bases necesitan tensión de alimentación, a 24V DC. Sin otras características que destacar.

3.3. Periferias descentralizadas

A la hora de desarrollar un proyecto también es importante abaratar los costes de este en la medida de lo posible, por esta razón, se pretende que un CT esté equipado con un PLC y los otros dos con periferias descentralizadas.

La función de las periferias descentralizadas es la de transmitir las variables que recogen al PLC que las controla desde una posición remota. Así, dos CT comunicarán constantemente el estado de sus entradas con el CT que lleva el PLC y maneja todos estos datos.

El modelo de periferia descentralizada seleccionado es un Siemens SIMATIC ET 200SP de la familia de los IM 155-6 con referencia 6ES7155-6AR00-0AN, con las siguientes especificaciones:



Figura 6. Periferia descentralizada

- Tensión de alimentación: 24V DC
- Interfaz PROFINET v2.3 IM 155-6 PN Basic
- 2 conectores hembra RJ45 para comunicación Ethernet
- Velocidad de transferencia de 100Mbits/s full dúplex
- Hasta 12 módulos por bastidor
- Hasta 32 bytes de espacio de direcciones por módulo para entradas o salidas

3.4. Analizadores de consumo

Los analizadores de consumo son unos dispositivos cuya función es monitorizar una celda de media tensión para conocer, por ejemplo, la tensión o corriente instantánea de cada fase. Tienen una pantalla de unas 4" donde mostrar los datos en vivo y una pequeña memoria para guardar algunos datos históricos relevantes como picos máximos o mínimos de potencia.

El modelo de analizador seleccionado es el SENTRON PAC4200, de Siemens, con referencia: 7KM4212-0BA00-3AA0. A continuación, se detallan algunas de sus propiedades:



Figura 7. SENTRON PAC4200

- Pantalla LCD con resolución 128x96 píxeles.
- Medición de la potencia activa, reactiva y aparente, de la frecuencia, la tensión, la corriente e impulsos.
- Rango de medición de tensión entre fase y neutro: de 11,5V AC a 480V AC con una tensión nominal máxima de 400V.

- Rango de medición de tensión entre fases: de 20V AC a 828V AC con una tensión nominal máxima de 690V.
- Tensión de alimentación: 24V AC o DC.
- 1 puerto Ethernet para comunicación mediante Modbus TCP, Modbus RTU o PROFINET.
- Tasa de transferencia de entre 10 y 100 Mbits/s.

3.5. Cable Ethernet

Para la conexión entre los diferentes componentes y sus respectivos *switch* se usa cable Ethernet con conectores RJ45 macho a ambos extremos.



Figura 8. Cable Ethernet RJ45

Dependiendo de las necesidades del cable, existen varias categorías de este. En la siguiente tabla, se puede apreciar una comparación entre las tres categorías más usadas:

	Cat5e	Cat6a	Cat7
Velocidad máxima	1 Gb/s	10 Gb/s	100 Gb/s
Ancho de banda máximo	100 MHz	500 MHz	1.000 MHz
Distancia	100 m	100 m	15 m

Tabla 1. Comparación categorías cable Ethernet

El cable de categoría 7 ofrece velocidades de transferencias muy altas, pero tiene un corto alcance, ya que a partir de los 15 metros comienza a disminuir su velocidad, llegando a 40 Gb/s a los 50 metros. La diferencia en cuanto a coste de las categorías 5e y 6a es mínima, por lo que la velocidad y ancho de banda extra que tiene la categoría 6a hace que esta sea la escogida finalmente.

3.6. Switch

Los *switch* nos ayudarán a crear una red Ethernet local, permitiendo la comunicación de todos los dispositivos entre ellos. Anteriormente se ha nombrado que la red tiene una topología de estrella, pero eso no es más que una vista general, ya que, siendo más específicos, y debido a la necesidad de más de un *switch*, se podría decir que es una red mixta, con un anillo central entre los *switch*, aportando redundancia para que si una de estas conexiones cae, la red siga

Sistema SCADA para el control de tres centros de transformación

totalmente comunicada, y una topología de estrella con los diferentes componentes en los extremos y su respectivo *switch* en el centro de ellos.

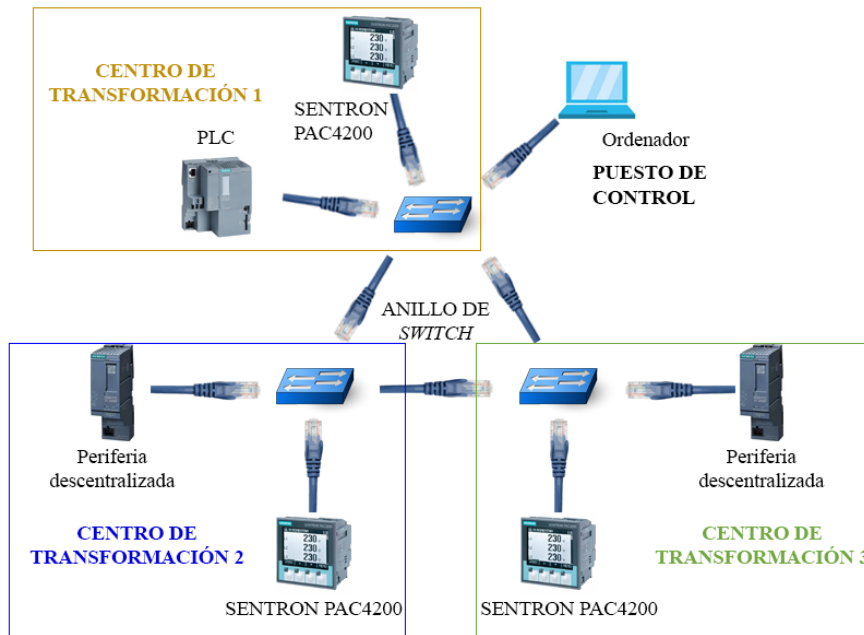


Figura 9. Vista en detalle de la topología

Cómo se puede apreciar en la figura anterior, se necesitarán 3 switch. El modelo escogido es el Siemens SCALANCE XB005 (6GK5005-0BA00-1AB2), un switch con 5 puertos RJ45, una tasa de transferencia entre 10Mbit/s y 100Mbit/s, y alimentado con una tensión de 24V AC o DC.



Figura 10. Switch

Otra opción es sustituir los tres SCALANCE XB005, por un SCALANCE XB116, con las mismas características, pero con 16 puertos RJ45, no obstante, debido a las limitaciones de la tecnología del cableado que se han explicado en el apartado 3.5, no puede haber más de 100 metros de distancia entre los componentes. Usando el *switch* de 16 puertos se complica la conexión de todos los dispositivos ya que se limita el radio de alcance del cableado.

De la misma manera que se ha dividido para este proyecto un *switch* en tres para tener un mayor alcance, cabe la posibilidad de añadir varios *switch* más, si es necesario, para llegar a un dispositivo o CT a más de 100 metros.

3.7. Software y ordenador

Una de las razones por las que se han seleccionado la mayoría de los componentes de Siemens, es por usar el software de la empresa para desarrollar este proyecto. Totally Integrated Automation Portal, más conocido como TIA Portal, es probablemente el software de automatización más completo y sencillo. TIA Portal optimiza cualquier proceso o maquinaria, su intuitiva interfaz de usuario, y la cantidad y facilidad de uso de las funciones hace que sea el programa de automatización ideal.

Se necesita una licencia SIMATIC STEP 7 para poder usar TIA Portal, y una licencia SIMATIC WINCC para poder visualizar y controlar el proceso desde el ordenador a través de Internet o intranet. La siguiente imagen detalla los productos a los que se tiene acceso con cada tipo de licencia:

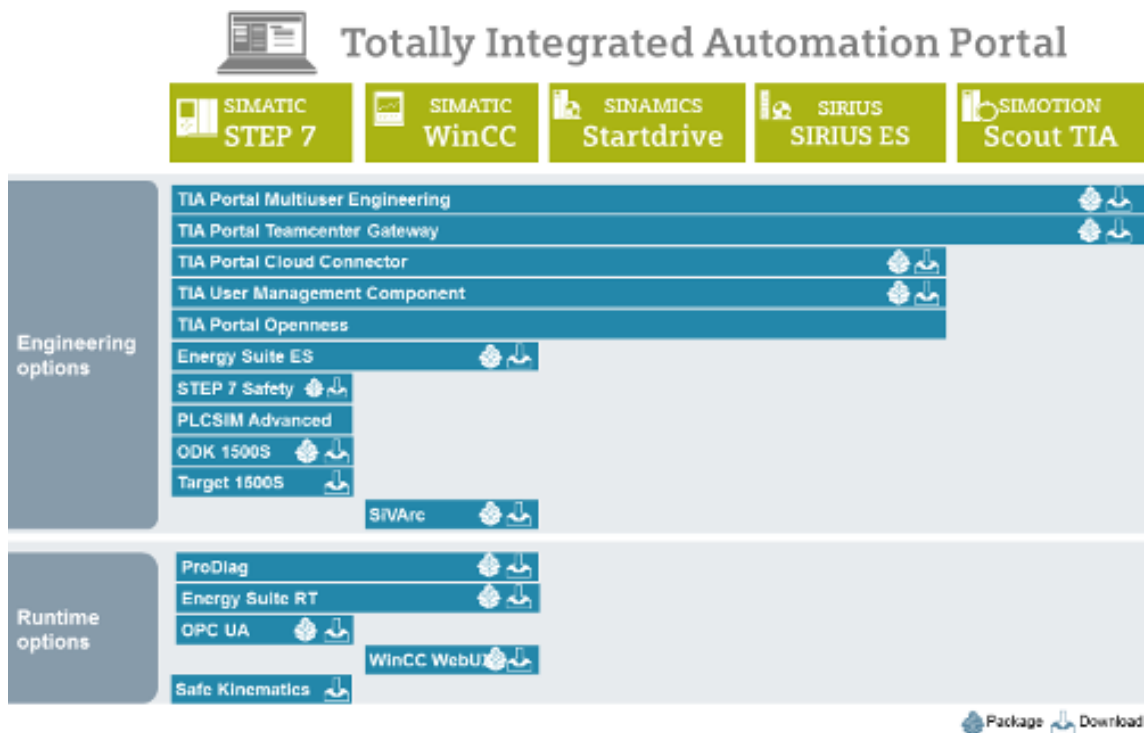


Figura 8. Servicios TIA Portal

Los requisitos recomendados de sistema, según Siemens, para que TIA Portal v15.1 funcione de manera correcta en el ordenador, tanto el SIMATIC STEP 7 como el SIMATIC WINCC, son los siguientes:

- Procesador Core i5-6440M, 3.4 GHz.
- Memoria RAM de 16GB.
- 50GB de espacio disponible en el disco duro. (SSD recomendado)
- 1920x1080 pixeles de resolución de pantalla.

La memoria RAM es la característica más difícil de cumplir, ya que para disponer de 16 GB de memoria RAM se necesita un ordenador con altas prestaciones. La elección del equipo no es relevante siempre y cuando se cumplan los requisitos mencionados anteriormente, a continuación se describe el equipo usado para desarrollar este proyecto.

Sistema SCADA para el control de tres centros de transformación

Se ha escogido el Toshiba Satellite Pro A50-E-1CH, un ordenador portátil que supera los requisitos recomendados por Siemens. Sus especificaciones más importantes son las siguientes:



Figura 9. Ordenador puesto de control

- Procesador Intel® Core™ i7-8550U (frecuencia base de 1,8 GHz, hasta 4 GHz con tecnología Intel® Turbo Boost, 8 MB de caché, 4 núcleos)
- Memoria RAM 16GB DDR4L
- Disco duro 256 GB SSD
- Display Pantalla panorámica (16:9) Toshiba TFT con tecnología LED Mate 15.6 HD 200 LD-Flat eDP 1366 x 768 Pixeles
- Controlador gráfico Intel® HD Graphics 520
- Conectividad
 - Ethernet Gigabit
 - Intel 802.11 ac
 - Bluetooth 4.0
- Conexiones
 - 2 x USB 3.0
 - 2 x USB 2.0
 - 1 x HDMI
 - 1 x RJ-45
- Sistema operativo Windows 10 Home 64 Bits

3.8. HMI

HMI son unas siglas inglesas que quieren decir *Human-Machine Interface*, es decir, interfaz humano-máquina. Este concepto, por lo tanto, es aplicable a los sistemas que nos facilitan, la visualización del estado y los datos de todo, o parte del proceso, es decir, una pantalla o panel de control que permite la interacción entre el usuario y una máquina.

En este estudio, el sistema HMI lo forman 4 componentes: el primero es el ordenador del puesto de mando desde el que se monitorizan todos los CT gracias a la herramienta Win CC Runtime, y los otros 3 componentes son pequeñas pantallas de 7" colocadas en cada CT para poder visualizar el estado del sistema también desde los CT.

Sistema SCADA para el control de tres centros de transformación

WinCC Runtime es un software de Siemens que permite la visualización, en un ordenador, de un HMI programado desde TIA Portal. El paquete de WinCC Runtime Advanced autoriza la visualización, señalización y creación de informes, la administración de usuarios, ampliable de forma flexible mediante scripts de Visual Basic, y es integrable en soluciones de automatización basadas en redes TCP/IP. Además cuenta con diferentes servicios dedicados a facilitar la automatización de procesos, como el envío de correos electrónicos a través de SMTP o la creación de un servidor en la nube desde el que visualizar el estado del PLC y sus salidas y entradas.

Las pantallas, modelo Siemens SIMATIC HMI KTP700 Basic PN (6AV2123-2GB03-0AX0), son unos paneles táctiles TFT de 7" y 8 botones físicos. Tienen un puerto Ethernet que permite la comunicación mediante TCP/IP, PROFINET o Modbus entre otros. Estas pantallas también están programadas desde TIA Portal.

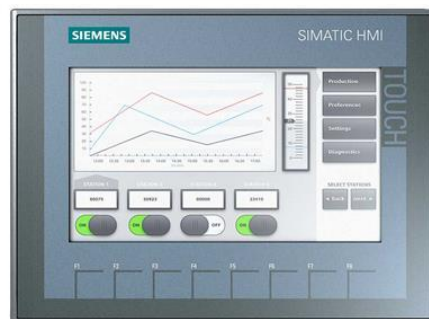


Figura 10. SIMATIC HMI KTP700

4. Conexión y comunicación entre componentes

Los módulos de entradas adicionales y los módulos de servidor se acoplan a las bases, y estas a su vez se acoplan al PLC o a la periferia descentralizada. El hecho de que estos dispositivos sean modulares y de la misma empresa, permite un montaje fácil y con un resultado compacto. En las siguientes figuras podemos ver los puertos que usan para conectarse entre sí.



Sistema SCADA para el control de tres centros de transformación



Figura 11. Puertos de expansión de módulos

La figura 15 ilustra el montaje de los componentes sobre un rail. Cabe destacar que se observa el montaje de una periferia descentralizada, sin embargo, el procedimiento sería el mismo para un PLC.

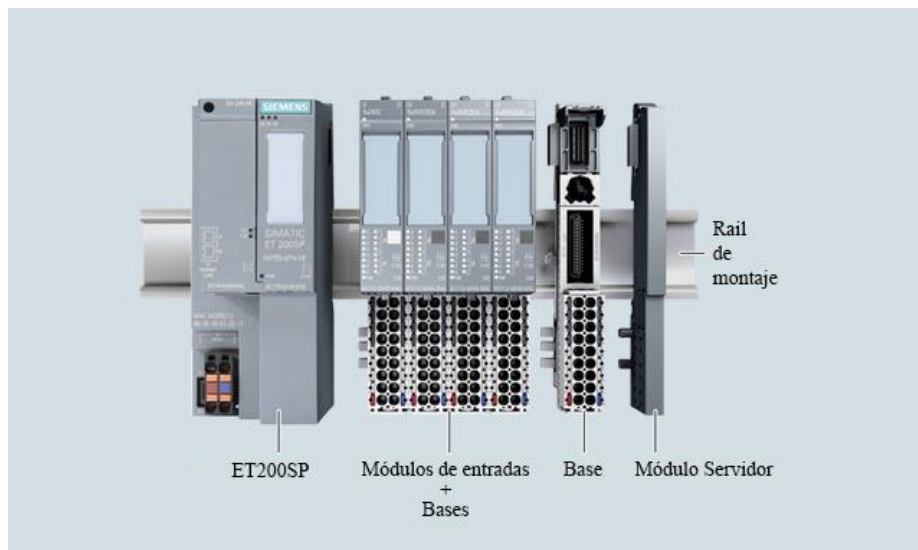


Figura 12. Montaje sobre rail

Cada CT cuenta con un *switch* al cual se conectan los componentes de ese centro que necesitan estar en la red. Concretamente, de los 5 puertos de cada *switch*, dos van a los otros *switch*, uno al PLC o periferia, otro a la pantalla KTP700 y otro al dispositivo SENTRON PAC4200. En el CT 1, la conexión del analizador se realiza directamente al PLC, sin usar el *switch*, de este modo, el puerto libre puede ser utilizado por el ordenador del puesto de control.

Todas estas conexiones deben ser realizadas con el cable Ethernet RJ45 seleccionado en el apartado 3.5.

La comunicación entre los distintos componentes se lleva a cabo gracias a los protocolos Ethernet/IP y Modbus.

4.1. Ethernet/IP

Se trata de un protocolo de comunicación desarrollado principalmente para el área de la industria basándose en el protocolo TCP/IP y usando el hardware y software de Ethernet para configurar, acceder y controlar dispositivos de automatización industrial.

La tecnología Ethernet se usa en gran medida hoy en día en la vida cotidiana de las personas, ordenadores personales, impresoras, videoconsolas, etc. Para estas funciones, Ethernet hace un papel perfecto, no obstante, cuando se está accediendo a datos del propio sistema operativo o de algún controlador de entradas y salidas, Ethernet deja en espera al usuario mientras tanto. En el ámbito de la industria, esos segundos en espera pueden ser decisivos para un correcto funcionamiento del proceso al completo, es por esto por lo que cuatro grandes organizaciones decidieron reunirse para crear una nueva tecnología adaptada a las necesidades de la industria.

La Open DeviceNet Vendor Association (ODVA), la Industrial Open Ethernet Association (IOANA), la Control Net International (CI) y la Industrial Ethernet Association (IEA) consiguieron desarrollar el estándar Ethernet/IP para atender a las exigentes demandas del sector de la industria.

Ethernet/IP clasifica cada componente del sistema como un elemento, lo que reduce los costos de puesta en marcha a la hora de añadir nuevos dispositivos o componentes al perímetro de la red. Esto disminuye el tiempo de respuesta e incrementa la capacidad de transferencia de datos respecto a otros estándares como DeviceNet o al ControlNet.

4.2. Modbus

Modbus es un protocolo de comunicaciones, basado en la arquitectura maestro/esclavo o cliente/servidor. Fue diseñado por Modicon, empresa posteriormente absorbida por Schneider Electric, como protocolo estándar para sus PLCs.

Cuando Schneider absorbió Modicon, éste hizo público su estándar, y al ser fácil de uso y requerir de poco desarrollo, rápidamente se extendió y se convirtió en un protocolo estándar de la comunicación entre dispositivos electrónicos industriales.

En esta comunicación maestro/esclavo, el maestro inicia las comunicaciones (por ejemplo un SCADA) preguntando datos a un esclavo (por ejemplo un PLC), que le responde siempre en función de la pregunta hecha por el maestro. Cada dispositivo de una red posee una dirección única. Modbus es un protocolo de aplicación, lo que significa que puede implementarse sobre diferentes capas físicas. Es por ello por lo que podemos encontrar versiones TCP/IP, como en nuestra aplicación, o también serie, como RTU y ASCII.

4.2.1. Modbus TCP/IP

“La especificación Modbus/TCP define un estándar interoperable en el campo de la automatización industrial, el cual es simple de implementar para cualquier dispositivo que soporte sockets TCP/IP. Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502 y normalmente usando comunicación half-duplex sobre una conexión dada. Es decir, no hay beneficio en enviar solicitudes adicionales sobre una conexión única mientras una respuesta está pendiente.” (Logicbus, s.f.)

5. Programación en TIA Portal

TIA Portal facilita en gran medida la programación de PLCs y otros componentes de los procesos industriales gracias a sus funciones, no obstante, el usuario del programa debe tener claro lo que está haciendo ya que hay muchos parámetros que configurar y puede cometer algún error que provoque un mal funcionamiento.

A continuación, se procede a explicar los conceptos de TIA Portal necesarios para el desarrollo de este trabajo.

5.1. Creación de un nuevo proyecto

Para empezar, se abre el programa y aparece la siguiente pantalla:

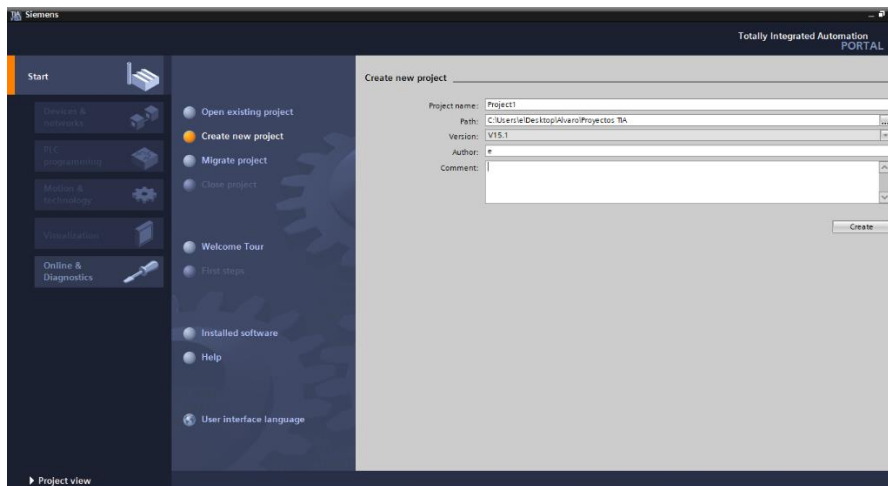


Figura 13. Nuevo proyecto TIA Portal

Seleccionando la pestaña “Create new project” y una vez rellenados los campos de la manera deseada, se pulsa el botón “Create” y se crea el proyecto. Seguidamente, aparece la pestaña “First steps”, como se observa en la figura 17, y se debe hacer clic en “Open the project view” para acceder a la pantalla de la figura 18 y poder comenzar con la identificación del hardware.

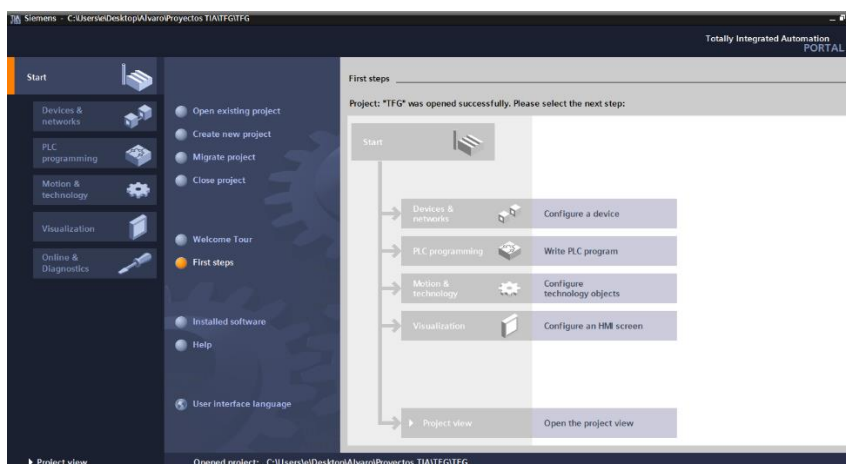


Figura 14. Primeros pasos TIA Portal

Sistema SCADA para el control de tres centros de transformación

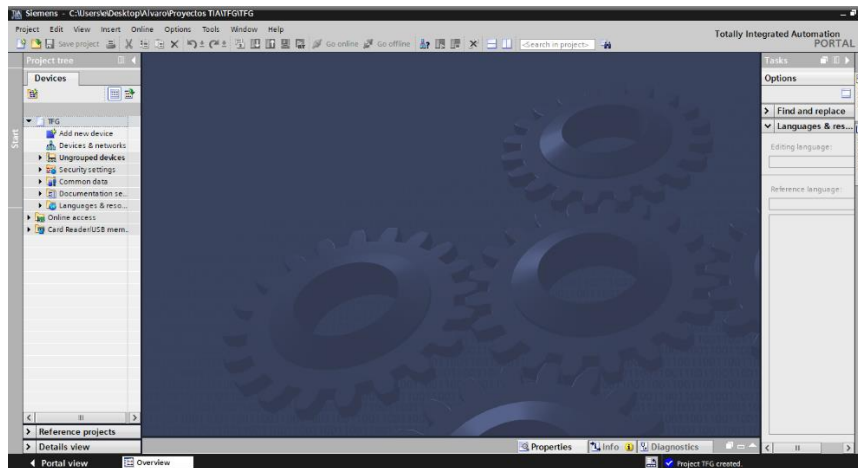


Figura 15. Inicio del proyecto en TIA Portal

5.2. Identificación del hardware

Ahora se procede a identificar en TIA Portal el PLC que estamos usando en este proyecto. Para ello, se conecta el PLC al ordenador con un cable Ethernet y se pulsa la opción “Add new device” del “Project Tree”, o Árbol del proyecto en castellano.

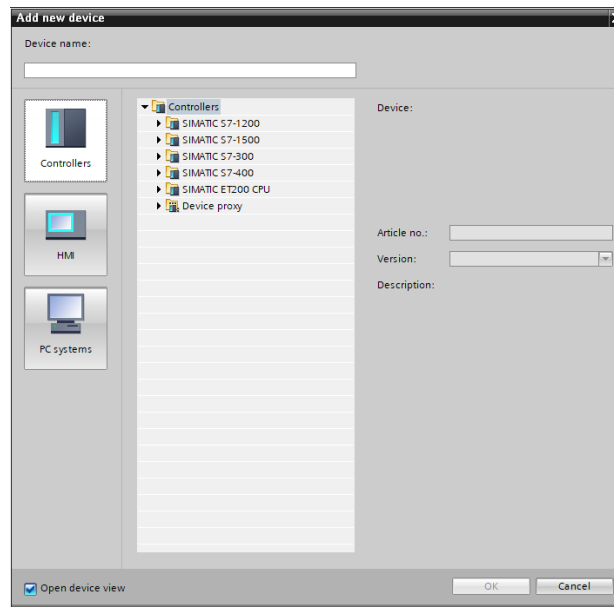


Figura 16. Identificación del PLC

La manera más sencilla y a prueba de errores de añadir nuestro PLC al proyecto es elegir la familia del PLC y hacer que TIA Portal detecte la CPU que este lleva.

En este caso, se selecciona la pestaña “Controllers” que se aprecia en la figura 19, se abre la carpeta “SIMATIC S7-1500” y se escoge la opción “Unspecified CPU 1500”. Tras esto, se abre el área de trabajo con la siguiente imagen:

Sistema SCADA para el control de tres centros de transformación

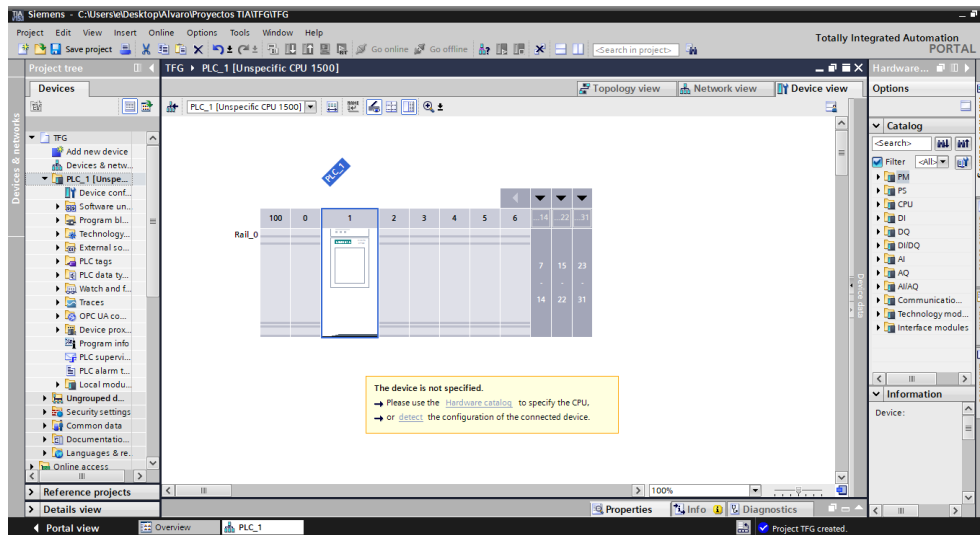


Figura 17. Detección de la CPU del PLC

Con el PLC conectado al ordenador, se debe pulsar sobre “detect” y TIA Portal detectará la CPU que usa el PLC y la añadirá automáticamente al proyecto.

Es aconsejable acceder a las propiedades del PLC y en el apartado “System and clock memory” marcar las casillas “Enable the use of system memory byte” y “Enable the use of clock memory byte”. Al marcar estas casillas, se crean unas variables predeterminadas del PLC las cuales pueden ser muy útiles. La primera casilla genera cuatro variables: un bit que se activa durante el primer ciclo de programa, un bit que indica cambios en el diagnóstico del PLC y 2 bits constantes, “siempre verdadero” y “siempre falso”. La segunda casilla crea variables de reloj las cuales se activan cada cierto tiempo de manera constante: se generan 8 variables de reloj con frecuencias de 0.5, 0.625, 1, 1.25, 2, 2.5, 5 y 10 Hz.

Para añadir los módulos de periferia descentralizada hay que buscar su número de referencia, 6ES7155-6AR00-0AN, en el catálogo de hardware o “Hardware catalog”, figura 21, situado en la parte derecha de la ventana de TIA Portal, y arrastrarlos al área de trabajo de nuestro proyecto en la ventana “Devices & network”.

Sistema SCADA para el control de tres centros de transformación

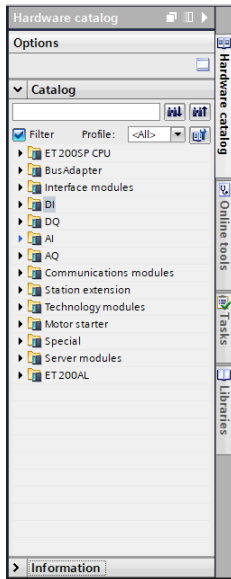


Figura 18. Catálogo de hardware

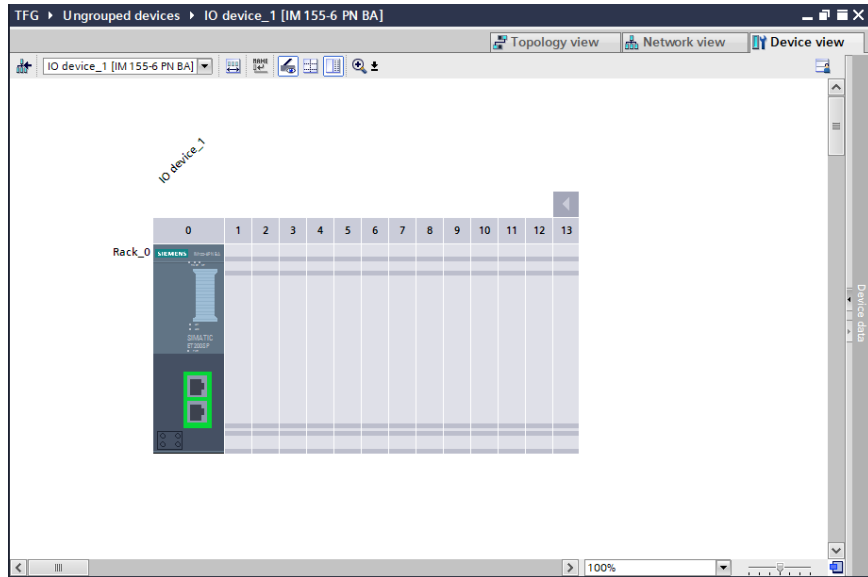


Figura 19. Identificación periferia descentralizada

Los módulos de entradas también se deben buscar en el catálogo de hardware, y añadirlos en el proyecto del mismo modo que los módulos de periferia, pero en su respectivo PLC o módulo de periferia.

Como se explica en el punto 3.2, se deben configurar 2 módulos de entradas para cada PLC o módulo de periferia, siendo una de ellas fuente de potencial. TIA Portal configura automáticamente los módulos de esta manera, aun así, esto siempre puede modificarse desde las propiedades del propio módulo de entradas, en el apartado "Potential group".

Además, se debe añadir el módulo de servidor, el cual contiene la información de los módulos de entradas, su estado y su diagnóstico. Para ello, se ha de buscar su código de artículo, 6ES7 193-6PA00-0AA0, en el catálogo de hardware.

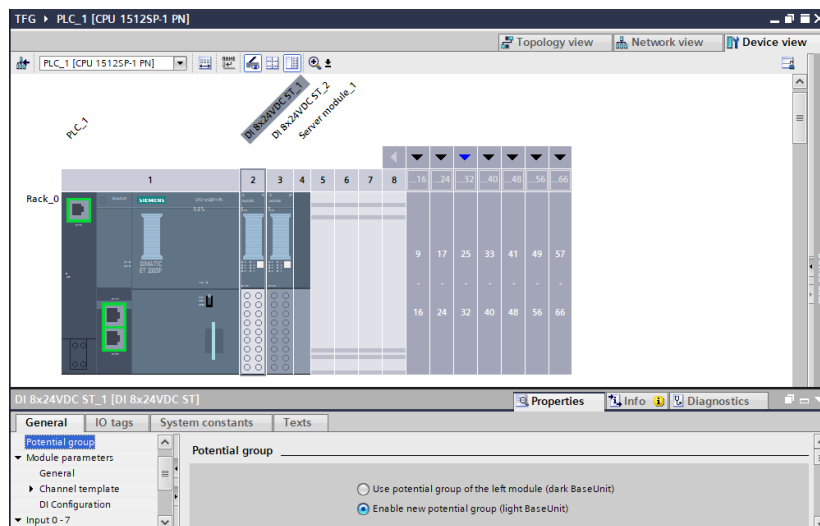


Figura 20. Configuración módulos de entradas en el PLC

Sistema SCADA para el control de tres centros de transformación

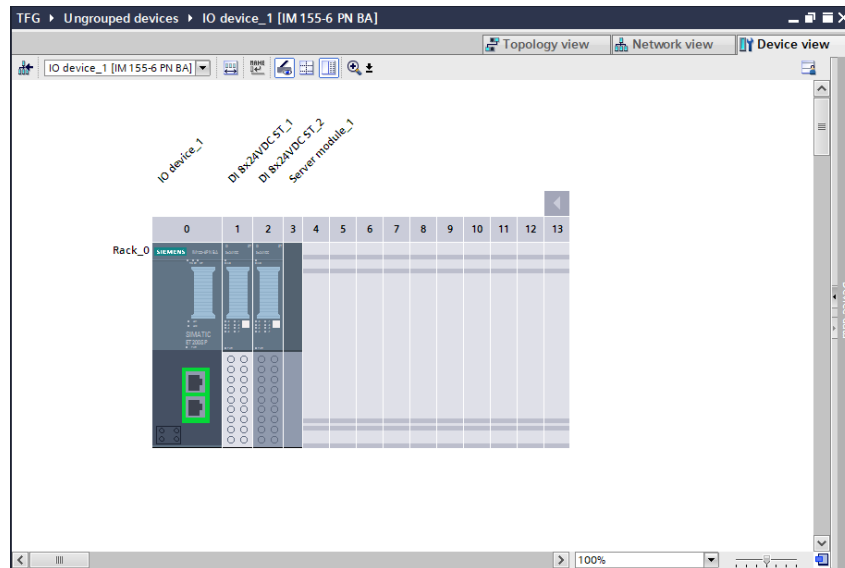


Figura 21. Configuración módulos de entradas en la periferia

Con el fin de mantener orden en el proyecto, se propone direccionar los módulos de entradas que han sido añadidos. Para ello, se debe acceder a las propiedades de los módulos, y haciendo doble clic en “I/O addresses”, dentro del apartado “Input 0-7”, se puede observar lo siguiente:

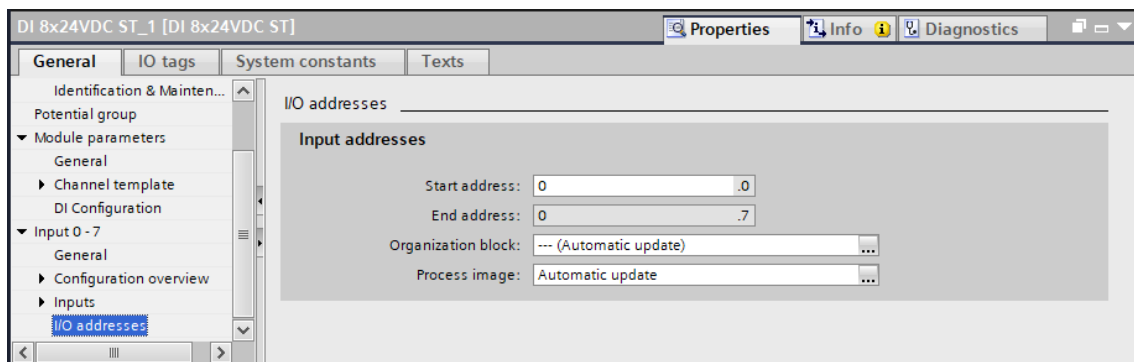


Figura 22. Redireccionamiento de las entradas

Modificando el valor del recuadro “Start address” redireccionaremos las entradas de la manera que prefiramos. En este caso se propone direccionar las entradas del PLC como “0.0” y “1.0”, las del módulo de periferia 1 como “10.0” y “11.0” y las de la última periferia como “20.0” y “21.0”.

De manera muy similar a como se ha añadido el PLC, se debe añadir también la pantalla KTP700, desde la pestaña “HMI” de la figura 19, escogiendo el modelo descrito en el punto 3.9 y procediendo de la misma manera para las tres pantallas.

Al seleccionar la pantalla, se abre un asistente de configuración de esta, con el siguiente aspecto:

Sistema SCADA para el control de tres centros de transformación

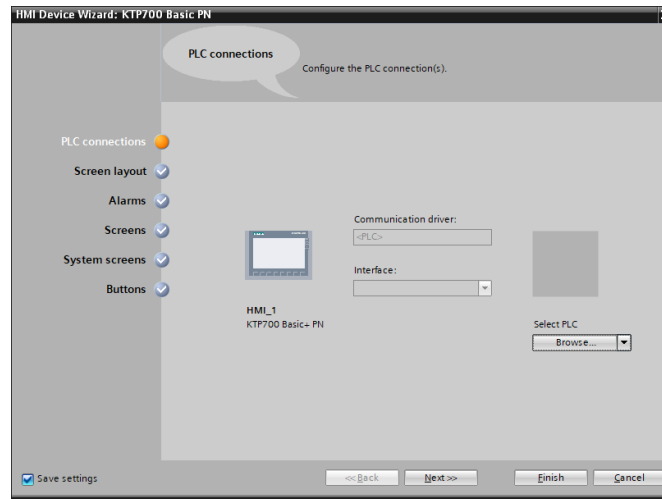


Figura 23. Asistente de configuración de HMI

Para configurar la conexión con el PLC, hay que abrir el desplegable “Browse...” y elegir el PLC que se ha configurado previamente. El resto de las pestañas del asistente ayudan al usuario creando algunas pantallas, botones o funciones básicas ya definidas. El problema de TIA Portal es que el estilo que usa por defecto en las pantallas se ve anticuado y obsoleto, por esta razón no usaremos estas ayudas que nos proporciona el asistente, solamente se va a crear la ventana “SIMATIC PLC” con la función “System diagnostics view”, lo que nos permitirá realizar diagnósticos a tiempo real del PLC y demás componentes del sistema.

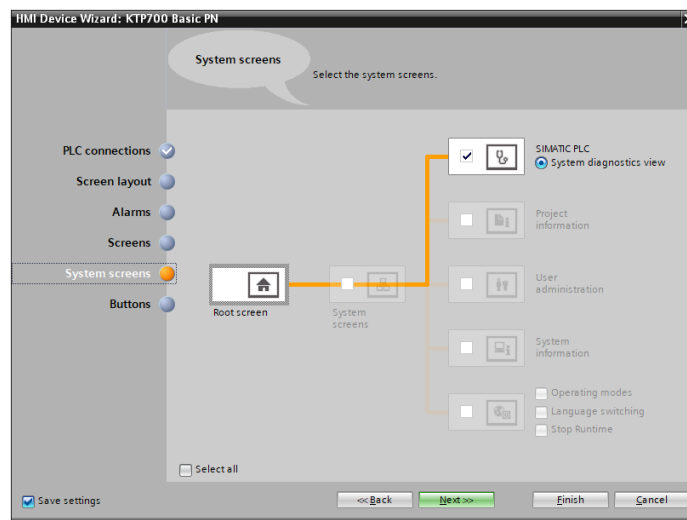


Figura 24. Creación ventana de diagnóstico

Por último, volviendo al menú “Add new device” de la figura 19, se configura WinCC Runtime, seleccionando la pestaña “PC systems” y buscando la herramienta “WinCC RT Advanced” dentro de la carpeta “SIMATIC HMI Application”.

Para terminar de configurar WinCC, hay que añadirle un módulo Ethernet, con el fin de asignarle una dirección IP. Para esto, se arrastra el módulo “IE general” del catálogo de hardware, figura 28, al área de trabajo, debiendo quedar con un aspecto similar a la figura 29.

Sistema SCADA para el control de tres centros de transformación

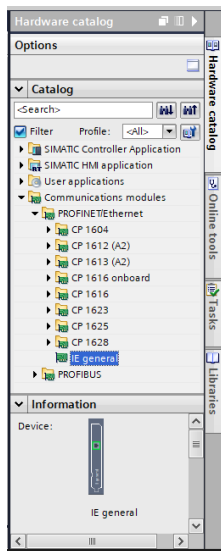


Figura 25. Selección del módulo de Internet

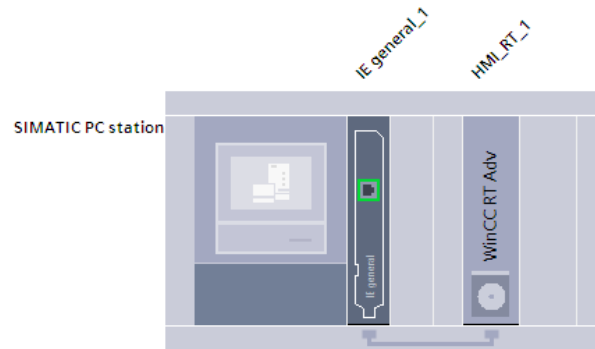


Figura 26. Identificación WinCC RT Advanced

Sistema SCADA para el control de tres centros de transformación

Una vez configurados todos los dispositivos, haciendo clic en la sección “Devices & networks” del árbol de proyecto se muestra una ventana similar a la figura 30.

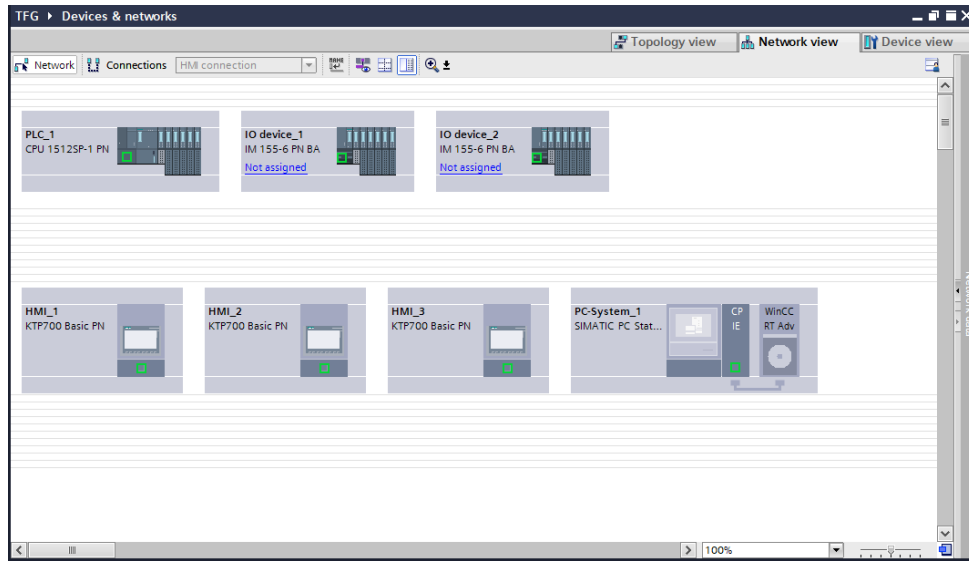


Figura 27. Componentes añadidos

No es necesario añadir los analizadores de consumo SENTRON PAC4200 como componentes a nuestro proyecto, ya que estos son identificados por bloques de funciones del PLC.

5.3. Asignación de direcciones IP

En la figura 30 se puede observar un recuadro verde en cada componente, estos recuadros simbolizan sus puertos RJ45. Haciendo clic y arrastrando desde un recuadro a otro se conectan los dispositivos entre sí creando una red que comunica todos los dispositivos entre ellos.

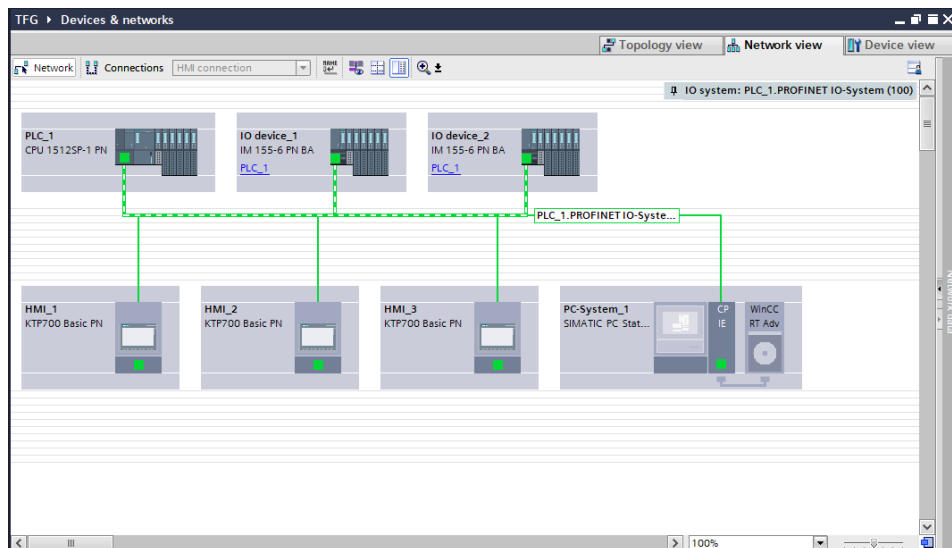


Figura 28. Conexión entre componentes en TIA Portal

Tras haber configurado la red, se debe asignar la dirección IP y la máscara subred deseada a cada puerto. TIA Portal asigna automáticamente una dirección IP, comenzando por 192.168.0.1, de manera sucesiva, hasta la dirección 192.168.0.255, y la máscara subred, asignando siempre el valor 255.255.255.0.

Para modificar estos parámetros en cualquiera de los componentes, se debe acceder a la carpeta del componente en cuestión en el árbol de proyecto, y abrir la ventana “Device configuration”. Una vez aquí, haciendo doble clic en el puerto RJ45 del componente se debe abrir la pestaña de propiedades. En la sección “Ethernet addresses”, se encuentra el apartado “IP protocol” con el siguiente aspecto:

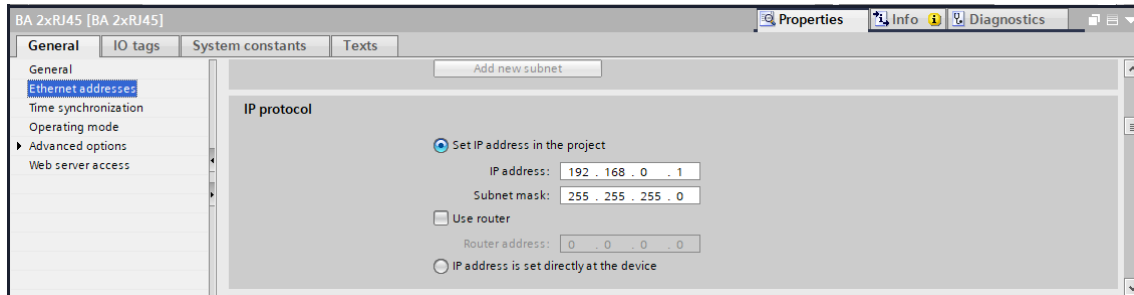


Figura 29. Asignación direcciones IP y máscaras subred

Aquí se pueden configurar la dirección IP y la máscara subred de la manera que se necesite. En este proyecto, los parámetros escogidos son los predeterminados, quedando los componentes con las direcciones y máscaras según la siguiente tabla.

Componente	Dirección IP	Máscara subred
PLC_1	192.168.0.1	255.255.255.0
IO device_1	192.168.0.2	255.255.255.0
IO device_2	192.168.0.3	255.255.255.0
HMI_1	192.168.0.4	255.255.255.0
HMI_2	192.168.0.5	255.255.255.0
HMI_3	192.168.0.6	255.255.255.0
SETRON PAC4200_1	192.168.0.7	255.255.255.0
SETRON PAC4200_2	192.168.0.8	255.255.255.0
SETRON PAC4200_3	192.168.0.9	255.255.255.0
PC-System_1	192.168.0.10	255.255.255.0

Tabla 2. Direcciones IP y máscaras subred

Los analizadores de consumo SETRON PAC4200 utilizan otro estándar de comunicación, Modbus, el cual se configura en el apartado 5.5 Bloques de programa.

Es muy importante para el funcionamiento de WinCC que la dirección IP y la máscara subred del componente PC-System_1 concuerden con las del propio ordenador desde el que se va a visualizar. Para ello, se pueden modificar los parámetros del proyecto, o los del ordenador en sí.

En la configuración del ordenador, accediendo a Panel de control>Redes e Internet>Conexiones de red y haciendo doble clic sobre la conexión en uso del ordenador, se abre una ventana del estado de la conexión, donde se puede ver la dirección IP y máscara subred actuales, pulsando el botón “Detalles”, o cambiarlas, pulsando en “Propiedades”>”Protocolo de Internet versión 4 (TCP/IPv4)” y eligiendo “Usar la siguiente dirección IP”.

Sistema SCADA para el control de tres centros de transformación

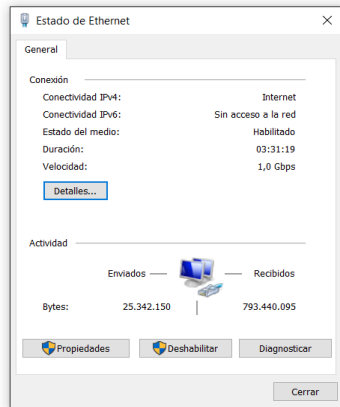


Figura 30. Estado de la conexión en uso

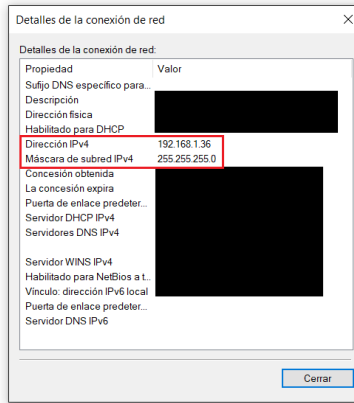


Figura 31. Detalles de la conexión en uso

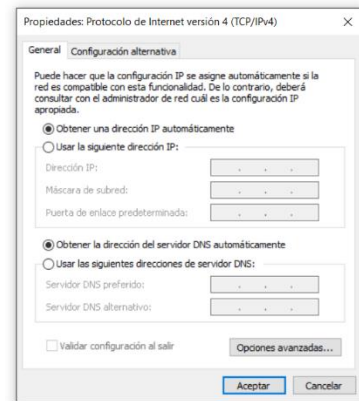
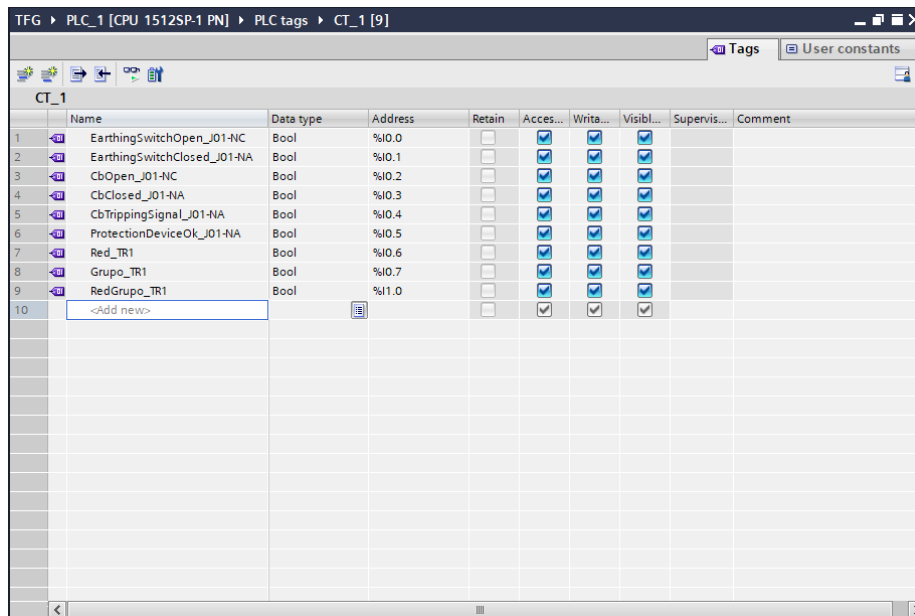


Figura 32. Propiedades protocolo TCP/IPv4

5.4. Tablas de variables

Para organizar las variables que se van a procesar haremos uso de las tablas de variables del PLC. En el árbol de proyecto, desplegando la carpeta del PLC, podemos ver el apartado "PLC tags" al cual podremos acceder para crear diferentes tablas para declarar variables, su nombre, tipo de dato y dirección en el PLC entre otras características.

Con la intención de mantener orden en el proyecto, se va a crear una tabla para cada centro de transformación. En ella se declaran las diferentes señales que se van a monitorizar de su CT. A continuación, se muestra una de las tablas, con las entradas que se propone estudiar.

The screenshot shows a table titled 'CT_1' within a software interface. The table has columns for Name, Data type, Address, Retain, Acces..., Write..., Visib..., Supervis..., and Comment. The first 9 rows contain specific variable declarations, and the 10th row is a placeholder for a new entry. The 'Acces...' and 'Write...' columns have checkboxes, some of which are checked. The 'Visib...' column also has checkboxes, some checked. The 'Supervis...' column has a dropdown menu.

	Name	Data type	Address	Retain	Acces...	Write...	Visib...	Supervis...	Comment
1	EarthingswitchOpen_J01-NC	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	EarthingswitchClosed_J01-NA	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	CbOpen_J01-NC	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	CbClosed_J01-NA	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	CbTrippingSignal_J01-NA	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	ProtectionDeviceOk_J01-NA	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Red_TR1	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Grupo_TR1	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	RedGrupo_TR1	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figura 33. Tabla de variables "CT_1"

Estas señales que se han seleccionado son las más importantes a controlar, no obstante, dependiendo de las necesidades del proyecto y de las cabinas que hayan instaladas en los CT, se pueden recoger más o menos variables.

"Earthing switch" indica si hay alguna derivación a tierra, "CbOpen" y "CbClosed" indican si el relé de protección se encuentra abierto o cerrado, "CbTrippingSignal" registra si sucede un

disparo del relé y “ProtectionDeviceOk” muestra si el relé se encuentra en buen estado. “Red_TR1” indica que el transformador está alimentando la red, mientras que “Grupo_TR1” se activará cuando el grupo electrógeno entre en funcionamiento, es decir, cuando el transformador no genere electricidad. “RedGrupo_TR1” indica que se está suministrando electricidad, ya sea por red o por grupo electrógeno.

5.5. Bloques de programa

La programación en TIA Portal de un PLC se lleva a cabo usando diferentes tipos de bloques de programa, como programas cíclicos (OB), bloques de funciones (FB) o bloques de datos (DB), los cuales pueden hacer uso de numerosos lenguajes de programación: diagramas tipo escalera (LAD), diagramas de bloques de funciones (FBD), listas de instrucciones (STL), lenguaje estructurado de control basado en texto (SCL), diagramas de funciones secuenciales (GRAPH)...

En este proyecto se usan cinco bloques de programa:

- Main
- Tags
- Data_SetCom
- SetCom
- Sequencer

Antes de comenzar a desarrollar los bloques de programa, se deben crear dos tipos de datos que son necesarios para la programación del proyecto.

Siemens ha provisto TIA Portal con numerosos tipos de datos, algunos de los comunes en todos los sistemas de programación, como booleano, entero o real, y otros frecuentemente usados en la automatización, como TCON_IP_v4 o HW_ANY. Sin embargo, este estudio necesita de algunas estructuras más complejas que recojan varios tipos de datos en uno solo.

Para ello, debemos dirigirnos al árbol de proyecto, y en la carpeta “PLC data types” hacer doble clic en “Add new data type”. Crearemos dos tipos de datos nuevos, “typeMbClient” y “typeDevice”.

La estructura “typeMbClient” define los parámetros que establecen y controlan la conexión entre el PLC y los SENTRON PAC4200, mientras que el tipo de dato “typeDevice” recoge las variables que se obtienen de los SENTRON PAC4200. A continuación pueden observarse dos tablas con la configuración requerida de ambos tipos de datos:

Estructura "typeMbClient"			
Nombre	Tipo de dato	Valor por defecto	Descripción
req	Bool	false	Activa la petición de transmisión cuando está en “TRUE”.
disconnect	Bool	false	Inicia la operación de desconexión.
mode	USInt	0	Especifica el tipo de petición: 0 – leer, 1 – escribir, 2 – diagnóstico.
mbDataAddr	UDInt	40066	Especifica la dirección donde comienzan los datos a los que se acceden a través del cliente

Sistema SCADA para el control de tres centros de transformación

			Modbus.
√ connect	TCON_IP_v4		Referencia a los parámetros de conexión.
InterfaceId	HW_ANY	64	Nº de identificación de hardware del submódulo de interfaz IE.
ID	CONN_OUC	1	Nº de identificación de la conexión.
ConnectionType	Byte	16#0B	Tipo de conexión: 11=TCP/IP, 19=UDP (17=TCP/IP)
ActiveEstablished	Bool	true	Establecer la conexión como activa (true) o pasiva (false).
√ RemoteAddress	IP_V4		Dirección IP remota (IPv4).
√ ADDR	Array[1..4] of Byte		Dirección IPv4
ADDR[1]	Byte	192	1º octeto de la dirección IPv4
ADDR[2]	Byte	168	2º octeto de la dirección IPv4
ADDR[3]	Byte	0	3º octeto de la dirección IPv4
ADDR[4]	Byte	16#0	4º octeto de la dirección IPv4
RemotePort	UInt	502	Nº del puerto UDP/TCP remoto (502 = puerto por defecto).
LocalPort	UInt	0	Nº del puerto UDP/TCP local (0 = cualquier puerto).

Tabla 3. Estructura "typeMbClient"

Estructura "typeDevice"			
Nombre	Tipo de dato	Valor por defecto	Descripción
ipOctet4	USInt	0	4º octeto de la dirección IPv4.
pTotal	Real	0	Potencia activa total (kW).
pTotalMax	Real	0	Potencia activa total máxima registrada (kW).
pTotalMin	Real	0	Potencia activa total mínima registrada (kW).
CorrienteL1	Real	0	Corriente de fase L1 (A).
CorrienteL2	Real	0	Corriente de fase L2 (A).
CorrienteL3	Real	0	Corriente de fase L3 (A).
VoltageL1-L2	Real	0	Tensión en la fase L1 sobre L2 (V).
VoltageL2-L3	Real	0	Tensión en la fase L2 sobre L3 (V).
VoltageL3-L1	Real	0	Tensión en la fase L3 sobre L1 (V).
vTHD_L1	Real	0	Distorsión armónica total (THD) en tensión L1 (%).
vTHD_L2	Real	0	THD en tensión L2 (%).
vTHD_L3	Real	0	THD en tensión L3 (%).
cTHD_L1	Real	0	THD en corriente L1 (%).
cTHD_L2	Real	0	THD en corriente L2 (%).

cTHD_L3	Real	0	THD en corriente L3 (%).
cosL1	Real	0	Cos φ en L1.
cosL2	Real	0	Cos φ en L2.
cosL3	Real	0	Cos φ en L3.

Tabla 4. Estructura "typeDevice"

El tipo de dato "typeMbClient" estructura los datos que controlan la comunicación y transmisión de datos entre los SENTRON PAC4200 y el PLC, mientras que la estructura "typeDevice" contiene la dirección IP del SENTRON con el que se esta comunicando, así como el almacenamiento de los datos que recogen.

Una vez instanciados los tipos de datos, se procede al desarrollo de los bloques de programa. Seguidamente se va a explicar la función y los aspectos más importantes de cada uno de los cinco bloques mencionados anteriormente.

➤ Data_SetCom [DB1]

Se trata de un bloque de datos que recoge las diferentes variables que establecen y controlan la comunicación entre los SENTRON PAC4200 y el PLC. En estos bloques, los datos se dividen en cinco tipos: "Input", "Output", "InOut", "Static" y "Temp". Las variables del campo "Input" son datos solo de entrada, las variables del campo "Output" son simplemente de salida y las variables "InOut" tienen las características de las dos anteriores. En cuanto a los tipos "Static" y "Temp" son variables auxiliares usadas dentro del bloque de programación.

A continuación se muestran las variables asignadas en cada campo del bloque de datos Data_SetCom y una pequeña descripción de cada una:

	Name	Data type	Start value	Comment
1	Input			
2	initialCall	Bool	false	Señal de primer ciclo del bloque cíclico

Figura 34. Variables de entrada bloque Data_SetCom

	Name	Data type	Start value	Comment
4	InOut			
5	device	Array[*] of *typeDevice*		Vector de datos de los SENTRON PAC

Figura 35. Variables de entrada/salida bloque Data_SetCom

	Name	Data type	Start value	Comment
4	Static			
5	statScanInterval	Time	T#2s	Los SENTRON PAC son procesados cada 2s.
6	instSequencer	*Sequencer*		Instanciación del bloque de programa Sequencer.
7	statMbClientInput	*typeMbClient*		Parámetros de entrada del bloque MB_CLIENT.
8	instMbClient	MB_CLIENT		Instanciación de los datos del bloque MB_CLIENT.
9	statNumberOfPacs	DInt	3	Nº de SENTRON PAC a comunicar
10	statPacIndex	Int	1	Índice del SENTRON PAC actualmente en comunicación.
11	statRestart	Bool	false	Marcador para la primera conexión con cada SENTRON.
12	statMbDataReal	Real	0.0	Transmisión de datos en formato Real.
13	statRetVal	Int	0	Transmisión de datos en formato Entero.
14	statNextPacPulse	Bool	false	Siguiente intervalo de escaneo.
15	instScanIntervalTimer	TON_TIME		Temporizador de intervalos.

Figura 36. Variables estáticas bloque Data_SetCom

Sistema SCADA para el control de tres centros de transformación

Al declarar “instMbClient” e “instScanIntervalTimer” como bloques de funciones ya existentes en la librería de TIA Portal; MB_CLIENT y TON_TIME, respectivamente, estas variables obtendrán las propiedades de sus funciones.

➤ Tags [DB2]

Tags es otro bloque de datos, pero mucho más simple, tan solo almacena las variables que se recogen de los SENTRON PAC4200, así como sus respectivas direcciones IP.

Creamos la variable “device”, la cual es un vector de tamaño 3, el número de analizadores que usamos, del tipo “typeDevice”. Cada elemento del vector referencia a cada SENTRON PAC, es por esto por lo que debemos modificar la variable “ipOctet4” de cada elemento, asignando el cuarto octeto de la dirección IPv4 de su respectivo analizador.

Como hemos visto en la tabla 2, el cuarto octeto del SENTRON PAC en el CT1 es 4, el del CT2 es 5 y el del CT3 es 6.

	Name	Data type	Start value
1	Static		
2	device	Array[1..3] of *typeDevice*	
3	device[1]	*typeDevice*	
4	ipOctet4	USInt	4
5	pTotal	Real	0.0
6	pTotalMax	Real	0.0
7	pTotalMin	Real	0.0
8	CorrienteL1	Real	0.0
9	CorrienteL2	Real	0.0
10	CorrienteL3	Real	0.0
11	VoltageL1-L2	Real	0.0
12	VoltageL2-L3	Real	0.0
13	VoltageL3-L1	Real	0.0
14	vTHD_L1	Real	0.0
15	vTHD_L2	Real	0.0
16	vTHD_L3	Real	0.0
17	cTHD_L1	Real	0.0
18	cTHD_L2	Real	0.0
19	cTHD_L3	Real	0.0
20	cosL1	Real	0.0
21	cosL2	Real	0.0
22	cosL3	Real	0.0
23	device[2]	*typeDevice*	
24	device[3]	*typeDevice*	

Figura 37. Bloque Tags

➤ SetCom [FB1]

SetCom es un bloque de funciones con lenguaje SCL, el cual nos permite programar escribiendo código. Esto facilita en gran medida la programación, ya que, mientras que con otros lenguajes estas limitado al uso de bloques de funciones que hay definidas, con SCL puedes usar estos bloques y realizar otro tipo de acciones las cuales serían imposibles o muy complicadas de desarrollar con otro lenguaje. Cabe destacar que nuestro PLC tiene la capacidad de ser programado en SCL ya que es un modelo muy completo, pero no todos los PLC pueden hacerlo.

Este bloque de programa trata de instanciar las funciones necesarias para establecer la comunicación entre los dispositivos.

Sistema SCADA para el control de tres centros de transformación

Antes de comenzar con el código, se debe diseñar la interfaz del bloque de función que estamos creando, es decir, declarar sus entradas, salidas y constantes. Para ello haremos clic en la flecha que se encuentra en la parte superior de la ventana del bloque SetCom, debajo del texto “Block interface”.

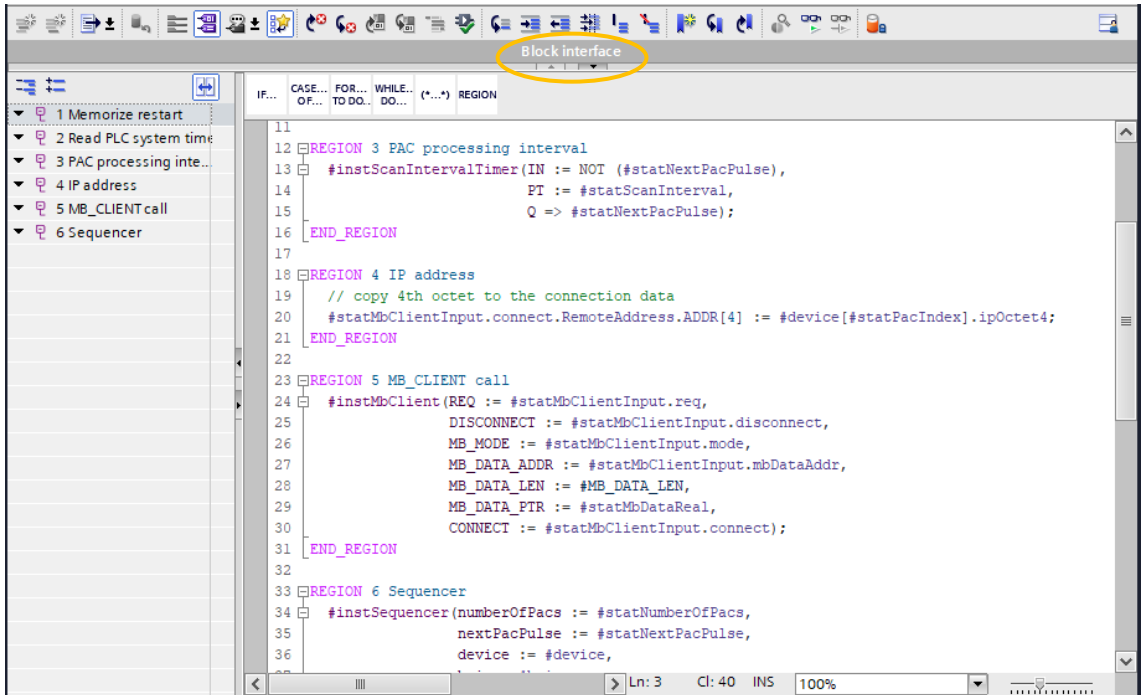


Figura 38. Abrir la interfaz de bloque

Al hacer clic en la flecha dos veces, la ventana tomará un aspecto similar a un bloque de datos. Como hemos visto anteriormente, se divide en diferentes campos dependiendo de la función de la variable.

En este caso, se añadirá la llamada del primer ciclo, “InitialCall” al campo Input, la variable “device” como un vector de tipo “typeDevice” en el campo InOut y la constante “MB_DATA_LEN”, que indica la longitud de los datos que se van a leer de los SENTRON PAC, cuyo valor será de dos registros, y que será declarada en el campo Constants.

	Name	Data type	Default value
1	Input		
2	initialCall	Bool	false

Figura 39. Entradas interfaz de bloque SetCom

	Name	Data type	Default value
5	InOut		
6	device	Array[*] of *typeDe...	

Figura 40. Entradas/Salidas interfaz de bloque SetCom

	Name	Data type	Default value
25	Constant		
26	MB_DATA_LEN	UInt	2

Figura 41. Constante interfaz de bloque SetCom

Sistema SCADA para el control de tres centros de transformación

Una vez quedan declaradas las variables y constantes, se procede a la programación del código. Las siguientes ilustraciones tratan de explicar a grandes rasgos su funcionamiento.

```
1 REGION 1 Memorize restart
2 IF (#initialCall = TRUE) THEN
3     #statRestart := TRUE; // set marker
4     #statNumberOfPacs := UPPER_BOUND(ARR := #device, DIM := 1); //count PACs
5 END_IF;
6 END_REGION
```

Figura 42. Región 1 del código SetCom

En su primer ciclo de trabajo, almacena el reinicio del sistema en una variable y cuenta el número de analizadores de consumo a controlar.

```
8 REGION 2 Read PLC system time
9     #statRetVal := RD_SYS_T(#hmi.sysTime);
10 END_REGION
```

Figura 43. Región 2 del código SetCom

En la región 2, obtiene la hora del sistema.

```
12 REGION 3 PAC processing interval
13 #instScanIntervalTimer(IN := NOT (#statNextPacPulse),
14                         PT := #statScanInterval,
15                         Q => #statNextPacPulse);
16 END_REGION
```

Figura 44. Región 3 del código SetCom

Como se ha mencionado previamente, la variable “#instScanIntervalTime” obtiene las propiedades del bloque de función TON_TIME, un temporizador. Esta variable es necesaria para sincronizar los intervalos de procesado de datos. Para configurar el temporizador se le asignan variables a sus entradas, IN (señal para comenzar la cuenta) y PT (tiempo establecido en segundos), y a la salida, Q (señal de finalización de la temporización).

```
18 REGION 4 IP address
19 // copy 4th octet to the connection data
20 #statMbClientInput.connect.RemoteAddress.ADDR[4] := #device[#statPacIndex].ipOctet4;
21 END_REGION
```

Figura 45. Región 4 del código SetCom

En esta región, se obtiene el cuarto octeto de la dirección IPv4 que va a ser procesada, accediendo al elemento adecuado declarado en el bloque de datos Tags.

```
23 REGION 5 MB_CLIENT call
24 #instMbClient(REQ := #statMbClientInput.req,
25              DISCONNECT := #statMbClientInput.disconnect,
26              MB_MODE := #statMbClientInput.mode,
27              MB_DATA_ADDR := #statMbClientInput.mbDataAddr,
28              MB_DATA_LEN := #MB_DATA_LEN,
29              MB_DATA_PTR := #statMbDataReal,
30              CONNECT := #statMbClientInput.connect);
31 END_REGION
```

Figura 46. Región 5 del código SetCom

Aquí se instancia el bloque de función MB_CLIENT, asignando las variables precisas de nuestro bloque de datos “Data_SetCom” a las entradas del bloque que sirven para su

control. Más tarde, en el bloque de programa “Sequencer”, se realizarán llamadas periódicas a esta región del código.

```
33 REGION 6 Sequencer
34 #instSequencer(numberOfPacs := #statNumberOfPacs,
35                 nextPacPulse := #statNextPacPulse,
36                 device := #device,
37                 hmi := #hmi,
38                 pacIndex := #statPacIndex,
39                 restart := #statRestart,
40                 instMbClient:=#instMbClient,
41                 mbClientInput:=#statMbClientInput,
42                 mbDataReal:=#statMbDataReal);
43 END_REGION
```

Figura 47. Región 6 del código SetCom

Finalmente, se procede, de una manera muy similar a la anterior región, a instanciar el bloque de programa que hemos creado, “Sequencer”, para poder realizar llamadas al bloque cuando sea necesario.

➤ Sequencer [FB2]

Este bloque de programa, al igual que SetCom, es un bloque de funciones en lenguaje SCL. La programación incluida en Sequencer es la encargada del control, la recogida y el procesamiento de los datos que provienen de los analizadores de consumo.

Primero se deben declarar como constantes las direcciones de los registros Modbus de los analizadores de consumo que van a ser usados. Los registros se declaran en la interfaz del bloque, del mismo modo que se ha declarado “MB_DATA_LEN” en SetCom. El valor de cada dirección de registro ha sido obtenido del manual de sistema de los SENTRON PAC. A continuación, se muestra la interfaz del bloque para ver que direcciones tienen las variables que se van a recoger en este estudio. Para consultar la tabla completa con todas las direcciones de registros que los analizadores son capaces de almacenar, acudir al anexo 8.3 Registros Modbus.

Sistema SCADA para el control de tres centros de transformación

	Name	Data type	Default value
27	Constant		
28	ADDR_TOTAL_ACTIVE...	UDInt	40066
29	ADDR_PTOTAL_MIN	UDInt	40198
30	ADDR_PTOTAL_MAX	UDInt	40140
31	ADDR_CURRENT_L1	UDInt	40014
32	ADDR_CURRENT_L2	UDInt	40016
33	ADDR_CURRENT_L3	UDInt	40018
34	ADDR_V_L1-L2	UDInt	40008
35	ADDR_V_L2-L3	UDInt	40010
36	ADDR_V_L3-L1	UDInt	40012
37	ADDR_vTHD_L1	UDInt	40262
38	ADDR_vTHD_L2	UDInt	40264
39	ADDR_vTHD_L3	UDInt	40266
40	ADDR_cTHD_L1	UDInt	40268
41	ADDR_cTHD_L2	UDInt	40270
42	ADDR_cTHD_L3	UDInt	40272
43	ADDR_COS_L1	UDInt	40244
44	ADDR_COS_L2	UDInt	40246
45	ADDR_COS_L3	UDInt	40248
46	READ	USInt	0
47	WRITE	USInt	1

Figura 48. Registros Modbus usados

Cada vez que SetCom llama a este bloque se ejecuta el código de manera secuencial hasta recoger todos los datos necesarios. Solo saldrá de la secuencia si sucede un error en la comunicación. El código cuenta con cuatro regiones o estados los cuales se encargan de establecer y mantener la correcta comunicación con los analizadores, la mayoría de los estados tratan de obtener los diferentes datos que se necesitan para el control. La siguiente tabla resume los posibles estados del código:

Estado	Función
Estado 0 (step 0)	Iniciar la comunicación.
Estados 10 y 11 (step 10 and 11)	Obtención de la potencia activa total.
Estados 12 y 13 (step 12 and 13)	Obtención de la potencia activa total máxima registrada.
Estados 14 y 15 (step 14 and 15)	Obtención de la potencia activa total mínima registrada.
Estados 16 y 17 (step 16 and 17)	Obtención de la corriente de la fase L1.
Estados 18 y 19 (step 18 and 19)	Obtención de la corriente de la fase L2.
Estados 20 y 21 (step 20 and 21)	Obtención de la corriente de la fase L3.
Estados 22 y 23 (step 22 and 23)	Obtención de la tensión en la fase L1 sobre L2.
Estados 24 y 25 (step 24 and 25)	Obtención de la tensión en la fase L2 sobre L3.
Estados 26 y 27 (step 26 and 27)	Obtención de la tensión en la fase L3 sobre L1.
Estados 28 y 29 (step 28 and 29)	Obtención de la distorsión armónica total (THD) en tensión de la fase L1.
Estados 30 y 31 (step 30 and 31)	Obtención de la THD en tensión de la fase L2.
Estados 32 y 33 (step 32 and 33)	Obtención de la THD en tensión de la fase L3.
Estados 34 y 35 (step 34 and 35)	Obtención de la THD en corriente de la fase L1.
Estados 36 y 37 (step 36 and 37)	Obtención de la THD en corriente de la fase L2.
Estados 38 y 39 (step 38 and 39)	Obtención de la THD en corriente de la fase L3.
Estados 40 y 41 (step 40 and 41)	Obtención del Cos φ en L1.
Estados 42 y 43 (step 42 and 43)	Obtención del Cos φ en L2.

Estados 44 y 45 (step 44 and 45)	Obtención del Cos ϕ en L3.
Estado 70 (step 70)	Espera a que el SENTRON PAC no esté ocupado.
Estado 80 (step 80)	Finaliza la conexión y cambia el SENTRON PAC que está siendo procesado.
process_comm_error	Comprueba si ha habido algún problema en la comunicación.

Tabla 5. Estados del código Sequencer

Todas las regiones del código tienen una estructura común la cual asegura el funcionamiento secuencial de la programación. Cada estado de la secuencia se denomina “step N”, siendo N el número de identificación de cada región. La estructura común es la siguiente:

```

1 REGION step N: Initial step after restart
2 IF (#statStep = N) THEN
3   IF (*condición*) THEN
4
5     //CÓDIGO AQUÍ
6
7     #statStep := N+1;
8   END_IF;
9   GOTO process_comm_error; // check communication error
10  END_IF;
11 END_REGION
    
```

Figura 49. Estructura común código Sequencer

La variable “#statStep” funciona como índice de la secuencia para saber en qué estado se encuentra o a cuál debe pasar, donde N es el número del índice del estado actual de la secuencia y N+1 el estado siguiente en la secuencia.

En la línea 2 del código, se verifica que ha entrado a la región de código adecuada según “#statStep”, si se cumple esta condición, se realizarán las acciones programadas en este estado de la secuencia y seguidamente cambiará el valor del índice al que debe dirigirse a continuación, N+1. Por último, se comprueba si ha habido algún problema en la comunicación, línea 9 del código, llamando a la región de código “process_comm_error”. Tras la comprobación, el programa volverá al estado de la secuencia correspondiente.

Seguidamente se procede a explicar la función de las regiones del código, más allá de la estructura que comparten.

```

1 REGION step 0: Initial step after restart
2 IF (#statStep = 0) THEN
3   IF (#nextPacPulse = TRUE) THEN // next PAC to communicate
4     #mbClientInput.disconnect := FALSE; // start connection
5     #statStep := 10;
6   END_IF;
7   GOTO process_comm_error; // check communication error
8   END_IF;
9 END_REGION
    
```

Figura 50. Estado 0 del código Sequencer

El estado 0 comienza la comunicación con el SENTRON PAC. Habiendo asignado la variable “#nextPacPulse” a la salida del temporizador TON_TIME, esta se activará cada vez que el tiempo de intervalo establecido se cumpla. De este modo, el código del bloque “Sequencer” comienza cuando “#nextPacPulse” cumpla la condición de ser verdadero, o

“TRUE”. “#MbClientInput” sale del modo desconexión, y se aumenta el índice de la secuencia al siguiente estado, el 10.

En la tabla 5 se puede apreciar que los estados del 10 al 45 se encargan de recoger los datos que nos proporcionan los analizadores de consumo. La obtención de cada uno de los datos es prácticamente idéntica, solo hay que cambiar la dirección del registro al que se accede, es por esto por lo que a continuación se explican simplemente las regiones 10 y 11, para comprobar el código completo, referir al anexo 8.1. Código Sequencer.

```

11 REGION step 10: pTot. Waiting for CONNECTED and NOT(BUSY)
12 IF (#statStep = 10) THEN
13     IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN
14         #mbClientInput.mbDataAddr := #ADDR_TOTAL_ACTIVE_POWER; // TOTAL_POWER
15         #mbClientInput.mode := #READ; // read
16         #mbClientInput.req := TRUE; // start MB_CLIENT
17         #statStep := 11;
18     END_IF;
19     #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred?
20     GOTO process_comm_error; // check communication error
21 END_IF;
22 END_REGION
    
```

Figura 51. Estado 10 del código Sequencer

```

24 REGION step 11: Waiting for reception pTot DONE
25 IF (#statStep = 11) THEN
26
27     IF (#instMbClient.DONE = TRUE) THEN
28
29         #tempReal := #mbDataReal / 1000.0;
30
31         #device[#pacIndex].pTotal := #tempReal;
32         #statStep := 12;
33
34     END_IF;
35
36     #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred?
37     GOTO process_comm_error; // check communication error
38
39 END_IF;
40 END_REGION
    
```

Figura 52. Estado 11 del código Sequencer

El estado 10 espera que el SENTRON PAC esté disponible, es decir, conectado y no ocupado. Cuando se cumple, envía una petición de lectura de dato a la dirección de registro indicada en la variable “#mbClientInput.mbDataAddr” y pasa al estado 11.

Una vez en el estado 11 se almacena el dato en su variable adecuada del analizador que está siendo procesado actualmente. El SENTRON PAC devuelve el valor de la potencia en vatios, pero en el caso de este estudio es preferible la visualización en kilovatios. Se puede observar en la línea 29 de la región 11, que se realiza la conversión del dato sirviéndose de una variable auxiliar “tempReal”. Esta conversión solo es necesaria en los datos relacionados con la potencia.

Finalmente, pasa al siguiente estado para recoger el valor del próximo registro, repitiendo el proceso que se acaba de describir. El último estado de recogida de datos, el paso 45, dirige al 70.

Sistema SCADA para el control de tres centros de transformación

```
543 REGION step 70: Waiting for NOT(BUSY)
544 IF (#statStep = 70) THEN
545
546 IF (#instMbClient.BUSY = FALSE) THEN // MB_CLIENT is stopped?
547     #mbClientInput.disconnect := TRUE; // disconnection
548     #statStep := 80;
549 END_IF;
550
551 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred?
552 GOTO process_comm_error; // check communication error
553
554 END_IF;
555 END_REGION
```

Figura 53. Estado 70 del código Sequencer

En el estado 70 el código espera a que se haya recogido la última medida requerida, y cuando el analizador no se encuentra ocupado envía la señal de desconexión de la función MB_CLIENT.

```
557 REGION step 80: Waiting for NOT(CONNECTED) and NOT(BUSY)
558 IF (#statStep = 80) THEN
559
560 IF (#instMbClient.Connected = FALSE) AND (#instMbClient.BUSY = FALSE) THEN
561
562     // resetting first scan marker
563 IF (#restart = TRUE) AND (#pacIndex = #numberOfPacs) THEN
564     #restart := FALSE;
565 END_IF;
566
567     #pacIndex := #pacIndex + 1; // increment PAC counter
568
569 IF (#pacIndex > #numberOfPacs) THEN
570     #pacIndex := 1; // reset PAC counter
571 END_IF;
572
573     #statStep := 0; // start sequencer for next PAC
574
575 END_IF;
576
577 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred?
578 GOTO process_comm_error; // check communication error
579
580 END_IF;
581 END_REGION
```

Figura 54. Estado 80 del código Sequencer

Por último, en el estado 80 se confirma la desconexión del bloque MB_CLIENT y se incrementa en uno la variable “#pacIndex” para proceder a la recogida de datos del siguiente analizador. Para terminar, devuelve el código a su comienzo, el estado 0, con el fin de continuar la monitorización de todos los C.T.

```

583 REGION React on communication timeout
584 process_comm_error:
585 IF (#tempTimeoutComm = TRUE) THEN // does puls for next PAC occurred?
586
587     #mbClientInput.req := FALSE; // stopping MB_CLIENT
588     #mbClientInput.disconnect := TRUE; // disconnection
589
590 IF (#device[#pacIndex].commBreakdown.started = FALSE) THEN
591     #device[#pacIndex].commBreakdown.started := TRUE; // set marker
592     // calculate time when "breakdown" message should be displayed
593     #device[#pacIndex].commBreakdown.errMsgTime := T_ADD(IN1 := #hmi.sysTime,
594                                                         IN2 := #statCommErrMsgDelay);
595
596     END_IF;
597
598     #statStep := 80; // check if MB_CLIENT is stopped and disconnected
599
600     END_IF;
601 END_REGION

```

Figura 55. Estado process_comm_error del código Sequencer

Al final de cada estado se realiza una comprobación para saber si ha habido problemas de comunicación. Si se cumple el intervalo establecido de 2 segundos, el bloque TON_TIME activa la señal “#nextPacPulse”, la cual a su vez dispara la variable “#tempTimeoutCom”. Si esta variable se encuentra activada cuando se realiza la llamada a process_comm_error, se procede a la desconexión del SENTRON PAC actual y se dirige el código al estado 80 con el fin de proseguir con el siguiente analizador de consumo.

➤ Main [OB1]

El bloque “Main” es un programa cíclico, en lenguaje FBD, encargado de llamar constantemente al bloque de función “SetCom”. Para configurar el bloque “Main” basta con hacer clic sobre “SetCom” en el árbol de proyecto y arrastrarlo a la ventana de programación de “Main”. Al soltarlo, TIA Portal genera un mensaje para determinar el bloque de datos que va a usar la función. De manera predefinida se crea un nuevo bloque de datos, pero en este caso se debe elegir el bloque de datos que nosotros hemos creado, “Data_SetCom”. El aspecto final de la ventana de “Main” es el siguiente:

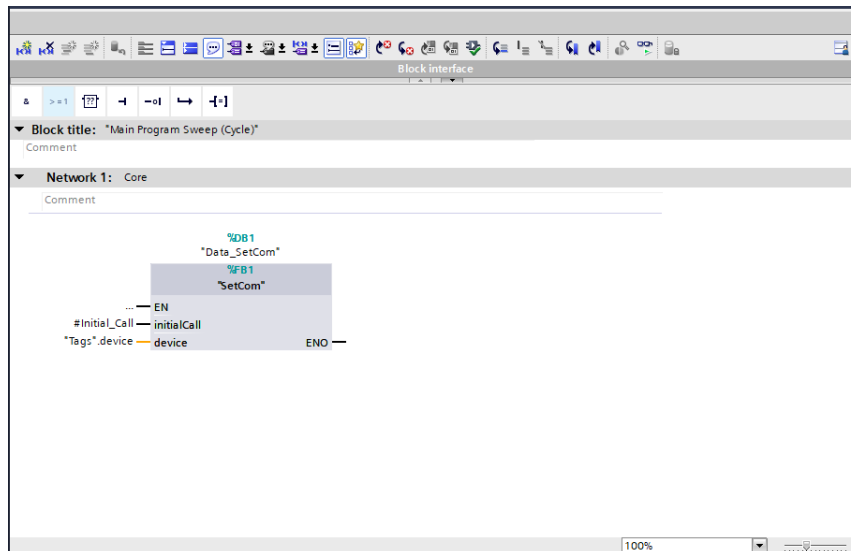


Figura 56. Bloque Main

5.6. Diseño del HMI

El diseño de las pantallas es una parte importante del proyecto, ya que deben ser lo más intuitivas posible y que aporten toda la información necesaria de manera visualmente atractiva. Cabe recordar que el diseño de todas las pantallas son un mero ejemplo, y que pueden ser modificadas al gusto del usuario siempre y cuando las variables que se usen sean las adecuadas.

En este apartado se pretende mostrar los diseños finales de las diferentes pantallas de los HMI, señalando y explicando los objetos con propiedades especiales, como eventos o animaciones, pero sin ahondar en características que no son relevantes o pueden ser apreciadas visualmente, como las medidas o colores de rectángulos, líneas, textos...

Antes de comenzar, se debe comprobar que se han seguido las indicaciones del apartado 5.2 sobre el asistente de configuración de los HMI. Si así ha sido, se debe haber creado una pantalla llamada "SIMATIC PLC system" en cada HMI, con el siguiente aspecto:

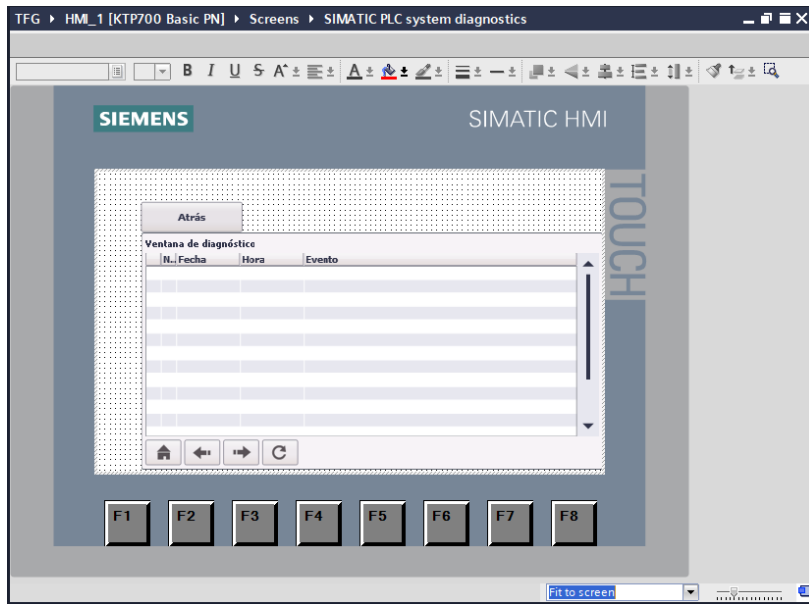


Figura 57. Ventana de diagnóstico del asistente

5.6.1. Diseño de la plantilla

Primero vamos a crear una plantilla para usarla en las pantallas y no tener que repetir ciertas partes del diseño. Para ello, en el árbol de proyecto abriremos la carpeta "Screen management", después, dentro de "Templates" haremos doble clic en "Add new template". Una vez creada la plantilla le cambiamos el nombre a "Plantilla" y la marcamos como plantilla predeterminada en este proyecto haciendo clic derecho sobre su nombre en el árbol de proyecto y seleccionando "Define as default template".

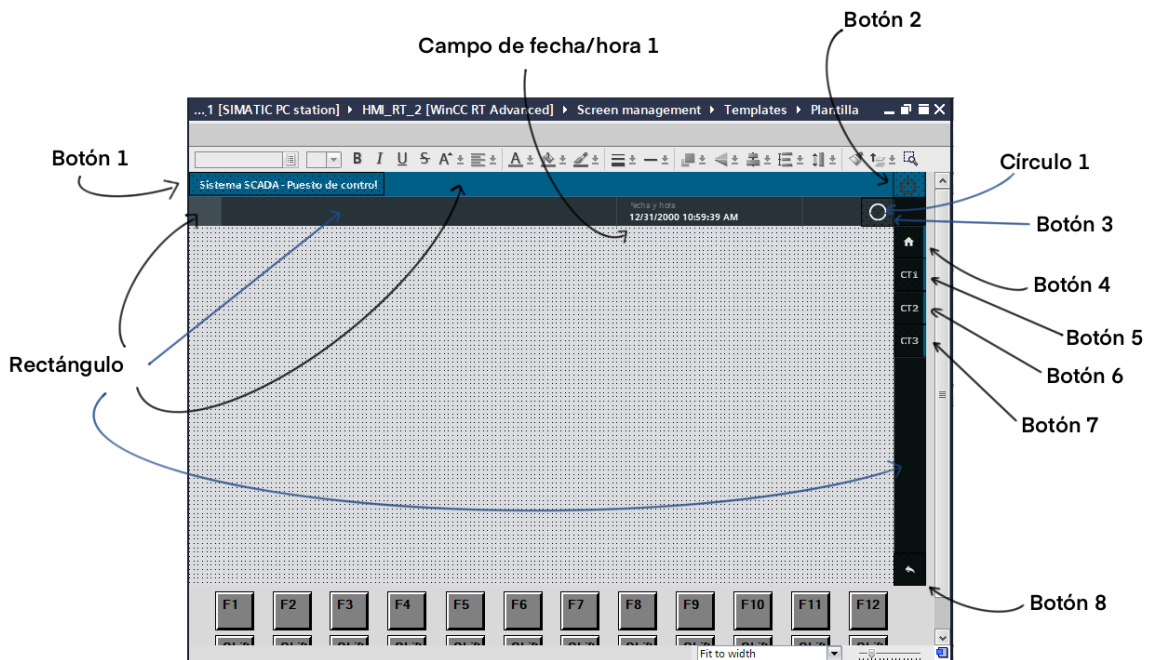


Figura 58. Diseño de la plantilla

En la parte derecha de TIA Portal se encuentra la “Toolbox” o sección de herramientas, de donde sacaremos todos los objetos que vayamos a usar en nuestros HMI. Para usarlos no hace falta más que hacer clic sobre ellos y arrastrarlos a la ventana que estamos diseñando.

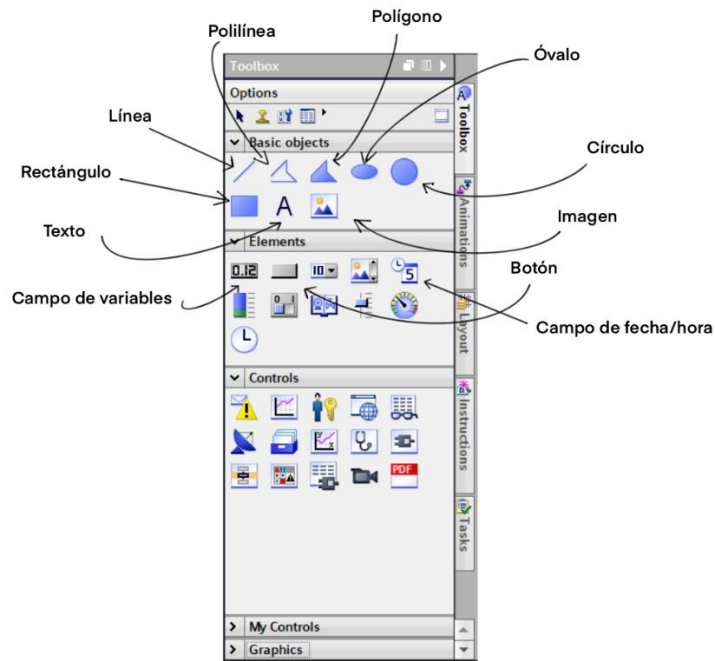


Figura 59. Sección de herramientas

➤ Botón 1

Se trata de un botón invisible que dirige a la pantalla de inicio. Los botones invisibles son muy útiles cuando quieres que una imagen, forma o texto pueda ser pulsado.

Cogemos un botón de las herramientas, lo colocamos en el sitio deseado y abrimos sus propiedades. En las propiedades generales, en la sección “Mode” marcamos la opción “Invisible”.

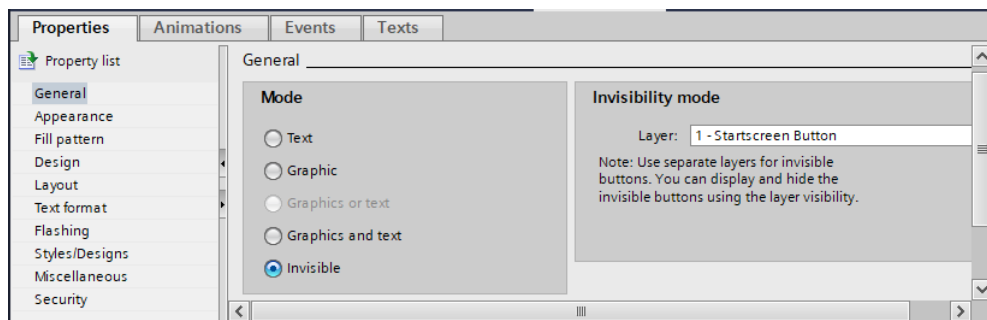


Figura 60. Configuración de la visibilidad de un botón

Luego, en la pestaña “Events” programamos el cambio de pantalla cuando se haga clic sobre este botón. Seleccionamos la opción “Click”, añadimos la función “ActivateScreen” e indicamos la pantalla a la que queremos que se dirija.

Sistema SCADA para el control de tres centros de transformación

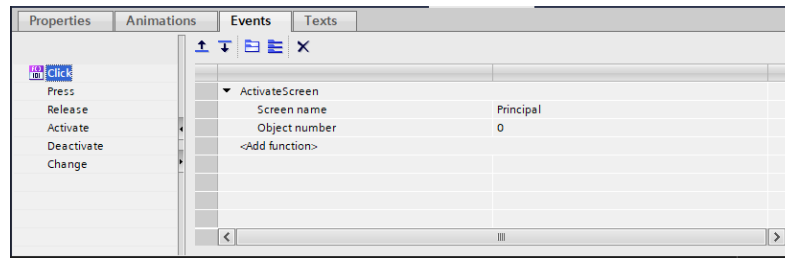


Figura 61. Configuración de eventos de botón

➤ Botón 2

En esta plantilla, todos los botones son invisibles. El botón 2 se encarga de cerrar la ventana de visualización del sistema, haciendo uso de la función “StopRuntime” en el evento de “Click”.

➤ Campo de fecha/hora 1

Arrastramos a la ventana el elemento y abrimos sus propiedades. Desde el apartado general podemos configurar que muestre la hora, la fecha o ambas cosas, así como indicar si debe usar la hora del sistema, como en este caso, u otra variable de tipo “DateTime”. La configuración usada para este campo es la siguiente:

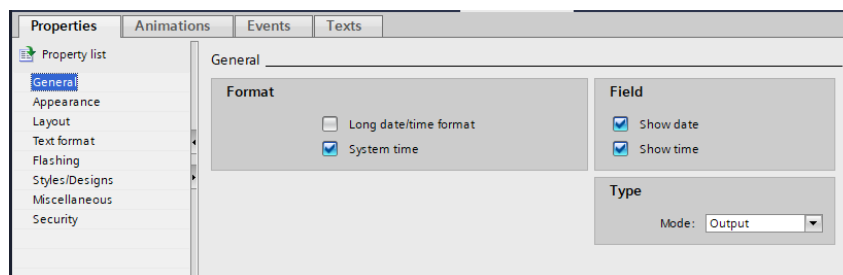


Figura 62. Configuración campo de fecha/hora

➤ Botón 3

El botón 3, también invisible, dirige al usuario a la pantalla “SIMATIC PLC system diagnostics” haciendo uso de la función “ActivateScreen” con el evento “Click”

➤ Círculo 1

Este círculo tiene una animación asociada a la variable de sistema “@DiagnosticsIndicatorTag”, la cual se activa cuando el diagnóstico detecta algún problema. Para conseguirlo, entramos en la pestaña “Animations” y dentro de la sección “Display” hacemos doble clic en “Add new animation”. Indicamos la variable que queremos procesar en el apartado “Tag”, y luego los cambios que queremos que adopte con sus respectivos valores de la variable.

En este caso, mientras la variable “@DiagnosticsIndicatorTag” se mantenga en 0, el interior del círculo es de color verde, y cuando la variable se activa (cambia de valor a 1), el interior cambia de color a rojo.

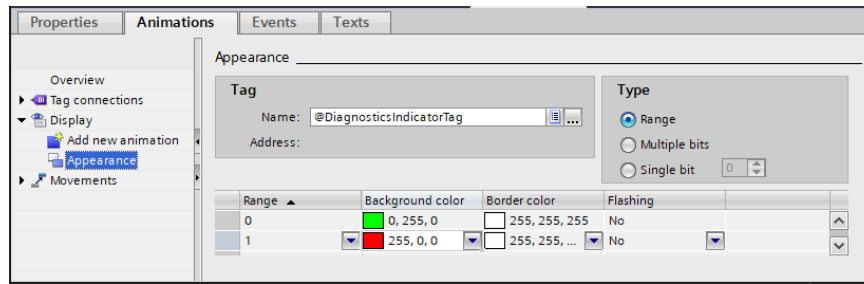


Figura 63. Configuración de la animación de apariencia de un objeto

➤ Botón 4

Este botón es idéntico al botón 1 descrito previamente.

➤ Botones 5, 6 y 7

Estos botones envían al usuario a las pantallas de los centros de transformación 1, 2 y 3 respectivamente. Para ello hacen uso de la función "ActivateScreen" con el evento "Click"

➤ Botón 8

Por último, este botón devuelve al usuario a la pantalla en la que se encontraba anteriormente. También conectado al evento "Click" hace uso de la función "ActivatePreviousScreen".

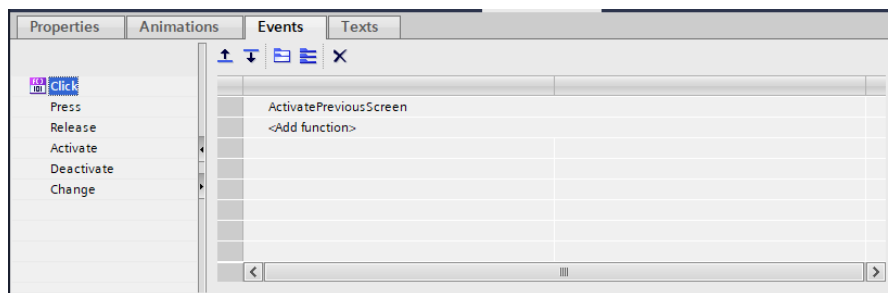


Figura 64. Evento "ActivatePreviousScreen"

5.6.2. Diseño del WinCC Runtime Advanced

Una vez creada la plantilla añadiremos las ventanas que vamos a necesitar en nuestro proyecto. Al tratarse de la pantalla del puesto de mando, tendremos ocho ventanas: una pantalla principal, la ventana de diagnóstico que hemos creado previamente y ventanas por cada CT, la del estado de sus variables y la de visualización de los datos obtenidos por los analizadores de consumo.

Desde el árbol de proyecto, en la carpeta "Screens" haremos doble clic en "Add new screen" ocho veces para crear todas las ventanas. Tras esto las renombraremos como queramos y definiremos la pantalla principal como ventana de inicio, todo esto haciendo clic derecho sobre las ventanas en el árbol.

Sistema SCADA para el control de tres centros de transformación

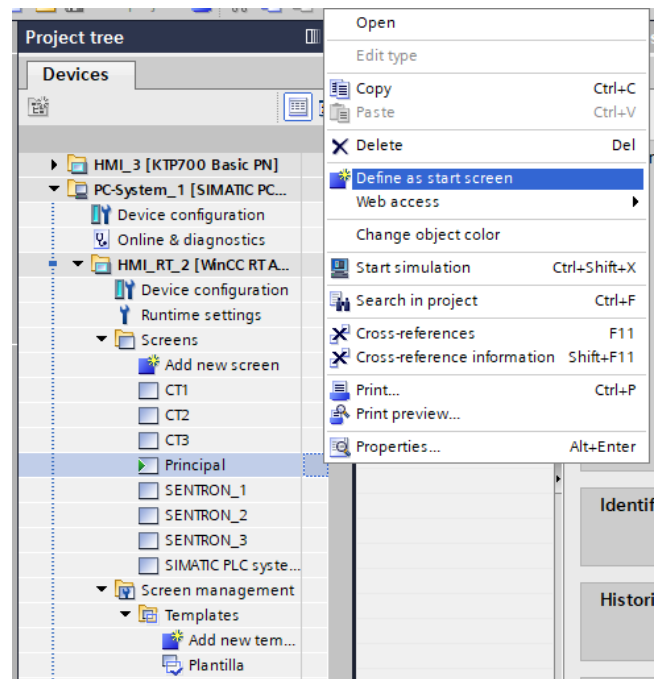


Figura 65. Creación de las pantallas para WinCC

➤ Ventana Principal

Desde esta pantalla se puede apreciar una visión general del sistema, donde se observa si los centros están proporcionando energía, ya sea de red o de grupo. Además se puede acceder a cada CT haciendo clic sobre sus respectivos nombres.

La ventana desarrollada es la siguiente:

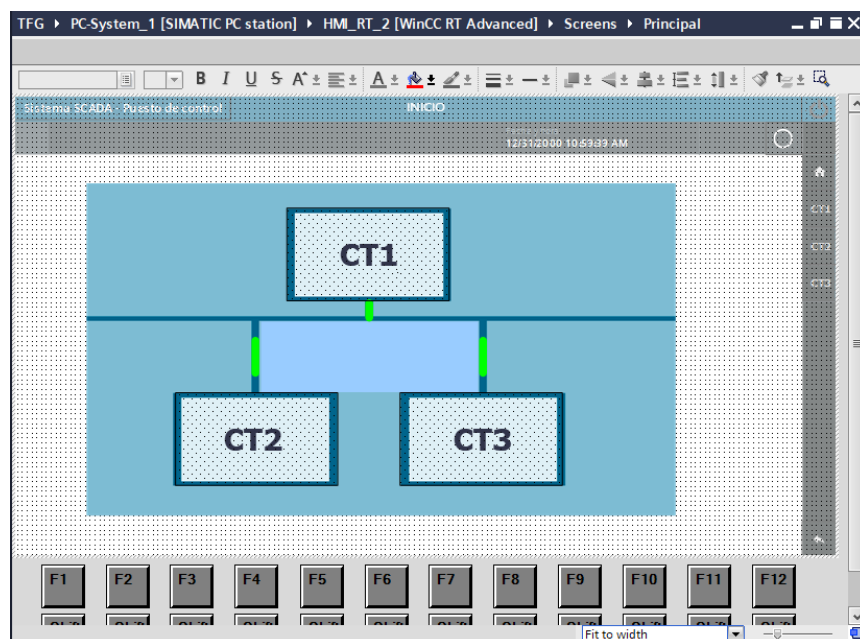


Figura 66. Pantalla principal de WinCC

Se ha seleccionado una imagen prediseñada adecuada al contexto como fondo, y se le han sobrepuesto elementos. Para añadir la imagen se debe arrastrar la opción de imagen de las herramientas a la ventana y seleccionar el archivo deseado.

Sistema SCADA para el control de tres centros de transformación

Tras esto, se han añadido 3 etiquetas de texto una para CT con un botón invisible para ir a la ventana de sus respectivos CT. Y, por último, las líneas verdes están condicionadas a las variables “RedGrupo_TR1”, “RedGrupo_TR2” y “RedGrupo_TR3”, del mismo modo que en la figura 66, pero en este caso con un “verdadero” en la variable, la línea es verde y al contrario, roja.

➤ Ventana SIMATIC PLC system diagnostics

En esta pantalla podremos ver el estado en vivo de cada componente del sistema, PLC, periféricas descentralizadas y HMIs.

De la pantalla que hemos generado con el asistente, nos quedamos solamente con la ventana de diagnóstico, y eliminamos los demás componentes. Finalmente debemos asegurarnos de que se haya aplicado la plantilla, ya que al haber creado esta pantalla antes que la plantilla, puede no tenerla. El aspecto final de esta ventana es el siguiente:

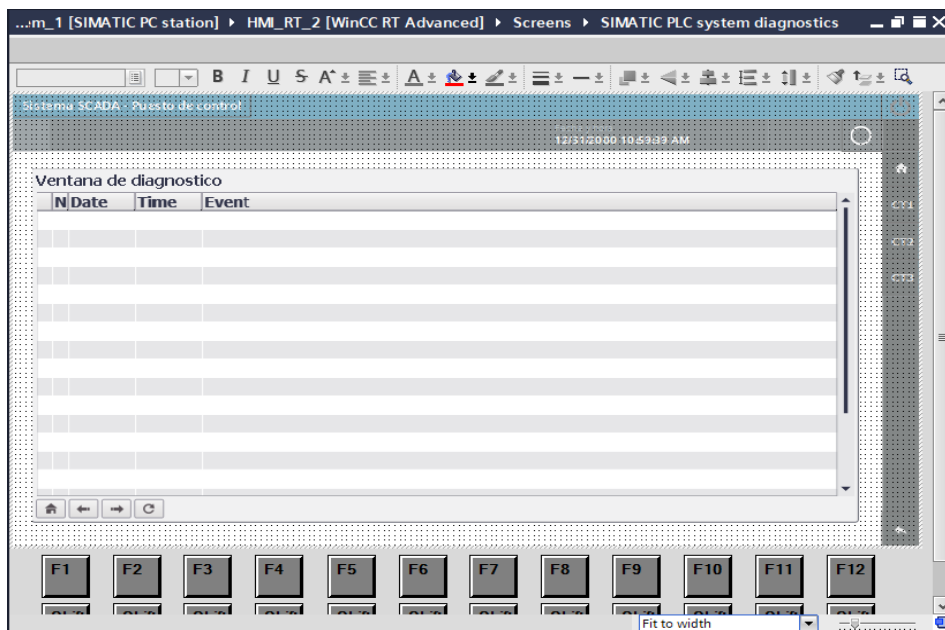


Figura 67. Pantalla de diagnóstico en WinCC

➤ Ventanas CT1, CT2 y CT3

Desde estas ventanas podemos comprobar el estado de las variables de cada CT y acceder a sus SENTRON PAC. Las ventanas de cada centro son idénticas salvo por unos detalles: en cada pantalla se recogerán las variables de sus respectivos CT, y además se ha creado un indicador en la barra superior para que el usuario conozca el CT que está visualizando.



Figura 68. Identificación pantallas CT

A continuación, se muestra el aspecto propuesto para estas ventanas.

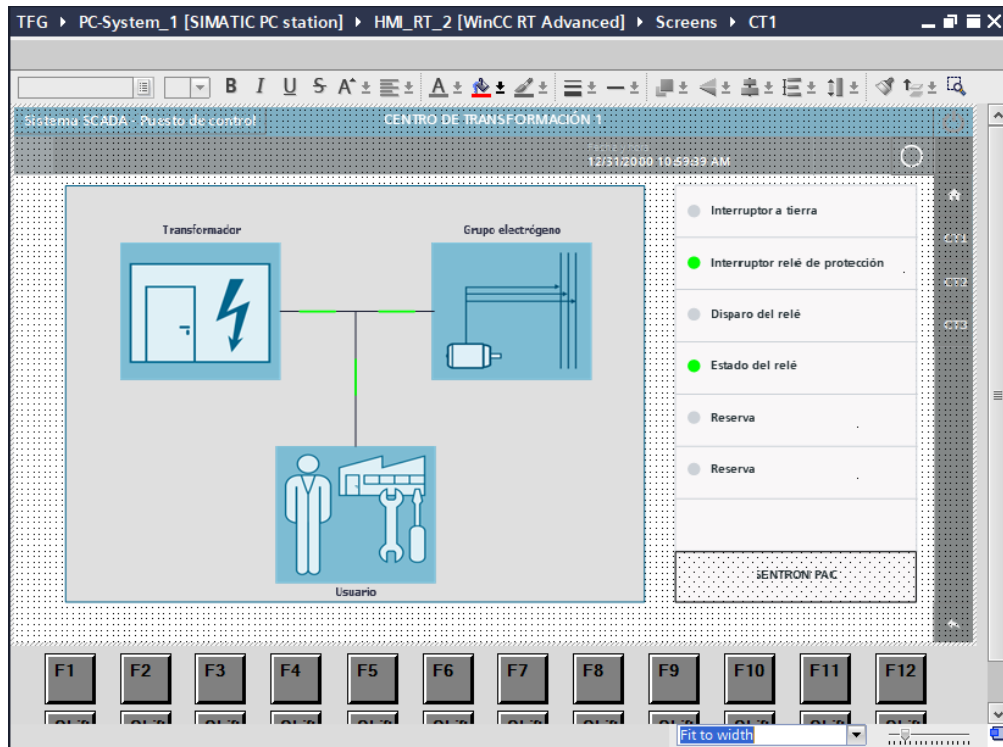


Figura 69. Pantalla CT1 en WinCC

Como se puede apreciar, se ha hecho uso de otras tres imágenes prediseñadas y varios rectángulos y líneas con carácter meramente ilustrativo. Los elementos importantes son los círculos que acompañan a las etiquetas de texto, el botón invisible sobre el texto “SENTRON PAC” y las líneas verdes.

Se ha dejado dos variables de reserva como posible ampliación del sistema. El estado del resto de variables, se definen por el color del círculo a su izquierda. Estos círculos están condicionados a variables, del mismo modo que se ha hecho anteriormente.

- Interruptor a tierra: el círculo está condicionado a la variable “EarthingSwitchOpen_J01-NC”. En condiciones normales, el círculo se mantiene gris, y si la variable cambia a 0, el círculo se vuelve rojo
- Interruptor relé de protección: controla el estado del círculo la variable “CbOpen_J01-NC”. Normalmente el círculo es verde, pero cuando el relé se abre y la variable se vuelve 0, el círculo cambia a rojo.
- Disparo de relé: manejado por la variable “CbTrippingSignal_J01-NA”, el círculo es gris en condiciones normales, pero cuando el relé se dispara, la variable se activa y el círculo pasa a ser rojo.
- Estado del relé: condicionado por la variable “ProtectionDeviceOk_J01-NA”, el círculo se encuentra de color verde mientras el relé esté bien, pero cuando este falle, la variable se activa y el círculo cambia a rojo.

Las líneas verdes indican de donde proviene la energía. La línea que sale del transformador está conectado a la variable “Red_TR1” y será verde mientras esta sea 1, en otro caso la línea será roja. Del mismo modo, la línea que sale del grupo electrógeno está condicionada a la variable “Grupo_TR1” y la que llega al usuario a la variable “RedGrupo_TR1”.

Finalmente, asignamos al botón invisible la función “ActivateScreen” con el evento “Click” para que nos lleve a la ventana “SENTRON_1”.

➤ Ventanas SENTRON_1, SENTRON_2 y SENTRON_3

En estas pantallas podremos observar a tiempo real los valores que están registrando los analizadores así como datos históricos de la potencia.

Igual que en el apartado anterior, estas tres ventanas son idénticas, cambiando las variables, y una etiqueta identificativa de lo que estamos visualizando.

Se ha diseñado una pantalla con rectángulos, etiquetas de texto y campos de variables de tal modo que su aspecto es similar a una tabla.

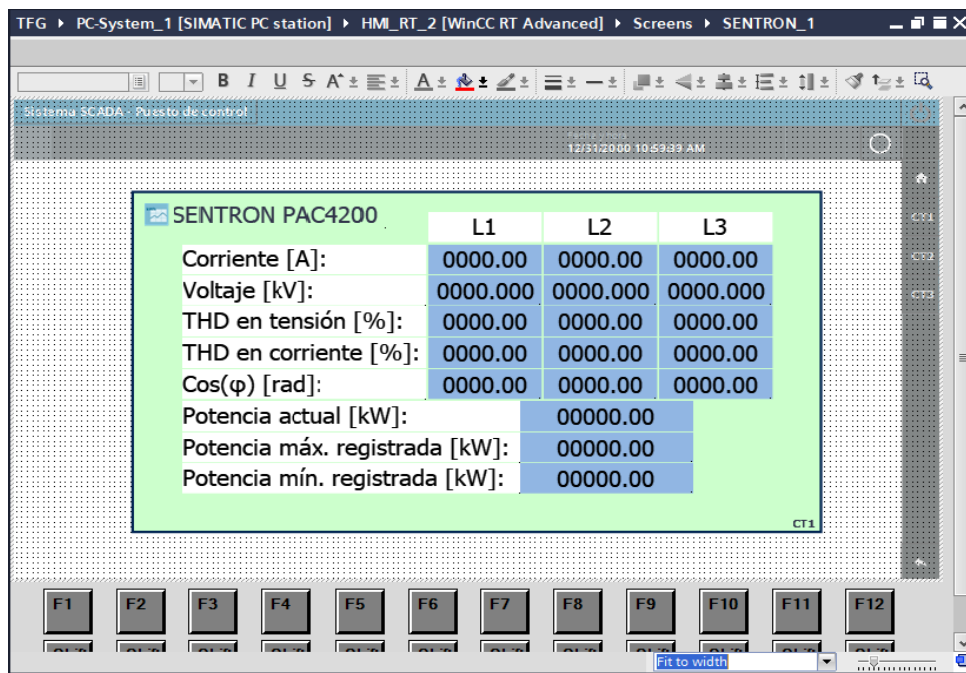


Figura 70. Pantalla SENTRON_1

En la esquina inferior derecha podemos observar la etiqueta identificativa de cada CT:

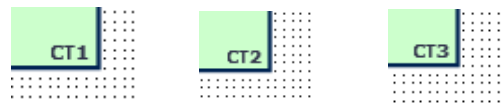


Figura 71. Identificación pantallas SENTRON

A continuación se muestra una tabla con las variables del bloque de datos “Tags” que se usan en las etiquetas que observamos en la figura 73. Hay que tener en cuenta que el direccionamiento completo de las variables es “Tags_device{X}_variable”, donde “X” es el elemento del vector device que se trata y “variable”, cualquiera de las variables del elemento en cuestión. En la tabla solo se muestra el nombre de la variable, por simplicidad.

	L1	L2	L3
Corriente	CorrienteL1	CorrienteL2	CorrienteL3
Voltaje	VoltageL1-L2	VoltageL2-L3	VoltageL3-L1
THD en tensión	vTHD_L1	vTHD_L2	vTHD_L3
THD en corriente	cTHD_L1	cTHD_L2	cTHD_L3

Cos φ	cosL1	cosL2	cosL3
Potencia actual		pTotal	
Pot. máx. registrada		pTotalMax	
Pot. min. registrada		pTotalMin	

Tabla 6. Campos de variable de las pantallas SENTRON

Para las pantallas de cada CT tendremos en cuenta su respectivo elemento del vector device, cambiando "X" en "Tags_device{X}_variable" por 1, 2 y 3.

Tras esto, ya tenemos diseñadas todas las ventanas de nuestro proyecto para WinCC.

5.6.3. Diseño de las pantallas KTP700

TIA Portal es un programa bastante avanzado y completo, otra de sus funcionalidades es que es capaz de redimensionar las ventanas que se diseñan a diferentes resoluciones. Es por esto por lo que podemos reutilizar las pantallas de punto anterior simplemente arrastrándolas en el árbol de proyecto, de la carpeta de WinCC a la de cada HMI.

La función de las pantallas es ofrecer información del estado del CT donde se encuentra de un vistazo, por eso son necesarias menos ventanas. Para cada HMI arrastraremos la plantilla y solo sus respectivas ventanas de CT y de SENTRON PAC, por ejemplo: en el HMI 1 debe haber tres pantallas, "CT1", "SENTRON_1" y "SIMATIC PLC system diagnostics".

6. Conclusiones

Podemos concluir que este estudio ha sido terminado de manera satisfactoria, cumpliendo los objetivos perseguidos desde su comienzo y habiendo desarrollado así un sistema compacto y eficiente que nos permite la visualización de todos los centros de transformación sin necesidad de salir del puesto de control.

El PLC seleccionado es un modelo muy potente el cual podría soportar muchos mas centros de transformación a controlar, además de ser fácilmente expandible con módulos compatibles. Es por esto por lo que este estudio, aunque reducido a tres centros de transformación y a la recogida de nueve variables, puede ser escalado a la cantidad de centros de transformación o variables a estudiar necesarios sin mucha complicación.

Así como las nociones de automática y electrónica digital obtenidas durante la carrera han sido útiles durante el desarrollo de este proyecto, existen otras competencias en las que he tenido que iniciarme, como la programación en TIA Portal, la programación SCL o la selección de componentes para el diseño del sistema.

7. Bibliografía

Logicbus. (s.f.). Obtenido de <https://www.logicbus.com.mx/Modbus.php>

Nagel, M. (31 de Enero de 2019). Obtenido de Siemens News:

<https://new.siemens.com/global/en/company/stories/industry/when-the-smart-factory-is-close-at-hand.html>

Serrano Gallego, A. (2017). Obtenido de

https://riunet.upv.es/bitstream/handle/10251/87579/45800771C-TFG_14993712123904276396153328323447.pdf?sequence=2&isAllowed=y

8. Anexos

8.1. Código Sequencer

Totally Integrated Automation Portal									
TFG / PLC_1 [CPU 1512SP-1 PN] / Program blocks									
Sequencer [FB2]									
Sequencer Properties									
General									
Name	Sequencer	Number	2	Type	FB				
Language	SCL	Numbering	Automatic						
Information									
Title		Author		Comment					
Family		Version	0.1	User-defined ID					
Sequencer									
Name	Data type	Default value	Retain	Access-ible from HMI/OPC UA	Wri-ta-ble in engi-neer-ing from HM I/O PC UA	Visible in HMI point	Set-point	Super- vision	Comment
▼ Input									
numberOfPacs	DInt	5	Non-retain	False	False	False	False		Number of PACs to communicate
nextPacPulse	Bool	false	Non-retain	True	True	True	False		Communication trigger to the next PAC
Output									
▼ InOut									
▼ device	Array[*] of "typeDe-vice"			False	False	False	False		Array of SENTRON PAC data
▼ device[*]	"typeDe-vice"			False	False	False	False		Array of SENTRON PAC data
ipOctet4	USInt			False	False	False	False		Fourth octet of the ip address
pTotal	Real			False	False	False	False		Average value of the total active power (in W)
pTotalMax	Real			False	False	False	False		
pTotalMin	Real			False	False	False	False		
CorrienteL1	Real			False	False	False	False		
CorrienteL2	Real			False	False	False	False		
CorrienteL3	Real			False	False	False	False		
VoltageL1-L2	Real			False	False	False	False		
VoltageL2-L3	Real			False	False	False	False		
VoltageL3-L1	Real			False	False	False	False		
vTHD_L1	Real			False	False	False	False		

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment	
vTHD_L2	Real			False	False	False	False			
vTHD_L3	Real			False	False	False	False			
cTHD_L1	Real			False	False	False	False			
cTHD_L2	Real			False	False	False	False			
cTHD_L3	Real			False	False	False	False			
cosL1	Real			False	False	False	False			
cosL2	Real			False	False	False	False			
cosL3	Real			False	False	False	False			
pacIndex	Int	1	Non-retain	True	True	True	False		Actual PAC number to communicate	
restart	Bool	false	Non-retain	True	True	True	False		Marker for the first communication of all PACs	
▼ mbClientInput	"type-MbClient"			False	False	False	False		Input parameter for MB_CLIENT	
req	Bool			False	False	False	False		Activates the requested transmission if TRUE	
disconnect	Bool			False	False	False	False		Initiates a disconnect operation	
mode	USInt			False	False	False	False		Specifies the type of request: read, write or diagnostic	
mbDataAddr	UDInt			False	False	False	False		Specifies the starting address of the data to be accessed in the Modbus client	
▼ connect	TCON_IP_v4			False	False	False	False		Reference to the connection parameters	
Interfaceld	HW_ANY			False	False	False	False		HW-identifier of IE-interface submodule	
ID	CONN_OUC			False	False	False	False		connection reference / identifier	
Connection-Type	Byte			False	False	False	False		type of connection: 11=TCP/IP, 19=UDP (17=TCP/IP)	
ActiveEstablished	Bool			False	False	False	False		active/passive connection establishment	
▼ RemoteAddress	IP_V4			False	False	False	False		remote IP address (IPv4)	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment	
▼ ADDR	Array[1..4] of Byte			False	False	False	False		IPv4 address	
ADDR[1]	Byte			False	False	False	False		IPv4 address	
ADDR[2]	Byte			False	False	False	False		IPv4 address	
ADDR[3]	Byte			False	False	False	False		IPv4 address	
ADDR[4]	Byte			False	False	False	False		IPv4 address	
RemotePort	UInt			False	False	False	False		remote UDP/TCP port number	
LocalPort	UInt			False	False	False	False		local UDP/TCP port number	
▼ instMbClient	MB_CLIENT			False	False	False	False		Instance data of MB_CLIENT	
▼ Input										
REQ	Bool		Non-retain	False	False	False	False		Activates the requested transmission if TRUE	
DISCONNECT	Bool		Non-retain	False	False	False	False		Initiates a disconnect operation	
MB_MODE	USInt		Non-retain	False	False	False	False		Specifies the type of request: read, write or diagnostic	
MB_DATA_ADDR	UDInt		Non-retain	False	False	False	False		Specifies the starting address of the data to be accessed in the Modbus client	
MB_DATA_LEN	UInt		Non-retain	False	False	False	False		Specifies the number of bits or words to be accessed	
▼ Output										
DONE	Bool		Non-retain	False	False	False	False		Instruction finished without error	
BUSY	Bool		Non-retain	False	False	False	False		Modbus transaction in progress	
ERROR	Bool		Non-retain	False	False	False	False		Instruction finished with error	
STATUS	Word		Non-retain	False	False	False	False		Detailed error information	
▼ InOut										
MB_DATA_PTR	Variant			False	False	False	False		Reference to the local source or destination address	
CONNECT	Variant			False	False	False	False		Reference to the connection parameters	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Access-ible from HMI/OP C UA	Wri-ta-ble from engi-neer-ing HM I/O PC UA	Visible in HMI	Set-point	Super- vision	Comment	
▼ Static										
▼ TCON	TCON			False	False	False	True		Local instance of the instruction TCON	
▼ Input										
REQ	Bool		Non-retain	False	False	False	False		Function to be executed on rising edge	
ID	CONN_OU C		Non-retain	False	False	False	False		Connection identifier	
▼ Output										
DONE	Bool		Non-retain	False	False	False	False		New data received	
BUSY	Bool		Non-retain	False	False	False	False		Function busy	
ERROR	Bool		Non-retain	False	False	False	False		Error detected	
STA-TUS	Word		Non-retain	False	False	False	False		Function result/ error message	
▼ InOut										
CON- NECT	Variant			False	False	False	False		Connection de- scription	
Static										
▼ TDISCON	TDISCON			False	False	False	True		Local instance of the instruction TDISCON	
▼ Input										
REQ	Bool		Non-retain	False	False	False	False		Function to be executed on rising edge	
ID	CONN_OU C		Non-retain	False	False	False	False		Connection identifier	
▼ Output										
DONE	Bool		Non-retain	False	False	False	False		Function per- formed	
BUSY	Bool		Non-retain	False	False	False	False		Function busy	
ERROR	Bool		Non-retain	False	False	False	False		Error detected	
STA-TUS	Word		Non-retain	False	False	False	False		Function result/ error message	
InOut										
Static										
▼ TSEND	TSEND			False	False	False	True		Local instance of the instruction TSEND	
▼ Input										
REQ	Bool		Non-retain	False	False	False	False		Function to be executed on rising edge	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI	Set-point	Supervision	Comment	
ID	CONN_OUTC		Non-retain	False	False	False	False		Connection identifier	
LEN	UDInt		Non-retain	False	False	False	False		Data length to send	
▼ Output										
DONE	Bool		Non-retain	False	False	False	False		Send performed	
BUSY	Bool		Non-retain	False	False	False	False		Function busy	
ERROR	Bool		Non-retain	False	False	False	False		Error detected	
STATUS	Word		Non-retain	False	False	False	False		Function result/error message	
▼ InOut										
DATA	VARIANT			False	False	False	False		Pointer on data area to send	
ADDR	VARIANT			False	False	False	False		Pointer on address of receiver	
Static										
▼ RECEIVE	TRCV			False	False	False	True		Local instance of the instruction TRCV	
▼ Input										
EN_R	Bool		Non-retain	False	False	False	False		EN_R=1: function enabled	
ID	CONN_OUTC		Non-retain	False	False	False	False		Connection identifier	
LEN	UDInt		Non-retain	False	False	False	False		Data length to receive	
ADHOC	Bool		Non-retain	False	False	False	False		Request adhoc mode	
▼ Output										
NDR	Bool		Non-retain	False	False	False	False		New data received	
BUSY	Bool		Non-retain	False	False	False	False		Function busy	
ERROR	Bool		Non-retain	False	False	False	False		Error detected	
STATUS	Word		Non-retain	False	False	False	False		Function result/error message	
RCVD_LEN	UDInt		Non-retain	False	False	False	False		Length of received data	
▼ InOut										
DATA	VARIANT			False	False	False	False		Buffer for received data	
ADDR	VARIANT			False	False	False	False		Address of sender	
Static										

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment	
▼ TRESET	T_RESET			False	False	False	True		Local instance of the instruction T_RESET	
▼ Input										
REQ	Bool		Non-retain	False	False	False	False		Function to be executed on rising edge	
ID	CONN_OUT		Non-retain	False	False	False	False		Connection identifier	
▼ Output										
DONE	Bool		Non-retain	False	False	False	False		Function performed	
BUSY	Bool		Non-retain	False	False	False	False		Function busy	
ERROR	Bool		Non-retain	False	False	False	False		Error detected	
STATUS	Word		Non-retain	False	False	False	False		Function result/error message	
InOut										
Static										
▼ TDIAG	T_DIAG			False	False	False	True		Local instance of the instruction T_DIAG	
▼ Input										
REQ	Bool		Non-retain	False	False	False	False		Function to be executed on rising edge	
ID	CONN_OUT		Non-retain	False	False	False	False		Connection identifier	
▼ Output										
DONE	Bool		Non-retain	False	False	False	False		Function completed	
BUSY	Bool		Non-retain	False	False	False	False		Function busy	
ERROR	Bool		Non-retain	False	False	False	False		Error detected	
STATUS	Word		Non-retain	False	False	False	False		Function result/error message	
▼ InOut										
RESULT	Variant			False	False	False	False		Diagnostics information	
Static										
▼ TDIAG_Status	TDiag_Status		Non-retain	False	False	False	True		TDiag_Status structure	
InterfaceId	HW_ANY		Non-retain	False	False	False	False		HW-identifier of IE-interface submodule	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Set-point	Supervision	Comment	
ID	CONN_OUTC		Non-retain	False	False	False	False		connection reference / identifier of monitored connection	
ConnectionType	Byte		Non-retain	False	False	False	False		type of monitored connection	
ActiveEstablished	Bool		Non-retain	False	False	False	False		active/passive connection establishment	
State	Byte		Non-retain	False	False	False	False		state of monitored connection	
Kind	Byte		Non-retain	False	False	False	False		kind of monitored connection	
SentBytes	UDInt		Non-retain	False	False	False	False		bytes sent via monitored connection	
ReceivedBytes	UDInt		Non-retain	False	False	False	False		bytes received on monitored connection	
Blocked_Proc_Timeout	Real		Non-retain	False	False	False	False		Time to wait upon a blocked Modbus Client instance before setting inactive	
Rcv_Timeout	Real		Non-retain	False	False	False	False		Amount of time that the client waits for the server to respond to a request	
MB_Unit_ID	Byte		Non-retain	False	False	False	False		The Modbus slave address	
MB_Transaction_ID	Word		Non-retain	False	False	False	False		Number of the current transaction	
MB_State	Word		Non-retain	False	False	False	False		Internal state of the Modbus client operation	
SAVED_MB_DATA_ADDR	UDInt		Non-retain	False	False	False	False		For internal use only	
SAVED_DATA_LEN	UInt		Non-retain	False	False	False	False		For internal use only	
SAVED_MB_MODE	USInt		Non-retain	False	False	False	False		For internal use only	
Connection_ID	Word		Non-retain	False	False	False	False		For internal use only	
Retries	Word		Non-retain	False	False	False	False		Number of retries that the client will attempt before returning a no response error	
▼ TEMP_DATA	Array[1..263] of Byte		Set in IDB	False	False	False	False		For internal use only	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[1]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[2]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[3]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[4]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[5]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[6]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[7]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[8]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[9]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[10]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[11]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[12]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[13]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[14]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[15]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[16]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[17]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[18]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[19]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[20]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[21]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[22]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[23]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[24]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[25]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[26]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[27]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[28]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[29]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[30]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[31]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[32]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[33]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[34]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[35]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[36]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[37]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[38]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[39]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[40]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[41]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[42]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[43]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[44]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[45]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[46]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[47]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[48]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[49]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[50]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[51]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[52]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[53]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[54]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[55]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[56]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[57]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[58]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[59]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[60]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[61]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[62]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[63]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[64]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[65]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[66]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[67]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[68]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[69]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[70]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[71]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[72]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[73]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[74]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[75]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[76]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[77]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[78]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[79]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[80]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[81]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[82]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[83]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[84]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[85]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[86]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[87]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[88]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[89]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[90]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[91]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[92]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[93]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[94]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[95]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[96]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[97]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[98]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[99]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[100]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[101]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[102]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[103]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[104]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[105]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[106]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[107]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[108]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[109]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[110]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[111]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[112]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[113]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[114]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[115]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[116]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[117]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[118]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[119]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[120]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[121]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[122]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[123]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[124]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[125]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[126]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[127]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[128]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[129]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[130]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[131]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[132]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[133]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[134]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[135]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[136]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[137]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[138]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[139]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[140]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[141]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[142]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[143]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[144]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[145]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[146]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[147]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[148]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[149]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[150]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[151]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[152]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[153]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[154]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[155]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[156]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[157]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[158]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[159]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[160]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[161]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[162]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[163]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[164]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[165]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[166]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[167]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[168]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[169]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[170]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[171]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[172]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[173]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[174]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[175]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[176]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[177]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[178]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[179]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[180]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[181]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[182]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[183]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[184]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[185]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[186]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[187]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[188]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[189]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[190]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[191]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[192]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[193]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[194]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[195]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[196]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[197]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[198]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[199]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[200]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Access-ible from HMI/OP C UA	Wri-ta-ble from HM I/O PC UA	Visible in HMI engineering	Set-point	Super- vision	Comment
TEMP_DATA[201]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[202]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[203]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[204]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[205]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[206]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[207]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[208]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[209]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[210]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[211]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[212]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[213]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[214]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[215]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[216]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[217]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[218]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[219]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[220]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[221]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[222]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[223]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[224]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[225]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[226]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[227]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[228]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[229]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[230]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[231]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[232]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[233]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[234]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[235]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[236]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[237]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[238]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[239]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[240]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[241]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[242]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[243]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[244]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[245]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[246]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[247]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[248]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[249]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[250]	Byte		Set in IDB	False	False	False	False		For internal use only

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPCUA	Writable from HMI/OPCUA	Visible in HMI engineering	Set-point	Supervision	Comment
TEMP_DATA[251]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[252]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[253]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[254]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[255]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[256]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[257]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[258]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[259]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[260]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[261]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[262]	Byte		Set in IDB	False	False	False	False		For internal use only
TEMP_DATA[263]	Byte		Set in IDB	False	False	False	False		For internal use only
▼ rvc_timer	TON_TIME		Non-retain	False	False	False	True		For internal use only
PT	Time		Non-retain	False	False	False	False		
ET	Time		Non-retain	False	False	False	False		
IN	Bool		Non-retain	False	False	False	False		
Q	Bool		Non-retain	False	False	False	False		
▼ wd_timer	TON_TIME		Non-retain	False	False	False	True		For internal use only
PT	Time		Non-retain	False	False	False	False		
ET	Time		Non-retain	False	False	False	False		
IN	Bool		Non-retain	False	False	False	False		
Q	Bool		Non-retain	False	False	False	False		
Connected	Bool		Non-retain	False	False	False	False		Connection state
Active	Bool		Non-retain	False	False	False	False		This call is the active instance
mbDataReal	Real	0.0	Set in IDB	True	True	True	False		Transfer data in Real

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Access-ible from HMI/OPC UA	Wri-ta-ble from HM I/O PC UA	Visible in HMI	Set-point	Super- vision	Comment	
mbDataUdint	UDInt		Set in IDB	False	False	False	False		Transfer data in UDInt	
▼ Static										
statCommErrMsg-Delay	Time	T#5m	Non-retain	False	False	False	False		Communication error message delay time	
statSampleMax	UInt	3	Non-retain	False	False	False	False		Number of samples for P averaging	
statStep	UInt	0	Non-retain	True	True	True	False			
statSampleRate	UInt	0	Non-retain	True	True	True	False			
statPsum	Real	0.0	Non-retain	True	True	True	False			
statPaverage	Real	0.0	Non-retain	True	True	True	False			
▼ Temp										
tempTimeout-Comm	Bool									
tempBool	Bool									
tempUdint	UDInt									
tempReal	Real									
▼ Constant										
ADDR_TOTAL_ACTIVE_POWER	UDInt	40066								
ADDR_PTO-TAL_MIN	UDInt	40198								
ADDR_PTO-TAL_MAX	UDInt	40140								
ADDR_CUR-RENT_L1	UDInt	40014								
ADDR_CUR-RENT_L2	UDInt	40016								
ADDR_CUR-RENT_L3	UDInt	40018								
ADDR_V_L1-L2	UDInt	40008								
ADDR_V_L2-L3	UDInt	40010								
ADDR_V_L3-L1	UDInt	40012								
ADDR_vTHD_L1	UDInt	40262								
ADDR_vTHD_L2	UDInt	40264								
ADDR_vTHD_L3	UDInt	40266								
ADDR_cTHD_L1	UDInt	40268								
ADDR_cTHD_L2	UDInt	40270								
ADDR_cTHD_L3	UDInt	40272								
ADDR_COS_L1	UDInt	40244								
ADDR_COS_L2	UDInt	40246								
ADDR_COS_L3	UDInt	40248								

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Access-ible from HMI/OPCUA	Wri-ta-ble from HM I/O PCUA	Visible in HMI point	Set-point	Super- vision	Comment
CON- VERT_WS_KWH	Real	3600000.0							
READ	USInt	0							
WRITE	USInt	1							
<pre> 0001 REGION step 0: Initial step after restart 0002 IF (#statStep = 0) THEN 0003 IF (#nextPacPulse = TRUE) THEN // next PAC to communicate 0004 #mbClientInput.disconnect := FALSE; // start connection 0005 #statStep := 10; 0006 END_IF; 0007 GOTO process_comm_error; // check communication error 0008 END_IF; 0009 END_REGION 0010 0011 REGION step 10: pTot. Waiting for CONNECTED and NOT(BUSY) 0012 IF (#statStep = 10) THEN 0013 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0014 #mbClientInput.mbDataAddr := #ADDR_TOTAL_ACTIVE_POWER; // TOTAL_POWER 0015 #mbClientInput.mode := #READ; // read 0016 #mbClientInput.req := TRUE; // start MB_CLIENT 0017 #statStep := 11; 0018 END_IF; 0019 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0020 GOTO process_comm_error; // check communication error 0021 END_IF; 0022 END_REGION 0023 0024 REGION step 11: Waiting for reception pTot DONE 0025 IF (#statStep = 11) THEN 0026 0027 IF (#instMbClient.DONE = TRUE) THEN 0028 0029 #tempReal := #mbDataReal / 1000.0; 0030 0031 #device[#pacIndex].pTotal := #tempReal; 0032 #statStep := 12; 0033 0034 END_IF; 0035 0036 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0037 GOTO process_comm_error; // check communication error 0038 0039 END_IF; 0040 END_REGION 0041 0042 REGION step 12: pMax Waiting for CONNECTED and NOT(BUSY) 0043 IF (#statStep = 12) THEN 0044 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0045 #mbClientInput.mbDataAddr := #ADDR_PTOTAL_MAX; // TOTAL_POWER 0046 #mbClientInput.mode := #READ; // read </pre>									

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal		
<pre> 0047 #mbClientInput.req := TRUE; // start MB_CLIENT 0048 #statStep := 13; 0049 END_IF; 0050 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0051 GOTO process_comm_error; // check communication error 0052 END_IF; 0053 END_REGION 0054 0055 REGION step 13: Waiting for reception pMax DONE 0056 IF (#statStep = 13) THEN 0057 0058 IF (#instMbClient.DONE = TRUE) THEN 0059 0060 #tempReal := #mbDataReal / 1000.0; 0061 0062 #device[#pacIndex].pTotalMax := #tempReal; 0063 #statStep := 14; 0064 0065 END_IF; 0066 0067 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0068 GOTO process_comm_error; // check communication error 0069 0070 END_IF; 0071 END_REGION 0072 0073 0074 REGION step 14: pMin Waiting for CONNECTED and NOT(BUSY) 0075 IF (#statStep = 14) THEN 0076 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0077 #mbClientInput.mbDataAddr := #ADDR_PTOTAL_MIN; // TOTAL_POWER 0078 #mbClientInput.mode := #READ; // read 0079 #mbClientInput.req := TRUE; // start MB_CLIENT 0080 #statStep := 15; 0081 END_IF; 0082 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0083 GOTO process_comm_error; // check communication error 0084 END_IF; 0085 END_REGION 0086 0087 REGION step 15: Waiting for reception pMin DONE 0088 IF (#statStep = 15) THEN 0089 0090 IF (#instMbClient.DONE = TRUE) THEN 0091 0092 #tempReal := #mbDataReal / 1000.0; 0093 0094 #device[#pacIndex].pTotalMin := #tempReal; 0095 #statStep := 16; 0096 0097 END_IF; 0098 0099 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0100 GOTO process_comm_error; // check communication error 0101 0102 END_IF; 0103 END_REGION 0104 0105 </pre>		

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal	
<pre> 0106 REGION step 16: Current-L1 Waiting for CONNECTED and NOT(BUSY) 0107 IF (#statStep = 16) THEN 0108 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0109 #mbClientInput.mbDataAddr := #ADDR_CURRENT_L1; // TOTAL_POWER 0110 #mbClientInput.mode := #READ; // read 0111 #mbClientInput.req := TRUE; // start MB_CLIENT 0112 #statStep := 17; 0113 END_IF; 0114 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0115 GOTO process_comm_error; // check communication error 0116 END_IF; 0117 END_REGION 0118 0119 REGION step 17: Waiting for reception Current-L1 DONE 0120 IF (#statStep = 17) THEN 0121 0122 IF (#instMbClient.DONE = TRUE) THEN 0123 0124 #device[#pacIndex].CorrienteL1 := #mbDataReal; 0125 #statStep := 18; 0126 0127 END_IF; 0128 0129 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0130 GOTO process_comm_error; // check communication error 0131 0132 END_IF; 0133 END_REGION 0134 0135 0136 REGION step 18: Current-L2 Waiting for CONNECTED and NOT(BUSY) 0137 IF (#statStep = 18) THEN 0138 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0139 #mbClientInput.mbDataAddr := #ADDR_CURRENT_L2; // TOTAL_POWER 0140 #mbClientInput.mode := #READ; // read 0141 #mbClientInput.req := TRUE; // start MB_CLIENT 0142 #statStep := 19; 0143 END_IF; 0144 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0145 GOTO process_comm_error; // check communication error 0146 END_IF; 0147 END_REGION 0148 0149 REGION step 19: Waiting for reception Current-L2 DONE 0150 IF (#statStep = 19) THEN 0151 0152 IF (#instMbClient.DONE = TRUE) THEN 0153 0154 #device[#pacIndex].CorrienteL2 := #mbDataReal; 0155 #statStep := 20; 0156 0157 END_IF; 0158 0159 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0160 GOTO process_comm_error; // check communication error 0161 0162 END_IF; 0163 END_REGION 0164 </pre>	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal		
<pre> 0165 0166 REGION step 20: Current-L3 Waiting for CONNECTED and NOT(BUSY) 0167 IF (#statStep = 20) THEN 0168 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0169 #mbClientInput.mbDataAddr := #ADDR_CURRENT_L3; // TOTAL_POWER 0170 #mbClientInput.mode := #READ; // read 0171 #mbClientInput.req := TRUE; // start MB_CLIENT 0172 #statStep := 21; 0173 END_IF; 0174 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0175 GOTO process_comm_error; // check communication error 0176 END_IF; 0177 END_REGION 0178 0179 REGION step 21: Waiting for reception Current-L3 DONE 0180 IF (#statStep = 21) THEN 0181 IF (#instMbClient.DONE = TRUE) THEN 0182 #device[#pacIndex].CorrienteL3 := #mbDataReal; 0183 #statStep := 22; 0184 END_IF; 0185 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0186 GOTO process_comm_error; // check communication error 0187 END_IF; 0188 END_REGION 0189 0190 REGION step 22: V_L1-L2 Waiting FOR CONNECTED AND NOT(BUSY) 0191 IF (#statStep = 22) THEN 0192 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0193 #mbClientInput.mbDataAddr := #"ADDR_V_L1-L2"; // TOTAL_POWER 0194 #mbClientInput.mode := #READ; // read 0195 #mbClientInput.req := TRUE; // start MB_CLIENT 0196 #statStep := 23; 0197 END_IF; 0198 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0199 GOTO process_comm_error; // check communication error 0200 END_IF; 0201 END_REGION 0202 0203 REGION step 23: Waiting for reception V_L1-L2 DONE 0204 IF (#statStep = 23) THEN 0205 IF (#instMbClient.DONE = TRUE) THEN 0206 #device[#pacIndex]."VoltageL1-L2" := #mbDataReal/1000; 0207 #statStep := 24; 0208 END_IF; 0209 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0210 GOTO process_comm_error; // check communication error 0211 END_IF; 0212 END_REGION 0213 0214 0215 0216 0217 0218 0219 0220 0221 0222 0223 </pre>		

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal	
<pre> 0224 REGION step 24: V_L2-L3 Waiting FOR CONNECTED AND NOT(BUSY) 0225 IF (#statStep = 24) THEN 0226 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0227 #mbClientInput.mbDataAddr := #"ADDR_V_L2-L3"; // TOTAL_POWER 0228 #mbClientInput.mode := #READ; // read 0229 #mbClientInput.req := TRUE; // start MB_CLIENT 0230 #statStep := 25; 0231 END_IF; 0232 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0233 GOTO process_comm_error; // check communication error 0234 END_IF; 0235 END_REGION 0236 0237 REGION step 25: Waiting for reception V_L2-L3 DONE 0238 IF (#statStep = 25) THEN 0239 0240 IF (#instMbClient.DONE = TRUE) THEN 0241 0242 #device[#pacIndex]."VoltageL2-L3" := #mbDataReal/1000; 0243 #statStep := 26; 0244 0245 END_IF; 0246 0247 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0248 GOTO process_comm_error; // check communication error 0249 0250 END_IF; 0251 END_REGION 0252 0253 REGION step 26: V_L3-L1 Waiting FOR CONNECTED AND NOT(BUSY) 0254 IF (#statStep = 26) THEN 0255 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0256 #mbClientInput.mbDataAddr := #"ADDR_V_L3-L1"; // TOTAL_POWER 0257 #mbClientInput.mode := #READ; // read 0258 #mbClientInput.req := TRUE; // start MB_CLIENT 0259 #statStep := 27; 0260 END_IF; 0261 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0262 GOTO process_comm_error; // check communication error 0263 END_IF; 0264 END_REGION 0265 0266 REGION step 27: Waiting for reception V_L3-L1 DONE 0267 IF (#statStep = 27) THEN 0268 0269 IF (#instMbClient.DONE = TRUE) THEN 0270 0271 #device[#pacIndex]."VoltageL3-L1" := #mbDataReal/1000; 0272 #statStep := 28; 0273 0274 END_IF; 0275 0276 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0277 GOTO process_comm_error; // check communication error 0278 0279 END_IF; 0280 END_REGION 0281 0282 REGION step 28: vTHD_L1 Waiting FOR CONNECTED AND NOT(BUSY) </pre>	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal		
<pre> 0283 IF (#statStep = 28) THEN 0284 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0285 #mbClientInput.mbDataAddr := #ADDR_vTHD_L1; // TOTAL_POWER 0286 #mbClientInput.mode := #READ; // read 0287 #mbClientInput.req := TRUE; // start MB_CLIENT 0288 #statStep := 29; 0289 END_IF; 0290 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0291 GOTO process_comm_error; // check communication error 0292 END_IF; 0293 END_REGION 0294 0295 REGION step 29: Waiting for reception vTHd_L1 DONE 0296 IF (#statStep = 29) THEN 0297 IF (#instMbClient.DONE = TRUE) THEN 0298 #device[#pacIndex].vTHD_L1 := #mbDataReal; 0300 #statStep := 30; 0301 END_IF; 0302 0303 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0304 GOTO process_comm_error; // check communication error 0305 END_IF; 0306 END_REGION 0307 0308 REGION step 30: vTHD-L2 Waiting FOR CONNECTED AND NOT(BUSY) 0309 IF (#statStep = 30) THEN 0310 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0311 #mbClientInput.mbDataAddr := #ADDR_vTHD_L2; // TOTAL_POWER 0312 #mbClientInput.mode := #READ; // read 0313 #mbClientInput.req := TRUE; // start MB_CLIENT 0314 #statStep := 31; 0315 END_IF; 0316 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0317 GOTO process_comm_error; // check communication error 0318 END_IF; 0319 END_REGION 0320 0321 REGION step 31: Waiting for reception vTHD-L2 DONE 0322 IF (#statStep = 31) THEN 0323 IF (#instMbClient.DONE = TRUE) THEN 0324 #device[#pacIndex].vTHD_L2 := #mbDataReal; 0325 #statStep := 32; 0326 END_IF; 0327 0328 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0329 GOTO process_comm_error; // check communication error 0330 END_IF; 0331 END_REGION 0332 0333 REGION step 32: vTHD_L3 Waiting FOR CONNECTED AND NOT(BUSY) 0334 IF (#statStep = 32) THEN </pre>		

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal	
<pre> 0342 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0343 #mbClientInput.mbDataAddr := #ADDR_vTHD_L3; // TOTAL_POWER 0344 #mbClientInput.mode := #READ; // read 0345 #mbClientInput.req := TRUE; // start MB_CLIENT 0346 #statStep := 33; 0347 END_IF; 0348 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0349 GOTO process_comm_error; // check communication error 0350 END_IF; 0351 END_REGION 0352 0353 REGION step 33: Waiting for reception vTHD_L3 DONE 0354 IF (#statStep = 33) THEN 0355 0356 IF (#instMbClient.DONE = TRUE) THEN 0357 0358 #device[#pacIndex].vTHD_L3 := #mbDataReal; 0359 #statStep := 34; 0360 0361 END_IF; 0362 0363 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0364 GOTO process_comm_error; // check communication error 0365 0366 END_IF; 0367 END_REGION 0368 0369 REGION step 34: cTHD_L1 Waiting FOR CONNECTED AND NOT(BUSY) 0370 IF (#statStep = 34) THEN 0371 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0372 #mbClientInput.mbDataAddr := #ADDR_cTHD_L1; // TOTAL_POWER 0373 #mbClientInput.mode := #READ; // read 0374 #mbClientInput.req := TRUE; // start MB_CLIENT 0375 #statStep := 35; 0376 END_IF; 0377 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0378 GOTO process_comm_error; // check communication error 0379 END_IF; 0380 END_REGION 0381 0382 REGION step 35: Waiting for reception cTHD_L1 DONE 0383 IF (#statStep = 35) THEN 0384 0385 IF (#instMbClient.DONE = TRUE) THEN 0386 0387 #device[#pacIndex].cTHD_L1 := #mbDataReal; 0388 #statStep := 36; 0389 0390 END_IF; 0391 0392 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0393 GOTO process_comm_error; // check communication error 0394 0395 END_IF; 0396 END_REGION 0397 0398 REGION step 36: cTHD_L2 Waiting FOR CONNECTED AND NOT(BUSY) 0399 IF (#statStep = 36) THEN 0400 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN </pre>	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal	
<pre> 0401 #mbClientInput.mbDataAddr := #ADDR_cTHD_L2; // TOTAL_POWER 0402 #mbClientInput.mode := #READ; // read 0403 #mbClientInput.req := TRUE; // start MB_CLIENT 0404 #statStep := 37; 0405 END_IF; 0406 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0407 GOTO process_comm_error; // check communication error 0408 END_IF; 0409 END_REGION 0410 0411 REGION step 37: Waiting for reception cTHD_L2 DONE 0412 IF (#statStep = 37) THEN 0413 0414 IF (#instMbClient.DONE = TRUE) THEN 0415 0416 #device[#pacIndex].cTHD_L2 := #mbDataReal; 0417 #statStep := 38; 0418 0419 END_IF; 0420 0421 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0422 GOTO process_comm_error; // check communication error 0423 0424 END_IF; 0425 END_REGION 0426 0427 REGION step 38: cTHD_L3 Waiting FOR CONNECTED AND NOT(BUSY) 0428 IF (#statStep = 38) THEN 0429 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0430 #mbClientInput.mbDataAddr := #ADDR_cTHD_L3; // TOTAL_POWER 0431 #mbClientInput.mode := #READ; // read 0432 #mbClientInput.req := TRUE; // start MB_CLIENT 0433 #statStep := 39; 0434 END_IF; 0435 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0436 GOTO process_comm_error; // check communication error 0437 END_IF; 0438 END_REGION 0439 0440 REGION step 39: Waiting for reception cTHD_L3 DONE 0441 IF (#statStep = 39) THEN 0442 0443 IF (#instMbClient.DONE = TRUE) THEN 0444 0445 #device[#pacIndex].cTHD_L3 := #mbDataReal; 0446 #statStep := 40; 0447 0448 END_IF; 0449 0450 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0451 GOTO process_comm_error; // check communication error 0452 0453 END_IF; 0454 END_REGION 0455 0456 REGION step 40: cos_L1 Waiting FOR CONNECTED AND NOT(BUSY) 0457 IF (#statStep = 40) THEN 0458 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0459 #mbClientInput.mbDataAddr := #ADDR_COS_L1; // TOTAL_POWER </pre>	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal	
<pre> 0460 #mbClientInput.mode := #READ; // read 0461 #mbClientInput.req := TRUE; // start MB_CLIENT 0462 #statStep := 41; 0463 END_IF; 0464 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0465 GOTO process_comm_error; // check communication error 0466 END_IF; 0467 END_REGION 0468 0469 REGION step 41: Waiting for reception cos_L1 DONE 0470 IF (#statStep = 41) THEN 0471 0472 IF (#instMbClient.DONE = TRUE) THEN 0473 0474 #device[#pacIndex].cosL1 := #mbDataReal; 0475 #statStep := 42; 0476 0477 END_IF; 0478 0479 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0480 GOTO process_comm_error; // check communication error 0481 0482 END_IF; 0483 END_REGION 0484 0485 REGION step 42: cos-L2 Waiting FOR CONNECTED AND NOT(BUSY) 0486 IF (#statStep = 42) THEN 0487 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0488 #mbClientInput.mbDataAddr := #ADDR_COS_L2; // TOTAL_POWER 0489 #mbClientInput.mode := #READ; // read 0490 #mbClientInput.req := TRUE; // start MB_CLIENT 0491 #statStep := 43; 0492 END_IF; 0493 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0494 GOTO process_comm_error; // check communication error 0495 END_IF; 0496 END_REGION 0497 0498 REGION step 43: Waiting for reception cos-L2 DONE 0499 IF (#statStep = 43) THEN 0500 0501 IF (#instMbClient.DONE = TRUE) THEN 0502 0503 #device[#pacIndex].cosL2 := #mbDataReal; 0504 #statStep := 44; 0505 0506 END_IF; 0507 0508 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0509 GOTO process_comm_error; // check communication error 0510 0511 END_IF; 0512 END_REGION 0513 0514 REGION step 44: cos_L3 Waiting FOR CONNECTED AND NOT(BUSY) 0515 IF (#statStep = 44) THEN 0516 IF (#instMbClient.Connected = TRUE) AND (#instMbClient.BUSY = FALSE) THEN 0517 #mbClientInput.mbDataAddr := #ADDR_COS_L3; // TOTAL_POWER 0518 #mbClientInput.mode := #READ; // read </pre>	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal	
<pre> 0519 #mbClientInput.req := TRUE; // start MB_CLIENT 0520 #statStep := 45; 0521 END_IF; 0522 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0523 GOTO process_comm_error; // check communication error 0524 END_IF; 0525 END_REGION 0526 0527 REGION step 45: Waiting for reception V_L1-L2 DONE 0528 IF (#statStep = 45) THEN 0529 0530 IF (#instMbClient.DONE = TRUE) THEN 0531 0532 #device[#pacIndex].cosL3 := #mbDataReal; 0533 #statStep := 70; 0534 0535 END_IF; 0536 0537 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0538 GOTO process_comm_error; // check communication error 0539 0540 END_IF; 0541 END_REGION 0542 0543 REGION step 70: Waiting for NOT(BUSY) 0544 IF (#statStep = 70) THEN 0545 0546 IF (#instMbClient.BUSY = FALSE) THEN // MB_CLIENT is stopped? 0547 #mbClientInput.disconnect := TRUE; // disconnection 0548 #statStep := 80; 0549 END_IF; 0550 0551 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? 0552 GOTO process_comm_error; // check communication error 0553 0554 END_IF; 0555 END_REGION 0556 0557 REGION step 80: Waiting for NOT(CONNECTED) and NOT(BUSY) 0558 IF (#statStep = 80) THEN 0559 0560 IF (#instMbClient.Connected = FALSE) AND (#instMbClient.BUSY = FALSE) THEN 0561 0562 // resetting first scan marker 0563 IF (#restart = TRUE) AND (#pacIndex = #numberOfPacs) THEN 0564 #restart := FALSE; 0565 END_IF; 0566 0567 #pacIndex := #pacIndex + 1; // increment PAC counter 0568 0569 IF (#pacIndex > #numberOfPacs) THEN 0570 #pacIndex := 1; // reset PAC counter 0571 END_IF; 0572 0573 #statStep := 0; // start sequencer for next PAC 0574 0575 END_IF; 0576 0577 #tempTimeoutComm := #nextPacPulse; // does puls for next PAC occurred? </pre>	

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal																																																																																										
<pre> 0578 GOTO process_comm_error; // check communication error 0579 0580 END_IF; 0581 END_REGION 0582 0583 REGION React on communication timeout 0584 process_comm_error: 0585 IF (#tempTimeoutComm = TRUE) THEN // does puls for next PAC occurred? 0586 0587 // set communication error (X0) 0588 // #device[#pacIndex].bitAlarms.%X0 := TRUE; 0589 FOR_counter_ := _start_count_ TO _end_count_ DO 0590 // Statement section FOR 0591 ; 0592 END_FOR; 0593 0594 #mbClientInput.req := FALSE; // stopping MB_CLIENT 0595 #mbClientInput.disconnect := TRUE; // disconnection 0596 0597 IF (#device[#pacIndex].commBreakdown.started = FALSE) THEN 0598 #device[#pacIndex].commBreakdown.started := TRUE; // set marker 0599 // calculate time when "breakdown" message should be displayed 0600 #device[#pacIndex].commBreakdown.errMsgTime := T_ADD(IN1 := #hmi.sysTime, 0601 IN2 := #statCommErrMsgDelay); 0602 END_IF; 0603 0604 #statStep := 80; // check if MB_CLIENT is stopped and disconnected 0605 0606 END_IF; 0607 END_REGION 0608 0609 </pre>																																																																																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Symbol</th> <th style="width: 20%;">Address</th> <th style="width: 20%;">Type</th> <th style="width: 40%;">Comment</th> </tr> </thead> <tbody> <tr><td>#"ADDR_V_L1-L2"</td><td>40008</td><td>UDInt</td><td></td></tr> <tr><td>#"ADDR_V_L2-L3"</td><td>40010</td><td>UDInt</td><td></td></tr> <tr><td>#"ADDR_V_L3-L1"</td><td>40012</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_COS_L1</td><td>40244</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_COS_L2</td><td>40246</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_COS_L3</td><td>40248</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_cTHD_L1</td><td>40268</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_cTHD_L2</td><td>40270</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_cTHD_L3</td><td>40272</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_CURRENT_L1</td><td>40014</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_CURRENT_L2</td><td>40016</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_CURRENT_L3</td><td>40018</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_PTOTAL_MAX</td><td>40140</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_PTOTAL_MIN</td><td>40198</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_TOTAL_AC-TIVE_POWER</td><td>40066</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_vTHD_L1</td><td>40262</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_vTHD_L2</td><td>40264</td><td>UDInt</td><td></td></tr> <tr><td>#ADDR_vTHD_L3</td><td>40266</td><td>UDInt</td><td></td></tr> <tr><td>#device[*].VoltageL1-L2"</td><td></td><td>Real</td><td>Array of SENTRON PAC data</td></tr> <tr><td>#device[*].VoltageL2-L3"</td><td></td><td>Real</td><td>Array of SENTRON PAC data</td></tr> <tr><td>#device[*].VoltageL3-L1"</td><td></td><td>Real</td><td>Array of SENTRON PAC data</td></tr> </tbody> </table>			Symbol	Address	Type	Comment	#"ADDR_V_L1-L2"	40008	UDInt		#"ADDR_V_L2-L3"	40010	UDInt		#"ADDR_V_L3-L1"	40012	UDInt		#ADDR_COS_L1	40244	UDInt		#ADDR_COS_L2	40246	UDInt		#ADDR_COS_L3	40248	UDInt		#ADDR_cTHD_L1	40268	UDInt		#ADDR_cTHD_L2	40270	UDInt		#ADDR_cTHD_L3	40272	UDInt		#ADDR_CURRENT_L1	40014	UDInt		#ADDR_CURRENT_L2	40016	UDInt		#ADDR_CURRENT_L3	40018	UDInt		#ADDR_PTOTAL_MAX	40140	UDInt		#ADDR_PTOTAL_MIN	40198	UDInt		#ADDR_TOTAL_AC-TIVE_POWER	40066	UDInt		#ADDR_vTHD_L1	40262	UDInt		#ADDR_vTHD_L2	40264	UDInt		#ADDR_vTHD_L3	40266	UDInt		#device[*].VoltageL1-L2"		Real	Array of SENTRON PAC data	#device[*].VoltageL2-L3"		Real	Array of SENTRON PAC data	#device[*].VoltageL3-L1"		Real	Array of SENTRON PAC data
Symbol	Address	Type	Comment																																																																																							
#"ADDR_V_L1-L2"	40008	UDInt																																																																																								
#"ADDR_V_L2-L3"	40010	UDInt																																																																																								
#"ADDR_V_L3-L1"	40012	UDInt																																																																																								
#ADDR_COS_L1	40244	UDInt																																																																																								
#ADDR_COS_L2	40246	UDInt																																																																																								
#ADDR_COS_L3	40248	UDInt																																																																																								
#ADDR_cTHD_L1	40268	UDInt																																																																																								
#ADDR_cTHD_L2	40270	UDInt																																																																																								
#ADDR_cTHD_L3	40272	UDInt																																																																																								
#ADDR_CURRENT_L1	40014	UDInt																																																																																								
#ADDR_CURRENT_L2	40016	UDInt																																																																																								
#ADDR_CURRENT_L3	40018	UDInt																																																																																								
#ADDR_PTOTAL_MAX	40140	UDInt																																																																																								
#ADDR_PTOTAL_MIN	40198	UDInt																																																																																								
#ADDR_TOTAL_AC-TIVE_POWER	40066	UDInt																																																																																								
#ADDR_vTHD_L1	40262	UDInt																																																																																								
#ADDR_vTHD_L2	40264	UDInt																																																																																								
#ADDR_vTHD_L3	40266	UDInt																																																																																								
#device[*].VoltageL1-L2"		Real	Array of SENTRON PAC data																																																																																							
#device[*].VoltageL2-L3"		Real	Array of SENTRON PAC data																																																																																							
#device[*].VoltageL3-L1"		Real	Array of SENTRON PAC data																																																																																							

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal			
Symbol	Address	Type	Comment
#device[*].CorrienteL1		Real	Array of SENTRON PAC data
#device[*].CorrienteL2		Real	Array of SENTRON PAC data
#device[*].CorrienteL3		Real	Array of SENTRON PAC data
#device[*].cosL1		Real	Array of SENTRON PAC data
#device[*].cosL2		Real	Array of SENTRON PAC data
#device[*].cosL3		Real	Array of SENTRON PAC data
#device[*].cTHD_L1		Real	Array of SENTRON PAC data
#device[*].cTHD_L2		Real	Array of SENTRON PAC data
#device[*].cTHD_L3		Real	Array of SENTRON PAC data
#device[*].pTotal		Real	Array of SENTRON PAC data
#device[*].pTotalMax		Real	Array of SENTRON PAC data
#device[*].pTotalMin		Real	Array of SENTRON PAC data
#device[*].vTHD_L1		Real	Array of SENTRON PAC data
#device[*].vTHD_L2		Real	Array of SENTRON PAC data
#device[*].vTHD_L3		Real	Array of SENTRON PAC data
#instMbClient.BUSY		Bool	Modbus transaction in progress
#instMbClient.Connected		Bool	Connection state
#instMbClient.DONE		Bool	Instruction finished without error
#mbClientInput.disconnect		Bool	Initiates a disconnect operation
#mbClientInput.mbDataAddr		UDInt	Specifies the starting address of the data to be accessed in the Modbus client
#mbClientInput.mode		USInt	Specifies the type of request: read, write or diagnostic
#mbClientInput.req		Bool	Activates the requested transmission if TRUE
#mbDataReal		Real	Transfer data in Real
#nextPacPulse		Bool	Communication trigger to the next PAC
#numberOfPacs		DInt	Number of PACs to communicate
#pacIndex		Int	Actual PAC number to communicate
#READ	0	USInt	
#restart		Bool	Marker for the first communication of all PACs
#statCommErrMsgDelay		Time	Communication error message delay time
#statStep		UInt	
#tempReal		Real	
#tempTimeoutComm		Bool	

8.2. Código SetCom

Totally Integrated Automation Portal									
SetCom [FB1]									
SetCom Properties									
General									
Name	SetCom	Number	1	Type	FB				
Language	SCL	Numbering	Automatic						
Information									
Title		Author		Comment					
Family		Version	0.1	User-defined ID					
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in engineering	Set-point	Supervision	Comment
▼ Input									
initialCall	Bool	false	Non-retain	True	True	True	False		Initial call of the called OB
Output									
▼ InOut									
▼ device	Array[*] of "typeDevice"			False	False	False	False		Array of SENTRON PAC data
▼ device[*]	"typeDevice"			False	False	False	False		Array of SENTRON PAC data
ipOctet4	USInt			False	False	False	False		Forth octet of the ip address
pTotal	Real			False	False	False	False		Average value of the total active power (in W)
pTotalMax	Real			False	False	False	False		
pTotalMin	Real			False	False	False	False		
CorrienteL1	Real			False	False	False	False		
CorrienteL2	Real			False	False	False	False		
CorrienteL3	Real			False	False	False	False		
VoltageL1-L2	Real			False	False	False	False		
VoltageL2-L3	Real			False	False	False	False		
VoltageL3-L1	Real			False	False	False	False		
vTHD_L1	Real			False	False	False	False		
vTHD_L2	Real			False	False	False	False		
vTHD_L3	Real			False	False	False	False		

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Access-ible from HMI/OPC UA	Wri-ta-ble from engineering HM I/O PC UA	Visible in HMI	Set-point	Super- vision	Comment
cTHD_L1	Real			False	False	False	False		
cTHD_L2	Real			False	False	False	False		
cTHD_L3	Real			False	False	False	False		
cosL1	Real			False	False	False	False		
cosL2	Real			False	False	False	False		
cosL3	Real			False	False	False	False		
Static									
Temp									
▼ Constant									
MB_DATA_LEN	UInt	2							Specifies the number of bits or words to be accessed
<pre> 0001 REGION 1 Memorize restart 0002 IF (#initialCall = TRUE) THEN 0003 #statRestart := TRUE; // set marker 0004 #statNumberOfPacs := UPPER_BOUND(ARR := #device, DIM := 1); //count PACs 0005 END_IF; 0006 END_REGION 0007 0008 REGION 2 Read PLC system time 0009 #statRetVal := RD_SYS_T(#hmi.sysTime); 0010 END_REGION 0011 0012 REGION 3 PAC processing interval 0013 #instScanIntervalTimer(IN := NOT (#statNextPacPulse), 0014 PT := #statScanInterval, 0015 Q => #statNextPacPulse); 0016 END_REGION 0017 0018 REGION 4 IP address 0019 // copy 4th octet to the connection data 0020 #statMbClientInput.connect.RemoteAddress.ADDR[4] := #device[#statPacIndex].ipOctet4; 0021 END_REGION 0022 0023 REGION 5 MB_CLIENT call 0024 #instMbClient(REQ := #statMbClientInput.req, 0025 DISCONNECT := #statMbClientInput.disconnect, 0026 MB_MODE := #statMbClientInput.mode, 0027 MB_DATA_ADDR := #statMbClientInput.mbDataAddr, 0028 MB_DATA_LEN := #MB_DATA_LEN, 0029 MB_DATA_PTR := #statMbDataReal, 0030 CONNECT := #statMbClientInput.connect); </pre>									

Sistema SCADA para el control de tres centros de transformación

Totally Integrated Automation Portal			
<pre> 0031 END_REGION 0032 0033 REGION 6 Sequencer 0034 #inst.Sequencer(numberOfPacs := #statNumberOfPacs, 0035 nextPacPulse := #statNextPacPulse, 0036 device := #device, 0037 hmi := #hmi, 0038 pacIndex := #statPacIndex, 0039 restart := #statRestart, 0040 instMbClient:=#instMbClient, 0041 mbClientInput:=#statMbClientInput, 0042 mbDataReal:=#statMbDataReal); 0043 END_REGION 0044 </pre>			
Symbol	Address	Type	Comment
#device		Array	Array of SENTRON PAC data
#device[*].ipOctet4		USInt	Array of SENTRON PAC data
#initialCall		Bool	Initial call of the called OB
#MB_DATA_LEN	2	UInt	Specifies the number of bits or words to be accessed

8.3. Registros Modbus

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
1	2	Tensión L1-N	Float	V	-	R
3	2	Tensión L2-N	Float	V	-	R
5	2	Tensión L3-N	Float	V	-	R
7	2	Tensión L1-L2	Float	V	-	R
9	2	Tensión L2-L3	Float	V	-	R
11	2	Tensión L3-L1	Float	V	-	R
13	2	Corriente L1	Float	A	-	R
15	2	Corriente L2	Float	A	-	R
17	2	Corriente L3	Float	A	-	R
19	2	Potencia aparente L1	Float	VA	-	R
21	2	Potencia aparente L2	Float	VA	-	R
23	2	Potencia aparente L3	Float	VA	-	R
25	2	Potencia activa L1	Float	W	-	R
27	2	Potencia activa L2	Float	W	-	R
29	2	Potencia activa L3	Float	W	-	R
31	2	Potencia reactiva L1 (Qn)	Float	var	-	R
33	2	Potencia reactiva L2 (Qn)	Float	var	-	R
35	2	Potencia reactiva L3 (Qn)	Float	var	-	R
37	2	Factor de potencia L1	Float	-	0 ... 1	R
39	2	Factor de potencia L2	Float	-	0 ... 1	R
41	2	Factor de potencia L3	Float	-	0 ... 1	R
43	2	THD en tensión L1-L2	Float	%	0 ... 100	R
45	2	THD en tensión L2-L3	Float	%	0 ... 100	R
47	2	THD en tensión L3-L1	Float	%	0 ... 100	R
49	2	Reserva				R

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
51	2	Reserva				R
53	2	Reserva				R
55	2	Frecuencia de red	Float	Hz	45 ... 65	R
57	2	Media en 3 fases de la tensión L-N	Float	V	-	R
59	2	Media en 3 fases de la tensión L-L	Float	V	-	R
61	2	Media en 3 fases de la corriente L-L	Float	A	-	R
63	2	Potencia aparente total	Float	VA	-	R
65	2	Potencia activa total	Float	W	-	R
67	2	Potencia reactiva total (Qn)	Float	var	-	R
69	2	Factor de potencia total	Float	-	-	R
71	2	Desbalance de amplitudes de tensión	Float	%	0 ... 100	R
73	2	Desbalance de amplitudes de corriente	Float	%	0 ... 100	R
75	2	Tensión máxima L1-N	Float	V	-	R
77	2	Tensión máxima L2-N	Float	V	-	R
79	2	Tensión máxima L3-N	Float	V	-	R
81	2	Tensión máxima L1-L2	Float	V	-	R
83	2	Tensión máxima L2-L3	Float	V	-	R
85	2	Tensión máxima L3-L1	Float	V	-	R
87	2	Corriente máxima L1	Float	A	-	R
89	2	Corriente máxima L2	Float	A	-	R
91	2	Corriente máxima L3	Float	A	-	R
93	2	Potencia aparente máxima L1	Float	VA	-	R
95	2	Potencia aparente máxima L2	Float	VA	-	R
97	2	Potencia aparente máxima L3	Float	VA	-	R
99	2	Potencia activa máxima L1	Float	W	-	R
101	2	Potencia activa máxima L2	Float	W	-	R
103	2	Potencia activa máxima L3	Float	W	-	R

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
105	2	Potencia reactiva máxima L1 (Qn)	Float	var	-	R
107	2	Potencia reactiva máxima L2 (Qn)	Float	var	-	R
109	2	Potencia reactiva máxima L3 (Qn)	Float	var	-	R
111	2	Factor de potencia máximo L1	Float	-	0 ... 1	R
113	2	Factor de potencia máximo L2	Float	-	0 ... 1	R
115	2	Factor de potencia máximo L3	Float	-	0 ... 1	R
117	2	THD máxima en tensión L1-L2	Float	%	0 ... 100	R
119	2	THD máxima en tensión L2-L3	Float	%	0 ... 100	R
121	2	THD máxima en tensión L3-L1	Float	%	0 ... 100	R
123	2	Reserva	-	-	-	
125	2	Reserva	-	-	-	
127	2	Reserva	-	-	-	
129	2	Frecuencia de red máxima	Float	Hz	45 ... 65	R
131	2	Media máxima en 3 fases de la tensión L-N	Float	V	-	R
133	2	Media máxima en 3 fases de la tensión L-L	Float	V	-	R
135	2	Media máxima en 3 fases de la corriente L-L	Float	A	-	R
137	2	Potencia aparente total máxima	Float	VA	-	R
139	2	Potencia activa total máxima	Float	W	-	R
141	2	Potencia reactiva total máxima (Qn)	Float	var	-	R
143	2	Factor de potencia total máximo	Float	-	-	R
145	2	Tensión mínima L1-N	Float	V	-	R
147	2	Tensión mínima L2-N	Float	V	-	R
149	2	Tensión mínima L3-N	Float	V	-	R
151	2	Tensión mínima L1-L2	Float	V	-	R
153	2	Tensión mínima L2-L3	Float	V	-	R
155	2	Tensión mínima L3-L1	Float	V	-	R

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
157	2	Corriente mínima L1	Float	A	-	R
159	2	Corriente mínima L2	Float	A	-	R
161	2	Corriente mínima L3	Float	A	-	R
163	2	Potencia aparente mínima L1	Float	VA	-	R
165	2	Potencia aparente mínima L2	Float	VA	-	R
167	2	Potencia aparente mínima L3	Float	VA	-	R
169	2	Potencia activa mínima L1	Float	W	-	R
171	2	Potencia activa mínima L2	Float	W	-	R
173	2	Potencia activa mínima L3	Float	W	-	R
175	2	Potencia reactiva mínima L1 (Qn)	Float	var	-	R
177	2	Potencia reactiva mínima L2 (Qn)	Float	var	-	R
179	2	Potencia reactiva mínima L3 (Qn)	Float	var	-	R
181	2	Factor de potencia mínimo L1	Float	-	0 ... 1	R
183	2	Factor de potencia mínimo L2	Float	-	0 ... 1	R
185	2	Factor de potencia mínimo L3	Float	-	0 ... 1	R
187	2	Frecuencia de red mínima	Float	Hz	45 ... 65	R
189	2	Media mínima en 3 fases de la tensión L-N	Float	V	-	R
191	2	Media mínima en 3 fases de la tensión L-L	Float	V	-	R
193	2	Media mínima en 3 fases de la corriente L-L	Float	A	-	R
195	2	Potencia aparente total mínima	Float	VA	-	R
197	2	Potencia activa total mínima	Float	W	-	R
199	2	Potencia reactiva total mínima (Qn)	Float	var	-	R

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
201	2	Factor de potencia total mínimo	Float	var	-	R
203	2	Violaciones de límite*	Unsigned long	-	Byte 3 bit 0 límite 0 Byte 3 bit 1 límite 1 Byte 3 bit 2 límite 2 Byte 3 bit 3 límite 3 Byte 3 bit 4 límite 4 Byte 3 bit 5 límite 5 Byte 3 bit 6 límite 6 Byte 3 bit 7 límite 7 Byte 2 bit 0 límite 8 Byte 2 bit 1 límite 9 Byte 2 bit 2 límite 10 Byte 2 bit 3 límite 11 Byte 0 bit 0 límite LÓGICO Byte 0 bit 1 resultado lógico 1 de los límites en las entradas 0 ... 3 Byte 0 bit 2 resultado lógico 2 de los límites en las entradas 4 ... 7 Byte 0 bit 3 resultado lógico 3 de los límites en las entradas 8 ... 11 Byte 0 bit 4 resultado lógico 4 de los límites en las entradas 12 ... 15	R
205	2	Diagnóstico PMD y estado*	Unsigned long	-	Byte 0 estado de sistema	R

Sistema SCADA para el control de tres centros de transformación

					Byte 1 estado de dispositivo	
					Byte 2 diagnóstico de dispositivo	
					Byte 3 diagnóstico de componentes	
207	2	Estado de salidas digitales*	Unsigned long	-	Byte 3 bit 0 salida 0	R
					Byte 3 bit 1 salida 1	
209	2	Estado de entradas digitales*	Unsigned long	-	Byte 3 bit 0 entrada 0	R
					Byte 3 bit 1 entrada 1	

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido		Acceso
211	2	Tarifa activa	Unsigned long	-	0 =	Tarifa 1	R
					1 =	Tarifa 2	
213	2	Contador de horas de funcionamiento**)	Unsigned long	s	0 ... 999999999		RW
215	2	Contador universal**)	Unsigned long	-	0 ... 999999999		RW
217	2	Contador de cambios en parámetros básicos	Unsigned long	-	-		R
219	2	Contador de cambios en todos los parámetros	Unsigned long	-	-		R
221	2	Contador de cambios de límites	Unsigned long	-	-		R
223	2	Contador de todos los eventos	Unsigned long	-	-		R
225	2	Contador de todas las alarmas	Unsigned long	-	-		R
227	2	Contador de todas las entradas en curvas de carga	Unsigned long	-	-		R
229	2	Contador para otros	Unsigned long	-	-		R
231	2	Estado de salidas digitales, módulo 1*)	Unsigned long	-	Byte 3 bit 0 salida 0	R	
					Byte 3 bit 1 salida 1		

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
233	2	Estado de entradas digitales, módulo 1 ^{*)}	Unsigned long	-	Byte 3 bit 0 entrada 0	R
					Byte 3 bit 1 entrada 1	
235	2	Estado de salidas digitales, módulo 2 ^{*)}	Unsigned long	-	Byte 3 bit 0 salida 0	R
					Byte 3 bit 1 salida 1	
237	2	Estado de entradas digitales, módulo 2 ^{*)}	Unsigned long	-	Byte 3 bit 0 entrada 0	R
					Byte 3 bit 1 entrada 1	
243	2	Cos ϕ L1	Float	-	-	R
245	2	Cos ϕ L2	Float	-	-	R
247	2	Cos ϕ L3	Float	-	-	R
249	2	Desfase L1	Float	°	-	R
251	2	Desfase L2	Float	°	-	R
253	2	Desfase L3	Float	°	-	R
255	2	Ángulo de fase L1-L1	Float	°	-	R
257	2	Ángulo de fase L1-L2	Float	°	-	R
259	2	Ángulo de fase L1-L3	Float	°	-	R
261	2	THD en tensión L1	Float	%	0 ... 100	R
263	2	THD en tensión L2	Float	%	0 ... 100	R
265	2	THD en tensión L3	Float	%	0 ... 100	R
267	2	THD en corriente L1	Float	%	0 ... 100	R
269	2	THD en corriente L2	Float	%	0 ... 100	R
271	2	THD en corriente L3	Float	%	0 ... 100	R
273	2	Distorsión de corriente L1	Float	A	-	R
275	2	Distorsión de corriente L2	Float	A	-	R
277	2	Distorsión de corriente L3	Float	A	-	R
279	2	Potencia reactiva total L1 (Qtot)	Float	var	-	R

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
281	2	Potencia reactiva total L2 (Qtot)	Float	var	-	R
283	2	Potencia reactiva total L3 (Qtot)	Float	var	-	R
285	2	Potencia reactiva L1 (Q1)	Float	var	-	R
287	2	Potencia reactiva L1 (Q1)	Float	var	-	R
289	2	Potencia reactiva L1 (Q1)	Float	var	-	R
291	2	Desbalance de tensión	Float	%	0 ... 100	R
293	2	Desbalance de corriente	Float	%	0 ... 100	R
295	2	Corriente por neutro	Float	A	-	R
297	2	Potencia reactiva total (Qtot)	Float	var	-	R
299	2	Potencia reactiva total (Q1)	Float	var	-	R
301	2	Media móvil de tensión L1-N	Float	V	-	R
303	2	Media móvil de tensión L2-N	Float	V	-	R
305	2	Media móvil de tensión L3-N	Float	V	-	R
307	2	Media móvil de tensión L1-L2	Float	V	-	R
309	2	Media móvil de tensión L2-L3	Float	V	-	R
311	2	Media móvil de tensión L3-L1	Float	V	-	R
313	2	Media móvil de corriente L1	Float	A	-	R
315	2	Media móvil de corriente L2	Float	A	-	R
317	2	Media móvil de corriente L3	Float	A	-	R
319	2	Demanda móvil de potencia aparente L1	Float	VA	-	R
321	2	Demanda móvil de potencia aparente L2	Float	VA	-	R
323	2	Demanda móvil de potencia aparente L3	Float	VA	-	R
325	2	Demanda móvil de potencia activa L1	Float	W	-	R
327	2	Demanda móvil de potencia activa L2	Float	W	-	R
329	2	Demanda móvil de potencia activa L3	Float	W	-	R
331	2	Demanda móvil de potencia reactiva L1 (Qn)	Float	var	-	R

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
333	2	Demanda móvil de potencia reactiva L2 (Qn)	Float	var	-	R
335	2	Demanda móvil de potencia reactiva L3 (Qn)	Float	var	-	R
337	2	Demanda móvil de potencia reactiva total L1 (Qtot)	Float	var	-	R
339	2	Demanda móvil de potencia reactiva total L2 (Qtot)	Float	var	-	R
341	2	Demanda móvil de potencia reactiva total L3 (Qtot)	Float	var	-	R
343	2	Demanda móvil de potencia reactiva L1 (Q1)	Float	var	-	R
345	2	Demanda móvil de potencia reactiva L2 (Q1)	Float	var	-	R
347	2	Demanda móvil de potencia reactiva L3 (Q1)	Float	var	-	R
349	2	Media móvil de factor de potencia L1	Float	-	0 ... 1	R
351	2	Media móvil de factor de potencia L2	Float	-	0 ... 1	R
353	2	Media móvil de factor de potencia L3	Float	-	0 ... 1	R
355	2	Demanda móvil de potencia aparente total	Float	VA	-	R
357	2	Demanda móvil de potencia activa total	Float	W	-	R
359	2	Demanda móvil de potencia reactiva total (Qn)	Float	var	-	R
361	2	Demanda móvil de potencia reactiva total (Qtot)	Float	var	-	R
363	2	Demanda móvil de potencia reactiva total (Q1)	Float	var	-	R
365	2	Media móvil de factor de potencia total	Float	-	-	R
367	2	Media móvil de corriente por neutro	Float	A	-	R
369	2	Contador de horas de funcionamiento del proceso**)	Unsigned long	s	0 ... 999 999 999	RW
371	2	Contador universal 2**)	Unsigned long	-	0 ... 999 999 999	RW

Sistema SCADA para el control de tres centros de transformación

Offset	Número de registros	Nombre	Formato	Unidad	Rango admitido	Acceso
373	2	Contador de impulsos 0**)	Unsigned long	-	0 ... 999 999 999	RW
375	2	Contador de impulsos 02**)	Unsigned long	-	0 ... 999 999 999	RW
377	2	Contador de impulsos 03**)	Unsigned long	-	0 ... 999 999 999	RW
379	2	Contador de impulsos 04**)	Unsigned long	-	0 ... 999 999 999	RW
381	2	Contador de impulsos 05**)	Unsigned long	-	0 ... 999 999 999	RW
383	2	Contador de impulsos 06**)	Unsigned long	-	0 ... 999 999 999	RW
385	2	Contador de impulsos 07**)	Unsigned long	-	0 ... 999 999 999	RW
387	2	Contador de impulsos 08**)	Unsigned long	-	0 ... 999 999 999	RW
389	2	Contador de impulsos 09**)	Unsigned long	-	0 ... 999 999 999	RW
391	2	Contador de impulsos 10**)	Unsigned long	-	0 ... 999 999 999	RW

**.) En todas las magnitudes medidas marcadas con un doble asterisco también se puede utilizar el código de función Modbus 0x10.

Significado de las abreviaturas de la columna "Acceso"

Abreviatura	Significado
R	Read; lectura
W	Write; escritura
RW	Read Write; lectura y escritura