

Alfonso Arranz García

Monitorización de un Sistema de Detección de Incendios

TRABAJO FIN DE GRADO

Dirigido por Enric Vidal Idiarte

Grado de Ingeniería Electrónica Industrial y Automática



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2021

1	Memoria Descriptiva.....	6
1.1	Objeto.....	6
1.2	Alcance.....	6
1.3	Antecedentes.....	6
1.4	Normas y Referencias.....	7
1.4.1	<i>Disposiciones Legales y Normas Aplicadas.....</i>	<i>7</i>
1.4.2	<i>Programas.....</i>	<i>9</i>
1.4.3	<i>Plan de Gestión de la Calidad.....</i>	<i>9</i>
1.4.4	<i>Bibliografía.....</i>	<i>9</i>
1.5	Definiciones y Abreviaturas.....	10
1.5.1	<i>Definiciones.....</i>	<i>10</i>
1.5.2	<i>Abreviaturas.....</i>	<i>11</i>
1.5.3	<i>Código Indicador de Dispositivos.....</i>	<i>12</i>
1.6	Requisitos de Diseño.....	16
1.6.1	<i>Situación y Emplazamiento de las Instalaciones.....</i>	<i>16</i>
1.6.2	<i>Datos de Partida.....</i>	<i>16</i>
1.6.3	<i>Centrales de Detección de Incendios.....</i>	<i>16</i>
1.6.4	<i>Condicionantes.....</i>	<i>19</i>
1.7	Análisis de Soluciones.....	20
1.7.1	<i>Solución Basada en Paneles Repetidores.....</i>	<i>20</i>
1.7.2	<i>Solución Basada en Software Gráfico del Fabricante.....</i>	<i>20</i>
1.7.3	<i>Solución Basada en Software Gráfico de Terceros.....</i>	<i>21</i>
1.7.4	<i>Solución Basada en la Integración con el Sistema de Control.....</i>	<i>21</i>
1.7.5	<i>Solución Adoptada.....</i>	<i>22</i>
1.8	Resultados Finales.....	23
1.8.1	<i>Infraestructura de Comunicación.....</i>	<i>23</i>
1.8.2	<i>Arquitectura de Integración con el Sistema de Control.....</i>	<i>36</i>
1.8.3	<i>Protocolo MODBUS-TCP.....</i>	<i>41</i>
1.8.4	<i>Conceptos Básicos de los Sistemas de Detección de Incendios.....</i>	<i>48</i>
1.8.5	<i>Comunicaciones con Notifier ID60 de Honeywell.....</i>	<i>49</i>
1.8.6	<i>Comunicaciones con ZP2 de Ziton.....</i>	<i>52</i>
1.8.7	<i>Sistema de Control PCS7.....</i>	<i>54</i>
1.9	Planificación.....	90
2	Anexos.....	93
2.1	Direccionamiento Plant Bus.....	94
2.2	Switch SCALANCE XC206-2SFP.....	95
2.3	Certificación Transceptor SFP992-1.....	107

2.4	CPU 317-2 PN/DP	111
2.5	CP 343-1	120
2.6	PS 307 (5 A)	123
2.7	Pasarela IBOX Modbus Server	126
2.8	Ziton Fire Panel Modbus Protocol	166
2.9	PCS7: Configuración de Hardware y Redes	177
2.10	PCS7: Jerarquía de Planta.....	182
2.11	PCS7: Jerarquía en Pantallas del Sistema.....	185
2.12	FB550: NOTIF60	190
2.13	FB551: NOTMOD.....	210
2.14	FB552: NOTSEN.....	220
2.15	FB553: NOTDEV	230
2.16	FB554: ZITONP2.....	238
2.17	FB555: ZITMOD	255
2.18	FB556: ZITDEV	265
2.19	FB557: ZITALM.....	273
2.20	PCS7: Librería Block Icons	278
3	Planos.....	280
3.1	Situación.....	281
3.2	Emplazamiento.....	282
3.3	Red General Contra Incendios	283
3.4	Detectores Edif. Oficinas Planta Baja	284
3.5	Detectores Edif. Oficinas Primera Planta.....	285
3.6	Detectores Edif. Laboratorios	286
3.7	Detectores Edif. Capability Lab	287
3.8	Detectores Edif. Vestuarios/Comedor	288
3.9	Detectores Varios	289
3.10	Detectores Servidores Edif. Laboratorios.....	290
3.11	Detectores Servidores Plantas.....	291
3.12	Detectores Sala Eléctrica Planta Finos	292
3.13	Distribución CDI.....	293
3.14	Trazado Fibra Óptica	294
3.15	Arquitectura del Sistema de Control	295
4	Pliego de Condiciones.....	298
4.1	Condiciones Técnicas.....	298
4.1.1	Condiciones de la Instalación Eléctrica.....	298
4.1.2	Condiciones de la Instalación del PLC	298

4.1.3	<i>Condiciones del Cableado</i>	299
4.1.4	<i>Condiciones de Alimentación a 24V</i>	299
4.2	Condiciones Facultativas	299
4.2.1	<i>Responsabilidades del Contratista</i>	299
4.2.2	<i>Responsabilidades del Contratante</i>	299
4.3	Condiciones Económicas	299
4.3.1	<i>Precios</i>	299
4.3.2	<i>Revisión de Precios</i>	300
4.3.3	<i>Penalizaciones</i>	300
4.4	Condiciones Legales	300
4.4.1	<i>Contrato</i>	300
4.4.2	<i>Rescisión de Contrato</i>	300
5	Presupuesto	303
5.1	Justificación de Precios	303
5.1.1	<i>Mano de obra</i>	303
5.1.2	<i>Materiales</i>	303
5.1.3	<i>Maquinaria</i>	304
5.1.4	<i>Licencias de Software</i>	305
5.2	Mediciones	306
5.2.1	<i>Instalación Infraestructura de red</i>	306
5.2.2	<i>Instalación Switches</i>	307
5.2.3	<i>Instalación PLC</i>	307
5.2.4	<i>Instalación Servidor OS</i>	308
5.2.5	<i>Estudio y Desarrollo del Programa</i>	308
5.3	Capítulos	309
5.3.1	<i>Instalación Infraestructura de red</i>	309
5.3.2	<i>Instalación Switches</i>	310
5.3.3	<i>Instalación PLC</i>	311
5.3.4	<i>Instalación Servidor OS</i>	312
5.3.5	<i>Estudio y Desarrollo del Programa</i>	313
5.4	Resumen del Presupuesto	314

Monitorización de un Sistema de Detección de Incendios

MEMORIA DESCRIPTIVA



UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2021

1 Memoria Descriptiva

1.1 Objeto

El objeto de este proyecto es centralizar la monitorización de las diferentes centrales de protección contra incendios (PCI) que conforman el sistema de detección y alarma de incendios en la planta química IFF Benicarló, debido a la necesidad de su atención permanente por el personal de seguridad o personal autorizado como recomienda la norma.

UNE 23007-14:2014. Punto 6.7: El equipo de control e indicación del Sistema de Detección y Alarma de Incendios debe estar preferiblemente en un lugar en el que haya personal permanentemente.

1.2 Alcance

El alcance del presente proyecto es la instalación de la infraestructura necesaria para la comunicación de las todas las centrales PCI que conforman el sistema de detección y alarma de incendios con un sistema centralizado de supervisión, así como la instalación y programación de los diferentes niveles de este último.

1.3 Antecedentes

Las exigencias normativas en materia de protección contra incendios en los edificios, industrias, instalaciones e infraestructuras definen la necesidad de disponer de sistemas de protección contra incendios y en particular de sistemas de detección y alarma de incendios como medida de protección activa en las instalaciones, según determinadas condiciones para el cumplimiento de la legislación vigente.

Dichos sistemas de detección y alarma de incendios reciben señales de los diferentes sistemas de protección activa contra incendios de las instalaciones protegidas, con el fin de supervisar el estado operativo de los mismos y transmitir cualquier alarma o fallo para su atención prioritaria.

En las instalaciones protegidas con estos sistemas que no disponen de personal formado para su atención de forma permanente y no están conectadas a un centro de control o receptor con dicho personal, no se garantiza la atención de las señales de incendio y por tanto de un posible siniestro, aumentando el riesgo de daños colaterales, costes, responsabilidad penal y sanciones derivadas.

La relación directa entre el retraso en la gestión e intervención en caso de alarma, frente al riesgo de que un posible incendio se desarrolle y cobre una dimensión que haga inviable su control, exige que los sistemas de protección contra incendios estén debidamente atendidos y se comuniquen las alarmas con el menor retardo posible.

Igualmente, la necesaria intervención en caso de alarma de los diferentes cuerpos públicos de actuación (fuerzas de seguridad, bomberos y diferentes organismos para la emergencia), hacen imprescindible que se vele por la correcta gestión y transmisión de las alarmas desde su inicio, con objeto de evitar usos indebidos de los recursos públicos y potenciales siniestros en caso de falsas alarmas o de alarmas transmitidas con retraso dada su incorrecta gestión desde su origen.

Las centrales PCI que conforman el sistema de detección y alarma de incendios en la planta química IFF Benicarló están localizadas en diferentes emplazamientos de las instalaciones,

de las cuales, algunas no se puede asegurar su atención permanentemente como recomienda la norma.

Es por ello que, dentro del ámbito de la protección de riesgos laborales, es responsabilidad de la empresa adoptar las medidas de lucha contra incendios y evacuación de los trabajadores garantizando la rapidez y eficacia de los mismos.

LEY 31/1995 PRL Artículo 20. Medidas de emergencia:

El empresario, teniendo en cuenta el tamaño y la actividad de la empresa, así como la posible presencia de personas ajenas a la misma, deberá analizar las posibles situaciones de emergencia y adoptar las medidas necesarias en materia de primeros auxilios, lucha contra incendios y evacuación de los trabajadores, designando para ello al personal encargado de poner en práctica estas medidas y comprobando periódicamente, en su caso, su correcto funcionamiento. El citado personal deberá poseer la formación necesaria, ser suficiente en número y disponer del material adecuado, en función de las circunstancias antes señaladas.

Para la aplicación de las medidas adoptadas, el empresario deberá organizar las relaciones que sean necesarias con servicios externos a la empresa, en particular en materia de primeros auxilios, asistencia médica de urgencia, salvamento y lucha contra incendios, de forma que quede garantizada la rapidez y eficacia de las mismas.

1.4 Normas y Referencias

1.4.1 Disposiciones Legales y Normas Aplicadas

- UNE-EN ISO11442:2006: Documentación Técnica de los documentos.
- UNE-EN ISO 81714: Diseño de símbolos gráficos en la documentación técnica de productos.
- UNE-EN ISO 9000:2005: Sistemas de gestión de calidad. Fundamentos y vocabulario.
- UNE-EN ISO 7200:2004: Documentación técnica de productos. Gestión de documentos.
- UNE-EN ISO 5457:2000: Documentación técnica de productos. Formatos y presentación de los elementos gráficos.
- UNE-EN ISO 5455:1996: Dibujos técnicos. Escalas.
- UNE-EN ISO 3098: Documentación técnica de producto. Escritura.
- UNE-EN ISO 80416: Principios básicos para símbolos gráficos utilizables en los equipos.
- UNE-EN ISO 61131: Autómatas programables.
- UNE-EN ISO 61082-1:2007: Preparación de documentos utilizados en electrotecnia.
- UNE-EN 60445:2012: Principios fundamentales y de Seguridad para la interfaz hombre máquina, el marcador y la identificación. Identificación de bornes de equipos y terminales de ciertos conductores designados y reglas generales para un sistema alfanumérico.
- UNE-EN 60027: Símbolos literales utilizados en electrotecnia.
- UNE 157001:2014: Criterios generales para la elaboración de proyectos.
- UNE 1135:1989: Dibujos técnicos. Lista de elementos.
- UNE 1039:1994: Dibujos técnicos. Acotación. Principios generales, definiciones, métodos de ejecución e indicaciones especiales.
- UNE 1032:1982: Dibujos técnicos. Principios generales de representación.
- UNE 1027:1995: Dibujos técnicos. Plegado de planos.
- EN ISO 10209:2012: Documentación técnica de producto. Vocabulario. Términos relacionados con los diseños técnicos, la definición de productos y productos relacionados.

- EN ISO 6433:2010: Dibujos técnicos. Referencia de partes.
- RD 2200/1995: Reglamento de la Infraestructura para calidad y seguridad industrial.
- RD 486/1997: Disposiciones mínimas de seguridad y Salud en los puestos de trabajo.
- RD 842/2002: Reglamento Electrotécnico de Baja Tensión, incluyendo las modificaciones de RD 560/2010 y según RD 1053/2014.
- LEY 54/2003 del 12 de diciembre, de reforma del marco normativo de la prevención de riesgos laborales.
- LEY 31/1995, de 8 de noviembre sobre Prevención de riesgos Laborales. Normativa Española relevante sobre la Seguridad y Salud en el Trabajo. Actualizado el 24 de febrero de 2012.
- UNE-EN ISO 11354-1:2012: Tecnologías avanzadas de automatización y sus aplicaciones. Requisitos para establecer la interpretación entre procesos de fabricación.
- IEC 61508- Seguridad en Sistemas Electrónicos Programables.
- IEC 61131- Conjunto de normas e informes técnicos de la comisión electrotécnica Internacional.
- Estándar IEC 1131-3
- UNE-EN 61131-1:2003: Autómatas programables. Parte 1: Información general.
- UNE-EN 61131-2:2007: Autómatas programables. Parte 2: Requisitos y ensayos de los equipos.
- UNE-EN 61131-3:2003: Autómatas programables. Parte 3: Lenguajes de programación. (Ratificada por AENOR en mayo de 2004)
- UNE-EN 61131-4:2004: Autómatas programables. Parte 4: Guías de usuario.
- UNE-EN 61131-5:2001: Autómatas programables. Parte 5: Comunicaciones. (Ratificada por AENOR en febrero de 2002).
- UNE-EN 61131-6:2012: Autómatas programables. Parte 6: Seguridad funcional. (Ratificada por AENOR en enero de 2013).
- IEC TR 61131-8:2000: Autómatas programables. Parte 8: Directrices para la aplicación e implementación de lenguajes de programación.
- UNE-EN ISO/IEC 7498-1:1995: Tecnología de la información. Interconexión de sistemas abiertos. Modelo de referencia básico: el modelo básico (Ratificada por AENOR en noviembre de 1995).
- EN 54 parte 2: Equipos de control e indicación (Central de detección de incendios o paneles de incendio).
- EN 54 parte 3: Equipos de control e indicación. Dispositivo de Notificación acústico o Sirena.
- EN 54 parte 13: Evaluación de compatibilidad de los componentes de un sistema.
- RD 513/2017: Reglamento de instalaciones de protección contra incendios.
- RD 393/2007: Norma Básica de Autoprotección de los centros, establecimientos y dependencias dedicados a actividades que puedan dar origen a situaciones de emergencia.
- RFC 1918: Asignación direcciones para redes de privadas.
- UNE-EN 62439-2:2017: Redes de comunicación industrial. Redes de automatización de alta disponibilidad. Parte 2: Protocolo de redundancia de medios (MRP).
- UNE-EN 60079-1:2015: Atmósferas explosivas. Parte 1: Protección del equipo por envoltentes antideflagrantes "d".
- UNE-EN 60079-28:2016: Atmósferas explosivas. Parte 28: Protección de material y sistemas de transmisión que utilizan radiación óptica.

1.4.2 Programas

Para el desarrollo del proyecto se han utilizado los siguientes programas:

- *PCS7 SP3*: Sistema integrado de Siemens para el control de procesos. Este software incluye todos los componentes necesarios para la programación del control y el entorno gráfico del sistema de supervisión.
- *TIA Portal V16*: Entorno de programación de Siemens para la programación de paneles de campo Confort Panel.
- *Notepad ++*: Editor de texto gratuito para la edición y formato del código fuente.
- *Ananas64*: Programa servidor de protocolo Modbus/TCP para la simulación de centrales de protección contra incendios.
- *Gantt Project*: Software de gestión para la planificación del proyecto.
- *MS Office*: Software ofimático para la redacción del proyecto y tareas auxiliares como la creación de tablas con listados de señales, descripciones, etc.
- *MS Visio*: Software de diseño gráfico para la elaboración de esquemas de red.
- *Autocad 2019 LT*: Software de diseño gráfico para la creación de planos.
- *Corel Draw 2018*: Software de diseño gráfico para la elaboración de objetos gráficos utilizados en esquemas de red, y en el entorno gráfico del sistema de supervisión.

1.4.3 Plan de Gestión de la Calidad

Proyecto redactado siguiendo el plan de calidad ISO 9001 implantado en IFF Benicarló.

1.4.4 Bibliografía

Bibliografía utilizada para el desarrollo del proyecto:

- Documentación técnica PCS7 SP3 disponible en el siguiente enlace:
<https://new.siemens.com/global/en/products/automation/process-control/simatic-pcs-7/technical-documentation.html>
- Configuration of Messages and Alarms in WinCC (TIA Portal).
- Libraries APL Style Guide. Siemens Programming Manual.
- Process Control System PCS 7 V7.0 Programming Instructions for Blocks.
- Structured Control Language (SCL) for S7-300/S7-400 Programming Manual.
- S7-SCL V5.3 for S7-300/400. Siemens Programming Manual.
- Technical Reference – Modbus TCP/IP. Acromag Manual
- WinCC V7.4. Scripting (VBS, ANSI-C, VBA). Siemens Programming Manual.
- Pointers and Indirect Addressing in S7 PLCs, Volume 3. Siemens Application Notes.
- Proyecto final de grado: Automatización y control de un proceso de fabricación de cilindros de simple efecto. Autor: Aleix Povill Ricomà. 2017.

- Conexiones de Sistemas de Protección contra Incendios a Centros de recepción de Alarmas y Averías. Asociación de Empresas de Seguridad (AES).
- Diferente contenido de páginas web, tales como Wikipedia.org , modbus.org, etc.

1.5 Definiciones y Abreviaturas

1.5.1 Definiciones

- **Sensor:** Dispositivo que convierte el valor de una magnitud física (presión, flujo, temperatura, etc.) en una señal eléctrica codificada (analógica o digital).
- **Dispositivo de señal simple:** Elemento de protección activa de un sistema de detección y alarma de incendios que es capaz de comunicar una única señal de estado a una central de detección de incendios.
- **Dispositivo multi-senal:** Elemento de protección activa de un sistema de detección y alarma de incendios que es capaz de comunicar varias señales de estado a una central de detección de incendios.
- **Interfaz:** Conexión funcional entre dos sistemas, programas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles, permitiendo el intercambio de información.
- **Ethernet:** Estándar de redes de área local para dispositivos de comunicación, por sus siglas en español Acceso Múltiple con Escucha de Portadora y Detección de Colisiones (CSMA/CD).
- **TCP/IP:** Conjunto de protocolos que permiten la comunicación entre los elementos pertenecientes a una red. La sigla TCP/IP significa Protocolo de control de transmisión/ Protocolo de Internet.
- **ModBus/TCP:** Protocolo de comunicaciones desarrollado por Modicon, que ha evolucionado para trabajar sobre una interfaz TCP/IP que comunica vía Ethernet.
- **Servidor:** Dispositivo de una red capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.
- **Cliente:** Dispositivo de una red que consume un servicio remoto en otro dispositivo conocido como servidor.
- **Hub:** O concentrador, es un dispositivo de interconexión utilizado para conectar equipos en red. A través de un Hub la información enviada por el dispositivo de origen se replica en todos sus puertos.
- **Switch:** O conmutador, es un dispositivo de interconexión utilizado para conectar equipos en red. A través de un Switch la información enviada por el dispositivo de origen va directamente al dispositivo de destino sin replicarse en el resto de puertos.
- **Gateway:** O pasarela de comunicaciones, es un dispositivo normalmente hardware que permite que dispositivos de una misma red con protocolos de comunicación diferentes puedan comunicarse entre sí. Se trataría de un dispositivo traductor de protocolos.
- **Bus:** Es una vía de transmisión que sirve para enlazar diferentes equipos. Los datos se intercambian de forma bidireccional entre equipos mediante conductores eléctricos o fibra óptica.
- **Modelo OSI:** Es un modelo de referencia para los protocolos de red que tiene por objetivo conseguir interconectar sistemas de procedencia distinta para que estos puedan intercambiar información. Está conformado por 7 capas o niveles de abstracción, cada uno de ellos con sus propias funciones.
- **Topología en anillo:** Es una configuración de red donde las conexiones de los dispositivos crean una ruta circular de los datos. Cada dispositivo en la red está conectado

a otros dos, el de adelante y el de atrás, formando así una ruta continua única para transmitir la señal, como los puntos en un círculo.

- **Autómata Programable (PLC):** Son controles o comandos electrónicos en los que la función a desarrollar está memorizada en forma de programa. Su estructura y cableado no dependen de la función del autómata. Un Autómata está constituido por una CPU y opcionalmente módulos de entradas y salidas y módulos de comunicaciones.
- **PCS7:** Sistema de control de Siemens orientado al control de procesos. Este software de arquitectura abierta cliente/servidor ofrece todos los recursos necesarios para la programación y control de procesos continuos o por lotes (Batch)
- **Hardware:** Equipos físicos de un sistema.
- **Software:** Programa o conjunto de programas informáticos.
- **Tag:** O etiqueta, es un código alfanumérico que identifica cada uno de los dispositivos o señales de un sistema de forma unívoca.
- **Bit:** Unidad elemental de información constituida por los valores 1 o 0.
- **Byte:** Conjunto de 8 bits.
- **Word:** Conjunto de 16 bits.
- **Bloque:** Subrutina de un programa de PLC.
- **Instrucción:** La instrucción es la unidad independiente más pequeña de un programa.

1.5.2 Abreviaturas

- **PCI:** Protección contra incendios.
- **CCM:** Cuarto de control de maniobras eléctricas.
- **CEX:** Central de extinción de incendios.
- **CDI:** Central de detección de incendios.
- **CAH:** Central de aspiración de humos.
- **FO:** Fibra óptica.
- **OS:** Operation System.
- **AS:** Automation System.
- **ES:** Engineering System.
- **PLC:** Programmable Logic Controller.
- **CPU:** Central process unit.
- **CP:** Communications processor.
- **PC:** Personal Computer.
- **CFC:** Continuous Function Chart.
- **FB:** Function Block.
- **DB:** Data Block.
- **SCL:** Structured Control Language.
- **FC:** Function Command.
- **ADU:** Application Data Unit.
- **PDU:** Protocol Data Unit.
- **SDK:** Software development kit

1.5.3 Código Indicador de Dispositivos

Derivado de la necesidad de tener una mejor gestión y control de los elementos del sistema, se definen una serie de criterios, por los cuales, cada uno de los dispositivos que conforman el sistema de detección y alarma de incendios tendrá un identificador único (TAG) con el siguiente formato:

AAA-BBB-CDD(-EEE)

Donde los respectivos dígitos se definen como:

- AAA: Numero de área en la cual está instalado el dispositivo.
- BBB: Código que identifica el tipo de dispositivo.
- C: Número de elevación (0 para planta baja, 1 primera planta, etc.)
- DD: Numero de orden del dispositivo del mismo tipo dentro de cada elevación
- EEE: Código que identifica el tipo de señal para dispositivos multi-señal.

1.5.3.1 Número de Área (AAA)

Tipo de Área	Núm. Área	Descripción
Plantas de Producción (100-299)	100	Planta Piloto
	110	Planta Síntesis I
	120	Planta Síntesis II
	130	Planta Hidrogenación
	140	Planta Pirolisis
	150	Planta Especialidades
	160	Planta Acetatos
	170	Planta Finos
	180	Planta Aguarrás
	190	Planta Naturales
Almacenamiento de móviles (300-399)	200	Planta de mezclas
	300	Almacén L1
	301	Almacén L4
	302	Almacén L2
	303	Almacén L3
	304	Almacén sólidos
	310	Almacén refrigerado M1
	311	Almacén refrigerado M3
	312	Almacén refrigerado M2
	320	Almacén Corrosivos
	330	Almacén TEAL
	331	L7 Botellones
	332	Almacén Anhídrido Sulfúrico
	333	Almacén CL2
334	Bahías Hidrógeno	

Tipo de Área	Núm. Área	Descripción
Almacenamiento fijo (400-499)	400	Tanques Aguarrás
	401	Tanques de Sosa y H2SO4
	402	Almacenamiento gasoil
	403	Tanques de sosa
	404	Tanques planta de Finos
	405	Tanques planta de Acetatos
	406	Tanques H830
	407	Tanques Hidrogenación
	408	Tanques aguarrás y Mirceno
	409	Tanque Isopreno
	410	Tanques Síntesis II
	411	Tanque Acetaldehído
	412	Cargadero Acetaldehído
	413	Cargadero especialidades
	414	Cargadero Pirolisis
	415	Cargadero Síntesis I
	416	Cargadero Bromo
417	Cargadero Sulfúrico	
Oficinas y laboratorios (500-599)	500	Edificio Oficinas
	510	Edificio Laboratorios
	520	Laboratorio Capability
	530	Laboratorio hidrogenación
	540	Laboratorios Naturales
	550	Portería
	560	Vestuarios y comedor
	570	Vestuarios contratas
580	Vestuarios móviles	
Auxiliares (600-699)	600	Expediciones
	610	Taller mecánico
	614	Taller eléctrico
	620	Estufas Especialidades
	621	Estufas Síntesis I
	622	Estufas Naturales
Salas eléctricas y GE (700-799)	700	CT1
	701	CCM NSPP
	702	CCM Finos
	703	CT-2
	704	CCM Acetatos
	705	CCM Naturales
	706	GE Naturales
	707	CT-3
	708	CCM Síntesis 2
	709	CCM Especialidades
710	CCM Síntesis 1	

Tipo de Área	Núm. Área	Descripción
Depuradora y residuos	800	PTAR
(800-899)	810	Filtro prensa
	820	Sala de control PTAR
	830	Evapocondensación
	840	tanques residuos calderas
	850	Balsas frente a naturales
	860	Balsa 1 M
	870	Balsa 6 M
Servicios de planta	900	Calderas
(900-999)	910	aire comprimido
	920	Nitrógeno
	930	agua de pozo
	940	BCI-01 y BCI-02
	941	BCI-03
	950	Torres refrigeración Finos
	951	Torres refrigeración Acetatos
	952	Torres refrigeración Sint I e Hyd
	953	Torres refrigeración Especialidades
	954	Torres refrigeración Síntesis II
	960	Servicios de agua Fría
	970	Servicios de agua Glicolada
	980	Oxígeno
	990	Aceite térmico Especialidades

1.5.3.2 Tipo de Dispositivo (BBB)

Código	Descripción
DG	Detector de gas
DLL	Detector de llama
DH	Detector de humo
DLI	Detector lineal óptico de humo (barrera de detección infrarroja)
DT	Detector térmico
PAL	Pulsador de alarma
PIN	Pulsador de inhibición
PEX	Pulsador de disparo de extinción
PPC	Pulsador local de agua (puesto de control diluvio)
PC	Puesto de control tanto para húmedo como seco
EXG	Sistema de extinción por agente limpio gaseoso
H	Hidrante
BIE	Boca de incendios equipada
I	Cañón agua-espuma
E	Sistema de extinción de espuma (lanza)
DEF	Depósito de espumógeno fijo
DEM	Depósito de espumógeno móvil
BCI	Bomba contraincendios
BJ	Bomba Jockey
CEX	Central de extinción

Código	Descripción
CDI	Central detección incendios
CAH	Central detección de humo por aspiración
BOC	Bocinas de megafonía
SIR	Sirena
SSD	Sistema de bloqueos de planta (Shutdown)
PAA	Sistema de parada aire acondicionado
CCV	Sistema para cerrar compuertas de ventilación
PA	Presostato de alarma
FSL	Detector de flujo
SV	Solenoides
FA	Fuente de alimentación
BAT	Batería
CCF	Compuerta cortafuegos
IAL	Indicador de alarma

1.5.3.3 Tipo de Señal (EEE)

Código	Descripción
PAL	Pre alarma
ACT	Activado
STP	En paro
RUN	En marcha
MAN	En modo manual
AUT	En modo automático
INH	Anulado
EXT	Extinción realizada
FLT	Avería
DH[n]	Central aspiración de humo (CAH). [n] Zona de aspiración
EX[n]	Central extinción (CEX). [n] Zona de extinción

1.5.3.4 Ejemplos de Aplicación

Para dispositivos de señal simple:

- 520-DH-102: Segundo detector de humo del 1º piso del laboratorio de capability.
- 530-SIR-001: Sirena ubicada en la planta baja del laboratorio de hidrogenación.

Para dispositivos multi-señal:

- 940-BCI-001-MAN: Bomba contraincendios 1 en modo manual.
- 702-CEX-001-EXT: Extinción realizada en la central de extinción ubicada en la planta baja del CCM de Finos.
- 702-CEX-001-EX1: Zona de extinción 1 activa en la central de extinción ubicada en la planta baja del CCM de Finos.
- 702-CAH-001-DH2: Detección de humo en zona de aspiración 2 para la central de aspiración ubicada en la planta baja del CCM de Finos.

1.6 Requisitos de Diseño

1.6.1 Situación y Emplazamiento de las Instalaciones

El presente proyecto está dirigido a las instalaciones de IFF Benicarló, Avd. de Felipe Klein, 2, 12580 Benicarló, Castellón. Su situación y emplazamiento quedan reflejados en los planos 3.1 Situación y 3.2 Emplazamiento.

1.6.2 Datos de Partida

Las instalaciones de IFF Benicarló disponen en la actualidad de un sistema de detección y extinción de incendios, cuya distribución general pueden verse en el plano 3.3. *Red General Contra Incendios*. Para el interior de edificios, los planos comprendidos entre 3.4. *Detectores Edif. Oficinas Planta Baja* y 3.12. *Detectores Sala Eléctrica Planta Finos* muestran la distribución de elementos tales como detectores de humo, pulsadores de emergencia, etc. que se tomaran en cuenta para la ejecución de este proyecto. Todos estos planos han sido suministrados por la propiedad.

Este sistema dispone de *ocho* centrales de detección de incendios (CDI) ubicadas en diferentes emplazamientos de las instalaciones, cuya distribución se puede ver en el plano 3.13. *Distribución CDI*.

Cada una de estas CDI reciben señales de los diferentes sistemas de protección activa contra incendios de las instalaciones protegidas próximas a ellas, con el fin de supervisar el estado operativo de los mismos. De las CDI actuales podemos encontrar dos modelos diferentes. Por un lado, en los edificios de oficinas y de laboratorios están instaladas el modelo ZP2 de la marca Ziton y para en el resto de instalaciones encontramos el modelo Notifier ID60 de la marca Honeywell. Cada uno de estos modelos es incompatible entre sí y con característica técnicas diferentes.

1.6.3 Centrales de Detección de Incendios

1.6.3.1 ZP2 de Ziton

Los sistemas de la serie ZP2 aportan la velocidad y funcionalidad de procesamiento inteligente de alta gama para aplicaciones de tamaño pequeño y mediano. Con la detección direccionable y una gama completa de configuraciones, tarjetas y módulos opcionales, así como conectividad USB y Ethernet.



Figura 1. Central ZP2 de Ziton

Entre sus características más destacables tenemos:

- Puerto Ethernet con TCP/IP para programación y mantenimiento remotos.
- LCD gráfica compatible con EN54 con iconos y símbolos para facilitar reconocimiento de eventos.
- Jog Dial con 4 botones suaves para un control de usuario simple e intuitivo.
- Diseño agradable y estético con puerta sintética especial para pintar.
- Puerta y chasis fácilmente extraíbles para una instalación rápida y limpia.
- Todos los conectores son enchufables.
- 3 puertos USB con soporte para lápiz de memoria y RS232 para soporte de impresora.
- Notificación por correo electrónico de eventos directamente desde el panel (4 cuentas de usuario)
- Salida de suministro auxiliar de 24V CC con soporte de reinicio.
- Estructura de menú de 3 niveles de operador con nombre de usuario y contraseña.
- Memoria de registro de historial para 9999 eventos.
- Hasta 72 horas en espera y 30 minutos de alarma.
- Compatible con la base y las sirenas / balizas alimentadas por bucle Ziton, y sirenas y detectores de la serie Ziton.
- 1 bucle que admite hasta 254 dispositivos (127 direcciones) en 512 zonas.
- 2 salidas de enrutado de incendios y sirenas supervisadas, que pueden ser utilizadas como salidas programables.
- 2 salidas de relé convencionales y 2 salidas supervisadas trabajando en pareja y dedicadas a condiciones comunes de incendio y falla.
- 2 entradas de usuario configurables para monitoreo y control.

Desde el punto de vista de conectividad remota cuenta con diferentes opciones de ampliación. En el siguiente listado se definen estas opciones, así como las referencias del fabricante:

- *2010-2-PAK-RMSDK*: Licencia de activación para habilitar la integración en plataformas externas a través de SDK vía Ethernet.
- *2010-2-PAK-RMMB*: Licencia de activación de protocolo estándar Modbus/TCP.
- *2010-2-NB*: Tarjeta de red basada en comunicaciones RS485 con protocolo propietario Firenet. Permite crear un bus con topología en anillo de hasta 32 nodos, con una distancia máxima de 1200m entre nodos.
- *ZP2-FR-C-99*: Panel repetidor ZP2. Conectado al bus Firenet, se utiliza para monitorizar y controlar el resto de centrales del bus de forma remota.



Figura 2. Panel repetidor ZP2

1.6.3.2 Notifier ID60 de Honeywell

Central microprocesada analógica algorítmica de 1 lazo para la detección y alarmas de incendio que monitoriza y controla individualmente los elementos del sistema.



Figura 3. Central Notifier ID60 de Honeywell

Entre sus características más destacables tenemos:

- Sistema compacto con fuente de alimentación incorporada y capacidad de 1 lazo de detección analógica inteligente con 6 circuitos de salida, 2 de entrada configurables y 1 de salida AUX 24Vcc. para alimentar equipos externos.
- Capacidad de protección activa de hasta 16 zonas.
- El lazo puede controlar un máximo de 99 detectores analógicos inteligentes más 99 módulos monitores (entrada) o de control (salida), con hasta un total de 198 puntos identificables individualmente.
- Incluye aisladores de lazo en la entrada y salida de cableado del lazo.
- Puede alimentar sirenas y detectores de rayo a través del lazo SLC (según especificaciones). Incorpora algoritmos AWACS para poder conectar sensores NFXI-VIEW y SMART 4.
- Dispone de pantalla alfanumérica de cristal líquido LCD de 2 líneas de 40 caracteres, teclado de membrana con teclas de funciones y control y leds para la visualización del estado del sistema e indicación individual de las 16 zonas en alarma y avería/fallo/anulado.

Desde el punto de vista de conectividad remota cuenta con diferentes opciones, algunas de ellas como ampliación. En el siguiente listado se definen estas opciones, así como las referencias del fabricante:

- Incluido en la propia central un interfaz de comunicación RS232 con conector DB9 para la conexión a impresora, programa de configuración, integración o Software Gráfico de control remoto. Con protocolo propietario Notifier ID
- 020-553: Tarjeta de red basada en comunicaciones RS485 con protocolo propietario para la conexión de repetidores remotos.
- 002-450-001: Panel repetidor. Conectado al bus RS485, puede monitorizar y controlar la central a la que está conectada de forma remota.

- *IBOX-MBS-NID3K*: Pasarela de comunicaciones (Gateway) que convierte el protocolo de las centrales serie ID en el estándar Modbus Server para comunicarse vía RS232 o RS485 con equipos Modbus/RTU o vía Ethernet a través de RJ45 a Modbus/TCP.



Figura 4. Pasarela NotifierID / ModbusTCP

1.6.4 *Condicionantes*

En el diseño del proyecto y la solución a adoptar, se fijan tres condicionantes básicos que tendrán que adoptarse en la medida de lo posible.

Las condiciones de trabajo en las instalaciones hacen que no se puede asegurar que haya una persona permanentemente en un centro de control. Este motivo obliga a que el sistema de supervisión tenga que ser flexible en el número y la ubicación de los puestos de monitorización, asegurando de este modo que haya una atención permanente máxima.

El sistema de supervisión tiene que ser de fácil manejo, sin requerir de un gran esfuerzo en formación. Cualquier trabajador de las instalaciones tiene que ser capaz de interpretar la información que brinda el sistema de una forma intuitiva, independientemente de su grado de formación o responsabilidad, con el fin de poder actuar de forma preventiva.

La necesidad de instalar una infraestructura de comunicaciones nueva con la inversión económica que ello supone, hace que se requiera una solución de comunicaciones estandarizada y que pueda ser aprovechada para otros sistemas. De este modo, como preferencia en el diseño, se busca una solución basada en redes ethernet.

1.7 Análisis de Soluciones

Entre las diferentes alternativas estudiadas en función de las posibilidades de conectividad remota de cada uno de los modelos de CDI, y teniendo en cuenta los condicionantes requeridos del apartado *1.6.4 Condicionantes* destacamos las siguientes.

1.7.1 Solución Basada en Paneles Repetidores

Esta solución se basa en la instalación de paneles repetidores de cada modelo en los puestos de supervisión.

1.7.1.1 Ventajas

- Se trata de una solución basada prácticamente en instalación de hardware, por lo que la inversión en tiempo de ingeniería para el desarrollo del proyecto es mínima.

1.7.1.2 Inconvenientes

- Infraestructura de comunicación compleja y no estandarizada.
- Necesidad de una infraestructura basada en comunicaciones RS485 para las CDI ZP2 y otra independiente para las CDI Notifier, debido a la incompatibilidad de protocolos entre ellas.
- Puestos de supervisión limitados en número y ubicación, ya que están ligados al trazado de la infraestructura de comunicaciones.
- Necesidad de dos estaciones remotas por cada puesto de supervisión, incrementando el coste y el mantenimiento.
- Necesidad de un alto grado de formación.

1.7.2 Solución Basada en Software Gráfico del Fabricante

Esta solución se basa en la instalación del software gráfico de supervisión que ofrece cada uno de los fabricantes. Por un lado, el *ATS Advisor Management Software (ATS8610)* para las CDI ZP2 y por otro, el *TG-ID50/60* para las CDI Notifier ID60.

1.7.2.1 Ventajas

- Infraestructura de comunicación basada en redes ethernet añadiendo elementos adicionales.
- Software gráfico e intuitivo sin necesidad de un alto grado de formación.
- Puestos de supervisión flexibles en número y ubicación ya que ambos softwares tienen opcionalmente una arquitectura cliente/servidor.

1.7.2.2 Inconvenientes

- Necesidad de dos softwares diferentes para la supervisión completa.
- Coste de licencias elevado dependiente del número de puestos de supervisión y el número de CDI a conectar.
- Coste de puestos de supervisión dedicados donde se instala el software cliente.
- En el caso de las Notifier ID60, necesidad de hardware adicional (TG-IP1-SEC) por cada CDI. Este módulo es un servidor de puerto RS232 para la conexión de centrales al software de gestión gráfica TG, mediante protocolo IP.
- Tiempo de ingeniería para el desarrollo del proyecto medio/alto, por la necesidad del aprendizaje y configuración del software.

1.7.3 Solución Basada en Software Gráfico de Terceros

Esta solución se basa en la instalación un software gráfico de terceros compatible con los dos modelos de CDI. Estudiando las posibilidades que existen en el mercado, solo se ha encontrado el *VIGIPlus* de *DESICO*.

1.7.3.1 Ventajas

- Infraestructura de comunicación basada en redes ethernet añadiendo elementos adicionales.
- Software gráfico e intuitivo sin necesidad de un alto grado de formación.
- Puestos de supervisión flexibles en número y ubicación.

1.7.3.2 Inconvenientes

- Coste de licencias dependiente del número de puestos de supervisión y el número de CDI a conectar.
- Coste de puestos de supervisión dedicados donde se instala el software cliente.
- En el caso de las Notifier ID60, necesidad de hardware adicional (TG-IP1-SEC) por cada CDI. Este módulo es un servidor de puerto RS232 para la conexión de centrales al software de gestión, mediante protocolo IP.
- En el caso de las ZP2, necesidad de adquirir una licencia adicional (2010-2-PAK-RMSDK) por cada CDI. Licencia de activación para habilitar la integración en plataformas externas a través de SDK vía Ethernet.
- Tiempo de ingeniería para el desarrollo del proyecto medio/alto, por la necesidad de del aprendizaje y configuración del software.

1.7.4 Solución Basada en la Integración con el Sistema de Control

Esta solución se basa en la integración de la supervisión con el sistema de control actual PCS7 de las plantas de producción.

1.7.4.1 Ventajas

- Infraestructura de comunicación basada en redes ethernet añadiendo elementos adicionales.
- Integración en el sistema de control PCS7 actual de las plantas con un grado de formación mínimo.
- Puestos de supervisión integrados en el propio sistema de control PCS7. La propia infraestructura de control constituida por estaciones cliente en salas de control, servicios de control vía web y pantallas de campo, brindan un despliegue importante de supervisión que garantiza una máxima atención permanente sin coste añadido.

1.7.4.2 Inconvenientes

- En el caso de las Notifier ID60, necesidad de hardware adicional (IBOX-MBS-NID3K) por cada CDI. Pasarela de comunicaciones que convierte el protocolo de las centrales a Modbus/TCP IP.
- En el caso de las ZP2, necesidad de adquirir una licencia adicional (2010-2-PAK-RMMB) por cada CDI. Licencia de activación de protocolo estándar Modbus/TCP.
- Costes añadidos derivados de la instalación de un PLC dedicado que actúe como enlace entre las CDI y el sistema de control.

- Tiempo de ingeniería para el desarrollo del proyecto alto, por la necesidad de la programación de librerías específicas.

1.7.5 Solución Adoptada

De las distintas alternativas que se han estudiado, y teniendo en cuenta los condicionantes de inicio, se llega a la conclusión de que la solución a adoptar estaría entre la propuesta basada en un software gráfico de terceros o la basada en la integración con el sistema de control.

Desde el punto de vista económico y el tiempo de ingeniería necesario para el desarrollo del proyecto, la primera opción sería la más adecuada. Aunque la inversión económica en licencias de software y elementos adicionales de la primera opción podría aproximarse a la inversión en un PLC dedicado de la segunda, el tiempo de desarrollo en esta última, es claramente mucho más alto.

Sin embargo, se estima que el tiempo de desarrollo inicial para crear las librerías del sistema de control se puede compensar en un futuro con la falta de necesidad en formación específica para la programación del software que se requiere en la primera. Esto brinda la posibilidad de que una empresa externa integradora de sistemas de control pueda hacer ampliaciones a futuro sin necesidad de esta formación específica.

Al margen del coste económico, el punto clave que inclina la balanza, es el hecho de garantizar una máxima atención permanente y, por lo tanto, decidir que la solución a adoptar sea finalmente la integración con el sistema de control.

Además, esta solución abre las puertas a la interacción entre los dos sistemas dándole valor añadido, y permitiendo que acciones o enclavamientos de seguridad del sistema de control estén supeditados a estados de alarma del sistema de detección de incendios.

De este último punto y a modo de ejemplo, se podría interrumpir de forma automática la carga de un producto controlada por el sistema de control cuando existe una alarma de incendios en la zona afectada.

1.8 Resultados Finales

De la solución adoptada con la integración en el sistema de control PCS7, abordaremos cada uno de los elementos que intervienen en el desarrollo del proyecto.

1.8.1 Infraestructura de Comunicación

Las instalaciones de IFF Benicarló cuentan con varias plantas de producción que históricamente han trabajado de manera independiente de forma que, cada una de ellas cuenta con un sistema de control aislado para actuar sobre los dispositivos de la propia planta. De manera excepcional hay interacciones puntuales entre el control de varias plantas por la necesidad de flexibilizar la producción y el almacenamiento. La evolución de los mercados y en continuo cambio en la demanda de productos hace cada día más patente la necesidad de buscar una solución para normalizar estas interacciones, buscando de este modo una infraestructura de comunicaciones global.

Esta infraestructura global, no solo permite abrir el aislamiento de las plantas, también permite la comunicación con otros sistemas como el de detección de incendios de este proyecto u otros en un futuro.

1.8.1.1 Direccionamiento de Red

La red implementada se basa en el protocolo Ethernet, un estándar de redes de área local (LAN) que controla el método con el que se comunican las computadoras y los dispositivos. El Instituto de Ingenieros Eléctricos y Electrónicos define Ethernet como el protocolo 802.3. Es un protocolo de enlace de datos de capa 2 que a menudo se usa como parte de la pila TCP/IP. TCP/IP reside en la capa 3 o 4, lo que hace de Ethernet la base para la comunicación mediante el protocolo TCP/IP. Los datos se transmiten en tramas Ethernet que la capa 3 de IP divide en paquetes de longitud variable.

En este tipo de redes, cada uno de los participantes tiene asignado una dirección IP que se compone de un llamado identificador de red para encontrar el camino de enrutamiento y de un identificador de dispositivo para llegar hasta un dispositivo concreto o *Host*.

En la actualidad existen dos versiones diferentes de direcciones, la IPv4 y la IPv6. La diferencia fundamental radica en el tamaño de la dirección y por lo tanto en el número de dispositivos posibles a direccionar. Al tratarse de una red de uso local y en previsión al número de dispositivos que podemos tener, la IPv4 es suficiente.

Las direcciones IPv4 están formadas por 32 bits y suelen representarse como una combinación de cuatro números decimales con valores entre 0 y 255 separados por puntos. En combinación a esta dirección se utiliza una *máscara de red*, que es una combinación de bits que sirve para delimitar el ámbito de la red. Su función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al Host.

Mediante la máscara de red, un sistema podrá saber si debe enviar un paquete dentro o fuera de la subred en la que está conectado. Por ejemplo, si el dispositivo tiene la dirección IP 192.168.1.1 y máscara de red 255.255.255.0, entiende que todo lo que se envía a una dirección IP con formato 192.168.1.x, se envía hacia la red local, mientras que direcciones con distinto formato de dirección IP serán enviadas hacia afuera.

Dentro de las posibilidades de direccionamiento, una red privada es una red que usa los espacios de direcciones IP especificadas en el documento RFC 1918, los cuales están reservados para la creación de redes de área local. Según este documento podemos encontrar estos rangos:

Bloque	Rango de direcciones IP	Nº de Redes	Nº de Hosts	Clase	Mayor CIDR (máscara de red)
24 bits	10.0.0.0 – 10.255.255.255	1	16.777.214	Clase A	10.0.0.0/8 (255.0.0.0)
20 bits	172.16.0.0 – 172.31.255.255	16	65.534	Clase B	172.16.0.0/12 (255.240.0.0)
16 bits	192.168.0.0 – 192.168.255.255	256	254	Clase C	192.168.0.0/16 (255.255.255.0)
16 bits	169.254.0.0 – 169.254.255.255	1	65.534	Clase B simple	169.254.0.0/16 (255.255.0.0)

Habiendo visto estas características y normas en el direccionamiento, podemos definir para el presente proyecto un rango lo suficientemente amplio con el siguiente criterio:

- Direccionamiento de red privada clase B usando una única red con rango 172.16.x.x.
- La red se divide en 4 subredes mediante la máscara 255.255.192.0 (CIDR/18)

Esto permite disponer de 4 subredes con un direccionamiento de 16382 Hosts por cada una de ellas, según la tabla:

Network	Host Min	Host Max	Broadcast
172.016.000.000/18	172.016.000.001	172.016.063.254	172.016.063.255
172.016.064.000/18	172.016.064.001	172.016.127.254	172.016.127.255
172.016.128.000/18	172.016.128.001	172.016.191.254	172.016.191.255
172.016.192.000/18	172.016.192.001	172.016.255.254	172.016.255.255

La red implementada y que denominamos **Plant Bus**, utiliza el primer rango de subred de la tabla anterior, dejando los tres restantes para futuras ampliaciones. En esta red, definimos el siguiente criterio de ordenación para la asignación de direcciones IP:

- El tercer octeto define la zona donde está instalado el dispositivo

Zona	Rango
Servicios Generales de Red	172.016.000.001 – 172.016.000.255
Planta Finos	172.016.001.001 – 172.016.001.255
Planta Acetatos	172.016.002.001 – 172.016.002.255
Planta Síntesis 1	172.016.003.001 – 172.016.003.255
Planta Síntesis 2	172.016.004.001 – 172.016.004.255
Planta Especialidades	172.016.005.001 – 172.016.005.255

Zona	Rango
Planta Hidrogenación	172.016.006.001 – 172.016.006.255
Planta Piloto	172.016.007.001 – 172.016.007.255
Depuradora	172.016.008.001 – 172.016.008.255
Planta Naturales	172.016.009.001 – 172.016.009.255
Planta Aguarrás	172.016.010.001 – 172.016.010.255
Edificio Laboratorios	172.016.011.001 – 172.016.011.255
Edificio Oficinas	172.016.012.001 – 172.016.012.255

Como excepción, se reserva el octeto 000 para dispositivos generales de red independientemente de su ubicación física.

- El cuarto octeto se subdivide en rangos para clasificar el tipo de dispositivo según su funcionalidad

Rango	Nº de Hosts	Tipo de dispositivo
172.016.xxx.001 – 172.016.xxx.032	32	Dispositivos gestores de red y comunicación tales como switches, routers, etc
172.016.xxx.033 – 172.016.xxx.064	32	Computadoras tales como servidores de control, estaciones de programación e Ingeniería, etc.
172.016.xxx.065 – 172.016.xxx.128	64	Dispositivos Master (Toman la iniciativa de comunicación) tales como PLCs, HMIs, etc.
172.016.xxx.129 – 172.016.xxx.255	127	Dispositivos Slave (Esperan las peticiones del Master) tales como periferia de I/O, servidores de datos, etc.

En el anexo 2.1 *Direccionamiento Plant Bus* podemos ver una tabla resumen de los dispositivos de red asociados a su dirección IP según el criterio explicado.

1.8.1.2 Arquitectura y Topología de Red

En el diseño de la arquitectura se buscan dos puntos clave: Un alto grado de disponibilidad y una disposición de conectividad redundante en dispositivos con doble puerto de comunicaciones.

Para alcanzar esto, nos basamos en la elección de la topología y el protocolo de enlace más adecuados, definiendo como topología de red, el mapa físico o lógico de una red para intercambiar datos. En otras palabras, es la forma en que está diseñada la red, sea en el plano físico o lógico. De esta forma y entre las topologías básicas, la más adecuada para conexión entre plantas es una **red en anillo** sustentada por el protocolo de enlace **MRP**.

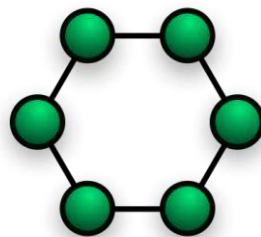


Figura 5. Topología de red en anillo

El protocolo MRP o *Media Redundancy Protocol* se describe en el estándar UNE-E N62439-2 y se utiliza en entornos de alta disponibilidad de redes industriales. Este protocolo opera en capa 2 del modelo OSI, siendo específico para redes en anillo que cuenten con hasta 50 equipos. De esta manera se garantiza siempre un comportamiento determinístico si se necesita realizar un cambio de estado.

Para que esto sea posible, se definen tiempos máximos en los que se deben recuperar las comunicaciones en función de la configuración elegida. Los tiempos tienen valores máximos de 10, 30 y 100 ms, siendo lo común que los cambios de ruta de los paquetes se den entre un 50 y un 75% más rápido.

Una parte importante del protocolo MRP son los nodos. Uno va a ser elegido MRM o *Media Redundancy Manager*, que bloqueará todo el reenvío de paquetes en uno de sus dos puertos de anillo designados (Fig. 6) para romper el bucle. El resto de nodos serán MRC o *Media Redundancy Client* que retransmiten las tramas de control enviadas por el MRM entre puertos para poder encontrar fallos si los hubiese y conectar el puerto bloqueado.

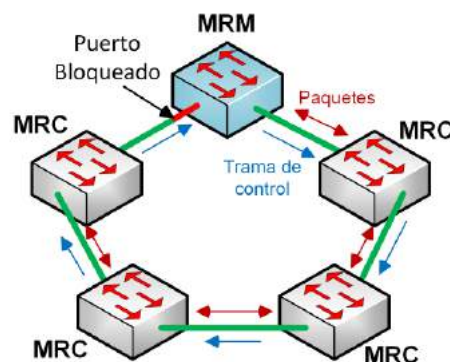


Figura 6. MRP en condiciones normales

El MRM del anillo se encuentra constantemente en contacto con sus pares MRC. Si uno de los dos vínculos de cualquier otro nodo de anillo detecta un fallo, el nodo del cliente enviará un mensaje al MRM de anillo que desbloqueará el puerto bloqueado permitiendo así que se vuelva a establecer la comunicación entre todos los nodos (Fig. 7).

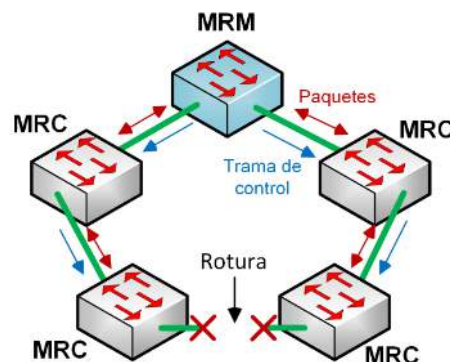


Figura 7. MRP en condiciones de fallo

La topología en anillo con MRP resuelve la disponibilidad entre plantas, pero fijándonos localmente en cada una de ellas, encontramos que esta no resuelve la conectividad redundante para dispositivos con doble puerto.

Las plantas con PCS7 tienen como base PLCs con CPU410-5H y tarjetas de comunicaciones CP443-1, las cuales disponen por cada una de sus líneas de comunicación de dos puertos con switches integrados (Fig. 8).

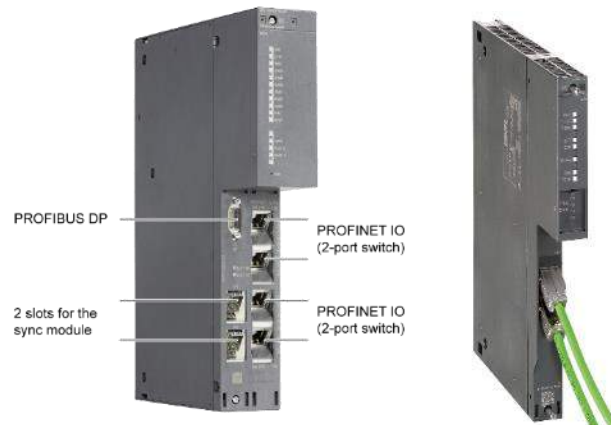


Figura 8. CPU410-5H y CP443-1

Esto permite la conexión de cada línea a dos nodos diferentes del anillo, permitiendo así un camino alternativo en el caso de producirse un fallo en uno de los nodos del MRP (Fig. 9).

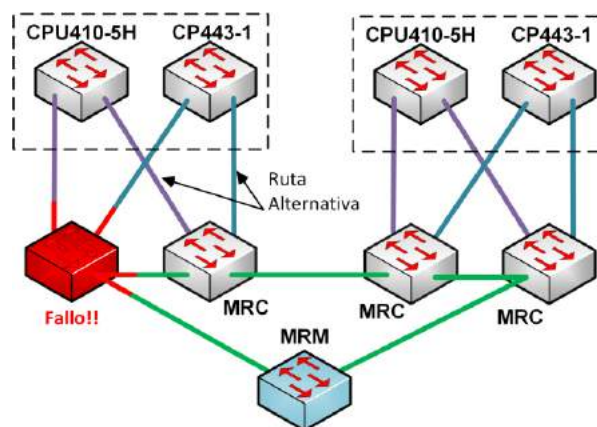


Figura 9. Redundancia en PLCs

Sin embargo, esta configuración presenta un problema ya que, al existir enlaces redundantes, cuando los switches envían paquetes broadcast (sin una dirección de destino) por todos sus puertos, estos se reenvían indefinidamente, generando *tormentas de broadcast* (Fig. 10). Las tormentas de broadcast generan un efecto “bola de nieve” que hacen que se consuma todo el ancho de banda y dejan la red inutilizable en poco tiempo.

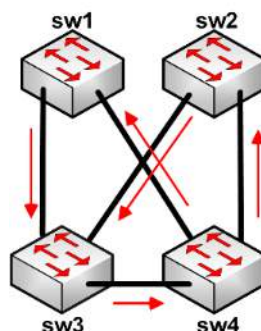


Figura 10. Bucles de red (tormenta de broadcast)

Este problema es muy común en redes complejas, y es por ello que se ha desarrollado el protocolo de enlace **RSTP**.

Rapid Spanning Tree Protocol (RSTP) es un protocolo de red que, al igual que el MRP, opera en capa 2 del modelo OSI y que gestiona enlaces redundantes. Especificado en IEEE 802.1w, es una evolución del Spanning Tree Protocol (STP), que reduce significativamente el tiempo de convergencia de la topología de red cuando ocurre un cambio en esta.

Su función es la de gestionar la presencia de bucles en topologías de red debido a la existencia de enlaces redundantes. El protocolo permite a los dispositivos de interconexión activar o desactivar automáticamente los enlaces de conexión, de forma que se garantice la eliminación de bucles. El algoritmo transforma una red física con forma de malla, por una red lógica en forma de árbol. Los puentes se comunican mediante mensajes de configuración llamados Bridge Protocol Data Units (BPDU).

El protocolo establece identificadores por puente y elige el que tiene la prioridad más alta, es decir, cada puerto tiene un parámetro configurable (*Span path cost*), y entre todos los puentes que conectan un segmento de red, se elige el de menor coste, bloqueando el resto de caminos. En el caso que haya el mismo coste en dos puentes, el protocolo elige el puente con la velocidad de transmisión más alta, el que tenga el menor identificador ("dirección MAC") y en último caso el número de puerto más bajo.

RSTP explora constantemente la red, de forma que cualquier fallo o adición en un enlace, es detectado al instante. Cuando cambia la topología de red, el RSTP reconfigura los puertos de los nodos para evitar una pérdida total de la conectividad (Fig. 11).

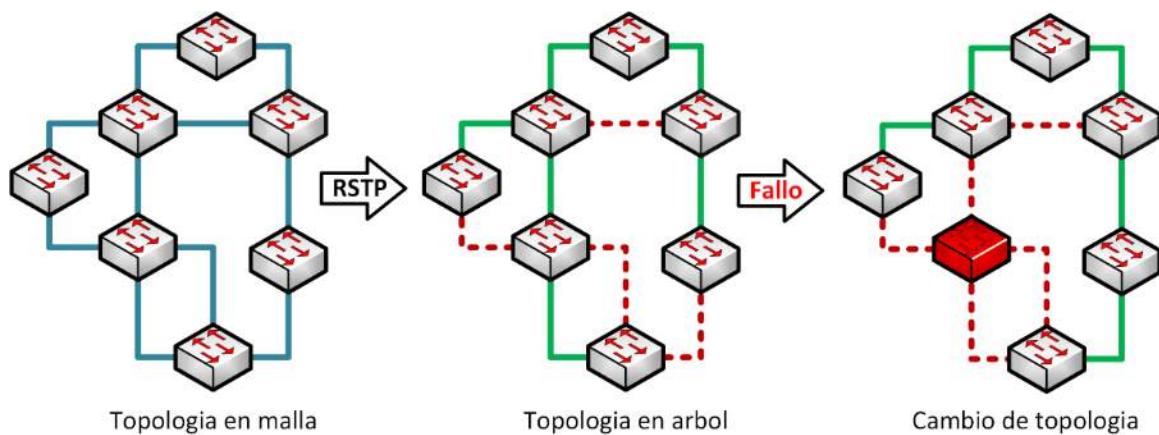


Figura 11. Rapid Spanning Tree Protocol (RSTP)

Podemos concluir finalmente que, para los enlaces entre plantas, el uso de MRP es el más adecuado por su pequeño tiempo de convergencia comparado con el RSTP. A nivel local, y dado que la red en malla a gestionar es muy pequeña, el RSTP es una buena solución, destacando la necesidad de instalar mínimo dos switches por planta para poder tener el doble enlace requerido.

1.8.1.3 Red de Fibra Óptica

Los enlaces entre plantas presentan dos impedimentos importantes a la hora de utilizar cable de cobre UTP/STP (cable de par trenzado) para las conexiones. En algunos casos, encontramos distancias de más de 200 m, cuando la distancia máxima recomendada es de 100 m. Además, el trazado atraviesa zonas de atmósfera explosiva ATEX que dificulta enormemente encontrar switches certificados para trabajar con cable de cobre en estas zonas. Es por eso que la instalación de fibra óptica es la opción más adecuada.

En la selección del tipo de fibra óptica se ha tenido en cuenta aspectos tales como distancias y velocidades de transmisión máximas según la tabla siguiente.

Distancias Max		Velocidad transmisión					
Tipo F.O.		100 Mb FX	1Gb SX	1Gb LX	10 Gb SR	40 Gb SR4	100 Gb SR10
Monomodo	OS2	2 Km	5Km	5Km	10Km	-	-
Multimodo	OM	2 Km	275m	550m (requiere acondici. del modo)	-	-	-
	OM	2 Km	550m		-	-	-
	OM	2 Km	550m		300m	100m	100m
	OM	2 Km	1 Km		400m	150m	150m
	OM	2 Km	1 Km		400m	400m	400m

Como veremos en el siguiente apartado, los transceptores de F.O. para el anillo MRP trabajan a velocidades de 1Gb/s. Además, cada una de las conexiones necesita un par de fibras ya que una trabaja como canal de transmisión y la otra de recepción. Teniendo en cuenta esto y en previsión a futuras ampliaciones tanto en la infraestructura como en el incremento de velocidad, finalmente se instala un **cable de 8 fibras Multimodo OM3 con grado de protección Gc (EPL)** según la norma UNE-EN 60079-1:2015.

Otro aspecto de la instalación de fibra óptica son las cajas de interconexión. En este caso se adopta el estándar de IFF, con cajas mural metálicas o *Patch panels* para armarios de Rack de 19” con conectores ST.



Figura 12. Conector de F.O. tipo ST (macho)

El trazado de la instalación de fibra óptica, así como la ubicación de las cajas de conexión quedan reflejados en el plano 3.14 *Trazado Fibra Óptica*.

1.8.1.4 Switches

Los dos puntos anteriores marcan los criterios a la hora de seleccionar el modelo de switch a instalar. Junto a estas características, y debido a que tendrán que trabajar en un entorno industrial, encontramos en la serie SCALANCE X de Siemens una opción válida.

Además, la elección de un Switch de Siemens garantiza la fácil integración con su sistema de control PCS7 al pertenecer al mismo fabricante.

Dentro de las posibilidades que ofrece Siemens tenemos varias gamas de switches, los cuales quedan clasificados en la tabla (Fig. 13).

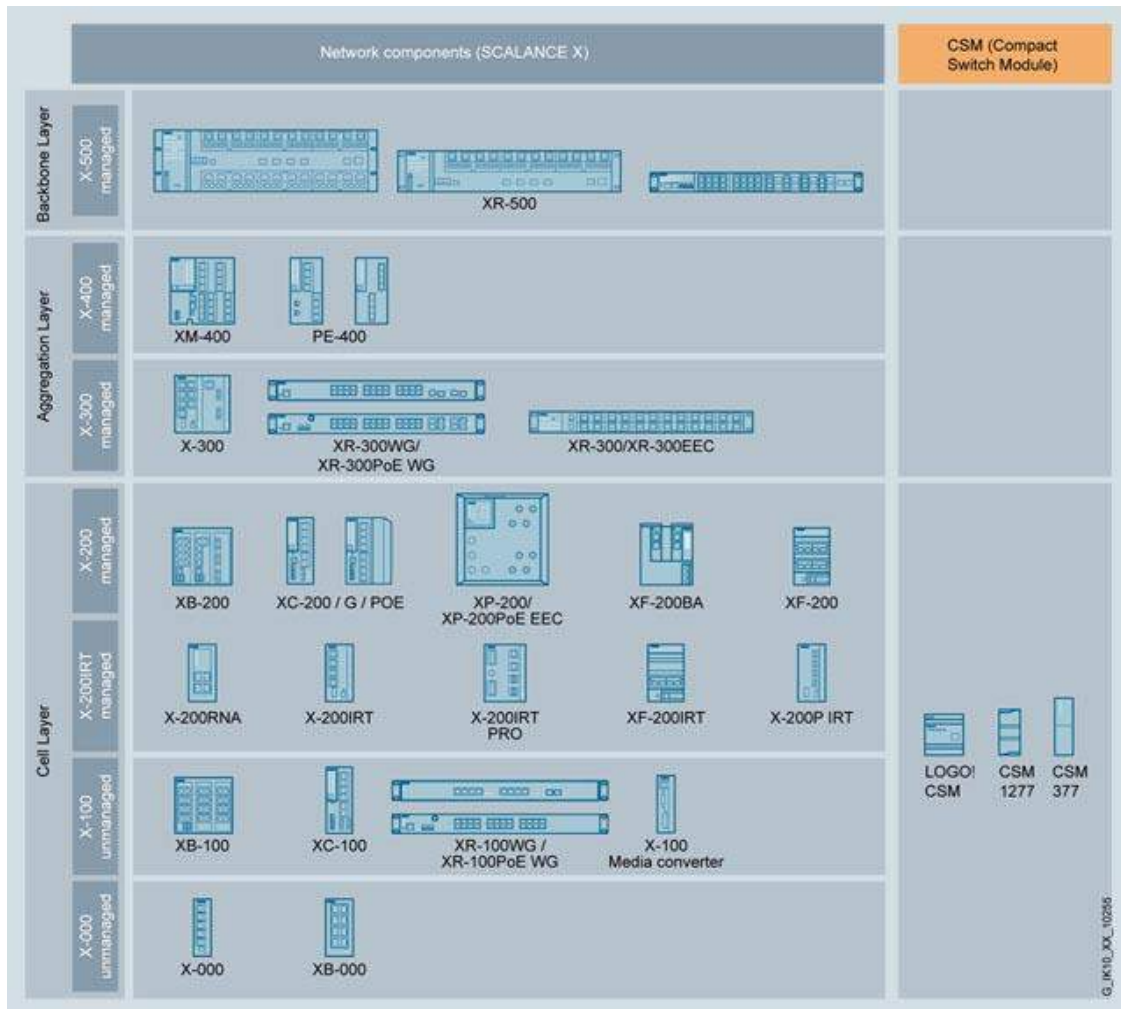


Figura 13. Sinopsis de la gama de switches, gestionados y no gestionados.

El campo de aplicación recomendado para su uso queda resumido en la tabla (Fig. 14)

	Areas of application / Type of networks / Requires	Office connection	Plant networking	Industry-related applications	Process automation	Power generation and distribution	Wind farms	Machinery and plant engineering	Unit networking	Standard mechanical engineering	Machine-internal networking	Network setup through SIMATIC S7-300, S7-1200 or LOGO!
X-500	Powerful backbone network with very high requirements on functionality / port density / availability as well as interface to Office IT	•	•	•								
X-400	Powerful plant network with high demand on functionality and availability	•	•	•								
X-300	Large networks with high demand on functionality and availability		•	•	•	•						
X-200	Networks with high demand on functionality and availability		•	•	•	•	•	•	•	•		
X-100	Networks with low demand on functionality						•	•	•	•		
X-000	Networks with low demand on functionality and ruggedness									•	•	
CSM	Networks or interface extension for SIMATIC S7-300, S7-1200, LOGO!											•

• applies to selected versions

Figura 14. Switches Industrial Ethernet SCALANCE X: Campos de aplicación.

Las posibilidades que ofrece Siemens en cuanto a las funciones de configuración de sus switches gestionables quedan reflejadas en la tabla (Fig. 15).

		SCALANCE ¹⁾												
		X-200						X-300			X-400	X-500		
		XB-200	XC-200	XP-200	XF-200	XF-200BA	X-200IRT	X-200RNA	X-300	XR-300WG	XR-300	XM-400	XR-500	
Software	Diagnostics	FO diagnostics		•		•			•		•	•	•	
		PROFINET	•	•	•	•	•	•	•	•	•	•	•	
		EtherNet/IP	•	•	•	•	•	•	•	•	•	•	•	
		Loop Detection	•	•	•	•	•	•	•	•	•	•	•	
		Port mirroring / RSPAN	•/-	•/-	•/-	•/-	•/-	•/-	•	•/-	•/-	•/-	•/-	•/-
		SNMP / Syslog	•	•	•	•	•	•	•	•	•	•	•	•
	Configuration	Web Interface, SNMP	•	•	•	•	•	•	•	•	•	•	•	
		TIA Portal	•	•	•	•	•	•	•	•	•	•	•	
		SINEC NMS	•	•	•	•	•	•	•	•	•	•	•	
		Command Line Interface (CLI)	•	•	•	•	•	•	•	•	•	•	•	
	Security	Access Control List (ACL)							•		•	•	•	
		IEEE 802.1X (RADIUS)	•	•	•		•		•	•	•	•	•	
		MAC authentication		•	•		•		•		•	•	•	
	Networking structuring	Virtual networks (VLAN) / Private VLAN	•/-	•/•	•/•		•/-		•/•	•/-	•/•	•/-	•/-	
		NAT/NAPT		•	•				•		•	•	•	
		IPv6 Routing										•	•	
		VRRP, OSPF, RIPv2										•	•	
		IGMP / PIM	•/-	•/-	•/-		•/-			•/-		•/•	•/•	
		Broad-/Multicast Blocking	•	•	•		•		•	•	•	•	•	
	Redundancy	Link Aggregation		•	•		•		•		•	•	•	
		HRP, MRP, STP/RSTP, Passive Listening	•	•	•	•	•	•	•	•	•	•	•	
		MSTP		•	•				•		•	•	•	
		HSR, PRP						•						
		MRP Interconnect / RSTP+	-/•	•/•	•/•		•/•			-/•		•/•	•/•	
	PROFINET	MRPD						•						
		Diagnostics	•	•	•	•	•	•	•	•	•	•	•	
		IRT						•						
	PA ready	H-Sync		•	•		•					•	•	
CiR/H-CiR			•	•		•					•	•		
S2 diagnostics			•	•		•					•	•		
Time synchronization	IEEE 1588 (PTP)							•		•	•	•		
	SNTP / NTP	•	•	•	•	•	•	•	•	•	•	•		
	secure NTP		•	•	•	•	•			•	•	•		

• applies

1) the unmanaged Switches are found in the function overview hardware

Figura 15. Switches Industrial Ethernet SCALANCE X: Sinopsis de funciones del software.

Fijándonos en el apartado de redundancia, necesitamos un modelo con capacidad RSTP+. RSTP+ se utiliza principalmente en la integración redundante de anillos MRP en una red RSTP. En principio sería posible administrar una red de este tipo exclusivamente con RSTP. Sin embargo, en una topología en anillo es más eficiente y rápido el método MRP. El método de redundancia de anillo MRP no se ve afectado por RSTP+, porque ambos métodos funcionan de forma independiente entre sí.

En principio, todos los dispositivos situados en los puntos de unión entre la red RSTP y el anillo MRP deben soportar el método RSTP+. Todos los demás dispositivos del anillo MRP deben reenviar BPDU (Bridge Protocol Data Unit).

La integración redundante de anillos MRP en una red RSTP no es posible sin RSTP+, porque el funcionamiento paralelo de RSTP y MRP en un puerto no está permitido. Solo deben soportar RSTP+ los dispositivos del anillo MRP que están conectados a la red RSTP. En el ejemplo de topología representado, son los dos dispositivos RSW1 y RSW4 (Fig. 16).

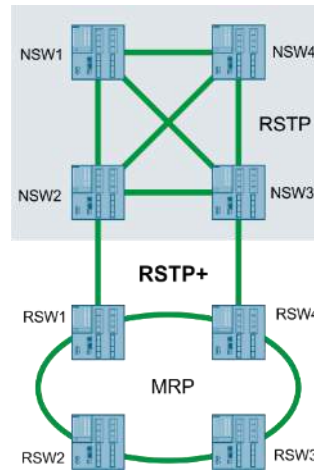


Figura 16. Red RSTP y anillo MRP

RSTP+ también se utiliza en la conexión de dos o más áreas de red RSTP a través de un anillo MRP. RSTP+ debe estar activado para todos los dispositivos del anillo MRP que están conectados a una de las redes RSTP. En el ejemplo representado, son los dispositivos RSW1, RSW2, RSW3 y RSW4 (Fig. 17).

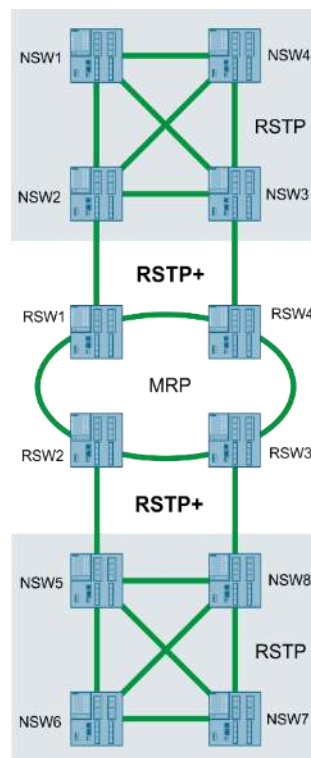


Figura 17. Anillo MRP con dos áreas de red RSTP.

RSTP+ también puede utilizarse para conectar entre sí varios anillos MRP a través de RSTP. En este caso, RSTP+ garantiza que MRP siga administrando la redundancia de anillo sin verse influido por RSTP (Fig. 18).

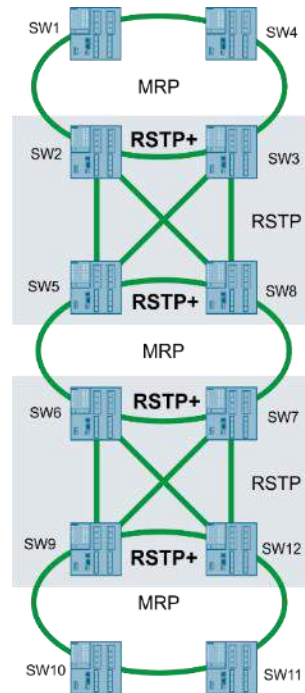


Figura 18. Anillos MRP a través de RSTP.

Por sus características, así como su relación versatilidad/precio, la serie XC-200 es la más adecuada para la instalación, siendo el modelo **SCALANCE XC206-2SFP** el elegido.

En el Anexo 2.2 *Switch SCALANCE XC206-2SFP*, se encuentra la hoja de datos del dispositivo con todas sus características, entre las que encontramos la disponibilidad de 2 puertos SFP para la conexión de transceptores a velocidades de hasta 1Gb/s (Fig. 19).



Figura 19. SCALANCE XC206-2SFP y transceptor SFP.

Estos puertos permiten conectar transceptores SFP de fibra óptica homologados según norma *UNE-EN60079-28:2016* (Atmósferas explosivas-Parte 28: Protección de material y sistemas de transmisión que utilizan radiación óptica), permitiendo que el trazado pueda atravesar zonas de atmósfera explosiva ATEX.

Dentro de la gama de transceptores SFP disponibles a 1Gb/s tenemos,

Modelo	Descripción	Referencia
SFP992-1	1 x 1000 Mbps, LC port optical for glass FO cable (multimode), up to max. 750 m	6GK5 992-1AL00-8AA0
SFP992-1(C)	1 x 1000 Mbps, LC port optical, for glass FO cable (multimode), up to max. 750 m, varnished	6GK5 992-1AL00-8FA0
SFP992-1+	1 x 1000 Mbps, LC port optical for glass FO cable (multimode), up to max. 2 km	6GK5 992-1AG00-8AA0
SFP992-1LD(C)	1 x 1000 Mbps LC port optical for glass FO cable (single mode) up to max. 10 km	6GK5 992-1AM00-8FA0
SFP992-1LD+	1 x 1000 Mbps LC port optical for glass FO cable (single mode) up to max. 30 km	6GK5 992-1AM30-8AA0
SFP992-1LH	1 x 1000 Mbps LC port optical for glass FO cable (single mode) up to max. 40 km	6GK5 992-1AN00-8AA0
SFP992-1LH+	1 x 1000 Mbps LC port optical for glass FO cable (single mode) up to max. 70 km	6GK5 992-1AP00-8AA0
SFP992-1ELH	1 x 1000 Mbps LC port optical for glass FO cable(single mode) up to max. 120 km	6GK5 992-1AQ00-8AA0

del que se ha elegido por sus características, el modelo **SFP992-1**. Ver Anexo 2.3 *Certificación Transceptor SFP992-1*.

El Switch cuenta con un servidor HTTP integrado para poder ser configurado fácilmente desde un explorador de internet. Por defecto, no tiene asignada ninguna dirección IP, por lo que será necesario el uso de un servidor DHCP para esta primera asignación.

El protocolo DHCP (*Dynamic Host Configuration Protocol*), es una extensión del protocolo *Bootstrap* (BOOTP) desarrollado en 1985 para conectar dispositivos como terminales y estaciones de trabajo sin disco duro con un Bootserver. Este protocolo de red se basa en un modelo cliente-servidor. En este escenario, un dispositivo host solicita en el momento de conectarse a una red la configuración IP a un servidor DHCP a través de los puertos UDP 67 y 68, que por su parte consulta en una base de datos las direcciones y parámetros de red asignables antes de dar una respuesta. Una vez realizada la consulta, el servidor envía los parámetros al cliente a través de la red.

Establecidos los parámetros de red en el Switch, el acceso por HTTP se hace posible, permitiendo dejar estos parámetros fijos con la desactivación del DHCP (Fig. 20).

① Opción del Menu Principal
 ② Configuración Dirección de Red
 ③ Desactivación DHCP

Figura 20. Configuración parámetros de red.

Para la configuración del protocolo RSTP+ deben observarse las reglas siguientes, que son válidas independientemente de una topología de red determinada:

- La función del administrador de redundancia no debe recaer en uno de los dos dispositivos del acoplamiento RSTP/MRP.
- Entre los dos puertos en anillo de los dispositivos acoplados debe haber una conexión LAN directa.

Para la configuración de redundancia de anillo MRP se deben seguir los siguientes pasos disponibles en el menú "Layer 2 > Ring Redundancy" (Fig.21):

- Activar la casilla "Ring Redundancy" para activar MRP.
- Asignar la función MRP. En la lista desplegable "Ring Redundancy Mode" debe seleccionarse la entrada "MRP Client" o bien "MRP Auto-Manager". La función del administrador de redundancia no debe recaer en uno de los dos dispositivos del acoplamiento RSTP/MRP.
- Definir los puertos en anillo seleccionando las entradas adecuadas para los puertos en anillo en las dos listas desplegables. En nuestro caso los puertos 7 y 8 pertenecientes a los transceptores SFP de fibra óptica.

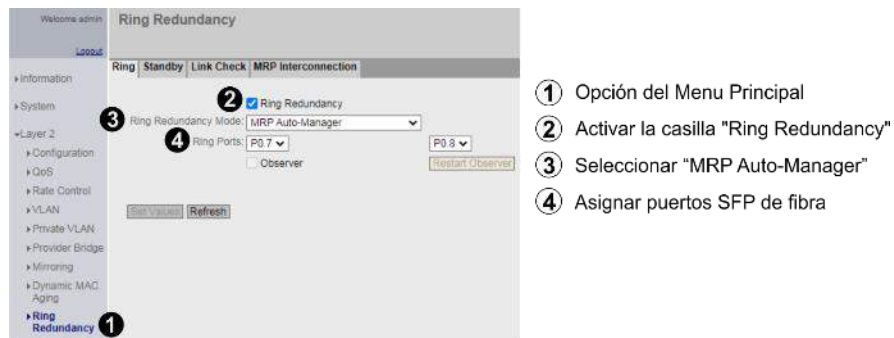


Figura 21. Configuración MRP.

Para la configuración de *Spanning Tree* para RSTP+ se deben seguir los siguientes pasos disponibles en el menú "Layer 2 > Spanning Tree" (Fig.22 y 23):

- Configurar los puertos en anillo en la ficha "CIST Port" desactivando las casillas de verificación de los dos puertos en anillo en la columna "Spanning Tree Status" de la tabla.
- Desactivar las casillas de verificación de los dos puertos en anillo en la columna "Restr. Role" de la tabla. Esto debe hacerse para que el comportamiento de los puertos en anillo sea controlado únicamente por MRP, es decir, por el administrador de redundancia.

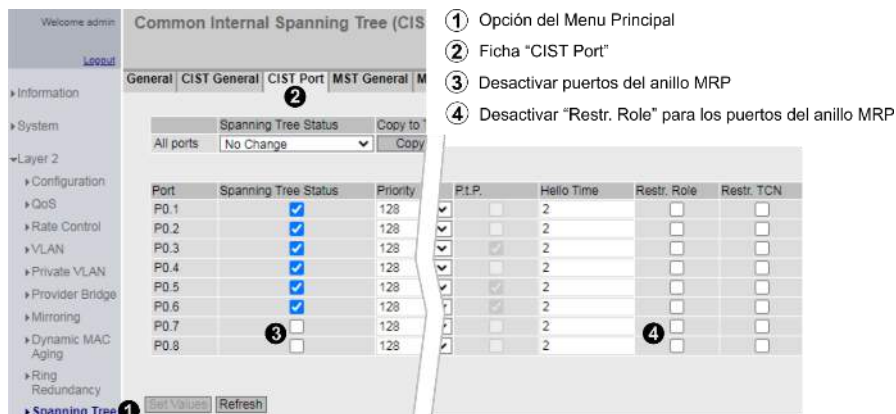


Figura 22. Configuración Spanning Tree

- Activar la casilla de verificación "*Spanning Tree*" y en la lista desplegable "*Protocol Compatibility*" seleccionar la entrada "RSTP".
- Definir la ID de dominio de la interconexión MRP para RSTP+. La ID de dominio de la interconexión MRP para RSTP+ debe ser unívoca en toda la red y debe ser distinta de una posible ID de dominio de la interconexión MRP que deba configurarse. Se requieren ID distintas para poder distinguir las TCN (*Topology Change Notifications*) de la red RSTP de las TCN del anillo MRP. En nuestro caso, la topología de red no presenta ninguna interconexión MRP, por lo que este valor es irrelevante.

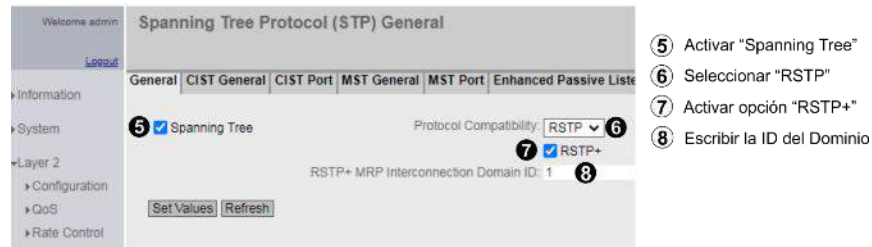


Figura 23. Configuración RSTP+

1.8.2 Arquitectura de Integración con el Sistema de Control

El sistema de control PCS7 en IFF, presenta la siguiente arquitectura estándar para cada una de las plantas de producción (Fig. 24).

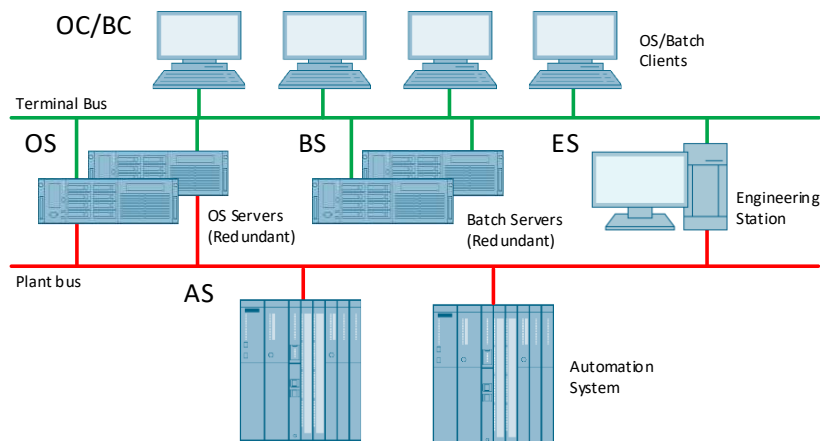


Figura 24. Arquitectura estándar plantas de producción IFF

En el nivel inferior encontramos los PLCs de control (AS) conectados de forma local a los servidores de operación (OS) y a la estación de Ingeniería (ES) a través de la red Plant Bus. Por un lado, los OS son los encargados de gestionar la parte gráfica del sistema permitiendo el flujo de información entre los clientes de operación (OC) y el AS. Por el otro, la estación de ingeniería (ES) es el centro de programación donde se encuentra el software necesario para la programación y configuración de todos los niveles del sistema y cuya información se almacena en un multiproyecto de PCS7. En este mismo nivel encontramos los servidores de Batch (BS) los cuales interactúan con los OS, encargándose de gestionar la parte productiva basada en la fabricación por lotes según norma ISA88.

Por último, en el nivel superior encontramos los clientes de operación y/o Batch (OC/BC). Estos clientes, que se comunican con los servidores a través de la red Terminal Bus existente, son la interfaz entre el sistema de control y los operarios de planta, permitiéndoles interactuar con el sistema.

Cada planta de producción cuenta con su propio sistema de control independiente del resto, es decir, existe un multiproyecto de PCS7 por cada una de ellas. Un multiproyecto de PCS7 es una distribución lógica del sistema formada por proyectos individuales, cada uno de los cuales tiene la información de un área de aplicación específica. Así, encontramos un proyecto AS donde se definen y programan los PLCs asociados, un proyecto OS donde se define y programa la parte gráfica del sistema, etc. El multiproyecto, al ser el nexo común de todos los proyectos, hace de puente entre la interrelación de cada uno de ellos permitiendo crear una estructura global de comunicaciones, arquitectura, jerarquía de planta, etc.

En el estudio de integración vemos dos flujos de datos a la hora de definir cuál es la mejor estrategia a seguir. En primer lugar, existe una comunicación entre las diferentes CDIs y el AS. Al no existir un multiproyecto común, integrar el sistema de monitorización en todas las plantas con PCS7 implica disponer de un proyecto de AS accesible desde todos los nodos, para que cualquier modificación sobre este, pueda propagarse en cada uno de los multiproyectos. Es por eso que este AS tiene que ser independiente, siendo recomendable la instalación de un nuevo PLC para separar la parte de monitorización de CDIs de la parte productiva.

El segundo flujo de datos lo encontramos entre el AS y los OS, ya que toda la información recogida por el PLC tiene que ser procesada y mostrada en los OC. Utilizar los OS de cada planta para este fin, implica replicar en todos ellos la configuración de conexión, la jerarquía de visualización, pantallas esquemáticas, etc. con todo el trabajo de desarrollo y mantenimiento que ello supone. Al igual que con el AS, la mejor opción es disponer de un OS independiente y aprovechar la característica de multicliente que ofrece PCS7 (Fig.25).

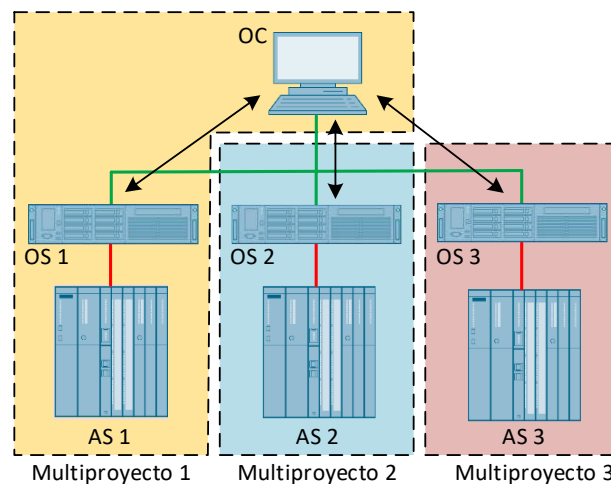


Figura 25. Sistema multicliente de PCS7

Mediante esta característica multicliente, los servidores OS y los sistemas de automatización AS se pueden distribuir en las respectivas zonas de las instalaciones utilizando un multiproyecto propio para cada una de ellas, pero si se tiene la necesidad de manejar varias zonas (servidor OS) desde un cliente, este cliente OC puede ser asignado a los multiproyectos servidor OS correspondientes.

Podemos concluir que la mejor estrategia a seguir para la implantación, es disponer de un PLC y un servidor OS dedicados a la monitorización del sistema de detección de incendios. Utilizando la nueva infraestructura de fibra óptica Plant Bus para enlazar este PLC con las diferentes CDIs y haciendo uso de la red *Terminal Bus* existente, para permitir que cualquier cliente OC pueda conectarse a este servidor OS dedicado.

De esta forma los clientes OC de las plantas de producción pueden mostrar no solo el proceso productivo de su propia planta, sino también, e integrado en la jerarquía de visualización, el sistema de detección de incendios (Fig.26).

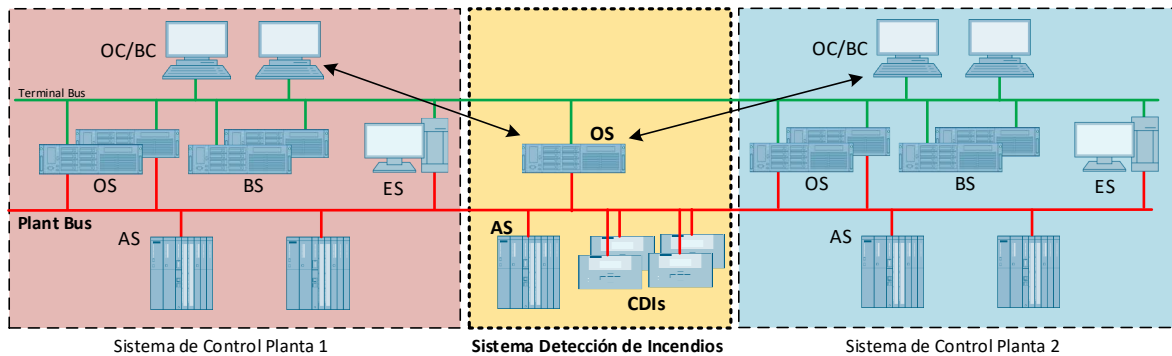


Figura 26. Arquitectura de Integración

La arquitectura general del sistema de control queda reflejada en el plano *3.15 Arquitectura del sistema de Control*.

1.8.2.1 PLC (AS)

El sistema de control PCS7 permite integrar de forma directa en su entorno de programación, CPUs de las gamas S7-400 y S7-300, siendo esta última la elegida por las características del proyecto y por su coste económico.

El PLC Siemens S7-300, ha sido desarrollado para dar solución a la automatización de rango medio. Se trata de un autómata modular que, si bien lleva mucho tiempo en el mercado demostrando su fiabilidad, mantiene renovaciones a nivel firmware, y esto lo hace una de las opciones más versátiles para la industria.

Este PLC cuenta con características entre las que podemos destacar:

- **Programación Modular:** Reduce costos tanto en ingeniería como en operación. De esta forma es posible reutilizar el programa y sus librerías.
- **Memoria Ampliable:** Utiliza una Micro Memory Card (MMC) para almacenar el programa, con la posibilidad adicional de almacenar datos y facilitar actualizaciones de firmware. Esto representa una ventaja para su mantenimiento, ya que la sustitución de la CPU solo requiere de la inserción de la MMC de PLC anterior. Además, se dispone de un catálogo de memorias MMC que van desde los 128 Kb hasta los 8 MB de almacenamiento, permitiendo ampliar sus capacidades de programa en función de las necesidades.
- **Gestión de Alarmas:** Dispone de un canal asíncrono para la notificación de alarmas que son gestionadas y almacenadas en un buffer de la propia CPU. Las alarmas son notificadas por evento a todos los sistemas de supervisión conectados permitiendo que, el reconocimiento de las mismas por parte de uno de estos sistemas, se propague al resto.
- **Puerto Ethernet Integrado:** Dispone de un puerto Ethernet integrado con posibilidad de 32 conexiones simultáneas.

Entre los modelos de la gama S7-300 se elige una **CPU 317-2 PN/DP** con una MMC de memoria de 4Mb, por sus características técnicas y porque actualmente ya hay instalado este modelo en otras partes de las instalaciones. Esto garantiza la disponibilidad de material de recambio en caso de avería, reduciendo el tiempo de inoperatividad del sistema. En el Anexo 2.4 *CPU 317-2 PN/DP* , se encuentra la hoja de datos del dispositivo con todas sus características.

Para las comunicaciones, se decide utilizar un módulo de comunicaciones **CP- 343-1** adicional separando el flujo de datos hacia las CDIs del flujo de datos hacia el servidor OS. Esto permite liberar conexiones del puerto ethernet de la propia CPU para futuras ampliaciones. En el Anexo 2.5 *CP 343-1* , se encuentra la hoja de datos del dispositivo con todas sus características.

El conjunto de CPU y CP, se alimenta por medio de un módulo **PS-307 (5 A)** de la misma serie. Esta fuente de alimentación que puede suministrar hasta 5 A, cubre sobradamente las necesidades de consumo. Para su selección se aplica el mismo criterio de disponibilidad de material de recambio. En el Anexo 2.6 *PS 307(5 A)* , se encuentra la hoja de datos del dispositivo con todas sus características.

1.8.2.2 Servidor OS virtualizado

La virtualización consiste en ejecutar, gracias a un software, una versión virtual o no física de un sistema operativo o una plataforma de hardware. Este software llamado *hipervisor* se conecta directamente con el hardware y permite dividir el sistema en entornos separados, distintos y seguros conocidos como "máquinas virtuales" (VM), simulando los recursos que tiene una máquina: CPU, RAM, Disco duro, Placa Base, red, y todo lo que compone un ordenador a partir de los recursos reales que tiene la máquina física. Es decir, la virtualización de servidores permite ejecutar múltiples sistemas operativos en un solo servidor físico (Fig.27).

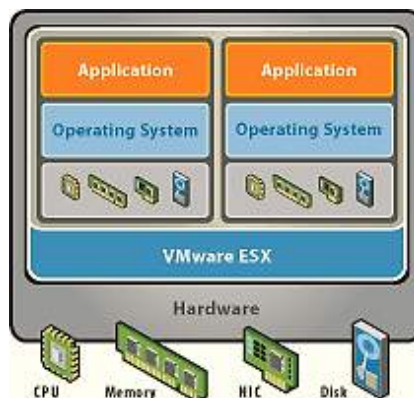


Figura 27. Hipervisor VMWare ESXi

El hecho de que IFF tenga disponible un servidor Dell Power Edge R640, con el software hipervisor VMWare ESXi 6.5, hace innecesaria la compra de un ordenador donde instalar el Servidor OS. Este servidor ESXi de gran capacidad, dispone de los recursos necesarios para albergar el Servidor OS en una nueva máquina virtual. Entre estos recursos destacar que dispone de 16 CPUs x Intel(R) Xeon(R) Gold 6134 CPU @ 3.20GHz, 128 GB de memoria RAM, 7 TB de capacidad de disco duro SSD en RAID 5.

El uso de la característica multicliente descrita anteriormente, requiere que las versiones de PCS7 entre el cliente OC y el servidor OS sean iguales, es por ello que la versión instalada sea PCS7 v9.0 SP3.

Los requerimientos de hardware recomendados según el manual de instalación para esta versión de Servidor OS son:

- Processor Min. INTEL Core 2 Duo; ≥ 2.4 GHz, INTEL Core 2 Quad
- Work memory (RAM) ≥ 8 GB (64-bit operating system)
- Hard disk Partition size ≥ 200 GB HDD/SSD C:\ 100 - 128 GB
- Network adapter (Terminal bus): RJ45 on-board gigabit Ethernet
- Network adapter (Plant bus): CP 1623 or BCE network adapter
- Opt. drive DVD

Por lo tanto, para la creación de la máquina virtual se utilizan los recursos de hardware del servidor hipervisor mostrados (Fig.28):

Hardware Configuration	
CPU	2 vCPUs
Memory	16 GB
Hard disk 1	100 GB
Hard disk 2	100 GB
USB controller	USB 2.0
Network adapter 1	prd-cluster01_Vlan_203 (Connected)
Network adapter 2	PlantBus (Connected)
Video card	36 MB
CD/DVD drive 1	ATAPI
Others	Additional Hardware

Figura 28. Recursos VM Servidor OS

Utilizando el primer disco para la partición c:\ donde se instalará el sistema operativo y el software de OS, y el segundo para la partición d:\ reservado para guardar ficheros de configuración, datos, etc. Este segundo disco virtual se configura en modo de *aprovisionamiento thin* por el cual, consume solo el espacio que necesita inicialmente y crece con el tiempo según la demanda, permitiendo ahorrar espacio de almacenamiento físico.

El software PCS7 requiere de la instalación de una serie de licencias según el tipo de producto instalado (Servidor OS, Cliente OC, etc.) y el número de elementos a visualizar y/o programar. Para estos elementos denominados “*Process Objects*” (PO) existen paquetes de licencia por número de objetos permitidos para la programación del PLC (AS RT PO) y paquetes de licencia por número de objetos para la visualización en el servidor OS (OS RT PO). El uso de un PLC S-300 no requiere del primer licenciamiento, ya que esta gama de PLCs no admite esta característica. Sin embargo, para las licencias OS RT PO, podemos estimar que por cada elemento a monitorizar dentro del sistema de detección de incendios (CDI, Detector de humo, pulsador de alarma, etc.) usaremos 1 OS RT PO. Teniendo en cuenta que el número total de elementos del sistema está entorno a los 400, y haciendo una previsión de futuras ampliaciones, se instala un paquete de 1000 OS RT PO.

Para las comunicaciones entre el servidor OS y el PLC, cuando se utilizan tarjetas de red específicas de Siemens (SIMATIC NET CP1623) no se requiere ningún tipo de licencia, sin embargo, al utilizar una máquina virtual, el driver de este tipo de tarjetas no está soportado y es necesario utilizar una tarjeta de red virtual estándar. El uso de tarjetas ethernet estándar requiere de la licencia adicional SIMATIC NET BCE.

A modo de resumen, la siguiente tabla muestra las características del Servidor OS:

Operating System	Windows Server 2016 Standard (64-Bit)
Host Name	ESBEP7OS50
OS Software	PCS7 v9.0 SP3
Network Adapter 1 (Terminal Bus)	10.8.8.35
Network Adapter 2 (Plant Bus)	172.16.12.33
PCS7 Licenses	PCS7 OS Server BCE 1000 OS RT POs

1.8.3 Protocolo MODBUS-TCP

El protocolo Modbus fue desarrollado en 1979 por Modicon para sistemas de automatización industrial y controladores programables Modicon. Desde entonces se ha convertido en un método estándar de la industria para la transferencia de información.

Modbus es un protocolo de mensajería que opera en capa 7 del modelo OSI (capa de aplicación), y que proporciona la comunicación entre dispositivos conectados en diferentes tipos de buses o redes.

Los dispositivos Modbus se comunican mediante una técnica maestro-esclavo (cliente-servidor), técnica en la que solo un dispositivo (maestro/cliente) puede iniciar las transacciones (llamadas consultas). Los demás dispositivos (esclavos/servidores) responden proporcionando los datos solicitados al maestro, o tomando la acción solicitada en la consulta.

Modbus TCP/IP (denominado Modbus-TCP) es esencialmente el protocolo Modbus RTU, que nació originalmente como una interfaz de comunicaciones serie, encapsulado en una envoltura Ethernet TCP/IP.

TCP/IP se refiere al Protocolo de control de transmisión y al Protocolo de Internet, que proporciona el medio de transmisión para la mensajería Modbus TCP/IP. La función principal de TCP es asegurarse de que todos los paquetes de datos se reciban correctamente, mientras que IP se asegura que los mensajes se dirijan y enruten correctamente. Es decir, la combinación TCP/IP es simplemente un protocolo de transporte, y no define el significado de los datos o cómo se interpretan.

La estructura de mensajería Modbus es el protocolo de aplicación que define la reglas para organizar e interpretar los datos independientemente del medio de transmisión.

En resumen, Modbus TCP/IP combina una red física (Ethernet), con una red estándar (TCP/IP) y un método estándar para representar datos (Modbus como el protocolo de aplicación), utilizando como estándar el puerto 502 de la pila TCP/IP.

En la práctica, Modbus TCP incorpora una estructura de datos Modbus estándar en una trama TCP como se muestra en el siguiente diagrama (Fig.29):

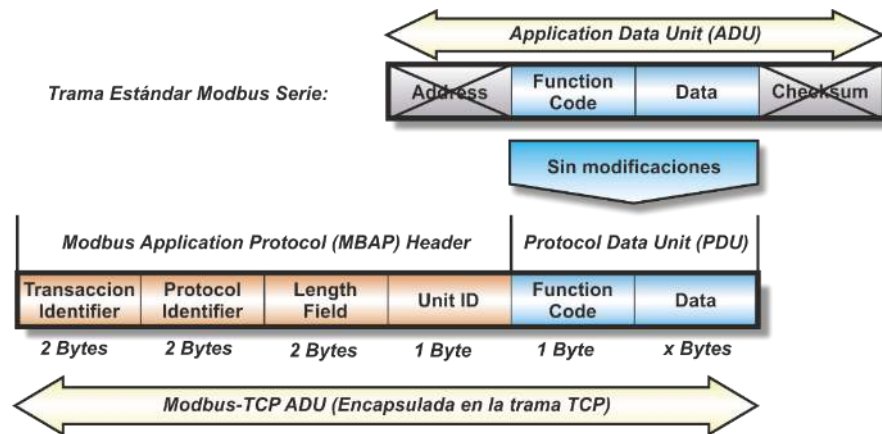


Figura 29. Estructura trama Modbus-TCP

Los comandos Modbus y los datos de usuario están encapsulados en el contenedor de datos de un telegrama TCP/IP sin modificaciones. Sin embargo, el campo de comprobación de errores de Modbus (*Checksum*) no se utiliza, ya que los métodos de suma de comprobación de la capa de enlace Ethernet TCP/IP estándar garantizan la integridad de los datos. Además, el campo de dirección de la trama Modbus es reemplazado por el identificador de la unidad en Modbus TCP/IP, y se convierte en parte del encabezado del protocolo de aplicación Modbus (MBAP).

En la figura 29, vemos que el código de función y los campos de datos son tomados en su forma original. Así, una unidad de datos de aplicación (ADU) para Modbus-TCP toma la forma de un encabezado formado por un identificador de transacción, un identificador de protocolo, una longitud y un identificador de unidad, junto a la unidad de datos del protocolo (PDU) formado por el código de función y los datos. El encabezado MBAP tiene 7 bytes de longitud e incluye los siguientes campos:

- *Identificador de Transacción (2 Bytes)*: Este campo se utiliza para el emparejamiento de transacciones cuando un cliente envía múltiples mensajes por la misma conexión TCP sin esperar una respuesta previa.
- *Identificador de protocolo (2 Bytes)*: Este campo siempre es 0 para los servicios de Modbus y se reservan otros valores para futuras ampliaciones.
- *Longitud (2 Bytes)*: Este campo es un recuento de bytes de los campos restantes incluyendo el byte del identificador de la unidad, el byte del código de función y los datos.
- *Identificador de unidad (1 Byte)*: Este campo se utiliza para identificar un servidor ubicado en una red que no sea TCP/IP (para puertas de enlace TCP/Serie). En una aplicación típica Modbus-TCP, el ID de unidad se establece en 00 o FF, siendo ignorado por el servidor y simplemente repetido en la respuesta.

Para la PDU, el protocolo Modbus define tres tipos: De solicitud, de respuesta y de respuesta de excepción. La solicitud de servicio se compone de un código de función, y algún número de bytes de datos adicionales dependiendo de la función. En la mayoría de los casos, los datos adicionales suelen ser una referencia, como una dirección de registro, ya que la mayoría de las funciones Modbus operan en registros de memoria para configurar, monitorizar y controlar las I/O del dispositivo. Estos dispositivos suelen incluir un mapa de registros que debe ser consultado para obtener una mejor comprensión de su funcionamiento.

El direccionamiento y los datos utilizan una configuración “Big-Endian”. Esto significa que cuando se transmite un dato numérico mayor de 1 Byte, el Byte más significativo es el que se transmite primero. Así, por ejemplo:

<u>Tamaño</u>	<u>Valor</u>	<u>Big-Endian</u>
16-bits	0x1234	Primer Byte 0x12, segundo Byte 0x34

El tamaño de la PDU está limitado por la restricción de tamaño heredada de la primera implementación Modbus en las redes de línea serie (RS485 ADU máx. = 256 bytes):

- PDU comunicaciones serie = 256 - Dirección (1 byte) - CRC (2 bytes) = 253 bytes.

Como consecuencia:

- Modbus-TCP ADU = 253 bytes + MBAP (7 bytes) = 260 bytes.

Los registros Modbus de un dispositivo se organizan entorno a cuatro tipos básicos de datos, identificándose, además, por el número inicial de la dirección de referencia de la siguiente forma:

Ref.	Tipo	Tamaño	Acceso	Descripción
0xxxx	Salidas Digitales (Coils)	1 bit	R/W	Se utiliza una dirección de referencia 0x para enviar datos de salida a un canal de salida digital.
1xxxx	Entradas Digitales	1 bit	R	El estado ON/OFF de una dirección de referencia 1x está controlado por el canal de entrada digital correspondiente.
3xxxx	Registros de Entrada	1 Word (16 bits)	R	Un registro 3x contiene un número de 16 bits recibido de una fuente externa, como por ejemplo una señal analógica.
4xxxx	Registros de Salida (Holding Reg.)	1 Word (16 bits)	R/W	Un registro 4x se utiliza para almacenar 16 bits de un dato numérico, o para enviar el dato desde la CPU a un canal de salida.

Las "x" que siguen al dígito principal representa una ubicación de dirección Hexadecimal de cuatro cifras en la memoria de datos de usuario (0 a 65535 en decimal). El dígito principal, generalmente está implícito en el código de función y se omite del especificador de dirección para una función determinada.

Todas las direcciones de datos en los mensajes Modbus se referencian a 0, con la primera ocurrencia de un elemento de datos direccionado como elemento número 1. Por ejemplo, el registro 40001 se direcciona como registro 0000 en el campo de dirección de datos del mensaje. El código de función que opera en este registro especifica una operación de "registro de salida" donde el grupo de referencia “4xxxx” está implícito. Por lo tanto, el registro 40108 se direcciona realmente como registro 0x006B (107 decimal).

En una consulta, el cliente envía una trama (ADU) de solicitud. Esta trama es recibida por el servidor y procesada siguiendo el diagrama de flujo de la figura 30. Una vez que la solicitud ha sido procesada por el servidor, se construye una respuesta que es retornada al cliente.

Dependiendo del resultado del procesamiento se construyen dos tipos de respuesta: Una respuesta positiva donde la PDU contendrá el código de función de respuesta igual al código de función de solicitud, más los datos solicitados. O una respuesta de excepción cuyo objetivo es proporcionar al cliente información relevante sobre el error detectado durante el procesamiento, y donde la PDU contendrá el código de función igual al código de función de solicitud + 0x80, más un código de excepción indicando el motivo del error.

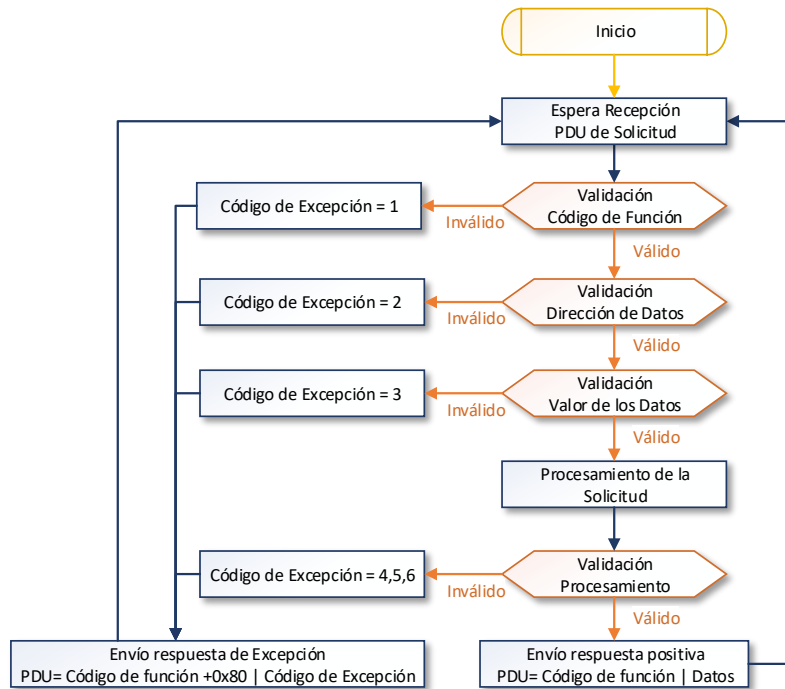


Figura 30. diagrama de procesamiento de una PDU de respuesta.

De los posibles códigos de excepción tendremos:

- **FUNCION ILEGAL (0x01)**

El código de función recibido en la consulta no es una acción permitida por el servidor. Esto puede deberse a que el código de función solo se aplica a los dispositivos más nuevos y no se implementó en la unidad seleccionada. También podría indicar que el servidor está en un estado incorrecto para procesar una solicitud de este tipo, por ejemplo, porque no está configurado y se le pide que devuelva valores de registro.

- **DIRECCIÓN DE DATOS ILEGALES (0x02)**

La dirección de datos recibida en la consulta no es una dirección permitida por el servidor. Es decir, la combinación de número de referencia y longitud de transferencia no es válida. Para un controlador con 100 registros, la PDU direcciona el primer registro como 0 y el último como 99. Si se envía una solicitud con una dirección de registro inicial de 96 y una cantidad de registros de 4, esta solicitud funcionará correctamente en los registros 96, 97, 98, 99. Si se envía una solicitud con una dirección de registro inicial de 96 y una cantidad de registros de 5, esta solicitud fallará con el código de excepción 0x02 ya que intenta operar en los registros 96, 97, 98, 99 y 100, y no existe ningún registro con dirección 100.

- **VALOR DE DATOS ILEGALES (0x03)**

Un valor contenido en el campo de datos de la consulta no es un valor permitido por el servidor. Esto indica una falla en la estructura de la solicitud, como que la longitud implícita es incorrecta o que el número de datos solicitados excede al máximo permitido. Si se envía

una solicitud de lectura de 126 registros, la solicitud fallará con el código de excepción 0x03 ya que el máximo permitido es de 125 derivado del tamaño máximo de la PDU.

- **FALLO DEL SERVIDOR (0x04)**

Se produjo un error irrecuperable mientras el servidor intentaba realizar la acción solicitada.

- **RECONOCIMIENTO (0X05)**

El servidor ha aceptado la solicitud y la está procesando, pero será necesario un período de tiempo prolongado para hacerlo. Esta respuesta se devuelve para evitar que se produzca un error de tiempo de espera en el cliente. Posteriormente, el cliente puede emitir un mensaje de “Encuesta de procesamiento completado” para determinar si se completó el procesamiento.

- **DISPOSITIVO SERVIDOR OCUPADO (0X06)**

El servidor está procesando un comando de programa de larga duración. El cliente debe retransmitir el mensaje más tarde cuando el servidor esté libre.

- **ERROR DE PARIDAD DE MEMORIA (0X08)**

El servidor intentó leer el archivo de registro, pero detectó un error de paridad en la memoria. El cliente puede volver a intentar la solicitud

- **RUTA DE GATEWAY NO DISPONIBLE (0x0A)**

El uso especializado junto con las puertas de enlace indica que la puerta de enlace no pudo asignar una ruta de comunicación interna desde el puerto de entrada al puerto de salida para procesar la solicitud. Por lo general, significa que la puerta de enlace está mal configurada o sobrecargada.

- **EL DISPOSITIVO GATEWAY NO RESPONDE (0x0B)**

El uso especializado junto con pasarelas indica que no se obtuvo respuesta del dispositivo de destino. Por lo general, significa que el dispositivo no está presente en la red.

Para el código de función, aunque el protocolo Modbus define tres categorías, nos centraremos en los Códigos de Función Pública para acceso a registros. Estos códigos están validados por la comunidad Modbus, garantizando su definición y su exclusividad.

Tamaño	Tipo	Acceso	Código	Hex
Bit	Entradas Digitales (1x)	Lectura Múltiple	02	0x02
		Lectura Múltiple	01	0x01
	Salidas Digitales (0x)	Escritura Simple	05	0x05
		Escritura Múltiple	15	0x0F
Word (16 Bits)	Registros de Entrada (3x)	Lectura Múltiple	04	0x04
	Registros de Salida (4x)	Lectura Múltiple	03	0x03
		Escritura Simple	06	0x06
		Escritura Múltiple	16	0x10
		Lectura/Escritura Múltiple	23	0x17
		Escritura con máscara	22	0x16
		Lectura cola FIFO	24	0x18

A su vez, y para no extendernos demasiado, solo se describe en detalle el uso de los códigos de función coloreados en la tabla anterior.

1.8.3.1 Lectura de Salidas o Entradas Digitales (0x01/0x02)

Estos códigos de función se utilizan para leer de 1 a 2000 estados contiguos de salidas o entradas digitales. La PDU de solicitud especifica la dirección del primer valor digital y el número de valores. Los estados en el mensaje de respuesta se empaquetan uno por cada bit. El bit de menor peso (LSB) del primer Byte de datos contiene el estado del valor digital direccionado en la consulta. Los otros estados siguen hacia el extremo de orden superior de este Byte, y de orden inferior a orden superior en Bytes posteriores. Si la cantidad de estados devuelta no es un múltiplo de ocho, los bits restantes en el Byte de datos final se rellenan con ceros (hacia el extremo de orden superior del Byte).

Solicitud

Function Code	1 Byte	0x01/0x02
Starting Address	2 Bytes	0x0000 a 0xFFFF
Quantity	2 Bytes	1 a 2000 (0x07D0)

Respuesta positiva

Function Code	1 Byte	0x01/0x02
Byte Count	1 Byte	N
Data	N Bytes	N=RoundUp(Quantity/8)

Ejemplo

Lectura de las salidas digitales 20 a 37 (18 entradas):

Solicitud		Respuesta	
Campo	Hex	Campo	Hex
Function Code	0x01	Function Code	0x01
Starting Address Hi	0x00	Byte Count	0x03
Starting Address Lo	0x13	Outputs Status 27-20	0xCD
Quantity Hi	0x00	Outputs Status 35-28	0x6B
Quantity Lo	0x12	Outputs Status 37-36	0x03

1.8.3.2 Lectura de Registros de Salida o Entrada (0x03/0x04)

Estos códigos de función se utilizan para leer de 1 a 125 registros contiguos de salida o entrada. La PDU de solicitud especifica la dirección del registro inicial y el número de registros. Los datos del registro en el mensaje de respuesta se empaquetan como dos bytes por registro, con el contenido binario justificado a la derecha dentro de cada byte. Para cada registro, el primer Byte contiene los bits de orden superior y el segundo contiene los bits de orden inferior.

Solicitud

Function Code	1 Byte	0x03/0x04
Starting Address	2 Bytes	0x0000 a 0xFFFF
Quantity	2 Bytes	1 a 125 (0x007D)

Respuesta positiva

Function Code	1 Byte	0x03/0x04
Byte Count	1 Byte	N
Data	N Bytes	N=Quantity*2

Ejemplo

Lectura de los registros de entrada 9 y 10:

Solicitud		Respuesta	
Campo	(Hex)	Campo	(Hex)
Function Code	0x04	Function Code	0x04
Starting Address Hi	0x00	Byte Count	0x04
Starting Address Lo	0x08	Input Reg. 9 Hi	0x12
Quantity Hi	0x00	Input Reg. 9 Lo	0x34
Quantity Lo	0x02	Input Reg. 10 Hi	0x56
		Input Reg. 10 Lo	0x78

1.8.3.3 Escritura de Salida Digital (0x05)

Este código de función se utiliza para escribir una sola salida digital en ON o OFF. La PDU de solicitud especifica la dirección de la salida que se va a escribir y un valor 0xFF00 para activarla (ON) o 0x0000 para apagarla (OFF). Los demás valores son ilegales y no afectan a la salida.

Solicitud

Function Code	1 Byte	0x05
Output Address	2 Bytes	0x0000 a 0xFFFF
Output Value	2 Bytes	0x0000 o 0xFF00 (OFF/ON)

Respuesta positiva

Function Code	1 Byte	0x05
Output Address	2 Byte	0x0000 a 0xFFFF
Output Value	2 Bytes	0x0000 o 0xFF00 (OFF/ON)

Ejemplo

Escritura a ON de la salida digital 173:

Solicitud		Respuesta	
Campo	Hex	Campo	Hex
Function Code	0x05	Function Code	0x05
Output Address Hi	0x00	Byte Count	0x00
Output Address Lo	0xAC	Outputs Status 27-20	0xAC
Output Value Hi	0xFF	Outputs Status 35-28	0xFF
Output Value Lo	0x00	Outputs Status 37-36	0x00

1.8.3.4 Escritura de Registro de Salida (0x06)

Este código de función se utiliza para escribir un solo registro de salida. La PDU de solicitud especifica la dirección del registro que se va a escribir y el valor de dicho registro.

Solicitud

Function Code	1 Byte	0x06
Register Address	2 Bytes	0x0000 a 0xFFFF
Register Value	2 Bytes	0x0000 a 0xFFFF

Respuesta positiva

Function Code	1 Byte	0x06
Register Address	2 Bytes	0x0000 a 0xFFFF
Register Value	2 Bytes	0x0000 a 0xFFFF

Ejemplo

Escritura del registro de la salida 2 con un valor de 10 (decimal):

Solicitud		Respuesta	
Campo	Hex	Campo	Hex
Function Code	0x06	Function Code	0x06
Register Address Hi	0x00	Register Address Hi	0x00
Register Address Lo	0x01	Register Address Lo	0x01
Register Value Hi	0x00	Register Value Hi	0x00
Register Value Lo	0x0A	Register Value Lo	0x0A

1.8.4 Conceptos Básicos de los Sistemas de Detección de Incendios

Básicamente cualquier sistema de detección de incendios está formado por los siguientes componentes:

- Unidad de Control o Central Detectora de Incendios (CDI).
- Equipos de detección (Detectores de humo, etc).
- Accionamientos manuales (Pulsadores de emergencia, etc).
- Señalización (Sirenas, etc).
- Líneas de conexión.
- Alimentación (Energía y Baterías de Backup).
- Elementos auxiliares (Módulos de entradas y salidas para censado y control opcional de otros equipos y sistemas).

Entre los diferentes tipos de sistemas, destacaremos los analógicos direccionables por ser los instalados en IFF Benicarló. Estos sistemas, comúnmente denominados sistemas inteligentes, se diferencian principalmente de los sistemas convencionales por su capacidad de identificación del punto exacto de una posible incidencia y por sus amplias posibilidades de configuración y programación.

El corazón de los sistemas analógicos direccionables es la central de detección de incendios (CDI), que es dónde se llevarán a cabo la mayoría de las opciones de programación y configuración del sistema. La CDI normalmente está compuesta por una fuente de alimentación, una placa base con su CPU y sus tarjetas de lazo, así como una fuente de alimentación con cargador de baterías.

Su instalación, en lugar de ramales, utiliza un número determinado de lazos (Fig.31) dependiendo del fabricante y del modelo de central. Cada lazo se compone de un bus de comunicaciones de lectura bidireccional, normalmente a dos hilos, donde se intercalan los dispositivos de campo. Estos dispositivos tienen asignada una dirección única programable que permite su identificación por separado ante una situación de alarma o fallo teniendo, además, la posibilidad de recibir y ajustar parámetros específicos de cada uno de manera remota. Esta identificación hace que quede reflejado cada equipo de forma unívoca en la central, lo que permite etiquetar cada elemento por su posición y/o Zona.

Esto significa que si, por ejemplo, en la habitación 207 de un hotel instalamos el detector N°52, en la central podremos asignar al detector 52 la etiqueta "habitación 207". Lo que provocará una respuesta más rápida y eficaz por parte del personal de seguridad, ya que se indica el lugar exacto de la alarma.

Además de este etiquetado, este tipo de centrales permite realizar una serie de programaciones de acción y respuesta ante alarmas. Volviendo al ejemplo del hotel, si la planta 2ª tiene asignada la zona 2, se puede programar que la central de detección actúe solo sobre los retenedores de las puertas cortafuego de esa zona, para sectorizar el hotel, o que sólo se disparen las sirenas de esa planta para evacuar, pero haya un retardo antes de evacuar el resto del hotel. Todo este potencial implica un trabajo de programación con el software propio de la central.

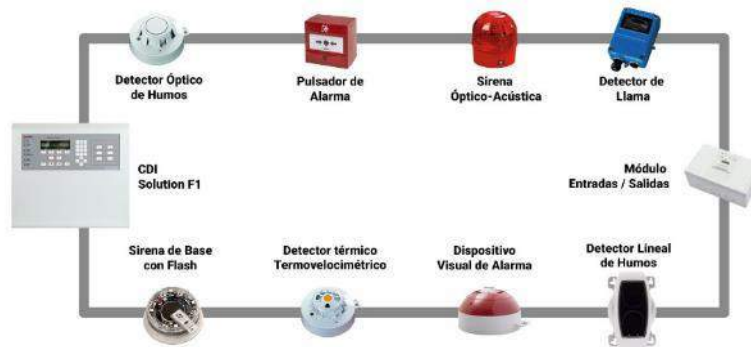


Figura 31. Ejemplo de lazo cerrado de un Sistema Detección de Incendios

1.8.5 Comunicaciones con Notifier ID60 de Honeywell

Como se ha descrito en el apartado 1.6.3.2 *Notifier ID60 de Honeywell*, las centrales ID60 no disponen de una conexión con protocolo Modbus-TCP, siendo necesario el uso de un dispositivo IBOX (Fig.4) que actúa como pasarela de comunicaciones (Gateway), convirtiendo el protocolo propietario de esta CDI en el estándar Modbus-TCP con rol de Server.

El mapeado del IBOX está organizado en registros cuya lectura se puede hacer con los códigos de función 0x03 o 0x04 indistintamente, y para la escritura de los registros permitidos es necesario el uso de la función 0x06. Esta organización de memoria se divide en 6 áreas, de las cuales usaremos las 4 primeras para la monitorización y control del sistema.

La primera área contiene los registros asociados a eventos y ordenes de la propia CDI y cuyo mapeado es el mostrado en la tabla:

Registro	R/W	Descripción
1	R	Error Comunicación
2	R	Último Evento del Sistema
3	R	Alarma de Fuego
4	R	Pre-Alarma
5	R	Fallo
6	R	Dispositivo Desconectado
7	R	Evacuación
8	R	Prueba
9	R	Modo Ingeniero

Registro	R/W	Descripción
10	R	Fallo Comunicaciones ID3000 (No usado)
11	R	Ingeniero Silenciado (No usado)
12	R	Bocinas en Modo Retardo
13	R	Bocinas Desconectadas
14	R	(No usado)
15	R	Sistema en Modo Día
16	R	Equipo de Enrutamiento de Alarmas Desc. (No usado)
17	W	Comando Rearme del Sistema
18	W	Comando Finalizar Prueba
19	W	Comando Silenciar Bocinas
20	W	Comando Silenciar Zumbador interno CDI
21	W	Comando Evacuar
22	W	Comando Probar Bocinas
23	W	Comando Habilitar Bocinas

- El registro 1 corresponde a la señal de error de comunicación, habrá un valor de 0 cuando la comunicación con la central sea correcta, y un 1 cuando la central no esté respondiendo al IBOX.
- El registro 2 corresponde al último evento recibido desde la central, en este registro estará el código de evento recibido desde la central para propósitos de información y seguimiento de errores. Los códigos pueden consultarse al final del anexo 2.7. *Pasarela IBOX MB Server*.
- Los registros 3 a 16 corresponden a eventos generales recibidos desde la CDI, habrá un valor de 0 cuando el evento no esté activo, y un 1 cuando esté activo.
- Los registros 17 a 23 corresponden a comandos generales para enviar a la central. Escribiendo un 1 en estos registros se enviará el comando correspondiente a la CDI.

La segunda área contiene los registros asociados a los detectores del sistema (de humo, de llama, etc.). Cada detector tiene asociado un registro en función de su dirección dentro del lazo de la CDI siguiendo la siguiente regla: **Nº Registro=256+@Detector**, siendo posible direccionar un máximo de 99 detectores (Del registro 257 al 355).

Estos registros son de tipo lectura/escritura, disponiendo de doble funcionalidad. La lectura del registro devuelve el valor correspondiente al estado del detector codificado en bits, y la escritura corresponde al comando enviado al detector asociado.

Lectura Registro		Escritura Registro	
Bit	Descripción	Valor	Descripción
0	Alarma	0	Habilitar
1	Pre-Alarma	1	Deshabilitar
2	Fallo		
3	Deshabilitado		
4	En Prueba		

La tercera área contiene los registros asociados a los módulos del sistema (pulsadores de alarma, bocinas, etc.). Cada módulo tiene asociado un registro en función de su dirección dentro del lazo de la CDI siguiendo la siguiente regla: **Nº Registro=356+@Módulo**, siendo posible direccionar un máximo de 99 módulos (Del registro 357 al 455).

Estos registros son de tipo lectura/escritura, disponiendo de doble funcionalidad. La lectura del registro devuelve el valor correspondiente al estado del módulo codificado en bits, y la escritura corresponde al comando enviado al módulo asociado.

Lectura Registro		Escritura Registro	
Bit	Descripción	Valor	Descripción
0	Alarma	0	Habilitar
1	Pre-Alarma	1	Deshabilitar
2	Fallo	2	Activar
3	Deshabilitado	3	Desactivar
4	En Prueba		

- La activación y desactivación (valores 2 y 3) corresponde a la activación de la señal digital asociada. En el caso de una bocina, por ejemplo, la activación provocaría que la bocina sonara.

La cuarta área contiene los registros asociados a las zonas del sistema (distribución lógica de los dispositivos). Cada zona tiene asociado un registro en función de su número dentro siguiendo la siguiente regla: **Nº Registro=2304+Nº Zona**, siendo posible direccionar un máximo de 32 zonas para la CDI utilizada (Del registro 2305 al 2336).

Estos registros son de tipo lectura/escritura, disponiendo de doble funcionalidad. La lectura del registro devuelve el valor correspondiente al estado de la zona codificado en bits, y la escritura corresponde al comando enviado para la zona asociada.

Lectura Registro		Escritura Registro	
Bit	Descripción	Valor	Descripción
0	Alarma	0	Habilitar
1	Pre-Alarma	1	Deshabilitar
2	Fallo	2	Activar Prueba
3	En Prueba	3	Desactivar Prueba
4	Deshabilitado Parcial		
5	Deshabilitado Total		

Los estados hacen referencia a la zona y, por ende, a cualquier detector o módulo asociado a esa zona. Así, el deshabilitado parcial (bit 4) hace referencia a que hay algún módulo o detector asociado a esa zona que está deshabilitado, y el deshabilitado total (bit 5), a que toda la zona está deshabilitada. Del mismo modo, la activación y desactivación de prueba afectaría a todos los módulos y detectores asociados a esa zona.

Las zonas 5 y 6 corresponden a los valores analógicos de los Detectores y Módulos (temperatura, etc.) que no se utilizarán en este proyecto. En el anexo 2.7. *Pasarela IBOX MB Server* se encuentra el manual completo del IBOX donde se detalla el software necesario para su configuración, así como la descripción completa de sus funciones y un extracto del manual “Comunicador telefónico ID-CRA” donde se ha podido encontrar el listado de códigos de eventos indicados en el registro 2 del IBOX.

1.8.6 Comunicaciones con ZP2 de Ziton

las centrales ZP2 de Ziton disponen de conexión con protocolo Modbus-TCP, previo licenciamiento de activación. Estas CDI se pueden interconectar utilizando un protocolo propietario en el cual cada central tiene un identificador único, creando así, una topología de sistema distribuido. Esto permite, por ejemplo, que desde una central se pueda consultar el estado de los dispositivos conectados en otra. Esta topología distribuida abre la posibilidad a que la conexión Modbus-TCP se haga solo sobre una CDI, pudiendo consultar el estado de las demás a través de esta misma conexión, y ahorrando así en el número de licencias de activación del protocolo. Sin embargo, esto presenta varios inconvenientes:

- La pérdida de conexión con la central que dispone de protocolo Modbus-TCP inhabilita la comunicación con el resto.
- La central dispone de una limitación importante respecto a la velocidad de consultas permitidas. Según el manual de protocolo para esta CDI, tras enviar una trama de consulta, hay que esperar 1 segundo antes de poder enviar la siguiente.
- Cada consulta de petición, según el manual de protocolo, tiene un límite de lectura de 4 registros consecutivos.

De los dos últimos puntos anteriores se deduce que, si ya existe una la limitación para leer los registros de una CDI, esta se vería agravada si por la misma conexión se leen los registros de las demás. Es por eso, la decisión de que se establezca una conexión ModBus-TCP por cada una de las CDI de manera individual.

La ZP2 tiene la posibilidad de conectar 4 lazos con 256 dispositivos en cada uno, además de contar con 512 zonas configurables. Esta gran cantidad de registros disponibles junto a las limitaciones de velocidad de consulta descritos, hace que se adopten los siguientes criterios limitantes en las comunicaciones:

- El control de estas CDI será solo de lectura, no permitiendo enviar órdenes para ahorrar en el número de transacciones. Puesto que el objetivo principal del proyecto es la monitorización del sistema, esto no es un inconveniente.
- Cada zona tiene asociado un registro en la CDI, ocupando además un word en la memoria del PLC. De las 512 zonas configurables se decide hacer uso solo de las 32 primeras.
- Del mismo modo, cada módulo tiene asociado un registro en la CDI. De los 256 módulos configurables se decide hacer uso de los 127 primeros. siendo un tamaño suficiente para la instalación actual y posibles ampliaciones.

El mapeado de la ZP2 está organizado en registros de 16 bits cuya lectura se puede hacer con el código de función 0x03. Esta organización de memoria se divide en varias áreas dependiendo del modo configurado. Modo "ZONE" con posibilidad de direccionar hasta 128 nodos (CDIs), o como en nuestro caso, modo "ZONEPOINT" con posibilidad de direccionar hasta 32 nodos y permitiendo, además, monitorizar el estado individual de cada dispositivo. Hay que tener en cuenta que cada área está dimensionada para 32 nodos, por lo que solo haremos uso de los registros asociados al nodo 1, por usar conexión ModBus-TCP individual.

Registros	Descripción
0x2001 – 0x2080	Estado CDI (32 Nodos)
0x3001 – 0x7000	Estado Zonas (32 Nodos, 512 Zonas)
0x7001 – 0xF000	Estado Dispositivos (32 Nodos, 4 Lazos, 256 Disp.)

La primera área contiene los registros asociados a los estados de la propia central. Aunque se trata de los registros comprendidos entre el 0x2001 y 0x2004, solo tienen funcionalidad los dos primeros, codificando cada bit como se muestra a continuación.

- Registro 0x2001

Byte Alto								Byte Bajo							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Estados Bocinas								Condiciones Funcionales							

Bit	Descripción	Bit	Descripción
15	No Usado	7	No Usado
14	En prueba	6	No Usado
13	Tiempo Transcurrido Anulado	5	Alarma MCP (Pulsadores)
12	Desactivadas	4	Modo Noche
11	Salidas Silenciadas	3	En Prueba
10	Salidas Activadas	2	Desactivado
9	Activación Retardo en Progreso	1	Fallo
8	Retardo Habilitado	0	Alarma

- Registro 0x2002

Byte Alto								Byte Bajo							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Estados Protección Fuego								Estados Bomberos							

Bit	Descripción	Bit	Descripción
15	No Usado	7	No Usado
14	No Usado	6	No Usado
13	En Prueba	5	En Prueba
12	Desactivado	4	Desactivado
11	Alarma Reconocida	3	Alarma Reconocida
10	Salidas Activadas	2	Salidas Activadas
9	Activación Retardo en Progreso	1	Activación Retardo en Progreso
8	Retardo Habilitado	0	Retardo Habilitado

La segunda área contiene los registros asociados a las zonas del sistema (distribución lógica de los dispositivos). Aunque para el nodo 1, se trata de los registros comprendidos entre el 0x3001 y 0x3200, solo tendremos en cuenta los 32 primeros, correspondiendo cada uno de ellos a una zona determinada y codificando cada bit como se muestra a continuación.

- Registros 0x3001 al 0x3020

Byte	Byte Bajo							
15-8	7	6	5	4	3	2	1	0
				Deshabilitada	Prueba	Fallo	Alarma	PreAlarma

Los estados hacen referencia a la zona y, por ende, a cualquier dispositivo asociado a esa zona. Así, por ejemplo, el estado de fallo en la zona indica que uno o varios dispositivos se encuentran en fallo.

La tercera área contiene los registros asociados a los dispositivos del sistema (detectores y módulos). El área está dividida en 4 sub-áreas correspondientes a cada lazo de comunicaciones de la CDI, cada uno de ellos, a su vez, con hasta 256 dispositivos.

Registros	Descripción
0x7001 – 0x7100	Lazo 1. Dispositivos del 1 al 256
0x7101 – 0x7200	Lazo 2. Dispositivos del 1 al 256
0x7201 – 0x7300	Lazo 3. Dispositivos del 1 al 256
0x7301 – 0x7400	Lazo 4. Dispositivos del 1 al 256

Del mismo modo que los registros de las zonas, cada registro de dispositivo presenta la siguiente codificación de bit.

Byte	Byte Bajo							
	7	6	5	4	3	2	1	0
				Deshabilitado	Prueba	Fallo	Alarma	PreAlarma

En el anexo 2.8. *Ziton Fire Panel Modbus Protocol*. Se encuentra un extracto del manual donde se detalla la descripción completa de sus funciones.

1.8.7 Sistema de Control PCS7

En el apartado 1.8.2 *Arquitectura de Integración con el Sistema de Control* hemos visto la arquitectura general del sistema desde el punto de vista de hardware. En este apartado nos centraremos en la configuración y programación del software sin entrar en detalle de todas las funcionalidades que puede ofrecer.

En la estación de ingeniería (ES) se encuentra instalado el software *Simatic Manager*, que es el puesto central para la ejecución de las tareas de ingeniería de un multiproyecto de PCS7. Un multiproyecto de PCS7 está formado por proyectos individuales siguiendo una distribución lógica del sistema, cada uno de los cuales tiene la información de un área de aplicación específica. Para el acceso a los datos del multiproyecto, el software ofrece varios tipos de vistas de las que destacaremos la *vista de planta*, que muestra el multiproyecto con sus niveles jerárquicos, lo que viene a ser una representación lógica de la planta real, y la *vista de componentes*, que muestra el multiproyecto desde el punto de vista de hardware, conteniendo los sistemas de bus, sistemas de automatización (AS), Servidores de operación (OS), etc. (Fig. 32).

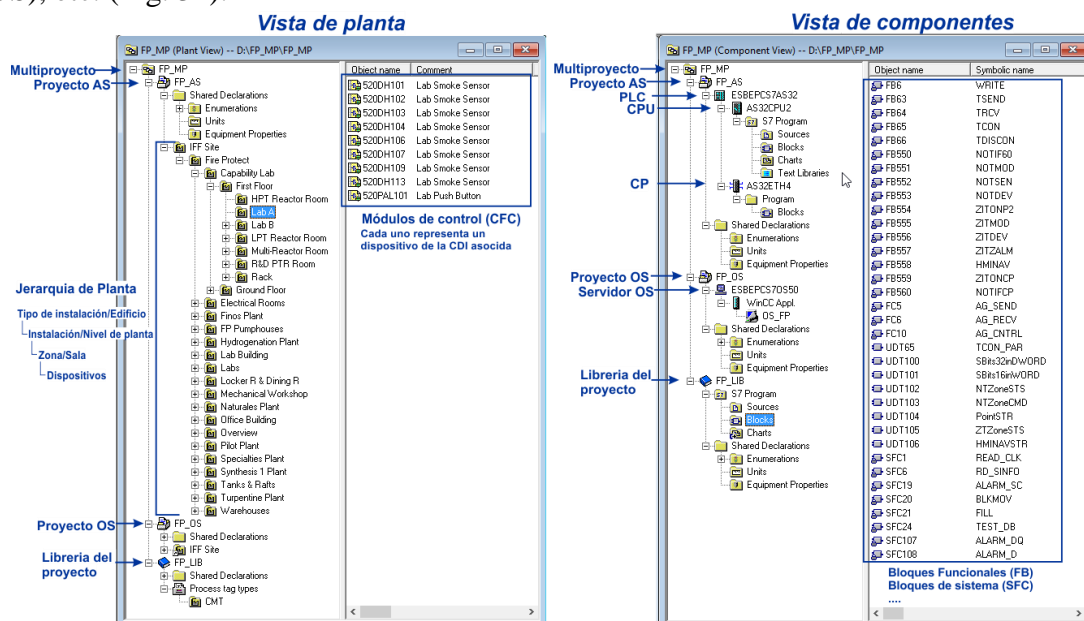


Figura 32. Simatic Manager. Vistas del multiproyecto PCS7

1.8.7.1 Configuración de Hardware y Redes

Para la configuración del hardware del proyecto, PCS7 dispone del componente *HWConfig*. Mediante esta herramienta se pueden definir todos los elementos del sistema y configurar sus propiedades. Esta herramienta brinda un amplio catálogo de hardware de Siemens con la posibilidad de importar componentes de otros fabricantes e incluirlos en los buses de campo (Profibus/Profinet). (Fig. 35)

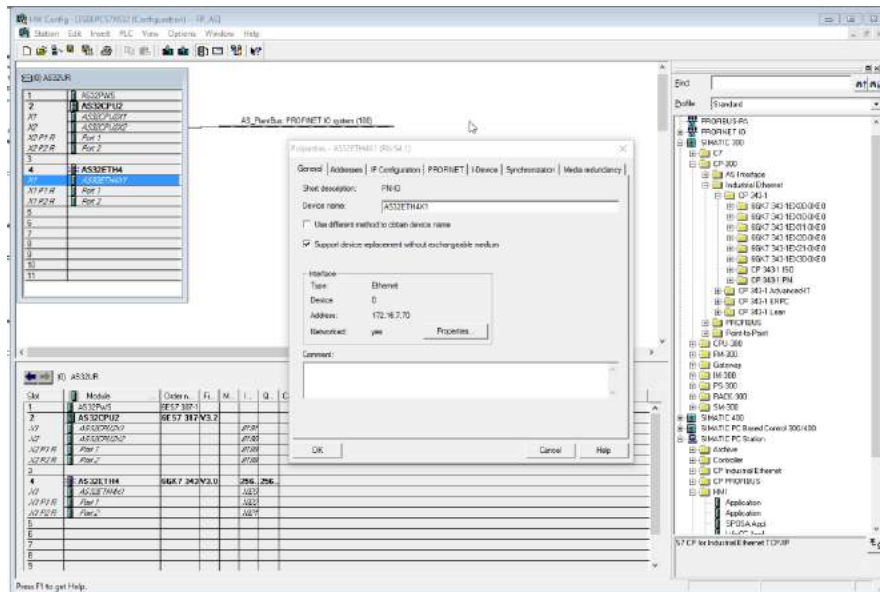


Figura 35. HWConfig. Configuración del Hardware

Una vez definido el hardware, y mediante el componente *NetPro*, es posible configurar las redes de comunicaciones, incluyendo aspectos como su topología o enlaces de comunicación de los diferentes dispositivos del proyecto. (Fig.36)

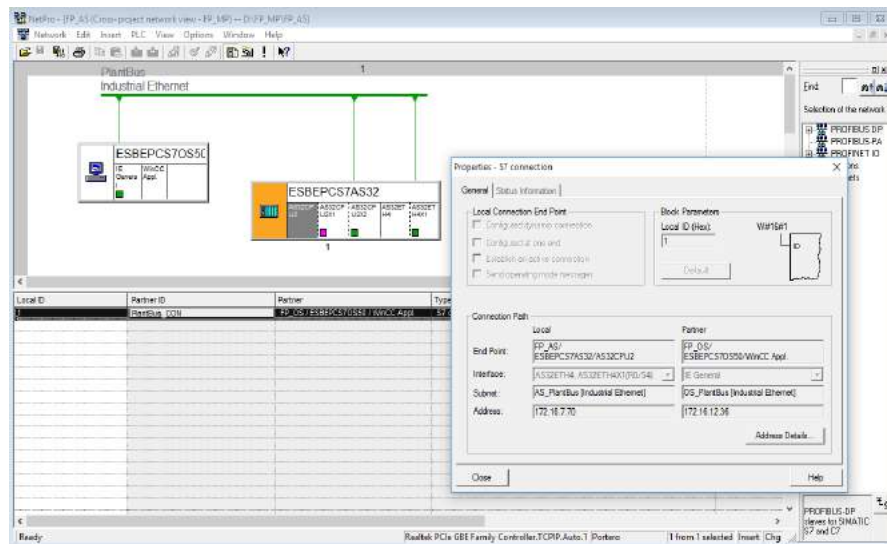


Figura 36. NetPro. Configuración de redes.

En el anexo 2.9 *PCS7: Configuración de Hardware y Redes*, quedan detalladas ambas configuraciones para este proyecto.

1.8.7.2 Jerarquía de Planta

La jerarquía de planta, además de ser una representación lógica de la planta real, organiza los diferentes niveles de visualización en la parte gráfica del sistema, pudiéndose crear diferentes pantallas por cada uno de ellos. Esta jerarquía se puede consultar en el anexo 2.10 PCS7: *Jerarquía de Planta*, y está distribuida en niveles siguiendo la siguiente estructura y criterios:

- Nivel 1: Tipo de instalación o Edificio
Este nivel presenta una vista general resaltando los tipos de instalación o edificios y muestra el estado del sistema solo para las instalaciones seleccionadas. El objetivo es clasificar las instalaciones por funcionalidad ya que este nivel también hace referencia a la distribución de áreas del sistema. Las áreas del sistema agrupan aspectos tales como alarmas, control de acceso, etc. permitiendo, por ejemplo, poder controlar que un cliente de operación (OC) solo muestre las alarmas de un área determinada.
Este nivel jerárquico también conforma la distribución del menú principal dentro del sistema de operación. En el anexo 2.11 *Jerarquía en Pantallas del Sistema* se pueden ver varios ejemplos.
- Nivel 2: Instalación o Elevación
Este nivel presenta una vista general de la instalación seleccionada o una elevación determinada en el caso de edificios con más de un piso de altura. El objetivo es tener una visión general de la distribución de la instalación o de la distribución de estancias en el piso. En función de la extensión de la instalación, este nivel jerárquico puede contener *block icons*. En el anexo 2.11 *Jerarquía en Pantallas del Sistema* se pueden ver varios ejemplos.
- Nivel 3: Zona o Estancia
Este nivel es una visión ampliada del Nivel 2 para una zona concreta. El objetivo es tener una visión detallada de la zona seleccionada donde se muestran los dispositivos del sistema acorde a su posición real. Estos dispositivos están representados por *block icons* y disponen de *faceplates* asociados para su gestión individual. En el anexo 2.11 *Jerarquía en Pantallas del Sistema* se pueden ver varios ejemplos.

1.8.7.3 Librería de Bloques Funcionales

Los bloques funcionales (FB) son bloques de código reutilizables que permiten modularizar un programa, haciendo más fácil su configuración y entendimiento. Estos bloques toman los valores de sus parámetros de entradas, los procesan y muestran los resultados en sus parámetros de salidas. Cada instancia de un FB tiene asociado un bloque de datos (DB) de memoria en el que puede almacenar información. Esta información puede ser recuperada posteriormente para su procesamiento en las siguientes ejecuciones de la instancia.

El PLC dispone de bloques de organización (OB), cada uno con funcionalidades diferentes. Así, por ejemplo, tenemos el OB100 que se ejecuta al arranque del PLC, el OB85 que se ejecuta cuando existe un error en la ejecución del programa, OBs que se ejecutan cíclicamente con diferentes periodos de tiempo, como el OB32 cada segundo o el OB34 cada 200 ms, etc. Estos OBs cíclicos son la base de ejecución de los FBs. De esta forma, la instancia de un FB puede incluirse en un OB cíclico determinado, asegurando su ejecución periódica en la base de tiempos del OB.

Para la escritura del código de un FB, se dispone de diferentes lenguajes de programación. Sin embargo, por su potencial y simplicidad a la hora de depurar el código se utiliza el lenguaje SCL, un lenguaje de programación textual y estructurado. En la figura 37 puede verse una comparativa de las ventajas de utilizar este tipo de lenguaje.

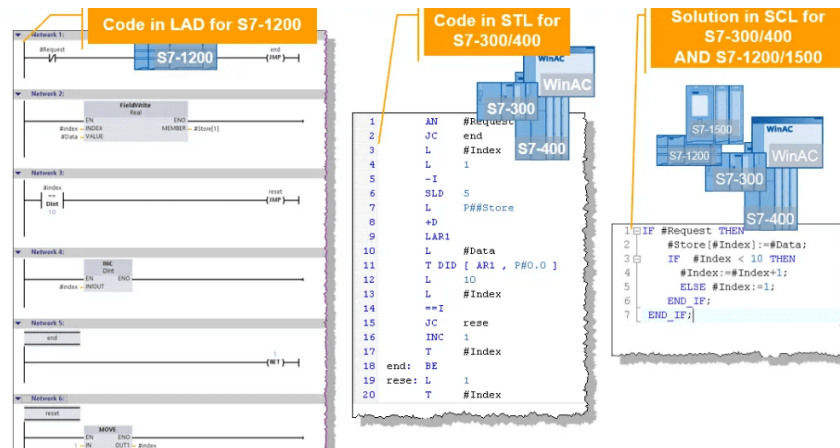


Figura 37. Comparativa SCL con otros lenguajes.

Otras ventajas destacables a la hora de utilizar el lenguaje SCL:

- Lenguaje estructurado de control lenguaje de programación textual y estructurado.
- Se corresponde con la norma IEC 61131-3 (ST).
- Certificado de acuerdo con PLCopen.
- Ejecución de cálculos complejos y algoritmos.
- Operaciones para la manipulación memoria.
- Uso de estructuras de datos complejas y punteros.
- Código de programa intercambiable con las distintas plataformas de Siemens.

La estructura del código para la creación de un FB en lenguaje SCL se compone de:

- **Declaración de las opciones de compilación y creación del bloque:**
En esta sección se pueden definir los valores por defecto que utilizará el compilador de SCL a la hora de generar el bloque y que serán tenidos en cuenta por encima de las opciones que tenga definidas por defecto el compilador.
- **Declaración de tipos:**
En esta sección se declaran los tipos de datos generales tales como estructuras de datos que pueden ser usados por cualquier FB. El compilador SCL genera un bloque con la estructura de los tipos declarados denominado UDT (User Data Type).
- **Cabecera:**
En esta sección se declara la información general del FB, como el número de FB, Título, Descripción, Versión, etc.
- **Variables de entrada:**
En esta sección se declaran todas las variables de entrada que podrán ser procesadas por el FB. Estas variables pueden ser de tipos predefinidos (excepto *Arrays*), estructuras o tipos de datos declarados por el usuario (UDT). Cada variable de entrada puede tener asociado uno o varios atributos que son utilizados para establecer cómo va a ser tratada en la compilación del proyecto OS (La parte gráfica del sistema), o para definir sus características dentro del entorno gráfico de programación CFC donde aparecen como

patas de entrada del bloque. Para cada una de las entradas, el compilador SCL genera un espacio de memoria en la estructura del DB asociado.

- **Variables de salida:**

En esta sección se declaran todas las variables de salida que actualiza el FB tras el procesamiento del código. Al igual que las entradas, pueden ser de tipos predefinidos (excepto *Arrays*), estructuras o tipos de datos declarados por el usuario (UDT). Cada variable de salida puede tener asociado uno o varios atributos que son utilizados para establecer cómo va a ser tratada en la compilación del proyecto OS (La parte gráfica del sistema), o para definir sus características dentro del entorno gráfico de programación CFC donde aparecen como patas de salida del bloque. Para cada una de las salidas, el compilador SCL genera un espacio de memoria en la estructura del DB asociado.

- **Variables:**

En esta sección se declaran todas las variables internas que utiliza el código. Pueden ser de tipos predefinidos incluyendo *Arrays*, estructuras o tipos de datos declarados por el usuario (UDT). Para cada una de las variables, el compilador SCL genera un espacio de memoria en la estructura del DB asociado.

- **Variables Temporales:**

En esta sección se declaran todas las variables temporales, las cuales tienen las mismas características que las variables del apartado anterior, con la excepción de que no se guardan en memoria. El compilador SCL no genera un espacio de memoria en la estructura del DB asociado y, por lo tanto, no se pueden recuperar en posteriores ejecuciones de la instancia.

- **Código de Procesamiento:**

En esta sección, se escribe el propio código de procesamiento en lenguaje estructurado SCL. Este lenguaje, al igual que otros denominados de “alto nivel”, permite crear estructuras complejas como iteraciones, bucles, condiciones, etc.

La filosofía utilizada para crear los diferentes FB del sistema, ha sido seguir la misma distribución lógica de una CDI. (Fig.38)

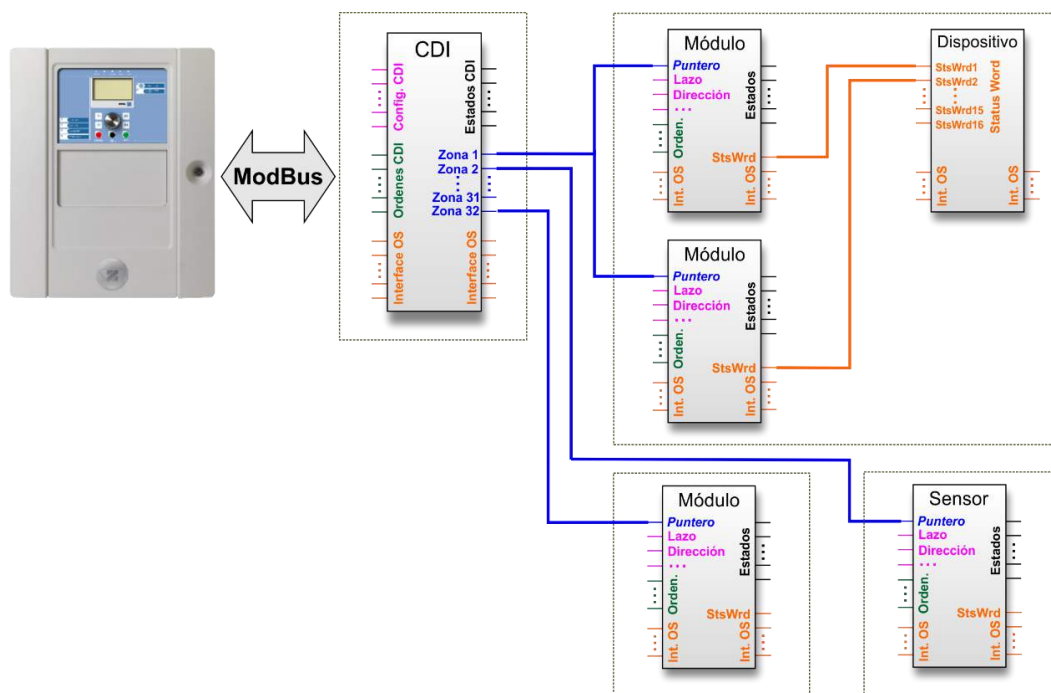


Figura 38. Distribución y conexionado de FBs.

En esta distribución tendremos los siguientes tipos Bloques Funcionales:

- **FB CDI:**
Representa a la propia central con la que establece comunicaciones. Este tipo de FB es el núcleo del sistema y el nexo común con el resto de bloques. Almacena toda la información leída y manda los comandos solicitados por los FB de Módulos y Sensores.
- **FB Módulo:**
Representa a un elemento de campo de tipo Módulo (Dispositivo de señal simple). Este tipo de bloque se conecta al FB CDI por medio de un puntero permitiéndole leer y escribir directamente en su DB mediante una única conexión.
- **FB Sensor:**
Representa a un elemento de campo de tipo Sensor (Dispositivo de señal simple), solo aplicable a CDI Notifier ID60. Al igual que los Módulos, se conecta al FB CDI por medio de un puntero permitiéndole leer y escribir directamente en su DB mediante una única conexión.
- **FB Dispositivo:**
Representa a un elemento de campo de tipo multi-señal. Cada una de las señales pertenecientes al dispositivo llegan al sistema como señal simple. Este tipo de bloque las aglutina pudiéndose representar gráficamente acorde al elemento de campo real. Así, por ejemplo, una central de extinción (CEX) puede tener varias señales simples (extinción anulada, fallo extinción, etc.), que pueden ser representadas como un único *block icon* en pantalla.
- **FB Zona:**
Representa a una zona para las centrales Ziton ZP2. Este tipo de bloque solo se utiliza para generar alarmas en el sistema OS (La parte gráfica del sistema) acordes con el estado de la zona. Como veremos más adelante, la necesidad de este tipo de FB está ligada a la limitación de velocidad de consultas para estas CDI.

En los siguientes apartados haremos la descripción de los aspectos más importantes para cada uno de los bloques funcionales. El objetivo no es explicar en detalle el código, pero sí dar una idea de su diseño para su mejor comprensión.

1.8.7.3.1 FB550: NOTIF60 (FB CDI)

Para el establecimiento de la conexión se necesita asignar un número de identificador en el parámetro *Id* del bloque. Este parámetro puede tomar valores entre 0x0001 y 0x0FFF teniendo que ser único dentro del PLC, ya que fija el canal de conexión que utilizará dentro de los 32 disponibles para la CPU de este proyecto. El bloque permite además configurar la dirección IP y el puerto de conexión de la CDI.

La lectura de las variables se hace con la función 03 de ModBus (Lectura Múltiple de Registros) y la escritura con la función 06 (Escritura Simple de Registro).

En una consulta de datos, el campo *Identificador de Transacción* del protocolo se utiliza para identificar el área de datos consultado. Este identificador devuelto en la respuesta se utilizará para saber qué área de memoria se tiene que actualizar.

Las consultas para envío de ordenes tienen prioridad sobre las de lectura de estados durante un tiempo configurable en el parámetro *Max_CMD_Time*. Transcurrido este tiempo, se realiza un ciclo completo de actualización de estados, aunque queden ordenes pendientes de procesar. En cada ejecución, y si no hay ordenes a procesar, el bloque lee un área de estados

diferente. Teniendo en cuenta que por defecto el FB se ejecuta cada 200 ms, obtenemos una actualización completa de estados cada segundo.

El DB asociado dispone de 7 zonas de memoria donde se guardan los valores de cada área de la CDI. Los bloques de módulos y sensores pueden acceder a estas zonas por medio de un puntero para actualizar su estado o indicarle al bloque NOTIF60 que envíe una orden.

Variable	Tipo	Direcciones (Bytes)	Descripción
PanelCMD	ARRAY [1..7] OF WORD	596 - 608	Panel Commands
ZonesCMD	ARRAY [1..32] OF WORD	610 - 672	Zones Commands
SensorsCMD	ARRAY [1..99] OF WORD	674 - 870	Sensors Commands
ModulesCMD	ARRAY [1..99] OF WORD	872 - 1068	Modules Commands
ZonesSTS	ARRAY [1..32] OF WORD	1070 - 1132	Zones Status
SensorsSTS	ARRAY [1..99] OF WORD	1134 - 1330	Sensors Status
ModulesSTS	ARRAY [1..99] OF WORD	1332 - 1528	Modules Status

Este bloque es capaz de generar 3 tipos de mensaje asociados a eventos de la propia central: Alarma de Fuego, Alarma de evacuación y Fallo general, que serán mostrados en el sistema OS. Los mensajes asociados a elementos de campo, son gestionados por los propios FB que representan estos elementos.

En las siguientes figuras (Fig.39, 40 y 41) quedan reflejados los diagramas de flujo para las comunicaciones de este bloque. En el apartado 2.12 *FB550:NOTIF60* se puede consultar el código completo.

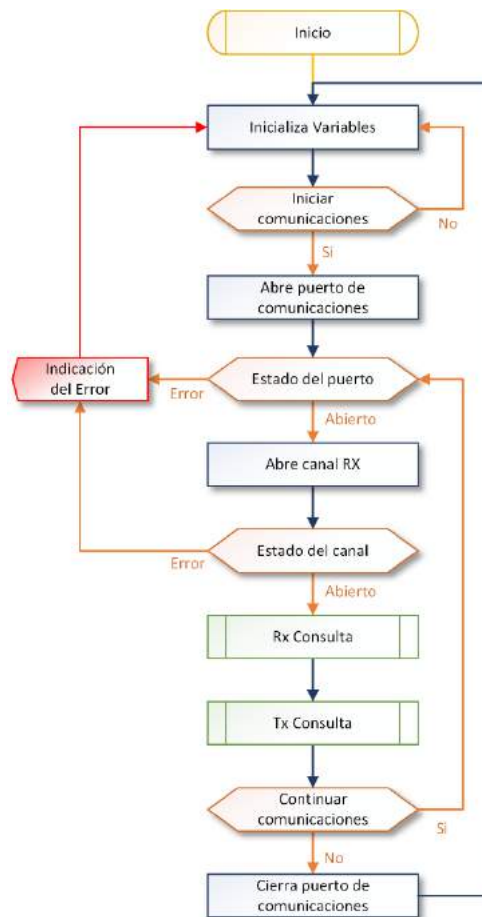


Figura 39. Función principal bloque NOTIF60.

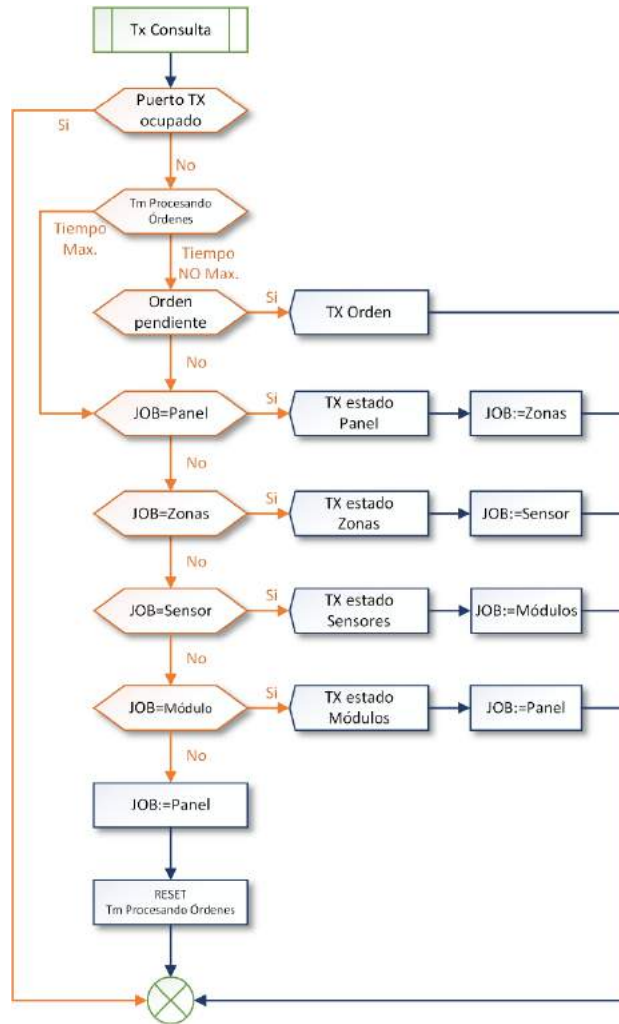


Figura 40. Función de transmisión bloque NOTIF60.

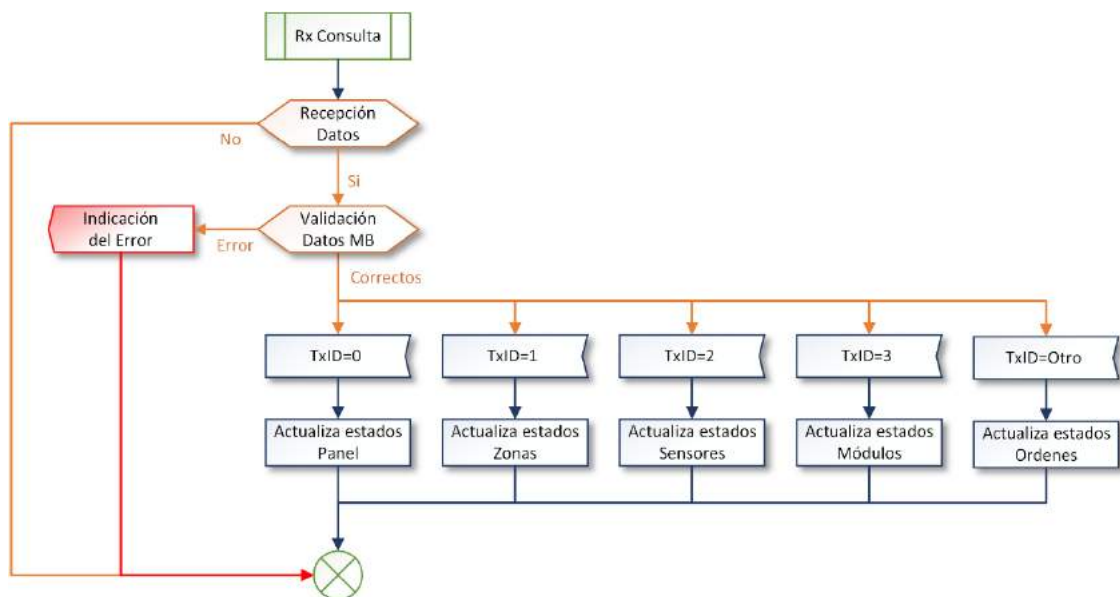


Figura 41. Función de recepción bloque NOTIF60.

1.8.7.3.2 FB551: NOTMOD (FB Módulo)

El bloque accede al FB NOTIF60 por medio de un puntero definido en su parámetro de entrada *Zone_STS*. Este, debe conectarse a uno de los parámetros de salida *Zonexx_STS* del FB NOTIF60, donde xx corresponde a la zona de la CDI donde está configurado el módulo.

Conocido el número de Zona, y gracias al parámetro *Mod_Id* (Identificador de módulo), el bloque es capaz de calcular los registros de estado y órdenes al que debe acceder dentro de las zonas de memoria del FB NOTIF60, *ModulesSTS* y *ModulesCMD* respectivamente.

El bloque es capaz de generar 3 tipos de mensaje asociados a los eventos del módulo: Alarma, Pre-Alarma y Fallo, que serán mostrados en el sistema OS.

Este bloque está preparado para conectar con el bloque *FB560: NOTIFCP* de tipo FB CDI, quedando fuera del alcance de este proyecto. La funcionalidad del NOTIFCP es la misma que la del NOTIF60, pero la conexión con la CDI se realiza por el puerto de la tarjeta de comunicaciones CP 343-1 en lugar del puerto de la propia CPU. Esto supone un cambio en la manera de trabajar con el puerto, lo que implica una modificación del código que hace desplazar las zonas de memoria del FB CDI donde accede NOTMOD.

En el anexo 2.13 *FB551:NOTMOD* se puede consultar el código completo.

1.8.7.3.3 FB552: NOTSEN (FB Sensor)

El bloque accede al FB NOTIF60 por medio de un puntero definido en su parámetro de entrada *Zone_STS*. Este, debe conectarse a uno de los parámetros de salida *Zonexx_STS* del FB NOTIF60, donde xx corresponde a la zona de la CDI donde está configurado el sensor.

Conocido el número de Zona, y gracias al parámetro *Sen_Id* (Identificador de sensor), el bloque es capaz de calcular los registros de estado y órdenes al que debe acceder dentro de las zonas de memoria del FB NOTIF60, *SensorsSTS* y *SensorsCMD* respectivamente.

El bloque es capaz de generar 3 tipos de mensaje asociados a los eventos del sensor: Alarma, Pre-Alarma y Fallo, que serán mostrados en el sistema OS.

Este bloque está preparado para conectar con el bloque *FB560: NOTIFCP* de tipo FB CDI, quedando fuera del alcance de este proyecto. La funcionalidad del NOTIFCP es la misma que la del NOTIF60, pero la conexión con la CDI se realiza por el puerto de la tarjeta de comunicaciones CP 343-1 en lugar del puerto de la propia CPU. Esto supone un cambio en la manera de trabajar con el puerto, lo que implica una modificación del código que hace desplazar las zonas de memoria del FB CDI donde accede NOTMOD.

En el anexo 2.14 *FB552:NOTSEN* se puede consultar el código completo.

1.8.7.3.4 FB553: NOTDEV (FB Dispositivo)

Cada una de las entradas del bloque *STInx* se debe conectar al parámetro se salida *Status1* de los bloques NOTMOD y NOTSEN que forman parte del dispositivo multi-señal, siendo posible la conexión de hasta 16 bloques.

Cada parámetro *Status1* del FB de origen, contiene los estados más relevantes codificados por bit. NOTDEV realiza una OR lógica de todos ellos, mostrando un resumen del estado global.

El bloque es capaz de generar 3 tipos de mensaje asociados a los eventos de los módulo y sensores asociados: Alarma, Pre-Alarma y Fallo, que serán mostrados en el sistema OS. Al igual que los estados, los mensajes se generan a partir de una OR lógica de los mensajes de origen.

En el anexo 2.15 *FB553:NOTDEV* se puede consultar el código completo.

1.8.7.3.5 FB554: ZITONP2 (FB CDI)

Para el establecimiento de la conexión se necesita asignar un número de identificador en el parámetro *Id* del bloque. Este parámetro puede tomar valores entre 0x0001 y 0x0FFF teniendo que ser único dentro del PLC, ya que fija el canal de conexión que utilizará dentro de los 32 disponibles para la CPU de este proyecto. El bloque permite además configurar la dirección IP y el puerto de conexión de la CDI.

La lectura de las variables se hace con la función 03 de ModBus (Lectura Múltiple de Registros), con un valor de 4 registros por consulta debido a las limitaciones de la CDI.

En una consulta de datos, el campo *Identificador de Transacción* del protocolo se utiliza para identificar el área de datos consultado. Este identificador devuelto en la respuesta se utilizará para saber qué área de memoria se tiene que actualizar. El formato de este identificador sigue los siguientes criterios:

	Byte Alto		Byte Bajo	
Panel	0000	0000	0000	0000
Zonas	0000	Nº Zona	0000	0001
Módulos	Nº Módulo		Nº Lazo	0010

Siendo el Nº de Zona y Nº de Módulo el primero de los 4 registros leídos en la consulta.

La gran cantidad de registros de los que dispone la CDI, y sus limitaciones de comunicación (1 consulta por segundo de 4 registros consecutivos), plantean un problema. Si se intenta hacer una lectura completa de todos ellos de forma cíclica, nos encontramos tiempos de escaneo demasiado elevados para un sistema de seguridad.

	Nº Registros	Tiempo (seg.)
Panel	2	1
Zonas	32*	8
Módulos (4 Lazos)	508*	127
Total		136

*Valores menores que la capacidad real de la CDI, justificado en el apartado 1.8.6 *Comunicaciones con ZP2*

Pudiéndose encontrar retardos de más de 2 minutos desde que se produce un evento hasta que el bloque lo pueda apereibir, tal y como se muestra en la tabla anterior.

Para minimizar esto, se adoptan los siguientes criterios en el diseño del FB:

- El FB dispone de los parámetros de entrada: *Zones*, *Loop1_Mod*, *Loop2_Mod*, *Loop3_Mod* y *Loop4_Mod*. Estos parámetros configuran el valor más alto para esa área que el bloque debe consultar, evitando leer los que no están configurados en la CDI.
- Los estados del panel y de las zonas tienen mayor peso que los estados de los módulos, ya que los primeros informan de eventos de forma global, aunque posteriormente se necesite de los segundos para concretar quién generó el evento. Bajo esta premisa, el FB escanea con mayor asiduidad los estados del panel y las zonas, sacrificando tiempo para escanear el completo de todos los módulos. Para ello, y mediante el parámetro *Mod_cycle* se puede definir cuantos módulos se escanean consecutivamente antes de que el bloque vuelva a hacer una consulta de los estados del panel y las zonas.
- Ante el evento de una zona, todos los módulos conectados a esa zona realizan una acción de “levantar la mano”. El bloque ZITONP2 no guarda esta relación de módulo/zona asociada, y es por eso que son los propios módulos los que tienen que informar. Con esta acción, el bloque entra en modo “Alta prioridad” por el cual, durante un ciclo completo de direcciones, solo lee los registros asociados a los módulos que han “levantado la mano”.
- Aunque este criterio no forme parte del diseño del FB, se tiene en consideración en este apartado por tener relación directa con las limitaciones de comunicación de la CDI. Los bloques de tipo FB CDI solo generan mensajes asociados a eventos de la propia central, sin tener en consideración los eventos asociados a las zonas. Esto es debido a que son los propios módulos los que concretan la causa del evento con sus mensajes, evitando duplicidad de alarmas en el sistema OS. Es decir, si un módulo de campo entra en alarma, el FB asociado genera un mensaje de alarma en el OS, que de manera implícita concreta el problema sin necesidad de mostrar otro mensaje de alarma para la zona, ya que este sería trivial. En el caso del bloque ZITONP2, el modo “Alta prioridad” optimiza el modo de lectura de registros, pero no elimina el retardo entre la generación del evento en una zona, y concretar quién lo generó. Es por eso, la necesidad del tipo **FB Zona**, el cual añade esta funcionalidad específica, generando mensajes en el sistema OS en base a los estados de las zonas y permitiendo un apereibimiento menos concreto, pero más rápido.

El DB asociado dispone de 6 zonas de memoria donde se guardan los valores de cada área de la CDI. Los bloques de módulos y zona pueden acceder a estas zonas por medio de un puntero para actualizar su estado.

Variable	Tipo	Direcciones (Bytes)	Descripción
ZonesSTS	ARRAY [1..32] OF WORD	522 - 584	Zones Status
L1ModSTS	ARRAY [1..128] OF WORD	586 - 840	Modules Loop1
L2ModSTS	ARRAY [1..128] OF WORD	842 - 1096	Modules Loop2
L3ModSTS	ARRAY [1..128] OF WORD	1098 - 1352	Modules Loop3
L4ModSTS	ARRAY [1..128] OF WORD	1354 - 1608	Modules Loop4
AckZonesSTS	ARRAY [1..32] OF WORD	1610 - 1672	Zones Acknowledged

Este bloque es capaz de generar 3 tipos de mensaje asociados a eventos de la propia central: Alarma general, Alarma MCP (por pulsación de botón de emergencia) y Fallo general, que

serán mostrados en el sistema OS. Los mensajes asociados a elementos de campo, son gestionados por los propios FB que representan estos elementos.

En las siguientes figuras (Fig.42, 43 y 44) quedan reflejados los diagramas de flujo para las comunicaciones de este bloque. En el apartado 2.16 *FB554:ZITONP2* se puede consultar el código completo.

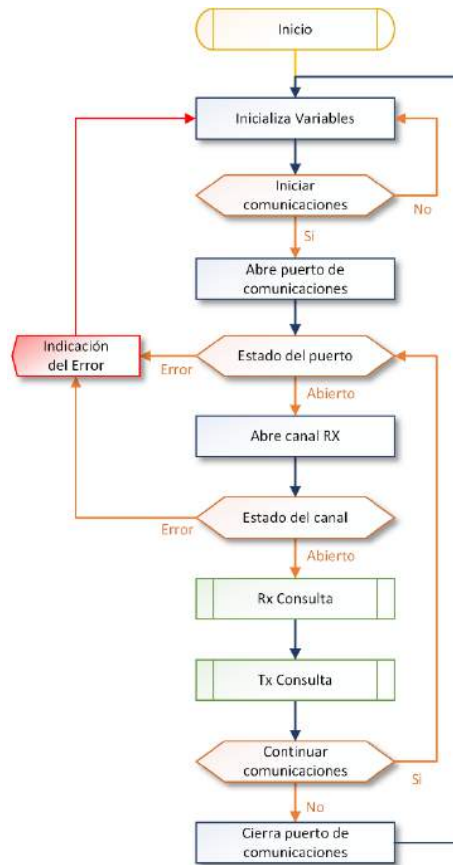


Figura 42. Función principal bloque ZITONP2.

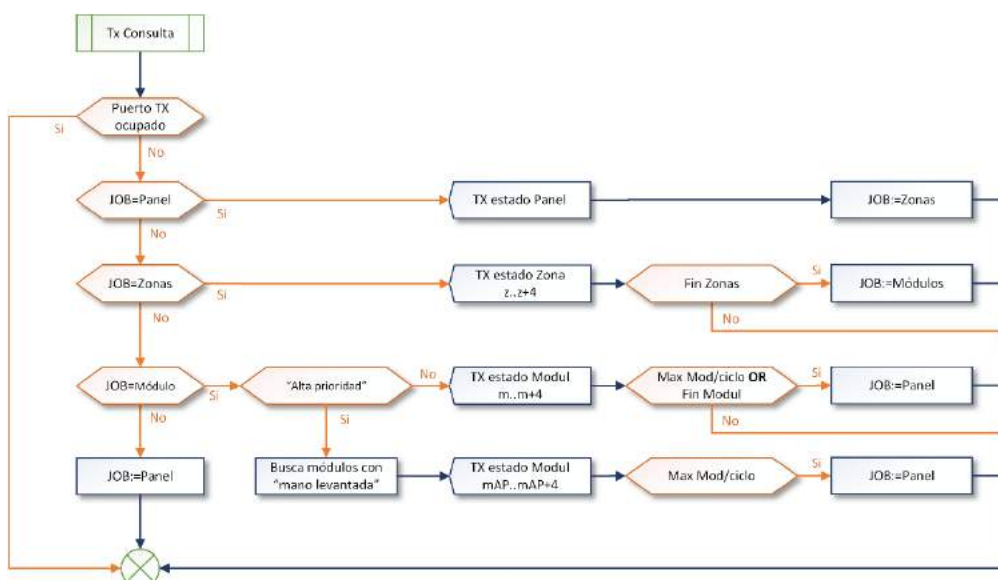


Figura 43. Función de transmisión bloque ZITONP2.

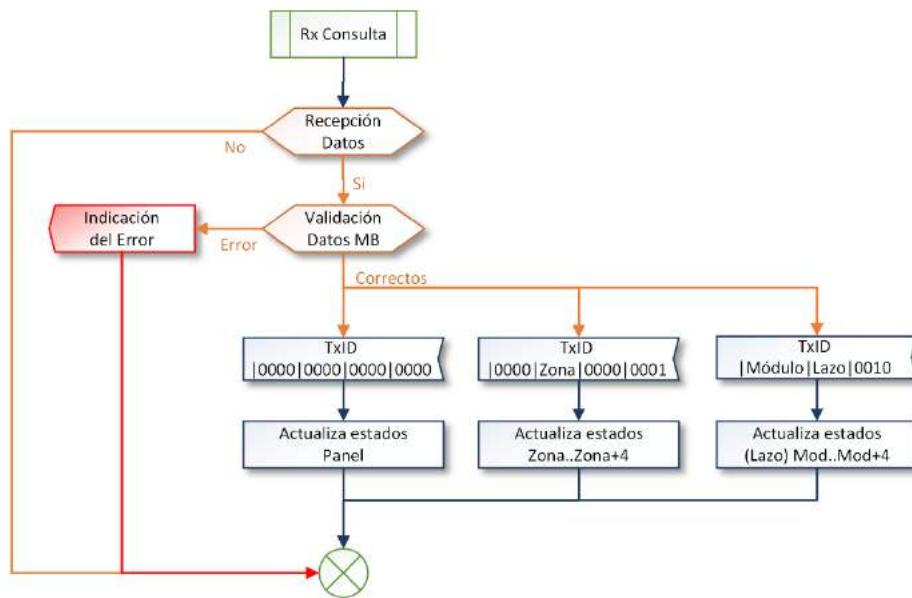


Figura 44. Función de recepción bloque ZITONP2.

1.8.7.3.6 FB555: ZITMOD (FB Módulo)

El bloque accede al FB ZITONP2 por medio de un puntero definido en su parámetro de entrada *Zone_STS*. Este, debe conectarse a uno de los parámetros de salida *Zonexx_STS* del FB ZITONP2, donde xx corresponde a la zona de la CDI donde está configurado el módulo.

Conocido el número de zona, y gracias a los parámetros *Mod_Id* (Identificador de módulo) y *Loop_Id* (Identificador del lazo), es capaz de calcular el registro de estado al que debe acceder dentro de las áreas de memoria del ZITONP2 ($L''Loop_Id \text{ "ModSTS[Mod_Id]}$).

Cuando se produce un evento de la zona asociada, el bloque realiza la acción de “levantar la mano”, consistente en poner a 1 el bit14 del registro de estado correspondiente, en el FB ZITONP2 (que no está en uso por la CDI). Acto seguido, pone a 1 el bit correspondiente a la zona en el parámetro *HPScanCMD* del FB ZITONP2, provocando que el FB ZITONP2 se ponga en modo de “Alta prioridad”. Una vez leído el registro de estado para este módulo por parte del FB ZITONP2, el bit14 se habrá puesto a 0 ya que es su valor por defecto en la CDI. Si el bloque reconoce que es el causante del evento, pondrá a 1 los bits correspondientes a los eventos en el parámetro *AckZonesSTS[Zone]* del FB ZITONP2. Este registro será leído por el bloque FB ZITALM, provocando que desactive los mensajes de evento que ha enviado al sistema OS.

El bloque es capaz de generar 3 tipos de mensaje asociados a los eventos del módulo: Alarma, Pre-Alarma y Fallo, que serán mostrados en el sistema OS.

Este bloque está preparado para conectar con el bloque *FB559: ZITONCP* de tipo FB CDI, quedando fuera del alcance de este proyecto. La funcionalidad del *ZITONCP* es la misma que la del *ZITONP2*, pero la conexión con la CDI se realiza por el puerto de la tarjeta de comunicaciones CP 343-1 en lugar del puerto de la propia CPU. Esto supone un cambio en la manera de trabajar con el puerto, lo que implica una modificación del código que hace desplazar las zonas de memoria del FB CDI donde accede *ZITONP2*.

En el anexo 2.17 *FB555:ZITMOD* se puede consultar el código completo.

1.8.7.3.7 FB556: ZITDEV (FB Dispositivo)

Cada una de las entradas del bloque *STInx* se debe conectar al parámetro de salida *Status1* de los bloques ZITMOD que forman parte del dispositivo multi-señal, siendo posible la conexión de hasta 16 bloques.

Cada parámetro *Status1* del FB de origen, contiene los estados más relevantes codificados por bit. ZITDEV realiza una OR lógica de todos ellos, mostrando un resumen del estado global.

El bloque es capaz de generar 3 tipos de mensaje asociados a los eventos de los módulos y sensores asociados: Alarma, Pre-Alarma y Fallo, que serán mostrados en el sistema OS. Al igual que los estados, los mensajes se generan a partir de una OR lógica de los mensajes de origen.

En el anexo 2.18 *FB556:ZITDEV* se puede consultar el código completo.

1.8.7.3.8 FB557: ZITALM (FB Zona)

El bloque accede al FB ZITONP2 por medio de un puntero definido en su parámetro de entrada *Zone_STS*. Este, debe conectarse a uno de los parámetros de salida *Zonexx_STS* del FB ZITONP2, donde xx corresponde a la zona de la CDI donde está configurado el módulo.

Cuando se produce un evento de la zona asociada, el bloque envía un mensaje de alarma al sistema OS que puede ser monitorizado en pantalla. Este mensaje permanece activo mientras el evento se mantenga o los bits correspondientes a los eventos en el parámetro *AckZonesSTS[Zone]* del FB ZITONP2 estén a 0. Una vez estos bits pasen a 1, será indicativo de que el FB Módulo que provocó el evento ha generado su propio mensaje de alarma. Este mensaje al ser más concreto, hace que el de zona se vuelva trivial y, por lo tanto, el bloque lo desactive.

El bloque es capaz de generar 3 tipos de mensaje asociados a los eventos del módulo: Alarma, Pre-Alarma y Fallo, que serán mostrados en el sistema OS.

Este bloque está preparado para conectar con el bloque *FB559: ZITONCP* de tipo FB CDI, quedando fuera del alcance de este proyecto. La funcionalidad del *ZITONCP* es la misma que la del *ZITONP2*, pero la conexión con la CDI se realiza por el puerto de la tarjeta de comunicaciones CP 343-1 en lugar del puerto de la propia CPU. Esto supone un cambio en la manera de trabajar con el puerto, lo que implica una modificación del código que hace desplazar las zonas de memoria del FB CDI donde accede *ZITONP2*.

En el anexo 2.19 *FB557:ZITALM* se puede consultar el código completo.

1.8.7.4 Librería de Objetos Gráficos Block Icons

Los bloques descritos en los apartados anteriores tienen asociados objetos gráficos (*Block Icon*), y pantallas de detalle (*faceplate*) a excepción del FB ZITALM. Estos objetos pueden ser insertados en las pantallas del sistema OS de forma automática durante la compilación del proyecto OS. Esto se consigue gracias a que cada FB dispone de un parámetro de configuración donde se le puede indicar que *Block Icon* de la librería tiene asociado (Fig.45)

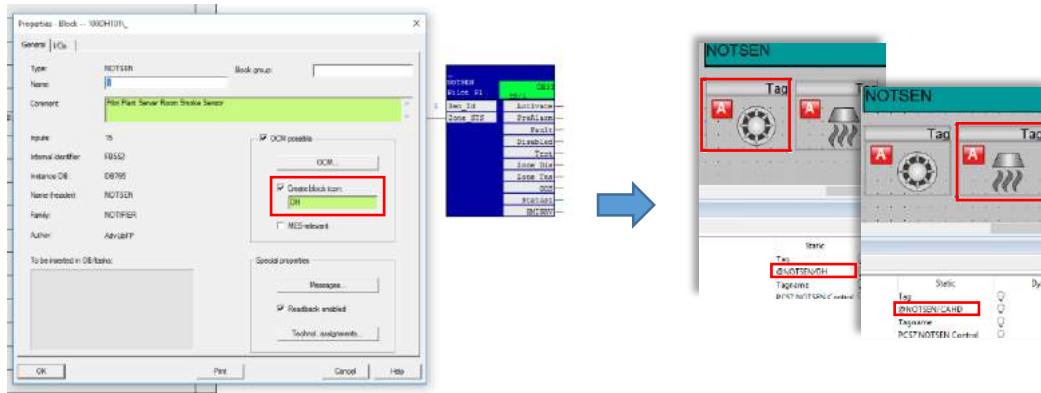


Figura 45. Asociación del block icon para FB NOTSEN.

El *Block Icon* presenta de forma dinámica el estado del FB asignado mediante los objetos gráficos que lo componen. Estos objetos pueden tener scripts asociados (pequeños programas escritos en C o en VBA) que se ejecutan ante un evento del bloque (como el cambio de un valor), un evento de pantalla (al clicar sobre el block icon), un evento del sistema (el cambio de usuario), etc., quedando fuera del alcance de esta memoria, analizar y detallar cada uno de ellos.

En el anexo 2.20 *Librería Block Icons*, se puede ver la librería completa que forma parte del sistema.

1.8.7.4.1 BI Módulo y BI Sensor

Englobamos estos tipos de *Block Icon* en un único apartado por compartir la misma representación gráfica (Fig.46), estando compuesto por:



Figura 46. Block Icon Módulo/Sensor.

- Nombre del Módulo/Sensor (Se puede mostrar u ocultar con la tecla F12)



- Eventos de Mensaje (Transparente si no hay)

	Fallo
	PreAlarma (Warning)
	Alarma
	Mensajes Anulados

- Nota de texto (Configurable en el *faceplate* permite dejar una anotación)



Nota

- Representación del Módulo/Sensor (Asociado al estado de Activación/Alarma)

On	Off	Descripción
		Detector de humo
		Pulsador de emergencia
		Sirena
...

- Estados del Módulo/Sensor (Transparente si no hay)

	Fuera de Servicio (Fallo comunicaciones CDI)
	Zona Deshabilitada
	Zona en Test
	Módulo/Sensor en Fallo
	Módulo/Sensor Deshabilitado
	Módulo/Sensor en Test

1.8.7.4.2 BI Dispositivo

Este tipo de BI está asociado, normalmente, a centrales externas (aspiración de humos, extinción, etc.) o cuadros de control de maniobra (Bombas contra Incendios, etc.). A diferencia de los módulos o sensores, su representación gráfica es genérica (Fig.47), indicando las siglas del tipo de dispositivo.



Figura 47. Block Icon Dispositivo.

- Nombre del Dispositivo (Se puede mostrar u ocultar con la tecla F12)



- Eventos de Mensaje (Transparente si no hay)

Representa una OR lógica de los eventos de los bloques individuales que lo conforman.

	Fallo
	PreAlarma (Warning)
	Alarma
	Mensajes Anulados

- Nota de texto (Configurable en el *faceplate* permite dejar una anotación)



Nota

- Estados del Dispositivo (Transparente si no hay)

Representa una OR lógica de los estados de los bloques individuales que lo conforman.

	Fuera de Servicio (Fallo comunicaciones CDI)
	Alguna Zona asociada Deshabilitada
	Alguna Zona asociada en Test
	Dispositivo en Fallo
	Dispositivo Deshabilitado
	Dispositivo en Test

- Representación del estado general del dispositivo

Asociado al estado de activación/alarma más relevante de los bloques individuales que lo conforman. La prioridad viene dada por el número de conexión al FB, siendo *STIn1* el de mayor prioridad y pudiendo configurar el texto a mostrar y el color, para cada conexión. El texto por defecto (Fig. 48) será mostrado cuando no hay ningún bloque con su estado de activación a 1. A modo de ejemplo, para un FB Dispositivo tipo CEX (central de extinción), su entrada *STIn1* está unida a un FB Módulo que indica extinción activada, su entrada *STIn2* está unida a un FB Módulo que indica CEX en modo manual, etc. En el caso de poner el CEX en manual se mostraría (Fig.48) (a). Si en manual disparamos la extinción se mostraría (Fig.48) (b), siguiendo el orden de prioridad descrito.

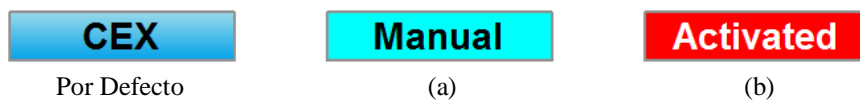


Figura 48. Estado General del Dispositivo.

1.8.7.4.3 BI CDI

Este tipo de *Block Icon* presenta el mismo aspecto que un BI Dispositivo, sin embargo, los textos y colores de su estado general están prefijados, y la codificación de los iconos de estados siguen un criterio diferente (Fig.49).







Figura 49. Block Icon CDI.

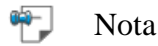
- Nombre del Dispositivo (Se puede mostrar u ocultar con la tecla F12)



- Eventos de Mensaje (Transparente si no hay)

	Fallo
	Alarma de Fuego o de Evacuación
	Mensajes Anulados
	Comunicaciones Deshabilitadas






- Nota de texto (Configurable en el *faceplate* permite dejar una anotación)









Nota

- Estados de la CDI (Transparente si no hay)

Para los FB NOTIF60:

	<p>Avisos relacionados con elementos de campo:</p> <ul style="list-style-type: none"> • Pre-Alarma General • Algún Elemento de Campo Deshabilitado • Sirenas Retardadas • Sirenas Silenciadas
	Fallo en el Panel CDI
	<p>CDI en modo Ingeniería:</p> <p>Avisa de una posible manipulación local en la configuración de la CDI.</p>
	CDI o Zona en Test.
	Comunicaciones Deshabilitadas

Para los FB ZITONP2:

	<p>Relacionados con estados de Bocinas, Protección de Fuego y Bomberos:</p> <ul style="list-style-type: none"> • Deshabilitado • Retardado
	<p>Relacionados con estados de Bocinas, Protección de Fuego y Bomberos:</p> <ul style="list-style-type: none"> • En Test
	<p>Relacionado con condiciones funcionales:</p> <ul style="list-style-type: none"> • Fallo
	<p>Relacionado con condiciones funcionales:</p> <ul style="list-style-type: none"> • Deshabilitado
	<p>Relacionado con condiciones funcionales:</p> <ul style="list-style-type: none"> • En Test
	Comunicaciones Deshabilitadas

- Representación del estado general de la CDI ordenados por prioridad

Para los FB NOTIF60:

Conn. Error	Error de Conexión con la CDI
EVACUATION	Estado de Alarma de Evacuación
FIRE	Estado de Alarma de Fuego
PreAlarm	Estado de Pre-Alarma
FAULT	Estado de Fallo
CDI	Sin Estado

Para los FB ZITONP2:

Conn. Error	Error de Conexión con la CDI
ALARM	Alarma General
MPC Alarm	Alarma por Pulsador
FAULT	Estado de Fallo
CDI	Sin Estado

1.8.7.5 Librería de Objetos Gráficos Faceplates

Cada uno de los *Block Icons* descritos en el apartado anterior dan acceso a sus pantallas de detalle denominadas *Faceplates* (Fig. 50), las cuales permiten configurar y actuar sobre el FB asociado, así como mostrar funcionalidades adicionales que no pueden ser representadas en el *Block Icon*. Al igual que los *Block Icons*, los *Faceplates* están compuestos por objetos dinámicos, los cuales también pueden tener scripts asociados.

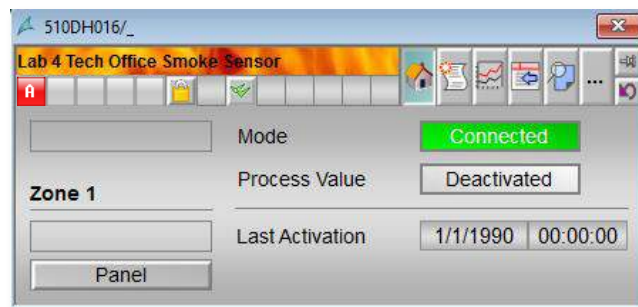
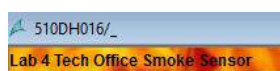


Figura 50. Ejemplo de Faceplate.





La estructura de los *Faceplates* sigue el estilo de diseño del resto del sistema, siendo la parte superior común a todos ellos. En esta cabecera podemos distinguir los siguientes elementos:

- Nombre del FB y descripción



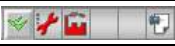

- Eventos de Mensaje

En cada cuadro se muestra un tipo de mensaje, pudiéndose ver al mismo tiempo eventos diferentes, a diferencia de los *Block Icons* donde solo se muestra uno según su grado de importancia:








	Alarma
	Aviso
	Fallo
	Petición al operador (No utilizado)

- Estados



Con la misma representación que en los *Block Icons*, pudiéndose ver al mismo tiempo estados diferentes.

	(Ver estados <i>Block Icons</i>)
	Reconocimiento de Alarmas

- Botones de Navegación

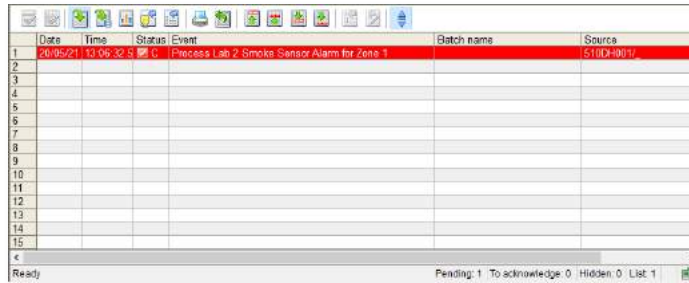
	Pantalla Principal
	Listado de Mensajes
	Gráficas
	Parámetros
	Vista Avanzada
	Muestra siguiente conjunto de botones
	Notas

- Botones Auxiliares

	Evita que se cierre el <i>Faceplate</i> en el cambio de pantalla.
	Permite regresar a la pantalla donde se encuentra el <i>Block Icon</i> asociado.

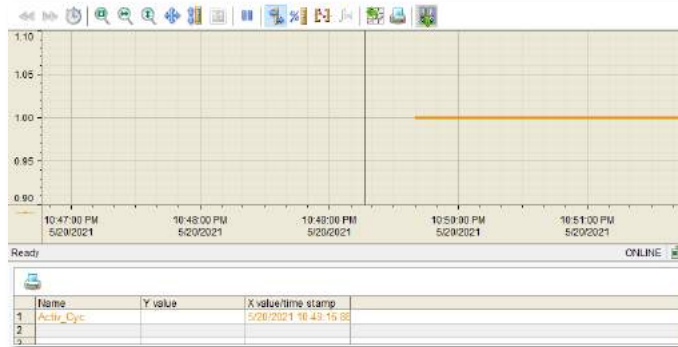
Algunas de las pantallas de navegación son también comunes, entre las que encontramos:


- Listado de Mensajes
Muestra el listado de mensajes asociados al FB

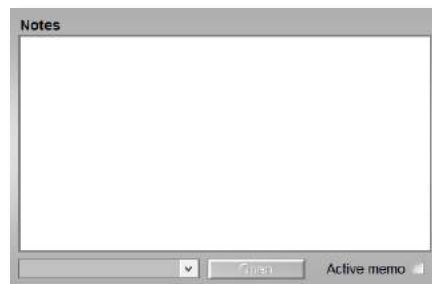


Date	Time	Status	Event	Batch name	Source
20/05/21	13:09:32	5	Process Lab 2 Smoke Sensor Alarm for Zone 1		511(LX-001)

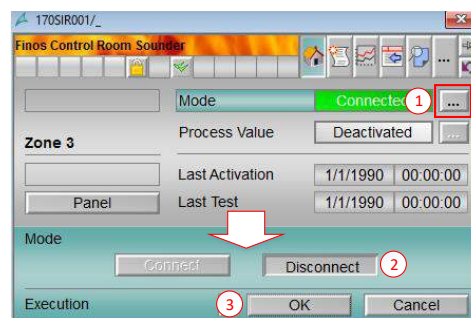
- Gráficas
Muestra una gráfica con valores asociados al FB



- Notas
Permite escribir un texto libre y activarlo para que aparezca el icono  informando.



Los botones de órdenes hacia el FB necesitan de una acción de 3 pasos para evitar errores.



1.8.7.5.1 Faceplate NOTIF60

- Pantalla Principal



1. Último Evento recibido desde la CDI.
Los eventos pueden consultarse al final del anexo 2.7. *Pasarela IBOX MB Server.*

2. Estados Generales de la CDI.

3. Para cada Zona, estados de Test, Alarma o Pre-Alarma

	Estado no activo
	Zona en Test
	Zona en Alarma
	Zona en Pre-Alarma

4. Para cada Zona, estado de Fallo

	Estado no activo
	Zona en Fallo

5. Para cada Zona, estados de Deshabilitado Total o Parcia

	Estado no activo
	Zona Totalmente Deshabilitada
	Parcialmente Deshabilitada

6. Para cada Zona, descripción de la Zona.

7. Para cada Zona, orden para Iniciar/Parar Test de la Zona.

8. Para cada Zona, orden para Habilitar/Deshabilitar la Zona.

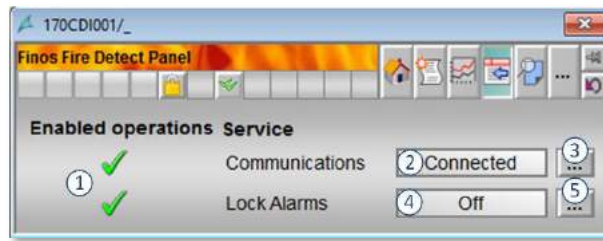
9. Estado de la conexión con la CDI

	Conexión Habilitada
	Conexión Deshabilitada
	Error de conexión con el IBOX
	Error de conexión del IBOX con la CDI

10. Órdenes Generales a la CDI.

11. Muestra Conjunto de Zonas 1-16 / 17-32.

- Pantalla de Parámetros



1. Permisos de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Modo de Funcionamiento

Connected	Conexión Habilitada
Disconnected	Conexión Deshabilitada

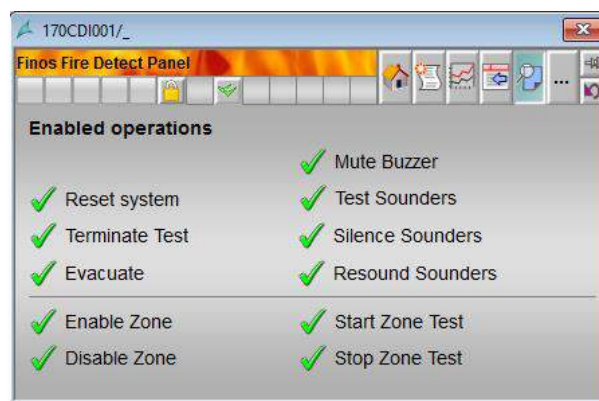
3. Orden para Habilitar/Deshabilitar la conexión con la CDI.

4. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

5. Orden para Bloquear/Desbloquear Mensajes.

- Pantalla de Vista Avanzada

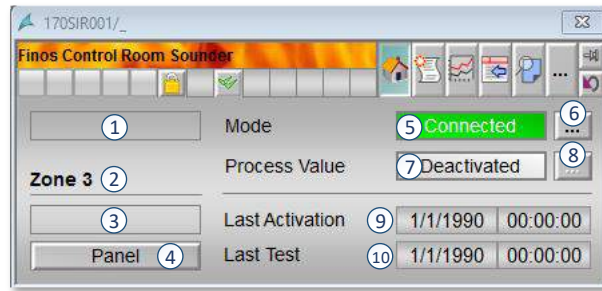


1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

1.8.7.5.2 Faceplate NOTMOD

- Pantalla Principal



1. Estado del Módulo

PreAlarm	Módulo en Pre-Alarma
Fault	Módulo en Fallo
Test	Módulo en Test

2. Número de zona asignado.

3. Estado de la zona

Test	Zona en Test
Disabled	Zona Deshabilitada

4. Abre el *Faceplate* de la CDI de la que depende.

5. Modo de Funcionamiento.

Connected	Módulo Habilitado
Disconnected	Módulo Deshabilitado
Out of Service	Fuera de Servicio (Fallo comunicaciones CDI)

6. Orden para Habilitar/Deshabilitar el Módulo.

7. Valor de la Entrada o Salida del Módulo.

Deactivated	Desactivada
Activated	Activada (Sin alarma asociada)
Activated	Activada (Con alarma asociada)

8. Orden para Activar/Desactivar la Salida del Módulo cuando está en Test.
Si es un Módulo de Entrada, el botón no aparece.

9. Con fines de mantenimiento, registro de la última activación fuera de Test.

10. Con fines de mantenimiento, registro de la última activación en Test.

- Pantalla de Parámetros



1. Permisos de Operación

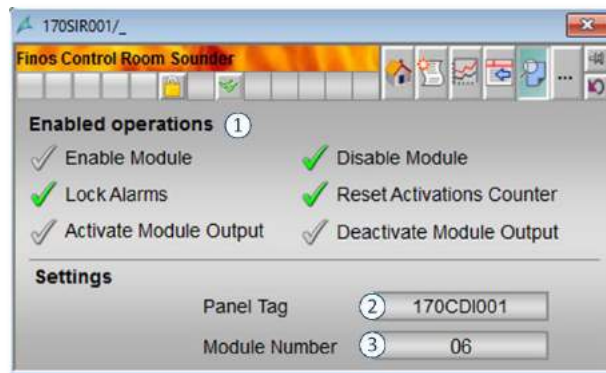
✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

3. Orden para Bloquear/Desbloquear Mensajes.
4. Con fines de mantenimiento, registro del número de Activaciones del Módulo.
5. Orden para poner a cero el registro del número de Activaciones del Módulo.

- Pantalla de Vista Avanzada



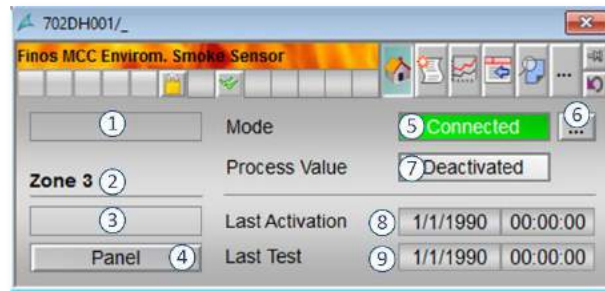
1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Nombre de la CDI de la que depende.
3. Número de Módulo configurado en la CDI.

1.8.7.5.3 Faceplate NOTSEN

- Pantalla Principal



1. Estado del Sensor

PreAlarm	Sensor en Pre-Alarma
Fault	Sensor en Fallo
Test	Sensor en Test

2. Número de zona asignado.

3. Estado de la zona

Test	Zona en Test
Disabled	Zona Deshabilitada

4. Abre el *Faceplate* de la CDI de la que depende.

5. Modo de Funcionamiento.

Connected	Sensor Habilitado
Disconnected	Sensor Deshabilitado
Out of Service	Fuera de Servicio (Fallo comunicaciones CDI)

6. Orden para Habilitar/Deshabilitar el Sensor.

7. Valor de la Entrada del Sensor.

Deactivated	Desactivada
Activated	Activada

8. Con fines de mantenimiento, registro de la última activación fuera de Test.

9. Con fines de mantenimiento, registro de la última activación en Test.

- Pantalla de Parámetros



1. Permisos de Operación

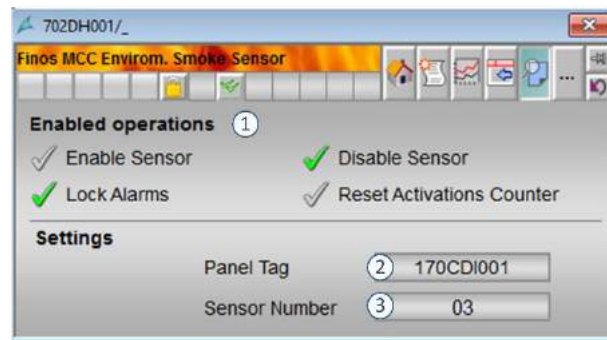
✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

3. Orden para Bloquear/Desbloquear Mensajes.
4. Con fines de mantenimiento, registro del número de Activaciones del Sensor.
5. Orden para poner a cero el registro del número de Activaciones del Sensor.

- Pantalla de Vista Avanzada



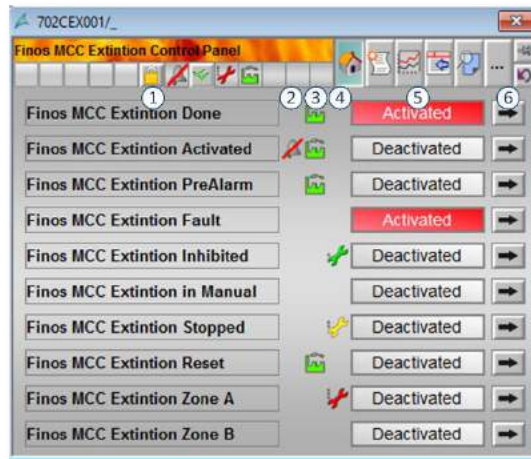
1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Nombre de la CDI de la que depende.
3. Número de Sensor configurado en la CDI.

1.8.7.5.4 Faceplate NOTDEV

- Pantalla Principal



1. Descripción del Módulo.

2. Estado de Mensajes del Módulo

	Mensajes no Anulados
	Mensajes Anulados

3. Estados de la Zona asociada al Módulo.

	Fuera de Servicio (Fallo comunicaciones CDI)
	Zona Deshabilitada
	Zona en Test

4. Estados del Módulo.

	Módulo en Fallo
	Módulo Deshabilitado
	Módulo en Test

5. Valor de la Entrada o Salida del Módulo.

	Desactivada
	Pre-Alarma
	Activada (Sin alarma asociada)
	Activada (Con alarma asociada)

6. Abre el *Faceplate* del Módulo.

- Pantalla de Parámetros



1. Permisos de Operación

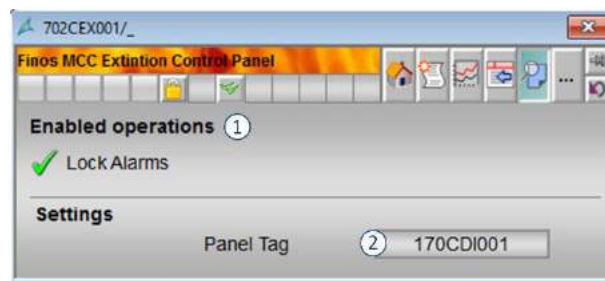
✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

3. Orden para Bloquear/Desbloquear Mensajes.

- Pantalla de Vista Avanzada



1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Nombre de la CDI de la que depende.

1.8.7.5.5 Faceplate ZITONP2

- Pantalla Principal



1. Estados Generales de la CDI.
2. Estado de la conexión con la CDI
3. Condiciones Funcionales.
4. Estados Bocinas.
5. Estados Bomberos.
6. Estados Protección Fuego.
7. Para cada Zona, estados de Test, Alarma o Pre-Alarma

	Conexión Habilitada
	Conexión Deshabilitada
	Error de conexión

8. Para cada Zona, estado de Fallo

	Estado no activo
	Zona en Fallo

9. Para cada Zona, estados de Deshabilitado Total

	Estado no activo
	Zona Totalmente Deshabilitada

10. Para cada Zona, descripción de la Zona.
11. Muestra Conjunto de Zonas 1-16 / 17-32.

- Pantalla de Parámetros



1. Permisos de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Modo de Funcionamiento

Connected	Conexión Habilitada
Disconnected	Conexión Deshabilitada

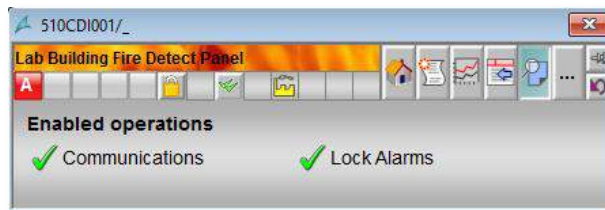
3. Orden para Habilitar/Deshabilitar la conexión con la CDI.

4. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

5. Orden para Bloquear/Desbloquear Mensajes.

- Pantalla de Vista Avanzada

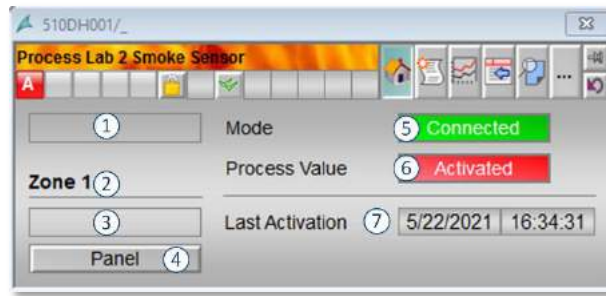


1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

1.8.7.5.6 Faceplate ZITMOD

- Pantalla Principal



1. Estado del Módulo

PreAlarm	Módulo en Pre-Alarma
Fault	Módulo en Fallo
Test	Módulo en Test

2. Número de zona asignado

3. Estado de la zona

Test	Zona en Test
Disabled	Zona Deshabilitada

4. Abre el *Faceplate* de la CDI de la que depende.

5. Modo de Funcionamiento.

Connected	Módulo Habilitado
Disconnected	Módulo Deshabilitado
Out of Service	Fuera de Servicio (Fallo comunicaciones CDI)

6. Valor de la Entrada o Salida del Módulo.

Deactivated	Desactivada
Activated	Activada (Sin alarma asociada)
Activated	Activada (Con alarma asociada)

7. Con fines de mantenimiento, registro de la última activación fuera de Test.

- Pantalla de Parámetros



1. Permisos de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

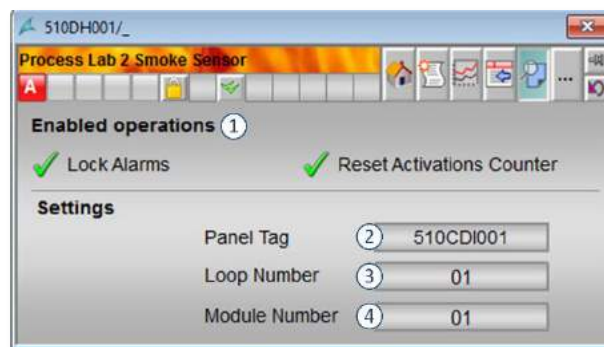
2. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

3. Orden para Bloquear/Desbloquear Mensajes

4. Con fines de mantenimiento, registro del número de Activaciones del Módulo.
5. Orden para poner a cero el registro del número de Activaciones del Módulo.

- Pantalla de Vista Avanzada



1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Nombre de la CDI de la que depende.
3. Número de Lazo configurado en la CDI.
4. Número de Módulo configurado en la CDI.

1.8.7.5.7 Faceplate ZITDEV

- Pantalla Principal



1. Descripción del Módulo.

2. Estado de Mensajes del Módulo

	Mensajes no Anulados
	Mensajes Anulados

3. Estados de la Zona asociada al Módulo.

	Fuera de Servicio (Fallo comunicaciones CDI)
	Zona Deshabilitada
	Zona en Test

4. Estados del Módulo.

	Módulo en Fallo
	Módulo Deshabilitado
	Módulo en Test

5. Valor de la Entrada o Salida del Módulo.

	Desactivada
	Pre-Alarma
	Activada (Sin alarma asociada)
	Activada (Con alarma asociada)

6. Abre el *Faceplate* del Módulo.

- Pantalla de Parámetros



1. Permisos de Operación

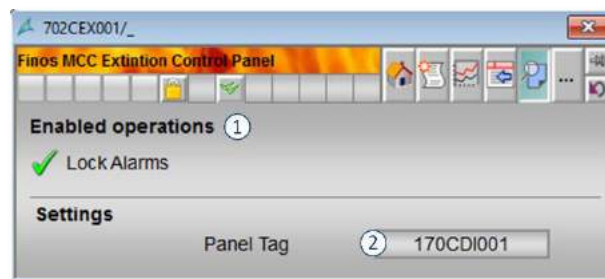
✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Estado del Bloqueo de Mensajes

Off	Mensajes no Bloqueados (se muestran)
On	Mensajes Bloqueados (no se muestran)

3. Orden para Bloquear/Desbloquear Mensajes.

- Pantalla de Vista Avanzada

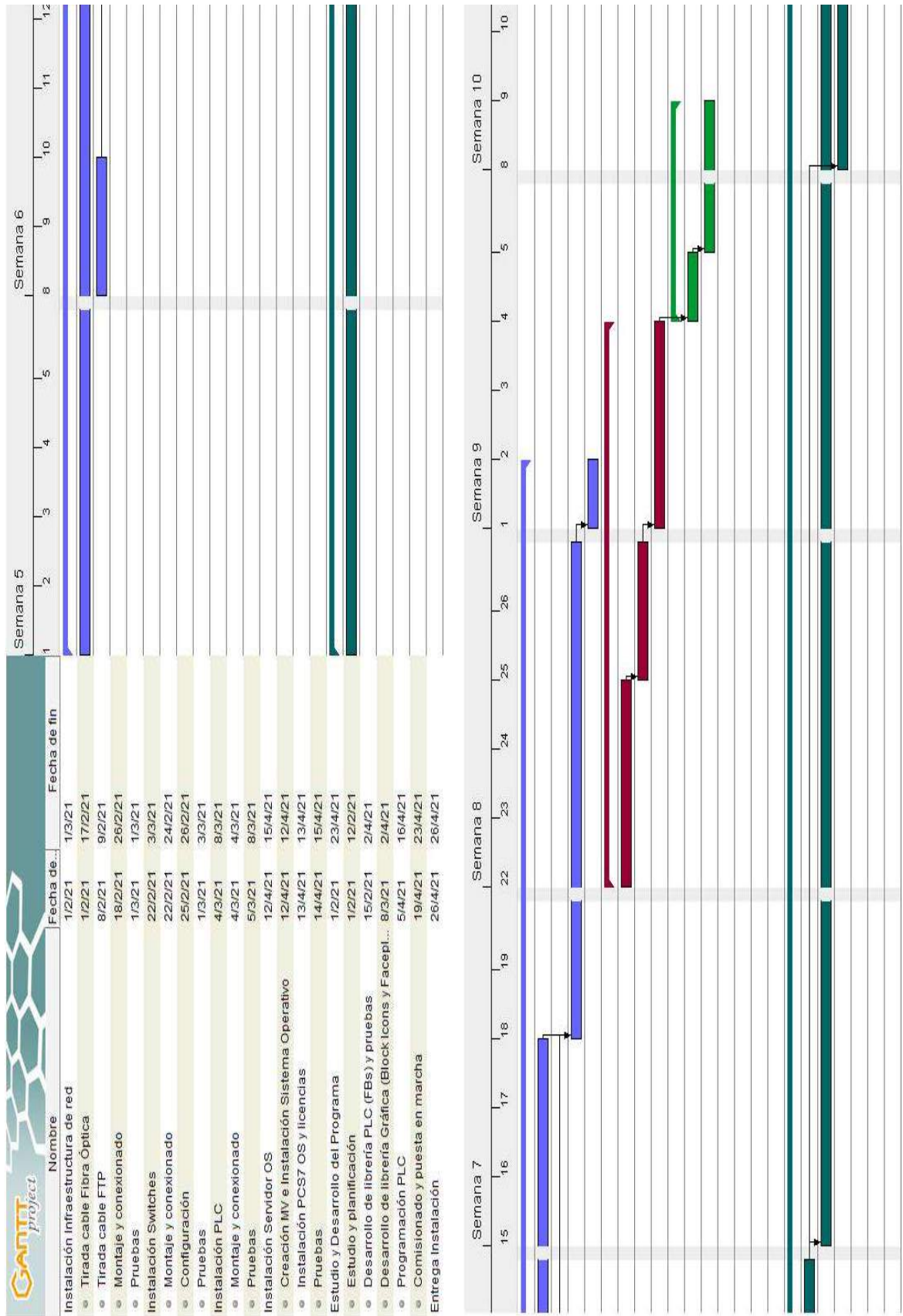


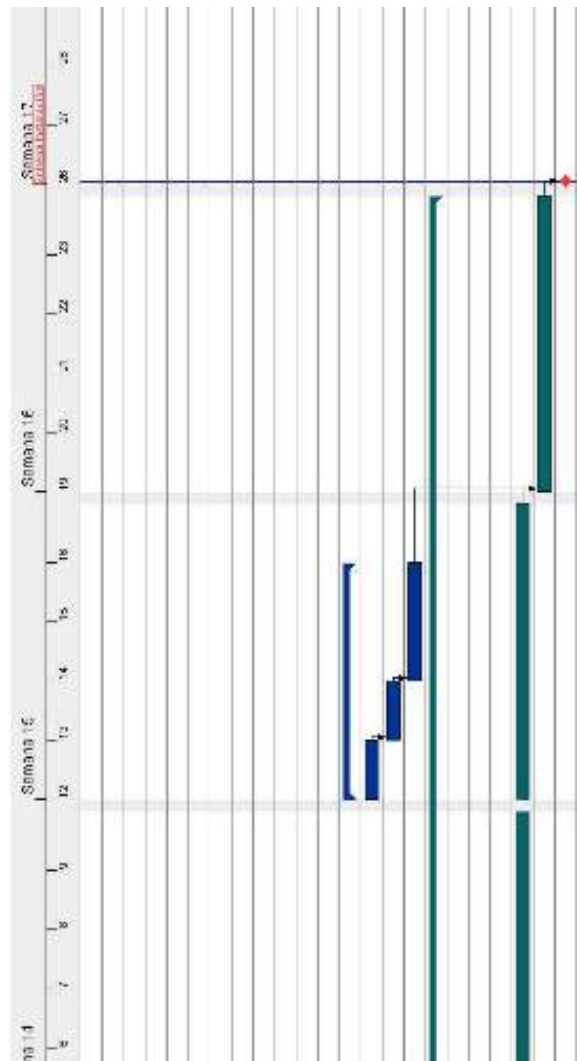
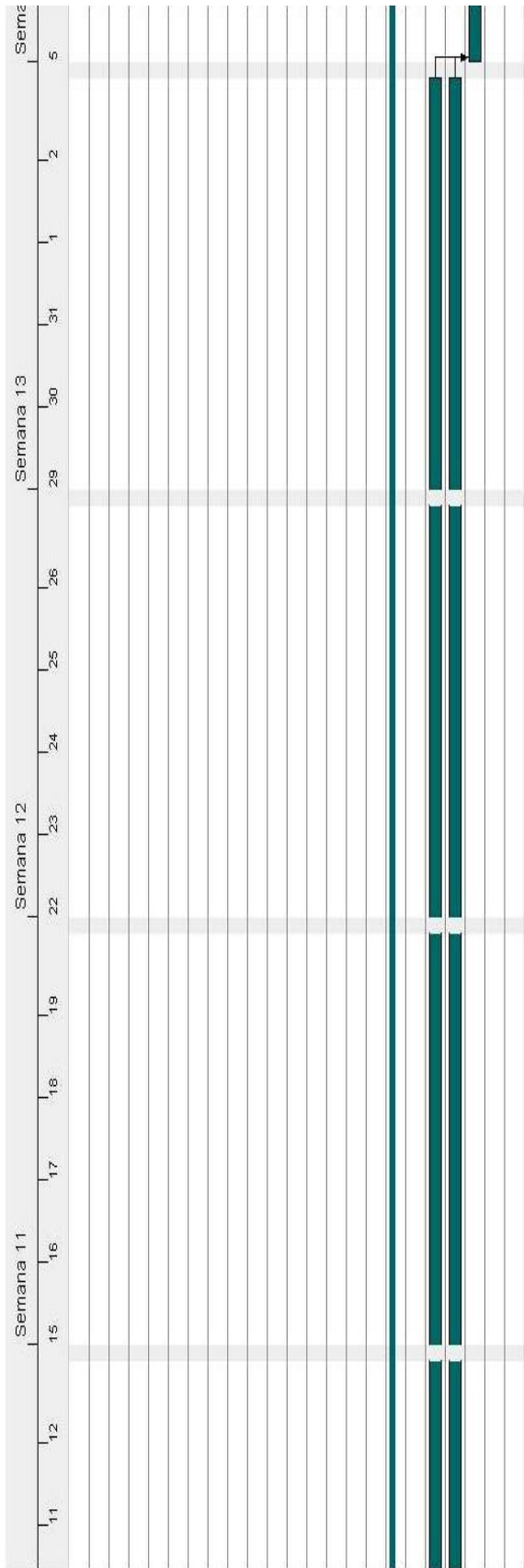
1. Permisos Globales de Operación

✓	Parámetro controlable
✓	Parámetro no controlable en este momento
✗	Parámetro no controlable

2. Nombre de la CDI de la que depende.

1.9 Planificación





Monitorización de un Sistema de Detección de Incendios

ANEXOS



UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2021

2 Anexos

2.1	Direccionamiento Plant Bus.....	94
2.2	Switch SCALANCE XC206-2SFP.....	95
2.3	Certificación Transceptor SFP992-1.....	107
2.4	CPU 317-2 PN/DP.....	111
2.5	CP 343-1.....	120
2.6	PS 307 (5 A).....	123
2.7	Pasarela IBOX Modbus Server.....	126
2.8	Ziton Fire Panel Modbus Protocol.....	166
2.9	PCS7: Configuración de Hardware y Redes.....	177
2.10	PCS7: Jerarquía de Planta.....	182
2.11	PCS7: Jerarquía en Pantallas del Sistema.....	185
2.12	FB550: NOTIF60.....	190
2.13	FB551: NOTMOD.....	210
2.14	FB552: NOTSEN.....	220
2.15	FB553: NOTDEV.....	230
2.16	FB554: ZITONP2.....	238
2.17	FB555: ZITMOD.....	255
2.18	FB556: ZITDEV.....	265
2.19	FB557: ZITALM.....	273
2.20	PCS7: Librería Block Icons.....	278

2.1 Direccionamiento Plant Bus

IP Address	SERVICES		FINOS PLANT		ACETATES		SYN1		SYN2		SFFC		RIBH	
	Host	Description	Host	Description	Host	Description	Host	Description	Host	Description	Host	Description	Host	Description
172.016.xxx.001	NTL	NTL	NS0201	SCALANCE_XC206-25FP	NS0201	SCALANCE_XC206-25FP	NS0301	SCALANCE_XC206-25FP	NS0301	SCALANCE_XC206-25FP	NS0501	SCALANCE_XC206-25FP	NS0601	SCALANCE_XC206-25FP
172.016.xxx.002	NTL	NTL	NS0202	SCALANCE_XC206-25FP	NS0202	SCALANCE_XC206-25FP	NS0302	SCALANCE_XC206-25FP	NS0302	SCALANCE_XC206-25FP				
172.016.xxx.003	NTL	NTL	NS0202	SCALANCE_XC206-25FP	NS0202	SCALANCE_XC206-25FP	NS0302	SCALANCE_XC206-25FP	NS0302	SCALANCE_XC206-25FP				
172.016.xxx.004	NTL	NTL	NS0202	SCALANCE_XC206-25FP	NS0202	SCALANCE_XC206-25FP	NS0302	SCALANCE_XC206-25FP	NS0302	SCALANCE_XC206-25FP				
172.016.xxx.031	Domain Server	Domain Server	ESBPCS70501	Acct. OS/BE primary	ESBPCS70501	Acct. OS/BE primary	ESBPCS70502	Syn1 OS primary						
172.016.xxx.032	Domain Server	Domain Server	ESBPCS70502	Acct. OS/BE backup	ESBPCS70502	Acct. OS/BE backup	ESBPCS70503	Syn1 OS backup						
172.016.xxx.033	SMTP Proxy Server	SMTP Proxy Server	ESBPCS70503	Acct. ES	ESBPCS70503	Acct. ES	ESBPCS70504	Syn1 ES						
172.016.xxx.064			AS01CPUS	Acct. CPU 410-5H1 KS port1	AS02CPUS	Acct. CPU 410-5H1 KS port1	AS03CPUS	Syn1 CPU 410-5H1 KS port1						
172.016.xxx.065			AS01ETH7		AS02ETH7		AS03ETH7	Syn1 CP 443						
172.016.xxx.066			AS01ETH8		AS02ETH8		AS03ETH8	Syn1 CP 443						
172.016.xxx.126														
172.016.xxx.127														
172.016.xxx.128														
172.016.xxx.129														
172.016.xxx.251														
172.016.xxx.252														
172.016.xxx.253														
172.016.xxx.254														
172.016.xxx.254	Fire Protect Panel Error	Fire Protect Panel Error	130CD001		130CD001		130CD001	Fire Protect Panel Lab Syn1					130CD001	Fire Protect Panel Error

IP Address	NPP		EFFLUENT		NATUBALS		TURPERTINE		LAB BUILDING		OFFICE BUILDING			
	Host	Description	Host	Description	Host	Description	Host	Description	Host	Description	Host	Description		
172.016.xxx.001	NS0701	SCALANCE_XC206-25FP	NS0801	SCALANCE_XC206-25FP	NS0901	SCALANCE_XC206-25FP	NS1001	SCALANCE_XC206-25FP	NS1101	SCALANCE_XC206-25FP	NS1201	SCALANCE_XC206-25FP		
172.016.xxx.002	NS0702	SCALANCE_XC206-25FP	NS0702	SCALANCE_XC206-25FP	NS0901	SCALANCE_XC206-25FP	NS1001	SCALANCE_XC206-25FP	NS1101	SCALANCE_XC206-25FP	NS1201	SCALANCE_XC206-25FP		
172.016.xxx.003	NS0702	SCALANCE_XC206-25FP	NS0702	SCALANCE_XC206-25FP	NS0901	SCALANCE_XC206-25FP	NS1001	SCALANCE_XC206-25FP	NS1101	SCALANCE_XC206-25FP	NS1201	SCALANCE_XC206-25FP		
172.016.xxx.004	NS0702	SCALANCE_XC206-25FP	NS0702	SCALANCE_XC206-25FP	NS0901	SCALANCE_XC206-25FP	NS1001	SCALANCE_XC206-25FP	NS1101	SCALANCE_XC206-25FP	NS1201	SCALANCE_XC206-25FP		
172.016.xxx.031														
172.016.xxx.032														
172.016.xxx.033														
172.016.xxx.034														
172.016.xxx.035														
172.016.xxx.036														
172.016.xxx.064														
172.016.xxx.065														
172.016.xxx.066														
172.016.xxx.067														
172.016.xxx.068														
172.016.xxx.126														
172.016.xxx.127														
172.016.xxx.128														
172.016.xxx.129														
172.016.xxx.251														
172.016.xxx.252														
172.016.xxx.253														
172.016.xxx.254														
172.016.xxx.254	Fire Protect Panel P10K P.	Fire Protect Panel P10K P.	110CD101		110CD101		110CD101	Fire Protect Panel Lab Syn1					110CD101	Fire Protect Panel Office B.

2.2 Switch SCALANCE XC206-2SFP

SIEMENS

Hoja de datos

6GK5206-2BS00-2AC2



SCALANCE XC206-2SFP switch IE gestionable capa 2; con certificación IEC 62443-4-2; 6 puertos RJ45 10/100 Mbits/s; 2 puertos SFP 100/1000 Mbits/s; 1 puerto de consola; LED de diagnóstico; alimentación redundante; rango de temperaturas de -40 °C a +70 °C; montaje: perfil DIN/soporte S7/pared funciones de redundancia Office Features (RSTP, VLAN, ...); dispositivo PROFINET IO; conforme con Ethernet/IP; ranura para C-PLUG;

denominación del tipo de producto	SCALANCE XC206-2SFP
tasa de transferencia	
tasa de transferencia	10 Mbit/s, 100 Mbit/s, 1000 Mbit/s
interfaces / para comunicaciones / integrada	
número de conexiones eléctricas	
• para componentes de red o equipos terminales	6; RJ45
número de puertos RJ45 a 10/100 Mbits/s / integrada	
• con collar de sujeción	6
interfaces / para comunicaciones / enchufables	
número de conexiones eléctricas	
• para SFP	2; Transceptor enchufable SFP 100 Mbits/s o 1000 Mbits/s
interfaces / otras	
número de conexiones eléctricas	
• para consola de mando	1
• para contacto de señalización	1
• para alimentación	1
tipo de conexión eléctrica	
• para consola de mando	RJ11
• para contacto de señalización	Bloque de bornes de 2 polos
• para alimentación	Bloque de bornes de 4 polos
tipo de soporte de datos intercambiable	
• C-PLUG	Sí
entradas/salidas de señal	
tensión de empleo / de los contactos de señalización	
• con DC / valor nominal	24 V
intensidad de empleo / de los contactos de señalización	
• con DC / máx.	0,1 A
tensión de alimentación, consumo de corriente, pérdidas	
componente del producto / conexión para alimentación redundante	Sí
tipo de corriente / 1 / de la tensión de alimentación	DC
• tensión de alimentación / 1 / valor nominal	24 V
• pérdidas [W] / 1 / valor nominal	6 W
• corriente consumida / 1 / con valor nominal de la tensión de alimentación / máx.	0,25 A
• tensión de alimentación / 1 / valor asignado	9,6 ... 31,2 V
• corriente consumida / 1 / máx.	0,5 A
• tipo de conexión eléctrica / 1 / para alimentación	Bloque de bornes de 4 polos

<ul style="list-style-type: none"> componente del producto / 1 / protección con fusibles en entrada de alimentación 	Sí
condiciones ambientales	
temperatura ambiente	
<ul style="list-style-type: none"> durante el funcionamiento durante el almacenamiento durante el transporte 	-40 ... +70 °C -40 ... +85 °C -40 ... +85 °C
humedad relativa del aire	
<ul style="list-style-type: none"> con 25 °C / sin condensación / durante el funcionamiento / máx. 	95 %
grado de protección IP	IP20
diseño, dimensiones y pesos	
forma constructiva	Diseño compacto
anchura	60 mm
altura	147 mm
profundidad	125 mm
peso neto	0,52 kg
materia / de la caja	Polycarbonato (PC-GF10) / fundición inyectada de aluminio
tipo de fijación	
<ul style="list-style-type: none"> montaje sobre perfil DIN de 35 mm montaje en pared montaje en perfil soporte S7-300 montaje en perfil soporte S7-1500 	Sí Sí Sí Sí
propiedades, funciones y componentes del producto / general	
conexión en cascada con anillo redundante / en tiempo de reconfiguración < 0,3 s	50
conexión en cascada con topología en estrella	libre (sólo depende del tiempo de propagación de la señal)
función del producto / QoS según DSCP	Sí
propiedad del producto	
<ul style="list-style-type: none"> método de conmutación Cut Through método de conmutación Store & Forward 	No Sí
funciones del producto / gestión, configuración, ajustes	
función del producto	
<ul style="list-style-type: none"> CLI gestión basada en web soporte de MIB TRAP vía e-mail configuración con STEP 7 RMON servidor SMTP Portmirroring Mirroring multipuerto CoS diagnóstico PROFINET IO 	Sí Sí Sí Sí Sí Sí No Sí Sí Sí Sí Sí
clase de conformidad PROFINET	B
función del producto / gestionada por switch	Sí
longitud de telegrama / con Ethernet / máx.	1632 byte
protocolo / soportado	
<ul style="list-style-type: none"> Telnet HTTP HTTPS TFTP BOOTP GMPR DCP LLDP EtherNet/IP SNMP v1 	Sí Sí Sí Sí No Sí Sí Sí Sí Sí

• SNMP v2	Sí
• SNMP v3	Sí
• IGMP (Snooping/Querier)	Sí
función de Identificación y Mantenimiento	
• I&M0 - Información específica del dispositivo	Sí
• I&M1 - ID de la instalación/ID de situación	Sí
funciones del producto / diagnóstico	
función del producto	
• Port Diagnostics	Sí
• Packet Size Statistics	Sí
• Packet Type Statistics	Sí
• Error Statistics	Sí
• SysLog	Sí
funciones del producto / VLAN	
función del producto	
• VLAN - port based	Sí
• VLAN - protocol based	No
• VLAN - IP based	No
número VLAN / máx.	257
número VLAN - dynamic / máx.	257
protocolo / soportado / GVRP	Sí
funciones del producto / DHCP	
función del producto	
• servidor DHCP	Sí
• cliente DHCP	Sí
• DHCP opción 82	Sí
• DHCP opción 66	Sí
• DHCP opción 67	Sí
funciones del producto / redundancia	
función del producto	
• de PROFINET IO-Device / soportado / redundancia de sistema PROFINET	Sí
• redundancia de anillo	Sí
• procedimiento de redundancia HSR	Sí
• High Speed Redundancy Protocol (HRP) con gestor de redundancia	Sí
• High Speed Redundancy Protocol (HRP) con redundancia standby	Sí
protocolo / soportado / procedimiento de redundancia MRP	Sí
función del producto	
• Media Redundancy Protocol (MRP) con gestor de redundancia	Sí
• Media Redundancy Protocol Interconnection (MRP-I)	Sí
• de PROFINET IO-Device / soportado / H-Sync Forwarding	Sí
• procedimiento de redundancia STP	Sí
• procedimiento de redundancia RSTP	Sí
• procedimiento de redundancia RSTP+	Sí
• procedimiento de redundancia MSTP	Sí
• protocolo de redundancia en paralelo (PRP)/uso en red PRP	Sí
• protocolo de redundancia en paralelo (PRP)/Redundant Network Access (RNA)	No
• Passive Listening	Sí
protocolo / soportado	
• LACP	Sí
funciones del producto / seguridad	
función del producto	

<ul style="list-style-type: none"> • IEEE 802.1x (radio) 	Sí
<ul style="list-style-type: none"> • Broadcast/Multicast/Unicast Limiter 	Sí
<ul style="list-style-type: none"> • Broadcast Blocking 	Sí
protocolo / soportado	
<ul style="list-style-type: none"> • SSH 	Sí
<ul style="list-style-type: none"> • SSL 	Sí
funciones del producto / hora	
función del producto	
<ul style="list-style-type: none"> • soporte de SICLOCK 	Sí
<ul style="list-style-type: none"> • cliente NTP 	Sí
<ul style="list-style-type: none"> • cliente SNTP 	Sí
protocolo / soportado	
<ul style="list-style-type: none"> • NTP 	Sí
<ul style="list-style-type: none"> • SNTP 	Sí
modificación de la instalación durante el funcionamiento	
función del producto / configuración en RUN vía CiR/H-CiR	Sí
normas, especificaciones, homologaciones	
seguridad informática para sistemas de automatización industrial / según IEC 62443-4-2:2019	Sí
MTBF	47 y
normas, especificaciones, homologaciones / CE	
certificado de aptitud / marcado CE	Sí
conformidad del producto / según Directiva CEM	2014/30/EU
norma	
<ul style="list-style-type: none"> • para emisión de perturbaciones CEM 	EN 61000-6-4, EN 50121-4
<ul style="list-style-type: none"> • para inmunidad a perturbaciones CEM 	EN 61000-6-2, EN 50121-4
certificado de aptitud / conformidad con las Directivas sobre restricción de sustancias peligrosas (RoHS)	Sí; 2011/65/EU
normas, especificaciones, homologaciones / entornos peligrosos	
certificado de aptitud	
<ul style="list-style-type: none"> • ATEX 	Sí; EN 60079-0, EN 60079-15
<ul style="list-style-type: none"> • para ATEX / como marcado 	II 3 G Ex nA II T4 Gc
<ul style="list-style-type: none"> • para ATEX / como certificado 	KEMA 07ATEX0145 X
<ul style="list-style-type: none"> • IECEx 	Sí; IEC 60079-0, IEC 60079-15
<ul style="list-style-type: none"> • para IECEx / como marcado 	Ex nA II T4 Gc
<ul style="list-style-type: none"> • para IECEx / como certificado 	IECEX DEK 14.0025X
<ul style="list-style-type: none"> • para cULus HazLoc / como marcado 	CL1, DIV2, Group A,B,C,D T4, CL1, Zone 2, Group IIC, T4
<ul style="list-style-type: none"> • para cULus HazLoc / como File Nr. 	E240480 (NWHF, NWHF7)
<ul style="list-style-type: none"> • homologación FM 	Sí; FM Class 3600:2011, FM Class 3611:2004, FM Class 3810:2005, ANSI/ISA-61010-1:2004
<ul style="list-style-type: none"> • para FM / como marcado 	CL1, DIV2, Group A,B,C,D T4, CL1, Zone 2, Group IIC, T4 Ta: -40°C +70°C
<ul style="list-style-type: none"> • para FM / como certificado 	FM16US0205X
normas, especificaciones, homologaciones / otras	
certificado de aptitud	
<ul style="list-style-type: none"> • aplicaciones ferroviarias según EN 50121-4 	Sí
<ul style="list-style-type: none"> • Regulatory Compliance Mark (RCM) 	Sí
<ul style="list-style-type: none"> • para UL 61010-2-201 / como File Nr. 	E85972 (NRAQ, NRAQ7)
<ul style="list-style-type: none"> • para UL 60950-1 / como File Nr. 	E115352 (NWWGQ, NWWGQ7)
<ul style="list-style-type: none"> • homologación EAC 	Sí
normas, especificaciones, homologaciones / clasificación naval	
sociedad de clasificación naval	
<ul style="list-style-type: none"> • American Bureau of Shipping Europe Ltd. (ABS) 	Sí
<ul style="list-style-type: none"> • Bureau Veritas (BV) 	Sí
<ul style="list-style-type: none"> • DNV GL 	Sí
<ul style="list-style-type: none"> • Korean Register of Shipping (KRS) 	Sí
<ul style="list-style-type: none"> • Lloyds Register of Shipping (LRS) 	Sí
<ul style="list-style-type: none"> • Nippon Kaiji Kyokai (NK) 	Sí

<ul style="list-style-type: none"> • Polski Rejestr Statkow (PRS) • Royal Institution of Naval Architects (RINA) 	<p>Si</p> <p>Si</p>
accesorios	
ampliación del producto / opcional / C-PLUG	Si
información adicional / enlaces de Internet	
<p>enlace de Internet</p> <ul style="list-style-type: none"> • a la página web: Comunicación Industrial • a la página web: Industry Mall • a la página web: Centro de información y descarga • a la página web: Archivo gráfico • a la página web: CAX-Download-Manager • a la página web: Industry Online Support 	<p>http://www.siemens.com/simatic-net</p> <p>https://mall.industry.siemens.com</p> <p>http://www.siemens.com/industry/infocenter</p> <p>http://automation.siemens.com/bilddb</p> <p>http://www.siemens.com/cax</p> <p>https://support.industry.siemens.com</p>
información de seguridad	
información de seguridad	<p>Siemens suministra productos y soluciones con funciones de seguridad industrial que contribuyen al funcionamiento seguro de instalaciones, soluciones, máquinas, equipos y redes. Dichas funciones son un componente importante de un sistema global de seguridad industrial. En consideración de lo anterior, los productos y soluciones de Siemens son objeto de mejoras continuas. Por ello, le recomendamos que se informe periódicamente sobre las actualizaciones de nuestros productos. Para el funcionamiento seguro de los productos y soluciones de Siemens, es preciso tomar medidas de protección adecuadas (como el concepto de protección de células) e integrar cada componente en un sistema de seguridad industrial integral que incorpore los últimos avances tecnológicos. También deben tenerse en cuenta los productos de otros fabricantes que se estén utilizando. Encontrará más información sobre seguridad industrial en http://www.siemens.com/industrialsecurity. Si desea mantenerse al día de las actualizaciones de nuestros productos, regístrese para recibir un boletín de noticias específico del producto que desee. Encontrará más información en http://support.automation.siemens.com. (V3.4)</p>

Última modificación:

18/12/2020 

Homologaciones

Los productos SIMATIC NET descritos en estas Instrucciones de servicio disponen de las homologaciones indicadas a continuación.

Nota

Homologaciones otorgadas en la placa de características del dispositivo

Las homologaciones indicadas se considerarán concedidas cuando el producto lleve el distintivo correspondiente. Las homologaciones que han sido concedidas a su producto, se reconocen por los distintivos que figuran en la placa de características del mismo.

Certificados actuales en Internet

En las páginas de Internet del Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/es/ps/15273/cert>) encontrará las homologaciones actuales del producto.

Indicaciones para los fabricantes de máquinas

Los dispositivos no son máquinas en el sentido de la directiva de la CE sobre máquinas. Por esta razón no existe para estos dispositivos ninguna declaración de conformidad según la directiva de la CE sobre máquinas 2006/42/CE.

Si los dispositivos forman parte del equipamiento de una máquina, el fabricante de la máquina debe tenerlos en cuenta en el procedimiento de declaración de conformidad.

Declaración de conformidad de la Unión Europea



Los productos SIMATIC NET descritos en estas instrucciones de servicio cumplen los requisitos y los objetivos en materia de seguridad de las directivas de la UE siguientes y, además, cumple las normas armonizadas europeas (EN) que se mencionan en los documentos oficiales de la UE.

- **2014/34/UE (directiva de protección frente a explosiones ATEX)**
Directiva del Parlamento Europeo y del Consejo, de 26 de febrero de 2014, relativa a la aproximación de las legislaciones de los Estados miembros sobre los aparatos y sistemas de protección para uso en atmósferas potencialmente explosivas; Boletín Oficial de la UE L96, 29/03/2014, págs. 309-356
- **2014/30/UE (CEM)**
Directiva CEM del Parlamento Europeo y del Consejo, de 26 de febrero de 2014, relativa a la aproximación de las legislaciones de los Estados miembros en materia de compatibilidad electromagnética; Boletín Oficial de la UE L96, 29/03/2014, págs. 79-106
- **2011/65/UE (RoHS)**
Directiva del Parlamento Europeo y del Consejo, de 8 de junio de 2011, sobre restricciones a la utilización de determinadas sustancias peligrosas en aparatos eléctricos y electrónicos


Homologaciones

En las páginas de Internet del Siemens Industry Online Support (<http://support.automation.siemens.com/WW/view/es/33118389/134200>) encontrará la declaración de conformidad de la Unión Europea para estos productos.

La declaración de conformidad de la UE se encuentra a disposición de todas las autoridades competentes en:

Siemens Aktiengesellschaft
Division Process Industries and Drives
Process Automation
DE-76181 Karlsruhe
Alemania

ATEX (directiva de protección contra explosión)

 ADVERTENCIA
En caso de utilizar productos SIMATIC NET en áreas con peligro de explosión zona 2, tenga en cuenta las condiciones especiales relacionadas con ello que figuran en el documento "SIMATIC NET Product Information Use of subassemblies/modules in a Zone 2 Hazardous Area".
Encontrará este documento:
<ul style="list-style-type: none">• En el soporte de datos incluido en el volumen de suministro de algunos aparatos.• En las páginas de Internet del Siemens Industry Online Support (http://support.automation.siemens.com/WW/view/es).
Introduzca como término de búsqueda el número de identificación del documento "C234".

Los productos SIMATIC NET cumplen hasta el 19.04.2016 los requisitos de la Directiva CE 94/9/CE "Aparatos y sistemas de protección para uso en atmósferas potencialmente explosivas", y a partir del 20.04.2016 la Directiva de la Unión Europea 2014/34/UE.

Clasificación ATEX:

II 3 G Ex nA IIC T4 Gc

KEMA 07ATEX0145 X


Los productos cumplen los requisitos de las normas:

- EN 60079-15 (Electrical apparatus for potentially explosive atmospheres; Type of protection "n")
- EN 60079-0 (Atmósferas explosivas - Parte 0: Equipo. Requisitos generales)

Encontrará las versiones actuales de las normas en los certificados ATEX vigentes.

9.1 ATEX (KEMA 07 ATEX0145 X)[op is]

ATEX (directiva de protección contra explosión) - radiación óptica

<p> ADVERTENCIA</p> <p>En caso de utilizar productos SIMATIC NET en áreas con peligro de explosión zona 2, tenga en cuenta las condiciones especiales relacionadas con ello que figuran en el documento "SIMATIC NET Product Information Use of subassemblies/modules in a Zone 2 Hazardous Area".</p> <p>Encontrará este documento:</p> <ul style="list-style-type: none"> • En el soporte de datos incluido en el volumen de suministro de algunos aparatos. • En las páginas de Internet del Siemens Industry Online Support (http://support.automation.siemens.com/WW/view/es). <p>Introduzca como término de búsqueda el número de identificación del documento "C234".</p>

Los productos SIMATIC NET cumplen hasta el 19.04.2016 los requisitos de la Directiva CE 94/9/CE "Aparatos y sistemas de protección para uso en atmósferas potencialmente explosivas", y a partir del 20.04.2016 la Directiva de la Unión Europea 2014/34/UE.

Clasificación ATEX:

II 3 (2) G Ex nA [op is Gb] IIC T4 Gc

DEKRA 11ATEX0060 X

Los productos cumplen los requisitos de las normas:

- EN 60079-0
- EN 60079-15
- EN 60079-28

Encontrará las versiones actuales de las normas en los certificados ATEX vigentes.

IECEX

Los productos SIMATIC NET cumplen los requisitos de protección contra explosión según IECEX.

Clasificación IECEX:

Ex nA IIC T4 Gc

DEK 14.0025X

Los productos cumplen los requisitos de las normas:

- IEC 60079-15 (Atmósferas explosivas - Parte 15: Protección del equipo por modo de protección "n")
- IEC 60079-0 (Atmósferas explosivas - Parte 0: Equipo. Requisitos generales)

Las versiones actuales de las normas se encuentran en los certificados IECEX vigentes.

*Homologaciones**9.2 IECEx (Ex nA [op is Gb] IIC T4 Gc)***9.2 IECEx (Ex nA [op is Gb] IIC T4 Gc)****IECEx (radiación óptica)**

Los productos SIMATIC NET cumplen los requisitos de protección contra explosión según IECEx.

Clasificación IECEx:

Ex nA [op is Gb] IIC T4 Gc

DEK 14.0026X

Los productos cumplen los requisitos de las normas:

- IEC 60079-15 (Atmósferas explosivas - Parte 15: Protección del equipo por modo de protección "n")
- IEC 60079-0 (Atmósferas explosivas - Parte 0: Equipo. Requisitos generales)
- IEC 60079-28 (Atmósferas explosivas - Parte 28: Protección de material y sistemas de transmisión que utilizan radiación óptica)

Las versiones actuales de las normas se encuentran en los certificados IECEx vigentes.

Directiva CEM (compatibilidad electromagnética)

Los productos SIMATIC NET descritos en las presentes instrucciones de servicio cumplen a fecha de 19.04.2016 los requisitos de la Directiva 2004/108/CE "Compatibilidad electromagnética" (Directiva CEM), y desde el 20.04.2016 la Directiva 2014/30/UE.

Campo de aplicación	Requisitos en cuanto a	
	Emisión de perturbaciones	Inmunidad a las perturbaciones
Industria	EN 61000-6-4	EN 61000-6-2

Encontrará las versiones actuales de las normas en la declaración de conformidad CE actualmente vigente.

RoHS (restricciones a la utilización de determinadas sustancias peligrosas)

Los productos SIMATIC NET descritos en las presentes instrucciones de servicio cumplen los requisitos de la directiva de la UE 2011/65/UE sobre restricciones a la utilización de determinadas sustancias peligrosas en aparatos eléctricos y electrónicos.

Norma aplicada:

- EN 50581

FM

El producto cumple las exigencias de las normas:

- Factory Mutual Approval Standard Class Number 3611
- FM Hazardous (Classified) Location Electrical Equipment:
Non Incendive / Class I / Division 2 / Groups A,B,C,D / T4 y
Non Incendive / Class I / Zone 2 / Group IIC / T4

Homologación cULus Industrial Control Equipment

cULus Listed IND. CONT. EQ.

Underwriters Laboratories Inc. según

- UL 61010-2-201
- CAN/CSA-IEC 61010-2-201

Report número E85972

Homologación cULus Information Technology Equipment

cULus Listed I. T. E.

Underwriters Laboratories Inc. según

- UL 60950-1 (Information Technology Equipment)
- CSA C22.2 No. 60950-1-03

Report número E115352

Homologación cULus Hazardous Location

cULus Listed I. T. E. FOR HAZ. LOC.

Underwriters Laboratories Inc. según

- UL 60950-1 (Information Technology Equipment)
- ANSI/ISA 12.12.01-2007
- CSA C22.2 No. 213-M1987

Approved for use in

Cl. 1, Div. 2, GP A, B, C, D T4

Cl. 1, zona 2, GP IIC T4

Report número E240480

RCM

El producto cumple las exigencias de la norma AS/NZS 2064 (Class A).

*Homologaciones**9.2 IECEx (Ex nA [op is Gb] IIC T4 Gc)***MSIP 요구사항 - For Korea only****A 급 기기(업무용 방송통신기자재)**

이 기기는 업무용(A 급) 전자파 적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정 외의 지역에서 사용하는 것을 목적으로 합니다.

Certificación de la unión aduanera euroasiática

EAC (Eurasian Conformity)

Unión aduanera de Rusia, Bielorrusia y Kazajistán

Declaración de conformidad según las normas técnicas de la unión aduanera (TR CU)

Certificaciones FDA e IEC


Los siguientes dispositivos cumplen los requisitos de la FDA e IEC indicados más abajo:

Dispositivo	CLASS 1 LASER Product	CLASS 1 LED Product
SCALANCE XC206-2 (ST/BFOC)	-	•
SCALANCE XC206-2 (SC)	-	•
SCALANCE XC206-2SFP	-	-
SCALANCE XC208	-	-
SCALANCE XC216	-	-
SCALANCE XC224	-	-

En dispositivos modulares, la certificación tiene lugar mediante los correspondientes transceptores enchufables.



Figura 9-1 Certificaciones FDA e IEC

 PRECAUCIÓN
La utilización de otros controladores/reguladores/elementos de control, ajustes o la ejecución de procedimientos diferentes a los indicados aquí pueden provocar exposiciones a radiaciones peligrosas.

Cumplimiento de las directivas de instalación

Los dispositivos cumplen los requisitos si en su instalación y su uso se respetan las directivas de instalación y las instrucciones de seguridad que se describen en esta documentación y en otras sucesivas.

- Manual de sistema "Industrial Ethernet / PROFINET Industrial Ethernet"
- Manual de sistema "Industrial Ethernet / PROFINET - Componentes de red pasivos"
Encontrará más información sobre los manuales de sistema en el apartado "Documentación complementaria" del capítulo "Introducción (Página 5)".
- Manual de configuración "Directrices de montaje CEM"
60612658 (<http://support.automation.siemens.com/WW/view/es/60612658>)

**ADVERTENCIA****Pueden producirse daños personales y materiales**

Por la instalación de ampliaciones no autorizadas para los productos SIMATIC NET o sus sistemas de destino se pueden dejar de cumplir las exigencias y prescripciones de seguridad y compatibilidad electromagnética.

Utilice únicamente ampliaciones que estén autorizadas para el sistema.


Nota


El test se efectuó con el dispositivo y una estación de comunicación conectada, que también cumplía las normas mencionadas anteriormente.

En caso de utilizar el dispositivo con una estación de comunicación que no cumpla dichas normas, no se puede garantizar el respeto de los valores correspondientes.

2.3 Certificación Transceptor SFP992-1

		<h2 style="margin: 0;">IECEX Certificate of Conformity</h2>	
<p>INTERNATIONAL ELECTROTECHNICAL COMMISSION IEC Certification Scheme for Explosive Atmospheres <small>for rules and details of the IECEX Scheme visit www.iecex.com</small> Ex COMPONENT CERTIFICATE</p>			
Certificate No.:	IECEX DEK 18.0050U	Issue No: 0	<u>Certificate history:</u> Issue No. 0 (2019-03-06)
Status:	Current	Page 1 of 3	
Date of Issue:	2019-03-06		
Applicant:	SIEMENS AG Östliche Rheinbrückenstraße 50 76187 Karlsruhe Germany		
Ex Component:	Optical Transceivers type 6GK599. ...		
<p>This component is NOT intended to be used alone and requires additional consideration when incorporated into other equipment or systems for use in explosive atmospheres (refer to IEC 60079-0).</p>			
Type of Protection:	Ex nA		
Marking:	Ex nA IIC Gc		
Approved for issue on behalf of the IECEX Certification Body:	R. Schuller		
Position:	Certification Manager		
Signature: (for printed version)			
Date:	2019-03-06		
<p>1. This certificate and schedule may only be reproduced in full. 2. This certificate is not transferable and remains the property of the issuing body. 3. The Status and authenticity of this certificate may be verified by visiting the Official IECEX Website.</p>			
Certificate issued by:	DEKRA Certification B.V. Meander 1051, 6825 MJ Arnhem The Netherlands		
			

	IECEX Certificate of Conformity	
Certificate No:	IECEX DEK 18.0050U	Issue No: 0
Date of Issue:	2019-03-06	Page 2 of 3
Manufacturer:	Siemens AG Östliche Rheinbrückenstraße 50 76187 Karlsruhe Germany	
Additional Manufacturing location(s):		
<p>This certificate is issued as verification that a sample(s), representative of production, was assessed and tested and found to comply with the IEC Standard list below and that the manufacturer's quality system, relating to the Ex Component covered by this certificate, was assessed and found to comply with the IECEx Quality system requirements. This certificate is granted subject to the conditions as set out in IECEx Scheme Rules, IECEx 02 and Operational Documents as amended.</p>		
STANDARDS:		
The Ex Component and any acceptable variations to it specified in the schedule of this certificate and the identified documents, was found to comply with the following standards:		
IEC 60079-0 : 2011 Edition:6.0	Explosive atmospheres - Part 0: General requirements	
IEC 60079-15 : 2010 Edition:4	Explosive atmospheres - Part 15: Equipment protection by type of protection "n"	
<p><i>This Certificate does not indicate compliance with electrical safety and performance requirements other than those expressly included in the Standards listed above.</i></p>		
TEST & ASSESSMENT REPORTS:		
<i>A sample(s) of the Ex Component listed has successfully met the examination and test requirements as recorded in</i>		
<u>Test Report:</u>		
NL/DEK/ExTR18.0051/00		
<u>Quality Assessment Report:</u>		
NL/DEK/QAR12.0079/04		

	IECEX Certificate of Conformity	
Certificate No:	IECEX DEK 18.0050U	Issue No: 0
Date of Issue:	2019-03-06	Page 3 of 3
Schedule		
Ex Component(s) covered by this certificate is described below:		
Optical Transceivers type 6GK599. ... for use in SIMATIC Network Components series.		
Fiber Outputs: Laser Class 1 approved according IEC 60825-1. The optical cables may lead either into or through hazardous areas requiring equipment of EPL Gb, Gc, Db or Dc.		
The type code, the service temperature range and the temperature class of the modules shall be taken from table 1, See Annex.		
Electrical data		
The electrical data of the supply and the input and output circuits shall be taken from Table 1, see Annex.		
SCHEDULE OF LIMITATIONS:		
The Optical Transceivers Type 6GK599. ... may only be used with SIMATIC Network Components series as covered in IECEx DEK 14.0025X.		
Service temperature: Suitable for T4 classification when installed in SIMATIC Network Components series as covered in IECEx DEK 14.0025X for an ambient temperature up to 70 °C.		
Annex:		
223043100-ExTR18.0051.00-Annex 1.pdf		



**Annex 1 to Certificate of Conformity IECEx DEK 18.0050U/
Testreport NL/DEK/ExTR18.0051.00**

Table 1

Description	Type/MLFB No.	Vendor ID	Service temperature	Technical Data
SFP991-1	6GK5991-1AD00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP991-1	6GK5991-1AD00-8AA0	0X02	-40 ... +85°C	3.3V / max.110 mA
SFP991-1 Coated	6GK5991-1AD00-8FA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP991-1 Coated	6GK5991-1AD00-8FA0	0X02	-40 ... +85°C	3.3V / max.110 mA
SFP991-1LD	6GK5991-1AF00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP991-1LD	6GK5991-1AF00-8AA0	0X01	-40 ... +85°C	3.3V / max.300 mA
SFP991-1LD Coated	6GK5991-1AF00-8FA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP991-1LH+	6GK5991-1AE00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP991-1LH+	6GK5991-1AE00-8AA0	0X01	-40 ... +85°C	3.3V / max.300 mA
SFP991-1ELH200	6GK5991-1AE30-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1	6GK5992-1AL00-8AA0	0X00	-40 ... +85°C	3.3V / max.240 mA
SFP992-1	6GK5992-1AL00-8AA0	0X01	-40 ... +85°C	3.3V / max.144 mA
SFP992-1+	6GK5992-1AG00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1+	6GK5992-1AG00-8AA0	0X04	-40 ... +85°C	3.3V / max.250 mA
SFP992-1LD	6GK5992-1AM00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1LD	6GK5992-1AM00-8AA0	0X01	-40 ... +85°C	3.3V / max.300 mA
SFP992-1LD Coated	6GK5992-1AM00-8FA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1LD+	6GK5992-1AM30-8AA0	0X04	-40 ... +85°C	3.3V / max.300 mA
SFP992-1LH	6GK5992-1AN00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1LH+	6GK5992-1AP00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1LH+	6GK5992-1AP00-8AA0	0X01	-40 ... +85°C	3.3V / max.300 mA
SFP992-1ELH	6GK5992-1AQ00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1BXMT	6GK5992-1AL00-8TA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1BXMT	6GK5992-1AL00-8TA0	0X04	-40 ... +85°C	3.3V / max.250 mA
SFP992-1BXM	6GK5992-1AL00-8RA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP992-1BXM	6GK5992-1AL00-8RA0	0X04	-40 ... +85°C	3.3V / max.250 mA
SFP992-1BX10T	6GK5992-1AM00-8TA0	0X04	-40 ... +85°C	3.3V / max.300 mA
SFP992-1BX10R	6GK5992-1AM00-8RA0	0X04	-40 ... +85°C	3.3V / max.300 mA
SFP993-1	6GK5993-1AT00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP993-1	6GK5993-1AT00-8AA0	0X01	-40 ... +85°C	3.3V / max.289 mA
SFP993-1LD	6GK5993-1AU00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SFP993-1LD	6GK5993-1AU00-8AA0	0X01	-40 ... +85°C	3.3V / max.285 mA
SFP993-1LH	6GK5993-1AV00-8AA0	0X00	-40 ... +85°C	3.3V / max.450 mA
SFP993-1LH	6GK5993-1AV00-8AA0	0X01	-40 ... +85°C	3.3V / max.454 mA
STP991-1	6GK5991-1AB00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
STP991-1LD	6GK5991-1AC00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA
SCP992-1	6GK5992-1AJ00-8AA0	0X00	-40 ... +85°C	3.3V / max.240 mA
SCP992-1LD	6GK5992-1AK00-8AA0	0X00	-40 ... +85°C	3.3V / max.300 mA

2.4 CPU 317-2 PN/DP

SIEMENS

Hoja de datos

6ES7317-2EK14-0AB0



SIMATIC S7-300 CPU 317-2 PN/DP, Módulo central con memoria de trabajo de 1 MB, 1.ª interfaz MPI/DP 12 Mbits/s, 2.ª interfaz Ethernet PROFINET, con switch de 2 puertos, Se necesita Micro Memory Card

Información general	
Función del producto	
• Modo isócrono	Sí; a través de la interfaz PROFIBUS DP o PROFINET
Ingeniería con	
• Paquete de programación	STEP 7 V5.5 o superior
Tensión de alimentación	
Valor nominal (DC)	24 V
Rango admisible, límite inferior (DC)	20,4 V
Rango admisible, límite superior (DC)	28,8 V
Protección externa para líneas de alimentación (recomendación)	min. 2 A
Puenteo de caídas de red y tensión	
• Puenteo de caídas de red/de tensión	5 ms
• Tasa de repetición, mín.	1 s
Intensidad de entrada	
Consumo (valor nominal)	750 mA
Consumo (en marcha en vacío), tip.	150 mA
Intensidad de cierre, tip.	4 A
I_t	1 A ² ·s
Pérdidas	
Pérdidas, tip.	4,65 W
Memoria	
Memoria de trabajo	
• integrada	1 024 kbyte
• ampliable	No
Memoria de carga	
• enchufable (MMC)	Sí
• enchufable (MMC), máx.	8 Mbyte
• Conservación de datos en MMC (tras última programación), mín.	10 y
Respaldo	
• existente	Sí; garantizado por la MMC (sin mantenimiento)
• sin pila	Sí; Programa y datos
Tiempos de ejecución de la CPU	
para operaciones de bits, tip.	0,025 µs
para operaciones a palabras, tip.	0,03 µs
para aritmética de coma fija, tip.	0,04 µs
para aritmética de coma flotante, tip.	0,16 µs

CPU-bloques	
Nº de bloques (total)	2 048; (DB, FC, FB); la cantidad máxima de bloques cargables puede verse reducida por la MMC utilizada por el usuario.
DB	
• Número, máx.	2 048; Banda numérica: 1 a 16000
• Tamaño, máx.	64 kbyte
FB	
• Número, máx.	2 048; Banda numérica: 0 a 7999
• Tamaño, máx.	64 kbyte
FC	
• Número, máx.	2 048; Banda numérica: 0 a 7999
• Tamaño, máx.	64 kbyte
OB	
• Tamaño, máx.	64 kbyte
• Nº de OBs de ciclo libre	1; OB 1
• Nº de OBs de alarma horaria	1; OB 10
• Nº de OBs de alarma de retardo	2; OB 20, 21
• Nº de OBs de alarma cíclica	4; OB 32, 33, 34, 35
• Nº de OBs de alarma de proceso	1; OB 40
• Nº de OBs de alarmas DPV1	3; OB 55, 56, 57
• Nº de OBs de modo isócrono	1; OB 61 - Modo isócrono sobre DP o sobre PROFINET IO posible (no simultáneamente)
• Nº de OBs de arranque	1; OB 100
• Nº de OBs de errores asíncronos	6; OB 80, 82, 83, 85, 86, 87 (OB 83 solo para PROFINET IO)
• Nº de OBs de errores síncronos	2; OB 121, 122
Profundidad de anidamiento	
• por cada prioridad	16
• adicional, dentro de un OB de error	4
Contadores, temporizadores y su remanencia	
Contadores S7	
• Cantidad	512
Remanencia	
— Configurable	Si
— Límite inferior	0
— Límite superior	511
— predeterminado	Z 0 a Z 7
Rango de contaje	
— Configurable	Si
— Límite inferior	0
— Límite superior	999
Contadores IEC	
• existente	Si
• Clase	SFB
• Cantidad	ilimitado (limitado sólo por la memoria de trabajo)
Temporizadores S7	
• Cantidad	512
Remanencia	
— Configurable	Si
— Límite inferior	0
— Límite superior	511
— predeterminado	sin remanencia
Rango de tiempo	
— Límite inferior	10 ms
— Límite superior	9 990 s
Temporizadores IEC	
• existente	Si
• Clase	SFB
• Cantidad	ilimitado (limitado sólo por la memoria de trabajo)

Áreas de datos y su remanencia	
Área de datos remanentes (incl. temporizadores, contadores, marcas), máx.	256 kbyte
Marcas	
• Tamaño, máx.	4 096 byte
• Remanencia disponible	Sí; de MB 0 a MB 4 095
• Remanencia predeterminada	MB 0 a MB 15
• Nº de marcas de ciclo	8; 1 byte de marcas
Bloques de datos	
• Remanencia configurable	Sí; ajustando apropiadamente la propiedad de volatilidad del DB
• Remanencia predeterminada	Sí
Datos locales	
• por cada prioridad, máx.	32 768 byte; máx. 2048 bytes por bloque
Área de direcciones	
Área de direcciones de periferia	
• Entradas	8 192 byte
• Salidas	8 192 byte
de ellas, descentralizadas	
— Entradas	8 192 byte
— Salidas	8 192 byte
Imagen del proceso	
• Entradas	8 192 byte
• Salidas	8 192 byte
• Entradas, configurables	8 192 byte
• Salidas, configurables	8 192 byte
• Entradas, predeterminado	256 byte
• Salidas, predeterminado	256 byte
Imágenes de subproceso	
• Nº de imágenes de subproceso, máx.	1; en PROFINET IO la longitud de los datos útiles está limitada a 1600 bytes
Canales digitales	
• Entradas	65 536
— de las cuales centralizadas	1 024
• Salidas	65 536
— de las cuales centralizadas	1 024
Canales analógicos	
• Entradas	4 096
— de las cuales centralizadas	256
• Salidas	4 096
— de las cuales centralizadas	256
Configuración del hardware	
Número de aparatos de ampliación, máx.	3
Nº de maestros DP	
• integrada	1
• vía CP	4
Nº de FM y CP utilizables (recomendación)	
• FM	8
• CP PaP	8
• CP, LAN	10
Bastidores	
• Bastidores, máx.	4
• Módulos por bastidor, máx.	8
Hora	
Reloj	
• Reloj de hardware (en tiempo real)	Sí
• respaldado y sincronizable	Sí
• Duración del respaldo	6 wk; a 40 °C de temperatura ambiente
• Desviación diaria, máx.	10 s; típ.: 2 s

<ul style="list-style-type: none"> Comportamiento del reloj tras RED CON Comportamiento del reloj tras agotamiento de batería 	<p>El reloj continúa funcionando tras el corte de alimentación</p> <p>El reloj continúa corriendo con la hora a la que se produjo la RED DES</p>
Contador de horas de funcionamiento	
<ul style="list-style-type: none"> Cantidad Número/banda numérica Rango de valores Granularidad remanente 	<p>4</p> <p>0 a 3</p> <p>0 a 2³¹ horas (si se usa el SFC 101)</p> <p>1 h</p> <p>Sí; tiene que reiniciarse en cada re arranque</p>
Sincronización de la hora	
<ul style="list-style-type: none"> Soporta en MPI, maestro en MPI, esclavo en DP, maestro en DP, esclavo en el autómata, maestro en el autómata, esclavo por Ethernet via NTP 	<p>Sí</p> <p>Sí</p> <p>Sí</p> <p>Sí; para esclavo DP, solo hora de esclavo</p> <p>Sí</p> <p>Sí</p> <p>Sí</p> <p>Sí; Como cliente</p>
Entradas digitales	
Nº de entradas digitales	0
Salidas digitales	
Número de salidas	0
Entradas analógicas	
Nº de entradas analógicas	0
Salidas analógicas	
Nº de salidas analógicas	0
Interfaces	
Nº de interfaces Industrial Ethernet	1; 2 puertos (switch) RJ45
Nº de interfaces PROFINET	1; 2 puertos (switch) RJ45
Nº de interfaces RS 485	1; MPI/PROFIBUS DP combinado
Nº de interfaces RS 422	0
1. Interfaz	
Tipo de interfaz	Interfaz RS485 integrada
con aislamiento galvánico	Sí
Física de la interfaz	
<ul style="list-style-type: none"> RS 485 Intensidad de salida de la interfaz, máx. 	<p>Sí</p> <p>200 mA</p>
Protocolos	
<ul style="list-style-type: none"> MPI Maestro PROFIBUS DP Esclavo PROFIBUS DP Acoplamiento punto a punto 	<p>Sí</p> <p>Sí</p> <p>Sí</p> <p>No</p>
MPI	
<ul style="list-style-type: none"> Velocidad de transferencia, máx. 	12 Mbit/s
Servicios	
<ul style="list-style-type: none"> Comunicación PG/OP Enrutado Comunicación de datos globales Comunicación S7 básica Comunicación S7 Comunicación S7, como cliente Comunicación S7, como servidor 	<p>Sí</p> <p>Sí</p> <p>Sí</p> <p>Sí</p> <p>Sí</p> <p>No; pero a través de CP y FB cargables</p> <p>Sí</p>
Maestro PROFIBUS DP	
<ul style="list-style-type: none"> Velocidad de transferencia, máx. Nº de esclavos DP, máx. 	<p>12 Mbit/s</p> <p>124</p>
Servicios	
<ul style="list-style-type: none"> Comunicación PG/OP Enrutado 	<p>Sí</p> <p>Sí</p>

— Comunicación de datos globales	No
— Comunicación S7 básica	Sí; sólo bloques 1
— Comunicación S7	Sí
— Comunicación S7, como cliente	No
— Comunicación S7, como servidor	Sí
— Equidistancia	Sí
— Modo isócrono	Sí; OB 61 Modo isócrono solo posible en PROFIBUS DP o PROFINET IO alternativamente
— SYNC/FREEZE	Sí
— Activar/desactivar esclavos DP	Sí
— Nº de esclavos DP activables/desactivables simultáneamente, máx.	8
— Comunicación directa de datos (esclavo-esclavo)	Sí; como suscriptor
— DPV1	Sí
Área de direcciones	
— Entradas, máx.	8 kbyte
— Salidas, máx.	8 kbyte
Datos útiles por esclavo DP	
— Entradas, máx.	244 byte
— Salidas, máx.	244 byte
Esclavo PROFIBUS DP	
• Velocidad de transferencia, máx.	12 Mbit/s
• Búsqueda automática de velocidad de transferencia	Sí; sólo con interfaz pasiva
• Área de direcciones, máx.	32
• Datos útiles por área de direcciones, máx.	32 byte
Servicios	
— Comunicación PG/OP	Sí
— Enrutado	Sí; sólo con interfaz activa
— Comunicación de datos globales	No
— Comunicación S7 básica	No
— Comunicación S7	Sí
— Comunicación S7, como cliente	No
— Comunicación S7, como servidor	Sí; Sólo conexión de configuración unidireccional
— Comunicación directa de datos (esclavo-esclavo)	Sí
— DPV1	No
Memoria de transferencia	
— Entradas	244 byte
— Salidas	244 byte
2. Interfaz	
con aislamiento galvánico	Sí
Detección automática de la velocidad de transferencia	Sí; 10/100 Mbits/s
Autonegociación	Sí
Autocrossing	Sí
Cambio de dirección IP en tiempo de ejecución, función soportada	Sí
Física de la interfaz	
• RJ 45 (Ethernet)	Sí
• Número de puertos	2
• Switch integrado	Sí
Protocolos	
• MPI	No
• PROFINET IO-Controller	Sí; también con funcionalidad de IO-Device simultánea
• PROFINET IO-Device	Sí; también con funcionalidad de IO-Controller simultánea
• PROFINET CBA	Sí
• Maestro PROFIBUS DP	No
• Esclavo PROFIBUS DP	No
• Comunicación IE abierta	Sí; mediante TCP/IP, ISO on TCP, UDP

• Servidores web	Sí
• Redundancia del medio	Sí
PROFINET IO-Controller	
• Velocidad de transferencia, máx.	100 Mbit/s
Servicios	
— Comunicación PG/OP	Sí
— Enrutado	Sí
— Comunicación S7	Sí; con FB cargables, conexiones configurables máx.: 16, máx. número de instancias: 32
— Modo isócrono	Sí; OB 61 Modo isócrono solo posible en PROFIBUS DP o PROFINET IO alternativamente
— IRT	Sí
— Shared Device	Sí
— Arranque priorizado	Sí
— Número de dispositivos IO con arranque preferente, máx.	32
— Nº de IO Devices que se pueden conectar en total, máx.	128
— de los cuales, IO devices con IRT, máx.	64
— de ellos, en línea, máx.	64
— Nº de IO Devices con IRT y la opción "alta flexibilidad"	128
— de ellos, en línea, máx.	61
— Nº de IO-Devices conectables para RT, máx.	128
— de ellos, en línea, máx.	128
— Activar/desactivar IO Devices	Sí
— Nº de IO-Devices activables/desactivables simultáneamente, máx.	8
— IO-Devices (puertos asociados) que cambian en servicio, soportado	Sí
— Nº de IO-Devices por herramienta, máx.	8
— Cambio de aparato sin soporte removible	Sí
— Tiempos de ciclo de envío	250 µs, 500 µs, 1 ms, 2 ms, 4 ms (no con IRT y opción "Alta flexibilidad")
— Tiempo de actualización	250 µs a 512 ms (dependiendo del modo de servicio; más detalles en el manual de producto "S7-300 CPU 31xC y CPU 31x, Datos técnicos")
Área de direcciones	
— Entradas, máx.	8 kbyte
— Salidas, máx.	8 kbyte
— Coherencia de datos útiles, máx.	1 024 byte
PROFINET IO-Device	
Servicios	
— Comunicación PG/OP	Sí
— Enrutado	Sí
— Comunicación S7	Sí; con FB cargables, conexiones configurables máx.: 16, máx. número de instancias: 32
— Modo isócrono	No
— IRT	Sí
— PROFINergy	Sí; Con SFB 73 / 74 preparado para FB estándar PROFINergy para I-Device
— Shared Device	Sí
— Nº de IO Controller con Shared Device, máx.	2
Memoria de transferencia	
— Entradas, máx.	1 440 byte; por cada IO Controller con Shared Device
— Salidas, máx.	1 440 byte; por cada IO Controller con Shared Device
Submódulos	
— Número, máx.	64
— Datos útiles por submódulo, máx.	1 024 byte
PROFINET CBA	
• Transferencia acíclica	Sí
• Transferencia cíclica	Sí
Comunicación IE abierta	

<ul style="list-style-type: none"> • Número de conexiones máx. • Números de puerto locales utilizados en el sistema • Función Keep-Alive, soportada 	<p>16</p> <p>0, 20, 21, 23, 25, 80, 102, 135, 161, 443, 8080, 34962, 34963, 34964, 65532, 65533, 65534, 65535</p> <p>Sí</p>
Protocolos	
Funcionamiento redundante:	
Redundancia del medio	
<ul style="list-style-type: none"> — Tiempo de conmutación en caso de rotura de cable, tip. — Nº de estaciones en el anillo, máx. 	<p>200 ms; PROFINET MRP</p> <p>50</p>
Comunicación IE abierta:	
<ul style="list-style-type: none"> • TCP/IP <ul style="list-style-type: none"> — Número de conexiones máx. — Tamaño de datos con tipo de conexión 01H, máx. — Tamaño de datos con tipo de conexión 11H, máx. — varias conexiones pasivas por puerto, función soportada • ISO-on-TCP (RFC1006) <ul style="list-style-type: none"> — Número de conexiones máx. — Tamaño de datos, máx. • UDP <ul style="list-style-type: none"> — Número de conexiones máx. — Tamaño de datos, máx. 	<p>Sí; a través de interfaz PROFINET y FB cargables</p> <p>16</p> <p>1 460 byte</p> <p>32 768 byte</p> <p>Sí</p> <p>Sí; a través de interfaz PROFINET y FB cargables</p> <p>16</p> <p>32 768 byte</p> <p>Sí; a través de interfaz PROFINET y FB cargables</p> <p>16</p> <p>1 472 byte</p>
Servidores web	
<ul style="list-style-type: none"> • Soporta • Páginas web definidas por el usuario • N.º de clientes HTTP 	<p>Sí</p> <p>Sí</p> <p>5</p>
Funciones de comunicación	
Comunicación PG/OP	Sí
Enrutado de registros	Sí
Comunicación de datos globales	
<ul style="list-style-type: none"> • Soporta • Nº de círculos GD, máx. • Nº de paquetes GD, máx. • Nº de paquetes GD, emisor, máx. • Nº de paquetes GD, receptor, máx. • Tamaño de paquetes GD, máx. • Tamaño de paquetes GD (de ellos, coherentes), máx. 	<p>Sí</p> <p>8</p> <p>8</p> <p>8</p> <p>8</p> <p>22 byte</p> <p>22 byte</p>
Comunicación S7 básica:	
<ul style="list-style-type: none"> • Soporta • Datos útiles por petición, máx. • Datos útiles por petición (de ellos, coherentes), máx. 	<p>Sí</p> <p>76 byte</p> <p>76 byte; 76 bytes (con X_SEND o X_RCV), 64 bytes (con X_PUT o X_GET como servidor)</p>
Comunicación S7:	
<ul style="list-style-type: none"> • Soporta • como servidor • Como cliente • Datos útiles por petición, máx. 	<p>Sí</p> <p>Sí</p> <p>Sí; a través de la interfaz PROFINET integrada y FB cargables o a través de CP y FB cargables</p> <p>ver ayuda en línea de STEP 7 ("Parámetros comunes de los SFB/FB y las SFC/FC de la Comunicación S7")</p>
Comunicación compatible con S5:	
<ul style="list-style-type: none"> • Soporta 	Sí; a través de CP y FC cargables
PROFINET CBA (con carga de comunicación ajustada a su valor teórico)	
<ul style="list-style-type: none"> • Ajuste teórico de la carga de comunicación de la CPU • Nº de interlocutores de interconexión remotos • Nº de funciones maestro/esclavo 	<p>50 %</p> <p>32</p> <p>30</p>

• Suma de todas las conexiones maestro/esclavo	1 000
• Tamaño de los datos de todas las conexiones maestro/esclavo entrantes, máx.	4 000 byte
• Tamaño de los datos de todas las conexiones maestro/esclavo salientes, máx.	4 000 byte
• Número de interconexiones internas del dispositivo y por PROFIBUS	500
• Tamaño de los datos de las interconexiones PROFIBUS y las interconexiones internas de los dispositivos, máx.	4 000 byte
• Tamaño de los datos de cada conexión, máx.	1 400 byte
Interconexiones remotas con transferencia acíclica	
— Intervalo de muestreo, mín.	500 ms
— Número de interconexiones entrantes	100
— Número de interconexiones salientes	100
— Tamaño de los datos de todas las interconexiones entrantes, máx.	2 000 byte
— Tamaño de los datos de todas las interconexiones salientes, máx.	2 000 byte
— Tamaño de los datos de cada conexión, máx.	1 400 byte
Interconexiones remotas con transferencia cíclica	
— Frecuencia de transferencia: intervalo de transferencia, mín.	10 ms
— Número de interconexiones entrantes	200
— Número de interconexiones salientes	200
— Tamaño de los datos de todas las interconexiones entrantes, máx.	2 000 byte
— Tamaño de los datos de todas las interconexiones salientes, máx.	2 000 byte
— Tamaño de los datos de cada conexión, máx.	450 byte
Variables HMI vía PROFINET (acíclicas)	
— Número de estaciones conectables para variables HMI (PN OPC/iMAP)	3; 2x PN OPC/1x iMap
— Actualización de variables HMI	500 ms
— Número de variables HMI	200
— Tamaño de datos de todas las variables HMI, máx.	2 000 byte
Funcionalidad de proxy PROFIBUS	
— Soporta	Sí
— Número de dispositivos PROFIBUS acoplados	16
— Tamaño de los datos de cada conexión, máx.	240 byte; en función del esclavo
Nº de conexiones	
• total	32
• usable para comunicación PG	31
— reservadas para comunicación PG	1
— configurables para comunicación PG, mín.	1
— configurables para comunicación PG, máx.	31
• usable para comunicación OP	31
— reservadas para comunicación OP	1
— configurables para comunicación OP, mín.	1
— configurables para comunicación OP, máx.	31
• usable para comunicación básica S7	30
— reservadas para comunicación básica S7	0
— configurables para comunicación básica S7, mín.	0
— configurables para comunicación básica S7, máx.	30
• usables para la comunicación S7	16
— reservadas para comunicación S7	0
— configurables para comunicación S7, mín.	0
— configurables para comunicación S7, máx.	16
• Nº total de instancias, máx.	32

• usable para enrutado	X1 como MPI: máx. 10; X1 como maestro DP: máx. 24; X1 como esclavo DP (activo): máx. 14; X2 como PROFINET: máx. 24
Funciones de aviso S7	
Número de estaciones conectables para funciones de aviso, máx.	32; depende de las conexiones configuradas para la comunicación PG/OP y S7 básica
Avisos de diagnóstico de proceso	Sí
Bloques Alarm-S activos simultáneamente, máx.	300
Funciones de test y puesta en marcha	
Estado de bloques	Sí; hasta 2 simultáneas
Paso individual	Sí
Nº de puntos de parada	4
Estado/forzado	
• Estado/forzado de variables	Sí
• Variables	Entradas, salidas, marcas, DB, tiempos, contadores
• Nº de variables, máx.	30
— de ellas, estado de variables, máx.	30
— de ellas, forzado de variables, máx.	14
Forzado permanente	
• Forzado permanente	Sí
• Forzado permanente, variables	Entradas, salidas
• Nº de variables, máx.	10
Búfer de diagnóstico	
• existente	Sí
• Nº de entradas, máx.	500
— Configurable	No
— de ellos seguros contra caída de red	100; Sólo son remanentes las 100 últimas entradas
• N.º de entradas legibles en RUN, máx.	499
— Configurable	Sí; de 10 a 499
— predeterminado	10
Datos de servicio técnico	
• Legibles	Sí
Condiciones ambientales	
Temperatura ambiente en servicio	
• mín.	0 °C
• máx.	60 °C
Configuración	
Software de configuración	
• STEP 7	Sí; V 5.5 o superior
programación	
• Juego de operaciones	Ver Lista de operaciones
• Niveles de paréntesis	8
• Funciones de sistema (SFC)	Ver Lista de operaciones
• Bloques de función de sistema (SFB)	Ver Lista de operaciones
Lenguaje de programación	
— KOP	Sí
— FUP	Sí
— AWL	Sí
— SCL	Sí
— CFC	Sí
— GRAPH	Sí
— HiGraph®	Sí
Protección de know-how	
• Protección de programas de usuario/Protección por contraseña	Sí
• Codificación de bloque	Sí; con bloque S7 Privacy
Dimensiones	
Ancho	40 mm
Altura	125 mm
Profundidad	130 mm
Pesos	
Peso, aprox.	340 g

2.5 CP 343-1

SIEMENS

Hoja de datos

6GK7343-1EX30-0XE0



procesador de comunicaciones CP 343-1 para conectar SIMATIC S7-300 a Industrial Ethernet via ISO y TCP/IP, controlador PROFINET IO o dispositivo PROFINET IO, switch de 2 puertos integrado ERTEC 200, comunicación S7, fetch/write, SEND/RECEIVE RFC1006, Multicast, DHCP, NTC- CPU Sync, con y sin diagnóstico, inicialización via LAN, 2 conexiones RJ45 para LAN con 10/100 Mbits/s

tasa de transferencia	
tasa de transferencia	
<ul style="list-style-type: none"> en la interfaz 1 	10 ... 100 Mbit/s
interfaces	
número de interfaces / según Industrial Ethernet	2
número de conexiones eléctricas	
<ul style="list-style-type: none"> en la interfaz 1 / según Industrial Ethernet 	2
<ul style="list-style-type: none"> para alimentación 	1
tipo de conexión eléctrica	
<ul style="list-style-type: none"> de la interfaz Industrial Ethernet 	Puerto RJ45
<ul style="list-style-type: none"> en la interfaz 1 / según Industrial Ethernet 	Puerto RJ45
tipo de conexión eléctrica	
<ul style="list-style-type: none"> para alimentación 	Regleta de bornes enchufable de 2 polos
tensión de alimentación, consumo de corriente, pérdidas	
tipo de corriente / de la tensión de alimentación	DC
tensión de alimentación / 1 / del bus de fondo	5 V
tensión de alimentación	24 V
tensión de alimentación / externa	24 V
tensión de alimentación / externa / con DC / valor nominal	24 V
tolerancia positiva relativa / con DC / con 24 V	20 %
tolerancia negativa relativa / con DC / con 24 V	15 %
corriente consumida	
<ul style="list-style-type: none"> del bus de fondo / con DC / con 5 V / típico 	0,2 A
<ul style="list-style-type: none"> de la tensión de alimentación externa / con DC / con 24 V / típico 	0,16 A
<ul style="list-style-type: none"> de la tensión de alimentación externa / con DC / con 24 V / máx. 	0,2 A
pérdidas [W]	5,8 W
condiciones ambientales	
temperatura ambiente	
<ul style="list-style-type: none"> con instalación vertical / durante el funcionamiento 	0 ... 40 °C
<ul style="list-style-type: none"> con posición de montaje vertical / durante el funcionamiento 	0 ... 60 °C
<ul style="list-style-type: none"> durante el almacenamiento 	-40 ... +70 °C
<ul style="list-style-type: none"> durante el transporte 	-40 ... +70 °C
humedad relativa del aire	
<ul style="list-style-type: none"> con 25 °C / sin condensación / durante el funcionamiento / máx. 	95 %

grado de protección IP	IP20
diseño, dimensiones y pesos	
formato de módulos	Módulo compacto S7-300 de anchura simple
anchura	40 mm
altura	125 mm
profundidad	120 mm
peso neto	0,22 kg
tipo de fijación	
<ul style="list-style-type: none"> montaje en perfil soporte S7-300 	Sí
datos de potencia / comunicación abierta	
número de conexiones posibles / para comunicación abierta / mediante bloques SEND/RECEIVE / máx.	16
volumen de datos	
<ul style="list-style-type: none"> como datos útiles por conexión ISO / para comunicación abierta / mediante bloques SEND/RECEIVE / máx. 	8 Kíbyte
<ul style="list-style-type: none"> como datos útiles por conexión ISO on TCP / para comunicación abierta / mediante bloques SEND/RECEIVE / máx. 	8 Kíbyte
<ul style="list-style-type: none"> como datos útiles por conexión TCP / para comunicación abierta / mediante bloques SEND/RECEIVE / máx. 	8 Kíbyte
<ul style="list-style-type: none"> como datos útiles por conexión UDP / para comunicación IE abierta / mediante bloques SEND/RECEIVE / máx. 	2 Kíbyte
número de estaciones Multicast	16
datos de potencia / comunicación S7	
número de conexiones posibles / para comunicación S7	
<ul style="list-style-type: none"> máx. 	16
datos de potencia / modo multiprotocolo	
número de conexiones activas / con modo multiprotocolo	32
datos de potencia / comunicación PROFINET / como PN IO-Controller	
número de PN IO-Devices / en PROFINET IO-Controller / utilizable / total	32
número de tramos PN IO externos / con PROFINET / por bastidor	1
volumen de datos	
<ul style="list-style-type: none"> como datos útiles para variables de entrada / como PROFINET IO-Controller / máx. 	1 Kíbyte
<ul style="list-style-type: none"> como datos útiles para variables de salida / como PROFINET IO-Controller / máx. 	1 Kíbyte
<ul style="list-style-type: none"> como datos útiles para variables de entrada por PN IO-Device / como PROFINET IO-Controller / máx. 	1433 byte
<ul style="list-style-type: none"> como datos útiles para variables de salida por PN IO-Device / como PROFINET IO-Controller / máx. 	1433 byte
<ul style="list-style-type: none"> como datos útiles para variables de entrada por PN IO-Device / por submódulo como PROFINET IO-Controller / máx. 	240 byte
<ul style="list-style-type: none"> como datos útiles para variables de salida por PN IO-Device / por submódulo como PROFINET IO-Controller / máx. 	240 byte
datos de potencia / comunicación PROFINET / como PN IO-Device	
función del producto / PROFINET IO-Device	Sí
volumen de datos	
<ul style="list-style-type: none"> como datos útiles para variables de entrada / como PROFINET IO-Device / máx. 	512 byte
<ul style="list-style-type: none"> como datos útiles para variables de salida / como PROFINET IO-Device / máx. 	512 byte
<ul style="list-style-type: none"> como datos útiles para variables de entrada / por submódulo como PROFINET IO-Device 	240 byte
<ul style="list-style-type: none"> como datos útiles para variables de salida / por submódulo como PROFINET IO-Device 	240 byte
<ul style="list-style-type: none"> como datos útiles para el área de coherencia por 	240 byte

submódulo	
número de submódulos / por PROFINET IO-Device	32
datos de potencia / Telecontrol	
protocolo / soportado	
• TCP/IP	Sí
funciones del producto / gestión, configuración, ajustes	
función del producto / soporte de MIB	Sí
protocolo / soportado	
• SNMP v1	Sí
• DCP	Sí
• LLDP	Sí
software de configuración	
• necesario	STEP 7 V5.4 SP2 o superior / STEP 7 Professional V11 (TIA Portal) o superior
función de Identificación y Mantenimiento	
• I&M0 - Información específica del dispositivo	Sí
• I&M1 - ID de la instalación/ID de situación	Sí
funciones del producto / diagnóstico	
función del producto / diagnóstico basado en web	Sí
funciones del producto / Switch	
equipamiento del producto / Switch	Sí
función del producto	
• gestionada por switch	No
• con IRT / Switch PROFINET IO	Sí
• configuración con STEP 7	Sí
funciones del producto / redundancia	
función del producto	
• redundancia de anillo	Sí
• gestor de redundancia	No
protocolo / soportado / procedimiento de redundancia MRP	Sí
funciones del producto / seguridad	
función del producto	
• protección por contraseña bus para aplicaciones Web	No
• ACL - IP based	Sí
• ACL - IP based para PLC/Routing	No
• desconexión de servicios no necesarios	Sí
• bloqueo de la comunicación mediante puertos físicos	Sí
• archivo de registro para acceso no autorizado	No
funciones del producto / hora	
función del producto / soporte de SICLOCK	Sí
función del producto / retransmisión de sincronización horaria	Sí
protocolo / soportado	
• NTP	Sí
información adicional / enlaces de Internet	
enlace de Internet	
• a la página web: Guía de selección TIA Selection Tool	http://www.siemens.com/tia-selection-tool
• a la página web: Comunicación industrial	http://www.siemens.com/simatic-net
• a la página web: Industry Mall	https://mall.industry.siemens.com
• a la página web: Centro de información y descarga	http://www.siemens.com/industry/infocenter
• a la página web: Archivo gráfico	http://automation.siemens.com/bilddb
• a la página web: CAx-Download-Manager	http://www.siemens.com/cax
• a la página web: Industry Online Support	https://support.industry.siemens.com
información de seguridad	
información de seguridad	Siemens suministra productos y soluciones con funciones de seguridad

2.6 PS 307 (5 A)

SIEMENS

Hoja de datos

6ES7307-1EA01-0AA0



SIMATIC PS307/1AC/DC24V/5A

SIMATIC S7-300 fuente de alimentación estabilizada PS307
 entrada: AC 120/230 V salida:
 DC 24 V/5 A

Entrada	
Entrada	AC monofásica
• Observación	Cambio de rango automático
tensión de alimentación	
• 1 con AC valor nominal	120 V
• 2 con AC valor nominal	230 V
tensión de entrada	
• 1 con AC	85 ... 132 V
• 2 con AC	170 ... 264 V
Entrada de rango amplio	No
Resistencia a sobretensiones	2,3 x U _e nom, 1,3 ms
Respaldo de red	Con U _e = 93/187 V
Respaldo de red con la nom, mín.	20 ms; Con U _e = 93/187 V
Frecuencia nominal de red 1	50 Hz
Frecuencia nominal de red 2	60 Hz
Rango de frecuencia de red	47 ... 63 Hz
intensidad de entrada	
• con valor nominal de la tensión de entrada 120 V	2,3 A
• con valor nominal de la tensión de entrada 230 V	1,2 A
Limitación de la intensidad de conexión (+ 25 °C), máx.	20 A
duración de la limitación de intensidad de conexión con 25 °C	
• máx.	3 ms
I ² t, máx.	1,2 A ² ·s
Fusible de entrada incorporado	T 3,15 A/250 V (no accesible)
Protección del cable de red (IEC 898)	Interruptor magnetotérmico recomendado: a partir de 6 A característica C
Salida	
Salida	Tensión continua estabilizada y aislada galvánicamente
Tensión nominal U _s nom DC	24 V
• tensión de salida en la salida 1 con DC valor nominal	24 V
Tolerancia total, estática ±	3 %
Compens. estática variación de red, aprox.	0,1 %
Compens. estática variación de carga, aprox.	0,5 %

Ondulación residual entre picos, máx.	50 mV
Ondulación residual entre picos, típ.	10 mV
Spikes entre picos, máx. (ancho de banda aprox. 20 MHz)	150 mV
Spikes entre picos, típ. (ancho de banda aprox. 20 MHz)	20 mV
función del producto tensión de salida es ajustable	No
Ajuste de la tensión de salida	-
Pantalla normal	LED verde para 24 V O.K.
Comportamiento al conectar desconectar	Sin rebase transitorio de Ua (arranque suave)
Retardo de arranque, máx.	2 s
Subida de tensión, típ.	10 ms
Intensidad nominal la nom	5 A
Rango de intensidad	0 ... 5 A
potencia activa entregada típico	120 W
intensidad de sobrecarga breve	
• con cortocircuito durante el arranque típico	20 A
• con cortocircuito en servicio típico	20 A
duración de la capacidad de sobrecarga con sobreintensidad	
• con cortocircuito durante el arranque	100 ms
• con cortocircuito en servicio	100 ms
Posibilidad de conex. en paralelo para aumento de potencia	Sí
Rendimiento	
Rendimiento con Ua nominal, la nominal, aprox.	87 %
Pérdidas con Ua nom, la nom, aprox.	18 W
Regulación	
Compens. dinám. variación de red (Ue nom \pm 15%), máx.	0,1 %
Compens. dinám. variación de carga (Ia: 50/100/50%), Ua \pm típ.	1 %
Tiempo de recuperación escalón de carga 50 a 100%, típ.	0,3 ms
Tiempo de recuperación escalón de carga 100 a 50%, típ.	0,3 ms
Protección y vigilancia	
Protección sobretensión en salida	Lazo de regulación adicional, desconexión < 28,8 V, rearmado automático
Limitación de intensidad	5,5 ... 6,5 A
propiedad de la salida resistente a cortocircuitos	Sí
Prot. contra cortocircuito	Corte electrónico, rearmado automático
intensidad de cortocircuito sostenido valor eficaz	
• máx.	7 A
Seguridad	
Aislamiento galvánico primario secundario	Sí
aislamiento galvánico	Tensión de salida MBTS/SELV Us según EN 60950-1 y EN 50178
Clase de protección	Clase I
corriente de fuga	
• máx.	3,5 mA
• típico	0,5 mA
Grado de protección (EN 60529)	IP20
Homologaciones	
Marcado CE	Sí
Aprobación UL/cUL (CSA)	cULus-Listed (UL 508, CSA C22.2 No. 142), File E143289
Protección contra explosiones	ATEX (EX) II 3G Ex nA II T4; cULus (ANSI/ISA 12.12.01, CSA C22.2 No.213) Class I, Div. 2, Group ABCD, T4, File E330455
certificado de aptitud NEC Class 2	No
Homologación FM	Class I, Div. 2, Group ABCD, T4
Homologación CB	No
certificado de aptitud homologación EAC	Sí
Homologación para la construcción naval	En el sistema S7-300
CEM	
Emisión de interferencias	EN 55022 clase B

Limitación de armónicos en red	EN 61000-3-2
Inmunidad a interferencias	EN 61000-6-2
condiciones ambientales	
temperatura ambiente	0 ... 60 °C
<ul style="list-style-type: none"> ● durante el funcionamiento <ul style="list-style-type: none"> — Observación ● durante el transporte ● durante el almacenamiento 	Con convección natural -40 ... +85 °C -40 ... +85 °C
Clase de humedad según EN 60721	Clase climática 3K3, 5 ... 95% sin condensación
Mecánica	
Sistema de conexión	conexión por tornillo
Conexiones	L, N, PE: 1 borne de tornillo resp. para 0,5 ... 2,5 mm ² monofilar/flexible L+, M: 3 bornes de tornillo resp. para 0,5 ... 2,5 mm ²
<ul style="list-style-type: none"> ● entrada de red ● salida ● contactos auxiliares 	-
anchura de la caja	60 mm
altura de la caja	125 mm
profundidad de la caja	120 mm
distancia que debe respetarse	
<ul style="list-style-type: none"> ● arriba ● abajo ● izquierda ● derecha 	40 mm 40 mm 0 mm 0 mm
Peso aprox.	0,6 kg
propiedad del producto de la caja carcasa disponible en hilera	Sí
Montaje	Para montar en perfil soporte S7
accesorios mecánicos	Adaptador para fijación sobre perfil normalizado (6EP1971-1BA00)
MTBF con 40 °C	2 480 589 h
notas adicionales	Siempre que no se diga lo contrario, son aplicables todos los datos para la tensión nominal de entrada y una temperatura ambiente de +25 °C

2.7 Pasarela IBOX Modbus Server



Honeywell Life Safety Iberia
C/Pau Vila 15-19; 08911
Badalona Barcelona
T. 902 03 05 45; Internacional:+34932424236
www.honeywelllifesafety.es
infohlsiberia@honeywell.com

iBox Modbus Server

Pasarela para la integración de centrales de
detección de incendio

ID3000 / ID3002 / ID60 / ID50

en sistemas de supervisión y control con
protocolo Modbus

Manual de usuario

MN-DT-958
20 JUNIO 2011
(IBOX-MBS-NID3000 / v10 r13 esp)

Toda la información contenida en este documento puede ser modificada sin previo aviso

ÍNDICE

1.	Descripción	3
1.1	Introducción	3
1.2	Funcionalidad	4
1.3	Capacidad de iBox ModBus	5
2.	El interfaz Modbus de iBox ModBus	6
2.1	Descripción	6
2.2	Definición de señales	6
2.3	Funciones soportadas	6
3.	LNK-BoxMB. Software de configuración y monitorización para la gama iBox ModBus Server	7
3.1	Introducción	7
3.2	Definición del proyecto.....	8
3.3	Configuración de las conexiones.....	12
3.4	Señales.....	14
3.5	Valores de eventos	16
3.6	Valores codificados a bit.....	18
3.7	Enviar la configuración a iBox ModBus	19
3.8	El visor de señales.....	20
3.9	Comandos de sistema	21
3.10	Ficheros	22
4.	Proceso de configuración y solución de problemas	23
4.1	Pre-requisitos	23
4.2	Proceso de puesta en marcha	23
5.	Conexiones	26
6.	Características técnicas	28
7.	Características funcionales	29
8.	Dimensiones	30
9.	Mapa de direcciones Modbus	31
10.	Proceso para configurar el puerto RS232 en la central Notifier.....	35

1. Descripción

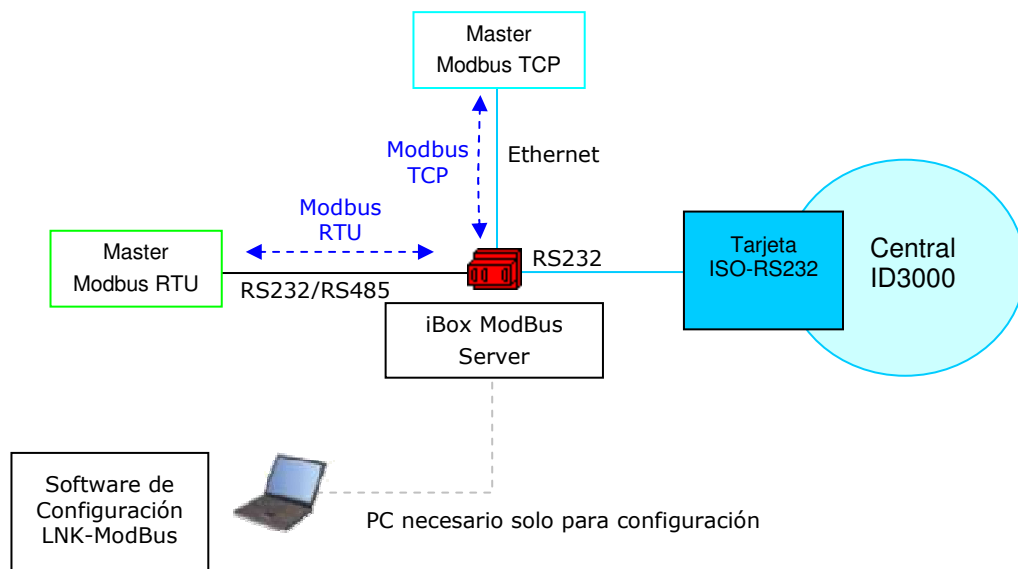
1.1 Introducción

Integración de centrales de detección de incendios Notifier ID3000 series en sistemas de supervisión/control que incorporen conectividad Modbus master, usando la pasarela *iBox Modbus Server - Notifier ID3000*.

El objetivo de esta integración es hacer disponible los estados de los puntos de las centrales de detección de incendio Notifier ID3000 / ID3002 / ID60 / ID50 desde sistemas de supervisión/control con conectividad Modbus master. Para ello, la pasarela trabaja, desde el punto de vista del sistema Modbus, actuando como un dispositivo esclavo Modbus respondiendo a interrogaciones de datos provenientes del Modbus master, y desde el punto de vista del sistema Notifier, actuando como un dispositivo serie conectado a su puerto RS232, y sirviendo los datos recibidos de Notifier hacia el lado Modbus.

iBox ModBus se conecta a la central Notifier por el puerto RS232 de la placa base o por el puerto ISO-RS232 (tarjeta opcional), este último es el recomendado, y usando una velocidad de transmisión de 2400bps.

Si hay más de una central Notifier conectadas en red, cualquiera de ellas puede ser integrada usando iBox ModBus, pero solamente una. iBox ModBus se puede conectar físicamente a una central y comunicarse con otra que esté conectada a la red Notifier.



Integración de centrales de detección de incendio Notifier ID3000 series usando *iBox Modbus Server*.

1.2 Funcionalidad

Visión general

El protocolo de comunicación Notifier ID3000 está basado en eventos, el estado de los elementos del sistema (detectores, módulos, etc.) se transmite mediante el protocolo en forma de eventos cuando estos se producen.

La misión de iBox ModBus consiste en asociar los estados de los elementos del sistema Notifier ID3000 con valores en direcciones de registros Modbus.

iBox ModBus emplea una asociación fija entre elementos de Notifier ID3000 y direcciones de registros Modbus, es decir, cada elemento de la central se corresponde con una dirección de registro Modbus prefijada. El valor en el registro Modbus para representar cada estado de la central es configurable usando el software LNK-BoxMB de una forma simple e intuitiva.

El proceso de configuración de iBox ModBus consiste básicamente en lo siguiente:

- Introducir los parámetros de comunicación deseados para el lado Modbus y para el lado Notifier.
- Asignar los valores deseados en el lado Modbus para cada estado a integrar de los elementos de Notifier.
- Una vez la configuración ha sido hecha con el software LNK-BoxMB, debe enviarse a iBox ModBus vía conexión serie a través del puerto de consola, e iBox ModBus reiniciará con la nueva configuración activa.

El valor numérico que representará, en los registros Modbus, los diferentes estados posibles de los puntos Notifier puede ser seleccionado en el proceso de configuración.

iBox ModBus se puede configurar como esclavo Modbus TCP o como esclavo Modbus RTU (RS232/RS485).

iBox ModBus soporta la capacidad completa de una central Notifier.

Se permite el control de la central Notifier, se permite enviar comandos hacia la central.

Todos los elementos de una central Notifier ID3000 completa (detectores, salidas y zonas) están configurados por defecto en iBox ModBus.

También todos los eventos generales son detectados por iBox ModBus y traducidos a Modbus:

La operativa de la integración es como sigue:

Una vez iBox ModBus está configurado y conectado a ambos sistemas (Notifier y Modbus), se mantiene un mensaje de "mantenimiento de la comunicación" con la central Notifier, siendo este mensaje la petición/respuesta del estado del panel. También se "escucha" continuamente a la espera de recibir nuevos eventos de Notifier. Con cada evento recibido, el nuevo estado es actualizado en la memoria de iBox ModBus y queda disponible para ser leído por el dispositivo Modbus master.

Como se ha mencionado antes, el protocolo de comunicación de Notifier está basado en mensajes espontáneos, es decir, solamente se envían mensajes cuando hay cambio de estado de los elementos, en el momento en que estos se producen. Debido a esto, cuando iBox ModBus se pone en marcha, el estado actual del sistema Notifier es desconocido, para proceder a la sincronización de ambos equipos, iBox ModBus envía

mensajes de petición de estado a la central y ésta informa a iBox ModBus sobre el estado actual, tanto de la propia central como de los elementos. A parte de ello, cíclicamente se van realizando peticiones de cierta información que no se envía de forma espontanea por la central, como por ejemplo los valores analógicos.

1.3 Capacidad de iBox ModBus

Elemento	Max.	Notas
Número de centrales	1	iBox ModBus puede integrar una sola central, no importa si esta en la red con otras.
Número de Puntos	Todos los de una central completa	Número de puntos definidos en iBox ModBus.

Ref.: IBOX-MBS-NID3000

2. El interfaz Modbus de iBox ModBus

2.1 Descripción

iBox ModBus actúa como un dispositivo esclavo en su interfaz Modbus, este interfaz puede ser el puerto Ethernet (si se usa Modbus TCP), o el puerto RS232 o el puerto RS485 (si se usa Modbus RTU). Para acceder a los puntos y recursos de iBox ModBus desde el sistema Modbus, se deben especificar como direcciones de registro aquellas configuradas de forma fija en iBox ModBus, correspondientes a elementos Notifier. Vea en detalle el mapa de direcciones de registros Modbus más abajo en este documento.

2.2 Definición de señales

Cada señal definida en iBox ModBus corresponde a un elemento Notifier. Cada estado posible del elemento (FIRE, FIRE DISABLED, TEST...) en el sistema Notifier puede ser libremente asociado a un valor numérico en Modbus. Este valor numérico será el valor del punto leído desde Modbus cuando el elemento Notifier asociado está en este estado. Desde el punto de vista de Modbus todos los registros son de tipo analógico.

2.3 Funciones soportadas

Las funciones Modbus 03 y 04 (*read holding registers* y *read input registers*) se pueden usar para leer registros Modbus.

La función Modbus 06 debe ser usada para escribir registros Modbus.

Si se usan '*poll records*' para leer más de un registro, es necesario que el rango de direcciones solicitadas contenga direcciones validas, sino el correspondiente código de error Modbus será retornado.

Todos los registros son de 2 bytes y su contenido se expresa en MSB..LSB.

Los códigos de error Modbus están totalmente soportados, serán enviados siempre que sea solicitada una acción Modbus o una dirección no válida.

3. LNK-BoxMB. Software de configuración y monitorización para la gama iBox ModBus Server

3.1 Introducción

LNK-BoxMB es un software compatible con Windows® desarrollado específicamente para monitorizar y configurar la gama de pasarelas iBox ModBus Server. Es posible configurar todos los protocolos externos disponibles en la gama iBox ModBus Server, y mantener diferentes configuraciones de clientes basadas en un proyecto LNK-BoxMB para cada instalación distinta. Manteniendo siempre en el disco duro una copia de los ficheros de la última configuración para cada cliente y protocolo externo, es decir para cada proyecto.

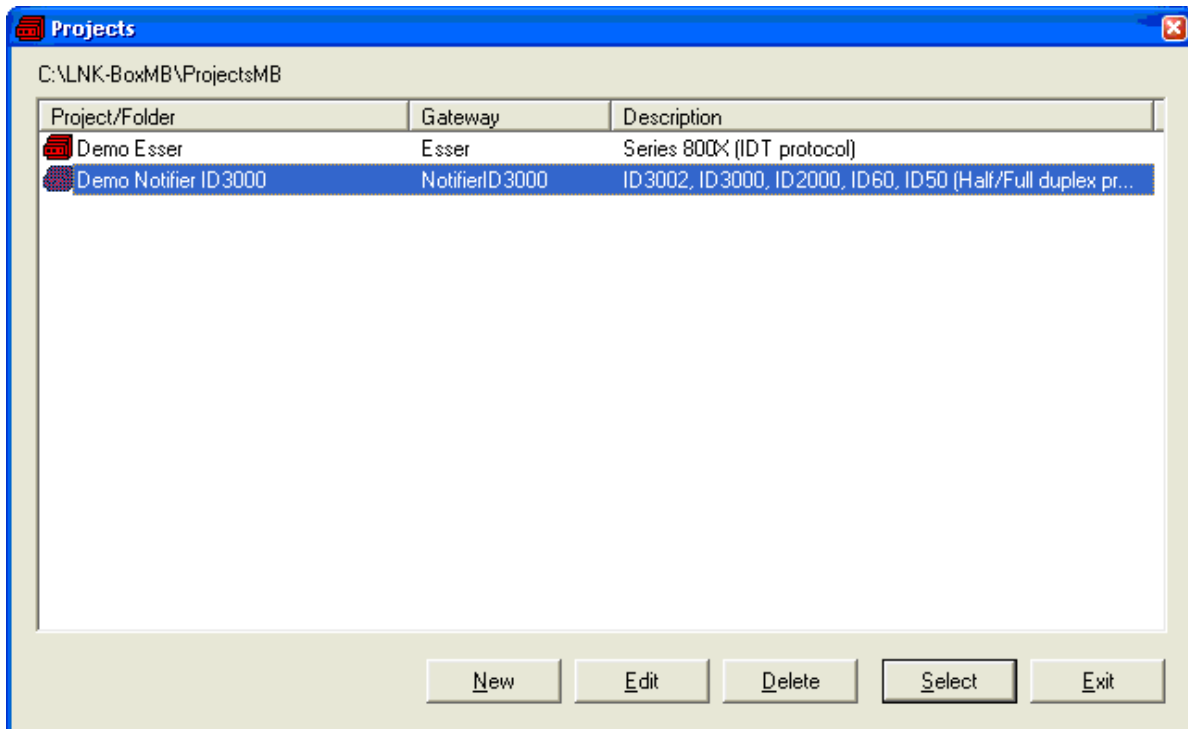
Desde LNK-BoxMB, así como permite configurar la lista de las señales de integración y los parámetros de conexión para cada protocolo externo, también permite seleccionar el puerto serie a usar para conectar con iBox ModBus Server y el uso de alguna herramienta para monitorizar y depurar el dispositivo. En este documento se explican algunas de estas herramientas pero solo alguna de ellas, el resto de herramientas y comandos disponibles para depurar no se explican aquí por que son para uso exclusivo bajo las recomendaciones del soporte técnico de Notifier.

LNK-BoxMB permite configurar todas las series iBox ModBus Server independientemente del sistema externo o el protocolo usado. Para cada sistema externo, LNK-BoxMB tiene una ventana diferente de configuración. Periódicamente, se proporcionan nuevas versiones gratuitas de este programa en las que están disponibles las nuevas integraciones de sistemas externos.

3.2 Definición del proyecto

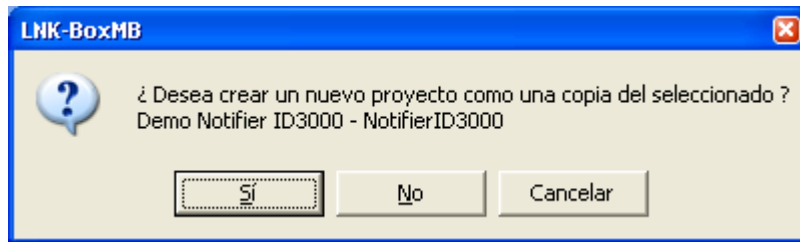
El primer paso a realizar en LNK-BoxMB para una nueva instalación es crear el proyecto de instalación dándole un nombre descriptivo. Al crear un proyecto, se crea una nueva carpeta con el nombre del proyecto conteniendo los ficheros de configuración necesarios que dependerán del protocolo externo seleccionado para el proyecto. Es muy recomendable que se cree un nuevo proyecto para cada nueva instalación, para evitar sobrescribir la configuración de otros proyectos existentes que usen el mismo protocolo externo y perder así los datos de configuración. La carpeta de proyectos esta ubicada en AppFolder\ProjectsMB, donde AppFolder es la carpeta de instalación de LNK-BoxMB (Por defecto C:\Archivos de Programa\LNK-BoxMB). Dentro de la carpeta de proyectos, se creará una nueva carpeta para cada proyecto definido en LNK-BoxMB con los ficheros necesarios para el proyecto.

Cuando se abre LNK-BoxMB, aparece la ventana de selección de proyecto invitando a seleccionar un proyecto existente o a crear uno nuevo. Con la instalación estándar de LNK-BoxMB se instala un proyecto *demo* para cada protocolo externo soportado. Puede crear un nuevo proyecto o seleccionar un proyecto *demo* del protocolo externo que desee, y crear el nuevo proyecto a partir de este *demo* seleccionado.

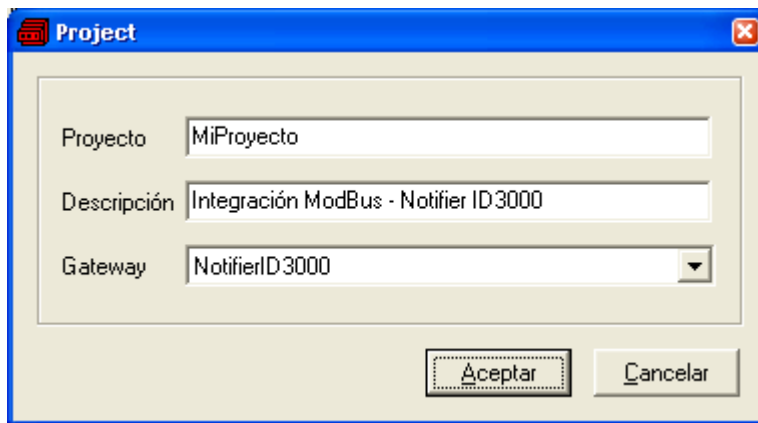


Ventana de selección de proyecto

Para crear un nuevo proyecto, seleccione un proyecto que use el mismo protocolo externo que usted quiere usar en su nuevo proyecto y presione el botón *Nuevo*. Entonces LNK-BoxMB le preguntara si quiere crear una copia del proyecto seleccionado (útil para instalaciones idénticas o similares) o si desea crear un proyecto completamente nuevo.



Si selecciona *Sí*, deberá especificar un nombre y una descripción del nuevo proyecto, que contendrá una copia de la configuración del proyecto seleccionado. Si selecciona *No*, puede especificar un nombre, una descripción y un protocolo externo a usar de la lista de protocolos externos disponibles.



Al *Aceptar*, se creará una nueva carpeta dentro de la carpeta de proyectos con el nombre dado al proyecto, esta carpeta contendrá los ficheros plantilla de configuración si el proyecto es uno completamente nuevo, o una copia de los ficheros de configuración si es una copia de un proyecto seleccionado.

Puede encontrar una descripción de los ficheros creados para un proyecto basado en protocolo Notifier ID3000 en la sección *Ficheros* de este documento.

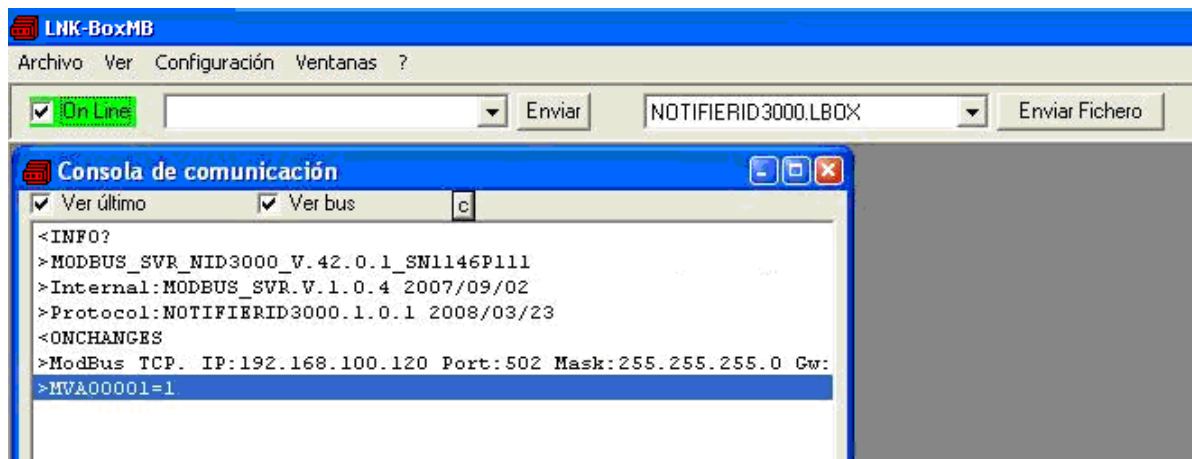
De todas las posibilidades de LNK-BoxMB, mientras se está desconectado de iBox ModBus (trabajando off-line) solo se puede modificar la configuración de la integración y generar el fichero binario de configuración a enviar a iBox ModBus, permitiéndole de esta forma hacer estas tareas más cómodamente en la oficina sin la necesidad de estar conectado físicamente a iBox ModBus. Antes de poder realizar cualquier acción de monitorización o envío de configuración hacia iBox ModBus, se debe establecer la conexión entre iBox ModBus y el PC donde se ejecuta LNK-BoxMB. Para realizarlo seguir los pasos siguientes:

1. Asegurarse de que iBox ModBus está funcionando y correctamente conectado al sistema Modbus vía la conexión Ethernet (Modbus TCP) o la conexión serie (Modbus RTU) y a la central Notifier vía la conexión RS232 (consultar los detalles para esta conexión y la asignación de pines en la sección *Conexiones* de este documento).
2. Conectar un puerto serie libre del PC al puerto serie de iBox ModBus marcado como **PC Console**. (Use el cable serie estándar suministrado junto con iBox ModBus o cree su propio cable siguiendo la asignación de pines especificadas en la sección *Conexiones* en este documento).

3. Seleccione en LNK-BoxMB el puerto serie del PC usado para la conexión con iBox ModBus. Use el menú: Configuración --> Conexión



4. Marque la casilla *off-line* situada bajo la barra de menú (cambiará automáticamente a *on-line*), en este momento LNK-BoxMB pedirá identificación a iBox ModBus a través de la conexión por el puerto de consola, si esta conexión esta bien, iBox ModBus responderá con su identificación (este proceso puede ser monitorizado en la ventana de la *Consola de Comunicación*, como se muestra en la figura de abajo).



Una vez conectado a iBox ModBus, todas las opciones de LNK-BoxMB están operativas.

Para monitorizar la comunicación entre iBox ModBus y el dispositivo Modbus master, seleccione el menú *Ver -> Bus -> Modbus*. Se abrirá la ventana del *Visor de comunicación Modbus*. Esta ventana muestra en tiempo real, tanto la comunicación entre iBox ModBus y el dispositivo Modbus master, como ciertos mensajes de funcionamiento referidos al protocolo interno (Modbus) enviados por iBox ModBus.

3.3 Configuración de las conexiones

Para configurar los parámetros de comunicación de iBox ModBus, y los valores Modbus para cada posible estado, seleccione el menú *Configuración -> Gateway*. Se abrirá la ventana *Configuración Notifier ID3000*.

Seleccione la pestaña *Conexión* para configurar los parámetros de comunicación.

Se configuran dos tipos de información en esta ventana, la referente al lado Modbus y la referente al lado Notifier.

Parámetros de configuración del lado Modbus:

The image shows a configuration window for Modbus. It has two main sections. The top section is for 'ModBus TCP' and contains four input fields: 'IP' with the value '192.168.100.120', 'Net Mask' with '255.255.255.0', 'Gateway' (empty), and 'Port' with '502'. The bottom section is for 'ModBus RTU RS232/485' and contains four dropdown menus: 'Conexión' (RS485), 'Baudios' (9600), 'Paridad' (none), and 'Slave' (1). A large arrow labeled '1' points to the radio button for 'ModBus TCP'. Smaller arrows labeled '2' through '9' point to each of the input fields in the two sections.

Configuración del Interfaz Modbus

1. Seleccione el tipo de conexión deseada (TCP o RTU).

Si selecciona Modbus TCP, entonces:

2. Entre la dirección IP para iBox ModBus.
3. Entre la máscara de subred para iBox ModBus.
4. Entre la dirección del router por defecto que tiene que usar iBox ModBus, dejar en blanco si no hay necesidad de dirección del router.
5. Entre el puerto TCP a usar, por defecto 502.

Si selecciona Modbus RTU, entonces:

6. Seleccione el puerto a usar (RS232 o RS485).
7. Seleccione la velocidad de transmisión a usar.
8. Seleccione la paridad a usar.
9. Entre el número de esclavo Modbus para iBox ModBus.

Parámetros de configuración del lado Notifier:

Configuración del Interfaz Notifier.

1. Número de la central Notifier a integrar, aunque solo se puede integrar una central, puede ser una central físicamente conectada a iBox ModBus o cualquier otra conectada en red (red Notifier), así el número a introducir aquí puede ser:
 - 0 en caso que la central a integrar sea la físicamente conectada a iBox ModBus.
 - 1 a 64 en caso de central en red ID2net.
2. Protocolo Notifier a usar:
 - Seleccione *Full Duplex* si se usa el puerto RS232 de la placa base de la central. Vea el anexo 1 para detalles del proceso a seguir para configurar este puerto en la central.
 - **Seleccione *Half Duplex* si usa la tarjeta opcional ISO-RS232. Esta es la conexión recomendada.** Vea el anexo 1 para detalles del proceso a seguir para configurar este puerto en la central.
3. Velocidad de transmisión a usar para comunicarse con la central.
4. Bits de datos a usar para comunicarse con la central.
5. Paridad a usar para comunicarse con la central.
6. Tiempo de espera de respuesta de la central antes de repetir de nuevo el mensaje (en milisegundos). Tras cuatro intentos sin respuesta de la central, la señal de error de comunicación será activada.
7. Tiempo de espera entre envíos hacia la central (en milisegundos).
8. Tiempo de espera entre procesos polling de estados (en segundos).
9. Tipo de codificación de de los datos en los registros modbus. Si se selecciona (**opción recomendada**), los estados de los elementos se codificarán en bits de los registros Modbus, si no está seleccionado los registros Modbus mostraran un valor numérico relacionado con el último evento recibido de cada elemento.

3.4 Señales

Seleccione la pestaña *Puntos* para una descripción de las señales de iBox ModBus.

#	Dirección/Formula	R/W	Señal	Valores to read	Valores to write
1		R	Error comunicación	0-Normal, 1-Activo	
2		R	System Events (last event received)	128..365 (see Notifier documentation)	
3		R	Fire	0-Normal, 1-Activo	
4		R	Pre-Alarm	0-Normal, 1-Activo	
5		R	Fault	0-Normal, 1-Activo	
6		R	Device(s) disconnected	0-Normal, 1-Activo	
7		R	Evacuate	0-Normal, 1-Activo	
8		R	Test in progress	0-Normal, 1-Activo	
9		R	Engineer mode	0-Normal, 1-Activo	
10		R	ID3000 Network communications Fault	0-Normal, 1-Activo	
11		R	Engineer Mute	0-Normal, 1-Activo	
12		R	Sounders delayed mode	0-Normal, 1-Activo	
13		R	Sounders Off	0-Normal, 1-Activo	
14		R	Not used		
15		R	System in Day Mode	0-Normal, 1-Activo	
16		R	TX to Alarm Routing Equipment disabled	0-Normal, 1-Activo	
17		RW	CMD System Reset	0-Always	1-To execute
18		RW	CMD Terminate Test	0-Always	1-To execute
19		RW	CMD Silence Sounders	0-Always	1-To execute
20		RW	CMD Mute Buzzer	0-Always	1-To execute
21		RW	CMD Evacuate	0-Always	1-To execute
22		RW	CMD Test Sounders	0-Always	1-To execute
23		RW	CMD Resound Sounders	0-Always	1-To execute
	(Loop X 256) + Detector	RW	Detector Status (Loop:1..8, Detector:1..99)	(see Events Values table)	Enable/Disable
	(Loop X 256) + Module + 100	RW	Module Status (Loop:1..8, Module:1..99)	(see Events Values table)	Enable/Disable, Test Active/Deac
	2304 + Zone	RW	Zone Status (Zone:1..255)	(see Events Values table)	Enable/Disable, Start Test
	2304 + (Loop X 256) + Detector	R	Detector Analog value (Loop:1..8, Module: %		
	2304 + (Loop X 256) + Module + 100	R	Module Analog value (Loop:1..8, Module:1 %		

Lista de puntos

Esta ventana es solo para propósitos de información sobre las señales que existen en iBox ModBus y su funcionalidad.

1. *Dir/Formula*. Indica la dirección Modbus correspondiente a cada señal, con las siguientes observaciones:

- La dirección Modbus 1 corresponde a la señal de error de comunicación, habrá un valor de cero en este registro cuando la comunicación con la central sea correcta, y un uno cuando la central no esté respondiendo a iBox ModBus. Este registro es de solo lectura.
- La dirección Modbus 2 corresponde al último evento recibido desde la central, en este registro Modbus habrá directamente el código de evento recibido desde la central, es solo para propósitos de información y seguimiento de errores. Vea la documentación de Notifier para consultar la lista de los códigos de eventos y sus significados. Este registro es de solo lectura.
- Las direcciones Modbus 3 a 16 corresponden a eventos generales recibidos desde la central, habrá un valor de cero en este registro cuando el evento no esté activo, y un uno cuando esté activo. Vea la descripción de cada evento en la columna "señal". Estos registros son de solo lectura.
- Las direcciones Modbus 17 a 23 corresponden a comandos generales para enviar a la central. Escribiendo un uno en estos registros se enviará el comando correspondiente a la central. Vea una descripción de cada comando en la

columna "señal". Estos son registros de lectura/escritura, aunque una lectura de ellos siempre retorna un cero.

- Las direcciones Modbus 257 a 2559 corresponden a estados de los Detectores, Módulos y Zonas. Estos registros son de tipo lectura/escritura, esto significa que se leerá en estos registros el valor correspondiente al estado del detector asociado, modulo o zona, el valor leído es el definido para el correspondiente estado en la pestaña *valores de eventos* (ver abajo), y puede escribir también el correspondiente valor para enviar un comando a la central. El único comando permitido para detectores es *habilitado/deshabilitado*, para módulos *habilitado/deshabilitado* y *test activado/desactivado*, y para zonas *habilitado/deshabilitado* e *iniciar test*.

Para saber que dirección de registro Modbus corresponde a que detector, modulo o zona, use la siguiente formula:

Para detectores

Dirección Modbus = (Lazo x 256) + detector

Lazo: 1 a 8

Detectores: 1 a 99

Para módulos

Dirección Modbus = (Lazo x 256) + módulo + 100

Lazo: 1 a 8

Módulos: 1 a 99

Para zonas

Dirección Modbus = zona + 2304

Zonas: 1 a 255

Al final de este documento existe una tabla con el mapa de direcciones Modbus.

- Las direcciones Modbus 2561 a 4551 corresponden a los valores analógicos de los Detectores y Módulos. Estos registros son de tipo lectura. Estos registros se actualizan muy periódicamente, cada Timeout polling se actualiza uno de ellos.

Para saber que dirección de registro Modbus corresponde a que detector y módulo, use la siguiente formula:

Para detectores

Dirección Modbus = 2304 + (Lazo x 256) + detector

Lazo: 1 a 8

Detectores: 1 a 99

Para módulos

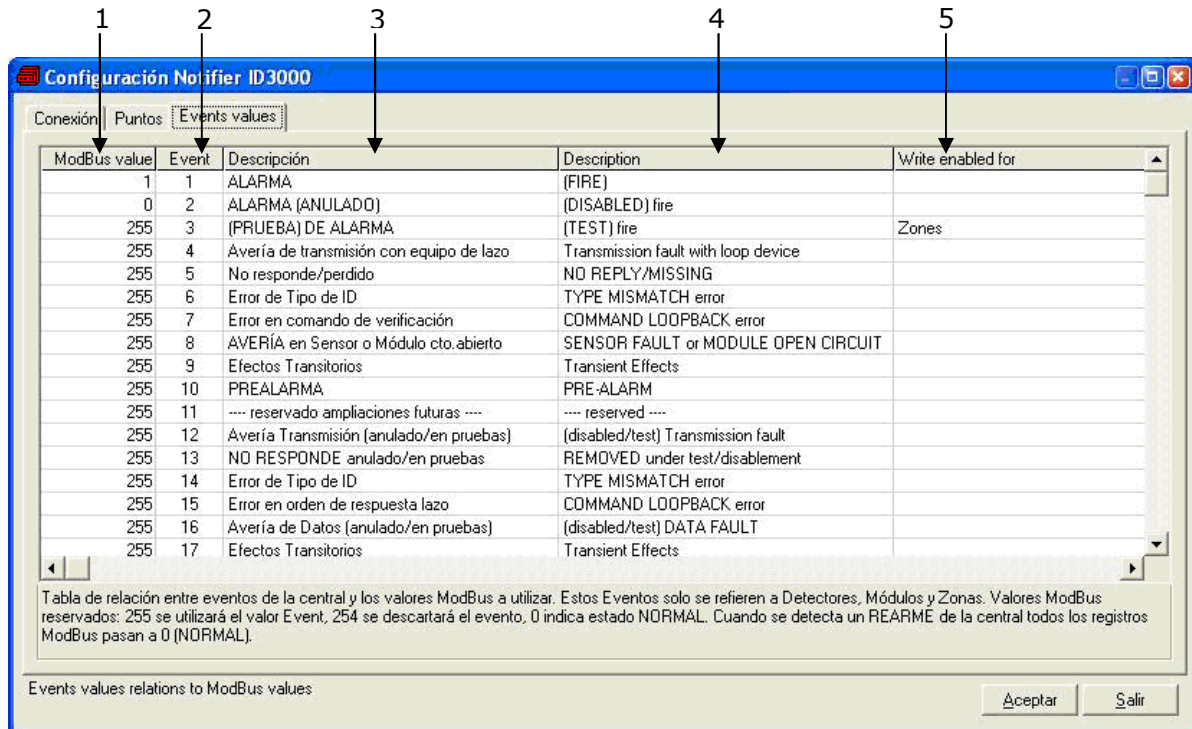
Dirección Modbus = 2304 + (Lazo x 256) + módulo + 100

Lazo: 1 a 8

Módulos: 1 a 99

3.5 Valores de eventos

Seleccione la pestaña *Valores de Eventos* para configurar los valores deseados en el registro Modbus para cada posible evento de la central. (**Se recomienda usar la opción de registros codificados a bit**, este tipo de codificación de valores directamente asociados a eventos puede no reflejar el estado correcto en determinadas situaciones).



Lista de puntos

La ventana es solamente una tabla de relaciones entre los códigos de eventos usados por la central y los valores que reflejan estos códigos de eventos en los registros Modbus.

1. *Valor Modbus*. Indica el valor que aparecerá en el registro Modbus cuando este evento ocurra en la central. Cuando el evento desaparezca, entonces el valor en el registro Modbus vuelve a cero. Entrar un valor de 255 aquí significa que el valor en el registro Modbus será el mismo que el del código para el evento en la central (el indicado en la columna "Evento"). Entrar un valor de 254 aquí significa que este evento no será tenido en cuenta por iBox Modbus.
2. *Evento*. Indica el código usado por la central para cada evento, solo a nivel informativo.
3. *Descripción*. La descripción de cada evento, solo a nivel informativo, en castellano.
4. *Descripción*. La descripción de cada evento, solo a nivel informativo, en inglés.
5. *Escritura habilitada para*. Indica que objeto en la central acepta este evento, solo a nivel informativo.

Comentarios:

- El comando test (Fuego) solo se usa en Zonas, y solo para probar la alarma de incendio de la zona.

- Los comandos `'device ENABLED'` y `'device DISABLED'` son utilizables por Detectores, Módulos y Zonas, y son solo para habilitar/deshabilitar.
- Los comandos `'output module test activation'` y `'output module test deactivation'` son utilizables solo por Módulos de tipo salida, y son solo para activar/desactivar el test del modulo.
- Después de recibir un evento de reinicio de la central, todos los valores en los registros Modbus vuelven a cero.

En la columna *valor Modbus* puede entrar el valor deseado individualmente por celda o puede auto enumerar celdas consecutivas, para lo cual tiene que seguir los pasos siguientes:

1. Seleccione usando el botón izquierdo del ratón (clic y arrastrar) todas las filas en la lista a las que quiere asignar valores automáticamente (deben ser filas consecutivas).
2. Haga clic con el botón derecho del ratón sobre los campos seleccionados y seleccione la opción *Auto Enumeración* del menú desplegable que aparecerá.

ModBus value	Event	Descripción
1	1	ALARMA
0	2	ALARMA (ANULADO)
255	3	(PRUEBA) DE ALARMA
255	4	Avería de transmisión
255	5	No responde/perdido
255	6	Error de Tipo de ID
255	7	Error en comando de
255	8	en Sensor o
255	9	Efectos Transitorios
255	10	PREALARMA
255	11 reservado ampliac
255	12	Avería Transmisión (a
255	13	NO RESPONDE anul

3. Entre el primer valor a asignar

4. Entre el incremento entre las asignaciones consecutivas. Por ejemplo seleccionando 255 como primer valor y un incremento de 0, los valores generados serán siempre 255.

3.6 Valores codificados a bit

Seleccione la pestaña *Valores codificados a bit* para ver el uso de los bits. Esta pestaña solo está disponible cuando se activa el check "States in Modbus Registers Bit Coded is checked" en la pestaña Conexión. Esta característica está disponible solo en los equipos con firmware V.42.0.8 or superior.

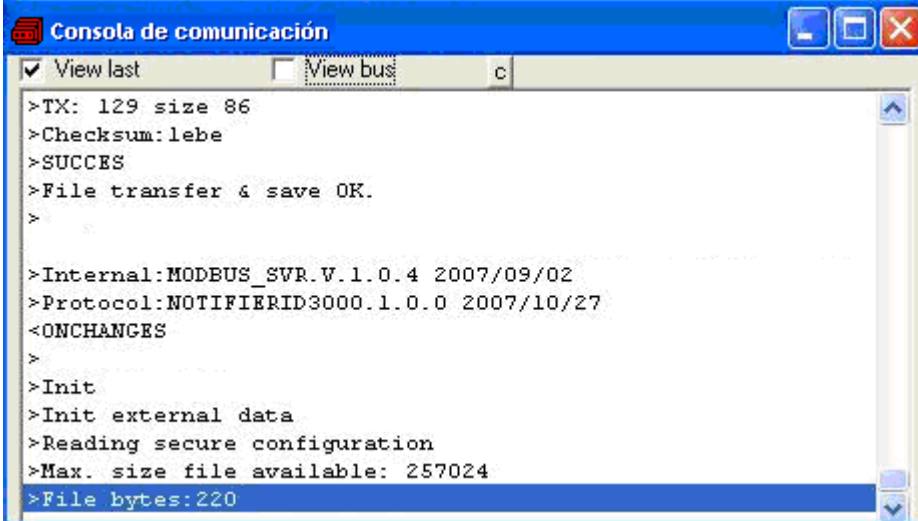
Element	Bit	Bit coded Status in Modbus register	Value	Value to write in Modbus register to execute Command
Detectors	0	Alarm	0	Enable detector
	1	PreAlarm	1	Disable detector
	2	Fault		
	3	Disabled		
	4	Test		
Modules	0	Activate	0	Enable module
	1	PreAlarm	1	Disable module
	2	Fault	2	Activate module
	3	Disabled	3	Deactivate module
	4	Test		
Zones	0	Alarm	0	Enable zone
	1	PreAlarm	1	Disable zone
	2	Fault	2	Start test zone
	3	Walk Test	3	Stop test zone
	4	Partially Disabled		
	5	Totally Disabled		

Se muestran los bits que se usarán para indicar los estados de los elementos.

1. Elemento. Indica el tipo de elemento al que aplican las siguientes columnas.
2. Bit. Indica el bit dentro del word (registro Modbus) donde se codifica el estado del elemento. Cuando el estado en el elemento está activo el valor del bit estará a 1, en otro caso será 0.
3. Estado. Indica el estado del elemento al que corresponde el bit.
4. Valor. Indica el valor analógico que tiene que escribirse en el registro Modbus para ejecutar el comando en la central, esto solo se usa para enviar comandos al elemento asociado en la central (por ejemplo escribiendo 1 se deshabilita el detector asociado).
5. Valor a escribir en registro Modbus para ejecutar el comando. Indicación del comando que será ejecutado en la central al escribir el dato en el registro Modbus.

3.7 Enviar la configuración a iBox ModBus

Cuando la configuración ha sido guardada (botón *Aceptar*) y el fichero binario de configuración para iBox ModBus ha sido generado (recuerde seleccionar sí cuando se le pregunte si quiere generar el fichero de v), para enviar el fichero de configuración a iBox ModBus haga clic en el botón **Enviar Fichero**. El proceso de transmisión del fichero puede monitorizarse en la ventana de la *Consola de Comunicación* de iBox ModBus. Si la transmisión del fichero es correcta, iBox ModBus se reiniciará automáticamente con la nueva configuración cargada.

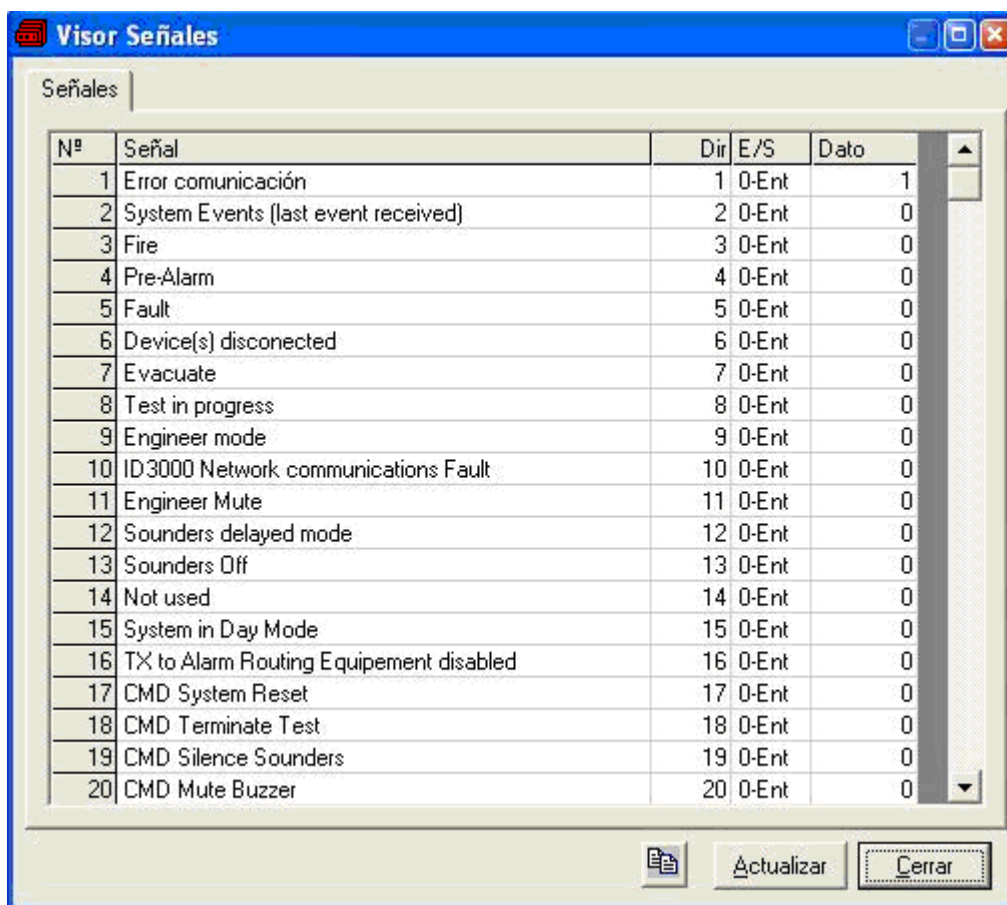


```
Consola de comunicación
View last View bus c
>TX: 129 size 86
>Checksum: lebe
>SUCCES
>File transfer & save OK.
>
>
>Internal:MODEBUS_SVR.V.1.0.4 2007/09/02
>Protocol:NOTIFIERID3000.1.0.0 2007/10/27
<ONCHANGES
>
>Init
>Init external data
>Reading secure configuration
>Max. size file available: 257024
>File bytes:220
```

Recuerde que guardar la configuración y generar el fichero binario para iBox ModBus solo guarda en el disco duro del PC los ficheros de configuración. **No olvide enviar el fichero binario a iBox ModBus (usando el botón *Enviar Fichero*) después de guardar la configuración.**

3.8 El visor de señales

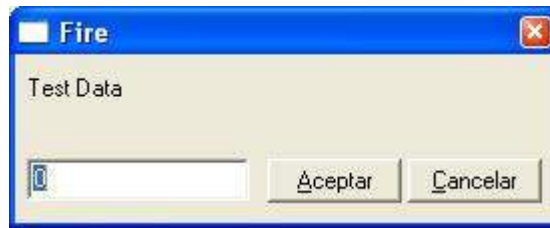
Una vez iBox ModBus esta funcionando con la configuración correcta, para supervisar el estado de las señales configuradas seleccione el menú *Ver -> Señales*. Se abrirá la ventana del Visor de Señales. Esta ventana muestra todas las señales activas en iBox ModBus con sus principales parámetros de configuración y sus valores en tiempo real en la columna *Dato*. Después de un reinicio de iBox ModBus o tras enviar un fichero de configuración a iBox ModBus, todos los valores de las señales se actualizarán automáticamente en el visor de señales, en caso de que conecte con iBox ModBus cuando éste ya este funcionando, debe pulsar el botón *Actualizar* para forzar actualización de los valores en la ventana, presione solo una vez el botón para actualizar todos los valores de las señales, a partir de ese momento los valores de las señales se mantendrán actualizados hasta que la conexión se cierre.



Nº	Señal	Dir	E/S	Dato
1	Error comunicación	1	0-Ent	1
2	System Events (last event received)	2	0-Ent	0
3	Fire	3	0-Ent	0
4	Pre-Alarm	4	0-Ent	0
5	Fault	5	0-Ent	0
6	Device(s) disconnected	6	0-Ent	0
7	Evacuate	7	0-Ent	0
8	Test in progress	8	0-Ent	0
9	Engineer mode	9	0-Ent	0
10	ID3000 Network communications Fault	10	0-Ent	0
11	Engineer Mute	11	0-Ent	0
12	Sounders delayed mode	12	0-Ent	0
13	Sounders Off	13	0-Ent	0
14	Not used	14	0-Ent	0
15	System in Day Mode	15	0-Ent	0
16	TX to Alarm Routing Equipement disabled	16	0-Ent	0
17	CMD System Reset	17	0-Ent	0
18	CMD Terminate Test	18	0-Ent	0
19	CMD Silence Sounders	19	0-Ent	0
20	CMD Mute Buzzer	20	0-Ent	0

El visor de señales se puede usar aunque solo se conecte uno de los sistemas a integrar a iBox ModBus, *Notifier* o *Modbus*, y es muy útil para supervisión y testeo.

Es posible forzar un valor específico para cualquier señal para su testeo, para hacerlo solo tiene que hacer doble clic en la fila y seleccionar el valor deseado, luego pulse *Aceptar* en la ventana de Test de datos. El nuevo valor entrado estará disponible a través del interfaz *Modbus*, de la misma forma que si hubiera sido recibido desde la central *Notifier*.

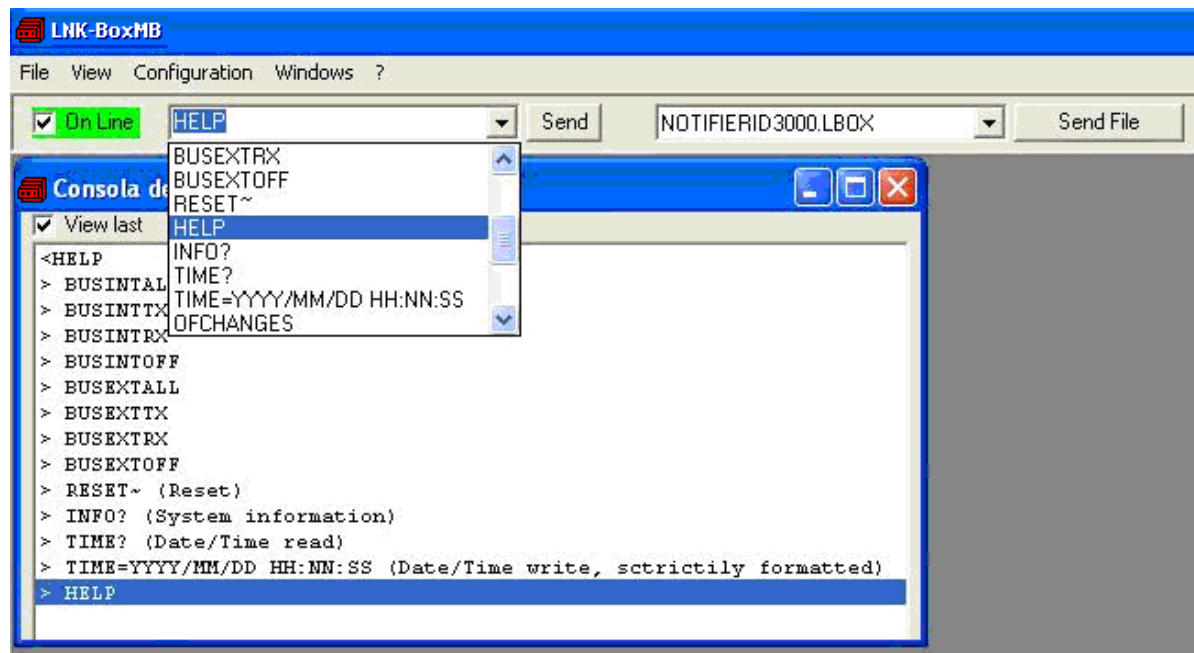


Esta ventana es muy útil para probar la comunicación en el lado Modbus desde el dispositivo Modbus master por ejemplo, sin necesidad de tener la central Notifier conectada y funcionando.

La ventana del visor de señales tiene un botón para copiar al Portapapeles de Windows todos los datos que hay en la ventana (en formato separado por tabuladores).

3.9 Comandos de sistema

LNK-BoxMB incluye una opción para enviar a iBox ModBus un juego de comandos de sistema con el propósito testear y controlar su funcionamiento; esta juego de comandos esta disponible en la *lista de comandos* como se muestra en la figura de abajo. Para enviar un comando a iBox ModBus simplemente selecciónelo de la lista, o tecléelo con el formato correcto, y pulse *Enter* o haga clic en el botón *Enviar*. iBox ModBus actuará de acuerdo con el comando recibido; este proceso puede ser monitorizado en la ventana de la *Consola de Comunicación*. El uso de algunos de estos comandos puede ser crítico para el funcionamiento normal de iBox ModBus, teniendo esto presente use estos comandos bajo las recomendaciones del soporte técnico de Notifier. iBox ModBus devolverá una lista de los comandos más comúnmente usados y el su formato correcto tras enviar el comando HELP.



3.10 Ficheros

LNK-BoxMB guarda la configuración de la integración en los siguientes ficheros dentro de la carpeta de proyecto:

PROJECT.INI	Fichero .ini que contiene información general referente al proyecto.
NOTIFIERID3000.INI	Fichero .ini que contiene información referente a la ventana de conexión y otros ajustes especiales.
NOTIFIERID3000.EVT	Fichero de texto (valores separados por tabuladores) con la información de los valores de los eventos (Lista de los valores de los eventos).
NOTIFIERID3000.LBOX	Archivo binario creado a partir de la información de los ficheros anteriores. Este es el fichero que realmente se envía a iBox ModBus.

Se recomienda realizar una copia de seguridad, en un medio externo, de la carpeta del proyecto que contiene estos ficheros una vez que el proceso de configuración ha finalizado. De esta forma, podrá realizar cambios de configuración futuros en caso de reinstalación de LNK-BoxMB, por ejemplo en caso de un fallo en el disco duro del PC donde LNK-BoxMB se instaló previamente.

La configuración no se puede transferir desde iBox ModBus a LNK-BoxMB, solo desde LNK-BoxMB a iBox ModBus.

4. Proceso de configuración y solución de problemas

4.1 Pre-requisitos

Es necesario tener el dispositivo Modbus master operativo y bien conectado al puerto Modbus de iBox ModBus, recuerde respetar el máximo de 15 metros de distancia de cable si se usa comunicación RS232.

Es necesario tener el panel Notifier con un puerto RS232 operativo y a una distancia máxima de 15 metros del lugar de instalación de iBox ModBus (debido a la comunicación RS232).

Notifier no suministra los conectores, cables de conexión ni PC para LNK-BoxMB con esta integración estándar. Los artículos suministrados por Notifier para esta integración son:

- Dispositivo iBox Modbus Server con el Firmware del protocolo externo Notifier ID3000 cargado.
- Software LNK-BoxMB para configurar iBox ModBus.
- Cable de consola necesario para enviar la configuración a iBox ModBus.
- Documentación del producto.

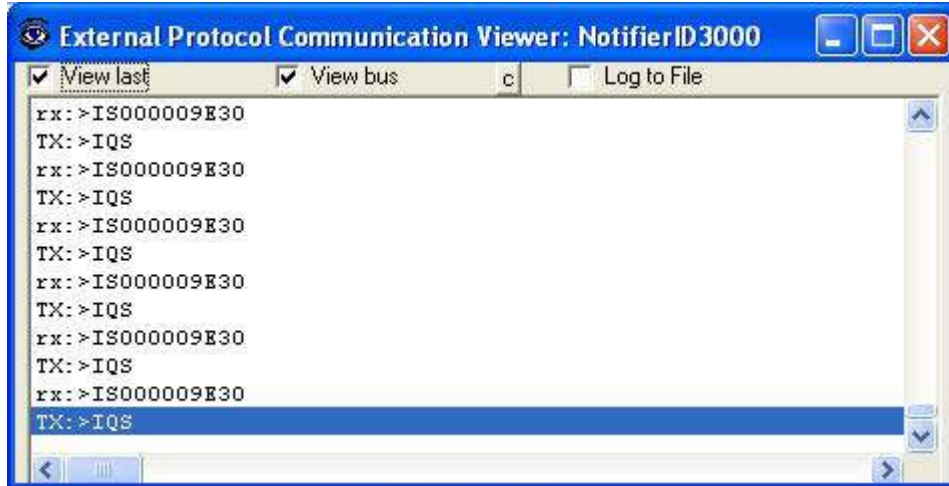
4.2 Proceso de puesta en marcha

1. Instale LNK-BoxMB en su ordenador portátil, use el programa de 'setup' suministrado para ello y siguiendo las instrucciones del asistente de instalación.
2. Instale iBox ModBus en el lugar deseado. El montaje puede ser en un carril DIN o en una superficie estable no vibratoria (Se recomienda en carril DIN y montado dentro de un armario industrial metálico conectada a tierra, junto a la central Notifier).
3. Conecte el cable de comunicación desde el dispositivo master Modbus al puerto marcado como **Modbus** de iBox ModBus (use RS232, RS485 o Ethernet dependiendo del tipo de comunicación Modbus a usar). (Vea los detalles para este cable de comunicación en la sección *Conexiones* de este documento).
4. Conecte el cable de comunicación proveniente del puerto RS232 de la central Notifier al puerto marcado como **Notifier** de iBox ModBus. (Vea los detalles para este cable de comunicación en la sección *Conexiones* de este documento).
5. Ponga en marcha iBox ModBus. El voltaje de alimentación puede ser de 9 a 30 Vcc o bien 24 Vac. Respete la polaridad.

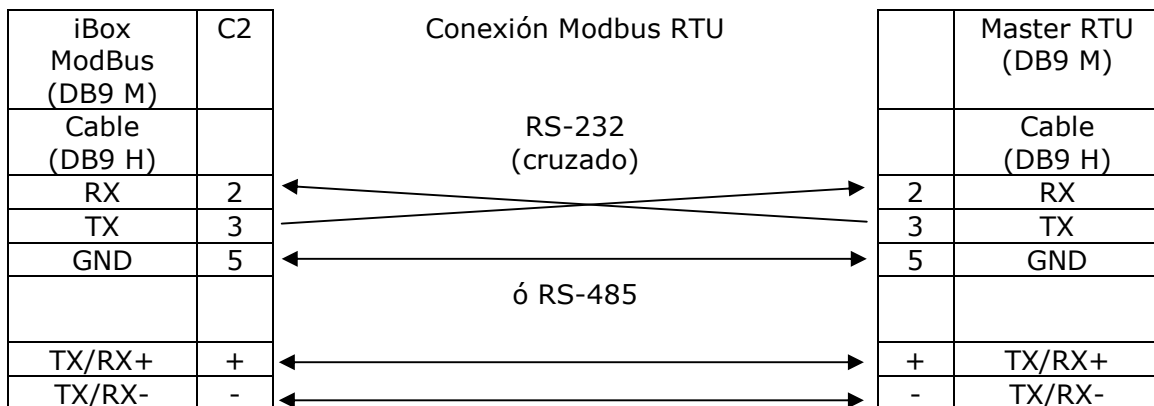
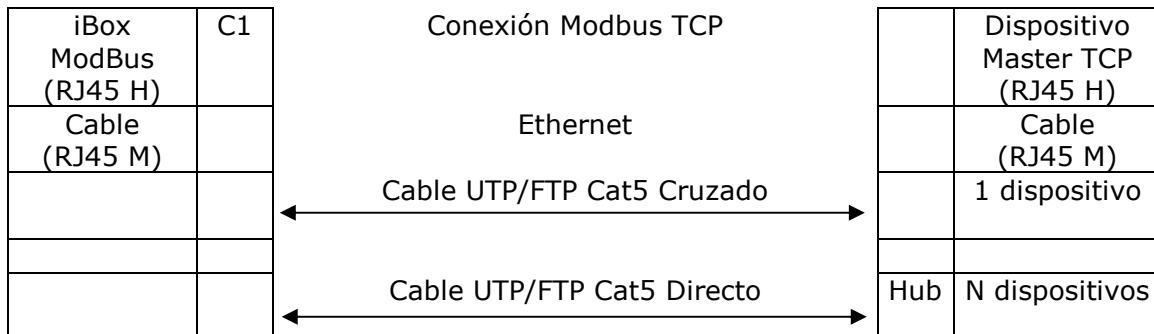
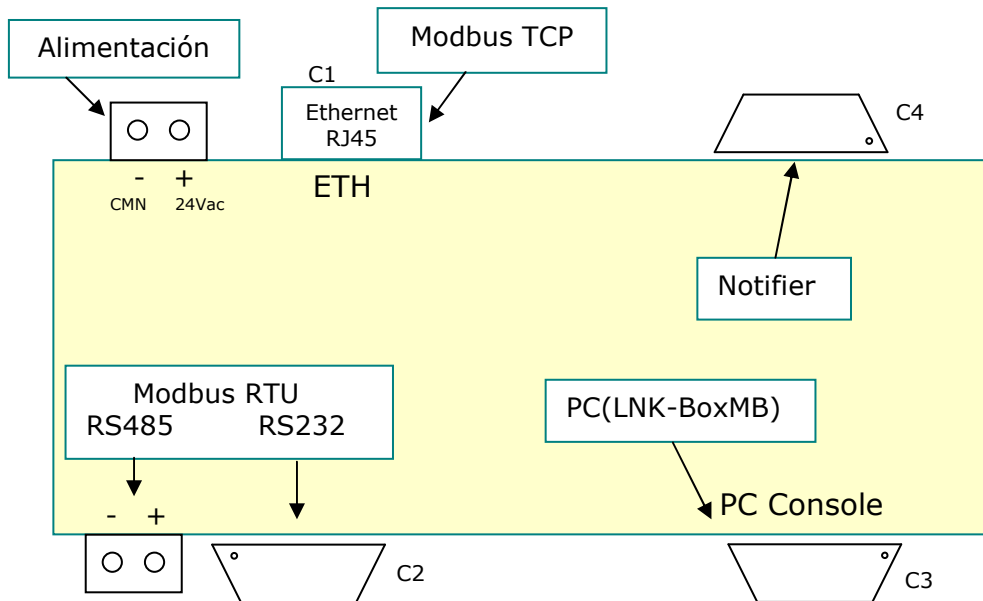
¡AVISO! Para evitar bucles de tierra que pueden dañar iBox ModBus y/o cualquier otro equipo conectado a él, recomendamos especialmente:

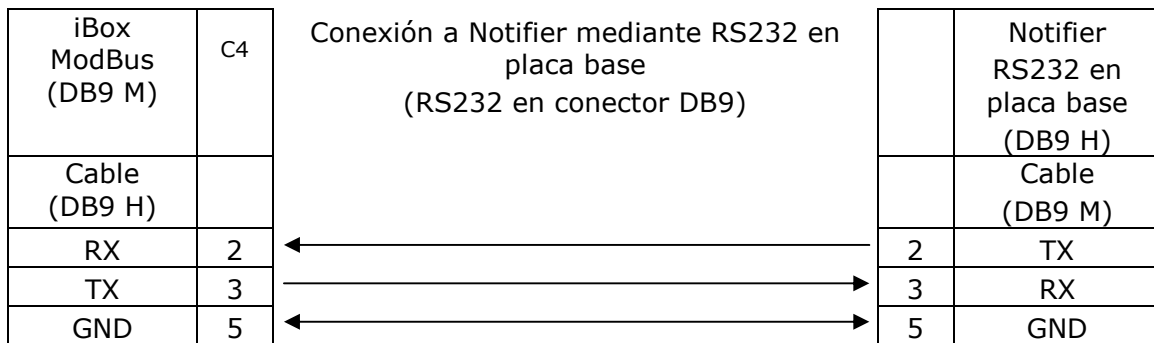
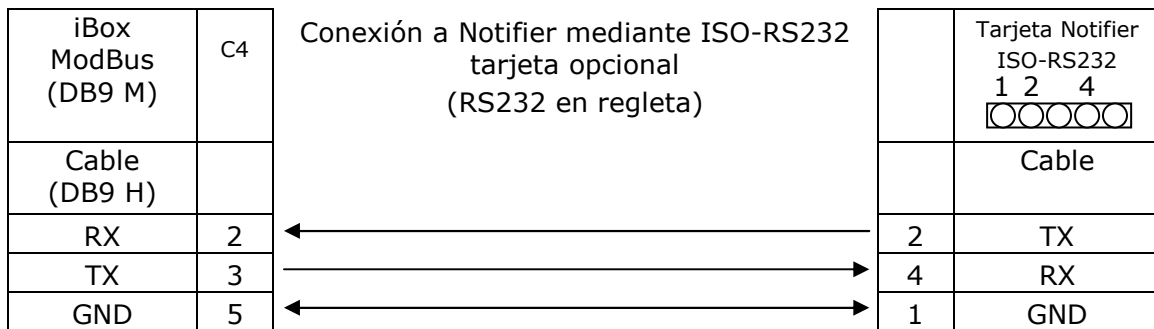
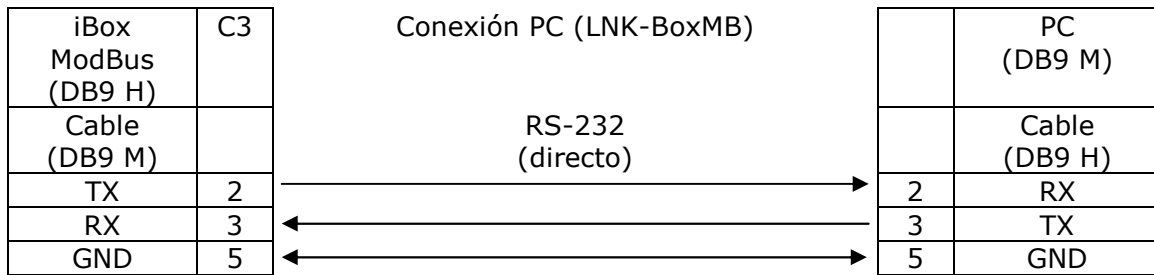
- El uso de una fuente de alimentación CC, flotante o con el terminal negativo conectado a tierra. **Nunca use una fuente de alimentación CC con el terminal positivo conectado a tierra.**
- Usar una fuente de alimentación CA solo si es flotante y no alimenta a ningún otro dispositivo.

tramas TX y otras rx en el visor, como se muestra en la figura de abajo. Esto significa que la comunicación con el panel Notifier esta bien. En caso de falta de actividad de comunicación entre iBox ModBus y Notifier, compruebe que el puerto RS232 del panel de Notifier esta operativo y bien configurado, y compruebe también el cable de comunicaciones usado para conectar ambos dispositivos. (Vea los detalles para este cable de comunicación en la sección *Conexiones* de este documento).

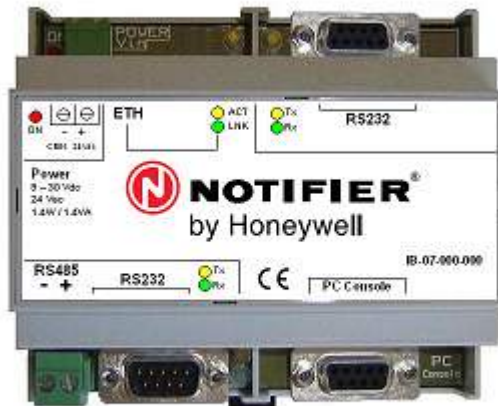


5. Conexiones





6. Características técnicas



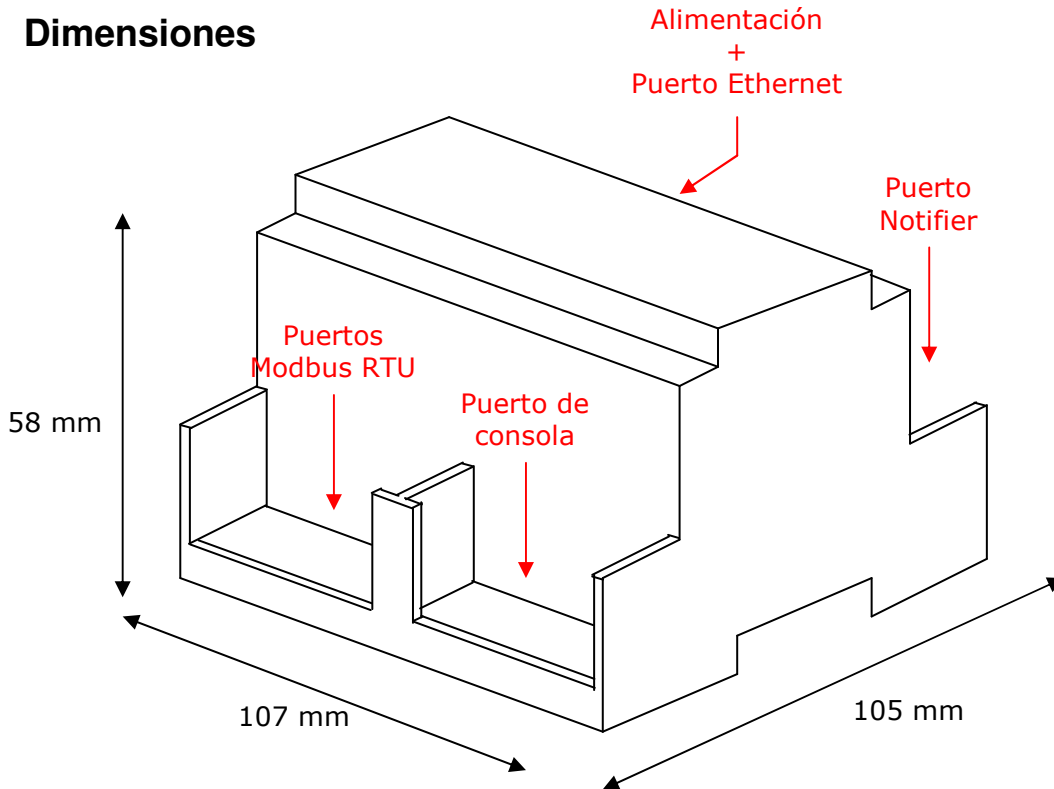
Envolvente	Plástico, tipo PC (UL 94 V-0). Dimensiones: 107mm x 105mm x 58mm.
Color	Gris. RAL 7035.
Alimentación	De 9 a 30Vcc +/-10% 1.4W. De 24Vca +/-10% 1.4VA. Conexión de alimentación tipo clema extraíble (2 bornes).
Montaje	Sobremesa. Mural. Carril DIN EN60715 TH35.
Puerto Modbus TCP	1 x Ethernet 10BT RJ45.
Puertos Modbus RTU	1 x RS232. Conector DB9 macho (DTE). 1 x RS485. Conector tipo clema extraíble (2 bornes).
Puerto Notifier	1 x RS232. Conector DB9 macho (DTE).
Indicadores LED	1 x Alimentación. 2 x Actividad del puerto Notifier (Tx, Rx). 2 x Actividad puerto Modbus RTU (Tx, Rx). 2 x Puerto Ethernet (LNK, ACT).
Puerto de consola	RS232. Conector DB9 hembra (DCE).
Configuración	Vía el puerto de consola. ¹
Firmware	Permite actualizaciones vía el puerto de consola.
Temperatura de funcionamiento	De -40°C a +70°C
Humedad relativa de funcionamiento	De 5% a 95%, sin condensación
Protección	IP20 (IEC60529).
Conformidad RoHS	Cumple con la directiva RoHS (2002/95/CE).
Certificaciones	CE

¹ Junto con el dispositivo se suministra un cable estándar DB9 macho - DB9 hembra de 1,8 metros para conexión directa al puerto serie de un PC para configuración y monitorización del dispositivo. El software de configuración, para sistemas operativos Windows, también se suministra.

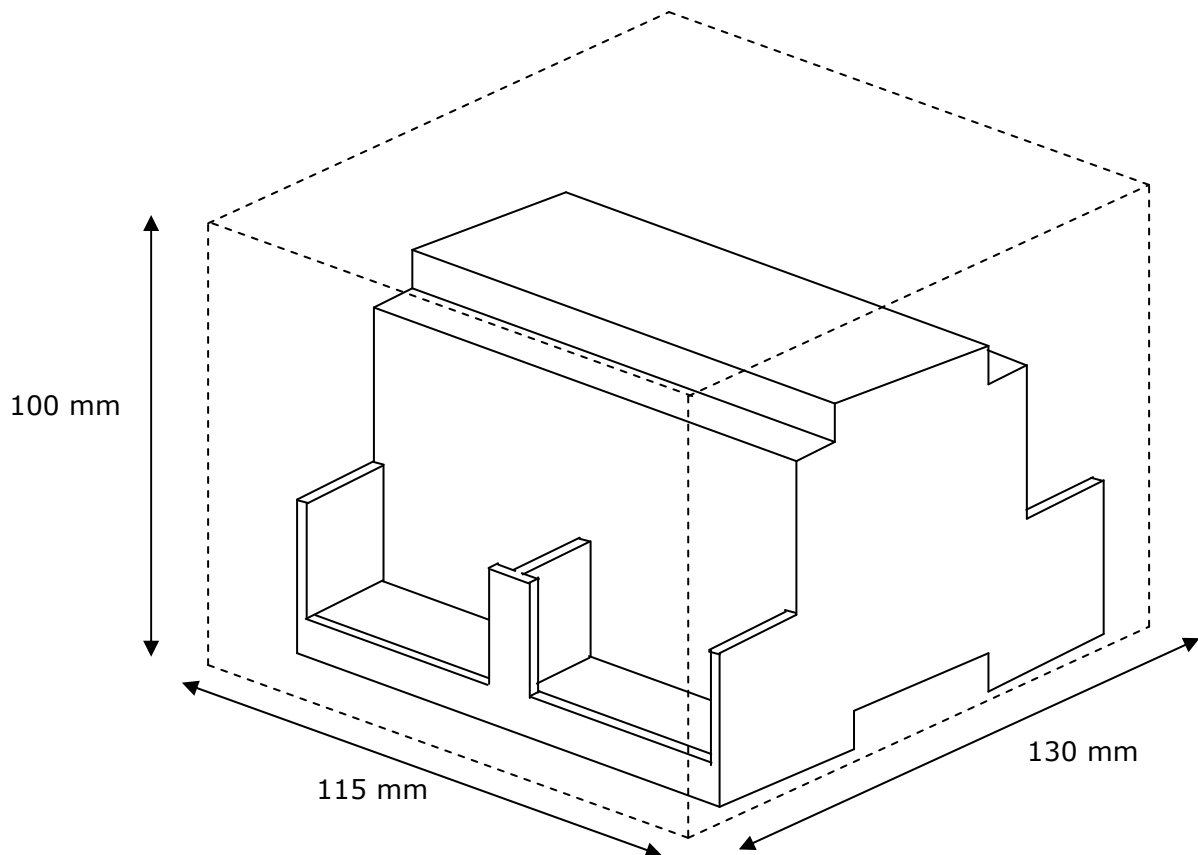
7. Características funcionales

Interfaz Notifier	
Tipo	Serie RS232 para conexión a puerto RS232 Notifier.
Parámetros configurables	<ul style="list-style-type: none"> • Número de central, permite integrar cualquier central en red. • Velocidad de transmisión, bits de datos y paridad. • Protocolo (Half o Full Duplex). • Tiempo de espera de respuesta de la central.
Interactividad con Notifier	<ul style="list-style-type: none"> • Permite lectura y escritura en Notifier (envío de comandos a la central). • El valor Modbus que refleja cada posible estado es totalmente configurable.
Interfaz Modbus	
Tipo de dispositivo	Esclavo.
Modos Modbus soportados	TCP, RTU RS232 o RS485.
Parámetros de configuración Modbus TCP	<ul style="list-style-type: none"> • Dirección IP. • Macara de Subred • Gateway por efecto. • Puerto TCP.
Parámetros de configuración Modbus RTU	<ul style="list-style-type: none"> • RS232/RS485. • Velocidad de transmisión, bits de datos y paridad. • Número de esclavo.
Puntos	
Tipos de registros Modbus	Todos los registros son de tipo UNSIGNED INT en el interfaz Modbus.

8. Dimensiones



Espacio libre recomendado para su instalación en un armario (sujeción mural o carril DIN), con previsión de espacio suficiente para las conexiones externas



9. Mapa de direcciones Modbus

La tabla siguiente muestra todo el mapa de direcciones Modbus de iBox ModBus (*Dir* significa dirección de registro Modbus).

Lazo	1				2				3				4			
	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir
1	257	1	357	1	513	1	613	1	769	1	869	1	1025	1	1125	
2	258	2	358	2	514	2	614	2	770	2	870	2	1026	2	1126	
3	259	3	359	3	515	3	615	3	771	3	871	3	1027	3	1127	
4	260	4	360	4	516	4	616	4	772	4	872	4	1028	4	1128	
5	261	5	361	5	517	5	617	5	773	5	873	5	1029	5	1129	
6	262	6	362	6	518	6	618	6	774	6	874	6	1030	6	1130	
7	263	7	363	7	519	7	619	7	775	7	875	7	1031	7	1131	
8	264	8	364	8	520	8	620	8	776	8	876	8	1032	8	1132	
9	265	9	365	9	521	9	621	9	777	9	877	9	1033	9	1133	
10	266	10	366	10	522	10	622	10	778	10	878	10	1034	10	1134	
11	267	11	367	11	523	11	623	11	779	11	879	11	1035	11	1135	
12	268	12	368	12	524	12	624	12	780	12	880	12	1036	12	1136	
13	269	13	369	13	525	13	625	13	781	13	881	13	1037	13	1137	
14	270	14	370	14	526	14	626	14	782	14	882	14	1038	14	1138	
15	271	15	371	15	527	15	627	15	783	15	883	15	1039	15	1139	
16	272	16	372	16	528	16	628	16	784	16	884	16	1040	16	1140	
17	273	17	373	17	529	17	629	17	785	17	885	17	1041	17	1141	
18	274	18	374	18	530	18	630	18	786	18	886	18	1042	18	1142	
19	275	19	375	19	531	19	631	19	787	19	887	19	1043	19	1143	
20	276	20	376	20	532	20	632	20	788	20	888	20	1044	20	1144	
21	277	21	377	21	533	21	633	21	789	21	889	21	1045	21	1145	
22	278	22	378	22	534	22	634	22	790	22	890	22	1046	22	1146	
23	279	23	379	23	535	23	635	23	791	23	891	23	1047	23	1147	
24	280	24	380	24	536	24	636	24	792	24	892	24	1048	24	1148	
25	281	25	381	25	537	25	637	25	793	25	893	25	1049	25	1149	
26	282	26	382	26	538	26	638	26	794	26	894	26	1050	26	1150	
27	283	27	383	27	539	27	639	27	795	27	895	27	1051	27	1151	
28	284	28	384	28	540	28	640	28	796	28	896	28	1052	28	1152	
29	285	29	385	29	541	29	641	29	797	29	897	29	1053	29	1153	
30	286	30	386	30	542	30	642	30	798	30	898	30	1054	30	1154	
31	287	31	387	31	543	31	643	31	799	31	899	31	1055	31	1155	
32	288	32	388	32	544	32	644	32	800	32	900	32	1056	32	1156	
33	289	33	389	33	545	33	645	33	801	33	901	33	1057	33	1157	
34	290	34	390	34	546	34	646	34	802	34	902	34	1058	34	1158	
35	291	35	391	35	547	35	647	35	803	35	903	35	1059	35	1159	
36	292	36	392	36	548	36	648	36	804	36	904	36	1060	36	1160	
37	293	37	393	37	549	37	649	37	805	37	905	37	1061	37	1161	
38	294	38	394	38	550	38	650	38	806	38	906	38	1062	38	1162	
39	295	39	395	39	551	39	651	39	807	39	907	39	1063	39	1163	
40	296	40	396	40	552	40	652	40	808	40	908	40	1064	40	1164	
41	297	41	397	41	553	41	653	41	809	41	909	41	1065	41	1165	
42	298	42	398	42	554	42	654	42	810	42	910	42	1066	42	1166	
43	299	43	399	43	555	43	655	43	811	43	911	43	1067	43	1167	
44	300	44	400	44	556	44	656	44	812	44	912	44	1068	44	1168	
45	301	45	401	45	557	45	657	45	813	45	913	45	1069	45	1169	
46	302	46	402	46	558	46	658	46	814	46	914	46	1070	46	1170	
47	303	47	403	47	559	47	659	47	815	47	915	47	1071	47	1171	
48	304	48	404	48	560	48	660	48	816	48	916	48	1072	48	1172	
49	305	49	405	49	561	49	661	49	817	49	917	49	1073	49	1173	
50	306	50	406	50	562	50	662	50	818	50	918	50	1074	50	1174	
51	307	51	407	51	563	51	663	51	819	51	919	51	1075	51	1175	
52	308	52	408	52	564	52	664	52	820	52	920	52	1076	52	1176	
53	309	53	409	53	565	53	665	53	821	53	921	53	1077	53	1177	
54	310	54	410	54	566	54	666	54	822	54	922	54	1078	54	1178	
55	311	55	411	55	567	55	667	55	823	55	923	55	1079	55	1179	
56	312	56	412	56	568	56	668	56	824	56	924	56	1080	56	1180	
57	313	57	413	57	569	57	669	57	825	57	925	57	1081	57	1181	
58	314	58	414	58	570	58	670	58	826	58	926	58	1082	58	1182	
59	315	59	415	59	571	59	671	59	827	59	927	59	1083	59	1183	
60	316	60	416	60	572	60	672	60	828	60	928	60	1084	60	1184	
61	317	61	417	61	573	61	673	61	829	61	929	61	1085	61	1185	
62	318	62	418	62	574	62	674	62	830	62	930	62	1086	62	1186	
63	319	63	419	63	575	63	675	63	831	63	931	63	1087	63	1187	
64	320	64	420	64	576	64	676	64	832	64	932	64	1088	64	1188	
65	321	65	421	65	577	65	677	65	833	65	933	65	1089	65	1189	
66	322	66	422	66	578	66	678	66	834	66	934	66	1090	66	1190	
67	323	67	423	67	579	67	679	67	835	67	935	67	1091	67	1191	

68	324	68	424	68	580	68	680	68	836	68	936	68	1092	68	1192
69	325	69	425	69	581	69	681	69	837	69	937	69	1093	69	1193
70	326	70	426	70	582	70	682	70	838	70	938	70	1094	70	1194
71	327	71	427	71	583	71	683	71	839	71	939	71	1095	71	1195
72	328	72	428	72	584	72	684	72	840	72	940	72	1096	72	1196
73	329	73	429	73	585	73	685	73	841	73	941	73	1097	73	1197
74	330	74	430	74	586	74	686	74	842	74	942	74	1098	74	1198
75	331	75	431	75	587	75	687	75	843	75	943	75	1099	75	1199
76	332	76	432	76	588	76	688	76	844	76	944	76	1100	76	1200
77	333	77	433	77	589	77	689	77	845	77	945	77	1101	77	1201
78	334	78	434	78	590	78	690	78	846	78	946	78	1102	78	1202
79	335	79	435	79	591	79	691	79	847	79	947	79	1103	79	1203
80	336	80	436	80	592	80	692	80	848	80	948	80	1104	80	1204
81	337	81	437	81	593	81	693	81	849	81	949	81	1105	81	1205
82	338	82	438	82	594	82	694	82	850	82	950	82	1106	82	1206
83	339	83	439	83	595	83	695	83	851	83	951	83	1107	83	1207
84	340	84	440	84	596	84	696	84	852	84	952	84	1108	84	1208
85	341	85	441	85	597	85	697	85	853	85	953	85	1109	85	1209
86	342	86	442	86	598	86	698	86	854	86	954	86	1110	86	1210
87	343	87	443	87	599	87	699	87	855	87	955	87	1111	87	1211
88	344	88	444	88	600	88	700	88	856	88	956	88	1112	88	1212
89	345	89	445	89	601	89	701	89	857	89	957	89	1113	89	1213
90	346	90	446	90	602	90	702	90	858	90	958	90	1114	90	1214
91	347	91	447	91	603	91	703	91	859	91	959	91	1115	91	1215
92	348	92	448	92	604	92	704	92	860	92	960	92	1116	92	1216
93	349	93	449	93	605	93	705	93	861	93	961	93	1117	93	1217
94	350	94	450	94	606	94	706	94	862	94	962	94	1118	94	1218
95	351	95	451	95	607	95	707	95	863	95	963	95	1119	95	1219
96	352	96	452	96	608	96	708	96	864	96	964	96	1120	96	1220
97	353	97	453	97	609	97	709	97	865	97	965	97	1121	97	1221
98	354	98	454	98	610	98	710	98	866	98	966	98	1122	98	1222
99	355	99	455	99	611	99	711	99	867	99	967	99	1123	99	1223

Lazo	5				6				7				8			
	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir
1	1281	1	1381	1	1537	1	1637	1	1793	1	1893	1	2049	1	2149	
2	1282	2	1382	2	1538	2	1638	2	1794	2	1894	2	2050	2	2150	
3	1283	3	1383	3	1539	3	1639	3	1795	3	1895	3	2051	3	2151	
4	1284	4	1384	4	1540	4	1640	4	1796	4	1896	4	2052	4	2152	
5	1285	5	1385	5	1541	5	1641	5	1797	5	1897	5	2053	5	2153	
6	1286	6	1386	6	1542	6	1642	6	1798	6	1898	6	2054	6	2154	
7	1287	7	1387	7	1543	7	1643	7	1799	7	1899	7	2055	7	2155	
8	1288	8	1388	8	1544	8	1644	8	1800	8	1900	8	2056	8	2156	
9	1289	9	1389	9	1545	9	1645	9	1801	9	1901	9	2057	9	2157	
10	1290	10	1390	10	1546	10	1646	10	1802	10	1902	10	2058	10	2158	
11	1291	11	1391	11	1547	11	1647	11	1803	11	1903	11	2059	11	2159	
12	1292	12	1392	12	1548	12	1648	12	1804	12	1904	12	2060	12	2160	
13	1293	13	1393	13	1549	13	1649	13	1805	13	1905	13	2061	13	2161	
14	1294	14	1394	14	1550	14	1650	14	1806	14	1906	14	2062	14	2162	
15	1295	15	1395	15	1551	15	1651	15	1807	15	1907	15	2063	15	2163	
16	1296	16	1396	16	1552	16	1652	16	1808	16	1908	16	2064	16	2164	
17	1297	17	1397	17	1553	17	1653	17	1809	17	1909	17	2065	17	2165	
18	1298	18	1398	18	1554	18	1654	18	1810	18	1910	18	2066	18	2166	
19	1299	19	1399	19	1555	19	1655	19	1811	19	1911	19	2067	19	2167	
20	1300	20	1400	20	1556	20	1656	20	1812	20	1912	20	2068	20	2168	
21	1301	21	1401	21	1557	21	1657	21	1813	21	1913	21	2069	21	2169	
22	1302	22	1402	22	1558	22	1658	22	1814	22	1914	22	2070	22	2170	
23	1303	23	1403	23	1559	23	1659	23	1815	23	1915	23	2071	23	2171	
24	1304	24	1404	24	1560	24	1660	24	1816	24	1916	24	2072	24	2172	
25	1305	25	1405	25	1561	25	1661	25	1817	25	1917	25	2073	25	2173	
26	1306	26	1406	26	1562	26	1662	26	1818	26	1918	26	2074	26	2174	
27	1307	27	1407	27	1563	27	1663	27	1819	27	1919	27	2075	27	2175	
28	1308	28	1408	28	1564	28	1664	28	1820	28	1920	28	2076	28	2176	
29	1309	29	1409	29	1565	29	1665	29	1821	29	1921	29	2077	29	2177	
30	1310	30	1410	30	1566	30	1666	30	1822	30	1922	30	2078	30	2178	
31	1311	31	1411	31	1567	31	1667	31	1823	31	1923	31	2079	31	2179	
32	1312	32	1412	32	1568	32	1668	32	1824	32	1924	32	2080	32	2180	
33	1313	33	1413	33	1569	33	1669	33	1825	33	1925	33	2081	33	2181	
34	1314	34	1414	34	1570	34	1670	34	1826	34	1926	34	2082	34	2182	
35	1315	35	1415	35	1571	35	1671	35	1827	35	1927	35	2083	35	2183	
36	1316	36	1416	36	1572	36	1672	36	1828	36	1928	36	2084	36	2184	
37	1317	37	1417	37	1573	37	1673	37	1829	37	1929	37	2085	37	2185	
38	1318	38	1418	38	1574	38	1674	38	1830	38	1930	38	2086	38	2186	
39	1319	39	1419	39	1575	39	1675	39	1831	39	1931	39	2087	39	2187	
40	1320	40	1420	40	1576	40	1676	40	1832	40	1932	40	2088	40	2188	
41	1321	41	1421	41	1577	41	1677	41	1833	41	1933	41	2089	41	2189	

42	1322	42	1422	42	1578	42	1678	42	1834	42	1934	42	2090	42	2190
43	1323	43	1423	43	1579	43	1679	43	1835	43	1935	43	2091	43	2191
44	1324	44	1424	44	1580	44	1680	44	1836	44	1936	44	2092	44	2192
45	1325	45	1425	45	1581	45	1681	45	1837	45	1937	45	2093	45	2193
46	1326	46	1426	46	1582	46	1682	46	1838	46	1938	46	2094	46	2194
47	1327	47	1427	47	1583	47	1683	47	1839	47	1939	47	2095	47	2195
48	1328	48	1428	48	1584	48	1684	48	1840	48	1940	48	2096	48	2196
49	1329	49	1429	49	1585	49	1685	49	1841	49	1941	49	2097	49	2197
50	1330	50	1430	50	1586	50	1686	50	1842	50	1942	50	2098	50	2198
51	1331	51	1431	51	1587	51	1687	51	1843	51	1943	51	2099	51	2199
52	1332	52	1432	52	1588	52	1688	52	1844	52	1944	52	2100	52	2200
53	1333	53	1433	53	1589	53	1689	53	1845	53	1945	53	2101	53	2201
54	1334	54	1434	54	1590	54	1690	54	1846	54	1946	54	2102	54	2202
55	1335	55	1435	55	1591	55	1691	55	1847	55	1947	55	2103	55	2203
56	1336	56	1436	56	1592	56	1692	56	1848	56	1948	56	2104	56	2204
57	1337	57	1437	57	1593	57	1693	57	1849	57	1949	57	2105	57	2205
58	1338	58	1438	58	1594	58	1694	58	1850	58	1950	58	2106	58	2206
59	1339	59	1439	59	1595	59	1695	59	1851	59	1951	59	2107	59	2207
60	1340	60	1440	60	1596	60	1696	60	1852	60	1952	60	2108	60	2208
61	1341	61	1441	61	1597	61	1697	61	1853	61	1953	61	2109	61	2209
62	1342	62	1442	62	1598	62	1698	62	1854	62	1954	62	2110	62	2210
63	1343	63	1443	63	1599	63	1699	63	1855	63	1955	63	2111	63	2211
64	1344	64	1444	64	1600	64	1700	64	1856	64	1956	64	2112	64	2212
65	1345	65	1445	65	1601	65	1701	65	1857	65	1957	65	2113	65	2213
66	1346	66	1446	66	1602	66	1702	66	1858	66	1958	66	2114	66	2214
67	1347	67	1447	67	1603	67	1703	67	1859	67	1959	67	2115	67	2215
68	1348	68	1448	68	1604	68	1704	68	1860	68	1960	68	2116	68	2216
69	1349	69	1449	69	1605	69	1705	69	1861	69	1961	69	2117	69	2217
70	1350	70	1450	70	1606	70	1706	70	1862	70	1962	70	2118	70	2218
71	1351	71	1451	71	1607	71	1707	71	1863	71	1963	71	2119	71	2219
72	1352	72	1452	72	1608	72	1708	72	1864	72	1964	72	2120	72	2220
73	1353	73	1453	73	1609	73	1709	73	1865	73	1965	73	2121	73	2221
74	1354	74	1454	74	1610	74	1710	74	1866	74	1966	74	2122	74	2222
75	1355	75	1455	75	1611	75	1711	75	1867	75	1967	75	2123	75	2223
76	1356	76	1456	76	1612	76	1712	76	1868	76	1968	76	2124	76	2224
77	1357	77	1457	77	1613	77	1713	77	1869	77	1969	77	2125	77	2225
78	1358	78	1458	78	1614	78	1714	78	1870	78	1970	78	2126	78	2226
79	1359	79	1459	79	1615	79	1715	79	1871	79	1971	79	2127	79	2227
80	1360	80	1460	80	1616	80	1716	80	1872	80	1972	80	2128	80	2228
81	1361	81	1461	81	1617	81	1717	81	1873	81	1973	81	2129	81	2229
82	1362	82	1462	82	1618	82	1718	82	1874	82	1974	82	2130	82	2230
83	1363	83	1463	83	1619	83	1719	83	1875	83	1975	83	2131	83	2231
84	1364	84	1464	84	1620	84	1720	84	1876	84	1976	84	2132	84	2232
85	1365	85	1465	85	1621	85	1721	85	1877	85	1977	85	2133	85	2233
86	1366	86	1466	86	1622	86	1722	86	1878	86	1978	86	2134	86	2234
87	1367	87	1467	87	1623	87	1723	87	1879	87	1979	87	2135	87	2235
88	1368	88	1468	88	1624	88	1724	88	1880	88	1980	88	2136	88	2236
89	1369	89	1469	89	1625	89	1725	89	1881	89	1981	89	2137	89	2237
90	1370	90	1470	90	1626	90	1726	90	1882	90	1982	90	2138	90	2238
91	1371	91	1471	91	1627	91	1727	91	1883	91	1983	91	2139	91	2239
92	1372	92	1472	92	1628	92	1728	92	1884	92	1984	92	2140	92	2240
93	1373	93	1473	93	1629	93	1729	93	1885	93	1985	93	2141	93	2241
94	1374	94	1474	94	1630	94	1730	94	1886	94	1986	94	2142	94	2242
95	1375	95	1475	95	1631	95	1731	95	1887	95	1987	95	2143	95	2243
96	1376	96	1476	96	1632	96	1732	96	1888	96	1988	96	2144	96	2244
97	1377	97	1477	97	1633	97	1733	97	1889	97	1989	97	2145	97	2245
98	1378	98	1478	98	1634	98	1734	98	1890	98	1990	98	2146	98	2246
99	1379	99	1479	99	1635	99	1735	99	1891	99	1991	99	2147	99	2247

Zona	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir	Detector	Dir	Módulo	Dir
1	2305	33	2337	65	2369	97	2401	129	2433	161	2465	193	2497	225	2529
2	2306	34	2338	66	2370	98	2402	130	2434	162	2466	194	2498	226	2530
3	2307	35	2339	67	2371	99	2403	131	2435	163	2467	195	2499	227	2531
4	2308	36	2340	68	2372	100	2404	132	2436	164	2468	196	2500	228	2532
5	2309	37	2341	69	2373	101	2405	133	2437	165	2469	197	2501	229	2533
6	2310	38	2342	70	2374	102	2406	134	2438	166	2470	198	2502	230	2534
7	2311	39	2343	71	2375	103	2407	135	2439	167	2471	199	2503	231	2535
8	2312	40	2344	72	2376	104	2408	136	2440	168	2472	200	2504	232	2536
9	2313	41	2345	73	2377	105	2409	137	2441	169	2473	201	2505	233	2537
10	2314	42	2346	74	2378	106	2410	138	2442	170	2474	202	2506	234	2538
11	2315	43	2347	75	2379	107	2411	139	2443	171	2475	203	2507	235	2539
12	2316	44	2348	76	2380	108	2412	140	2444	172	2476	204	2508	236	2540
13	2317	45	2349	77	2381	109	2413	141	2445	173	2477	205	2509	237	2541
14	2318	46	2350	78	2382	110	2414	142	2446	174	2478	206	2510	238	2542
15	2319	47	2351	79	2383	111	2415	143	2447	175	2479	207	2511	239	2543
16	2320	48	2352	80	2384	112	2416	144	2448	176	2480	208	2512	240	2544
17	2321	49	2353	81	2385	113	2417	145	2449	177	2481	209	2513	241	2545
18	2322	50	2354	82	2386	114	2418	146	2450	178	2482	210	2514	242	2546
19	2323	51	2355	83	2387	115	2419	147	2451	179	2483	211	2515	243	2547
20	2324	52	2356	84	2388	116	2420	148	2452	180	2484	212	2516	244	2548
21	2325	53	2357	85	2389	117	2421	149	2453	181	2485	213	2517	245	2549
22	2326	54	2358	86	2390	118	2422	150	2454	182	2486	214	2518	246	2550
23	2327	55	2359	87	2391	119	2423	151	2455	183	2487	215	2519	247	2551
24	2328	56	2360	88	2392	120	2424	152	2456	184	2488	216	2520	248	2552
25	2329	57	2361	89	2393	121	2425	153	2457	185	2489	217	2521	249	2553
26	2330	58	2362	90	2394	122	2426	154	2458	186	2490	218	2522	250	2554
27	2331	59	2363	91	2395	123	2427	155	2459	187	2491	219	2523	251	2555
28	2332	60	2364	92	2396	124	2428	156	2460	188	2492	220	2524	252	2556
29	2333	61	2365	93	2397	125	2429	157	2461	189	2493	221	2525	253	2557
30	2334	62	2366	94	2398	126	2430	158	2462	190	2494	222	2526	254	2558
31	2335	63	2367	95	2399	127	2431	159	2463	191	2495	223	2527	255	2559
32	2336	64	2368	96	2400	128	2432	160	2464	192	2496	224	2528		

10. Proceso para configurar el puerto RS232 en la central Notifier

Para habilitar el puerto RS232 de la placa base, siga estos pasos en el menú de la central (para desbloquear el teclado numérico y tener acceso al menú, gire la llave de servicio localizada en el panel frontal, junto al teclado numérico, un cuarto de vuelta a la derecha):

6. Configuración

6. Configuración Panel

18. Conf. RS232 en placa base

3. Protocolo Integración

9600 bps

Versión: 1: 003

Controles: habilitado/deshabilitado ***1**

Supervisión com.: deshabilitado

Supervisión tempo: 0

Resaltadas en negrita las selecciones obligatorias.

Contraseña del nivel 3 por defecto: 27835

Para habilitar el puerto RS232 de la tarjeta opcional ISO-RS232, siga estos pasos en el menú de la central (para desbloquear el teclado numérico y tener acceso al menú, gire la llave de servicio localizada en el panel frontal, junto al teclado numérico, un cuarto de vuelta a la derecha):

6. Configuración

6. Configuración Panel

18. Conf. Port RS232 isolated

3. Protocolo Integración

2400 bps

Versión: 2: 011 (solo si Half Duplex no está disponible)

Controles: habilitado/deshabilitado ***1**

Supervisión com.: deshabilitado

Supervisión tempo: 0

Resaltadas en negrita las selecciones obligatorias.

Contraseña del nivel 3 por defecto: 27835

***1 Si se deshabilitan los controles no se podrán dar órdenes a la central**

Zona 15 Alarma Módulo	115	015	ZA A="0"
Zona 16 Alarma Módulo	115	016	ZB A="0"
Fallo de comunicación con la central	300	000	Fc="1"
Fallo del comunicador*	350	000	Fo="1"
Fallo comunicación CRA 1	351	001	Fo="3"
Fallo comunicación CRA 2	352	001	Fo="3"
Entrada digital 1	000	000	
Entrada digital 2	000	000	

* Estos códigos solo estarán activos en caso de utilizar MÓDEM GSM.

En las centrales de la Serie ID se podrá seleccionar entre Transmisión por Zona o Punto.

Los Códigos Contact ID por defecto de los eventos de la central son:

Evento	Código	Descripción
1	110	ALARMA
2	000	ALARMA (ANULADO)
3	000	(PRUEBA) DE ALARMA
4	300	Avería de transmisión con equipo de lazo.
5	300	No responde/perdido
6	300	Error de Tipo de ID
7	300	Error en comando de verificación
8	300	AVERÍA en Sensor o Módulo cto.abierto
9	300	Efectos Transitorios
10	000	PREALARMA
11	000	---- reservado ampliaciones futuras ----
12	300	Avería Transmisión (anulado/en pruebas)
13	000	NO RESPONDE anulado/en pruebas
14	300	Error de Tipo de ID
15	300	Error en orden de respuesta lazo
16	000	Avería de Datos (anulado/en pruebas)
17	300	Efectos Transitorios
18	000	PREALARMA (anulado/en pruebas)
19	R110	fin/restablecimiento de ALARMA
20	R380	fin/restablecimiento de AVERÍA
21	R354	restablecimientos de comunicación
22	000	FALLO en Simulación de Test Alarma
23	R570	Equipo HABILITADO
24	570	Equipo ANULADO
25	000	Sensibilidad/Tipo de Equipo Cambiado
26	000	Cambiado Retardo de Alarma
27	000	Zona reasignada
28	300	Dirección duplicada
29	000	Nuevo equipo creado
30	000	Equipo Borrado
31	000	Descripción equipo cambiada
32	000	---- reservado ampliaciones futuras ----

33	300	Cortocircuito en cto.supervisión de Módulo
34	000	TEST ACTIVACIÓN Módulo de Salida
35	000	TEST DESACTIVACIÓN Módulo Salida
36	300	Dirección duplicada
37	000	Módulo AUXiliar ACTIVO
38	000	Módulo AUXiliar DESACTIVADO
39	000	'CDI-LMS' fault signal (VdS only)
40	000	Sensor VIEW reemplazado
42	000	'CDI-LMS' supervision fault (VdS only)
43-50	000	---- reservado ampliaciones futuras ----
Códigos de 51 a 129 No editables		
129	305	REARME DEL SISTEMA
130	000	TEST FINALIZADO
131	000	SILENCIAR SIRENAS
132	000	SILENCIAR ZUMBADOR INTERNO
133	000	TODOS LOS SENSORES VIEW del lazo cambiado
134	000	ERROR CLAVE DE ACCESO
135	000	INICIO TEST, ZONA
136	R570	HABILITAR TODA LA ZONA
137	570	ANULAR TODA LA ZONA
138	110	EVACUACIÓN
139	000	AJUSTADA HORA DEL SISTEMA
140	306	CONFIRMADO LOS CAMBIOS REALIZADO
141	306	CREADA LÍNEA MATRIZ DE CONTROL
142	306	BORRADA LÍNEA EN MATRIZ DE CONTROL
143	300	PROBABLE CORTE EN LAZO
144	000	CANCELADO LA EDICIÓN DE CAMBIOS
145	000	PRUEBA activación Sirena o Circuito de Relé n
146	850	REINICIO ALIMENTACIÓN/SINCRONIZAR
147	300	FALLO DE COMUNICACIONES
148	300	PÉRDIDA TOTAL DEL LAZO n
149	300	PÉRDIDA PARCIAL DEL LAZO n
150	300	LAZO FINAL 'B' EN AVERÍA, LAZO n
151	306	DESCRIPCIÓN DE CENTRAL EN RED CAMBIADA
152	306	CONFIGURACIÓN DE LA RED CAMBIADA
153	305	REARME GENERAL (Sólo red)
154	571	SILENCIO SIRENAS GENERAL (Sólo red)
155	422	SILENCIO INTERNO GENERAL (Sólo red)
156	306	RED ANULADA
157	601	REACTIVAR SIRENAS
158	000	INICIADO TEST ZONA REMOTO (No VdS)
159	571	SIRENAS ANULADAS
160	571	Anulada Salida Remota de Alarma
161	000	SILENCIO ZUMBADOR INTERNO (Sólo ID2000)
162	571	Anulada Salidas de Control
163	575	Anular sirena/Retardos de Investigación
164	306	Aumentado Retardo de Investigación
165	323	Salida Remota de Alarma Activada
166	R323	Habilitada Salidas de Control
167	000	Aumentado Retardo de Investigación Ampliado
168	R571	Habilitada Salida Remota de Alarma

169	R571	Habilitada Sirenas
170	000	Sirenas cambiadas a modo INMEDIATO
171	000	Sirenas cambiadas a modo RETARDADO
172	000	Sistema en modo DÍA
173	000	Sistema en modo NOCHE
174	000	Protección/Cubierta de Central quitada
175	000	Protección/Cubierta de Central cambiada
176	614	Test de Salida Remota de Alarma
177	571	Desactivada Llamada a Bomberos
178	350	Avería en Llamada a Bomberos
179	571	Anulados equipos Control de Incendio
180	R571	Habilitados equipos Control de Incendio
181	571	Anulado Sirena o Circuito de Relé
182	571	Anulado Relé de Alarma
183	324	Anulado Relé de Avería
184	R571	Habilitada Sirena o Circuito de Relé
185	R571	Habilitado Relé de Alarma
186	R571	Habilitado Relé de Avería
187	000	<i>'SST' devices Disabled (VdS only)</i>
188	000	<i>'SST' devices Enabled (VdS only)</i>
189	000	<i>Panel Expansion Cover removed (VdS only)</i>
190	000	<i>Panel Expansion Cover replaced (VdS only)</i>
191	000	<i>Zone n put into Test mode (VdS only)</i>
192	000	<i>Zone n Enabled (VdS only)</i>
193	000	<i>Zone n Disabled (VdS only)</i>
194	000	<i>Put Zone n into Test (VdS only)</i>
195	000	<i>Enable Zone n (VdS only)</i>
196	000	<i>Disable Zone n (VdS only)</i>
197	000	<i>Sounders Enabled in Zone n (VdS only)</i>
198	000	<i>Sounders Disabled in Zone n (VdS only)</i>
199	000	<i>Zone n Fault (VdS only)</i>
200	380	Zona n en Avería
201	300	Fallo Comunicaciones en Repetidor (Sólo ID ² net)
202-	000	TEST desactivar Sirena o Circuito de Relé
203-205	000	---- reservado ampliaciones futuras ----
Códigos de 206 a 245 No editables		
246	000	Petición a otras estaciones suspender mensajes no solicitados Ask other station to suspend uns
247	000	Petición a otras estaciones continuar con mensajes no solicitados
248-256	000	---- reservado ampliaciones futuras ----
257	304	Fallo en CPU de Tarjeta de lazo LIB
258-260	000	---- reservado ampliaciones futuras ----
261	300	Reinicio de CPU de Tarjeta de lazo LIB
262-264	000	---- reservado ampliaciones futuras ----
265	300	Cortocircuito en lazo tarjeta LIB
266-268	000	---- reservado ampliaciones futuras ----
269	300	Equipo con dirección 00 en lazo LIB
270-272	000	---- reservado ampliaciones futuras ----
273	300	No encontrada Tarjeta Lazo LIB
274-276	000	---- reservado ampliaciones futuras ----
277	300	Fallo en driver final de tarjeta de lazo LIB

278-280	000	---- reservado ampliaciones futuras ----
281	300	Señal degradada en lazo
282-284	000	---- reservado ampliaciones futuras ----
285	300	Error checksum ROM Tarjeta Lazo
286-287	000	---- reservado ampliaciones futuras ----
288	300	Fallo en conexión RS232
289	301	Fallo Alimentación de red/Alimentación
290	309	Fallo Fuente Alimentación/Cargador
291	302	Tensión Baja en Baterías
292	311	Fallo en Baterías
293-295	000	---- reservado ampliaciones futuras ----
296	300	Activado Watchdog principal de CPU
297	300	Error en Checksum EPROM de CPU
298	300	Error de Escritura en memoria E2PROM de CPU
299	300	Error en Checksum de memoria FLASH de CPU
300	300	Fallo en impresora
301	300	Error de Escritura en memoria FLASH de CPU
302	300	Fallo de Software
303	300	Fallo de hardware en tarjeta CPU/Display
304	300	Fallo de hardware en placa base/tarjeta expansión
305	300	Fallo en temporizador de Watchdog de CPU
306	322	Avería de cortocircuito en Circuito Sirena 2
307	321	Avería de cortocircuito en Circuito Sirena 1
308	322	Avería de circuito abierto en Circuito Sirena 2
309	321	Avería de circuito abierto en Circuito Sirena 1
310	322	Avería en Relé en Circuito Sirena 2
311	321	Avería en Relé en Circuito Sirena 1
312	310	Avería Derivación a Tierra
313	300	Avería de cortocircuito en Circuito Sirena 3
314	300	Avería de cortocircuito en Circuito Sirena 4
315	300	Avería de circuito abierto en Circuito Sirena 3
316	300	Avería de circuito abierto en Circuito Sirena 4
317	000	Avería en Relé en Circuito Sirena 3
318	000	Avería en Relé en Circuito Sirena 4
319	000	Teclado Central Bloqueado
320	000	Temporizador de Watchdog de CPU no habilitado
321	000	Reloj Ajustado posterior año 2099
322	300	Error Supervisión Clock de CPU
323	300	Instrucción de CPU incorrecta
324	300	Avería Salida Auxiliar 1
325	300	Avería Salida Auxiliar 2
326	300	Fallo Fuente Alim.: Activo Auxiliar
327	000	Config.requiere tarj.expansión LEDs
328	300	La Configuración Requiere Tarjeta RS485
329	300	La Configuración Requiere Tarjeta RS232
330	300	Tarjeta RS232/RS485/Impresora desplazada
331	302	Desconectada debido a baterías bajas
332	301	Avería Fuente Alimen.externa
333	301	Fallo doble alimentación FA Externa
334	302	Tensión baja en fuente alimentación externa
335	300	Fallo o falta tarjeta ELIB

339	300	Error al escribir en FLASH de ELIB
343	300	Error en envío datos a ELIB
347	000	Zone assignments incorrect (VdS only)
348	000	Reference no. assignments incorrect (VdS only)
349	300	Activado Watchdog de CPU (COP)
350	000	<i>Network Zone duplication (VdS only)</i>
351	300	Tarjeta LIB instalada incompatible (ELIB ID2000)
355	300	Tarjeta de Red ID ² net no encontrada o inoperativa
356	300	ID ² net: Fallo inicialización de Red
358	300	ID ² net: Fallo inicio de red – sin respuesta a orden de conexión a la red
360	300	ID ² net: Fallo inicio de red – error en orden de conexión a la tarjeta de red
361	300	ID ² net: Fallo en Rutina de Red
362	300	ID ² net: Fallo Conexión Canal 1
363	300	ID ² net: Fallo Conexión Canal 2
364	300	ID ² net: Error de Checksum en Flash
365	300	ID ² net: Tiempo máximo sobrecarga red

Notas el código 000 indica que está deshabilitado y no se transmite. La R indica que el evento es un rearme.

Los códigos a transmitir por las entradas digitales son configurables por el usuario.

2.8 Ziton Fire Panel Modbus Protocol

UTC Fire & Security

firmware
Group

2010-2 Fire Panel Modbus Protocol

Notice of Confidential Information

Information contained herein is Confidential, is the property of UTC Fire & Security, is to be used solely for the purpose provided, and is not to be disclosed to others without the written consent of UTC Fire & Security.

© 2015 UTCFS, Inc. All Rights Reserved

	Name	Role	Date
Prepared by:	Cesca Krassimirova	Firmware Engineer	17 Feb 2015
Reviewed by:	José Angel Ortega Gustavo Taus	Firmware Project Leader Firmware Team Leader	17 Feb 2015
Approved by:			



Version: 1.4
 Created By: Cesca Krassimirova
 Date: 29/04/2014
 Location: UTCFS BCN
 Last Modified By: Cesca Krassimirova
 Last Date Modified: 17/02/2015
 Number of Pages: 43
 File Name: 00-3207-206-0000-14 FW_2010-2_Modbus_Protocol

3 Register Mapping

- Modbus uses 16-bit registers.
- These registers represent a contiguous address mapping of 2010-2 fire panels, zones and devices status.

3.1 Global Register Map

3.1.1 2010-2 Modbus Zone/Point Mode

Start Address	End Address	Access	Use
0x0001	0x0001	W	Execute Reset.
0x0002	0x0002	W	Execute Panel Silence.
0x0003	0x0003	W	Set Sounders Start/Stop.
0x0004	0x0004	W	Set Sounders Delay On/Off.
0x0005	0x0005	W	Set Fire Protection Delay On/Off.
0x0006	0x0006	W	Execute Fire Protection Override Delay.
0x0007	0x0007	W	Set Fire Routing Delay On/Off.
0x0008	0x0008	W	Execute Fire Routing Override Delay.
0x1001	0x1002	R	2010-2 fire panels Global Status Flags (32 Nodes).
0x2001	0x2080	R	2010-2 fire panel Status Flags (32 Nodes).
0x3001	0x7000	R	2010-2 fire panel Zone Status (32 Nodes, 512 Zones).
0x7001	0xF000	R	2010-2 fire panel Device Status (32 Nodes, 4 Loops, 256 Devices).

3.1.2 2010-2 Modbus Zone Mode

Start Address	End Address	Access	Use
0x0001	0x0001	W	Execute Reset.
0x0002	0x0002	W	Execute Panel Silence.
0x0003	0x0003	W	Set Sounders Start/Stop.
0x0004	0x0004	W	Set Sounders Delay On/Off.
0x0005	0x0005	W	Set Fire Protection Delay On/Off.
0x0006	0x0006	W	Execute Fire Protection Override Delay.
0x0007	0x0007	W	Set Fire Routing Delay On/Off.
0x0008	0x0008	W	Execute Fire Routing Override Delay.
0x1001	0x1002	R	2010-2 fire panels Global Status Flags (128 Nodes).
0x2001	0x2200	R	2010-2 fire panel Status Flags (128 Nodes).
0x3001	0xB000	R	2010-2 fire panel Zone Status (128 Nodes, 512 Zones).



3.1.3 Heartbeat

Start Address	End Address	Access	Use
0xFFFF	0xFFFF	W	Heartbeat. It is used only to reset the inactivity time counter to prevent the TCP/IP connection from closing automatically.

4 2010-2 Fire Panel Read Holding Registers

4.1 Zone/Point Mode



4.1.1 2010-2 Fire Panels Global Status

- Consists of 2 holding register.

Start Address	End Address	Access	Use
0x1001	0x1001	R	GLOBAL_NODE_STATUS1 (1 to 32 Nodes)
0x1002	0x1002	R	GLOBAL_NODE_STATUS2 (1 to 32 Nodes)

- Each register is divided up into two bytes.
- These two registers represent the global status of 2010-2 fire panels of the network, as defined below:

4.1.1.1 Status 1

Upper Byte								Lower Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used		Sounders Status						Not used		Functional Conditions					

Upper Byte				Lower Byte			
Bit8	Sounders delay enabled.			Bit0	Alarm functional condition.		
Bit9	Sounders activation delay in progress.			Bit1	Fault functional condition.		
Bit10	Sounders outputs activated.			Bit2	Disable functional condition.		
Bit11	Sounders outputs silenced.			Bit3	Test functional condition.		
Bit12	Sounders disabled.			Bit4	Daynight mode (1=nightmode/0=daymode).		
Bit13	Sounders override time elapsed.			Bit5	MCP alarm.		
Bit14	Sounders on Test.			Bit6	Not used.		
Bit15	Not used.			Bit7	Not used.		

4.1.1.2 Status 2

Upper Byte								Lower Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used		Fire Protection Status						Not used		Fire Brigade Status					

Upper Byte				Lower Byte			
Bit8	Fire Protection: delay enabled.			Bit0	Fire Brigade: delay enabled.		
Bit9	Fire Protection: activation delay in progress.			Bit1	Fire Brigade: activation delay in progress.		
Bit10	Fire Protection: outputs activated.			Bit2	Fire Brigade: outputs activated.		
Bit11	Fire Protection: acknowledged.			Bit3	Fire Brigade: acknowledged.		
Bit12	Fire Protection: disabled.			Bit4	Fire Brigade: disabled.		
Bit13	Fire Protection: Test ON			Bit5	Fire Brigade: Test ON		



Bit14	Not used.	Bit6	Fire Brigade: extended activation delay in progress.
Bit15	Not used.	Bit7	Not used.

- **Global Status in Alarm:** if this bit is 1 then one or more 2010-2 fire panels are in alarm status.
- **Global Status in Fault:** if this bit is 1 then one or more 2010-2 fire panels are in fault status.
- **Global Status Disabled:** if this bit is 1 then one or more 2010-2 fire panels are disabled.
- **Global Status in Test:** if this bit is 1 then one or more 2010-2 fire panels are in test status.

4.1.2 2010-2 Fire Panel Status Flags

- Consists of 128 holding registers.

Start Address	End Address	Access	Use
0x2001	0x2080	R	NODE1_STATUS1 - NODE_32_STATUS4

- Each register is divided up into two bytes.
- 4 registers represent all status of one node (2010-2 fire panel), as defined below:

Address	Value	Address	Value	Address	Value	Address	Value	Address	Value
0x2001	NODE1_ST1	0x201D	NODE8_ST1	0x2039	NODE15_ST1	0x2055	NODE22_ST1	0x2071	NODE29_ST1
0x2002	NODE1_ST2	0x201E	NODE8_ST2	0x203A	NODE15_ST2	0x2056	NODE22_ST2	0x2072	NODE29_ST2
0x2003	NODE1_ST3	0x201F	NODE8_ST3	0x203B	NODE15_ST3	0x2057	NODE22_ST3	0x2073	NODE29_ST3
0x2004	NODE1_ST4	0x2020	NODE8_ST4	0x203C	NODE15_ST4	0x2058	NODE22_ST4	0x2074	NODE29_ST4
0x2005	NODE2_ST1	0x2021	NODE9_ST1	0x203D	NODE16_ST1	0x2059	NODE23_ST1	0x2075	NODE30_ST1
0x2006	NODE2_ST2	0x2022	NODE9_ST2	0x203E	NODE16_ST2	0x205A	NODE23_ST2	0x2076	NODE30_ST2
0x2007	NODE2_ST3	0x2023	NODE9_ST3	0x203F	NODE16_ST3	0x205B	NODE23_ST3	0x2077	NODE30_ST3
0x2008	NODE2_ST4	0x2024	NODE9_ST4	0x2040	NODE16_ST4	0x205C	NODE23_ST4	0x2078	NODE30_ST4
0x2009	NODE3_ST1	0x2025	NODE10_ST1	0x2041	NODE17_ST1	0x205D	NODE24_ST1	0x2079	NODE31_ST1
0x200A	NODE3_ST2	0x2026	NODE10_ST2	0x2042	NODE17_ST2	0x205E	NODE24_ST2	0x207A	NODE31_ST2
0x200B	NODE3_ST3	0x2027	NODE10_ST3	0x2043	NODE17_ST3	0x205F	NODE24_ST3	0x207B	NODE31_ST3
0x200C	NODE3_ST4	0x2028	NODE10_ST4	0x2044	NODE17_ST4	0x2060	NODE24_ST4	0x207C	NODE31_ST4
0x200D	NODE4_ST1	0x2029	NODE11_ST1	0x2045	NODE18_ST1	0x2061	NODE25_ST1	0x207D	NODE32_ST1
0x200E	NODE4_ST2	0x202A	NODE11_ST2	0x2046	NODE18_ST2	0x2062	NODE25_ST2	0x207E	NODE32_ST2
0x200F	NODE4_ST3	0x202B	NODE11_ST3	0x2047	NODE18_ST3	0x2063	NODE25_ST3	0x207F	NODE32_ST3
0x2010	NODE4_ST4	0x202C	NODE11_ST4	0x2048	NODE18_ST4	0x2064	NODE25_ST4	0x2080	NODE32_ST4



0x2011	NODE5_ST1	0x202D	NODE12_ST1	0x2049	NODE19_ST1	0x2065	NODE26_ST1		
0x2012	NODE5_ST2	0x202E	NODE12_ST2	0x204A	NODE19_ST2	0x2066	NODE26_ST2		
0x2013	NODE5_ST3	0x202F	NODE12_ST3	0x204B	NODE19_ST3	0x2067	NODE26_ST3		
0x2014	NODE5_ST4	0x2030	NODE12_ST4	0x204C	NODE19_ST4	0x2068	NODE26_ST4		
0x2015	NODE6_ST1	0x2031	NODE13_ST1	0x204D	NODE20_ST1	0x2069	NODE27_ST1		
0x2016	NODE6_ST2	0x2032	NODE13_ST2	0x204E	NODE20_ST2	0x206A	NODE27_ST2		
0x2017	NODE6_ST3	0x2033	NODE13_ST3	0x204F	NODE20_ST3	0x206B	NODE27_ST3		
0x2018	NODE6_ST4	0x2034	NODE13_ST4	0x2050	NODE20_ST4	0x206C	NODE27_ST4		
0x2019	NODE7_ST1	0x2035	NODE14_ST1	0x2051	NODE21_ST1	0x206D	NODE28_ST1		
0x201A	NODE7_ST2	0x2036	NODE14_ST2	0x2052	NODE21_ST2	0x206E	NODE28_ST2		
0x201B	NODE7_ST3	0x2037	NODE14_ST3	0x2053	NODE21_ST3	0x206F	NODE28_ST3		
0x201C	NODE7_ST4	0x2038	NODE14_ST4	0x2054	NODE21_ST4	0x2070	NODE28_ST4		

4.1.2.1 Status 1

Upper Byte								Lower Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used	Sounders Status							Not used		Functional Conditions					

Upper Byte				Lower Byte			
Bit8	Sounders delay enabled.			Bit0	Alarm functional condition.		
Bit9	Sounders activation delay in progress.			Bit1	Fault functional condition.		
Bit10	Sounders outputs activated.			Bit2	Disable functional condition.		
Bit11	Sounders outputs silenced.			Bit3	Test functional condition.		
Bit12	Sounders disabled.			Bit4	Daynight mode (1=nightmode/0=daymode).		
Bit13	Sounders override time elapsed.			Bit5	MCP alarm.		
Bit14	Sounders on Test.			Bit6	Not used.		
Bit15	Not used.			Bit7	Not used.		

4.1.2.2 Status 2

Upper Byte								Lower Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used	Fire Protection Status							Not used		Fire Brigade Status					

Upper Byte				Lower Byte			
Bit8	Fire Protection: delay enabled.			Bit0	Fire Brigade: delay enabled.		
Bit9	Fire Protection: activation delay in progress.			Bit1	Fire Brigade: activation delay in progress.		
Bit10	Fire Protection: outputs activated.			Bit2	Fire Brigade: outputs activated.		
Bit11	Fire Protection: acknowledged.			Bit3	Fire Brigade: acknowledged.		
Bit12	Fire Protection: disabled.			Bit4	Fire Brigade: disabled.		



Bit13	Fire Protection: Test ON	Bit5	Fire Brigade: Test ON
Bit14	Not used.	Bit6	Not used
Bit15	Not used.	Bit7	Not used.

4.1.2.3 Status 3

Upper Byte								Lower Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used								Not used							

4.1.2.4 Status 4

Upper Byte								Lower Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used								Not used							

4.1.3 2010-2 Fire Panel Zone Status

- Consists of 16384 holding registers.

Start Address	End Address	Access	Node Address	Use
0x3001	0x3200	R	NODE1	ZONE_1 STATUS ZONE_2 STATUS ZONE_3 STATUS ... ZONE_512 STATUS
0x3201	0x3400	R	NODE2	
0x3401	0x3600	R	NODE3	
0x3601	0x3800	R	NODE4	
0x3801	0x3A00	R	NODE5	
0x3A01	0x3C00	R	NODE6	
0x3C01	0x3E00	R	NODE7	
0x3E01	0x4001	R	NODE8	
0x4001	0x4200	R	NODE9	
0x4201	0x4400	R	NODE10	
0x4401	0x4600	R	NODE11	
0x4601	0x4800	R	NODE12	
0x4801	0x4A00	R	NODE13	
0x4A01	0x4C00	R	NODE14	
0x4C01	0x4E00	R	NODE15	
0x4E01	0x5000	R	NODE16	
0x5001	0x5200	R	NODE17	
0x5201	0x5400	R	NODE18	
0x5401	0x5600	R	NODE19	
0x5601	0x5800	R	NODE20	
0x5801	0x5A00	R	NODE21	
0x5A01	0x5C00	R	NODE22	
0x5C01	0x5E00	R	NODE23	



0x5E01	0x6000	R	NODE24	
0x6001	0x6200	R	NODE25	
0x6201	0x6400	R	NODE26	
0x6401	0x6600	R	NODE27	
0x6601	0x6800	R	NODE28	
0x6801	0x6A00	R	NODE29	
0x6A01	0x6C00	R	NODE30	
0x6C01	0x6E00	R	NODE31	
0x6E01	0x7000	R	NODE32	

- Each register is divided up into two bytes.
- The lower byte represents one zone status of one node (2010-2 fire panel) as defined below:

Upper Byte								Lower Byte								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Not used								Zone Disabled		Zone in Test		Zone in Fault		Zone in Alarm		Zone in Alert

- Zone in Alert:** if this bit is 1 then one or more fire detectors of the zone are in pre-alarm status.
- Zone in Alarm:** if this bit is 1 then one or more fire detectors of the zone are in alarm status.
- Zone in Fault:** if this bit is 1 then one or more devices of the zone are in fault status.
- Zone in Test:** if this bit is 1 then the zone is in test. When this bit is 0 the zone is in normal status.
- Zone Disabled:** the zone is disabled when this bit is 1. The zone is enabled when this bit is 0.

4.1.4 2010-2 Fire Panel Device Status

- Consists of 32768 holding registers.

Start Address	End Address	Access	Node Address	Use
0x7001	0x7100	R	NODE1	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7101	0x7200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7201	0x7300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7301	0x7400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x7401	0x7500	R	NODE2	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7501	0x7600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS



0x7601	0x7700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7701	0x7800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x7801	0x7900	R	NODE3	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7901	0x7A00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7A01	0x7B00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7B01	0x7C00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x7C01	0x7D00	R	NODE4	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x7D01	0x7E00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x7E01	0x7F00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x7F01	0x8000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8001	0x8100	R	NODE5	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8101	0x8200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8201	0x8300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8301	0x8400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8401	0x8500	R	NODE6	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8501	0x8600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8601	0x8700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8701	0x8800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8801	0x8900	R	NODE7	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8901	0x8A00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8A01	0x8B00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8B01	0x8C00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x8C01	0x8D00	R	NODE8	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x8D01	0x8E00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x8E01	0x8F00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x8F01	0x9000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9001	0x9100	R	NODE9	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9101	0x9200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9201	0x9300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9301	0x9400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9401	0x9500	R	NODE10	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9501	0x9600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9601	0x9700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9701	0x9800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9801	0x9900	R	NODE11	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9901	0x9A00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9A01	0x9B00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9B01	0x9C00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0x9C01	0x9D00	R	NODE12	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0x9D01	0x9E00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0x9E01	0x9F00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0x9F01	0xA000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xA001	0xA100	R	NODE13	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xA101	0xA200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xA201	0xA300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xA301	0xA400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xA401	0xA500	R	NODE14	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xA501	0xA600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xA601	0xA700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xA701	0xA800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS



0xA801	0xA900	R	NODE15	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xA901	0xAA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xAA01	0xAB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xAB01	0xAC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xAC01	0xAD00	R	NODE16	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xAD01	0xAE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xAE01	0xAF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xAF01	0xB000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xB001	0xB100	R	NODE17	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xB101	0xB200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xB201	0xB300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xB301	0xB400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xB401	0xB500	R	NODE18	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xB501	0xB600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xB601	0xB700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xB701	0xB800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xB801	0xB900	R	NODE19	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xB901	0xBA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xBA01	0xBB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xBB01	0xBC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xBC01	0xBD00	R	NODE20	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xBD01	0xBE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xBE01	0xBF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xBF01	0xC000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xC001	0xC100	R	NODE21	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xC101	0xC200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xC201	0xC300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xC301	0xC400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xC401	0xC500	R	NODE22	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xC501	0xC600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xC601	0xC700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xC701	0xC800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xC801	0xC900	R	NODE23	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xC901	0xCA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xCA01	0xCB00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xCB01	0xCC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xCC01	0xCD00	R	NODE24	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xCD01	0xCE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xCE01	0xCF00	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xCF01	0xD000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xD000	0xD100	R	NODE25	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xD101	0xD200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xD201	0xD300	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xD301	0xD400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xD401	0xD500	R	NODE26	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xD501	0xD600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xD601	0xD700	R		LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xD701	0xD800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xD801	0xD900	R	NODE27	LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xD901	0xDA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS



0xDA01	0xDB00	R	NODE28	LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xDB01	0xDC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xDC01	0xDD00	R		LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xDD01	0xDE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xDE01	0xDF00	R	NODE29	LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xDF01	0xE000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xE001	0xE100	R		LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xE101	0xE200	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xE201	0xE300	R	NODE30	LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xE301	0xE400	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xE401	0xE500	R		LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xE501	0xE600	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xE601	0xE700	R	NODE31	LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xE701	0xE800	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xE801	0xE900	R		LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xE901	0xEA00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xEA01	0xEB00	R	NODE32	LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xEB01	0xEC00	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS
0xEC01	0xED00	R		LOOP_1_DEV_1 - LOOP_1_DEV_256 STATUS
0xED01	0xEE00	R		LOOP_2_DEV_1 - LOOP_2_DEV_256 STATUS
0xEE01	0xEF00	R	NODE32	LOOP_3_DEV_1 - LOOP_3_DEV_256 STATUS
0xEF01	0xF000	R		LOOP_4_DEV_1 - LOOP_4_DEV_256 STATUS

- Each register is divided up into two bytes.
- The lower byte represents one device status as defined below:

Upper Byte										Lower Byte					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not used										Device Disabled	Device in Test	Device in Fault	Device in Alarm	Device in Pre-Alarm	

- **Device in Pre-Alarm:** if this bit is 1 then the device is in pre-alarm status.
- **Device in Alarm:** if this bit is 1 then the device is in in alarm status.
- **Device in Fault:** if this bit is 1 then the device is in fault status.
- **Device in Test:** if this bit is 1 then the device is in test. When this bit is 0 the device is in normal status.
- **Device Disabled:** if this bit is 1 then the device is disabled. When this bit is 0 the device is enabled.



2.9 PCS7: Configuración de Hardware y Redes

• Hardware

ESBPCS7AS32

UR - Rack (0)

Short description: UR
 Order no: 6ES7 390-1???0-0AA0
 Designation: AS32UR

Rack (0), Slot 1

Short description: PS 307 5A
 Order no: 6ES7 307-1EA01-0AA0
 Designation: AS32PWS
 Width: 1

Rack (0), Slot 2

Short description: CPU 317-2 PN/DP
 Firmware version: V3.2
 Order no: 6ES7 317-2EK14-0AB0
 Designation: AS32CPU2
 Width: 1

Rack (0), Slot 2, Interface X1

Short description: MPI/DP
 Order no: - - -
 Designation: AS32CPU2X1
 Width: 1
 PROFIBUS address: 1
 Addresses
 Inputs
 Start: 8191
 End: 8191
 Synchronization mode: None
 Time interval: None
 Operating mode: DP Master

Rack (0), Slot 2, Interface X2

Short description: PN-IO
 Device name: AS32CPU2X2
 Support device replacement without exchangeable medium: yes
 IP address: 172.16.7.69
 Diagnostic addresses
 Interface: 8190
 Controller: 8187
 Send clock: 1.000ms
 Communication (PROFINET IO): 0.0 %
 Max. IRT stations in line: 0
 Use system settings: activated
 Use module for PROFINET&CBA communication: disabled
 Communication allocation (PROFINET CBA): 100%
 Possible QoS with cyclic interconnections: 10-200ms
 OB82 call after communication interrupt: activated
 Force IEC V2.2 LLDP mode: disabled
 &I-device mode: no
 Name of sync domain: syncdomain-default
 Synchronization role: Not synchronized
 RT class: RT; IRT
 IRT option: - - -
 Media Redundancy Protocol (MRP)
 Name of the MRP domain: mrpdomain-1
 Role in the domain: Not node in the ring
 Send keepalive for connections Interval: 30s

Rack 0, Slot 2, Interface X2 P1 R

Short description: Port
 Device name: Port 1
 Diagnostic address: 8189*
 Interconnection
 Local port: ESBPCS7AS32\AS32CPU2X2\Port 1 (R0/S2/X2 P1 R)
 Partner port: Any partner
 Cable
 Medium: Copper
 Cable name: Copper
 Cable length: < 100m (Signal delay time: 0.60µs)
 Delay time: 0.60µs
 Connection
 Transmission parameters: Automatic settings (monitor)
 Monitor: yes
 Disable autonegotiation: no
 Boundaries

```

End of sync domain: no
End of detection of accessible nodes: no
End of topology discovery: no
Rack 0, Slot 2, Interface X2 P2 R
Short description: Port
Device name: Port 2
Diagnostic address: 8188*
Interconnection
Local port: ESBEPCS7AS32\AS32CPU2X2\Port 1 (R0/S2/X2 P2 R)
Partner port: Any partner
Cable
Medium: Copper
Cable name: Copper
Cable length: Not specified
Delay time: 0.00µs
Connection
Transmission parameters: Automatic settings (monitor)
Monitor: no
Disable autonegotiation: no
Boundaries
End of sync domain: no
End of detection of accessible nodes: no
End of topology discovery: no
Rack (0), Slot 4
Short description: CP 343-1
Order no: 6GK7 343-1EX30-0XE0
Name: AS32ETH4
Firmware version: V3.0
Location
Station: ESBEPCS7AS32
Width: 1
MPI address: - - -
Name of MPI network: - - -
Addresses
Inputs
Start: 256
End: 271
System default: No
Outputs
Start: 256
End: 271
System default: No
Security
Security enabled: - - -
Access over FTPS: - - -
Access over HTTPS: - - -
Expanded NTP configuration: - - -
Rack (0), Slot 4 Interface X1
Short description: PN-IO
Device name: AS32ETH4X1
Support device replacement
without exchangeable medium: yes
IP address: 172.16.7.70
Diagnostic addresses
Interface: 1023
Controller: -
IP Configuration
Address assignment mode: Properties-Ethernet Interface dialog
Network
Network type: Ind. Ethernet
Network name: AS_PlantBus
MAC address: 08-00-06-01-00-00
IP address: 172.16.7.70
Subnet mask: 255.255.192.0
Router address: 172.16.0.1
Standard router: yes
IO device mode
Enable IO device mode: No
Enable parallel operation as IRT controller: yes
Send clock: 1.000ms
Communication allocation (PROFINET IO): 0.0%
Max. IRT stations in line: 0
Use system settings: activated
Use module for PROFINET&CBA communication: disabled
Communication allocation (PROFINET CBA): 100%
Possible QoS with cyclic interconnections: 10-200ms
OB82 call after communication interrupt: disabled
Force IEC V2.2 LLDP mode: disabled
&I-device mode: no
Name of sync domain: syncdomain-default

```

```

Synchronization role: Not synchronized
RT class: RT
IRT option: - - -
Media Redundancy Protocol (MRP)
Name of the MRP domain: mrpdomain-1
Role in the domain: Not node in the ring
Send keepalive for connections Interval: 0s
Rack 0, Slot 4, Interface X1 P1 R
Short description: Port
Device name: Port 1
Diagnostic address: 1022*
Interconnection
Local port: ESBEP7AS32\AS32ETH4X1\Port 1 (R0/S4/X1 P1 R)
Partner port: Any partner
Cable
Medium: Copper
Cable name: Copper
Cable length: Not specified
Delay time: 0.00µs
Connection
Transmission parameters: Automatic settings (monitor)
Monitor: yes
Disable autonegotiation: no
Boundaries
End of sync domain: no
End of detection of accessible nodes: no
End of topology discovery: no
Rack 0, Slot 4, Interface X1 P2 R
Short description: Port
Device name: Port 2
Diagnostic address: 1021*
Interconnection
Local port: ESBEP7AS32\AS32ETH4X1\Port 2 (R0/S4/X1 P2 R)
Partner port: Any partner
Cable
Medium: Copper
Cable name: Copper
Cable length: Not specified
Delay time: 0.00µs
Connection
Transmission parameters: Automatic settings (monitor)
Monitor: no
Disable autonegotiation: no
Boundaries
End of sync domain: no
End of detection of accessible nodes: no
End of topology discovery: no
PROFINET IO System
Short description: PROFINET IO system
Name: PROFINET IO system
IO system No.: 100
Subnet: AS_PlantBus
Communication Allocation (PROFINET IO): 0.0%
Send clock: 1.000ms
Address overview
Addresses of: AS32CPU2
Address range from: 0
Address range to: 8191
Available address assignment: Yes
Rack/Slot: 0/2


| Module     | IF | addr.from | Addr.to | Type | PIP  | DP | PN | FF | R | S |
|------------|----|-----------|---------|------|------|----|----|----|---|---|
| AS32ETH4   | -  | 256       | 271     | I    | ---- | -  | -  | -  | 0 | 4 |
| AS32CPU2X2 | 2  | 8187      | 8187    | I*   | -    | -  | -  | -  | 0 | 2 |
| Port 2     | 2  | 8188      | 8188    | I*   | -    | -  | -  | -  | 0 | 2 |
| Port 1     | 4  | 8189      | 8189    | I*   | -    | -  | -  | -  | 0 | 2 |
| AS32CPU2X2 | 3  | 8190      | 8190    | I*   | -    | -  | -  | -  | 0 | 2 |
| AS32CPU2X1 | 2  | 8191      | 8191    | I*   | -    | -  | -  | -  | 0 | 2 |
| AS32ETH4   | 1  | 256       | 271     | Q    | ---- | -  | -  | -  | 0 | 4 |

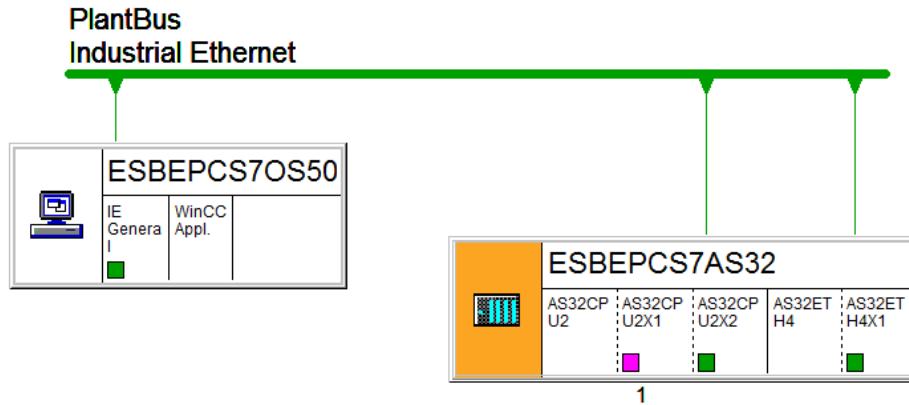

Addresses of: AS32ETH4
Address range from: 0
Address range to: 1023

```

Available address assignment: Yes
Rack/Slot: 0/4

Module	addr. from	Addr. to	Type	PIP	DP	PN	FF	R	S
IF									
AS32ETH4	256	271	I	----	-	-	-	0	4
-									
Port 2	1021	1021	I*	-	-	-	-	0	4
3									
Port 1	1022	1022	I*	-	-	-	-	0	4
2									
AS32ETH4X1	1023	1023	I*	-	-	-	-	0	4
1									
AS32ETH4	256	271	Q	----	-	-	-	0	4
-									

• **Redes**



Industrial Ethernet: PlantBus (S7 subnet ID: 0253 - 0003) contains the following network connections:

Station address:	Station:	Module:	R/S:
172.16.7.69	ESBEP7AS32	AS32CPU2	0/2
172.16.7.70	ESBEP7AS32	AS32ETH4	0/4
172.16.12.36	ESBEP70S50	IE General	-

Projects and cross-project subnets of the multiproject: FP_MP (D:\FP_MP\FP_MP)

FP_OS (D:\FP_MP\FP_OS)

The project contains the following cross-project subnets:

OS_PlantBus (Part of: PlantBus) (S7 subnet ID: 0253 - 0003)

FP_AS (D:\FP_MP\FP_AS)

The project contains the following cross-project subnets:

AS_PlantBus (Part of: PlantBus) (S7 subnet ID: 0253 - 0003)

Connection(s) from the local module:

ESBEP7AS32 / AS32CPU2

Connection 1

Local ID:	(HEX) 1
Partner ID:	PlantBus_CON
Local end point:	ESBEP7AS32/AS32CPU2
Partner end point:	FP_OS/ESBEP70S50/WinCC Appl.
Connection type:	S7 connection

Type-specific parameters:

Configured dynamic connection:	No
Connection establishment initiated at one end:	No
Active connection setup	No
Send operating mode messages	No

Connection path:

By means of local CP/CPU: AS32CPU2(R0/S2)

By means of remote CP/CPU:	IE General
Subnet (local):	AS_PlantBus [Industrial Ethernet]
Subnet (remote):	OS_PlantBus [Industrial Ethernet]
Station address (local):	172.16.7.70
Station address (remote):	172.16.12.36
Address details:	
Rack/Slot (local):	0/2
Connection resource (local)(hex):	10
Connection resource (remote)(hex):	10
TSAP (local):	10.02
TSAP (remote):	10.12
S7 subnet ID (local):	0253 - 0003
S7 subnet ID (remote):	0253 - 0003

2.10 PCS7: Jerarquía de Planta

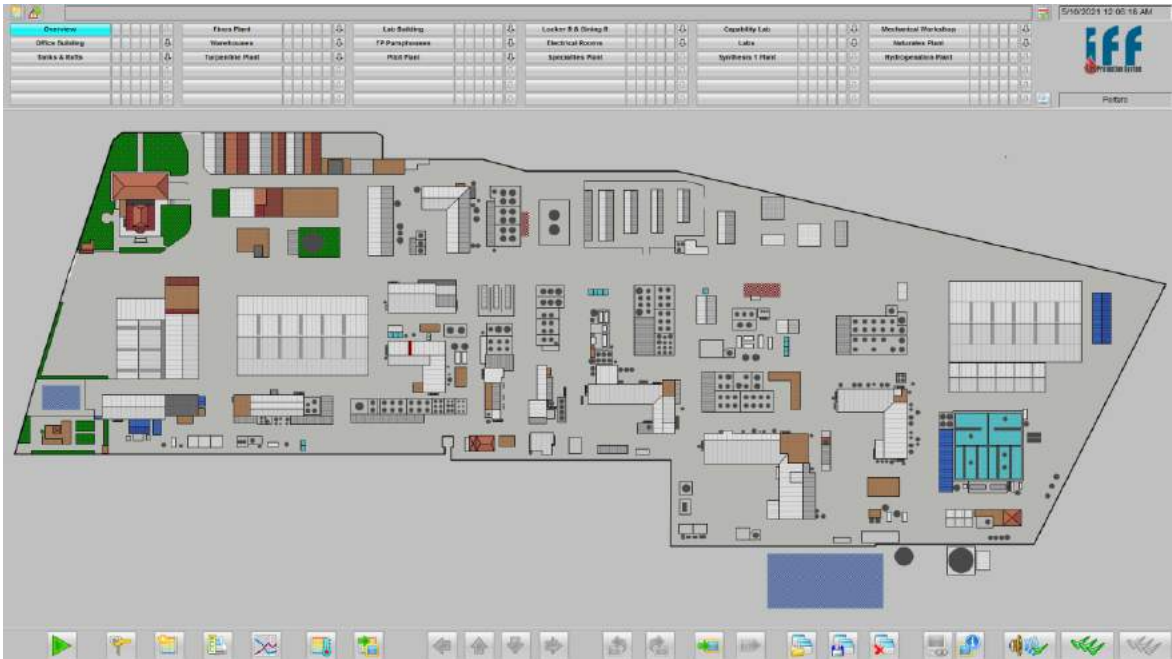
Proyecto	Nivel 1	Nivel 2	Nivel 3	
	Capability Lab	First Floor	HPT Reactor Room	
			Lab A	
			Lab B	
			LPT Reactor Room	
			Multi-Reactor Room	
			R&D PTR Room	
			Rack	
		Ground Floor	Corridor	
			Cosmetics	
			Female Toilet	
			Hall	
			Male Toilet	
			Meeting Room	
			Team Room	
		Electrical Rooms	CT-1	
			MCC Acetates	
			MCC Finos	
			MCC Naturales	
			MCC Pilot Plant	
			MCC Synthesis 1	
			MCC Turpentine	
		Finos Plant	Control Room	
			Production	
		FP Pumphouses	FP Pumphouse 1	
		Hydrogenation Plant	Control Room	
			Production	
		Lab Building	First Floor	Batches Archive
				Brands Lab
				Chromatography
				Contin Improvement
				Control Lab 1
				Control Lab 2
				EHS IT Room
			EHS Manager	
			EHS Office	
			Encapsul Corridor	
			Encapsul Manager	
			Encapsul Meeting R 1	
			Encapsul Meeting R 2	
			Encapsul Office 1	
			Encapsul Office 2	
			Encapsul Office 3	
			Encapsul Tech Office	
			Environment Lab	
			Main Corridor	
			Meeting R DAKSA	
			Meeting Room	
			Panel	
			Planning Manager	
			Planning Office	
			Plant Supp Manager	
			Plant Supp Office	
			Quality Office	
			Servers Room	
			Tag Printers	

Proyecto	Nivel 1	Nivel 2	Nivel 3
		Ground Floor	Analytical Lab 1 Analytical Lab 2 Chromatography Hall Instrument Room Lab 4 Lab 4 Corridor Lab 4 Office Lab 4 Tech Office Main Hall Material Room Performance Room Process Lab 1 Process Lab 2 Process Office Raw Material RMN Store
	Labs	Hydrogenation Lab Lab 3 Naturales Lab	
	Locker R & Dining R	First Floor Ground Floor	Corridor Locker Room 1 Locker Room 2 Toilets & Showers Dining Room Dining Room Toilets Female Locker Room Locker Rooms Hall Male Locker Room
	Mechanical Workshop	First Floor Ground Floor	
	Naturales Plant	Control Room Production	
	Office Building	Basement First Floor	EPIs Store Material Store Paper Crusher Parking Parking Entry Store 1 Store 2 Archive AZUL Meeting Room Business Meeting R Business Mngr Offi 1 Business Mngr Offi 2 Business Mngr Offi 3 Engineer Meeting R Engineering Manager Engineering Office Finance Manager Finance Office Hall IT Office Manager Assistant PICASSO Meeting R SDS Manager SDS Store Servers Room

Proyecto	Nivel 1	Nivel 2	Nivel 3
		Ground Floor	
			DALI Meeting Room
			HR Manager
			HR Office
			Main Hall
			MIRO Room
			Purchasing M Store
			Purchasing Manager
			Purchasing Office 1
			Purchasing Office 2
			Purchasing Office 3
			Sales Manager
			Sales Meeting Room
			Sales Office
			SOROLLA Room
			Store 1
			Store 2
			Store 3
			Store 4
			Works Council Room
		Second Floor	
			Finance Archive 1
			Finance Archive 2
			Hall
	Overview		
	Pilot Plant		
		Control Room	
		Production	
	Specialties Plant		
		Control Room	
		Production	
	Synthesis 1 Plant		
		Control Room	
		Production	
	Tanks & Rafts		
		Acetaldehyde	
		Boilers Tanks	
		Emergency Raft	
		Hydrogen Platform	
		Isoprene Tank	
		Turpentine Tanks	
	Turpentine Plant		
		Control Room	
		Production	
	Warehouses		
		Dispatch WH	
		Warehouse L1	
		Warehouse L2 & L3	
		Warehouse L4	
		Warehouse M1	
		Warehouse M2	
		Warehouse M3	

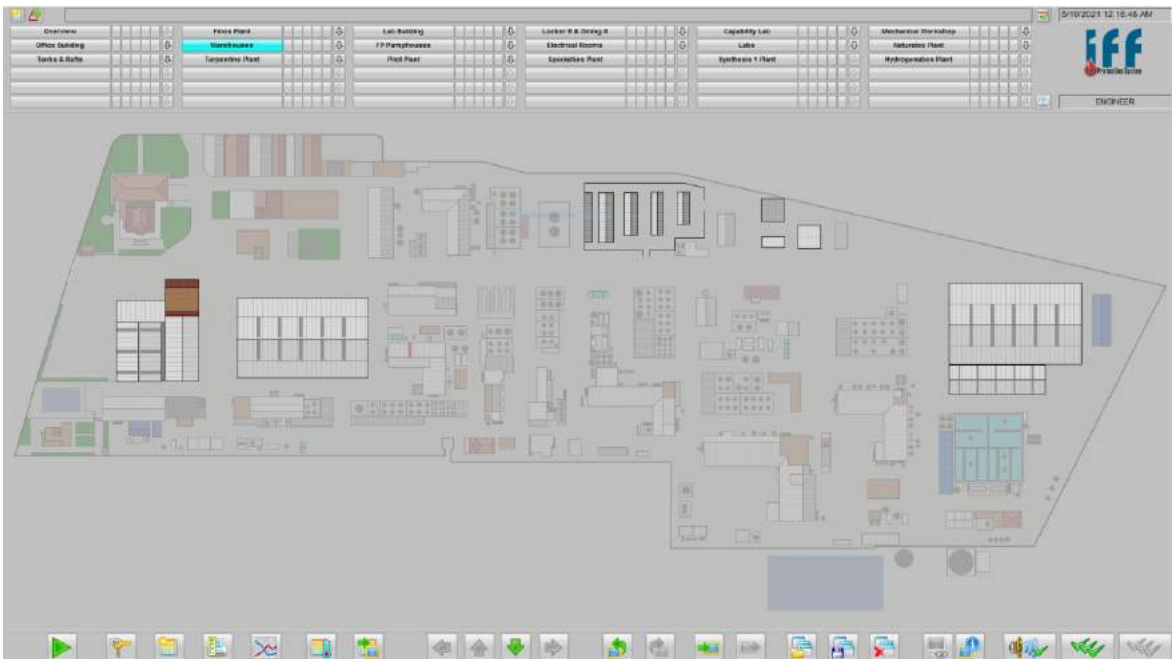
2.11 PCS7: Jerarquía en Pantallas del Sistema

Nivel 1: Vista general

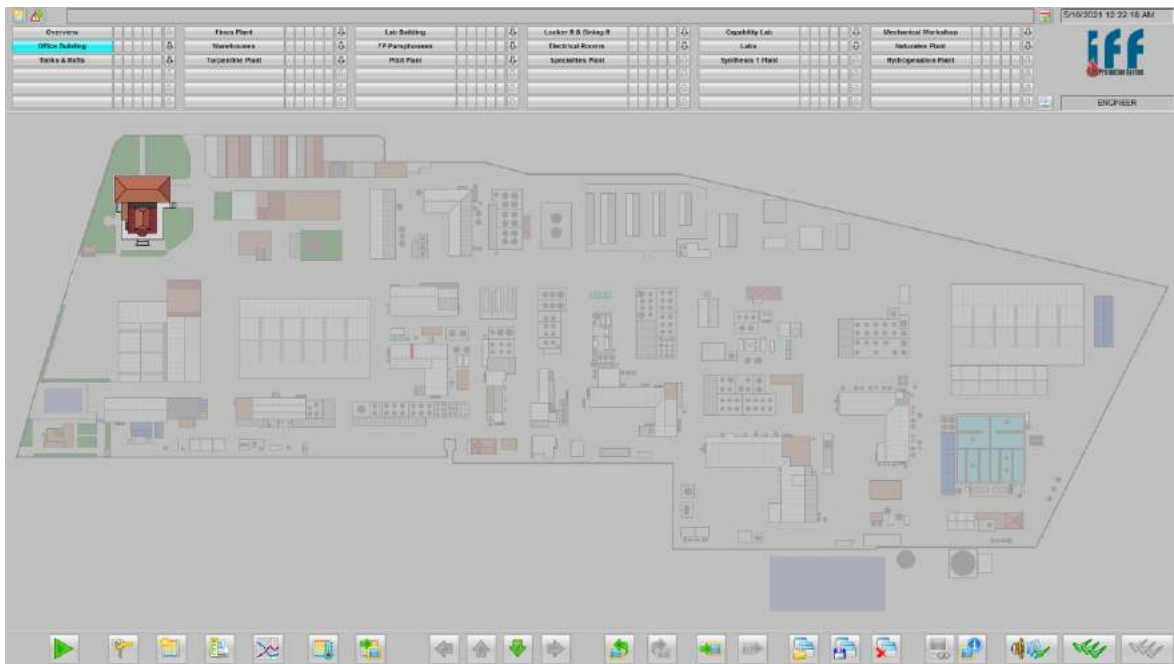


- La vista general, aunque pertenezca al Nivel 1, muestra todas las instalaciones

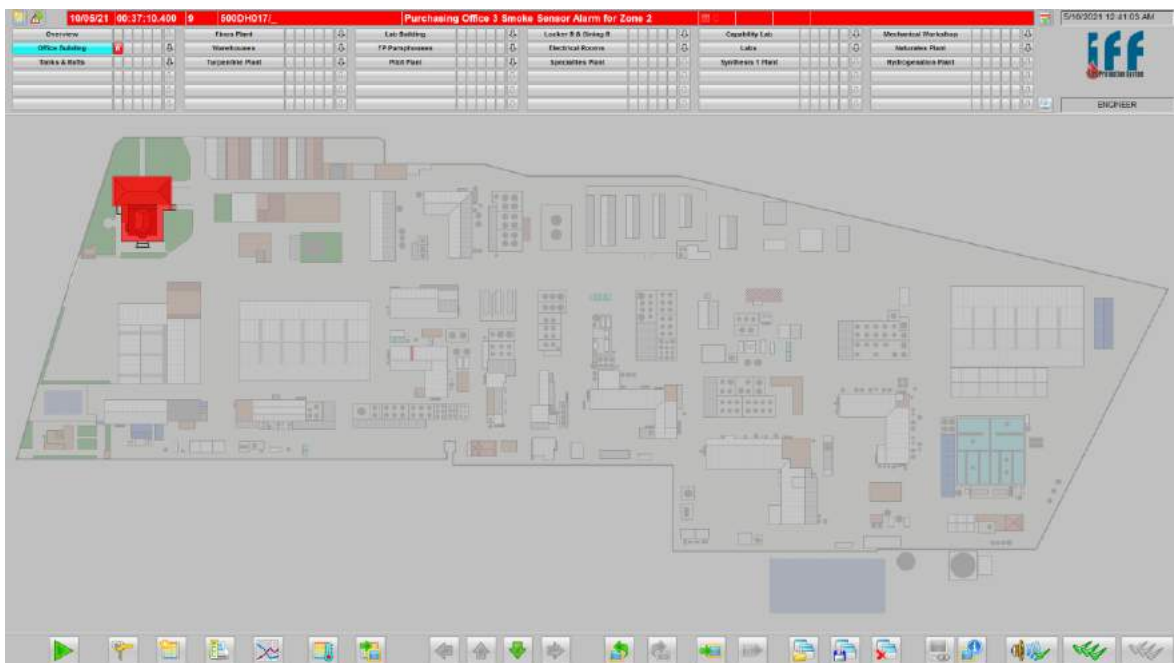
Nivel 1: Almacenes (Warehouses)



- Vista de las instalaciones del tipo almacenes. El menú principal muestra también el área seleccionada.

Nivel 1: Edificio de Oficinas (Office Building)

- Vista del edificio de oficinas. El menú principal muestra también el área seleccionada.

Nivel 1: Edificio de Oficinas (Office Building) en Alarma

- Vista del edificio de oficinas en alarma (rojo intermitente).
- El menú principal muestra también el estado de alarma para esa área (“A” roja recuadrada).
- La parte superior de la pantalla muestra la última alarma acontecida con mayor prioridad.

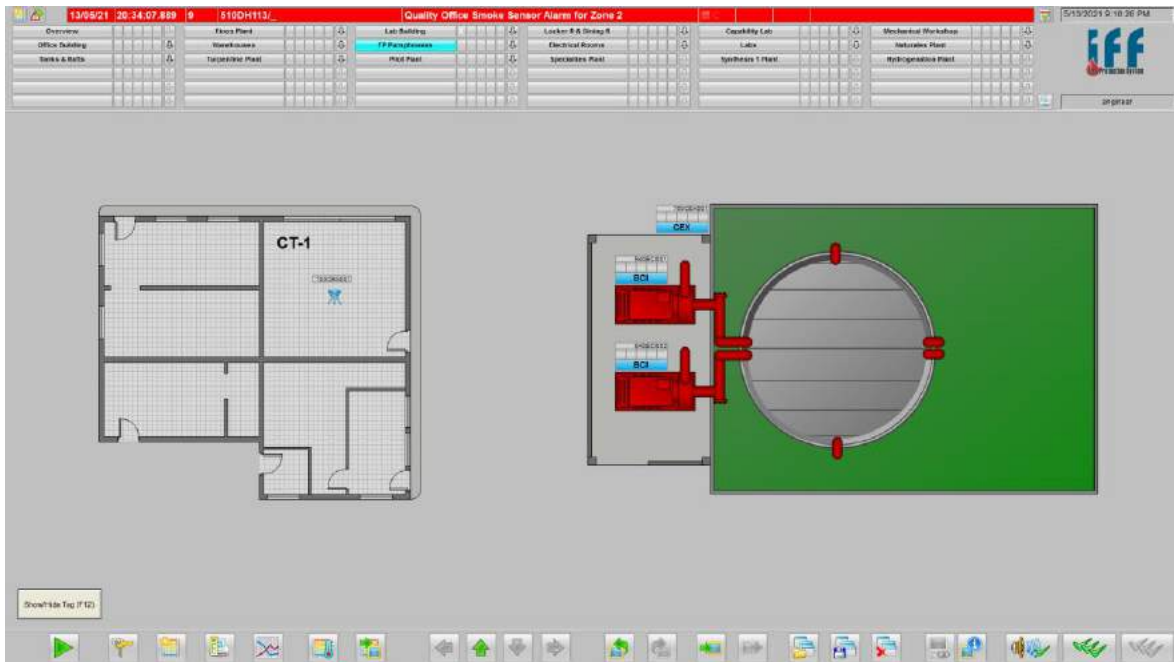
Nivel 2: Edificio de Oficinas-Piso 1 (Office Building) en PreAlarma



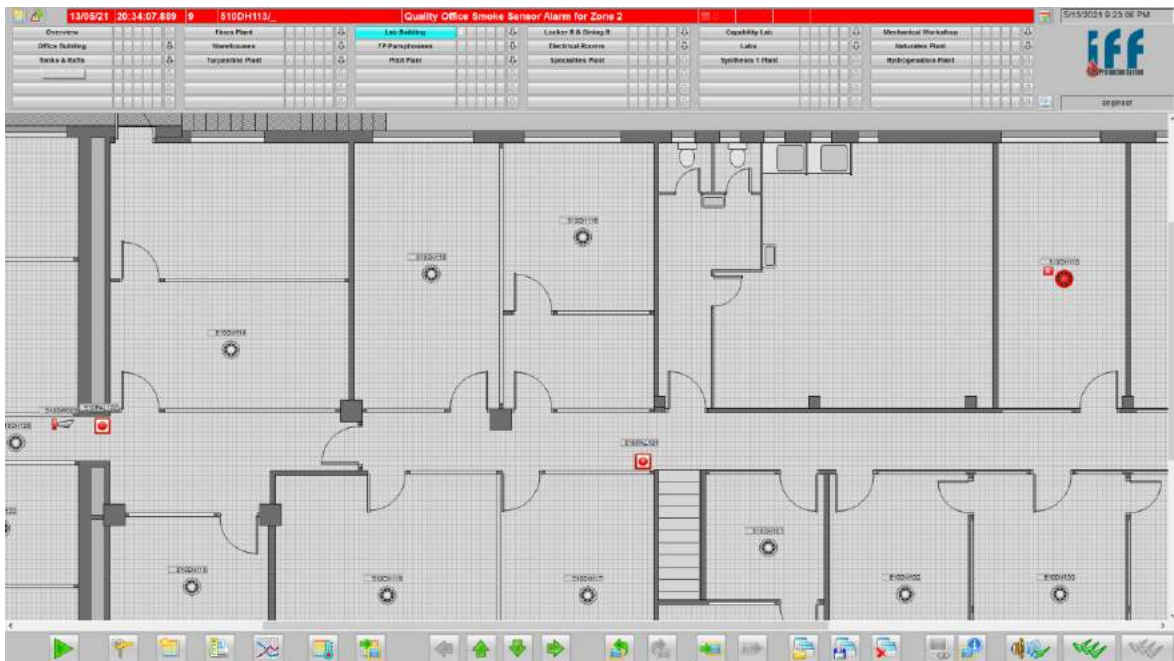
- Vista del 1º piso del edificio de oficinas con el Dpto. de Ingeniería en PreAlarma (amarillo intermitente).
- El menú principal muestra el estado de PreAlarma (Warning) para esa área/piso/estancia (“W” amarilla recuadrada).
- Pulsando sobre la flecha del menú principal se accede al submenú del nivel 2, y desplegando uno de estos sub-menús se puede ver el listado de las estancias. Esto se puede hacer independientemente de la pantalla en la que nos encontremos.

Nivel 2: Edificio de Laboratorios-Piso 1 (Lab Building) en Alarma



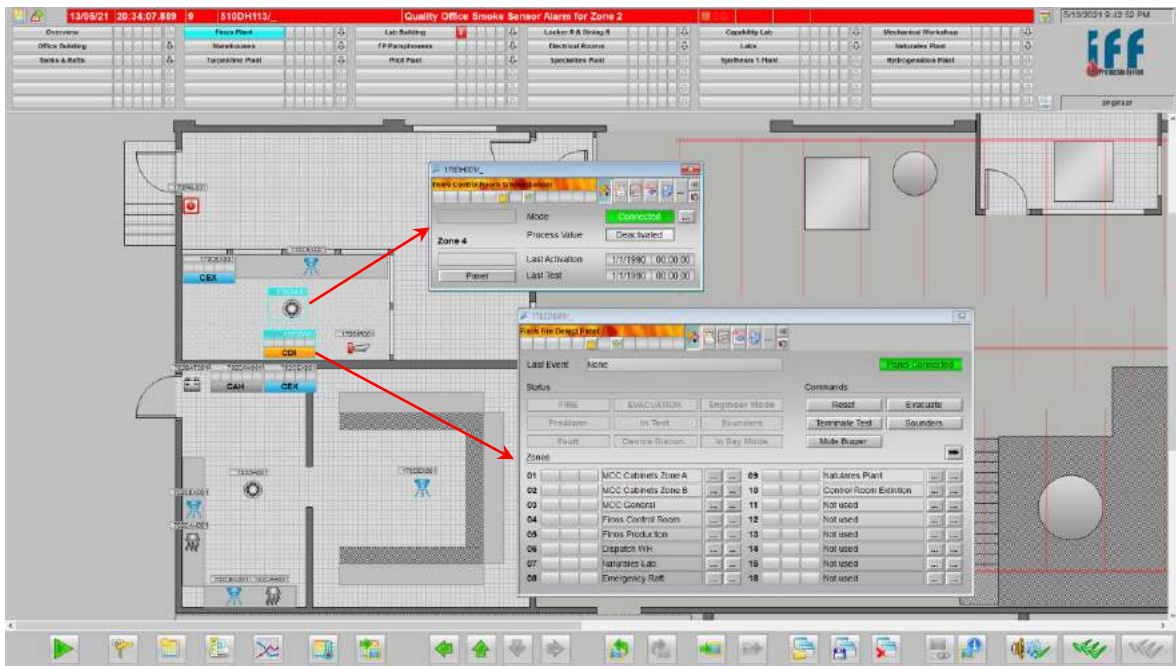
Nivel 2: Instalación Cuarto de Bombas Contraincendios.

- Esta instalación tiene una extensión pequeña y con el Nivel 2 es suficiente. En este nivel se incluyen *block icons* representando los dispositivos. Central de extinción, zona de extinción y bombas.

Nivel 3: Edificio de Laboratorios-Piso 1. Sensor de humo del Dpto. Calidad en Alarma

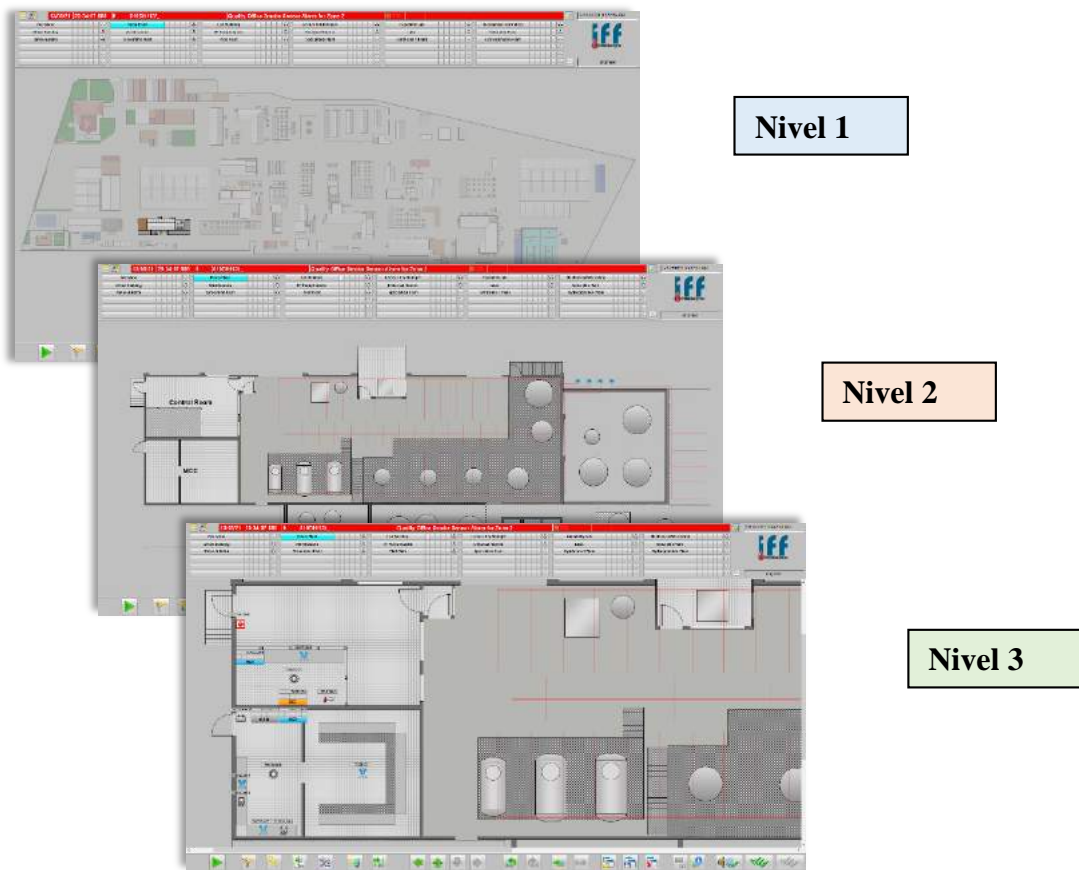
- El Nivel 3 es una vista ampliada del 2 cuando se pulsa sobre la pantalla. En este nivel se incluyen *Block Icons* representando los dispositivos y su ubicación real. Detectores de Humo, Pulsadores de Alarma, Sirenas, etc.
- Es posible desplazarse por esta vista con las barras de desplazamiento lateral e inferior.
- Los nombres de los dispositivos (Tags) se pueden mostrar u ocultar con la tecla F12.

Nivel 3: Planta de Finos-Sala de Control y Cuarto Eléctrico (MCC).



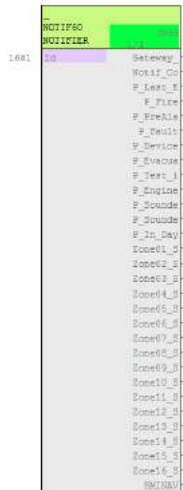
- Pulsando sobre el *Block Icon* de un elemento, se accede a su *Faceplate*, donde es posible gestionarlo de forma individual.

Visión General de Navegación por la Jerarquía para la Planta de Finos



2.12 FB550: NOTIF60

• Representación Gráfica



• Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
FBID	Word	0.0	W#16#0	Block ID =0 (DO NOT CHANGE)
Id	Word	2.0	W#16#1	Connection ID (W#16#01..W#16#0FFF)
Dev_ID	Byte	4.0	B#16#2	Communication interface (See TCON Block Help)
IP_Address	Struct	6.0		Master IP Address
Octet_1	Int	6.0	192	1th Octet (0..255)
Octet_2	Int	8.0	168	2nd Octet (0..255)
Octet_3	Int	10.0	0	3th Octet (0..255)
Octet_4	Int	12.0	1	4th Octet (0..255)
Port	Int	14.0	502	Master Port
Gtkw_Conn_Time	Real	16.0	5.000000e+000	Delay time Gateway connection Error [s]
Notif_Conn_Time	Real	20.0	5.000000e+000	Delay time NOTIFIER connection Error [s]
Max_CMD_Time	Real	24.0	5.000000e+000	Max Time for Sending Commands [s]
EnaComOp	Bool	28.0	FALSE	Operator Enable comms
SysRstOp	Bool	28.1	FALSE	Operator System Reset
TermTestOp	Bool	28.2	FALSE	Operator Terminate Test
SilSoundOp	Bool	28.3	FALSE	Operator Silence Sounder
MuteBuzzOp	Bool	28.4	FALSE	Operator Mute Buzzer
EvacuateOp	Bool	28.5	FALSE	Operator Evacuate
TestSoundOp	Bool	28.6	FALSE	Operator Test Sounder
ResoundOp	Bool	28.7	FALSE	Operator Resound Sounder
Zone01_CMD	Struct	30.0		Zone 01 CMD
EnableOp	Bool	30.0	FALSE	Enable Zone
DisableOp	Bool	30.1	FALSE	Disable Zone
StartTestOp	Bool	30.2	FALSE	Start Zone Test
StopTestOp	Bool	30.3	FALSE	Stop Zone Test
...
Zone32_CMD	Struct	92.0		Zone 32 CMD
EnableOp	Bool	92.0	FALSE	Enable Zone
DisableOp	Bool	92.1	FALSE	Disable Zone
StartTestOp	Bool	92.2	FALSE	Start Zone Test
StopTestOp	Bool	92.3	FALSE	Stop Zone Test
OS_Perm	Struct	94.0		Operator Permissions
Bit0	Bool	94.0	TRUE	1 = Operator can Reset system
Bit1	Bool	94.1	TRUE	1 = Operator can Terminate Test
Bit2	Bool	94.2	TRUE	1 = Operator can Silence Sounders
Bit3	Bool	94.3	TRUE	1 = Operator can Mute Buzzer
Bit4	Bool	94.4	TRUE	1 = Operator can Evacuate
Bit5	Bool	94.5	TRUE	1 = Operator can Test Sounders
Bit6	Bool	94.6	TRUE	1 = Operator can Resound Sounders
Bit7	Bool	94.7	TRUE	Reserved
Bit8	Bool	95.0	TRUE	1 = Operator can Enable Zone
Bit9	Bool	95.1	TRUE	1 = Operator can Disable Zone
Bit10	Bool	95.2	TRUE	1 = Operator can Start Zone Test
Bit11	Bool	95.3	TRUE	1 = Operator can Stop Zone Test
Bit12	Bool	95.4	TRUE	Reserved
Bit13	Bool	95.5	TRUE	Reserved
Bit14	Bool	95.6	TRUE	Reserved

Name	Data Type	Address	Initial Value	Comment
Bit15	Bool	95.7	TRUE	Reserved
Bit16	Bool	96.0	TRUE	Reserved
Bit17	Bool	96.1	TRUE	Reserved
Bit18	Bool	96.2	TRUE	Reserved
Bit19	Bool	96.3	TRUE	Reserved
Bit20	Bool	96.4	TRUE	Reserved
Bit21	Bool	96.5	TRUE	Reserved
Bit22	Bool	96.6	TRUE	Reserved
Bit23	Bool	96.7	TRUE	Reserved
Bit24	Bool	97.0	TRUE	Reserved
Bit25	Bool	97.1	TRUE	Reserved
Bit26	Bool	97.2	TRUE	Reserved
Bit27	Bool	97.3	TRUE	Reserved
Bit28	Bool	97.4	TRUE	Reserved
Bit29	Bool	97.5	TRUE	Reserved
Bit30	Bool	97.6	TRUE	1 = Operator can Lock Messages
Bit31	Bool	97.7	TRUE	1 = Operator can Enable/Disable Comms
SampleTime	Real	98.0	1.000000e-001	Sampling time [s]
MSG_EVID_1	DWord	102.0	DW#16#0	Message 1 ID (Alarm)
MSG_EVID_2	DWord	106.0	DW#16#0	Message 2 ID (Evacuation)
MSG_EVID_3	DWord	110.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	114.0	FALSE	Suppress Message 1
MsgSup2	Bool	114.1	FALSE	Suppress Message 2
MsgSup3	Bool	114.2	FALSE	Suppress Message 3
MsgLock	Bool	114.3	FALSE	Message Lock
OUT				
Func_Error	String[8]	116.0	"	Internal Function that Generated the Error
Func_Err_Num	Word	126.0	W#16#0	Function Error number
Gateway_Con_Err	Bool	128.0	FALSE	Gateway Connection Error
Notif_Con_Err	Bool	128.1	FALSE	Notifier Connection Error
P_Last_Event	Int	130.0	0	Panel Last Event Code
P_Fire	Bool	132.0	FALSE	Panel Fire
P_PreAlarm	Bool	132.1	FALSE	Panel PreAlarm
P_Fault	Bool	132.2	FALSE	Panel Fault
P_Device_Disc	Bool	132.3	FALSE	Panel Devices Disconnected
P_Evacuate	Bool	132.4	FALSE	Panel Evacuate
P_Test_in_prog	Bool	132.5	FALSE	Panel Test in Progress
P_Engineer_Mode	Bool	132.6	FALSE	Panel Engineer Mode
P_Sounders_Delay	Bool	132.7	FALSE	Panel Sounders Delayed
P_Sounders_Off	Bool	133.0	FALSE	Panel Sounders Off
P_In_Day_Mode	Bool	133.1	FALSE	Panel System in Day Mode
Zone01_STS	Struct	134.0		Zone 01 Status
Alarm	Bool	134.0	FALSE	Zone in Alarm
PreAlarm	Bool	134.1	FALSE	Zone in PreAlarm
Fault	Bool	134.2	FALSE	Zone in Fault
Walk_Test	Bool	134.3	FALSE	Zone in Test
Parti_Disab	Bool	134.4	FALSE	Zone Partial Disabled
Total_Disab	Bool	134.5	FALSE	Zone Total Disabled
...
Zone32_STS	Struct	196.0		Zone 32 Status
Alarm	Bool	196.0	FALSE	Zone in Alarm
PreAlarm	Bool	196.1	FALSE	Zone in PreAlarm
Fault	Bool	196.2	FALSE	Zone in Fault
Walk_Test	Bool	196.3	FALSE	Zone in Test
Parti_Disab	Bool	196.4	FALSE	Zone Partial Disabled
Total_Disab	Bool	196.5	FALSE	Zone Total Disabled
Status1	DWord	198.0	DW#16#0	Status1 Word
ZStatus1	DWord	202.0	DW#16#0	Status Z1-4 Word
ZStatus2	DWord	206.0	DW#16#0	Status Z5-8 Word
ZStatus3	DWord	210.0	DW#16#0	Status Z9-12 Word
ZStatus4	DWord	214.0	DW#16#0	Status Z13-16 Word
ZStatus5	DWord	218.0	DW#16#0	Status Z17-20 Word
ZStatus6	DWord	222.0	DW#16#0	Status Z21-24 Word
ZStatus7	DWord	226.0	DW#16#0	Status Z25-28 Word
ZStatus8	DWord	230.0	DW#16#0	Status Z29-32 Word
OS_PermlLog	DWord	234.0	DW#16#FFFFFF	Operator Panel permissions: output for OS
OS_PermlLogZ1	DWord	238.0	DW#16#FFFFFF	Operator Zones 1-8 permissions: output for OS
OS_PermlLogZ2	DWord	242.0	DW#16#FFFFFF	Operator Zones 9-16 permissions: output for OS
OS_PermlLogZ3	DWord	246.0	DW#16#FFFFFF	Operator Zones 17-24 permissions: output for OS
OS_PermlLogZ4	DWord	250.0	DW#16#FFFFFF	Operator Zones 25-32 permissions: output for OS
HMINAV	Struct	254.0		HMI Panel Interface
Status	Word	254.0	W#16#0	Status Word
DBNum	Int	256.0	0	DB Number
Btype	Int	258.0	0	Block Type
MsgSTS1	Word	260.0	W#16#0	Message 1 Status
MsgACK1	Bool	262.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	264.0	W#16#0	Message 2 Status
MsgACK2	Bool	266.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	268.0	W#16#0	Message 3 Status
MsgACK3	Bool	270.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:= 'y' ;
Scl_GenerateReferenceData := 'y' ;
Scl_S7ServerActive:= 'y' ;
Scl_CreateObjectCode:= 'y' ;
Scl_OptimizeObjectCode:= 'y' ;
Scl_MonitorArrayLimits:= 'y' ;
Scl_CreateDebugInfo := 'y' ;
Scl_SetOKFlag:= 'n' ;
Scl_SetMaximumStringLength:= '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu BYTE0 in memory
X24 : BOOL; // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL; // 0.7
// second highest Byte in accu BYTE1 in memory
X16 : BOOL; // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL; // 1.7
// third highest Byte in accu BYTE2 in memory
X8 : BOOL; // 2.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 2.7
// lowest Byte in accu BYTE3 in memory
X0 : BOOL; // 3.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 3.7
END_STRUCT
END_TYPE
// SBits32inDWORD UDT101
TYPE SBits16inWORD
STRUCT
// Highest Byte in accu BYTE1 in memory
X8 : BOOL; // 0.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 0.7
// Lowest Byte in accu BYTE2 in memory
X0 : BOOL; // 1.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 1.7
END_STRUCT
END_TYPE
// Zone Status UDT102
TYPE NTZoneSTS
STRUCT
Alarm : BOOL; //Zone in Alarm
PreAlarm : BOOL; //Zone in PreAlarm
Fault : BOOL; //Zone in Fault
Walk Test : BOOL; //Zone in Test
Parti_Disab : BOOL; //Zone Partial Disabled
Total_Disab : BOOL; //Zone Total Disabled
END_STRUCT
END_TYPE

```

```

// Zone Status UDT103
TYPE NTZoneCMD
  STRUCT
    EnableOp      : BOOL;      //Enable Zone
    DisableOp     : BOOL;      //Disable Zone
    StartTestOp   : BOOL;      //Start Zone Test
    StopTestOp    : BOOL;      //Stop Zone Test
  END_STRUCT
END_TYPE

// Pointer Struct UDT104
TYPE PointSTR
  STRUCT
    ID             : BYTE;      //Always B#16#10
    TypeCode       : BYTE;      //Pointer Type
    Quantity       : WORD;      //Quantity of Data
    DB_Num         : WORD;      //Data Block Number
    PArea          : DWORD;     //Data Area Pointer
  END_STRUCT;
END_TYPE

// HMINAV Link Struct UDT106
TYPE HMINAVSTR
  STRUCT
    Status         : WORD;      // Status Word
    DBNum          : INT;       // DB Number
    Btype         : INT;       // Block Type
  END_STRUCT;
END_TYPE

//=====
FUNCTION_BLOCK FB550
TITLE = 'NOTIFIER ID60'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'NOTIF60'
FAMILY: 'NOTIFIER'
KNOW_HOW_PROTECT
{S7_tasklist := 'OB100';
PO_Count := '1';
S7_m_c := 'true';
S7_alarm_ui := '1'}
VAR_INPUT
//Connection VAR
FBID {
  S7_visible :='false';
  S7_link :='false'} : WORD := 16#0000; //Block ID =0 (DO NOT CHANGE)
Id {
  S7_dynamic :='true';
  S7_edit :='para';
  S7_link :='false'} : WORD := W#16#01; //Connection ID (W#16#01..W#16#0FFF)
Dev_ID {
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} : BYTE := B#16#02; //Communication interface (See TCON Block Help)
IP_Address {
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} :
  STRUCT
    Octet_1 : INT := 192; //1th Octet (0..255)
    Octet_2 : INT := 168; //2nd Octet (0..255)
    Octet_3 : INT := 0; //3th Octet (0..255)
    Octet_4 : INT := 1; //4th Octet (0..255)
  END_STRUCT;
  BIPAddr AT IP_Address : ARRAY[1..8] OF BYTE;
Port {
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} : INT := 502; //Master Port
BPort AT Port : ARRAY[1..2] OF BYTE;
Gtw_Conn_Time {
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} : REAL := 5; //Delay time Gateway connection Error [s]
Notif_Conn_Time {
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} : REAL := 5; //Delay time NOTIFIER connection Error [s]
Max_CMD_Time {
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} : REAL := 5; //Max Time for Sending Commands [s]
//Operator Commands
EnaComOp {
  Op_Level :='3';
  S7_m_c :='true';
  S7_visible :='false';
  S7_edit :='para';
  S7_link :='false'} : BOOL; //Operator Enable comms

```

```

SysRstOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator System Reset
TermTestOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator Terminate Test
SilSoundOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator Silence Sounder
MuteBuzzOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator Mute Buzzer
EvacuateOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator Evacuate
TestSoundOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator Test Sounder
ResoundOp {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_visible    :='false';
  S7_edit       :='para';
  S7_link       :='false' } : BOOL;           //Operator Resound Sounder
Zone01_CMD {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_xm_c       :='EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible    :='false';
  S7_edit       :='para';
  BLK_Jump      := '1';
  S7_link       :='false' } : NTZoneCMD;     //Zone 01 CMD
Zone02_CMD {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_xm_c       :='EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible    :='false';
  S7_edit       :='para';
  BLK_Jump      := '1';
  S7_link       :='false' } : NTZoneCMD;     //Zone 02 CMD
Zone03_CMD {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_xm_c       :='EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible    :='false';
  S7_edit       :='para';
  BLK_Jump      := '1';
  S7_link       :='false' } : NTZoneCMD;     //Zone 03 CMD
Zone04_CMD {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_xm_c       :='EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible    :='false';
  S7_edit       :='para';
  BLK_Jump      := '1';
  S7_link       :='false' } : NTZoneCMD;     //Zone 04 CMD
Zone05_CMD {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_xm_c       :='EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible    :='false';
  S7_edit       :='para';
  BLK_Jump      := '1';
  S7_link       :='false' } : NTZoneCMD;     //Zone 05 CMD
Zone06_CMD {
  Op_Level      :='3';
  S7_m_c        :='true';
  S7_xm_c       :='EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible    :='false';

```

```

S7_edit      := 'para';
BLK_Jump    := '1';
S7_link     := 'false' } : NTZoneCMD;           //Zone 06 CMD
Zone07_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 07 CMD
Zone08_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 08 CMD
Zone09_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 09 CMD
Zone10_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 10 CMD
Zone11_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 11 CMD
Zone12_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 12 CMD
Zone13_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 13 CMD
Zone14_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 14 CMD
Zone15_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 15 CMD
Zone16_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
  S7_visible := 'false';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_link   := 'false' } : NTZoneCMD;           //Zone 16 CMD
Zone17_CMD {
  Op_Level  := '3';
  S7_m_c    := 'true';
  S7_xm_c   := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';

```

```

S7_visible := 'false';
S7_edit    := 'para';
BLK_Jump   := '1';
S7_link    := 'false' } : NTZoneCMD;           //Zone 17 CMD
Zone18_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 18 CMD
Zone19_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 19 CMD
Zone20_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 20 CMD
Zone21_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 21 CMD
Zone22_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 22 CMD
Zone23_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 23 CMD
Zone24_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 24 CMD
Zone25_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 25 CMD
Zone26_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 26 CMD
Zone27_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';
  S7_xm_c  := 'EnableOp,TRUE;DisableOp,TRUE;StartTestOp,TRUE;StopTestOp,TRUE;';
  S7_visible := 'false';
  S7_edit    := 'para';
  BLK_Jump   := '1';
  S7_link    := 'false' } : NTZoneCMD;           //Zone 27 CMD
Zone28_CMD {
  Op_Level := '3';
  S7_m_c   := 'true';

```

```

S7_xm_c      := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
S7_visible   := 'false';
S7_edit      := 'para';
BLK_Jump     := '1';
S7_link      := 'false' } : NTZoneCMD;          //Zone 28 CMD
Zone29_CMD {
Op_Level     := '3';
S7_m_c       := 'true';
S7_xm_c      := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
S7_visible   := 'false';
S7_edit      := 'para';
BLK_Jump     := '1';
S7_link      := 'false' } : NTZoneCMD;          //Zone 29 CMD
Zone30_CMD {
Op_Level     := '3';
S7_m_c       := 'true';
S7_xm_c      := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
S7_visible   := 'false';
S7_edit      := 'para';
BLK_Jump     := '1';
S7_link      := 'false' } : NTZoneCMD;          //Zone 30 CMD
Zone31_CMD {
Op_Level     := '3';
S7_m_c       := 'true';
S7_xm_c      := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
S7_visible   := 'false';
S7_edit      := 'para';
BLK_Jump     := '1';
S7_link      := 'false' } : NTZoneCMD;          //Zone 31 CMD
Zone32_CMD {
Op_Level     := '3';
S7_m_c       := 'true';
S7_xm_c      := 'EnableOp, TRUE; DisableOp, TRUE; StartTestOp, TRUE; StopTestOp, TRUE; ';
S7_visible   := 'false';
S7_edit      := 'para';
BLK_Jump     := '1';
S7_link      := 'false' } : NTZoneCMD;          //Zone 32 CMD
OS_Perm {
S7_visible:= 'false' } :
STRUCT
Bit0  : BOOL := 1; //1 = Operator can Reset system
Bit1  : BOOL := 1; //1 = Operator can Terminate Test
Bit2  : BOOL := 1; //1 = Operator can Silence Sounders
Bit3  : BOOL := 1; //1 = Operator can Mute Buzzer
Bit4  : BOOL := 1; //1 = Operator can Evacuate
Bit5  : BOOL := 1; //1 = Operator can Test Sounders
Bit6  : BOOL := 1; //1 = Operator can Resound Sounders
Bit7  : BOOL := 1; //Reserved
Bit8  : BOOL := 1; //1 = Operator can Enable Zone
Bit9  : BOOL := 1; //1 = Operator can Disable Zone
Bit10 : BOOL := 1; //1 = Operator can Start Zone Test
Bit11 : BOOL := 1; //1 = Operator can Stop Zone Test
Bit12 : BOOL := 1; //Reserved
Bit13 : BOOL := 1; //Reserved
Bit14 : BOOL := 1; //Reserved
Bit15 : BOOL := 1; //Reserved
Bit16 : BOOL := 1; //Reserved
Bit17 : BOOL := 1; //Reserved
Bit18 : BOOL := 1; //Reserved
Bit19 : BOOL := 1; //Reserved
Bit20 : BOOL := 1; //Reserved
Bit21 : BOOL := 1; //Reserved
Bit22 : BOOL := 1; //Reserved
Bit23 : BOOL := 1; //Reserved
Bit24 : BOOL := 1; //Reserved
Bit25 : BOOL := 1; //Reserved
Bit26 : BOOL := 1; //Reserved
Bit27 : BOOL := 1; //Reserved
Bit28 : BOOL := 1; //Reserved
Bit29 : BOOL := 1; //Reserved
Bit30 : BOOL := 1; //1 = Operator can Lock Messages
Bit31 : BOOL := 1; //1 = Operator can Enable/Disable Comms
END_STRUCT;
//Operator Permissions
dwOS_Perm AT OS_Perm : DWORD;
ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;
SampleTime {
S7_visible := 'false';
S7_link    := 'false';
S7_sampletime:= 'true';
S7_param   := 'false' } : REAL := 0.1; // Sampling time [s]
//Messages Configuration
MSG_EVID_1 {
S7_visible := 'false';
S7_link    := 'false';
S7_param   := 'false';
S7_server  := 'alarm_archiv';
S7_a_type  := 'alarm_s' } : DWORD:=0; //Message 1 ID (Alarm)

```

```

MSG_EVID_2 {
  S7_visible :='false';
  S7_link :='false';
  S7_param :='false';
  S7_server :='alarm_archiv';
  S7_a_type :='alarm_s'} : DWORD:=0; //Message 2 ID (Evacuation)
MSG_EVID_3 {
  S7_visible :='false';
  S7_link :='false';
  S7_param :='false';
  S7_server :='alarm_archiv';
  S7_a_type :='alarm_s'} : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
  S7_visible :='false';
  S7_m_c :='true';
  S7_string_1 :='activ';
  S7_string_0 :='inactiv'} : BOOL; //Suppress Message 1
MsgSup2 {
  S7_visible :='false';
  S7_m_c :='true';
  S7_string_0 :='inactiv';
  S7_string_1 :='activ'} : BOOL; //Suppress Message 2
MsgSup3 {
  S7_visible :='false';
  S7_m_c :='true';
  S7_string_0 :='inactiv';
  S7_string_1 :='activ'} : BOOL; //Suppress Message 3
MsgLock {
  Op_Level :='3';
  S7_m_c :='true';
  S7_visible :='false';
  S7_link :='false';
  S7_string_0 :='inactiv';
  S7_string_1 :='activ'} : BOOL; //Message Lock
END_VAR

//=====
VAR_OUTPUT
// Error Events
Func_Error {
  S7_visible :='false'} : STRING[8]; //Internal Function that Generated the Error
Func_Err_Num {
  S7_visible :='false'} : WORD; //Function Error number
Gateway_Con_Err {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Error'} : BOOL; //Gateway Connection Error
Notif_Con_Err {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Error'} : BOOL; //Notifier Connection Error

//Panel General Events
P_Last_Event {
  S7_m_c :='true';
  S7_visible :='true';
  S7_dynamic :='true'} : INT; //Panel Last Event Code
P_Fire {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Fire'} : BOOL; //Panel Fire
P_PreAlarm {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='PreAlarm'} : BOOL; //Panel PreAlarm
P_Fault {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Fault'} : BOOL; //Panel Fault
P_Device_Disc {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Disconnect'} : BOOL; //Panel Devices Disconnected
P_Evacuate {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Evacuate'} : BOOL; //Panel Evacuate
P_Test_in_prog {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Test'} : BOOL; //Panel Test in Progress
P_Engineer_Mode {
  S7_visible :='true';
  S7_dynamic :='true';
  S7_string_1 :='Engineer'} : BOOL; //Panel Engineer Mode
P_Sounders_Delay {
  S7_visible :='true';

```

```

    S7_dynamic := 'true';
    S7_string_1 := 'Delayed'} : BOOL; //Panel Sounders Delayed
P_Sounders_Off {
    S7_visible := 'true';
    S7_dynamic := 'true';
    S7_string_1 := 'Off'} : BOOL; //Panel Sounders Off
P_In_Day_Mode {
    S7_visible := 'true';
    S7_dynamic := 'true';
    S7_string_1 := 'In_Day'} : BOOL; //Panel System in Day Mode
//Zones
Zone01_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 01 Status
Zone02_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 02 Status
Zone03_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 03 Status
Zone04_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 04 Status
Zone05_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 05 Status
Zone06_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 06 Status
Zone07_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 07 Status
Zone08_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 08 Status
Zone09_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 09 Status
Zone10_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 10 Status
Zone11_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 11 Status
Zone12_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 12 Status
Zone13_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 13 Status
Zone14_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 14 Status
Zone15_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 15 Status
Zone16_STS {
    S7_visible := 'true'} : NTZoneSTS; //Zone 16 Status
Zone17_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 17 Status
Zone18_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 18 Status
Zone19_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 19 Status
Zone20_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 20 Status
Zone21_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 21 Status
Zone22_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 22 Status
Zone23_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 23 Status
Zone24_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 24 Status
Zone25_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 25 Status
Zone26_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 26 Status
Zone27_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 27 Status
Zone28_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 28 Status
Zone29_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 29 Status
Zone30_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 30 Status
Zone31_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 31 Status
Zone32_STS {
    S7_visible := 'false'} : NTZoneSTS; //Zone 32 Status
Status1 {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD; //Status1 Word
ZStatus1 {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD; //Status Z1-4 Word
ZStatus2 {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD; //Status Z5-8 Word
ZStatus3 {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD; //Status Z9-12 Word

```

```

ZStatus4 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD;           //Status Z13-16 Word
ZStatus5 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD;           //Status Z17-20 Word
ZStatus6 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD;           //Status Z21-24 Word
ZStatus7 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD;           //Status Z25-28 Word
ZStatus8 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD;           //Status Z29-32 Word
OS_PermLog {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD:=16#FFFFFFF; //Operator Panel permissions: output for OS
OS_PermLogZ1 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD:=16#FFFFFFF; //Operator Zones 1-8 permissions: output for OS
OS_PermLogZ2 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD:=16#FFFFFFF; //Operator Zones 9-16 permissions: output for OS
OS_PermLogZ3 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD:=16#FFFFFFF; //Operator Zones 17-24 permissions: output for OS
OS_PermLogZ4 {
    S7_visible:='false';
    S7_m_c:='true'}           : DWORD:=16#FFFFFFF; //Operator Zones 25-32 permissions: output for OS
HMINAV
//Messages Configuration
MsgSTS1 {
    S7_visible := 'false'}    : WORD;           //Message 1 Status
MsgACK1 {
    S7_visible := 'false'}    : BOOL;           //Message 1 Acknowledge
MsgSTS2 {
    S7_visible := 'false'}    : WORD;           //Message 2 Status
MsgACK2 {
    S7_visible := 'false'}    : BOOL;           //Message 2 Acknowledge
MsgSTS3 {
    S7_visible := 'false'}    : WORD;           //Message 3 Status
MsgACK3 {
    S7_visible := 'false'}    : BOOL;           //Message 3 Acknowledge
END_VAR

// Variables =====
VAR

// TCON Connection Parameters structure (UDT65)
CON_PAR:
STRUCT
    block_length      : WORD := W#16#40;           // Length: 64 Bytes (fixed)
    id                 : WORD := W#16#01;           // Connection id (W#16#0001 to W#16#0FFF)
    connection_type    : BYTE := B#16#11;           // Protocol TCP
    active_est         : BOOL := TRUE;              // Connection type active establishment
    local_device_id    : BYTE := B#16#02;           // Communication interface used
    //Codes:
    // B#16#02: Communication via the integrated IE interface FOR the CPUs 315-2 PN/DP AND 317-2 PN/DP
    // B#16#03: Communication via the integrated IE interface FOR the CPUs 315T-3 PN/DP,317T-3PN/DP AND 319-3PN/DP
    // B#16#05: Communication via the integrated IE interface X5 FOR the CPUs 41x AND 41xH (Rack 0)
    // B#16#06: Communication via the IE interface on interface slot 2 (IF2) with WinAC RTX (TCP only)
    // B#16#08: Communication via the integrated IE interface X8 FOR the CPUs 41x AND 41xH (Rack 0)
    // B#16#0B: Communication via the IE interface on interface slot 3 (IF3) with WinAC RTX (TCP only)
    // B#16#0F: Communication via the IE interface on interface slot 4 (IF4) with WinAC RTX (TCP only)
    // B#16#15: Communication via the integrated IE interface X5 FOR the CPUs 41xH (Rack 1)
    // B#16#18: Communication via the integrated IE interface X8 FOR the CPUs 41xH (Rack 1)
    local_tsap_id_len  : BYTE := B#16#00;           // Length of parameter local_tsap_id used
    rem_subnet_id_len  : BYTE := B#16#00;           // NOT used
    rem_staddr_len     : BYTE := B#16#04;           // Length of IP address for the remote connection endpoint
    rem_tsap_id_len    : BYTE := B#16#02;           // Length of the rem_tsap_id parameter used
    next_staddr_len    : BYTE := B#16#00;           // Length of parameter next_staddr used
    local_tsap_id      : ARRAY [1..16] OF BYTE := // Local port number assigned automatically
        16(B#16#00);
    rem_subnet_id      : ARRAY [1..6] OF BYTE := // NOT used. Assign 0
        6(B#16#00);
    rem_staddr         : ARRAY [1..6] OF BYTE := // IP of the remote connection, as instance, 192.168.002.003
        6(B#16#00);
        // rem_staddr[1]=B#16#C0(192),rem_staddr[2]=B#16#A8 (168),
        // rem_staddr[3]=B#16#02(002),rem_staddr[4]=B#16#03 (003)
    rem_tsap_id        : ARRAY [1..16] OF BYTE := // Remote port number (502 by default as MB standar port)
        B#16#01,
        B#16#F6,
        14(B#16#00);
        // rem_tsap_id[1]=high BYTE port no. in hex representation,
        // rem_tsap_id[2]=low BYTE port no. in hex representation,
        // rem_tsap_id[3-16]=B#16#00
    next_staddr        : ARRAY [1..6] OF BYTE := // NOT used for PN interface. Assign 0
        6(B#16#00);
    spare              : WORD := W#16#0000;         // NOT used
END_STRUCT;
BUFF_RD               : ARRAY [1..260] OF BYTE;    // Read Data Buffer

```

```

PanelCMD           : ARRAY [1..7] OF WORD; // Panel Commands
ZonesCMD           : ARRAY [1..32] OF WORD; // Zones Commands
SensorsCMD        : ARRAY [1..99] OF WORD; // Sensors Commands
ModulesCMD        : ARRAY [1..99] OF WORD; // Modules Commands
ZonesSTS          : ARRAY [1..32] OF WORD; // Zones Status
SensorsSTS        : ARRAY [1..99] OF WORD; // Sensors Status
ModulesSTS        : ARRAY [1..99] OF WORD; // Modules Status

// ModBus TCP/IP Application Data Unit Read
ADU_RD:
STRUCT
  Trans_Id         : WORD; // Transaction identifier
  Proto_Id        : WORD := W#16#00; // NOT Used
  Lenght          : WORD := W#16#06; // Remaining Fields byte count (To read, always 6)
  Unit_Id         : BYTE := B#16#00; // ID for serial bridging on a non TCP/IP network
  Func_Code       : BYTE := BYTE#03; // ModBus Function Code (Read Holding Registers)
  Start_Addr     : WORD; // Starting Address Register to be Read
  Qty_Reg        : WORD; // Quantity of Registers to be Read
END_STRUCT;

// ModBus TCP/IP Application Data Unit Write
ADU_WSR:
STRUCT
  Trans_Id         : WORD; // Transaction identifier
  Proto_Id        : WORD := W#16#00; // NOT Used
  Lenght          : WORD := W#16#06; // Remaining Fields byte count (To write single R, always 6)
  Unit_Id         : BYTE := B#16#00; // ID for serial bridging on a non TCP/IP network
  Func_Code       : BYTE := BYTE#06; // ModBus Function Code (Write Single Holding Register)
  Addr           : WORD; // Register Address to Write
  Value          : WORD; // Register Value to Write
END_STRUCT;
RD_MB_JOB         : INT; // Read ModBus Job
MB_JOB_Pre       : INT; // Previous Read ModBus Job

// System Functions
CONNECT          : TCON; // Connection Function
DISCONNECT       : TDISCON; // Disconnection Function
SEND             : TSEND; // Send Data Function
RECEIVE         : TRCV; // Receive Data Function

// Auxiliar Variables
xWrd_Z          : WORD := W#16#0000; // WORD Zero Value
xWrd_OOS        : WORD := W#16#8000; // WORD Out of Service
xCom_Start      : BOOL; // Start Communications
xCom_Run        : BOOL; // Communications Running
xNCom_Panel     : BOOL; // No Communications Panel
xTCON_Err      : BOOL; // TCON function Error
xGtw_Con_Err   : BOOL; // Connection Error with the Gateway
xMB_Err        : BOOL; // ModBus Comm Error
xRx_Err        : BOOL; // Receive Comm Error
xNotif_Con_Err : BOOL; // Connection Error with NOTIFIER
xTx_Cmd_RD     : BOOL; // Read Transmission Command
xTx_Cmd_WT     : BOOL; // Write Transmission Command
xCMD_Tm        : BOOL; // Commands Timer
xMax_CMD_Tm    : BOOL; // Max Time Reached for Commands
xAux_Err       : INT; // Auxiliar Return Function Error
xFirst_CMD_Idx : INT; // First Command Index
xLast_CMD_Idx  : INT := 237; // Last Command Index 7+32+99+99
xIdx           : INT; // Index FOR statement
xWValue        : WORD; // WORD Value
xArrWValue AT xWValue : SBits16inWORD; // WORD Value as Bits
xiWValue AT xWValue : INT; // WORD Value as Int
xBWValue AT xWValue : ARRAY [1..2] OF BYTE; // WORD Value as Bytes
xArrZCMD       : ARRAY [1..32] OF NTZoneCMD; // Zone Command Array
xWArrZCMD AT xArrZCMD : ARRAY [1..32] OF WORD; // Zone Command Array as Words
iRD_TId        : INT; // READ Transaction Identifier as integer
RD_TId AT iRD_TId : ARRAY [1..2] OF BYTE; // READ ID Coded
iLast_Evt      : INT; // READ Panel Last event as integer
BLast_Evt AT iLast_Evt : ARRAY [1..2] OF BYTE; // READ Panel Last event
pCom_Err       : BOOL; // Panel Communication Error
pLast_Event    : INT; // Panel Last Event
pFire          : BOOL; // Panel Fire
pPreAlarm      : BOOL; // Panel PreAlarm
pFault         : BOOL; // Panel Fault
pDevice_Disc   : BOOL; // Panel Devices Disconnected
pEvacuate      : BOOL; // Panel Evacuate
pTest_in_prog  : BOOL; // Panel Test in Progress
pEngineer_Mode : BOOL; // Panel Engineer Mode
pSounders_Delay : BOOL; // Panel Sounders Delayed
pSounders_Off  : BOOL; // Panel Sounders Off
pIn_Day_Mode   : BOOL; // Panel System in Day Mode
GtwMonTime     : REAL; // Current Time for xGtw_Con_Err
NotifMonTime   : REAL; // Current Time for xNotif_Con_Err
CmdMonTime     : REAL; // Current Time for xMax_CMD_Tm
xHMISTS        : WORD; // HMI Status
xBithMISTS AT xHMISTS : SBits16inWORD; // HMI Status as Bits
CMP_ID         : DWORD; // Area code

```

```

xMsgCond1      : BOOL;           // Condition MSG1
xMsgCond2      : BOOL;           // Condition MSG2
xMsgCond3      : BOOL;           // Condition MSG3
xMsgTrig1      : BOOL;           // Trigger MSG1
xMsgTrig2      : BOOL;           // Trigger MSG2
xMsgTrig3      : BOOL;           // Trigger MSG3
xMsgArch1      : INT;            // MSG1 Status
xMsgArch2      : INT;            // MSG2 Status
xMsgArch3      : INT;            // MSG3 Status
AuxValue:      :                // Msg Aux Value
  STRUCT
    Zone        : INT;           // Msg Zone
    DBNum       : WORD;          // Msg DB num
  END_STRUCT;
xRetSts       : BOOL;           // Msg Status
END_VAR

// Temporal Variables =====
VAR_TEMP

// Structures to call RD_SINFO (SFC6) in order to know OB running
sTOP_SI:
STRUCT
  EV_CLASS      : BYTE;         // Bits 0 to 3: Event ID· Bits 4 to 7: Event class
  EV_NUM        : BYTE;         // Event number
  PRIORITY      : BYTE;         // Number of the priority class (meaning OF B#16#FE: OB NOT avail or locked)
  NUM           : BYTE;         // OB number.
  TYP2_3       : BYTE;         // Data ID 2_3: identifies the information entered in ZI2_3
  TYP1         : BYTE;         // Data ID 1 : identifies the information entered in ZI1
  ZI1          : WORD;         // Additional information 1
  ZI2_3        : DWORD;        // Additional information 2_3
END_STRUCT;

sSTART_UP_SI:
STRUCT
  EV_CLASS      : BYTE;         // Bits 0 to 3: Event ID· Bits 4 to 7: Event class
  EV_NUM        : BYTE;         // Event number
  PRIORITY      : BYTE;         // Number of the priority class (meaning OF B#16#FE: OB NOT avail or locked)
  NUM           : BYTE;         // OB number.
  TYP2_3       : BYTE;         // Data ID 2_3: identifies the information entered in ZI2_3
  TYP1         : BYTE;         // Data ID 1 : identifies the information entered in ZI1
  ZI1          : WORD;         // Additional information 1
  ZI2_3        : DWORD;        // Additional information 2_3
END_STRUCT;

// Structures to copy data areas
PSource        : ANY;          // Pointer to Source Data Area
Psrc_STR AT PSource : PointSTR; // Pointer Structure
PDestin       : ANY;          // Pointer to Source Data Area
Pdst_STR AT PDestin : PointSTR; // Pointer Structure

// Variable for status
dwStatus       : DWORD;
ArrBitStatus AT dwStatus : SBits32inDWORD;
ArrByteStatus AT dwStatus : ARRAY[1..4] OF BYTE;

// Variable for Zones
xZSTS         : NTZoneSTS;     // Zone Structure (1 WORD)
xwZSTS AT xZSTS : ARRAY[1..2] OF BYTE; // Zone Structure as Bytes

// Variables for get the working OB
xOB_Cycle     : BOOL;         // Block is running in a cyclic interrupt OB
xOB_Start     : BOOL;         // Block is running in start OB
xSnErr        : INT;          // Error information.

// Variable for Operator Permissions
xdwOS_PermLog : DWORD;        // Temporary variable for the value of OS_PermLog
strxOS_PermLog AT xdwOS_PermLog : ARRAY[0..31] OF BOOL;
ArrxOS_PermLog AT xdwOS_PermLog : ARRAY[0..3] OF BYTE;
END_VAR

// CODE =====
BEGIN
xSnErr      := RD_SINFO(TOP_SI:= sTOP_SI, START_UP_SI:= sSTART_UP_SI);
xOB_Cycle   := (BYTE_TO_INT(sTOP_SI.NUM)>=30) AND (BYTE_TO_INT(sTOP_SI.NUM)<=38);
xOB_Start   := (BYTE_TO_INT(sTOP_SI.NUM)>=100) AND (BYTE_TO_INT(sTOP_SI.NUM)<=102);

xCom_Start := EnaComOp AND xOB_Cycle; // Block is Enabled and working in a Cyclic OB
xNCom_Panel := NOT(xCom_Start) OR Gateway_Con_Err OR Notif_Con_Err;

// Initialize variables
IF NOT(xCom_Start) THEN
  xTCON_Err := FALSE; // Reset Connection Error
  Func_Err_Num := W#16#00; // Reset Error Code
  Func_Error := ''; // Reset Function Name that generated the error
  RD_MB_JOB := 0; // Reset Read ModBus Job
  xRx_Err := FALSE; // Reset Watchdog receive data

```

```

xGtw_Con_Err      := FALSE;           // Reset Connection Error with the Gateway
xMB_Err           := FALSE;           // Reset ModBus Comm Error
xFirst_CMD_Idx    := 1;               // Reset First CMD Index
PSource           := Id;              // Pointer to Source to retrieve DBNum
CMP_ID := WORD_TO_DWORD(PSrc_STR.DB_Num); // DBNum for Messages
AuxValue.Zone     := 0;
AuxValue.DBNum    := PSrc_STR.DB_Num;
END_IF;

//Copy Zone Input commands to manage them in a loop
PSource := Zone01_CMD;               // Pointer to Source
PSrc_STR.TypeCode := BYTE#04;        // Work with WORDS
PSrc_STR.Quantity := WORD#32;        // 32 Words of Data

PDestin := xArrZCMD[1];               // Pointer to Destination
PDst_STR.TypeCode := BYTE#04;        // Work with WORDS
PDst_STR.Quantity := WORD#32;        // 32 Words of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data área

//Fill Zone Registers with the pin Commands
FOR xIdx:=1 TO 32 DO
  IF xArrZCMD[xIdx].EnableOp THEN ZonesCMD[xIdx]:=WORD#1; END_IF;
  IF xArrZCMD[xIdx].DisableOp THEN ZonesCMD[xIdx]:=WORD#2; END_IF;
  IF xArrZCMD[xIdx].StartTestOp THEN ZonesCMD[xIdx]:=WORD#3; END_IF;
  IF xArrZCMD[xIdx].StopTestOp THEN ZonesCMD[xIdx]:=WORD#4; END_IF;
END_FOR;

//Fill Panel Registers with the pin Commands
PanelCMD[1] := BOOL_TO_WORD(SysRstOp); // Operator System Reset
PanelCMD[2] := BOOL_TO_WORD(TermTestOp); // Operator Terminate Test
PanelCMD[3] := BOOL_TO_WORD(SilSoundOp); // Operator Silence Sounder
PanelCMD[4] := BOOL_TO_WORD(MuteBuzzOp); // Operator Mute Buzzer
PanelCMD[5] := BOOL_TO_WORD(EvacuateOp); // Operator Evacuate
PanelCMD[6] := BOOL_TO_WORD(TestSoundOp); // Operator Test Sounder
PanelCMD[7] := BOOL_TO_WORD(ResoundOp); // Operator Resound Sounder

// Reset Commands on error or not start
IF xNCom_Panel THEN
  PDestin := PanelCMD[1];             // Pointer to CMDs
  PDst_STR.TypeCode := BYTE#04;       // Work with WORDS
  PDst_STR.Quantity := WORD#237;      // 237 Words of Data (CMDs)
  xAux_Err:=FILL(BVAL:=xWrd_Z,BLK:=PDestin); // Fill CMDs data with 0 Value
END_IF;

// Initialize Connection Parameters
IF xCom_Start AND NOT(xCom_Run) THEN
  CON_PAR.id := Id;                  // Connection ID
  CON_PAR.local_device_id := Dev_ID; // Communication Interface
  CON_PAR.rem_staddr[1] := BIPAddr[2]; // IP address octet 1
  CON_PAR.rem_staddr[2] := BIPAddr[4]; // IP address octet 2
  CON_PAR.rem_staddr[3] := BIPAddr[6]; // IP address octet 3
  CON_PAR.rem_staddr[4] := BIPAddr[8]; // IP address octet 4
  CON_PAR.rem_tsap_id[1] := BPort[1]; // High Byte Remote Port number
  CON_PAR.rem_tsap_id[2] := BPort[2]; // Low Byte Remote Port number
END_IF;
xCom_Run := xCom_Start;

// Open Communications Channel
CONNECT(REQ:=xCom_Run, ID:=CON_PAR.id,CONNECT:=CON_PAR); //Open Communications Channel
IF CONNECT.ERROR AND (CONNECT.STATUS<> W#16#80A7) THEN //80A7 is generated when disconnect communi before
  Func_Err_Num:=CONNECT.STATUS; //connection is established.not considered as an Error
  Func_Error := 'TCON';
  xTCON_Err := TRUE;
END_IF;

// Communications Channel Opened
IF NOT(xTCON_Err) THEN
  RECEIVE(EN_R:=xCom_Run, ID:=CON_PAR.id,LEN:=0,DATA:=BUFF_RD); // Open Receive Channel
  DISCONNECT(REQ:=NOT(xCom_Run), ID:=CON_PAR.id); // Close Communications

// Evaluate Function Errors
IF DISCONNECT.ERROR THEN // TDISCON Function Error
  Func_Err_Num:=DISCONNECT.STATUS; // TDISCON Error Code is showed
  Func_Error:='TDISCON';
ELSIF RECEIVE.ERROR THEN // TRCV Function Error
  Func_Err_Num:=RECEIVE.STATUS; // TRCV Error Code is showed
  Func_Error:='TRCV';
ELSE
  Func_Err_Num:=W#16#00; // No Function Errors
  Func_Error:='';
END_IF;
xGtw_Con_Err:=RECEIVE.ERROR; // Check Connection with the Gateway

// Read Data
IF RECEIVE.NDR THEN // New Data Read
  xRx_Err := FALSE; // Reset watchdog receive data

```

```

RD_TId[1] := BUFF_RD[1]; // Read Trans ID
RD_TId[2] := BUFF_RD[2];
xMB_Err := BYTE_TO_INT(BUFF_RD[8]) > 128; // Function code h8x=response error
IF NOT (xMB_Err) THEN
CASE iRD_TId OF
0: //Panel Status
IF (RECEIVE.RCVD_LEN=41) THEN // HEAD(7)+FC(1)+BC(1)+DATA(16*2)
pCom_Err := BYTE_TO_BOOL(BUFF_RD[11]); // Panel Communication Error
BLast_Evt[1] := BUFF_RD[12];
BLast_Evt[2] := BUFF_RD[13];
pLast_Event := iLast_Evt; // Panel Last Event
pFire := BYTE_TO_BOOL(BUFF_RD[15]); // Panel Fire
pPreAlarm := BYTE_TO_BOOL(BUFF_RD[17]); // Panel PreAlarm
pFault := BYTE_TO_BOOL(BUFF_RD[19]); // Panel Fault
pDevice_Disc := BYTE_TO_BOOL(BUFF_RD[21]); // Panel Devices Disconnected
pEvacuate := BYTE_TO_BOOL(BUFF_RD[23]); // Panel Evacuate
pTest_in_prog := BYTE_TO_BOOL(BUFF_RD[25]); // Panel Test in Progress
pEngineer_Mode := BYTE_TO_BOOL(BUFF_RD[27]); // Panel Engineer Mode
pSounders_Delay := BYTE_TO_BOOL(BUFF_RD[33]); // Panel Sounders Delayed
pSounders_Off := BYTE_TO_BOOL(BUFF_RD[35]); // Panel Sounders Off
pIn_Day_Mode := BYTE_TO_BOOL(BUFF_RD[39]); // Panel System in Day Mode
ELSE
xMB_Err := TRUE; // Received Data Length Error
END_IF;
1: //Zones Status
IF (RECEIVE.RCVD_LEN=73) THEN // HEAD(7)+FC(1)+BC(1)+DATA(32*2)
PSource := BUFF_RD[10]; // Pointer to Source
PSrc_STR.TypeCode := BYTE#04; // Work with WORDS
PSrc_STR.Quantity := WORD#32; // 32 Words of Data

PDestin := ZonesSTS[1]; // Pointer to Destination
Pdst_STR.TypeCode := BYTE#04; // Work with WORDS
Pdst_STR.Quantity := WORD#32; // 32 Words of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data area
ELSE
xMB_Err := TRUE; // Received Data Length Error
END_IF;
2: //Sensors Status
IF (RECEIVE.RCVD_LEN=207) THEN // HEAD(7)+FC(1)+BC(1)+DATA(99*2)
PSource := BUFF_RD[10]; // Pointer to Source
PSrc_STR.TypeCode := BYTE#04; // Work with WORDS
PSrc_STR.Quantity := WORD#99; // 99 Words of Data

PDestin := SensorsSTS[1]; // Pointer to Destination
Pdst_STR.TypeCode := BYTE#04; // Work with WORDS
Pdst_STR.Quantity := WORD#99; // 99 Words of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data area
ELSE
xMB_Err := TRUE; // Received Data Length Error
END_IF;
3: //Modules Status
IF (RECEIVE.RCVD_LEN=207) THEN // HEAD(7)+FC(1)+BC(1)+DATA(99*2)
PSource := BUFF_RD[10]; // Pointer to Source
PSrc_STR.TypeCode := BYTE#04; // Work with WORDS
PSrc_STR.Quantity := WORD#99; // 99 Words of Data

PDestin := ModulesSTS[1]; // Pointer to Destination
Pdst_STR.TypeCode := BYTE#04; // Work with WORDS
Pdst_STR.Quantity := WORD#99; // 99 Words of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data area
ELSE
xMB_Err := TRUE; // Received Data Length Error
END_IF;
ELSE: //Commands
IF (RECEIVE.RCVD_LEN=12) THEN // HEAD(7)+FC(1)+R@ (2)+DATA(1*2)
xBWValue[1]:=BUFF_RD[9]; // Retrieve the written address
xBWValue[2]:=BUFF_RD[10];
CASE xiWValue OF
16..22: // Panel Commands
PanelCMD[xiWValue-15] :=WORD#0; // Command acknow->Reset Command
2304..2335: // Zones Commands
ZonesCMD[xiWValue-2303] :=WORD#0; // Command acknow->Reset Command
xWArrZCMD[xiWValue-2303]:=WORD#0;
256..354: // Sensors Commands
SensorsCMD[xiWValue-255]:=WORD#0; // Command acknow->Reset Command
356..454: // Modules Commands
ModulesCMD[xiWValue-355]:=WORD#0; // Command acknow->Reset Command
END_CASE;
ELSE
xMB_Err := TRUE; // Received Data Length Error
END_IF;
END_CASE;
END_IF;
END_IF;

```

```

// Request Data
IF NOT(xGtw_Con_Err) AND xCom_Run THEN // Gateway Connection OK
IF NOT(SEND.BUSY) THEN // Previous Transmission Expired

// Write Registers Job
IF NOT(xMax_CMD_Tm) THEN // Max Continuous Time Dedicated to CMDs
FOR xIdx:=xFirst_CMD_Idx TO xLast_CMD_Idx DO // Scan CMD Registers
CASE xIdx OF
1..7: // Panel Commands Registers Area
xWValue := PanelCMD[xIdx]; // Get Command Value
IF xiWValue>0 THEN // There is a Command
ADU_WSR.Value := WORD#01;
ADU_WSR.Addr := INT_TO_WORD(15+xIdx); // @16+(xIdx-1)
ADU_WSR.Trans_Id := ADU_WSR.Addr OR W#16#8000; // TId take Addr, setting Bit 15
xTx_Cmd_WT := TRUE;
xCMD_Tm := TRUE; // Enable Timer Dedicated to CMDs
MB_JOB_Pre := RD_MB_JOB; // Save RD Job to complete the cycle
EXIT;
END_IF;
8..39: // Zones Commands Registers Area
xWValue := ZonesCMD[xIdx-7]; // Get Command Value
IF xiWValue>0 THEN // There is a Command
xiWValue := xiWValue-1; // Value to send
ADU_WSR.Value := xWValue;
ADU_WSR.Addr := INT_TO_WORD(2296+xIdx); // @2304+(xIdx-8)
ADU_WSR.Trans_Id := ADU_WSR.Addr OR W#16#8000; // TId take Addr, setting Bit 15
xTx_Cmd_WT := TRUE;
xCMD_Tm := TRUE; // Enable Timer Dedicated to CMDs
MB_JOB_Pre := RD_MB_JOB; // Save RD Job to complete the cycle
EXIT;
END_IF;
40..138: // Sensors Commands Registers Area
xWValue := SensorsCMD[xIdx-39]; // Get Command Value
IF xiWValue>0 THEN // There is a Command
xiWValue := xiWValue-1; // Value to send
ADU_WSR.Value := xWValue;
ADU_WSR.Addr := INT_TO_WORD(216+xIdx); // @256+(xIdx-40)
ADU_WSR.Trans_Id := ADU_WSR.Addr OR W#16#8000; // TId take Addr, setting Bit 15
xTx_Cmd_WT := TRUE;
xCMD_Tm := TRUE; // Enable Timer Dedicated to CMDs
MB_JOB_Pre := RD_MB_JOB; // Save RD Job to complete the cycle
EXIT;
END_IF;
139..237: // Modules Commands
xWValue := ModulesCMD[xIdx-138]; // Get Command Value
IF xiWValue>0 THEN // There is a Command
xiWValue := xiWValue-1; // Value to send
ADU_WSR.Value := xWValue;
ADU_WSR.Addr := INT_TO_WORD(217+xIdx); // @356+(xIdx-139)
ADU_WSR.Trans_Id := ADU_WSR.Addr OR W#16#8000; // TId take Addr, setting Bit 15
xTx_Cmd_WT := TRUE;
xCMD_Tm := TRUE; // Enable Timer Dedicated to CMDs
MB_JOB_Pre := RD_MB_JOB; // Save RD Job to complete the cycle
EXIT;
END_IF;
END_CASE;
END_FOR;
IF xIdx>xLast_CMD_Idx THEN // FOR statement Reached the Last
xFirst_CMD_Idx := 1; // Reset CMD Index to the Begin
ELSE
xFirst_CMD_Idx := xIdx+1; // Save Idx to Continue on Next Cycle
END_IF;
END_IF;

// Read Registers Job
IF NOT(xTx_Cmd_WT) THEN // Write Job isn't Processing
CASE RD_MB_JOB OF // Read ModBus Job
0: // Panel
ADU_RD.Trans_Id := WORD#00;
ADU_RD.Start_Addr := WORD#00; // Start Address 0
ADU_RD.Qty_Reg := WORD#16; // 16 Registers
RD_MB_JOB := 1;
xTx_Cmd_RD := TRUE;
1: // Zones 1..32
ADU_RD.Trans_Id := WORD#01;
ADU_RD.Start_Addr := WORD#2304; // Start Address 2304
ADU_RD.Qty_Reg := WORD#32; // 32 Registers
RD_MB_JOB := 2;
xTx_Cmd_RD := TRUE;
2: // Sensors
ADU_RD.Trans_Id := WORD#02;
ADU_RD.Start_Addr := WORD#256; // Start Address 256
ADU_RD.Qty_Reg := WORD#99; // 99 Registers
RD_MB_JOB := 3;
xTx_Cmd_RD := TRUE;
3: // Modules
ADU_RD.Trans_Id := WORD#03;

```

```

        ADU_RD.Start_Addr := WORD#356;           // Start Address 356
        ADU_RD.Qty_Reg   := WORD#99;           // 99 Registers
        RD_MB_JOB       := 0;
        xTx_Cmd_RD      := TRUE;
    ELSE:
        RD_MB_JOB := 0;
    END_CASE;
    IF MB_JOB_Pre=RD_MB_JOB THEN xCMD_Tm :=FALSE; END_IF; // RD Job Cycle Completed
END_IF;
END_IF;
END_IF;

SEND(REQ:=xTx_Cmd_RD, ID:=CON_PAR.id, LEN:=12, DATA:=ADU_RD); // Send Data to Read
SEND(REQ:=xTx_Cmd_WT, ID:=CON_PAR.id, LEN:=12, DATA:=ADU_WSR); // Send Data to Write
xTx_Cmd_RD := FALSE;
xTx_Cmd_WT := FALSE;

// Gateway Connection Error Alarm
IF (xGtw_Con_Err OR xMB_Err OR xRx_Err) THEN
    IF (GtwMonTime < Gtw_Conn_Time) THEN
        GtwMonTime := GtwMonTime + SampleTime;
    ELSE
        Gateway_Con_Err := TRUE;
    END_IF;
ELSE
    GtwMonTime := 0.0;
    Gateway_Con_Err := FALSE;
END_IF;
IF NOT(xGtw_Con_Err) AND xCom_Run THEN xRx_Err := TRUE; END_IF; //Set watchdog receive data

// NOTIFIER Connection Error Alarm
IF (pCom_Err) THEN
    IF (NotifMonTime < Notif_Conn_Time) THEN
        NotifMonTime := NotifMonTime + SampleTime;
    ELSE
        Notif_Con_Err := TRUE;
    END_IF;
ELSE
    NotifMonTime := 0.0;
    Notif_Con_Err := FALSE;
END_IF;

// Time Spent Writing
IF (xCMD_Tm) THEN
    IF (CmdMonTime < Max_CMD_Time) THEN
        CmdMonTime := CmdMonTime + SampleTime;
    ELSE
        xMax_CMD_Tm := TRUE;
    END_IF;
ELSE
    CmdMonTime := 0.0;
    xMax_CMD_Tm := FALSE;
END_IF;
END_IF;

//Update pin Panel Commands with the data sent
SysRstOp := WORD_TO_BOOL(PanelCMD[1]); // Operator System Reset
TermTestOp := WORD_TO_BOOL(PanelCMD[2]); // Operator Terminate Test
SilSoundOp := WORD_TO_BOOL(PanelCMD[3]); // Operator Silence Sounder
MuteBuzzOp := WORD_TO_BOOL(PanelCMD[4]); // Operator Mute Buzzer
EvacuateOp := WORD_TO_BOOL(PanelCMD[5]); // Operator Evacuate
TestSoundOp := WORD_TO_BOOL(PanelCMD[6]); // Operator Test Sounder
ResoundOp := WORD_TO_BOOL(PanelCMD[7]); // Operator Resound Sounder

//Update pin Zone Commands with the data sent
PSource := xArrZCMD[1]; // Pointer to Source
Psrc_STR.TypeCode := BYTE#04; // Work with WORDS
Psrc_STR.Quantity := WORD#32; // 32 Words of Data

PDestin := Zone01_CMD; // Pointer to Destination
Pdst_STR.TypeCode := BYTE#04; // Work with WORDS
Pdst_STR.Quantity := WORD#32; // 32 Words of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data área

// Reset status on error or not start
IF xNCom_Panel THEN
    //Panel Status
    pLast_Event := 0; // Reset Panel Last Event
    pFire := FALSE; // Reset Panel Fire
    pPreAlarm := FALSE; // Reset Panel PreAlarm
    pFault := FALSE; // Reset Panel Fault
    pDevice_Disc := FALSE; // Reset Panel Devices Disconnected
    pEvacuate := FALSE; // Reset Panel Evacuate
    pTest_in_prog := FALSE; // Reset Panel Test in Progress
    pEngineer_Mode := FALSE; // Reset Panel Engineer Mode
    pSounders_Delay := FALSE; // Reset Panel Sounders Delayed

```

```

pSounders_Off      := FALSE;           // Reset Panel Sounders Off
pIn_Day_Mode       := FALSE;           // Reset Panel System in Day Mode

//Zones Status
PDestin            := ZonesSTS[1];     // Pointer to Zones
PDst_STR.TypeCode  := BYTE#04;         // Work with WORDS
PDst_STR.Quantity  := WORD#32;         // 32 Words of Data
xAux_Err:=FILL(BVAL:=xWrd_Z,BLK:=PDestin); // Fill Zones data with 0 Value

//Sensors & Modules Status
PDestin            := SensorsSTS[1];   // Pointer to STSs
PDst_STR.TypeCode  := BYTE#04;         // Work with WORDS
PDst_STR.Quantity  := WORD#198;        // 198 Words of Data (STSs)
xAux_Err:=FILL(BVAL:=xWrd_OOS,BLK:=PDestin); // Fill STSs data with OOS Value
END_IF;

// Update OUTPUTS
P_Last_Event       := pLast_Event;     // Panel Last Event
P_Fire              := pFire;           // Panel Fire
P_PreAlarm          := pPreAlarm;       // Panel PreAlarm
P_Fault             := pFault;          // Panel Fault
P_Device_Disc      := pDevice_Disc;    // Panel Devices Disconnected
P_Evacuate          := pEvacuate;       // Panel Evacuate
P_Test_in_prog     := pTest_in_prog;    // Panel Test in Progress
P_Engineer_Mode    := pEngineer_Mode;   // Panel Engineer Mode
P_Sounders_Delay   := pSounders_Delay;  // Panel Sounders Delayed
P_Sounders_Off     := pSounders_Off;    // Panel Sounders Off
P_In_Day_Mode      := pIn_Day_Mode;     // Panel System in Day Mode
FOR xIdx:=1 TO 32 DO
  xWValue           :=ZonesSTS[xIdx];   // Zone Status
  xwZSTS[1]         :=xBWValue[2];      // Take Lowest Byte
  CASE xIdx OF
    1: Zone01_STS:=xZSTS;
    2: Zone02_STS:=xZSTS;
    3: Zone03_STS:=xZSTS;
    4: Zone04_STS:=xZSTS;
    5: Zone05_STS:=xZSTS;
    6: Zone06_STS:=xZSTS;
    7: Zone07_STS:=xZSTS;
    8: Zone08_STS:=xZSTS;
    9: Zone09_STS:=xZSTS;
   10: Zone10_STS:=xZSTS;
   11: Zone11_STS:=xZSTS;
   12: Zone12_STS:=xZSTS;
   13: Zone13_STS:=xZSTS;
   14: Zone14_STS:=xZSTS;
   15: Zone15_STS:=xZSTS;
   16: Zone16_STS:=xZSTS;
   17: Zone17_STS:=xZSTS;
   18: Zone18_STS:=xZSTS;
   19: Zone19_STS:=xZSTS;
   20: Zone20_STS:=xZSTS;
   21: Zone21_STS:=xZSTS;
   22: Zone22_STS:=xZSTS;
   23: Zone23_STS:=xZSTS;
   24: Zone24_STS:=xZSTS;
   25: Zone25_STS:=xZSTS;
   26: Zone26_STS:=xZSTS;
   27: Zone27_STS:=xZSTS;
   28: Zone28_STS:=xZSTS;
   29: Zone29_STS:=xZSTS;
   30: Zone30_STS:=xZSTS;
   31: Zone31_STS:=xZSTS;
   32: Zone32_STS:=xZSTS;
  END_CASE;
END_FOR;

//Status1 DWord
dwStatus           := DWORD#0;
ArrBitStatus.X0   := Gateway_Con_Err;
ArrBitStatus.X1   := Notif_Con_Err;
ArrBitStatus.X2   := pFire;
ArrBitStatus.X3   := pPreAlarm;
ArrBitStatus.X4   := pFault;
ArrBitStatus.X5   := pDevice_Disc;
ArrBitStatus.X6   := pEvacuate;
ArrBitStatus.X7   := pTest_in_prog;
ArrBitStatus.X8   := pEngineer_Mode;
ArrBitStatus.X9   := pSounders_Delay;
ArrBitStatus.X10  := pSounders_Off;
ArrBitStatus.X11  := pIn_Day_Mode;
ArrBitStatus.X12  := NOT(EnaComOp);
ArrBitStatus.X13  := MsgLock;
Status1           := dwStatus;

//ZStatus DWord
FOR xIdx:=1 TO 32 BY 4 DO

```

```

xWVValue      := ZonesSTS[xIdx];
ArrByteStatus[4] := xBWVValue[2];
xWVValue      := ZonesSTS[xIdx+1];
ArrByteStatus[3] := xBWVValue[2];
xWVValue      := ZonesSTS[xIdx+2];
ArrByteStatus[2] := xBWVValue[2];
xWVValue      := ZonesSTS[xIdx+3];
ArrByteStatus[1] := xBWVValue[2];
CASE xIdx OF
1: ZStatus1 := dwStatus;
5: ZStatus2 := dwStatus;
9: ZStatus3 := dwStatus;
13: ZStatus4 := dwStatus;
17: ZStatus5 := dwStatus;
21: ZStatus6 := dwStatus;
25: ZStatus7 := dwStatus;
29: ZStatus8 := dwStatus;
END_CASE;
END_FOR;

//Panel OS_PermLog DWord
ArrBitStatus.X0 := NOT(xNCom_Panel) & OS_Perm.Bit0; //Operator can Reset system
ArrBitStatus.X1 := NOT(xNCom_Panel) & OS_Perm.Bit1; //Operator can Terminate Test
ArrBitStatus.X2 := NOT(xNCom_Panel) & OS_Perm.Bit2; //Operator can Silence Sounders
ArrBitStatus.X3 := NOT(xNCom_Panel) & OS_Perm.Bit3; //Operator can Mute Buzzer
ArrBitStatus.X4 := NOT(xNCom_Panel) & OS_Perm.Bit4; //Operator can Evacuate
ArrBitStatus.X5 := NOT(xNCom_Panel) & OS_Perm.Bit5; //Operator can Test Sounders
ArrBitStatus.X6 := NOT(xNCom_Panel) & OS_Perm.Bit6; //Operator can Resound Sounders
ArrBitStatus.X8 := NOT(xNCom_Panel) & OS_Perm.Bit8; //Operator can Enable Zone
ArrBitStatus.X9 := NOT(xNCom_Panel) & OS_Perm.Bit9; //Operator can Disable Zone
ArrBitStatus.X10 := NOT(xNCom_Panel) & OS_Perm.Bit10; //Operator can Start Zone Test
ArrBitStatus.X11 := NOT(xNCom_Panel) & OS_Perm.Bit11; //Operator can Stop Zone Test
ArrBitStatus.X30 := OS_Perm.Bit30; //Operator can Lock Messages
ArrBitStatus.X31 := OS_Perm.Bit31; //Operator can Enable/Disable Comms
OS_PermLog := dwStatus;

//Zones OS_PermLog DWords
IF xNCom_Panel THEN
// Look permissions on error or not start
PDestin := OS_PermLogZ1; // Pointer to OS_PermLogZx
PDst_STR.TypeCode := BYTE#06; // Work with DWORDS
PDst_STR.Quantity := WORD#04; // 4 DWords of Data
xAux_Err:=FILL(BVAL:=xWrd_Z,BLK:=PDestin); // Fill STSs data with OOS Value
ELSE
FOR xIdx:=1 TO 32 DO
xWVValue := ZonesSTS[xIdx];
CASE xIdx OF
1,9,17,25:
ArrBitStatus.X0 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X1 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X2 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X3 := xArrWVValue.X3 & OS_Perm.Bit11; //bit3=Walk test->Allow stop test
2,10,18,26:
ArrBitStatus.X4 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X5 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X6 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X7 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test
3,11,19,27:
ArrBitStatus.X8 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X9 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X10 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X11 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test
4,12,20,28:
ArrBitStatus.X12 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X13 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X14 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X15 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test
5,13,21,29:
ArrBitStatus.X16 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X17 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X18 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X19 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test
6,14,22,30:
ArrBitStatus.X20 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X21 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X22 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X23 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test
7,15,23,31:
ArrBitStatus.X24 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X25 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X26 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = Walk test -> Allow start test
ArrBitStatus.X27 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test
8,16,24,32:
ArrBitStatus.X28 := xArrWVValue.X5 & OS_Perm.Bit8; // bit5 = Totally disabled -> Allow enable
ArrBitStatus.X29 := NOT(xArrWVValue.X5) & OS_Perm.Bit9; //!bit5 = !Totally disabled -> Allow disable
ArrBitStatus.X30 := NOT(xArrWVValue.X3) & OS_Perm.Bit10; //!bit3 = !Walk test -> Allow start test
ArrBitStatus.X31 := xArrWVValue.X3 & OS_Perm.Bit11; // bit3 = Walk test -> Allow stop test

```

```

        END_CASE;
    CASE xIdx OF
        8: OS_PermLogZ1 :=dwStatus;
        16: OS_PermLogZ2 :=dwStatus;
        24: OS_PermLogZ3 :=dwStatus;
        32: OS_PermLogZ4 :=dwStatus;
    END_CASE;
END_FOR;
END_IF;

//Alarms Managment
xMsgCond1 := pFire AND NOT (MsgSup1 OR MsgLock);
xMsgCond2 := pEvacuate AND NOT (MsgSup2 OR MsgLock);
xMsgCond3 := Notif_Con_Err OR Gateway_Con_Err OR pFault AND NOT (MsgSup3 OR MsgLock);

//Alarm MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
    MsgSTS1:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond1
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_1
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xMsgArch1:=ALARM_SC(
    EV_ID := MSG_EVID_1
    ,STATE := xRetSts
    ,Q_STATE := MsgACK1);

//evacuate MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
    MsgSTS2:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond2
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_2
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xMsgArch2:=ALARM_SC(
    EV_ID := MSG_EVID_2
    ,STATE := xRetSts
    ,Q_STATE := MsgACK2);

//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
    MsgSTS3:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond3
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_3
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xMsgArch3:=ALARM_SC(
    EV_ID := MSG_EVID_3
    ,STATE := xRetSts
    ,Q_STATE := MsgACK3);

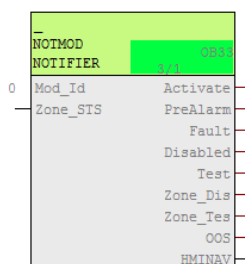
//HMIPAN
//Status
xBithMISTS.X0 := pFire OR pEvacuate;
xBithMISTS.X1 := false;
xBithMISTS.X2 := Notif_Con_Err OR Gateway_Con_Err OR pFault;
xBithMISTS.X3 := false;
//Msg activated
xBithMISTS.X4 := xMsgCond1 OR xMsgCond2;
xBithMISTS.X5 := false;
xBithMISTS.X6 := xMsgCond3;
xBithMISTS.X7 := false;
//Msg Acknowled
xBithMISTS.X8 := MsgACK1 AND MsgACK2;
xBithMISTS.X9 := false;
xBithMISTS.X10 := MsgACK3;
xBithMISTS.X11 := false;
//Msg in Archive (true if msg in Archive)
xBithMISTS.X12 := (xMsgArch1=0) OR (xMsgArch2=0);
xBithMISTS.X13 := false;
xBithMISTS.X14 := (xMsgArch3=0);
xBithMISTS.X15 := false;
HMINAV.Status := xHMISTS;
HMINAV.DBNum := WORD_TO_INT(AuxValue.DBNum);
HMINAV.Btype := 20; //NOTIF60 code=20

END_FUNCTION_BLOCK

```

2.13 FB551: NOTMOD

• Representación Gráfica



• Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
Mod_Id	Int	0	0	Module Id
EnableOp	Bool	2.0	FALSE	Enable Module
DisableOp	Bool	2.1	FALSE	Disable Module
ActivateOp	Bool	2.2	FALSE	Activate Module Output
DeactivateOp	Bool	2.3	FALSE	Deactivate Module Output
RstActCycOp	Bool	2.4	FALSE	Reset Acitvate Cycles
Zone_STS	Any	4.0		Connect to Panel ZoneX_STS
OS_Perm	Struct	14.0		Operator Permissions
Bit0	Bool	14.0	TRUE	1 = Operator can Enable Module
Bit1	Bool	14.1	TRUE	1 = Operator can Disable Module
Bit2	Bool	14.2	TRUE	1 = Operator can Activate Module Output
Bit3	Bool	14.3	TRUE	1 = Operator can Deactivate Module Output
Bit4	Bool	14.4	TRUE	1 = Operator can Reset Activated Cycles Totalizer
Bit5	Bool	14.5	TRUE	1 = Operator can Lock Messages
Bit6	Bool	14.6	TRUE	Reserved
Bit7	Bool	14.7	TRUE	Reserved
Bit8	Bool	15.0	TRUE	Reserved
Bit9	Bool	15.1	TRUE	Reserved
Bit10	Bool	15.2	TRUE	Reserved
Bit11	Bool	15.3	TRUE	Reserved
Bit12	Bool	15.4	TRUE	Reserved
Bit13	Bool	15.5	TRUE	Reserved
Bit14	Bool	15.6	TRUE	Reserved
Bit15	Bool	15.7	TRUE	Reserved
Bit16	Bool	16	TRUE	Reserved
Bit17	Bool	16.1	TRUE	Reserved
Bit18	Bool	16.2	TRUE	Reserved
Bit19	Bool	16.3	TRUE	Reserved
Bit20	Bool	16.4	TRUE	Reserved
Bit21	Bool	16.5	TRUE	Reserved
Bit22	Bool	16.6	TRUE	Reserved
Bit23	Bool	16.7	TRUE	Reserved
Bit24	Bool	17.0	TRUE	Reserved
Bit25	Bool	17.1	TRUE	Reserved
Bit26	Bool	17.2	TRUE	Reserved
Bit27	Bool	17.3	TRUE	Reserved
Bit28	Bool	17.4	TRUE	Reserved
Bit29	Bool	17.5	TRUE	Reserved
Bit30	Bool	17.6	TRUE	Reserved
Bit31	Bool	17.7	TRUE	Reserved
Feature	Struct	18.0		Status of various features
Bit0	Bool	18.0	TRUE	1 = Lock Messages on Module Disabled
Bit1	Bool	18.1	TRUE	1 = Lock Messages on Zone Disabled
Bit2	Bool	18.2	TRUE	1 = Lock Messages on Module Test
Bit3	Bool	18.3	TRUE	1 = Lock Messages on Zone Test
Bit4	Bool	18.4	TRUE	1 = Lock Alarm Messages when Zone is out of Alarm
Bit5	Bool	18.5	TRUE	1 = Lock PreAlarm Messages for Zone is out of PreAlarm
Bit6	Bool	18.6	FALSE	Reserved
Bit7	Bool	18.7	FALSE	Reserved
Bit8	Bool	19.0	FALSE	Reserved
Bit9	Bool	19.1	FALSE	Reserved
Bit10	Bool	19.2	FALSE	Reserved
Bit11	Bool	19.3	FALSE	Reserved
Bit12	Bool	19.4	FALSE	Reserved
Bit13	Bool	19.5	FALSE	Reserved
Bit14	Bool	19.6	FALSE	Reserved
Bit15	Bool	19.7	FALSE	1 = Output Module Type

Name	Data Type	Address	Initial Value	Comment
MSG_EVID_1	DWord	20.0	DW#16#0	Message 1 ID (Activate)
MSG_EVID_2	DWord	24.0	DW#16#0	Message 2 ID (PreAlarm)
MSG_EVID_3	DWord	28.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	32.0	FALSE	Suppress Message 1
MsgSup2	Bool	32.1	FALSE	Suppress Message 2
MsgSup3	Bool	32.2	FALSE	Suppress Message 3
MsgLock	Bool	32.3	FALSE	Message Lock
OUT				
Activate	Bool	34.0	FALSE	Module Activate
PreAlarm	Bool	34.1	FALSE	Module PreAlarm
Fault	Bool	34.2	FALSE	Module Fault
Disabled	Bool	34.3	FALSE	Module Disabled
Test	Bool	34.4	FALSE	Module in Test
Zone_Dis	Bool	34.5	FALSE	Zone Disabled
Zone_Test	Bool	34.6	FALSE	Zone in Test
Zone_Num	Int	36.0	0	Zone Number
QOS	Bool	38.0	FALSE	Out of Service
Test_D	DInt	40.0	L#0	Last Test Date
Test_T	DInt	44.0	L#0	Last Test Time
Activ_D	DInt	48.0	L#0	Last Activate Date
Activ_T	DInt	52.0	L#0	Last Activate Time
Activ_Cyc	DInt	56.0	L#0	Activate Cycles
OS_PermOut	DWord	60.0	DW#16#FFFFFFF	Operator permission: output for OS
OS_PermLog	DWord	64.0	DW#16#FFFFFFF	Operator permission: output for OS
Status1	DWord	68.0	DW#16#0	Status1 Word
HMINAV	Struct	72.0		HMI Panel Interface
Status	Word	72.0	W#16#0	Status Word
DBNum	Int	74.0	0	DB Number
Btype	Int	76.0	0	Block Type
MsgSTS1	Word	78.0	W#16#0	Message 1 Status
MsgACK1	Bool	80.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	82.0	W#16#0	Message 2 Status
MsgACK2	Bool	84.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	86.0	W#16#0	Message 3 Status
MsgACK3	Bool	88.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:=          'y' ;
Scl_GenerateReferenceData :=   'y' ;
Scl_S7ServerActive:=          'y' ;
Scl_CreateObjectCode:=        'y' ;
Scl_OptimizeObjectCode:=      'y' ;
Scl_MonitorArrayLimits:=      'y' ;
Scl_CreateDebugInfo :=        'y' ;
Scl_SetOKFlag:=                'n' ;
Scl_SetMaximumStringLength:=   '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu          BYTE0 in memory
X24 : BOOL;                    // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL;                    // 0.7
// second highest Byte in accu  BYTE1 in memory
X16 : BOOL;                    // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL;                    // 1.7
// third highest Byte in accu   BYTE2 in memory
X8  : BOOL;                    // 2.0
X9  : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL;                    // 2.7
// lowest Byte in accu          BYTE3 in memory
X0  : BOOL;                    // 3.0
X1  : BOOL;
X2  : BOOL;
X3  : BOOL;
X4  : BOOL;
X5  : BOOL;
X6  : BOOL;
X7  : BOOL;                    // 3.7
END_STRUCT
END_TYPE

// Zone Status UDT103
TYPE NTZoneSTS
STRUCT
Alarm      : BOOL;           //Zone in Alarm
PreAlarm   : BOOL;           //Zone in PreAlarm
Fault      : BOOL;           //Zone in Fault
Walk Test  : BOOL;           //Zone in Test
Parti_Disab : BOOL;         //Zone Partial Disabled
Total_Disab : BOOL;         //Zone Total Disabled
END_STRUCT
END_TYPE

// Pointer Struct UDT104
TYPE PointSTR
STRUCT
ID          : BYTE;          // Always B#16#10
TypeCode    : BYTE;          // Pointer Type
Quantity    : WORD;          // Quantity of Data
DB_Num      : WORD;          // Data Block Number
PArea       : DWORD;         // Data Area Pointer
END_STRUCT;
END_TYPE

// HMINAV Link Struct UDT106
TYPE HMINAVSTR
STRUCT
Status      : WORD;          // Status Word
DBNum       : INT;           // DB Number
Btype       : INT;           // Block Type
END_STRUCT;
END_TYPE

```

```

FUNCTION_BLOCK FB551
TITLE = 'NOTIFIER Module'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'NOTMOD'
FAMILY: 'NOTIFIER'
KNOW_HOW_PROTECT
{PO_Count := '1';
 S7_m_c := 'true';
 S7_alarm_ui := '1'}

//=====
VAR_INPUT
Mod_Id {
  S7_m_c := 'true';
  S7_visible := 'true';
  S7_edit := 'para';
  S7_link := 'false'} : INT; //Module Id
EnableOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Enable Module
DisableOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Disable Module
ActivateOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Activate Module Output
DeactivateOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Deactivate Module Output
RstActCycOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Reset Acitvate Cycles
Zone_STS {
  S7_edit := 'para';
  BLK_Jump := '1'} : ANY; //Connect to Panel ZoneX_STS
//Operator Permissions
OS_Perm {
  S7_visible:= 'false'} :
  STRUCT
  Bit0 : BOOL := 1; //1 = Operator can Enable Module
  Bit1 : BOOL := 1; //1 = Operator can Disable Module
  Bit2 : BOOL := 1; //1 = Operator can Activate Module Output
  Bit3 : BOOL := 1; //1 = Operator can Deactivate Module Output
  Bit4 : BOOL := 1; //1 = Operator can Reset Activated Cycles Totalizer
  Bit5 : BOOL := 1; //1 = Operator can Lock Messages
  Bit6 : BOOL := 1; //Reserved
  Bit7 : BOOL := 1; //Reserved
  Bit8 : BOOL := 1; //Reserved
  Bit9 : BOOL := 1; //Reserved
  Bit10 : BOOL := 1; //Reserved
  Bit11 : BOOL := 1; //Reserved
  Bit12 : BOOL := 1; //Reserved
  Bit13 : BOOL := 1; //Reserved
  Bit14 : BOOL := 1; //Reserved
  Bit15 : BOOL := 1; //Reserved
  Bit16 : BOOL := 1; //Reserved
  Bit17 : BOOL := 1; //Reserved
  Bit18 : BOOL := 1; //Reserved
  Bit19 : BOOL := 1; //Reserved
  Bit20 : BOOL := 1; //Reserved
  Bit21 : BOOL := 1; //Reserved
  Bit22 : BOOL := 1; //Reserved
  Bit23 : BOOL := 1; //Reserved
  Bit24 : BOOL := 1; //Reserved
  Bit25 : BOOL := 1; //Reserved
  Bit26 : BOOL := 1; //Reserved
  Bit27 : BOOL := 1; //Reserved
  Bit28 : BOOL := 1; //Reserved
  Bit29 : BOOL := 1; //Reserved
  Bit30 : BOOL := 1; //Reserved
  Bit31 : BOOL := 1; //Reserved
  END_STRUCT;
  dwOS_Perm AT OS_Perm : DWORD;
  ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;

```

```

//FB Features
Feature {
  S7_visible:='false':
  STRUCT
    Bit0   : BOOL := 1; //1 = Lock Messages on Module Disabled
    Bit1   : BOOL := 1; //1 = Lock Messages on Zone Disabled
    Bit2   : BOOL := 1; //1 = Lock Messages on Module Test
    Bit3   : BOOL := 1; //1 = Lock Messages on Zone Test
    Bit4   : BOOL := 1; //1 = Lock Alarm Messages when Zone is out of Alarm
    Bit5   : BOOL := 1; //1 = Lock PreAlarm Messages for Zone is out of PreAlarm
    Bit6   : BOOL; //Reserved
    Bit7   : BOOL; //Reserved
    Bit8   : BOOL; //Reserved
    Bit9   : BOOL; //Reserved
    Bit10  : BOOL; //Reserved
    Bit11  : BOOL; //Reserved
    Bit12  : BOOL; //Reserved
    Bit13  : BOOL; //Reserved
    Bit14  : BOOL; //Reserved
    Bit15  : BOOL; //1 = Output Module Type
  END_STRUCT; // Status of various features
//Messages Configuration
MSG_EVID_1 {
  S7_visible :='false';
  S7_link :='false';
  S7_param :='false';
  S7_server :='alarm_archiv';
  S7_a_type :='alarm_s'} : DWORD:=0; //Message 1 ID (Activate)
MSG_EVID_2 {
  S7_visible :='false';
  S7_link :='false';
  S7_param :='false';
  S7_server :='alarm_archiv';
  S7_a_type :='alarm_s'} : DWORD:=0; //Message 2 ID (PreAlarm)
MSG_EVID_3 {
  S7_visible :='false';
  S7_link :='false';
  S7_param :='false';
  S7_server :='alarm_archiv';
  S7_a_type :='alarm_s'} : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
  S7_visible :='false';
  S7_m_c :='true';
  S7_string_1 :='activ';
  S7_string_0 :='inactiv'} : BOOL; //Suppress Message 1
MsgSup2 {
  S7_visible :='false';
  S7_m_c :='true';
  S7_string_0 :='inactiv';
  S7_string_1 :='activ'} : BOOL; //Suppress Message 2
MsgSup3 {
  S7_visible :='false';
  S7_m_c :='true';
  S7_string_0 :='inactiv';
  S7_string_1 :='activ'} : BOOL; //Suppress Message 3
MsgLock {
  Op_Level :='3';
  S7_m_c :='true';
  S7_visible :='false';
  S7_link :='false';
  S7_string_0 :='inactiv';
  S7_string_1 :='activ'} : BOOL; //Message Lock
END_VAR

//=====
VAR_OUTPUT
Activate {
  S7_dynamic :='true';
  S7_m_c :='true';
  S7_string_0 :='';
  S7_string_1 :='Activate'} : BOOL; //Module Activate
PreAlarm {
  S7_dynamic :='true';
  S7_string_1 :='PreAlarm'} : BOOL; //Module PreAlarm
Fault {
  S7_dynamic :='true';
  S7_string_1 :='Fault'} : BOOL; //Module Fault
Disabled {
  S7_dynamic :='true';
  S7_string_1 :='Disabled'} : BOOL; //Module Disabled
Test {
  S7_dynamic :='true';
  S7_string_1 :='Test'} : BOOL; //Module in Test
Zone_Dis {
  S7_dynamic :='true';
  S7_string_1 :='Disabled'} : BOOL; //Zone Disabled
Zone_Test {

```

```

    S7_dynamic := 'true';
    S7_string_1 := 'Test' } : BOOL; //Zone in Test
Zone_Num {
    S7_m_c := 'true';
    S7_visible := 'false' } : INT; //Zone Number
OOS {
    S7_dynamic := 'true';
    S7_string_1 := 'OOS' } : BOOL; //Out of Service
Test_D {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Test Date
Test_T {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Test Time
Activ_D {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Activate Date
Activ_T {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Activate Time
Activ_Cyc {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Activate Cycles
OS_Permit {
    S7_visible := 'false';
    S7_m_c := 'true' } : DWORD := 16#FFFFFFFF; //Operator permission: output for OS
ArrOS_Permit AT OS_Permit : ARRAY[0..3] OF BYTE;
OS_PermitLog {
    S7_visible := 'false';
    S7_m_c := 'true' } : DWORD := 16#FFFFFFFF; //Operator permission: output for OS
ArrOS_PermitLog AT OS_PermitLog : ARRAY[0..3] OF BYTE;
Status1 {
    S7_visible := 'false';
    S7_m_c := 'true' } : DWORD; //Status1 Word
HMINAV : HMINAVSTR; //HMI Panel Interface
//Messages Configuration
MsgSTS1 {
    S7_visible := 'false' } : WORD; //Message 1 Status
MsgACK1 {
    S7_visible := 'false' } : BOOL; //Message 1 Acknowledge
MsgSTS2 {
    S7_visible := 'false' } : WORD; //Message 2 Status
MsgACK2 {
    S7_visible := 'false' } : BOOL; //Message 2 Acknowledge
MsgSTS3 {
    S7_visible := 'false' } : WORD; //Message 3 Status
MsgACK3 {
    S7_visible := 'false' } : BOOL; //Message 3 Acknowledge
END_VAR
//=====
CONST
ZBAddrCPU := 134; // Zones Byte Base Address
MCBAddrCPU := 872; // Mod Commands Byte Base Address
MSBAddrCPU := 1332; // Mod Status Byte Base Address
ZBAddrCP := 124; // Zones Byte Base Address
MCBAddrCP := 830; // Mod Commands Byte Base Address
MSBAddrCP := 1290; // Mod Status Byte Base Address
END_CONST
//=====
VAR
FBID : WORD; // Panel type
ZBAddr : INT; // Zones Byte Base Address
MCBAddr : INT; // Mod Commands Byte Base Address
MSBAddr : INT; // Mod Status Byte Base Address

xZonSTS : NTZoneSTS; // Zone Status
xModSTS : WORD; // Module Status
xBitModSTS AT xModSTS : SBits16inWORD; // Module as Bits
xHMISTS : WORD; // HMI Status
xBitHMISTS AT xHMISTS : SBits16inWORD; // HMI Status as Bits
OpCmd : INT; // Operator Command
xModCMD : WORD; // Module Command
xiModCMD : INT; // Module Command as INT

CMP_ID : DWORD; // Area code
xMsgCond1 : BOOL; // Condition MSG1
xMsgCond2 : BOOL; // Condition MSG2
xMsgCond3 : BOOL; // Condition MSG3
xMsgTrig1 : BOOL; // Trigger MSG1
xMsgTrig2 : BOOL; // Trigger MSG2
xMsgTrig3 : BOOL; // Trigger MSG3
xMsgArch1 : INT; // MSG1 Status
xMsgArch2 : INT; // MSG2 Status
xMsgArch3 : INT; // MSG3 Status

```

```

AuxValue:                                     // Msg Aux Value
  STRUCT
    Zone           : INT;           // Msg Zone
    DBNum          : WORD;          // Msg DB num
  END_STRUCT;
xRetSts         : BOOL;           // Msg Status
xMTimeTest      : BOOL;           // Mem Test Time flag
xMTimeActiv     : BOOL;           // Mem Activ Time flag
END_VAR
//=====
VAR_TEMP
xAux_Err        : INT;           // Auxiliar Return Function Error
xAddr           : DINT;
xFMsgLock       : BOOL;          // MsgLock by Features Bits
xMsg1Zone       : BOOL;          // MsgLock1 by Zone Alarm Status
xMsg2Zone       : BOOL;          // MsgLock2 by Zone PreAlarm Status

// Structures to copy data areas
PPanel          : ANY;           // Pointer to Panel Block
PPan_STR AT PPanel : PointSTR;   // Pointer Structure
PLocal          : ANY;           // Pointer to Local Var
PLoc_STR AT PLocal : PointSTR;   // Pointer Structure

// Variable for Operator Permissions
xdwOS_PermLog   : DWORD; // Temporary variable for the value of OS_PermLog
strxOS_PermLog AT xdwOS_PermLog :
  STRUCT
    Bit0      : BOOL; //1 = Operator can Enable Module
    Bit1      : BOOL; //1 = Operator can Disable Module
    Bit2      : BOOL; //1 = Operator can Activate Module Output
    Bit3      : BOOL; //1 = Operator can Deactivate Module Output
    Bit4      : BOOL; //1 = Operator can Reset Activated Cycle Totalizer
    Bit5      : BOOL; //1 = Operator can Lock Messages
    Bit6      : BOOL; //Not used
    Bit7      : BOOL; //Not used
    Bit8      : BOOL; //Not used
    Bit9      : BOOL; //Not used
    Bit10     : BOOL; //Not used
    Bit11     : BOOL; //Not used
    Bit12     : BOOL; //Not used
    Bit13     : BOOL; //Not used
    Bit14     : BOOL; //Not used
    Bit15     : BOOL; //Not used
    Bit16     : BOOL; //Not used
    Bit17     : BOOL; //Not used
    Bit18     : BOOL; //Not used
    Bit19     : BOOL; //Not used
    Bit20     : BOOL; //Not used
    Bit21     : BOOL; //Not used
    Bit22     : BOOL; //Not used
    Bit23     : BOOL; //Not used
    Bit24     : BOOL; //Not used
    Bit25     : BOOL; //Not used
    Bit26     : BOOL; //Not used
    Bit27     : BOOL; //Not used
    Bit28     : BOOL; //Not used
    Bit29     : BOOL; //Not used
    Bit30     : BOOL; //Not used
    Bit31     : BOOL; //Not used
  END_STRUCT;
ArrxOS_PermLog AT xdwOS_PermLog : ARRAY[0..3] OF BYTE;

// Variable for status
dwStatus      : DWORD;
ArrBitStatus AT dwStatus : SBits32inDWORD;

// Variable for Maintenance Time/Date
dtCPU_Time    : DATE_AND_TIME; // Actual CPU time
dtCPU_Time_st AT dtCPU_Time :
  STRUCT
    b0 : BYTE;
    b1 : BYTE;
    b2 : BYTE;
    b3 : BYTE;
    b4 : BYTE;
    b5 : BYTE;
    b6 : BYTE;
    b7 : BYTE;
  END_STRUCT;
SnREAD_CLK    : INT; // Error information READ_CLK;
cDate         : DINT; // Current Date
cTime         : DINT; // Current Time
TimeTest      : BOOL; // Test Time flag
TimeActiv     : BOOL; // Activate Time flag
END_VAR

```

```

//=====
BEGIN
IF Mod_Id>99 THEN Mod_Id:=99; END_IF;
IF Mod_Id<1 THEN Mod_Id:=1; END_IF;

//Panel Type to knows base addresses
PPanel := Zone_STS; // Pointer to Panel Block
PPan_STR.TypeCode := BYTE#04; // Work with WORDS
PPan_STR.Quantity := WORD#01; // 1 Word of Data
PPan_STR.PArea := DW#16#84000000; // First Word of the DB
PLocal := FBID;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ZoneX_STS
IF FBID=WORD#01 THEN // ZITONCP
    ZBAddr := ZBAddrCP;
    MCBAddr := MCBAddrCP;
    MSBAddr := MSBAddrCP;
ELSE
    ZBAddr := ZBAddrCPU;
    MCBAddr := MCBAddrCPU;
    MSBAddr := MSBAddrCPU;
END_IF;

//Zone outputs
PPanel := Zone_STS; // Pointer to Panel Block
PPan_STR.TypeCode := BYTE#02; // Work with BYTES
PPan_STR.Quantity := WORD#01; // 1 Byte of Data
PLocal := xZonSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve NTZoneSTS
Zone_Dis := xZonSTS.Total_Disab; // Zone Disabled
Zone_Test := xZonSTS.Walk_Test; // Zone Test
Zone_Num :=1+((DWORD_TO_INT(PPan_STR.PArea)-ZBAddr*8)/16); // Calculate Zone Number
CMP_ID := WORD_TO_DWORD(PLoc_STR.DB_Num); // DBNum for Messages
AuxValue.Zone := Zone_Num;
AuxValue.DBNum := PLoc_STR.DB_Num;

//Module Outputs
PPan_STR.TypeCode := BYTE#04; // Work with WORDS
PPan_STR.Quantity := WORD#01; // 1 Word of Data
xAddr := (MSBAddr+2*(Mod_Id-1))*8; // Addr in bits
PPan_STR.PArea := DINT_TO_DWORD(xAddr) OR DW#16#84000000;
PLocal := xModSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ModuleSTS[x]
Activate := xBitModSTS.X0;
PreAlarm := xBitModSTS.X1;
Fault := xBitModSTS.X2;
Disabled := xBitModSTS.X3;
Test := xBitModSTS.X4;
OOS := xBitModSTS.X15;

// Pointer to ModuleCMD[x]
PPan_STR.TypeCode := BYTE#04; // Work with WORDS
PPan_STR.Quantity := WORD#01; // 1 Word of Data
xAddr := (MCBAddr+2*(Mod_Id-1))*8; // Addr in bits
PPan_STR.PArea := DINT_TO_DWORD(xAddr) OR DW#16#84000000;

//Write New command
IF EnableOp THEN OpCmd:=1; END_IF;
IF DisableOp THEN OpCmd:=2; END_IF;
IF ActivateOp THEN OpCmd:=3; END_IF;
IF DeactivateOp THEN OpCmd:=4; END_IF;
IF OpCmd<>xiModCMD THEN // New Command
    xModCMD := INT_TO_WORD(OpCmd); // Value to send
    PLocal := xModCMD;
    xAux_Err :=BLKMOV(SRCBLK := PLocal, DSTBLK := PPanel); // Write ModuleCMD[x]
END_IF;

//Read Current Module Command
PLocal := xModCMD;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ModuleCMD[x]
xiModCMD := WORD_TO_INT(xModCMD);

//Update Operator Commands
IF xiModCMD=0 THEN OpCmd:=0; END_IF; // Reset Command
IF OpCmd<>1 THEN EnableOp:=0; END_IF;
IF OpCmd<>2 THEN DisableOp:=0; END_IF;
IF OpCmd<>3 THEN ActivateOp:=0; END_IF;
IF OpCmd<>4 THEN DeactivateOp:=0; END_IF;

//OS Permissives
ArrOS_PermOut[3] := ArrOS_Perm[0];
ArrOS_PermOut[2] := ArrOS_Perm[1];
ArrOS_PermOut[1] := ArrOS_Perm[2];
ArrOS_PermOut[0] := ArrOS_Perm[3];
IF OOS THEN
    OS_PermLog := DW#16#30 AND OS_PermOut; //MsgLock & Reset Totalizer Mask
ELSE
    xdwOS_PermLog := dwOS_Perm;

```

```

IF NOT(Disabled) THEN strxOS_PermLog.Bit0 := false; //Enable Module Permission
ELSE strxOS_PermLog.Bit0 := OS_Perm.Bit0; END_IF;
IF Disabled THEN strxOS_PermLog.Bit1 := false; //Disable Module Permission
ELSE strxOS_PermLog.Bit1 := OS_Perm.Bit1; END_IF;
IF Activate OR (NOT(Test) AND NOT(Zone_Test)) THEN strxOS_PermLog.Bit2 := false; //Act Out Perm
ELSE strxOS_PermLog.Bit2 := OS_Perm.Bit2; END_IF;
IF NOT(Activate) OR (NOT(Test) AND NOT(Zone_Test)) THEN strxOS_PermLog.Bit3 := false; //Deact out Pe
ELSE strxOS_PermLog.Bit3 := OS_Perm.Bit3; END_IF;
IF Activ_Cyc=0 THEN strxOS_PermLog.Bit4 := false; //Activated Cycle Tot Permission
ELSE strxOS_PermLog.Bit4 := OS_Perm.Bit4; END_IF;
strxOS_PermLog.Bit5 := OS_Perm.Bit5; //Lock Messages Permission

//OS_PermLog
ArrOS_PermLog[3] := ArrxOS_PermLog[0];
ArrOS_PermLog[2] := ArrxOS_PermLog[1];
ArrOS_PermLog[1] := ArrxOS_PermLog[2];
ArrOS_PermLog[0] := ArrxOS_PermLog[3];
END_IF;

//Status1 DWord
dwStatus := DWORD#0;
ArrBitStatus.X0 := Activate;
ArrBitStatus.X1 := PreAlarm;
ArrBitStatus.X2 := Fault;
ArrBitStatus.X3 := Disabled;
ArrBitStatus.X4 := Test;
ArrBitStatus.X5 := OOS;
ArrBitStatus.X6 := Zone_Dis;
ArrBitStatus.X7 := Zone_Test;
ArrBitStatus.X8 := MsgLock;
ArrBitStatus.X9 := MsgSup1;
ArrBitStatus.X10 := MsgSup2;
ArrBitStatus.X11 := MsgSup3;
ArrBitStatus.X12 := xMsgCond1;
ArrBitStatus.X13 := xMsgCond2;
ArrBitStatus.X14 := xMsgCond3;
ArrBitStatus.X15 := Feature.Bit15; //Output Module TYPE
Status1 := dwStatus;

//Alarms Managment
xFMsgLock := (Disabled AND Feature.Bit0) OR //Lock Messages on Module Disabled
              (Zone_Dis AND Feature.Bit1) OR //Lock Messages on Zone Disabled
              (Test AND Feature.Bit2) OR //Lock Messages on Module Test
              (Zone_Test AND Feature.Bit3); //Lock Messages on Zone Test

xMsg1Zone := NOT(xZonSTS.Alarm) AND Feature.Bit4; //Lock Alarm Messages when Zone is out of Alarm
xMsg2Zone := NOT(xZonSTS.PreAlarm) AND Feature.Bit5; //Lock PreAlarm Mess when Zone is out of PreAlarm
xMsgCond1 := Activate AND NOT(MsgSup1 OR MsgLock OR xFMsgLock OR xMsg1Zone);
xMsgCond2 := PreAlarm AND NOT(MsgSup2 OR MsgLock OR xFMsgLock OR xMsg2Zone);
xMsgCond3 := Fault AND NOT(MsgSup3 OR MsgLock OR xFMsgLock);
AuxValue.Zone := Zone_Num;

//Activate MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
    MsgSTS1:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond1
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_1
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xMsgArch1:=ALARM_SC(
    EV_ID := MSG_EVID_1
    ,STATE := xRetSts
    ,Q_STATE := MsgACK1);

//PreAlarm MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
    MsgSTS2:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond2
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_2
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xMsgArch2:=ALARM_SC(
    EV_ID := MSG_EVID_2
    ,STATE := xRetSts
    ,Q_STATE := MsgACK2);

//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
    MsgSTS3:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond3
        ,ID := W#16#EEEE

```

```

, EV_ID := MSG_EVID_3
, CMP_ID := CMP_ID
, SD := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xMsgArch3:=ALARM_SC(
    EV_ID := MSG_EVID_3
    , STATE := xRetSts
    , Q_STATE := MsgACK3);

//Maintenance Managment
TimeTest := (Zone_Test OR Test) AND Activate;
TimeActiv := NOT(Zone_Test) AND NOT(Test) AND Activate;
IF (TimeTest AND NOT(xMTimeTest)) OR (TimeActiv AND NOT(xMTimeActiv)) THEN
    SnREAD_CLK := READ_CLK(CDT:=dtCPU_Time);
    cDate := (BCD_TO_DINT(dtCPU_Time_st.b0)+10);
    cDate := (cDate*365)+(cDate/4);
    CASE (BCD_TO_INT(dtCPU_Time_st.b1)) OF
        1 : cDate:=cDate+0;
        2 : cDate:=cDate+31;
        3 : cDate:=cDate+59;
        4 : cDate:=cDate+90;
        5 : cDate:=cDate+120;
        6 : cDate:=cDate+151;
        7 : cDate:=cDate+181;
        8 : cDate:=cDate+212;
        9 : cDate:=cDate+243;
        10 : cDate:=cDate+273;
        11 : cDate:=cDate+304;
        12 : cDate:=cDate+334;
    END_CASE;
    cDate := cDate+BCD_TO_DINT(dtCPU_Time_st.b2);
    cTime := BCD_TO_DINT(dtCPU_Time_st.b3)*3600000;
    cTime := cTime+BCD_TO_DINT(dtCPU_Time_st.b4)*60000;
    cTime := cTime+BCD_TO_DINT(dtCPU_Time_st.b5)*1000;
    IF TimeTest AND NOT(xMTimeTest) THEN
        Test_D := cDate;
        Test_T := cTime;
    END_IF;
    IF TimeActiv AND NOT(xMTimeActiv) THEN
        Activ_D := cDate;
        Activ_T := cTime;
        Activ_Cyc := Activ_Cyc+1;
    END_IF;
END_IF;

xMTimeTest := TimeTest; // Time Test updated
xMTimeActiv := TimeActiv; // Time Activ updated
IF RstActCycOp THEN // Reset Activate Totalizer
    Activ_Cyc := 0;
    RstActCycOp := false;
END_IF;

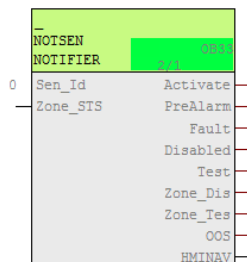
//HMINAV
//Status
xBithMISTS.X0 := Activate;
xBithMISTS.X1 := PreAlarm;
xBithMISTS.X2 := Fault;
xBithMISTS.X3 := false;
//Msg activated
xBithMISTS.X4 := xMsgCond1;
xBithMISTS.X5 := xMsgCond2;
xBithMISTS.X6 := xMsgCond3;
xBithMISTS.X7 := false;
//Msg Acknowled
xBithMISTS.X8 := MsgACK1;
xBithMISTS.X9 := MsgACK2;
xBithMISTS.X10 := MsgACK3;
xBithMISTS.X11 := false;
//Msg in Archive (true if msg in Archive)
xBithMISTS.X12 := (xMsgArch1=0);
xBithMISTS.X13 := (xMsgArch2=0);
xBithMISTS.X14 := (xMsgArch3=0);
xBithMISTS.X15 := false;
HMINAV.Status := xHMISTS;
HMINAV.DBNum := WORD_TO_INT(AuxValue.DBNum);
HMINAV.Btype := 21; //NOTMOD code=21

END_FUNCTION_BLOCK

```

2.14 FB552: NOTSEN

• Representación Gráfica



• Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
Mod_Id	Int	0.0	0	Module Id
EnableOp	Bool	2.0	FALSE	Enable Module
DisableOp	Bool	2.1	FALSE	Disable Module
ActivateOp	Bool	2.2	FALSE	Activate Module Output
DeactivateOp	Bool	2.3	FALSE	Deactivate Module Output
RstActCycOp	Bool	2.4	FALSE	Reset Acitvate Cycles
Zone_STS	Any	4.0		Connect to Panel ZoneX_STS
Sen_Id	Int	0.0	0	Sensor Id
EnableOp	Bool	2.0	FALSE	Enable Sensor
DisableOp	Bool	2.1	FALSE	Disable Sensor
RstActCycOp	Bool	2.2	FALSE	Reset Activate Cycles
Zone_STS	Any	4.0		Connect to Panel ZoneX_STS
OS_Perm	Struct	14.0		Operator Permissions
Bit0	Bool	14.0	TRUE	1 = Operator can Enable Sensor
Bit1	Bool	14.1	TRUE	1 = Operator can Disable Sensor
Bit2	Bool	14.2	TRUE	Reserved
Bit3	Bool	14.3	TRUE	Reserved
Bit4	Bool	14.4	TRUE	1 = Operator can Reset Activated Cycles Totalizer
Bit5	Bool	14.5	TRUE	1 = Operator can Lock Messages
Bit6	Bool	14.6	TRUE	Reserved
Bit7	Bool	14.7	TRUE	Reserved
Bit8	Bool	15.0	TRUE	Reserved
Bit9	Bool	15.1	TRUE	Reserved
Bit10	Bool	15.2	TRUE	Reserved
Bit11	Bool	15.3	TRUE	Reserved
Bit12	Bool	15.4	TRUE	Reserved
Bit13	Bool	15.5	TRUE	Reserved
Bit14	Bool	15.6	TRUE	Reserved
Bit15	Bool	15.7	TRUE	Reserved
Bit16	Bool	16.0	TRUE	Reserved
Bit17	Bool	16.1	TRUE	Reserved
Bit18	Bool	16.2	TRUE	Reserved
Bit19	Bool	16.3	TRUE	Reserved
Bit20	Bool	16.4	TRUE	Reserved
Bit21	Bool	16.5	TRUE	Reserved
Bit22	Bool	16.6	TRUE	Reserved
Bit23	Bool	16.7	TRUE	Reserved
Bit24	Bool	17.0	TRUE	Reserved
Bit25	Bool	17.1	TRUE	Reserved
Bit26	Bool	17.2	TRUE	Reserved
Bit27	Bool	17.3	TRUE	Reserved
Bit28	Bool	17.4	TRUE	Reserved
Bit29	Bool	17.5	TRUE	Reserved
Bit30	Bool	17.6	TRUE	Reserved
Bit31	Bool	17.7	TRUE	Reserved
Feature	Struct	18.0		Status of various features
Bit0	Bool	18.0	TRUE	1 = Lock Messages on Sensor Disabled
Bit1	Bool	18.1	TRUE	1 = Lock Messages on Zone Disabled
Bit2	Bool	18.2	TRUE	1 = Lock Messages on Sensor Test
Bit3	Bool	18.3	TRUE	1 = Lock Messages on Zone Test
Bit4	Bool	18.4	TRUE	1 = Lock Alarm Messages when Zone is out of Alarm
Bit5	Bool	18.5	TRUE	1 = Lock PreAlarm Messages for Zone is out of PreAlarm
Bit6	Bool	18.6	FALSE	Reserved
Bit7	Bool	18.7	FALSE	Reserved
Bit8	Bool	19.0	FALSE	Reserved
Bit9	Bool	19.1	FALSE	Reserved
Bit10	Bool	19.2	FALSE	Reserved
Bit11	Bool	19.3	FALSE	Reserved

Name	Data Type	Address	Initial Value	Comment
Bit12	Bool	19.4	FALSE	Reserved
Bit13	Bool	19.5	FALSE	Reserved
Bit14	Bool	19.6	FALSE	Reserved
Bit15	Bool	19.7	FALSE	Reserved
MSG_EVID_1	DWord	20.0	DW#16#0	Message 1 ID (Activate)
MSG_EVID_2	DWord	24.0	DW#16#0	Message 2 ID (PreAlarm)
MSG_EVID_3	DWord	28.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	32.0	FALSE	Suppress Message 1
MsgSup2	Bool	32.1	FALSE	Suppress Message 2
MsgSup3	Bool	32.2	FALSE	Suppress Message 3
MsgLock	Bool	32.3	FALSE	Message Lock
OUT				
Activate	Bool	34.0	FALSE	Module Activate
PreAlarm	Bool	34.1	FALSE	Sensor PreAlarm
Fault	Bool	34.2	FALSE	Sensor Fault
Disabled	Bool	34.3	FALSE	Sensor Disabled
Test	Bool	34.4	FALSE	Sensor in Test
Zone_Dis	Bool	34.5	FALSE	Zone Disabled
Zone_Test	Bool	34.6	FALSE	Zone in Test
Zone_Num	Int	36.0	0	Zone Number
OOS	Bool	38.0	FALSE	Out of Service
Test_D	DInt	40.0	L#0	Last Test Date
Test_T	DInt	44.0	L#0	Last Test Time
Activ_D	DInt	48.0	L#0	Last Activate Date
Activ_T	DInt	52.0	L#0	Last Activate Time
Activ_Cyc	DInt	56.0	L#0	Activate Cycles
OS_PermOut	DWord	60.0	DW#16#FFFFFFF	Operator permission: output for OS
OS_PermLog	DWord	64.0	DW#16#FFFFFFF	Operator permission: output for OS
Status1	DWord	68.0	DW#16#0	Status1 Word
HMINAV	Struct	72.0		HMI Panel Interface
Status	Word	72.0	W#16#0	Status Word
DBNum	Int	74.0	0	DB Number
Btype	Int	76.0	0	Block Type
MsgSTS1	Word	78.0	W#16#0	Message 1 Status
MsgACK1	Bool	80.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	82.0	W#16#0	Message 2 Status
MsgACK2	Bool	84.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	86.0	W#16#0	Message 3 Status
MsgACK3	Bool	88.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:=          'y' ;
Scl_GenerateReferenceData :=   'y' ;
Scl_S7ServerActive:=          'y' ;
Scl_CreateObjectCode:=        'y' ;
Scl_OptimizeObjectCode:=      'y' ;
Scl_MonitorArrayLimits:=      'y' ;
Scl_CreateDebugInfo :=        'y' ;
Scl_SetOKFlag:=               'n' ;
Scl_SetMaximumStringLength:=   '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu          BYTE0 in memory
X24 : BOOL;                    // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL;                    // 0.7
// second highest Byte in accu  BYTE1 in memory
X16 : BOOL;                    // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL;                    // 1.7
// third highest Byte in accu   BYTE2 in memory
X8  : BOOL;                    // 2.0
X9  : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL;                    // 2.7
// lowest Byte in accu          BYTE3 in memory
X0  : BOOL;                    // 3.0
X1  : BOOL;
X2  : BOOL;
X3  : BOOL;
X4  : BOOL;
X5  : BOOL;
X6  : BOOL;
X7  : BOOL;                    // 3.7
END_STRUCT
END_TYPE

// Zone Status UDT103
TYPE NTZoneSTS
STRUCT
Alarm      : BOOL;    //Zone in Alarm
PreAlarm   : BOOL;    //Zone in PreAlarm
Fault      : BOOL;    //Zone in Fault
Walk Test  : BOOL;    //Zone in Test
Parti_Disab : BOOL;   //Zone Partial Disabled
Total_Disab : BOOL;   //Zone Total Disabled
END_STRUCT
END_TYPE

// Pointer Struct UDT104
TYPE PointSTR
STRUCT
ID          : BYTE;    // Always B#16#10
TypeCode    : BYTE;    // Pointer Type
Quantity    : WORD;    // Quantity of Data
DB_Num      : WORD;    // Data Block Number
PArea       : DWORD;   // Data Area Pointer
END_STRUCT;
END_TYPE

// HMINAV Link Struct UDT106
TYPE HMINAVSTR
STRUCT
Status      : WORD;    // Status Word
DBNum       : INT;     // DB Number
Btype       : INT;     // Block Type
END_STRUCT;
END_TYPE

```

```

FUNCTION_BLOCK FB552
TITLE = 'NOTIFIER Sensor'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'NOTSEN'
FAMILY: 'NOTIFIER'
KNOW_HOW_PROTECT
{PO_Count := '1';
 S7_m_c := 'true';
 S7_alarm_ui := '1'}

//=====
VAR_INPUT
Sen_Id {
  S7_m_c := 'true';
  S7_visible := 'true';
  S7_edit := 'para';
  S7_link := 'false'} : INT; //Sensor Id
EnableOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Enable Sensor
DisableOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Disable Sensor
RstActCycOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Reset Activate Cycles
Zone_STS {
  S7_edit := 'para';
  BLK_Jump := '1'} : ANY; //Connect to Panel ZoneX_STS

//Operator Permissions
OS_Perm {
  S7_visible:= 'false'} :
  STRUCT
  Bit0 : BOOL := 1; //1 = Operator can Enable Sensor
  Bit1 : BOOL := 1; //1 = Operator can Disable Sensor
  Bit2 : BOOL := 1; //Reserved
  Bit3 : BOOL := 1; //Reserved
  Bit4 : BOOL := 1; //1 = Operator can Reset Activated Cycles Totalizer
  Bit5 : BOOL := 1; //1 = Operator can Lock Messages
  Bit6 : BOOL := 1; //Reserved
  Bit7 : BOOL := 1; //Reserved
  Bit8 : BOOL := 1; //Reserved
  Bit9 : BOOL := 1; //Reserved
  Bit10 : BOOL := 1; //Reserved
  Bit11 : BOOL := 1; //Reserved
  Bit12 : BOOL := 1; //Reserved
  Bit13 : BOOL := 1; //Reserved
  Bit14 : BOOL := 1; //Reserved
  Bit15 : BOOL := 1; //Reserved
  Bit16 : BOOL := 1; //Reserved
  Bit17 : BOOL := 1; //Reserved
  Bit18 : BOOL := 1; //Reserved
  Bit19 : BOOL := 1; //Reserved
  Bit20 : BOOL := 1; //Reserved
  Bit21 : BOOL := 1; //Reserved
  Bit22 : BOOL := 1; //Reserved
  Bit23 : BOOL := 1; //Reserved
  Bit24 : BOOL := 1; //Reserved
  Bit25 : BOOL := 1; //Reserved
  Bit26 : BOOL := 1; //Reserved
  Bit27 : BOOL := 1; //Reserved
  Bit28 : BOOL := 1; //Reserved
  Bit29 : BOOL := 1; //Reserved
  Bit30 : BOOL := 1; //Reserved
  Bit31 : BOOL := 1; //Reserved
  END_STRUCT; //Operator Permissions
dwOS_Perm AT OS_Perm : DWORD;
ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;
//FB Features
Feature {
  S7_visible:= 'false'}:
  STRUCT
  Bit0 : BOOL := 1; //1 = Lock Messages on Sensor Disabled
  Bit1 : BOOL := 1; //1 = Lock Messages on Zone Disabled
  Bit2 : BOOL := 1; //1 = Lock Messages on Sensor Test
  Bit3 : BOOL := 1; //1 = Lock Messages on Zone Test
  Bit4 : BOOL := 1; //1 = Lock Alarm Messages when Zone is out of Alarm
  Bit5 : BOOL := 1; //1 = Lock PreAlarm Messages for Zone is out of PreAlarm
  Bit6 : BOOL; //Reserved

```

```

    Bit7   : BOOL;    //Reserved
    Bit8   : BOOL;    //Reserved
    Bit9   : BOOL;    //Reserved
    Bit10  : BOOL;    //Reserved
    Bit11  : BOOL;    //Reserved
    Bit12  : BOOL;    //Reserved
    Bit13  : BOOL;    //Reserved
    Bit14  : BOOL;    //Reserved
    Bit15  : BOOL;    //Reserved
    END_STRUCT;      // Status of various features
//Messages Configuration
MSG_EVID_1 {
    S7_visible := 'false';
    S7_link    := 'false';
    S7_param   := 'false';
    S7_server  := 'alarm_archiv';
    S7_a_type  := 'alarm_s'} : DWORD:=0; //Message 1 ID (Activate)
MSG_EVID_2 {
    S7_visible := 'false';
    S7_link    := 'false';
    S7_param   := 'false';
    S7_server  := 'alarm_archiv';
    S7_a_type  := 'alarm_s'} : DWORD:=0; //Message 2 ID (PreAlarm)
MSG_EVID_3 {
    S7_visible := 'false';
    S7_link    := 'false';
    S7_param   := 'false';
    S7_server  := 'alarm_archiv';
    S7_a_type  := 'alarm_s'} : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
    S7_visible := 'false';
    S7_m_c     := 'true';
    S7_string_1 := 'activ';
    S7_string_0 := 'inactiv'} : BOOL;    //Suppress Message 1
MsgSup2 {
    S7_visible := 'false';
    S7_m_c     := 'true';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ'} : BOOL;    //Suppress Message 2
MsgSup3 {
    S7_visible := 'false';
    S7_m_c     := 'true';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ'} : BOOL;    //Suppress Message 3
MsgLock {
    Op_Level   := '3';
    S7_m_c     := 'true';
    S7_visible := 'false';
    S7_link    := 'false';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ'} : BOOL;    //Message Lock
END_VAR

//=====
VAR_OUTPUT
Activate {
    S7_dynamic := 'true';
    S7_string_1 := 'Activate'} : BOOL; //Sensor Activate
PreAlarm {
    S7_dynamic := 'true';
    S7_string_1 := 'PreAlarm'} : BOOL; //Sensor PreAlarm
Fault {
    S7_dynamic := 'true';
    S7_string_1 := 'Fault'} : BOOL; //Sensor Fault
Disabled {
    S7_dynamic := 'true';
    S7_string_1 := 'Disabled'} : BOOL; //Sensor Disabled
Test {
    S7_dynamic := 'true';
    S7_string_1 := 'Test'} : BOOL; //Sensor in Test
Zone_Dis {
    S7_dynamic := 'true';
    S7_string_1 := 'Disabled'} : BOOL; //Zone Disabled
Zone_Test {
    S7_dynamic := 'true';
    S7_string_1 := 'Test'} : BOOL; //Zone in Test
Zone_Num {
    S7_m_c     := 'true';
    S7_visible := 'false';
    S7_edit    := 'para'} : INT; //Zone Number
OOS {
    S7_dynamic := 'true';
    S7_string_1 := 'OOS'} : BOOL; //Out of Service
Test_D {
    S7_visible := 'false';
    S7_m_c     := 'true'} : DINT; //Last Test Date
Test_T {

```

```

    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Test Time
Activ_D {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Activate Date
Activ_T {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Last Activate Time
Activ_Cyc {
    S7_visible := 'false';
    S7_m_c := 'true' } : DINT; //Activate Cycles
OS_PermOut {
    S7_visible:= 'false';
    S7_m_c:= 'true' } : DWORD:=16#FFFFFFFF; // Operator permission: output for OS
ArrOS_PermOut AT OS_PermOut : ARRAY[0..3] OF BYTE;
OS_PermLog {
    S7_visible:= 'false';
    S7_m_c:= 'true' } : DWORD:=16#FFFFFFFF; // Operator permission: output for OS
ArrOS_PermLog AT OS_PermLog : ARRAY[0..3] OF BYTE;
Status1 {
    S7_visible:= 'false';
    S7_m_c:= 'true' } : DWORD; //Status1 Word
HMINAV : HMINAVSTR; //HMI Panel Interface

//Messages Configuration
MsgSTS1 {
    S7_visible := 'false' } : WORD; //Message 1 Status
MsgACK1 {
    S7_visible := 'false' } : BOOL; //Message 1 Acknowledge
MsgSTS2 {
    S7_visible := 'false' } : WORD; //Message 2 Status
MsgACK2 {
    S7_visible := 'false' } : BOOL; //Message 2 Acknowledge
MsgSTS3 {
    S7_visible := 'false' } : WORD; //Message 3 Status
MsgACK3 {
    S7_visible := 'false' } : BOOL; //Message 3 Acknowledge
END_VAR

//=====
CONST
ZBAddrCPU := 134; // Zones Byte Base Address
SCBAddrCPU := 674; // Sensor Commands Byte Base Address
SSBAddrCPU := 1134; // Sensor Status Byte Base Address
ZBAddrCP := 124; // Zones Byte Base Address
SCBAddrCP := 632; // Sensor Commands Byte Base Address
SSBAddrCP := 1092; // Sensor Status Byte Base Address
END_CONST

//=====
VAR
FBID : WORD; // Panel type
ZBAddr : INT; // Zones Byte Base Address
SCBAddr : INT; // Sens Commands Byte Base Address
SSBAddr : INT; // Sens Status Byte Base Address

xZonSTS : NTZoneSTS; // Zone Status
xSenSTS : WORD; // Sensor Status
xBitSenSTS AT xSenSTS : SBits16inWORD; // Sensor as Bits
xHMISTS : WORD; // HMI Status
xBitHMISTS AT xHMISTS : SBits16inWORD; // HMI Status as Bits
OpCmd : INT; // Operator Command
xSenCMD : WORD; // Sensor Command
xiSenCMD : INT; // Sensor Command as INT
CMP_ID : DWORD; // Area code
xMsgCond1 : BOOL; // Condition MSG1
xMsgCond2 : BOOL; // Condition MSG2
xMsgCond3 : BOOL; // Condition MSG3
xMsgTrig1 : BOOL; // Trigger MSG1
xMsgTrig2 : BOOL; // Trigger MSG2
xMsgTrig3 : BOOL; // Trigger MSG3
xMsgArch1 : INT; // MSG1 Status
xMsgArch2 : INT; // MSG2 Status
xMsgArch3 : INT; // MSG3 Status
AuxValue: // Msg Aux Value
    STRUCT
        Zone : INT; // Msg Zone
        DBNum : WORD; // Msg DB num
    END_STRUCT;
xRetSts : BOOL; // Msg Status
xSTimeTest : BOOL; // Mem Test Time flag
xSTimeActiv : BOOL; // Mem Activ Time flag
END_VAR

//=====
VAR_TEMP
xAux_Err : INT; // Auxiliar Return Function Error

```

```

xAddr          : DINT;
xFMsgLock      : BOOL;           // MsgLock by Features Bits
xMsg1Zone      : BOOL;           // MsgLock1 by Zone Alarm Status
xMsg2Zone      : BOOL;           // MsgLock2 by Zone PreAlarm Status

// Structures to copy data areas
PPanel         : ANY;           // Pointer to Panel Block
PPan_STR AT PPanel : PointSTR;  // Pointer Structure
PLocal        : ANY;           // Pointer to Local Var
PLoc_STR AT PLocal : PointSTR;  // Pointer Structure

// Variable for Operator Permissions
xdwOS_PermLog  : DWORD; // Temporary variable for the value of OS_PermLog
strxOS_PermLog AT xdwOS_PermLog :
STRUCT
  Bit0         : BOOL; //1 = Operator can Enable Sensor
  Bit1         : BOOL; //1 = Operator can Disable Sensor
  Bit2         : BOOL; //Not used
  Bit3         : BOOL; //Not used
  Bit4         : BOOL; //1 = Operator can Reset Activated Cycle Totalizer
  Bit5         : BOOL; //1 = Operator can Lock Messages
  Bit6         : BOOL; //Not used
  Bit7         : BOOL; //Not used
  Bit8         : BOOL; //Not used
  Bit9         : BOOL; //Not used
  Bit10        : BOOL; //Not used
  Bit11        : BOOL; //Not used
  Bit12        : BOOL; //Not used
  Bit13        : BOOL; //Not used
  Bit14        : BOOL; //Not used
  Bit15        : BOOL; //Not used
  Bit16        : BOOL; //Not used
  Bit17        : BOOL; //Not used
  Bit18        : BOOL; //Not used
  Bit19        : BOOL; //Not used
  Bit20        : BOOL; //Not used
  Bit21        : BOOL; //Not used
  Bit22        : BOOL; //Not used
  Bit23        : BOOL; //Not used
  Bit24        : BOOL; //Not used
  Bit25        : BOOL; //Not used
  Bit26        : BOOL; //Not used
  Bit27        : BOOL; //Not used
  Bit28        : BOOL; //Not used
  Bit29        : BOOL; //Not used
  Bit30        : BOOL; //Not used
  Bit31        : BOOL; //Not used
END_STRUCT;
ArrxOS_PermLog AT xdwOS_PermLog : ARRAY[0..3] OF BYTE;

// Variable for status
dwStatus       : DWORD;
ArrBitStatus AT dwStatus : SBits32inDWORD;

// Variable for Maintenance Time/Date
dtCPU_Time     : DATE_AND_TIME; // Actual CPU time
dtCPU_Time_st AT dtCPU_Time :
STRUCT
  b0 : BYTE;
  b1 : BYTE;
  b2 : BYTE;
  b3 : BYTE;
  b4 : BYTE;
  b5 : BYTE;
  b6 : BYTE;
  b7 : BYTE;
END_STRUCT;
SnREAD_CLK    : INT; // Error information READ_CLK;
cDate         : DINT; // Current Date
cTime        : DINT; // Current Time
TimeTest     : BOOL; // Test Time flag
TimeActiv    : BOOL; // Activate Time flag
END_VAR

//=====
BEGIN
IF Sen_Id>99 THEN Sen_Id:=99; END_IF;
IF Sen_Id<1 THEN Sen_Id:=1; END_IF;

//Panel Type to knows base addresses
PPanel      := Zone_STS; // Pointer to Panel Block
PPan_STR.TypeCode := BYTE#04; // Work with WORDS
PPan_STR.Quantity := WORD#01; // 1 Word of Data
PPan_STR.PArea := DW#16#84000000; // First Word of the DB
PLocal      := FBID;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ZoneX_STS
IF FBID=WORD#01 THEN // ZITONCP

```

```

ZBAddr      := ZBAddrCP;
SCBAddr     := SCBAddrCP;
SSBAddr     := SSBAddrCP;
ELSE
ZBAddr      := ZBAddrCPU;
SCBAddr     := SCBAddrCPU;
SSBAddr     := SSBAddrCPU;
END_IF;

//Zone outputs
PPanel      := Zone_STS;           // Pointer to Panel Block
PPan_STR.TypeCode := BYTE#02;     // Work with BYTES
PPan_STR.Quantity := WORD#01;     // 1 Byte of Data
PLocal      := xZonSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve NTZoneSTS
Zone_Dis   := xZonSTS.Total_Disab; // Zone Disabled
Zone_Test  := xZonSTS.Walk_Test;   // Zone Test
Zone_Num   :=1+((DWORD_TO_INT(PPan_STR.PArea)-ZBAddr*8)/16); // Calculate Zone Number
CMP_ID    := WORD_TO_DWORD(PLoc_STR.DB_Num); // DBNum for Messages
AuxValue.Zone := Zone_Num;
AuxValue.DBNum := PLoc_STR.DB_Num;

//Sensor Outputs
PPan_STR.TypeCode := BYTE#04;     // Work with WORDS
PPan_STR.Quantity := WORD#01;     // 1 Word of Data
xAddr      := (SSBAddr+2*(Sen_Id-1))*8; // Addr in bits
PPan_STR.PArea := DINT_TO_DWORD(xAddr) OR DW#16#84000000;
PLocal      := xSenSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve SensorSTS[x]
Activate   := xBitSenSTS.X0;
PreAlarm   := xBitSenSTS.X1;
Fault      := xBitSenSTS.X2;
Disabled   := xBitSenSTS.X3;
Test       := xBitSenSTS.X4;
OOS        := xBitSenSTS.X15;

// Pointer to SensorCMD[x]
PPan_STR.TypeCode := BYTE#04;     // Work with WORDS
PPan_STR.Quantity := WORD#01;     // 1 Word of Data
xAddr      := (SCBAddr+2*(Sen_Id-1))*8; // Addr in bits
PPan_STR.PArea := DINT_TO_DWORD(xAddr) OR DW#16#84000000;

//Write New command
IF EnableOp THEN OpCmd:=1; END_IF;
IF DisableOp THEN OpCmd:=2; END_IF;
IF OpCmd<>xiSenCMD THEN // New Command
  xSenCMD := INT_TO_WORD(OpCmd); // Value to send
  PLocal := xSenCMD;
  xAux_Err :=BLKMOV(SRCBLK := PLocal, DSTBLK := PPanel); // Write SensorCMD[x]
END_IF;

//Read Current Sensor Command
PLocal := xSenCMD;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve SensorCMD[x]
xiSenCMD := WORD_TO_INT(xSenCMD);

//Update Operator Commands
IF xiSenCMD=0 THEN OpCmd:=0; END_IF; // Reset Command
IF OpCmd<>1 THEN EnableOp:=0; END_IF;
IF OpCmd<>2 THEN DisableOp:=0; END_IF;

//OS Permissives
ArrOS_PermaOut[3] := ArrOS_Perma[0];
ArrOS_PermaOut[2] := ArrOS_Perma[1];
ArrOS_PermaOut[1] := ArrOS_Perma[2];
ArrOS_PermaOut[0] := ArrOS_Perma[3];
IF OOS THEN
  OS_PermaLog := DW#16#30 AND OS_PermaOut; //MsgLock & Reset Totalizer Mask
ELSE
  xdwOS_PermaLog := dwOS_Perma;
  IF NOT(Disabled) THEN strxOS_PermaLog.Bit0 := false; //Enable Sensor Permission
  ELSE strxOS_PermaLog.Bit0 := OS_Perma.Bit0; END_IF;
  IF Disabled THEN strxOS_PermaLog.Bit1 := false; //Disable Sensor Permission
  ELSE strxOS_PermaLog.Bit1 := OS_Perma.Bit1; END_IF;
  IF Activ_Cyc=0 THEN strxOS_PermaLog.Bit4 := false; //Activated Cycle Tot Permission
  ELSE strxOS_PermaLog.Bit4 := OS_Perma.Bit4; END_IF;
  strxOS_PermaLog.Bit5 := OS_Perma.Bit5; //Lock Messages Permission

//OS_PermaLog
ArrOS_PermaLog[3] := ArrxOS_PermaLog[0];
ArrOS_PermaLog[2] := ArrxOS_PermaLog[1];
ArrOS_PermaLog[1] := ArrxOS_PermaLog[2];
ArrOS_PermaLog[0] := ArrxOS_PermaLog[3];
END_IF;

```

```

//Status1 DWord
dwStatus      := DWORD#0;
ArrBitStatus.X0 := Activate;
ArrBitStatus.X1 := PreAlarm;
ArrBitStatus.X2 := Fault;
ArrBitStatus.X3 := Disabled;
ArrBitStatus.X4 := Test;
ArrBitStatus.X5 := OOS;
ArrBitStatus.X6 := Zone_Dis;
ArrBitStatus.X7 := Zone_Test;
ArrBitStatus.X8 := MsgLock;
ArrBitStatus.X9 := MsgSup1;
ArrBitStatus.X10 := MsgSup2;
ArrBitStatus.X11 := MsgSup3;
ArrBitStatus.X12 := xMsgCond1;
ArrBitStatus.X13 := xMsgCond2;
ArrBitStatus.X14 := xMsgCond3;
ArrBitStatus.X15 := 0;
Status1       := dwStatus;

//Alarms Management
xFMsgLock := (Disabled AND Feature.Bit0) OR //Lock Messages on Sensor Disabled
             (Zone_Dis AND Feature.Bit1) OR //Lock Messages on Zone Disabled
             (Test AND Feature.Bit2) OR //Lock Messages on Sensor Test
             (Zone_Test AND Feature.Bit3); //Lock Messages on Zone Test

xMsg1Zone := NOT(xZonSTS.Alarm) AND Feature.Bit4; //Lock Alarm Messages when Zone is out of Alarm
xMsg2Zone := NOT(xZonSTS.PreAlarm) AND Feature.Bit5; //Lock PreAlarm Mess when Z. is out of PreAlarm
xMsgCond1 := Activate AND NOT(MsgSup1 OR MsgLock OR xFMsgLock OR xMsg1Zone);
xMsgCond2 := PreAlarm AND NOT(MsgSup2 OR MsgLock OR xFMsgLock OR xMsg2Zone);
xMsgCond3 := Fault AND NOT(MsgSup3 OR MsgLock OR xFMsgLock);
AuxValue.Zone := Zone_Num;

//Activate MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
    MsgSTS1:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond1
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_1
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xMsgArch1:=ALARM_SC(
    EV_ID := MSG_EVID_1
    ,STATE := xRetSts
    ,Q_STATE := MsgACK1);

//PreAlarm MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
    MsgSTS2:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond2
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_2
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xMsgArch2:=ALARM_SC(
    EV_ID := MSG_EVID_2
    ,STATE := xRetSts
    ,Q_STATE := MsgACK2);

//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
    MsgSTS3:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond3
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_3
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xMsgArch3:=ALARM_SC(
    EV_ID := MSG_EVID_3
    ,STATE := xRetSts
    ,Q_STATE := MsgACK3);

//Maintenance Managment
TimeTest := (Zone_Test OR Test) AND Activate;
TimeActiv := NOT(Zone_Test) AND NOT(Test) AND Activate;
IF (TimeTest AND NOT(xSTimeTest)) OR (TimeActiv AND NOT(xSTimeActiv)) THEN
    SnREAD_CLK := READ_CLK(CDT:=dtCPU_Time);
    cDate := (BCD_TO_DINT(dtCPU_Time_st.b0)+10);
    cDate := (cDate*365)+(cDate/4);
    CASE (BCD_TO_INT(dtCPU_Time_st.b1)) OF
        1 : cDate:=cDate+0;
    
```

```

2 : cDate:=cDate+31;
3 : cDate:=cDate+59;
4 : cDate:=cDate+90;
5 : cDate:=cDate+120;
6 : cDate:=cDate+151;
7 : cDate:=cDate+181;
8 : cDate:=cDate+212;
9 : cDate:=cDate+243;
10: cDate:=cDate+273;
11: cDate:=cDate+304;
12: cDate:=cDate+334;
END_CASE;
cDate := cDate+BCD_TO_DINT(dtCPU_Time_st.b2);
cTime := BCD_TO_DINT(dtCPU_Time_st.b3)*3600000;
cTime := cTime+BCD_TO_DINT(dtCPU_Time_st.b4)*60000;
cTime := cTime+BCD_TO_DINT(dtCPU_Time_st.b5)*1000;
IF TimeTest AND NOT(xSTimeTest) THEN
  Test_D := cDate;
  Test_T := cTime;
END_IF;
IF TimeActiv AND NOT(xSTimeActiv) THEN
  Activ_D := cDate;
  Activ_T := cTime;
  Activ_Cyc := Activ_Cyc+1;
END_IF;

END_IF;
xSTimeTest := TimeTest; // Time Test updated
xSTimeActiv := TimeActiv; // Time Activ updated
IF RstActCycOp THEN // Reset Activate Totalizer
  Activ_Cyc := 0;
  RstActCycOp := false;
END_IF;

//HMINAV
//Status
xBithMISTS.X0 := Activate;
xBithMISTS.X1 := PreAlarm;
xBithMISTS.X2 := Fault;
xBithMISTS.X3 := false;
//Msg activated
xBithMISTS.X4 := xMsgCond1;
xBithMISTS.X5 := xMsgCond2;
xBithMISTS.X6 := xMsgCond3;
xBithMISTS.X7 := false;
//Msg Acknowled
xBithMISTS.X8 := MsgACK1;
xBithMISTS.X9 := MsgACK2;
xBithMISTS.X10 := MsgACK3;
xBithMISTS.X11 := false;
//Msg in Archive (true if msg in Archive)
xBithMISTS.X12 := (xMsgArch1=0);
xBithMISTS.X13 := (xMsgArch2=0);
xBithMISTS.X14 := (xMsgArch3=0);
xBithMISTS.X15 := false;

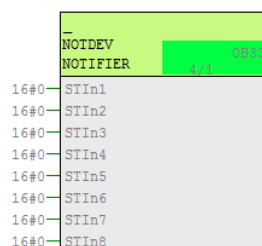
HMINAV.Status := xHMISTS;
HMINAV.DBNum := WORD_TO_INT(AuxValue.DBNum);
HMINAV.Btype := 22; //NOTSEN code=22

END_FUNCTION_BLOCK

```

2.15 FB553: NOTDEV

- Representación Gráfica



- Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
STIn1	DWord	0.0	DW#16#0	Connect to Module/Sensor Status1
STIn2	DWord	4.0	DW#16#0	Connect to Module/Sensor Status1
STIn3	DWord	8.0	DW#16#0	Connect to Module/Sensor Status1
STIn4	DWord	12.0	DW#16#0	Connect to Module/Sensor Status1
STIn5	DWord	16.0	DW#16#0	Connect to Module/Sensor Status1
STIn6	DWord	20.0	DW#16#0	Connect to Module/Sensor Status1
STIn7	DWord	24.0	DW#16#0	Connect to Module/Sensor Status1
STIn8	DWord	28.0	DW#16#0	Connect to Module/Sensor Status1
STIn9	DWord	32.0	DW#16#0	Connect to Module/Sensor Status1
STIn10	DWord	36.0	DW#16#0	Connect to Module/Sensor Status1
STIn11	DWord	40.0	DW#16#0	Connect to Module/Sensor Status1
STIn12	DWord	44.0	DW#16#0	Connect to Module/Sensor Status1
STIn13	DWord	48.0	DW#16#0	Connect to Module/Sensor Status1
STIn14	DWord	52.0	DW#16#0	Connect to Module/Sensor Status1
STIn15	DWord	56.0	DW#16#0	Connect to Module/Sensor Status1
STIn16	DWord	60.0	DW#16#0	Connect to Module/Sensor Status1
OS_Perm	Struct	64.0		Operator Permissions
Bit0	Bool	64.0	TRUE	Reserved
Bit1	Bool	64.1	TRUE	Reserved
Bit2	Bool	64.2	TRUE	Reserved
Bit3	Bool	64.3	TRUE	Reserved
Bit4	Bool	64.4	TRUE	Reserved
Bit5	Bool	64.5	TRUE	1 = Operator can Lock Messages
Bit6	Bool	64.6	TRUE	Reserved
Bit7	Bool	64.7	TRUE	Reserved
Bit8	Bool	65.0	TRUE	Reserved
Bit9	Bool	65.1	TRUE	Reserved
Bit10	Bool	65.2	TRUE	Reserved
Bit11	Bool	65.3	TRUE	Reserved
Bit12	Bool	65.4	TRUE	Reserved
Bit13	Bool	65.5	TRUE	Reserved
Bit14	Bool	65.6	TRUE	Reserved
Bit15	Bool	65.7	TRUE	Reserved
Bit16	Bool	66.0	TRUE	Reserved
Bit17	Bool	66.1	TRUE	Reserved
Bit18	Bool	66.2	TRUE	Reserved
Bit19	Bool	66.3	TRUE	Reserved
Bit20	Bool	66.4	TRUE	Reserved
Bit21	Bool	66.5	TRUE	Reserved
Bit22	Bool	66.6	TRUE	Reserved
Bit23	Bool	66.7	TRUE	Reserved
Bit24	Bool	67.0	TRUE	Reserved
Bit25	Bool	67.1	TRUE	Reserved
Bit26	Bool	67.2	TRUE	Reserved
Bit27	Bool	67.3	TRUE	Reserved
Bit28	Bool	67.4	TRUE	Reserved
Bit29	Bool	67.5	TRUE	Reserved
Bit30	Bool	67.6	TRUE	Reserved
Bit31	Bool	67.7	TRUE	Reserved
Feature	Struct	68.0		Status of various features
Bit0	Bool	68.0	FALSE	1 = Lock Messages for In1
Bit1	Bool	68.1	FALSE	1 = Lock Messages for In2
Bit2	Bool	68.2	FALSE	1 = Lock Messages for In3
Bit3	Bool	68.3	FALSE	1 = Lock Messages for In4
Bit4	Bool	68.4	FALSE	1 = Lock Messages for In5
Bit5	Bool	68.5	FALSE	1 = Lock Messages for In6
Bit6	Bool	68.6	FALSE	1 = Lock Messages for In7
Bit7	Bool	68.7	FALSE	1 = Lock Messages for In8

Name	Data Type	Address	Initial Value	Comment
Bit8	Bool	69.0	FALSE	1 = Lock Messages for In9
Bit9	Bool	69.1	FALSE	1 = Lock Messages for In10
Bit10	Bool	69.2	FALSE	1 = Lock Messages for In11
Bit11	Bool	69.3	FALSE	1 = Lock Messages for In12
Bit12	Bool	69.4	FALSE	1 = Lock Messages for In13
Bit13	Bool	69.5	FALSE	1 = Lock Messages for In14
Bit14	Bool	69.6	FALSE	1 = Lock Messages for In15
Bit15	Bool	69.7	FALSE	1 = Lock Messages for In16
MSG_EVID_1	DWord	70.0	DW#16#0	Message 1 ID (Activate)
MSG_EVID_2	DWord	74.0	DW#16#0	Message 2 ID (PreAlarm)
MSG_EVID_3	DWord	78.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	82.0	FALSE	Suppress Message 1
MsgSup2	Bool	82.1	FALSE	Suppress Message 2
MsgSup3	Bool	82.2	FALSE	Suppress Message 3
MsgLock	Bool	82.3	FALSE	Message Lock
OUT				
OS_PermOut	DWord	84.0	DW#16#FFFFFFF	Operator permission: output for OS
OS_PermLog	DWord	88.0	DW#16#FFFFFFF	Operator permission: output for OS
Status1	DWord	92.0	DW#16#0	Status1 Word
MsgSTS1	Word	96.0	W#16#0	Message 1 Status
MsgACK1	Bool	98.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	100.0	W#16#0	Message 2 Status
MsgACK2	Bool	102.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	104.0	W#16#0	Message 3 Status
MsgACK3	Bool	106.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:= 'y' ;
Scl_GenerateReferenceData := 'y' ;
Scl_S7ServerActive:= 'y' ;
Scl_CreateObjectCode:= 'y' ;
Scl_OptimizeObjectCode:= 'y' ;
Scl_MonitorArrayLimits:= 'y' ;
Scl_CreateDebugInfo := 'y' ;
Scl_SetOKFlag:= 'n' ;
Scl_SetMaximumStringLength:= '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu BYTE0 in memory
X24 : BOOL; // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL; // 0.7
// second highest Byte in accu BYTE1 in memory
X16 : BOOL; // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL; // 1.7
// third highest Byte in accu BYTE2 in memory
X8 : BOOL; // 2.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 2.7
// lowest Byte in accu BYTE3 in memory
X0 : BOOL; // 3.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 3.7
END_STRUCT
END_TYPE

// Pointer Struct UDT104
TYPE PointSTR
STRUCT
ID : BYTE; // Always B#16#10
TypeCode : BYTE; // Pointer Type
Quantity : WORD; // Quantity of Data
DB_Num : WORD; // Data Block Number
PArea : DWORD; // Data Area Pointer
END_STRUCT;
END_TYPE

FUNCTION_BLOCK FB553
TITLE = 'NOTIFIER Device'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'NOTDEV'
FAMILY: 'NOTIFIER'
KNOW_HOW_PROTECT
{PO_Count := '1';
S7_m_c := 'true';
S7_alarm_ui := '1'}

//=====
VAR_INPUT
STIn1 {
S7_m_c := 'true';
S7_edit := 'para';
BLK_Jump := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn1 AT STIn1 : SBits32inDWORD;
STIn2 {
S7_m_c := 'true';
S7_edit := 'para';
BLK_Jump := '1'} : DWORD; //Connect to Module/Sensor Status1

```

```

BSTIn2 AT STIn2          : SBits32inDWORD;
STIn3 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn3 AT STIn3          : SBits32inDWORD;
STIn4 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn4 AT STIn4          : SBits32inDWORD;
STIn5 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn5 AT STIn5          : SBits32inDWORD;
STIn6 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn6 AT STIn6          : SBits32inDWORD;
STIn7 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn7 AT STIn7          : SBits32inDWORD;
STIn8 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1'} : DWORD; //Connect to Module/Sensor Status1
BSTIn8 AT STIn8          : SBits32inDWORD;
STIn9 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn9 AT STIn9          : SBits32inDWORD;
STIn10 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn10 AT STIn10        : SBits32inDWORD;
STIn11 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn11 AT STIn11        : SBits32inDWORD;
STIn12 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn12 AT STIn12        : SBits32inDWORD;
STIn13 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn13 AT STIn13        : SBits32inDWORD;
STIn14 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn14 AT STIn14        : SBits32inDWORD;
STIn15 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn15 AT STIn15        : SBits32inDWORD;
STIn16 {
  S7_m_c    := 'true';
  S7_edit   := 'para';
  BLK_Jump  := '1';
  S7_visible := 'false'} : DWORD; //Connect to Module/Sensor Status1
BSTIn16 AT STIn16        : SBits32inDWORD;

//Operator Permissions
OS_Perm {
  S7_visible := 'false'} :
  STRUCT
  Bit0 : BOOL := 1; //Reserved
  Bit1 : BOOL := 1; //Reserved
  Bit2 : BOOL := 1; //Reserved

```

```

Bit3   : BOOL := 1; //Reserved
Bit4   : BOOL := 1; //Reserved
Bit5   : BOOL := 1; //1 = Operator can Lock Messages
Bit6   : BOOL := 1; //Reserved
Bit7   : BOOL := 1; //Reserved
Bit8   : BOOL := 1; //Reserved
Bit9   : BOOL := 1; //Reserved
Bit10  : BOOL := 1; //Reserved
Bit11  : BOOL := 1; //Reserved
Bit12  : BOOL := 1; //Reserved
Bit13  : BOOL := 1; //Reserved
Bit14  : BOOL := 1; //Reserved
Bit15  : BOOL := 1; //Reserved
Bit16  : BOOL := 1; //Reserved
Bit17  : BOOL := 1; //Reserved
Bit18  : BOOL := 1; //Reserved
Bit19  : BOOL := 1; //Reserved
Bit20  : BOOL := 1; //Reserved
Bit21  : BOOL := 1; //Reserved
Bit22  : BOOL := 1; //Reserved
Bit23  : BOOL := 1; //Reserved
Bit24  : BOOL := 1; //Reserved
Bit25  : BOOL := 1; //Reserved
Bit26  : BOOL := 1; //Reserved
Bit27  : BOOL := 1; //Reserved
Bit28  : BOOL := 1; //Reserved
Bit29  : BOOL := 1; //Reserved
Bit30  : BOOL := 1; //Reserved
Bit31  : BOOL := 1; //Reserved
END_STRUCT; //Operator Permissions
dwOS_Perm AT OS_Perm : DWORD;
ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;
//FB Features
Feature {
  S7_visible:='false';
  S7_edit:='para':
  STRUCT
    Bit0   : BOOL; //1 = Lock Messages for In1
    Bit1   : BOOL; //1 = Lock Messages for In2
    Bit2   : BOOL; //1 = Lock Messages for In3
    Bit3   : BOOL; //1 = Lock Messages for In4
    Bit4   : BOOL; //1 = Lock Messages for In5
    Bit5   : BOOL; //1 = Lock Messages for In6
    Bit6   : BOOL; //1 = Lock Messages for In7
    Bit7   : BOOL; //1 = Lock Messages for In8
    Bit8   : BOOL; //1 = Lock Messages for In9
    Bit9   : BOOL; //1 = Lock Messages for In10
    Bit10  : BOOL; //1 = Lock Messages for In11
    Bit11  : BOOL; //1 = Lock Messages for In12
    Bit12  : BOOL; //1 = Lock Messages for In13
    Bit13  : BOOL; //1 = Lock Messages for In14
    Bit14  : BOOL; //1 = Lock Messages for In15
    Bit15  : BOOL; //1 = Lock Messages for In16
  END_STRUCT; // Status of various features
//Messages Configuration
MSG_EVID_1 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s'} : DWORD:=0; //Message 1 ID (Activate)
MSG_EVID_2 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s'} : DWORD:=0; //Message 2 ID (PreAlarm)
MSG_EVID_3 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s'} : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
  S7_visible := 'false';
  S7_m_c := 'true';
  S7_string_1 := 'activ';
  S7_string_0 := 'inactiv'} : BOOL; //Suppress Message 1
MsgSup2 {
  S7_visible := 'false';
  S7_m_c := 'true';
  S7_string_0 := 'inactiv';
  S7_string_1 := 'activ'} : BOOL; //Suppress Message 2
MsgSup3 {
  S7_visible := 'false';
  S7_m_c := 'true';
  S7_string_0 := 'inactiv';

```

```

    S7_string_1 := 'activ' } : BOOL; //Suppress Message 3
MsgLock {
    Op_Level := '3';
    S7_m_c := 'true';
    S7_visible := 'false';
    S7_link := 'false';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ' } : BOOL; //Message Lock
END_VAR

//=====
VAR_OUTPUT
OS_PermOut {
    S7_visible:= 'false';
    S7_m_c:= 'true' } : DWORD:=16#FFFFFFFF; // Operator permission: output for OS
ArrOS_PermOut AT OS_PermOut : ARRAY[0..3] OF BYTE;
OS_PermLog {
    S7_visible:= 'false';
    S7_m_c:= 'true' } : DWORD:=16#FFFFFFFF; // Operator permission: output for OS
ArrOS_PermLog AT OS_PermLog : ARRAY[0..3] OF BYTE;
Status1 {
    S7_visible:= 'false';
    S7_m_c:= 'true' } : DWORD; //Status1 Word
//Messages Configuration
MsgSTS1 {
    S7_visible := 'false' } : WORD; //Message 1 Status
MsgACK1 {
    S7_visible := 'false' } : BOOL; //Message 1 Acknowledge
MsgSTS2 {
    S7_visible := 'false' } : WORD; //Message 2 Status
MsgACK2 {
    S7_visible := 'false' } : BOOL; //Message 2 Acknowledge
MsgSTS3 {
    S7_visible := 'false' } : WORD; //Message 3 Status
MsgACK3 {
    S7_visible := 'false' } : BOOL; //Message 3 Acknowledge
END_VAR

//=====
VAR
// Variable for Messages
CMP_ID : DWORD; // Area code
xMsgCond1 : BOOL; // Condition MSG1
xMsgCond2 : BOOL; // Condition MSG2
xMsgCond3 : BOOL; // Condition MSG3
xMsgTrig1 : BOOL; // Trigger MSG1
xMsgTrig2 : BOOL; // Trigger MSG2
xMsgTrig3 : BOOL; // Trigger MSG3
xRetVal : INT;
xRetSts : BOOL; // Msg Status
AuxValue:
    STRUCT
        Value : INT; // Msg Aux Values
    END_STRUCT;
END_VAR

//=====
VAR_TEMP
// Variable for Operator Permissions
xdwOS_PermLog : DWORD; // Temporary variable for the value of OS_PermLog
strxOS_PermLog AT xdwOS_PermLog :
    STRUCT
        Bit0 : BOOL; //Not used
        Bit1 : BOOL; //Not used
        Bit2 : BOOL; //Not used
        Bit3 : BOOL; //Not used
        Bit4 : BOOL; //Not used
        Bit5 : BOOL; //1 = Operator can Lock Messages
        Bit6 : BOOL; //Not used
        Bit7 : BOOL; //Not used
        Bit8 : BOOL; //Not used
        Bit9 : BOOL; //Not used
        Bit10 : BOOL; //Not used
        Bit11 : BOOL; //Not used
        Bit12 : BOOL; //Not used
        Bit13 : BOOL; //Not used
        Bit14 : BOOL; //Not used
        Bit15 : BOOL; //Not used
        Bit16 : BOOL; //Not used
        Bit17 : BOOL; //Not used
        Bit18 : BOOL; //Not used
        Bit19 : BOOL; //Not used
        Bit20 : BOOL; //Not used
        Bit21 : BOOL; //Not used
        Bit22 : BOOL; //Not used
        Bit23 : BOOL; //Not used
        Bit24 : BOOL; //Not used
    END_STRUCT;

```

```

Bit25      : BOOL; //Not used
Bit26      : BOOL; //Not used
Bit27      : BOOL; //Not used
Bit28      : BOOL; //Not used
Bit29      : BOOL; //Not used
Bit30      : BOOL; //Not used
Bit31      : BOOL; //Not used
END_STRUCT;
ArrxOS_PermLog AT xdwOS_PermLog : ARRAY[0..3] OF BYTE; // Variable for status
dwStatus    : DWORD;
ArrBitStatus AT dwStatus      : SBits32inDWORD;

//Local pointer
PLocal      : ANY; // Pointer to Local Var
PLoc_STR AT PLocal : PointSTR; // Pointer Structure
END_VAR

//=====
BEGIN
//OS Permissives
ArrOS_PermOut[3] := ArrOS_Perm[0];
ArrOS_PermOut[2] := ArrOS_Perm[1];
ArrOS_PermOut[1] := ArrOS_Perm[2];
ArrOS_PermOut[0] := ArrOS_Perm[3];
xdwOS_PermLog := dwOS_Perm;
strxOS_PermLog.Bit5 := OS_Perm.Bit5; //Lock Messages Permission

//OS_PermLog
ArrOS_PermLog[3] := ArrxOS_PermLog[0];
ArrOS_PermLog[2] := ArrxOS_PermLog[1];
ArrOS_PermLog[1] := ArrxOS_PermLog[2];
ArrOS_PermLog[0] := ArrxOS_PermLog[3];

//Status1 DWord
//| X0=Activate | X1=PreAlarm | X2=Fault | X3=Disabled |
//| X4=Test | X5=OOS | X6=Zone_Dis | X7=Zone_Test |
//| X8= MsgLock | X9=MsgSup1 | X10=MsgSup2 | X11=MsgSup3 |
//| X12=xMsgCondi | X13=xMsgCond2 | X14=xMsgCond3 | X15=LocMsgLck |
dwStatus := DWORD#0;
ArrBitStatus.X0 := BSTIn1.X0 OR BSTIn2.X0 OR BSTIn3.X0 OR BSTIn4.X0 OR BSTIn5.X0 OR BSTIn6.X0 OR
BSTIn7.X0 OR BSTIn8.X0 OR BSTIn9.X0 OR BSTIn10.X0 OR BSTIn11.X0 OR BSTIn12.X0 OR
BSTIn13.X0 OR BSTIn14.X0 OR BSTIn15.X0 OR BSTIn16.X0;
ArrBitStatus.X1 := BSTIn1.X1 OR BSTIn2.X1 OR BSTIn3.X1 OR BSTIn4.X1 OR BSTIn5.X1 OR BSTIn6.X1 OR
BSTIn7.X1 OR BSTIn8.X1 OR BSTIn9.X1 OR BSTIn10.X1 OR BSTIn11.X1 OR BSTIn12.X1 OR
BSTIn13.X1 OR BSTIn14.X1 OR BSTIn15.X1 OR BSTIn16.X1;
ArrBitStatus.X2 := BSTIn1.X2 OR BSTIn2.X2 OR BSTIn3.X2 OR BSTIn4.X2 OR BSTIn5.X2 OR BSTIn6.X2 OR
BSTIn7.X2 OR BSTIn8.X2 OR BSTIn9.X2 OR BSTIn10.X2 OR BSTIn11.X2 OR BSTIn12.X2 OR
BSTIn13.X2 OR BSTIn14.X2 OR BSTIn15.X2 OR BSTIn16.X2;
ArrBitStatus.X3 := BSTIn1.X3 OR BSTIn2.X3 OR BSTIn3.X3 OR BSTIn4.X3 OR BSTIn5.X3 OR BSTIn6.X3 OR
BSTIn7.X3 OR BSTIn8.X3 OR BSTIn9.X3 OR BSTIn10.X3 OR BSTIn11.X3 OR BSTIn12.X3 OR
BSTIn13.X3 OR BSTIn14.X3 OR BSTIn15.X3 OR BSTIn16.X3;
ArrBitStatus.X4 := BSTIn1.X4 OR BSTIn2.X4 OR BSTIn3.X4 OR BSTIn4.X4 OR BSTIn5.X4 OR BSTIn6.X4 OR
BSTIn7.X4 OR BSTIn8.X4 OR BSTIn9.X4 OR BSTIn10.X4 OR BSTIn11.X4 OR BSTIn12.X4 OR
BSTIn13.X4 OR BSTIn14.X4 OR BSTIn15.X4 OR BSTIn16.X4;
ArrBitStatus.X5 := BSTIn1.X5 OR BSTIn2.X5 OR BSTIn3.X5 OR BSTIn4.X5 OR BSTIn5.X5 OR BSTIn6.X5 OR
BSTIn7.X5 OR BSTIn8.X5 OR BSTIn9.X5 OR BSTIn10.X5 OR BSTIn11.X5 OR BSTIn12.X5 OR
BSTIn13.X5 OR BSTIn14.X5 OR BSTIn15.X5 OR BSTIn16.X5;
ArrBitStatus.X6 := BSTIn1.X6 OR BSTIn2.X6 OR BSTIn3.X6 OR BSTIn4.X6 OR BSTIn5.X6 OR BSTIn6.X6 OR
BSTIn7.X6 OR BSTIn8.X6 OR BSTIn9.X6 OR BSTIn10.X6 OR BSTIn11.X6 OR BSTIn12.X6 OR
BSTIn13.X6 OR BSTIn14.X6 OR BSTIn15.X6 OR BSTIn16.X6;
ArrBitStatus.X7 := BSTIn1.X7 OR BSTIn2.X7 OR BSTIn3.X7 OR BSTIn4.X7 OR BSTIn5.X7 OR BSTIn6.X7 OR
BSTIn7.X7 OR BSTIn8.X7 OR BSTIn9.X7 OR BSTIn10.X7 OR BSTIn11.X7 OR BSTIn12.X7 OR
BSTIn13.X7 OR BSTIn14.X7 OR BSTIn15.X7 OR BSTIn16.X7;
ArrBitStatus.X8 := BSTIn1.X8 OR BSTIn2.X8 OR BSTIn3.X8 OR BSTIn4.X8 OR BSTIn5.X8 OR BSTIn6.X8 OR
BSTIn7.X8 OR BSTIn8.X8 OR BSTIn9.X8 OR BSTIn10.X8 OR BSTIn11.X8 OR BSTIn12.X8 OR
BSTIn13.X8 OR BSTIn14.X8 OR BSTIn15.X8 OR BSTIn16.X8 OR MsgLock;
ArrBitStatus.X9 := BSTIn1.X9 OR BSTIn2.X9 OR BSTIn3.X9 OR BSTIn4.X9 OR BSTIn5.X9 OR BSTIn6.X9 OR
BSTIn7.X9 OR BSTIn8.X9 OR BSTIn9.X9 OR BSTIn10.X9 OR BSTIn11.X9 OR BSTIn12.X9 OR
BSTIn13.X9 OR BSTIn14.X9 OR BSTIn15.X9 OR BSTIn16.X9 OR MsgSup1;
ArrBitStatus.X10 := BSTIn1.X10 OR BSTIn2.X10 OR BSTIn3.X10 OR BSTIn4.X10 OR BSTIn5.X10 OR
BSTIn6.X10 OR BSTIn7.X10 OR BSTIn8.X10 OR BSTIn9.X10 OR BSTIn10.X10 OR
BSTIn11.X10 OR BSTIn12.X10 OR BSTIn13.X10 OR BSTIn14.X10 OR BSTIn15.X10 OR
BSTIn16.X10 OR MsgSup2;
ArrBitStatus.X11 := BSTIn1.X11 OR BSTIn2.X11 OR BSTIn3.X11 OR BSTIn4.X11 OR BSTIn5.X11 OR
BSTIn6.X11 OR BSTIn7.X11 OR BSTIn8.X11 OR BSTIn9.X11 OR BSTIn10.X11 OR
BSTIn11.X11 OR BSTIn12.X11 OR BSTIn13.X11 OR BSTIn14.X11 OR BSTIn15.X11 OR
BSTIn16.X11 OR MsgSup3;
ArrBitStatus.X12 := (BSTIn1.X12 AND NOT(Feature.Bit0) ) OR (BSTIn2.X12 AND NOT(Feature.Bit1) ) OR
(BSTIn3.X12 AND NOT(Feature.Bit2) ) OR (BSTIn4.X12 AND NOT(Feature.Bit3) ) OR
(BSTIn5.X12 AND NOT(Feature.Bit4) ) OR (BSTIn6.X12 AND NOT(Feature.Bit5) ) OR
(BSTIn7.X12 AND NOT(Feature.Bit6) ) OR (BSTIn8.X12 AND NOT(Feature.Bit7) ) OR
(BSTIn9.X12 AND NOT(Feature.Bit8) ) OR (BSTIn10.X12 AND NOT(Feature.Bit9) ) OR
(BSTIn11.X12 AND NOT(Feature.Bit10) ) OR (BSTIn12.X12 AND NOT(Feature.Bit11) ) OR
(BSTIn13.X12 AND NOT(Feature.Bit12) ) OR (BSTIn14.X12 AND NOT(Feature.Bit13) ) OR
(BSTIn15.X12 AND NOT(Feature.Bit14) ) OR (BSTIn15.X12 AND NOT(Feature.Bit15));
ArrBitStatus.X13 := (BSTIn1.X13 AND NOT(Feature.Bit0) ) OR (BSTIn2.X13 AND NOT(Feature.Bit1) ) OR
(BSTIn3.X13 AND NOT(Feature.Bit2) ) OR (BSTIn4.X13 AND NOT(Feature.Bit3) ) OR

```

```

(BSTIn5.X13 AND NOT (Feature.Bit4) ) OR (BSTIn6.X13 AND NOT (Feature.Bit5) ) OR
(BSTIn7.X13 AND NOT (Feature.Bit6) ) OR (BSTIn8.X13 AND NOT (Feature.Bit7) ) OR
(BSTIn9.X13 AND NOT (Feature.Bit8) ) OR (BSTIn10.X13 AND NOT (Feature.Bit9) ) OR
(BSTIn11.X13 AND NOT (Feature.Bit10)) OR (BSTIn12.X13 AND NOT (Feature.Bit11)) OR
(BSTIn13.X13 AND NOT (Feature.Bit12)) OR (BSTIn14.X13 AND NOT (Feature.Bit13)) OR
(BSTIn15.X13 AND NOT (Feature.Bit14)) OR (BSTIn15.X13 AND NOT (Feature.Bit15));
ArrBitStatus.X14 := (BSTIn1.X14 AND NOT (Feature.Bit0) ) OR (BSTIn2.X14 AND NOT (Feature.Bit1) ) OR
(BSTIn3.X14 AND NOT (Feature.Bit2) ) OR (BSTIn4.X14 AND NOT (Feature.Bit3) ) OR
(BSTIn5.X14 AND NOT (Feature.Bit4) ) OR (BSTIn6.X14 AND NOT (Feature.Bit5) ) OR
(BSTIn7.X14 AND NOT (Feature.Bit6) ) OR (BSTIn8.X14 AND NOT (Feature.Bit7) ) OR
(BSTIn9.X14 AND NOT (Feature.Bit8) ) OR (BSTIn10.X14 AND NOT (Feature.Bit9) ) OR
(BSTIn11.X14 AND NOT (Feature.Bit10)) OR (BSTIn12.X14 AND NOT (Feature.Bit11)) OR
(BSTIn13.X14 AND NOT (Feature.Bit12)) OR (BSTIn14.X14 AND NOT (Feature.Bit13)) OR
(BSTIn15.X14 AND NOT (Feature.Bit14)) OR (BSTIn15.X14 AND NOT (Feature.Bit15));
ArrBitStatus.X15 := MsgLock;
Status1 := dwStatus;

//Alarms Managment
PLocal := CMP_ID;
CMP_ID := WORD_TO_DWORD(PLoc_STR.DB_Num); // DBNum for Messages
xMsgCond1 := ArrBitStatus.X12 AND NOT (MsgSup1 OR MsgLock);
xMsgCond2 := ArrBitStatus.X13 AND NOT (MsgSup2 OR MsgLock);
xMsgCond3 := ArrBitStatus.X14 AND NOT (MsgSup3 OR MsgLock);

//Activate MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
  MsgSTS1:=INT_TO_WORD(ALARM_DQ(
    SIG := xMsgCond1
    ,ID := W#16#EEEE
    ,EV_ID := MSG_EVID_1
    ,CMP_ID := CMP_ID
    ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xRetVal:=ALARM_SC(
  EV_ID := MSG_EVID_1
  ,STATE := xRetSts
  ,Q_STATE := MsgACK1);

//PreAlarm MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
  MsgSTS2:=INT_TO_WORD(ALARM_DQ(
    SIG := xMsgCond2
    ,ID := W#16#EEEE
    ,EV_ID := MSG_EVID_2
    ,CMP_ID := CMP_ID
    ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xRetVal:=ALARM_SC(
  EV_ID := MSG_EVID_2
  ,STATE := xRetSts
  ,Q_STATE := MsgACK2);

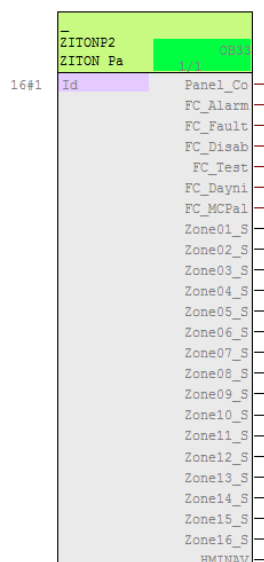
//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
  MsgSTS3:=INT_TO_WORD(ALARM_DQ(
    SIG := xMsgCond3
    ,ID := W#16#EEEE
    ,EV_ID := MSG_EVID_3
    ,CMP_ID := CMP_ID
    ,SD := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xRetVal:=ALARM_SC(
  EV_ID := MSG_EVID_3
  ,STATE := xRetSts
  ,Q_STATE := MsgACK3);

END_FUNCTION_BLOCK

```

2.16 FB554: ZITONP2

- Representación Gráfica



- Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
FBID	Word	0.0	W#16#0	Block ID =0 (DO NOT CHANGE)
Id	Word	2.0	W#16#1	Connection ID (W#16#01..W#16#0FFF)
Dev_ID	Byte	4.0	B#16#2	Communication interface (See TCON Block Help)
IP_Address	Struct	6.0		Master IP Address
Octet_1	Int	6.0	192	1th Octet (0..255)
Octet_2	Int	8.0	168	2nd Octet (0..255)
Octet_3	Int	10.0	0	3th Octet (0..255)
Octet_4	Int	12.0	1	4th Octet (0..255)
Port	Int	14.0	502	Master Port
Panel_Conn_Time	Real	16.0	5	Delay time Panel connection Error [s]
Mod_cycle	Int	20.0	4	Modules scanned per cycle (4..120)
EnaComOp	Bool	22.0	FALSE	Operator Enable comms
Zones	Int	24.0	1	Number of Zones configured (1..32)
Loop1_Mod	Int	26.0	1	Last Module configured for Loop1 (1..127)
Loop2_Mod	Int	28.0	0	Last Module configured for Loop2 (0..127)
Loop3_Mod	Int	30.0	0	Last Module configured for Loop3 (0..127)
Loop4_Mod	Int	32.0	0	Last Module configured for Loop4 (0..127)
OS_Perm	Struct	34.0		Operator Permissions
Bit0	Bool	34.0	TRUE	Reserved
Bit1	Bool	34.1	TRUE	Reserved
Bit2	Bool	34.2	TRUE	Reserved
Bit3	Bool	34.3	TRUE	Reserved
Bit4	Bool	34.4	TRUE	Reserved
Bit5	Bool	34.5	TRUE	Reserved
Bit6	Bool	34.6	TRUE	Reserved
Bit7	Bool	34.7	TRUE	Reserved
Bit8	Bool	35.0	TRUE	Reserved
Bit9	Bool	35.1	TRUE	Reserved
Bit10	Bool	35.2	TRUE	Reserved
Bit11	Bool	35.3	TRUE	Reserved
Bit12	Bool	35.4	TRUE	Reserved
Bit13	Bool	35.5	TRUE	Reserved
Bit14	Bool	35.6	TRUE	Reserved
Bit15	Bool	35.7	TRUE	Reserved
Bit16	Bool	36.0	TRUE	Reserved
Bit17	Bool	36.1	TRUE	Reserved
Bit18	Bool	36.2	TRUE	Reserved
Bit19	Bool	36.3	TRUE	Reserved
Bit20	Bool	36.4	TRUE	Reserved
Bit21	Bool	36.5	TRUE	Reserved
Bit22	Bool	36.6	TRUE	Reserved
Bit23	Bool	36.7	TRUE	Reserved
Bit24	Bool	37.0	TRUE	Reserved
Bit25	Bool	37.1	TRUE	Reserved

Name	Data Type	Address	Initial Value	Comment
Bit26	Bool	37.2	TRUE	Reserved
Bit27	Bool	37.3	TRUE	Reserved
Bit28	Bool	37.4	TRUE	Reserved
Bit29	Bool	37.5	TRUE	Reserved
Bit30	Bool	37.6	TRUE	1 = Operator can Lock Messages
Bit31	Bool	37.7	TRUE	1 = Operator can Enable/Disable Comms
SampleTime	Real	38.0	1	Sampling time [s]
MSG_EVID_1	DWord	42.0	DW#16#0	Message 1 ID (Alarm)
MSG_EVID_2	DWord	46.0	DW#16#0	Message 2 ID (MCP Alarm)
MSG_EVID_3	DWord	50.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	54.0	FALSE	Suppress Message 1
MsgSup2	Bool	54.1	FALSE	Suppress Message 2
MsgSup3	Bool	54.2	FALSE	Suppress Message 3
MsgLock	Bool	54.3	FALSE	Message Lock
OUT				
Func_Error	String[8]	56.0	"	Internal Function that Generated the Error
Func_Err_Num	Word	66.0	W#16#0	Function Error number
Panel_Con_Err	Bool	68.0	FALSE	Panel Connection Error
FC_Alarm	Bool	68.1	FALSE	Alarm Funcional Condition
FC_Fault	Bool	68.2	FALSE	Fault Funcional Condition
FC_Disable	Bool	68.3	FALSE	Disable Funcional Condition
FC_Test	Bool	68.4	FALSE	Test Funcional Condition
FC_Daynight	Bool	68.5	FALSE	Daynight Funcional Condition
FC_MCPalm	Bool	68.6	FALSE	Manual Call Points Alarm Funcional Condition
S_Delay	Bool	68.7	FALSE	Sounders Delay Enabled
S_DelayAct	Bool	69.0	FALSE	Sounders Activation Delay in progress
S_OutAct	Bool	69.1	FALSE	Sounders Outputs Activated
S_OutSil	Bool	69.2	FALSE	Sounders Outputs Silenced
S_Disable	Bool	69.3	FALSE	Sounders Outputs Disabled
S_OvrTime	Bool	69.4	FALSE	Sounders Override Time Elapsed
S_Test	Bool	69.5	FALSE	Sounders on Test
FB_Delay	Bool	69.6	FALSE	Fire Brigade: Delay Enabled
FB_DelayAct	Bool	69.7	FALSE	Fire Brigade: Activation Delay in progress
FB_OutAct	Bool	70.0	FALSE	Fire Brigade: Outputs Activated
FB_Ack	Bool	70.1	FALSE	Fire Brigade: Acknowledged
FB_Disable	Bool	70.2	FALSE	Fire Brigade: Disabled
FB_Test	Bool	70.3	FALSE	Fire Brigade: On Test
FP_Delay	Bool	70.4	FALSE	Fire Protection: Delay Enabled
FP_DelayAct	Bool	70.5	FALSE	Fire Protection: Activation Delay in progress
FP_OutAct	Bool	70.6	FALSE	Fire Protection: Outputs Activated
FP_Ack	Bool	70.7	FALSE	Fire Protection: Acknowledged
FP_Disable	Bool	71.0	FALSE	Fire Protection: Disabled
FP_Test	Bool	71.1	FALSE	Fire Protection: On Test
Zone01_STS	Struct	72.0		Zone 01 Status
PreAlarm	Bool	72.0	FALSE	Zone in PreAlarm
Alarm	Bool	72.1	FALSE	Zone in Alarm
Fault	Bool	72.2	FALSE	Zone in Fault
Test	Bool	72.3	FALSE	Zone in Test
Disabled	Bool	72.4	FALSE	Zone Disabled
...				
Zone32_STS	Struct	134.0		Zone 32 Status
PreAlarm	Bool	134.0	FALSE	Zone in PreAlarm
Alarm	Bool	134.1	FALSE	Zone in Alarm
Fault	Bool	134.2	FALSE	Zone in Fault
Test	Bool	134.3	FALSE	Zone in Test
Disabled	Bool	134.4	FALSE	Zone Disabled
Status1	DWord	136.0	DW#16#0	Status1 Word
ZStatus1	DWord	140.0	DW#16#0	Status Z1-4 Word
ZStatus2	DWord	144.0	DW#16#0	Status Z5-8 Word
ZStatus3	DWord	148.0	DW#16#0	Status Z9-12 Word
ZStatus4	DWord	152.0	DW#16#0	Status Z13-16 Word
ZStatus5	DWord	156.0	DW#16#0	Status Z17-20 Word
ZStatus6	DWord	160.0	DW#16#0	Status Z21-24 Word
ZStatus7	DWord	164.0	DW#16#0	Status Z25-28 Word
ZStatus8	DWord	168.0	DW#16#0	Status Z29-32 Word
OS_PermlLog	DWord	172.0	DW#16#FFFFFFF	Operator Panel permissions: output for OS
HMINAV	Struct	176.0		HMI Panel Interface
Status	Word	176.0	W#16#0	Status Word
DBNum	Int	178.0	0	DB Number
Btype	Int	180.0	0	Block Type
MsgSTS1	Word	182.0	W#16#0	Message 1 Status
MsgACK1	Bool	184.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	186.0	W#16#0	Message 2 Status
MsgACK2	Bool	188.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	190.0	W#16#0	Message 3 Status
MsgACK3	Bool	192.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:= 'y' ;
Scl_GenerateReferenceData := 'y' ;
Scl_S7ServerActive:= 'y' ;
Scl_CreateObjectCode:= 'y' ;
Scl_OptimizeObjectCode:= 'y' ;
Scl_MonitorArrayLimits:= 'y' ;
Scl_CreateDebugInfo := 'y' ;
Scl_SetOKFlag:= 'n' ;
Scl_SetMaximumStringLength:= '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu BYTE0 in memory
X24 : BOOL; // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL; // 0.7
// second highest Byte in accu BYTE1 in memory
X16 : BOOL; // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL; // 1.7
// third highest Byte in accu BYTE2 in memory
X8 : BOOL; // 2.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 2.7
// lowest Byte in accu BYTE3 in memory
X0 : BOOL; // 3.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 3.7
END_STRUCT
END_TYPE

// SBits32inDWORD UDT101
TYPE SBits16inWORD
STRUCT
// Highest Byte in accu BYTE1 in memory
X8 : BOOL; // 0.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 0.7
// Lowest Byte in accu BYTE2 in memory
X0 : BOOL; // 1.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 1.7
END_STRUCT
END_TYPE
// Pointer Struct UDT104
TYPE PointSTR
STRUCT
ID : BYTE; //Always B#16#10
TypeCode : BYTE; //Pointer Type
Quantity : WORD; //Quantity of Data
DB_Num : WORD; //Data Block Number
PArea : DWORD; //Data Area Pointer
END_STRUCT;
END_TYPE

```

```

// Zone Status UDT105
TYPE ZTZoneSTS
  STRUCT
    PreAlarm      : BOOL;    //Zone in PreAlarm
    Alarm         : BOOL;    //Zone in Alarm
    Fault         : BOOL;    //Zone in Fault
    Test          : BOOL;    //Zone in Test
    Disabled      : BOOL;    //Zone Disabled
  END_STRUCT
END_TYPE

// HMINAV Link Struct UDT106
TYPE HMINAVSTR
  STRUCT
    Status        : WORD;    // Status Word
    DBNum         : INT;     // DB Number
    Btype         : INT;     // Block Type
  END_STRUCT;
END_TYPE

//=====
FUNCTION_BLOCK FB554
  TITLE = 'ZITON ZP2'
  VERSION: '09.00'
  AUTHOR: 'AdvLibFP'
  NAME: 'ZITONP2'ç

  FAMILY: 'ZITON'
  KNOW_HOW_PROTECT
  {S7_tasklist := 'OB100';
  PO_Count    := '1';
  S7_m_c      := 'true';
  S7_alarm_ui := '1'}

  VAR_INPUT
  //Connection VAR
  FBID {
    S7_visible := 'false';
    S7_link    := 'false' : WORD := 16#000; //Block ID =0 (DO NOT CHANGE)
  Id {
    S7_dynamic := 'true';
    S7_edit    := 'para';
    S7_link    := 'false' : WORD := W#16#01; //Connection ID (W#16#01..W#16#0FFF)
  Dev_ID {
    S7_visible := 'false';
    S7_link    := 'false' : BYTE := B#16#02; //Communication interface (See TCON Block Help)
  IP_Address {
    S7_visible := 'false';
    S7_xedit   := 'Octet_1,para;Octet_2,para;Octet_3,para;Octet_4,para;';
    S7_link    := 'false' :
    STRUCT
      Octet_1 : INT := 192; //1th Octet (0..255)
      Octet_2 : INT := 168; //2nd Octet (0..255)
      Octet_3 : INT := 0; //3th Octet (0..255)
      Octet_4 : INT := 1; //4th Octet (0..255)
    END_STRUCT;
    BIPAddr AT IP_Address : ARRAY[1..8] OF BYTE;
  Port {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 502; //Master Port
  BPort AT Port : ARRAY[1..2] OF BYTE;
  Panel_Conn_Time {
    S7_visible := 'false';
    S7_link    := 'false' : REAL := 5; //Delay time Panel connection Error [s]
  Mod_cycle {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 4; //Modules scanned per cycle (4..120)
  EnaComOp {
    Op_Level := '3';
    S7_m_c   := 'true';
    S7_visible := 'false';
    S7_link    := 'false' : BOOL; //Operator Enable comms
  Zones {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 1; //Number of Zones configured (1..32)
  Loop1_Mod {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 1; //Last Module configured for Loop1 (1..127)
  Loop2_Mod {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 0; //Last Module configured for Loop2 (0..127)
  Loop3_Mod {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 0; //Last Module configured for Loop3 (0..127)
  Loop4_Mod {
    S7_visible := 'false';
    S7_link    := 'false' : INT := 0; //Last Module configured for Loop4 (0..127)

```

```

OS_Perm {
    S7_visible:='false'      :
    STRUCT
    Bit0   : BOOL := 1; //Reserved
    Bit1   : BOOL := 1; //Reserved
    Bit2   : BOOL := 1; //Reserved
    Bit3   : BOOL := 1; //Reserved
    Bit4   : BOOL := 1; //Reserved
    Bit5   : BOOL := 1; //Reserved
    Bit6   : BOOL := 1; //Reserved
    Bit7   : BOOL := 1; //Reserved
    Bit8   : BOOL := 1; //Reserved
    Bit9   : BOOL := 1; //Reserved
    Bit10  : BOOL := 1; //Reserved
    Bit11  : BOOL := 1; //Reserved
    Bit12  : BOOL := 1; //Reserved
    Bit13  : BOOL := 1; //Reserved
    Bit14  : BOOL := 1; //Reserved
    Bit15  : BOOL := 1; //Reserved
    Bit16  : BOOL := 1; //Reserved
    Bit17  : BOOL := 1; //Reserved
    Bit18  : BOOL := 1; //Reserved
    Bit19  : BOOL := 1; //Reserved
    Bit20  : BOOL := 1; //Reserved
    Bit21  : BOOL := 1; //Reserved
    Bit22  : BOOL := 1; //Reserved
    Bit23  : BOOL := 1; //Reserved
    Bit24  : BOOL := 1; //Reserved
    Bit25  : BOOL := 1; //Reserved
    Bit26  : BOOL := 1; //Reserved
    Bit27  : BOOL := 1; //Reserved
    Bit28  : BOOL := 1; //Reserved
    Bit29  : BOOL := 1; //Reserved
    Bit30  : BOOL := 1; //1 = Operator can Lock Messages
    Bit31  : BOOL := 1; //1 = Operator can Enable/Disable Comms
    END_STRUCT; //Operator Permissions
dwOS_Perm AT OS_Perm : DWORD;
ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;
SampleTime {
    S7_visible := 'false';
    S7_link := 'false';
    S7_samplertime:='true';
    S7_param := 'false' : REAL := 1; // Sampling time [s]
//Messages Configuration
MSG_EVID_1 {
    S7_visible := 'false';
    S7_link := 'false';
    S7_param := 'false';
    S7_server := 'alarm_archiv';
    S7_a_type := 'alarm_s' : DWORD:=0; //Message 1 ID (Alarm)
MSG_EVID_2 {
    S7_visible := 'false';
    S7_link := 'false';
    S7_param := 'false';
    S7_server := 'alarm_archiv';
    S7_a_type := 'alarm_s' : DWORD:=0; //Message 2 ID (MCP Alarm)
MSG_EVID_3 {
    S7_visible := 'false';
    S7_link := 'false';
    S7_param := 'false';
    S7_server := 'alarm_archiv';
    S7_a_type := 'alarm_s' : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
    S7_visible := 'false';
    S7_m_c := 'true';
    S7_string_1 := 'activ';
    S7_string_0 := 'inactiv' : BOOL; //Suppress Message 1
MsgSup2 {
    S7_visible := 'false';
    S7_m_c := 'true';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ' : BOOL; //Suppress Message 2
MsgSup3 {
    S7_visible := 'false';
    S7_m_c := 'true';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ' : BOOL; //Suppress Message 3
MsgLock {
    Op_Level := '3';
    S7_m_c := 'true';
    S7_visible := 'false';
    S7_link := 'false';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ' : BOOL; //Message Lock
END_VAR

```

```

//=====
VAR_OUTPUT
// Error Events
Func_Error {
    S7_visible :='false'      : STRING[8];           //Internal Function that Generated the Error
Func_Err_Num{
    S7_visible :='false'      : WORD;                //Function Error number
Panel_Con_Err {
    S7_visible :='true';
    S7_dynamic :='true';
    S7_string_1 :='Error'    : BOOL;                //Panel Connection Error

//Panel Funcional Conditions
FC_Alarm {
    S7_visible :='true';
    S7_dynamic :='true'     : BOOL;                //Alarm Funcional Condition
FC_Fault {
    S7_visible :='true';
    S7_dynamic :='true'     : BOOL;                //Fault Funcional Condition
FC_Disable {
    S7_visible :='true';
    S7_dynamic :='true'     : BOOL;                //Disable Funcional Condition
FC_Test {
    S7_visible :='true';
    S7_dynamic :='true'     : BOOL;                //Test Funcional Condition
FC_Daynight {
    S7_visible :='true';
    S7_dynamic :='true';
    S7_string_0 :='DayMode';
    S7_string_1 :='NightMode'} : BOOL;            //Daynight Funcional Condition
FC_MCPalm {
    S7_visible :='true';
    S7_dynamic :='true'     : BOOL;                //Manual Call Points Alarm Funcional Condition

//Panel Sounders Status
S_Delay {
    S7_visible :='false'    : BOOL;                //Sounders Delay Enabled
S_DelayAct {
    S7_visible :='false'    : BOOL;                //Sounders Activation Delay in progress
S_OutAct {
    S7_visible :='false'    : BOOL;                //Sounders Outputs Activated
S_OutSil {
    S7_visible :='false'    : BOOL;                //Sounders Outputs Silenced
S_Disable {
    S7_visible :='false'    : BOOL;                //Sounders Outputs Disabled
S_OvrTime {
    S7_visible :='false'    : BOOL;                //Sounders Override Time Elapsed
S_Test {
    S7_visible :='false'    : BOOL;                //Sounders on Test

//Panel Fire Brigade Status
FB_Delay {
    S7_visible :='false'    : BOOL;                //Fire Brigade: Delay Enabled
FB_DelayAct {
    S7_visible :='false'    : BOOL;                //Fire Brigade: Activation Delay in progress
FB_OutAct {
    S7_visible :='false'    : BOOL;                //Fire Brigade: Outputs Activated
FB_Ack {
    S7_visible :='false'    : BOOL;                //Fire Brigade: Acknowledged
FB_Disable {
    S7_visible :='false'    : BOOL;                //Fire Brigade: Disabled
FB_Test {
    S7_visible :='false'    : BOOL;                //Fire Brigade: On Test

//Panel Fire Protection Status
FP_Delay {
    S7_visible :='false'    : BOOL;                //Fire Protection: Delay Enabled
FP_DelayAct {
    S7_visible :='false'    : BOOL;                //Fire Protection: Activation Delay in progress
FP_OutAct {
    S7_visible :='false'    : BOOL;                //Fire Protection: Outputs Activated
FP_Ack {
    S7_visible :='false'    : BOOL;                //Fire Protection: Acknowledged
FP_Disable {
    S7_visible :='false'    : BOOL;                //Fire Protection: Disabled
FP_Test {
    S7_visible :='false'    : BOOL;                //Fire Protection: On Test

//Zones
Zone01_STS {
    S7_m_c :='true';
    BLK_Jump := '1';
    S7_edit :='signal';
    S7_visible :='true'    : ZTZoneSTS;           //Zone 01 Status
Zone02_STS {
    S7_m_c :='true';
    BLK_Jump := '1';

```

```

S7_edit      := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 02 Status
Zone03_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 03 Status
Zone04_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 04 Status
Zone05_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 05 Status
Zone06_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 06 Status
Zone07_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 07 Status
Zone08_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 08 Status
Zone09_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 09 Status
Zone10_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 10 Status
Zone11_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 11 Status
Zone12_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 12 Status
Zone13_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 13 Status
Zone14_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 14 Status
Zone15_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 15 Status
Zone16_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'true' }      : ZTZoneSTS;      //Zone 16 Status
Zone17_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'false' }     : ZTZoneSTS;      //Zone 17 Status
Zone18_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'false' }     : ZTZoneSTS;      //Zone 18 Status
Zone19_STS {
S7_m_c      := 'true';
BLK_Jump    := '1';
S7_edit     := 'signal';
S7_visible  := 'false' }     : ZTZoneSTS;      //Zone 19 Status

```

```

Zone20_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 20 Status
Zone21_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 21 Status
Zone22_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 22 Status
Zone23_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 23 Status
Zone24_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 24 Status
Zone25_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 25 Status
Zone26_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 26 Status
Zone27_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 27 Status
Zone28_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 28 Status
Zone29_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 29 Status
Zone30_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 30 Status
Zone31_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 31 Status
Zone32_STS {
  S7_m_c      := 'true';
  BLK_Jump   := '1';
  S7_edit    := 'signal';
  S7_visible := 'false'} : ZTZoneSTS;      //Zone 32 Status
Status1 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status1 Word
ZStatus1 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status Z1-4 Word
ZStatus2 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status Z5-8 Word
ZStatus3 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status Z9-12 Word
ZStatus4 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status Z13-16 Word
ZStatus5 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status Z17-20 Word
ZStatus6 {
  S7_visible:= 'false';
  S7_m_c:= 'true'} : DWORD;      //Status Z21-24 Word
ZStatus7 {

```

```

        S7_visible:='false';
        S7_m_c:='true'}           : DWORD;           //Status Z25-28 Word
ZStatus8 {
        S7_visible:='false';
        S7_m_c:='true'}           : DWORD;           //Status Z29-32 Word
OS_PermlLog {
        S7_visible:='false';
        S7_m_c:='true'}           : DWORD:=16#FFFFFFF; // Operator Panel permissions: output for OS
HMINAV          : HMINAVSTR;           //HMI Panel Interface

//Messages Configuration
MsgSTS1 {
        S7_visible := 'false'}     : WORD; //Message 1 Status
MsgACK1 {
        S7_visible := 'false' }    : BOOL; //Message 1 Acknowledge
MsgSTS2 {
        S7_visible := 'false'}     : WORD; //Message 2 Status
MsgACK2 {
        S7_visible := 'false' }    : BOOL; //Message 2 Acknowledge
MsgSTS3 {
        S7_visible := 'false'}     : WORD; //Message 3 Status
MsgACK3 {
        S7_visible := 'false' }    : BOOL; //Message 3 Acknowledge
END_VAR

// Variables =====
VAR
// TCON Connection Parameters structure (UDT65)
CON_PAR:
STRUCT
        block_length      : WORD := W#16#40; // Length: 64 Bytes (fixed)
        id                 : WORD := W#16#01; // Connection id (W#16#0001 to W#16#0FFF)
        connection_type   : BYTE := B#16#11; // Protocol TCP
        active_est        : BOOL := TRUE; // Connection type active establishment
        local_device_id   : BYTE := B#16#02;
        // Communication interface used
        // B#16#02: Communication via the integrated IE interface FOR the CPUs 315-2 PN/DP AND 317-2 PN/DP
        // B#16#03: Communication via the integrated IE interface FOR the CPUs 315T-3 PN/DP, 317T-3 PN/DP AND 319-3 PN/DP
        // B#16#05: Communication via the integrated IE interface X5 FOR the CPUs 41x AND 41xH (Rack 0)
        // B#16#06: Communication via the IE interface on interface slot 2 (IF2) with WinAC RTX (TCP only)
        // B#16#08: Communication via the integrated IE interface X8 FOR the CPUs 41x AND 41xH (Rack 0)
        // B#16#0B: Communication via the IE interface on interface slot 3 (IF3) with WinAC RTX (TCP only)
        // B#16#0F: Communication via the IE interface on interface slot 4 (IF4) with WinAC RTX (TCP only)
        // B#16#15: Communication via the integrated IE interface X5 FOR the CPUs 41xH (Rack 1)
        // B#16#18: Communication via the integrated IE interface X8 FOR the CPUs 41xH (Rack 1)
        local_tsap_id_len : BYTE := B#16#00; // Length of parameter local_tsap_id used
        rem_subnet_id_len : BYTE := B#16#00; // NOT used
        rem_staddr_len    : BYTE := B#16#04; // Length of IP address for the remote connection endpoint
        rem_tsap_id_len   : BYTE := B#16#02; // Length of the rem_tsap_id parameter used
        next_staddr_len   : BYTE := B#16#00; // Length of parameter next_staddr used
        local_tsap_id     : ARRAY [1..16] OF BYTE := // Local port number assigned automatically
        16(B#16#00);
        rem_subnet_id     : ARRAY [1..6] OF BYTE := // NOT used. Assign 0
        6(B#16#00);
        rem_staddr        : ARRAY [1..6] OF BYTE := //IP address remote conn endpoint, as instance, 192.168.2.3
        6(B#16#00); // rem_staddr[1] = B#16#C0 (192), rem_staddr[2] = B#16#A8 (168),
        // rem_staddr[3] = B#16#02 (002), rem_staddr[4] = B#16#03 (003)
        rem_tsap_id       : ARRAY [1..16] OF BYTE := // Remote port number (502 by default as MB standar port)
        B#16#01, // rem_tsap_id[1] = high BYTE OF the port no. in hex,
        B#16#F6, // rem_tsap_id[2] = low BYTE OF the port no. in hex,
        14(B#16#00); // rem_tsap_id[3-16] = B#16#00
        next_staddr       : ARRAY [1..6] OF BYTE := // NOT used for PN interface. Assign 0
        6(B#16#00);
        spare             : WORD := W#16#0000; // NOT used
END_STRUCT;
BUFF_RD : ARRAY [1..260] OF BYTE; // Read Data Buffer
PanFCSTS : BYTE; // Panel Functional Conditions Status
PanSOSTS : BYTE; // Panel Sounders Status
PanFBSTS : BYTE; // Panel Fire Brigade Status
PanFPSTS : BYTE; // Panel Fire Protection Status
ZonesSTS : ARRAY [1..32] OF WORD; // Zones Status
L1ModSTS : ARRAY [1..128] OF WORD; // Module Status Loop 1
L2ModSTS : ARRAY [1..128] OF WORD; // Module Status Loop 2
L3ModSTS : ARRAY [1..128] OF WORD; // Module Status Loop 3
L4ModSTS : ARRAY [1..128] OF WORD; // Module Status Loop 4
AckZonesSTS : ARRAY [1..32] OF WORD; // ZonesSTS for Modules acknowledgements
HPScanCMD : DWORD; // High Priority Scan command by any module
HPScanSTS : DWORD; // High Priority Scan status

xBZSTS AT ZonesSTS : ARRAY [1..64] OF BYTE; // ZonesSTS as Bytes to do a copy
xBPanFCSTS AT PanFCSTS : ARRAY [1..8] OF BOOL; // PanFCSTS as Bit to do a copy
xBPanSOSTS AT PanSOSTS : ARRAY [1..8] OF BOOL; // PanSOSTS as Bit to do a copy
xBPanFBSTS AT PanFBSTS : ARRAY [1..8] OF BOOL; // PanFBSTS as Bit to do a copy
xBPanFPSTS AT PanFPSTS : ARRAY [1..8] OF BOOL; // PanFPSTS as Bit to do a copy

// ModBus TCP/IP Application Data Unit Read
ADU_RD:
STRUCT
        Trans_Id : WORD; // Transaction identifier

```

```

Proto_Id      : WORD := W#16#00;      // NOT Used
Lenght        : WORD := W#16#06;      // Remaining Fields byte count (To read, always 6)
Unit_Id       : BYTE := B#16#00;      // ID for serial bridging on a non TCP/IP network
Func_Code     : BYTE := BYTE#03;      // ModBus Function Code (Read Holding Registers)
Start_Addr    : WORD;                  // Starting Address Register to be Read
Qty_Reg       : WORD;                  // Quantity of Registers to be Read
END_STRUCT;

// ModBus TCP/IP Application Data Unit Write
ADU_WSR:
STRUCT
  Trans_Id     : WORD;                  // Transaction identifier
  Proto_Id     : WORD := W#16#00;      // NOT Used
  Lenght       : WORD := W#16#06;      // Remaining Fields byte count (W single R, always 6)
  Unit_Id      : BYTE := B#16#00;      // ID for serial bridging on a non TCP/IP network
  Func_Code    : BYTE := BYTE#06;      // MB Function Code (Write Single Holding Register)
  Addr         : WORD;                  // Register Address to Write
  Value        : WORD;                  // Register Value to Write
END_STRUCT;

//MB Read Tasks Indexes
MB_JOB_Idx:
STRUCT
  Task         : INT;                  // Task Index (0:Panel,1:Zones,2:Modules)
  Zone         : INT;                  // Zone Index (1..32)
  Loop         : INT;                  // Loop Index (1..4)
  Module       : INT;                  // Module Index (1..127)
END_STRUCT;

// System Functions
CONNECT       : TCON;                  // Connection Function
DISCONNECT    : TDISCON;               // Disconnection Function
SEND          : TSEND;                 // Send Data Function
RECEIVE       : TRCV;                  // Receive Data Function

// Auxiliar Variables
xWrd_Z        : WORD := W#16#0000;     // WORD Zero Value
xWrd_OOS      : WORD := W#16#8000;     // WORD Out of Service
xCom_Start    : BOOL;                  // Start Communications
xCom_Run      : BOOL;                  // Communications Running
xNCom_Panel   : BOOL;                  // No Communications Panel
xTCON_Err     : BOOL;                  // TCON function Error
xPanel_Con_Err : BOOL;                 // Connection Error with Panel
xMB_Err       : BOOL;                  // ModBus Comm Error
xRx_Err       : BOOL;                  // Receive Comm Error
xTx_Cmd_RD    : BOOL;                  // Read Transmission Command
xAux_Err      : INT;                   // Auxiliar Return Function Error
xZone         : INT;                   // Zone num
xLoop         : INT;                   // Loop num
xMod          : INT;                   // Module num
xModCount     : INT;                   // Modules read count
xLLoop        : INT;                   // Last Loop num
xLMod         : INT;                   // Last Module num
xIdx          : INT;                   // Index FOR statement
xWValue       : WORD;                  // WORD Value
xArrWValue AT xWValue : SBits16inWORD; // WORD Value as Bits
xiWValue AT xWValue   : INT;           // WORD Value as Int
xBWValue AT xWValue   : ARRAY [1..2] OF BYTE; // WORD Value as Bytes
xModF         : BOOL;                  // Module found
PanMonTime    : REAL;                  // Current Time for xPanel_Con_Err
xHMISTS       : WORD;                  // HMI Status
xBithMISTS AT xHMISTS : SBits16inWORD; // HMI Status as Bits
CMP_ID        : DWORD;                 // Area code
xMsgCond1     : BOOL;                  // Condition MSG1
xMsgCond2     : BOOL;                  // Condition MSG2
xMsgCond3     : BOOL;                  // Condition MSG3
xMsgTrig1     : BOOL;                  // Trigger MSG1
xMsgTrig2     : BOOL;                  // Trigger MSG2
xMsgTrig3     : BOOL;                  // Trigger MSG3
xMsgArch1     : INT;                   // MSG1 Status
xMsgArch2     : INT;                   // MSG2 Status
xMsgArch3     : INT;                   // MSG3 Status
AuxValue:
  STRUCT
    Zone       : INT;                  // Msg Zone
    DBNum      : WORD;                  // Msg DB num
  END_STRUCT;
xRetSts       : BOOL;                  // Msg Status
END_VAR

// Temporal Variables =====
VAR_TEMP

// Structures to call RD_SINFO (SFC6) in order to know OB running
sTOP_SI:
STRUCT
  EV_CLASS    : BYTE;                  // Bits 0 to 3: Event ID· Bits 4 to 7: Event class

```

```

EV_NUM          : BYTE;    // Event number
PRIORITY        : BYTE;    // Num of the priority class (B#16#FE: OB NOT available or locked)
NUM             : BYTE;    // OB number.
TYP2_3         : BYTE;    // Data ID 2_3: identifies the information entered in ZI2_3
TYP1           : BYTE;    // Data ID 1 : identifies the information entered in ZI1
ZI1            : WORD;    // Additional information 1
ZI2_3         : DWORD;    // Additional information 2_3
END_STRUCT;

sSTART_UP_SI:
STRUCT
  EV_CLASS      : BYTE;    // Bits 0 to 3: Event ID. Bits 4 to 7: Event class
  EV_NUM        : BYTE;    // Event number
  PRIORITY      : BYTE;    // Num of the priority class (B#16#FE: OB NOT available or locked)
  NUM           : BYTE;    // OB number.
  TYP2_3       : BYTE;    // Data ID 2_3: identifies the information entered in ZI2_3
  TYP1         : BYTE;    // Data ID 1 : identifies the information entered in ZI1
  ZI1          : WORD;    // Additional information 1
  ZI2_3       : DWORD;    // Additional information 2_3
END_STRUCT;

// Structures to copy data areas
PSource        : ANY;     // Pointer to Source Data Area
PScs_STR AT PSource : PointSTR; // Pointer Structure
PDestin       : ANY;     // Pointer to Source Data Area
Pdst_STR AT PDestin : PointSTR; // Pointer Structure

// Variable for status
dwStatus       : DWORD;
ArrBitStatus AT dwStatus : SBits32inDWORD;

dwZStatus      : DWORD;
ArrByteZStatus AT dwZStatus : ARRAY[1..4] OF BYTE;

dwPerm        : DWORD;
ArrBitPerm AT dwPerm : SBits32inDWORD;

// Variables for get the working OB
xOB_Cycle     : BOOL;    // Block is running in a cyclic interrupt OB
xOB_Start     : BOOL;    // Block is running in start OB
xSnErr       : INT;     // Error information.

// Variable for Operator Permissions
xdwOS_PermlLog : DWORD; // Temporary variable for the value of OS_PermlLog
strxOS_PermlLog AT xdwOS_PermlLog : ARRAY[0..31] OF BOOL;
ArrxOS_PermlLog AT xdwOS_PermlLog : ARRAY[0..3] OF BYTE;
END_VAR

// CODE =====
BEGIN
// Narrow ranges
IF Zones>32 THEN Zones:=32; END_IF;
IF Zones<1 THEN Zones:=1; END_IF;
IF Loop1_Mod>127 THEN Loop1_Mod:=127; END_IF;
IF Loop1_Mod<1 THEN Loop1_Mod:= 1; END_IF;
IF Loop2_Mod>127 THEN Loop2_Mod:=127; END_IF;
IF Loop2_Mod<0 THEN Loop2_Mod:= 0; END_IF;
IF Loop3_Mod>127 THEN Loop3_Mod:=127; END_IF;
IF Loop3_Mod<0 THEN Loop3_Mod:= 0; END_IF;
IF Loop4_Mod>127 THEN Loop4_Mod:=127; END_IF;
IF Loop4_Mod<0 THEN Loop4_Mod:= 0; END_IF;
IF Mod_cycle>120 THEN Mod_cycle:=120; END_IF;
IF Mod_cycle<4 THEN Mod_cycle:= 4; END_IF;

//Read CPU working step
xSnErr := RD_SINFO(TOP_SI:= sTOP_SI, START_UP_SI:= sSTART_UP_SI);
xOB_Cycle := (BYTE_TO_INT(sTOP_SI.NUM)>=30) AND (BYTE_TO_INT(sTOP_SI.NUM)<=38);
xOB_Start := (BYTE_TO_INT(sTOP_SI.NUM)>=100) AND (BYTE_TO_INT(sTOP_SI.NUM)<=102);

xCom_Start := EnaComOp AND xOB_Cycle; // Block is Enabled and working in a Cyclic OB
xNCom_Panel := NOT(xCom_Start) OR Panel_Con_Err; // Panel is not communicating

// Initialize variables
IF NOT(xCom_Start) THEN
  xTCON_Err := FALSE; // Reset Connection Error
  Func_Err_Num := W#16#00; // Reset Error Code
  Func_Error := ''; // Reset Function Name that generated the error
  MB_JOB_Idx.Task := 0; // Reset ModBus Job Indexes
  MB_JOB_Idx.Loop := 1;
  MB_JOB_Idx.Module := 1;
  xRx_Err := FALSE; // Reset Watchdog receive data
  xPanel_Con_Err := FALSE; // Reset Connection Error with the Panel
  xMB_Err := FALSE; // Reset ModBus Comm Error
  PSource := Id; // Pointer to Source to retrieve DBNum
  CMP_ID := WORD_TO_DWORD(PSrc_STR.DB_Num); // DBNum for Messages
  AuxValue.Zone := 0;
  AuxValue.DBNum := PSrc_STR.DB_Num;

```

```

END_IF;

// Initialize Connection Parameters
IF xCom_Start AND NOT (xCom_Run) THEN
  CON_PAR.id := Id; // Connection ID
  CON_PAR.local_device_id := Dev_ID; // Communication Interface
  CON_PAR.rem_staddr[1] := BIPAddr[2]; // IP address octet 1
  CON_PAR.rem_staddr[2] := BIPAddr[4]; // IP address octet 2
  CON_PAR.rem_staddr[3] := BIPAddr[6]; // IP address octet 3
  CON_PAR.rem_staddr[4] := BIPAddr[8]; // IP address octet 4
  CON_PAR.rem_tsap_id[1] := BPort[1]; // High Byte Remote Port number
  CON_PAR.rem_tsap_id[2] := BPort[2]; // Low Byte Remote Port number
END_IF;
xCom_Run := xCom_Start;

// Open Communications Channel
CONNECT (REQ:=xCom_Run, ID:=CON_PAR.id, CONNECT:=CON_PAR); // Open Communications Channel
IF CONNECT.ERROR AND (CONNECT.STATUS<> W#16#80A7) THEN // 80A7 when disconnect communi before
  Func_Err_Num:=CONNECT.STATUS; // connection is establ.not considered as an Error
  Func_Error := 'TCON';
  xTCON_Err := TRUE;
END_IF;

// Communications Channel Opened
IF NOT (xTCON_Err) THEN
  RECEIVE (EN_R:=xCom_Run, ID:=CON_PAR.id, LEN:=0, DATA:=BUFF_RD); // Open Receive Channel
  DISCONNECT (REQ:=NOT (xCom_Run), ID:=CON_PAR.id); // Close Communications

  // Evaluate Function Errors
  IF DISCONNECT.ERROR THEN // TDISCON Function Error
    Func_Err_Num:=DISCONNECT.STATUS; // TDISCON Error Code is showed
    Func_Error:='TDISCON';
  ELSIF RECEIVE.ERROR THEN // TRCV Function Error
    Func_Err_Num:=RECEIVE.STATUS; // TRCV Error Code is showed
    Func_Error:='TRCV';
  ELSE
    Func_Err_Num:=W#16#00; // No Function Errors
    Func_Error:='';
  END_IF;
  xPanel_Con_Err:=RECEIVE.ERROR; // Check Panel Connection

  // Read Data
  IF RECEIVE.NDR THEN // New Data Read
    xRx_Err := FALSE; // Reset watchdog receive data
    xMB_Err := BYTE_TO_INT (BUFF_RD[8]) > 128; // Function code h8x=response error
    IF NOT (xMB_Err) THEN
      CASE (BYTE_TO_INT (BUFF_RD[2] AND B#16#0F)) OF // Last 4 bits Of TID to retrieve Area
        0: //Panel Status // TID |00000000|00000000|
          IF (RECEIVE.RCVD_LEN=13) THEN // HEAD(7)+FC(1)+BC(1)+DATA(2*2)
            PanFCSTS := BUFF_RD[11]; // Panel Functional Conditions Status
            PanSOSTS := BUFF_RD[10]; // Panel Sounders Status
            PanFBSTS := BUFF_RD[13]; // Panel Fire Brigade Status
            PanFPSTS := BUFF_RD[12]; // Panel Fire Protection Status
          ELSE
            xMB_Err := TRUE; // Received Data Length Error
          END_IF;
        1: //Zone Status // TID |__Zone|0000|0001|
          IF (RECEIVE.RCVD_LEN=17) THEN // HEAD(7)+FC(1)+BC(1)+DATA(4*2)
            PSource := BUFF_RD[10]; // Pointer to Source
            PSrc_STR.TypeCode := BYTE#04; // Work with WORDS
            PSrc_STR.Quantity := WORD#04; // 4 Words of Data

            xZone := BYTE_TO_INT (BUFF_RD[1]); // High Byte to retrieve Zone num
            IF (xZone>0 AND xZone<33) THEN // Check Range
              PDestin := ZonesSTS[xZone]; // Pointer to Destin
              PDst_STR.TypeCode := BYTE#04; // Work with WORDS
              PDst_STR.Quantity := WORD#04; // 4 Words of Data

              xAux_Err:=BLKMOV (SRCBLK := PSource, DSTBLK := PDestin); // Copy data area
            END_IF;
          ELSE
            xMB_Err := TRUE; // Received Data Length Error
          END_IF;
        2: //Modules Status // TID |__Module|Loop|0010|
          IF (RECEIVE.RCVD_LEN=17) THEN // HEAD(7)+FC(1)+BC(1)+DATA(4*2)
            PSource := BUFF_RD[10]; // Pointer to Source
            PSrc_STR.TypeCode := BYTE#04; // Work with WORDS
            PSrc_STR.Quantity := WORD#04; // 4 Words of Data

            xLoop:=BYTE_TO_INT (SHR (IN:=(BUFF_RD[2] AND B#16#F0), N:=4)); // Retr. Loop num.
            xMod := BYTE_TO_INT (BUFF_RD[1]); // High Byte to retrieve Zone num
            IF ((xLoop>0 AND xLoop<5) AND // Check Ranges
              (xMod>0 AND xMod<126)) THEN
              CASE (xLoop) OF

```

```

1: PDestin := L1ModSTS[xMod];           // Pointer to Destin
2: PDestin := L2ModSTS[xMod];           // Pointer to Destin
3: PDestin := L3ModSTS[xMod];           // Pointer to Destin
4: PDestin := L4ModSTS[xMod];           // Pointer to Destin
END_CASE;
Pdst_STR.TypeCode := BYTE#04;           // Work with WORDS
Pdst_STR.Quantity := WORD#04;           // 4 Words of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data area
END_IF;
ELSE
xMB_Err := TRUE;                         // Received Data Length Error
END_IF;
END_CASE;
END_IF;
END_IF;
END_IF;

// Request Data
IF NOT(xPanel_Con_Err) AND xCom_Run THEN // Panel Connection OK
IF NOT(SEND_BUSY) THEN                  // Previous Transmission Expired
CASE (MB_JOB_Idx.Task) OF              // Read ModBus Job
0:// Panel
// |00000000|00000000| = Panel
ADU_RD.Trans_Id := WORD#00;           // Panel
ADU_RD.Start_Addr := W#16#2000;       // Start Address 0x2000
ADU_RD.Qty_Reg := WORD#02;           // 2 Registers
MB_JOB_Idx.Task := 1;                 // Jump to Zones
MB_JOB_Idx.Zone := 1;
xTx_Cmd_RD := TRUE;

1:// Zones 1..32
// |___Zone|0000|0001| = Zone
xBWValue[1] := INT_TO_BYTE(MB_JOB_Idx.Zone);
xBWValue[2] := BYTE#01;
ADU_RD.Trans_Id := xWValue;

// Start Address 0x3000
xiWValue := INT#16#3000+(MB_JOB_Idx.Zone-1);
ADU_RD.Start_Addr := xWValue;

ADU_RD.Qty_Reg := WORD#04;           // Read 4 Zones Each time
xTx_Cmd_RD := TRUE;

MB_JOB_Idx.Zone := MB_JOB_Idx.Zone+4;
IF MB_JOB_Idx.Zone>Zones THEN        // Last Zone Reached
MB_JOB_Idx.Task := 2;                // Jump to Modules
xModCount := 0;                      // Reset Modules Count
END_IF;

2:// Modules
IF (HPScanCMD<>DWORD#0) AND (HPScanCMD<>HPScanSTS) THEN
xLMod := MB_JOB_Idx.Module;           //Save last Module scanned
xLLoop := MB_JOB_Idx.Loop;            //Save last Loop scanned
END_IF;
HPScanSTS:=HPScanCMD;

// High priority scan
IF (HPScanCMD<>DWORD#0) THEN
xModF := FALSE;                       // Reset Module Found
REPEAT                                 // Search modules in the affected zone
CASE (MB_JOB_Idx.Loop) OF
//Looks at module Base Address and the next 3 addresses
//because read command is for 4 registers
1: FOR xIdx:=0 TO 3 BY 1 DO
xWValue := L1ModSTS[MB_JOB_Idx.Module+xIdx];
IF (xArrWValue.X14) THEN
xModF := TRUE;
EXIT;
END_IF;
END_FOR;
2: FOR xIdx:=0 TO 3 BY 1 DO
xWValue := L2ModSTS[MB_JOB_Idx.Module+xIdx];
IF (xArrWValue.X14) THEN
xModF := TRUE;
EXIT;
END_IF;
END_FOR;
3: FOR xIdx:=0 TO 3 BY 1 DO
xWValue := L3ModSTS[MB_JOB_Idx.Module+xIdx];
IF (xArrWValue.X14) THEN
xModF := TRUE;
EXIT;
END_IF;
END_FOR;
4: FOR xIdx:=0 TO 3 BY 1 DO
xWValue := L4ModSTS[MB_JOB_Idx.Module+xIdx];

```

```

        IF (xArrWValue.X14) THEN
            xModF := TRUE;
            EXIT;
        END_IF;
    END_FOR;
END_CASE;

// Module was found
IF xModF THEN
    // |__Module|Loop|0010| = Module
    xBWValue[1] := INT_TO_BYTE(MB_JOB_Idx.Module);
    xBWValue[2] := INT_TO_BYTE((MB_JOB_Idx.Loop*16)+2);
    ADU_RD.Trans_Id := xWValue;

    // Start Address L1:0x7000, L2:0x7100...
    xiWValue := INT#16#7000+(MB_JOB_Idx.Loop-1)*256+(MB_JOB_Idx.Module-1);
    ADU_RD.Start_Addr := xWValue;

    ADU_RD.Qty_Reg := WORD#04; // Read 4 Modules Each time
    xTx_Cmd_RD := TRUE;

    xModCount := xModCount+4;
    IF (xModCount>=Mod_cycle) THEN // Max Mod. per cycle reached
        MB_JOB_Idx.Task := 0; // Jump to Panel
    END_IF;
END_IF;

MB_JOB_Idx.Module := MB_JOB_Idx.Module+4;
//Check ranges
IF (MB_JOB_Idx.Loop=1) AND
(MB_JOB_Idx.Module>Loop1_Mod) THEN // Last Module reached for Loop1
    MB_JOB_Idx.Loop := 2;
    MB_JOB_Idx.Module:= 1;
END_IF;
IF (MB_JOB_Idx.Loop=2) AND
(MB_JOB_Idx.Module>Loop2_Mod) THEN // Last Module reached for Loop2
    MB_JOB_Idx.Loop := 3;
    MB_JOB_Idx.Module:= 1;
END_IF;
IF (MB_JOB_Idx.Loop=3) AND
(MB_JOB_Idx.Module>Loop3_Mod) THEN // Last Module reached for Loop3
    MB_JOB_Idx.Loop := 4;
    MB_JOB_Idx.Module:= 1;
END_IF;
IF (MB_JOB_Idx.Loop=4) AND
(MB_JOB_Idx.Module>Loop4_Mod) THEN // Last Module reached for Loop4
    MB_JOB_Idx.Loop := 1;
    MB_JOB_Idx.Module:= 1;
END_IF;
UNTIL (MB_JOB_Idx.Module=xLMod AND MB_JOB_Idx.Loop=xLLoop) OR xModF END_REPEAT;

// A complete scan cycle is reached
IF NOT(xModF) THEN
    HPScanCMD:=DWORD#0;
END_IF;
END_IF;
// Normal scan
IF (HPScanCMD=0) THEN
    // |__Module|Loop|0010| = Module
    xBWValue[1] := INT_TO_BYTE(MB_JOB_Idx.Module);
    xBWValue[2] := INT_TO_BYTE((MB_JOB_Idx.Loop*16)+2);
    ADU_RD.Trans_Id := xWValue;

    // Start Address L1:0x7000, L2:0x7100...
    xiWValue := INT#16#7000+(MB_JOB_Idx.Loop-1)*256+(MB_JOB_Idx.Module-1);
    ADU_RD.Start_Addr := xWValue;

    ADU_RD.Qty_Reg := WORD#04; // Read 4 Modules Each time
    xTx_Cmd_RD := TRUE;

    xModCount := xModCount+4;
    IF (xModCount>=Mod_cycle) THEN // Max Mod. per cycle reached
        MB_JOB_Idx.Task := 0; // Jump to Panel
    END_IF;

    MB_JOB_Idx.Module := MB_JOB_Idx.Module+4;
    //Check ranges
    IF (MB_JOB_Idx.Loop=1) AND
(MB_JOB_Idx.Module>Loop1_Mod) THEN // Last Module reached for Loop1
        MB_JOB_Idx.Loop := 2;
        MB_JOB_Idx.Module:= 1;
    END_IF;
    IF (MB_JOB_Idx.Loop=2) AND
(MB_JOB_Idx.Module>Loop2_Mod) THEN // Last Module reached for Loop2
        MB_JOB_Idx.Loop := 3;
        MB_JOB_Idx.Module:= 1;
    END_IF;

```

```

        END_IF;
        IF (MB_JOB_Idx.Loop=3) AND
            (MB_JOB_Idx.Module>Loop3_Mod) THEN // Last Module reached for Loop3
            MB_JOB_Idx.Loop := 4;
            MB_JOB_Idx.Module:= 1;
        END_IF;
        IF (MB_JOB_Idx.Loop=4) AND
            (MB_JOB_Idx.Module>Loop4_Mod) THEN // Last Module reached for Loop4
            MB_JOB_Idx.Loop := 1;
            MB_JOB_Idx.Module:= 1;
            MB_JOB_Idx.Task := 0; // Jump to Panel
        END_IF;
    END_IF;
ELSE:
    MB_JOB_Idx.Task := 0;
END_CASE;
END_IF;
END_IF;

SEND (REQ:=xTx_Cmd_RD, ID:=CON_PAR.id, LEN:=12, DATA:=ADU_RD); // Send Data to Read
xTx_Cmd_RD := FALSE;

// Panel Connection Error Alarm
IF (xPanel_Con_Err OR xMB_Err OR xRx_Err) THEN
    IF (PanMonTime < Panel_Conn_Time) THEN
        PanMonTime := PanMonTime + SampleTime;
    ELSE
        Panel_Con_Err := TRUE;
    END_IF;
ELSE
    PanMonTime := 0.0;
    Panel_Con_Err := FALSE;
END_IF;
IF NOT(xPanel_Con_Err) AND xCom_Run THEN xRx_Err := TRUE; END_IF; //Set watchdog receive data
END_IF;

// Reset status on error or not start
IF xNCom_Panel THEN
    //Panel Status
    PanFCSTS := BYTE#0;
    PanSOSTS := BYTE#0;
    PanFBSTS := BYTE#0;
    PanFPSTS := BYTE#0;
    MB_JOB_Idx.Task := 0;
    MB_JOB_Idx.Loop := 1;
    MB_JOB_Idx.Module := 1;
    //Zones Status
    PDestin := ZonesSTS[1]; // Pointer to Zones
    PDst_STR.TypeCode := BYTE#04; // Work with WORDS
    PDst_STR.Quantity := WORD#32; // 32 Words of Data
    xAux_Err:=FILL (BVAL:=xWrd_Z, BLK:=PDestin); // Fill Zones data with 0 Value
    //Loop1 Modules Status
    PDestin := L1ModSTS[1]; // Pointer to Zones
    PDst_STR.TypeCode := BYTE#04; // Work with WORDS
    PDst_STR.Quantity := WORD#128; // 128 Words of Data
    xAux_Err:=FILL (BVAL:=xWrd_OOS, BLK:=PDestin); // Fill STSs data with OOS Value
    //Loop2 Modules Status
    PDestin := L2ModSTS[1]; // Pointer to Zones
    PDst_STR.TypeCode := BYTE#04; // Work with WORDS
    PDst_STR.Quantity := WORD#128; // 128 Words of Data
    xAux_Err:=FILL (BVAL:=xWrd_OOS, BLK:=PDestin); // Fill STSs data with OOS Value
    //Loop3 Modules Status
    PDestin := L3ModSTS[1]; // Pointer to Zones
    PDst_STR.TypeCode := BYTE#04; // Work with WORDS
    PDst_STR.Quantity := WORD#128; // 128 Words of Data
    xAux_Err:=FILL (BVAL:=xWrd_OOS, BLK:=PDestin); // Fill STSs data with OOS Value
    //Loop4 Modules Status
    PDestin := L4ModSTS[1]; // Pointer to Zones
    PDst_STR.TypeCode := BYTE#04; // Work with WORDS
    PDst_STR.Quantity := WORD#128; // 128 Words of Data
    xAux_Err:=FILL (BVAL:=xWrd_OOS, BLK:=PDestin); // Fill STSs data with OOS Value
    // ZonesSTS for Modules acknowledgements
    PDestin := AckZonesSTS[1]; // Pointer to Zones Ack
    PDst_STR.TypeCode := BYTE#04; // Work with WORDS
    PDst_STR.Quantity := WORD#32; // 32 Words of Data
    xAux_Err:=FILL (BVAL:=xWrd_OOS, BLK:=PDestin); // Fill Zones Ack data with OOS Value
END_IF;

//Zone event Acknowledgment by the modules
FOR xIdx:=1 TO 32 BY 1 DO
    AckZonesSTS[xIdx]:=AckZonesSTS[xIdx] AND ZonesSTS[xIdx];
END_FOR;

// Update OUTPUTS
//Panel Functional Conditions Status
FC_Alarm :=xBPanFCSTS[1];
FC_Fault :=xBPanFCSTS[2];

```

```

FC_Disable :=xBPanFCSTS[3];
FC_Test :=xBPanFCSTS[4];
FC_Daynight :=xBPanFCSTS[5];
FC_MCPalm :=xBPanFCSTS[6];
//Panel Sounders Status
S_Delay :=xBPanSOSTS[1];
S_DelayAct :=xBPanSOSTS[2];
S_OutAct :=xBPanSOSTS[3];
S_OutSil :=xBPanSOSTS[4];
S_Disable :=xBPanSOSTS[5];
S_OvrTime :=xBPanSOSTS[6];
S_Test :=xBPanSOSTS[7];
//Panel Fire Brigade Status
FB_Delay :=xBPanFBSTS[1];
FB_DelayAct :=xBPanFBSTS[2];
FB_OutAct :=xBPanFBSTS[3];
FB_Ack :=xBPanFBSTS[4];
FB_Disable :=xBPanFBSTS[5];
FB_Test :=xBPanFBSTS[6];
//Panel Fire Protection Status
FP_Delay :=xBPanFPSTS[1];
FP_DelayAct :=xBPanFPSTS[2];
FP_OutAct :=xBPanFPSTS[3];
FP_Ack :=xBPanFPSTS[4];
FP_Disable :=xBPanFPSTS[5];
FP_Test :=xBPanFPSTS[6];
//Zones
PSource := xBZSTS[2]; // Pointer to Source
Psrc_STR.TypeCode := BYTE#02; // Work with BYTES
Psrc_STR.Quantity := WORD#63; // 63 Bytes of Data

PDestin := Zone01_STS.PreAlarm; // Pointer to Destination
Pdst_STR.TypeCode := BYTE#02; // Work with BYTES
Pdst_STR.Quantity := WORD#63; // 63 Bytes of Data

xAux_Err:=BLKMOV(SRCBLK := PSource, DSTBLK := PDestin); // Copy data área

//Status1 DWord
dwStatus := DWORD#0;
ArrBitStatus.X0 := Panel_Con_Err;
ArrBitStatus.X1 := FC_Alarm;
ArrBitStatus.X2 := FC_MCPalm;
ArrBitStatus.X3 := FC_Fault;
ArrBitStatus.X4 := FC_Disable;
ArrBitStatus.X5 := FC_Test;
ArrBitStatus.X6 := FC_Daynight;
ArrBitStatus.X7 := S_Delay;
ArrBitStatus.X8 := S_DelayAct;
ArrBitStatus.X9 := S_OutAct;
ArrBitStatus.X10 := S_OutSil;
ArrBitStatus.X11 := S_Disable;
ArrBitStatus.X12 := S_OvrTime;
ArrBitStatus.X13 := S_Test;
ArrBitStatus.X14 := FB_Delay;
ArrBitStatus.X15 := FB_DelayAct;
ArrBitStatus.X16 := FB_OutAct;
ArrBitStatus.X17 := FB_Ack;
ArrBitStatus.X18 := FB_Disable;
ArrBitStatus.X19 := FB_Test;
ArrBitStatus.X20 := FP_Delay;
ArrBitStatus.X21 := FP_DelayAct;
ArrBitStatus.X22 := FP_OutAct;
ArrBitStatus.X23 := FP_Ack;
ArrBitStatus.X24 := FP_Disable;
ArrBitStatus.X25 := FP_Test;
ArrBitStatus.X26 := S_Disable OR S_Delay OR FB_Disable OR FB_Delay OR FP_Disable OR FP_Delay;
ArrBitStatus.X27 := S_Test OR FB_Test OR FP_Test;
ArrBitStatus.X28 := NOT(EnaComOp);
ArrBitStatus.X29 := MsgLock;
Status1 := dwStatus;

//ZStatus DWord
dwZStatus := DWORD#0;
FOR xIdx:=1 TO 32 BY 4 DO
  xWValue := ZonesSTS[xIdx];
  ArrByteZStatus[4] := xBWValue[2];
  xWValue := ZonesSTS[xIdx+1];
  ArrByteZStatus[3] := xBWValue[2];
  xWValue := ZonesSTS[xIdx+2];
  ArrByteZStatus[2] := xBWValue[2];
  xWValue := ZonesSTS[xIdx+3];
  ArrByteZStatus[1] := xBWValue[2];
CASE xIdx OF // Copy Bytes to Zone Status
  1: ZStatus1 := dwZStatus;
  5: ZStatus2 := dwZStatus;
  9: ZStatus3 := dwZStatus;
  13: ZStatus4 := dwZStatus;

```

```

17: ZStatus5 := dwZStatus;
21: ZStatus6 := dwZStatus;
25: ZStatus7 := dwZStatus;
29: ZStatus8 := dwZStatus;
END_CASE;
END_FOR;
//Panel OS_PermLog DWord
ArrBitPerm.X30 := OS_Perm.Bit30; //Operator can Lock Messages
ArrBitPerm.X31 := OS_Perm.Bit31; //Operator can Enable/Disable Comms
OS_PermLog := dwPerm;

//Alarms Managment
xMsgCond1 := FC_Alarm AND NOT(MsgSup1 OR MsgLock);
xMsgCond2 := FC_MCPalm AND NOT(MsgSup2 OR MsgLock);
xMsgCond3 := Panel_Con_Err OR FC_Fault AND NOT(MsgSup3 OR MsgLock);
//Alarm MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
  MsgSTS1:=INT_TO_WORD(ALARM_DQ(
    SIG := xMsgCond1
    ,ID := W#16#EEEE
    ,EV_ID := MSG_EVID_1
    ,CMP_ID := CMP_ID
    ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xMsgArch1:=ALARM_SC(
  EV_ID := MSG_EVID_1
  ,STATE := xRetSts
  ,Q_STATE := MsgACK1);

//evacuate MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
  MsgSTS2:=INT_TO_WORD(ALARM_DQ(
    SIG := xMsgCond2
    ,ID := W#16#EEEE
    ,EV_ID := MSG_EVID_2
    ,CMP_ID := CMP_ID
    ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xMsgArch2:=ALARM_SC(
  EV_ID := MSG_EVID_2
  ,STATE := xRetSts
  ,Q_STATE := MsgACK2);

//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
  MsgSTS3:=INT_TO_WORD(ALARM_DQ(
    SIG := xMsgCond3
    ,ID := W#16#EEEE
    ,EV_ID := MSG_EVID_3
    ,CMP_ID := CMP_ID
    ,SD := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xMsgArch3:=ALARM_SC(
  EV_ID := MSG_EVID_3
  ,STATE := xRetSts
  ,Q_STATE := MsgACK3);

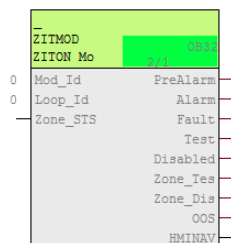
//HMINAV
//Status
xBithMISTS.X0 := FC_Alarm OR FC_MCPalm;
xBithMISTS.X1 := false;
xBithMISTS.X2 := Panel_Con_Err OR FC_Fault;
xBithMISTS.X3 := false;
//Msg activated
xBithMISTS.X4 := xMsgCond1 OR xMsgCond2;
xBithMISTS.X5 := false;
xBithMISTS.X6 := xMsgCond3;
xBithMISTS.X7 := false;
//Msg Acknowled
xBithMISTS.X8 := MsgACK1 AND MsgACK2;
xBithMISTS.X9 := false;
xBithMISTS.X10 := MsgACK3;
xBithMISTS.X11 := false;
//Msg in Archive (true if msg in Archive)
xBithMISTS.X12 := (xMsgArch1=0) OR (xMsgArch2=0);
xBithMISTS.X13 := false;
xBithMISTS.X14 := (xMsgArch3=0);
xBithMISTS.X15 := false;
HMINAV.Status := xHMISTS;
HMINAV.DBNum := WORD_TO_INT(AuxValue.DBNum);
HMINAV.Btype := 10; //ZITONP2 code=10

END_FUNCTION_BLOCK

```

2.17 FB555: ZITMOD

- Representación Gráfica



- Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
Mod_Id	Int	0.0	0	Module Id (1..127)
Loop_Id	Int	2.0	0	Loop Id (1..4)
RstActCycOp	Bool	4.0	FALSE	Reset Acitvate Cycles
Zone_STS	Any	6.0		Connect to ZITONP2 ZoneX_STS
OS_Perm	Struct	16.0		Operator Permissions
Bit0	Bool	16.0	TRUE	1 = Operator can Reset Activated Cycles Totalizer
Bit1	Bool	16.1	TRUE	1 = Operator can Lock Messages
Bit2	Bool	16.2	TRUE	Reserved
Bit3	Bool	16.3	TRUE	Reserved
Bit4	Bool	16.4	TRUE	Reserved
Bit5	Bool	16.5	TRUE	Reserved
Bit6	Bool	16.6	TRUE	Reserved
Bit7	Bool	16.7	TRUE	Reserved
Bit8	Bool	17.0	TRUE	Reserved
Bit9	Bool	17.1	TRUE	Reserved
Bit10	Bool	17.2	TRUE	Reserved
Bit11	Bool	17.3	TRUE	Reserved
Bit12	Bool	17.4	TRUE	Reserved
Bit13	Bool	17.5	TRUE	Reserved
Bit14	Bool	17.6	TRUE	Reserved
Bit15	Bool	17.7	TRUE	Reserved
Bit16	Bool	18.0	TRUE	Reserved
Bit17	Bool	18.1	TRUE	Reserved
Bit18	Bool	18.2	TRUE	Reserved
Bit19	Bool	18.3	TRUE	Reserved
Bit20	Bool	18.4	TRUE	Reserved
Bit21	Bool	18.5	TRUE	Reserved
Bit22	Bool	18.6	TRUE	Reserved
Bit23	Bool	18.7	TRUE	Reserved
Bit24	Bool	19.0	TRUE	Reserved
Bit25	Bool	19.1	TRUE	Reserved
Bit26	Bool	19.2	TRUE	Reserved
Bit27	Bool	19.3	TRUE	Reserved
Bit28	Bool	19.4	TRUE	Reserved
Bit29	Bool	19.5	TRUE	Reserved
Bit30	Bool	19.6	TRUE	Reserved
Bit31	Bool	19.7	TRUE	Reserved
Feature	Struct	20.0		Status of various features
Bit0	Bool	20.0	TRUE	1 = Lock Messages on Module Disabled
Bit1	Bool	20.1	TRUE	1 = Lock Messages on Zone Disabled
Bit2	Bool	20.2	TRUE	1 = Lock Messages on Module Test
Bit3	Bool	20.3	TRUE	1 = Lock Messages on Zone Test
Bit4	Bool	20.4	TRUE	1 = Lock Alarm Messages when Zone is out of Alarm
Bit5	Bool	20.5	TRUE	1 = Lock PreAlarm Messages for Zone is out of PreAlarm
Bit6	Bool	20.6	TRUE	1 = Lock Fault Messages for Zone is out of Fault
Bit7	Bool	20.7	FALSE	Reserved
Bit8	Bool	21.0	FALSE	Reserved
Bit9	Bool	21.1	FALSE	Reserved
Bit10	Bool	21.2	FALSE	Reserved
Bit11	Bool	21.3	FALSE	Reserved
Bit12	Bool	21.4	FALSE	Reserved
Bit13	Bool	21.5	FALSE	1 = Lock High priority scan when Zone is in PreAlarm
Bit14	Bool	21.6	FALSE	1 = Lock High priority scan when Zone is in Alarm
Bit15	Bool	21.7	FALSE	1 = Lock High priority scan when Zone in Fault
MSG_EVID_1	DWord	22.0	DW#16#0	Message 1 ID (Alarm)
MSG_EVID_2	DWord	26.0	DW#16#0	Message 2 ID (PreAlarm)
MSG_EVID_3	DWord	30.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	34.0	FALSE	Suppress Message 1

Name	Data Type	Address	Initial Value	Comment
MsgSup2	Bool	34.1	FALSE	Suppress Message 2
MsgSup3	Bool	34.2	FALSE	Suppress Message 3
MsgLock	Bool	34.3	FALSE	Message Lock
OUT				
PreAlarm	Bool	36.0	FALSE	Module PreAlarm
Alarm	Bool	36.1	FALSE	Module Activate
Fault	Bool	36.2	FALSE	Module Fault
Test	Bool	36.3	FALSE	Module in Test
Disabled	Bool	36.4	FALSE	Module Disabled
Zone_Test	Bool	36.5	FALSE	Zone in Test
Zone_Dis	Bool	36.6	FALSE	Zone Disabled
Zone_Num	Int	38.0	0	Zone Number
OOS	Bool	40.0	FALSE	Out of Service
Test_D	DInt	42.0	L#0	Last Test Date
Test_T	DInt	46.0	L#0	Last Test Time
Activ_D	DInt	50.0	L#0	Last Activate Date
Activ_T	DInt	54.0	L#0	Last Activate Time
Activ_Cyc	DInt	58.0	L#0	Activate Cycles
OS_PermOut	DWord	62.0	DW#16#FFFFFFFF	Operator permission: output for OS
OS_PermLog	DWord	66.0	DW#16#FFFFFFFF	Operator permission: output for OS
Status1	DWord	70.0	DW#16#0	Status1 Word
HMINAV	Struct	74.0		HMI Panel Interface
Status	Word	74.0	W#16#0	Status Word
DBNum	Int	76.0	0	DB Number
Btype	Int	78.0	0	Block Type
MsgSTS1	Word	80.0	W#16#0	Message 1 Status
MsgACK1	Bool	82.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	84.0	W#16#0	Message 2 Status
MsgACK2	Bool	86.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	88.0	W#16#0	Message 3 Status
MsgACK3	Bool	90.0	FALSE	Message 3 Acknowledge

1

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:= 'y' ;
Scl_GenerateReferenceData := 'y' ;
Scl_S7ServerActive:= 'y' ;
Scl_CreateObjectCode:= 'y' ;
Scl_OptimizeObjectCode:= 'y' ;
Scl_MonitorArrayLimits:= 'y' ;
Scl_CreateDebugInfo := 'y' ;
Scl_SetOKFlag:= 'n' ;
Scl_SetMaximumStringLength:= '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu          BYTE0 in memory
X24 : BOOL;                    // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL;                    // 0.7
// second highest Byte in accu  BYTE1 in memory
X16 : BOOL;                    // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL;                    // 1.7
// third highest Byte in accu   BYTE2 in memory
X8  : BOOL;                    // 2.0
X9  : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL;                    // 2.7
// lowest Byte in accu          BYTE3 in memory
X0  : BOOL;                    // 3.0
X1  : BOOL;
X2  : BOOL;
X3  : BOOL;
X4  : BOOL;
X5  : BOOL;
X6  : BOOL;
X7  : BOOL;                    // 3.7
END_STRUCT
END_TYPE
// Pointer Struct UDT104
TYPE PointSTR
STRUCT
ID          : BYTE;           // Always B#16#10
TypeCode   : BYTE;           // Pointer Type
Quantity   : WORD;           // Quantity of Data
DB_Num     : WORD;           // Data Block Number
PArea     : DWORD;           // Data Area Pointer
END_STRUCT;
END_TYPE

// Zone Status UDT105
TYPE ZTZoneSTS
STRUCT
PreAlarm   : BOOL;           //Zone in PreAlarm
Alarm      : BOOL;           //Zone in Alarm
Fault      : BOOL;           //Zone in Fault
Test       : BOOL;           //Zone in Test
Disabled   : BOOL;           //Zone Disabled
END_STRUCT
END_TYPE

// HMINAV Link Struct UDT106
TYPE HMINAVSTR
STRUCT
Status     : WORD;           // Status Word
DBNum     : INT;            // DB Number
Btype     : INT;            // Block Type
END_STRUCT;
END_TYPE

```

```

FUNCTION_BLOCK FB555
TITLE = 'ZITON Module'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'ZITMOD'
FAMILY: 'ZITON'
KNOW_HOW_PROTECT
{PO_Count := '1';
 S7_m_c := 'true';
 S7_alarm_ui := '1'}

//=====
VAR_INPUT
Mod_Id {
  S7_m_c := 'true';
  S7_visible := 'true';
  S7_edit := 'para';
  S7_link := 'false'} : INT; //Module Id (1..127)
Loop_Id {
  S7_m_c := 'true';
  S7_visible := 'true';
  S7_edit := 'para';
  S7_link := 'false'} : INT; //Loop Id (1..4)
RstActCycOp {
  Op_Level := '3';
  S7_m_c := 'true';
  S7_visible := 'false';
  S7_link := 'false'} : BOOL; //Reset Acitvate Cycles
Zone_STS {
  S7_edit := 'para';
  BLK_Jump := '1'} : ANY; //Connect to ZITONP2 ZoneX_STS
//Operator Permissions
OS_Perm {
  S7_visible:= 'false'} :
  STRUCT
  Bit0 : BOOL := 1; //1 = Operator can Reset Activated Cycles Totalizer
  Bit1 : BOOL := 1; //1 = Operator can Lock Messages
  Bit2 : BOOL := 1; //Reserved
  Bit3 : BOOL := 1; //Reserved
  Bit4 : BOOL := 1; //Reserved
  Bit5 : BOOL := 1; //Reserved
  Bit6 : BOOL := 1; //Reserved
  Bit7 : BOOL := 1; //Reserved
  Bit8 : BOOL := 1; //Reserved
  Bit9 : BOOL := 1; //Reserved
  Bit10 : BOOL := 1; //Reserved
  Bit11 : BOOL := 1; //Reserved
  Bit12 : BOOL := 1; //Reserved
  Bit13 : BOOL := 1; //Reserved
  Bit14 : BOOL := 1; //Reserved
  Bit15 : BOOL := 1; //Reserved
  Bit16 : BOOL := 1; //Reserved
  Bit17 : BOOL := 1; //Reserved
  Bit18 : BOOL := 1; //Reserved
  Bit19 : BOOL := 1; //Reserved
  Bit20 : BOOL := 1; //Reserved
  Bit21 : BOOL := 1; //Reserved
  Bit22 : BOOL := 1; //Reserved
  Bit23 : BOOL := 1; //Reserved
  Bit24 : BOOL := 1; //Reserved
  Bit25 : BOOL := 1; //Reserved
  Bit26 : BOOL := 1; //Reserved
  Bit27 : BOOL := 1; //Reserved
  Bit28 : BOOL := 1; //Reserved
  Bit29 : BOOL := 1; //Reserved
  Bit30 : BOOL := 1; //Reserved
  Bit31 : BOOL := 1; //Reserved
  END_STRUCT;
//Operator Permissions
dwOS_Perm AT OS_Perm : DWORD;
ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;
//FB Features
Feature {
  S7_visible:= 'false'}:
  STRUCT
  Bit0 : BOOL := 1; //1 = Lock Messages on Module Disabled
  Bit1 : BOOL := 1; //1 = Lock Messages on Zone Disabled
  Bit2 : BOOL := 1; //1 = Lock Messages on Module Test
  Bit3 : BOOL := 1; //1 = Lock Messages on Zone Test
  Bit4 : BOOL := 1; //1 = Lock Alarm Messages when Zone is out of Alarm
  Bit5 : BOOL := 1; //1 = Lock PreAlarm Messages for Zone is out of PreAlarm
  Bit6 : BOOL := 1; //1 = Lock Fault Messages for Zone is out of Fault
  Bit7 : BOOL; //Reserved
  Bit8 : BOOL; //Reserved
  Bit9 : BOOL; //Reserved
  Bit10 : BOOL; //Reserved
  Bit11 : BOOL; //Reserved

```

```

    Bit12 : BOOL; //Reserved
    Bit13 : BOOL; //1 = Lock High priority scan when Zone is in PreAlarm
    Bit14 : BOOL; //1 = Lock High priority scan when Zone is in Alarm
    Bit15 : BOOL; //1 = Lock High priority scan when Zone in Fault
    END_STRUCT; // Status of various features
//Messages Configuration
MSG_EVID_1 {
    S7_visible :='false';
    S7_link :='false';
    S7_param :='false';
    S7_server :='alarm_archiv';
    S7_a_type :='alarm_s' : DWORD:=0; //Message 1 ID (Alarm)
MSG_EVID_2 {
    S7_visible :='false';
    S7_link :='false';
    S7_param :='false';
    S7_server :='alarm_archiv';
    S7_a_type :='alarm_s' : DWORD:=0; //Message 2 ID (PreAlarm)
MSG_EVID_3 {
    S7_visible :='false';
    S7_link :='false';
    S7_param :='false';
    S7_server :='alarm_archiv';
    S7_a_type :='alarm_s' : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
    S7_visible :='false';
    S7_m_c :='true';
    S7_string_1 :='activ';
    S7_string_0 :='inactiv' : BOOL; //Suppress Message 1
MsgSup2 {
    S7_visible :='false';
    S7_m_c :='true';
    S7_string_0 :='inactiv';
    S7_string_1 :='activ' : BOOL; //Suppress Message 2
MsgSup3 {
    S7_visible :='false';
    S7_m_c :='true';
    S7_string_0 :='inactiv';
    S7_string_1 :='activ' : BOOL; //Suppress Message 3
MsgLock {
    Op_Level :='3';
    S7_m_c :='true';
    S7_visible :='false';
    S7_link :='false';
    S7_string_0 :='inactiv';
    S7_string_1 :='activ' : BOOL; //Message Lock
END_VAR
//=====
VAR_OUTPUT
PreAlarm {
    S7_dynamic :='true';
    S7_string_1 :='PreAlarm' : BOOL; //Module PreAlarm
Alarm {
    S7_m_c :='true';
    S7_dynamic :='true';
    S7_string_1 :='Alarm' : BOOL; //Module Activate
Fault {
    S7_dynamic :='true';
    S7_string_1 :='Fault' : BOOL; //Module Fault
Test {
    S7_dynamic :='true';
    S7_string_1 :='Test' : BOOL; //Module in Test
Disabled {
    S7_dynamic :='true';
    S7_string_1 :='Disabled' : BOOL; //Module Disabled
Zone_Test {
    S7_dynamic :='true';
    S7_string_1 :='Test' : BOOL; //Zone in Test
Zone_Dis {
    S7_dynamic :='true';
    S7_string_1 :='Disabled' : BOOL; //Zone Disabled
Zone_Num {
    S7_m_c :='true';
    S7_visible :='false' : INT; //Zone Number
OOS {
    S7_dynamic :='true';
    S7_string_1 :='OOS' : BOOL; //Out of Service
Test_D {
    S7_visible :='false';
    S7_m_c :='true' : DINT; //Last Test Date
Test_T {
    S7_visible :='false';
    S7_m_c :='true' : DINT; //Last Test Time
Activ_D {
    S7_visible :='false';
    S7_m_c :='true' : DINT; //Last Activate Date

```

```

Activ_T {
    S7_visible := 'false';
    S7_m_c := 'true'} : DINT; //Last Activate Time
Activ_Cyc {
    S7_visible := 'false';
    S7_m_c := 'true'} : DINT; //Activate Cycles
OS_PermOut {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD:=16#FFFFFFFF; //Operator permission: output for OS
ArrOS_PermOut AT OS_PermOut : ARRAY[0..3] OF BYTE;
OS_PermLog {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD:=16#FFFFFFFF; //Operator permission: output for OS
ArrOS_PermLog AT OS_PermLog : ARRAY[0..3] OF BYTE;
Status1 {
    S7_visible:= 'false';
    S7_m_c:= 'true'} : DWORD; //Status1 Word
HMINAV : HMINAVSTR; //HMI Panel Interface
//Messages Configuration
MsgSTS1 {
    S7_visible := 'false'} : WORD; //Message 1 Status
MsgACK1 {
    S7_visible := 'false'} : BOOL; //Message 1 Acknowledge
MsgSTS2 {
    S7_visible := 'false'} : WORD; //Message 2 Status
MsgACK2 {
    S7_visible := 'false'} : BOOL; //Message 2 Acknowledge
MsgSTS3 {
    S7_visible := 'false'} : WORD; //Message 3 Status
MsgACK3 {
    S7_visible := 'false'} : BOOL; //Message 3 Acknowledge
END_VAR

//=====
CONST
ZBAddrCPU := 72; // Zones Byte Base Address
MSL1BAddrCPU := 586; // Modules Status Loop1 Byte Base Address
MSL2BAddrCPU := 842; // Modules Status Loop2 Byte Base Address
MSL3BAddrCPU := 1098; // Modules Status Loop3 Byte Base Address
MSL4BAddrCPU := 1354; // Modules Status Loop4 Byte Base Address
AckZBAddrCPU := 1610; // ZonesSTS Acknowledgements Base Address
xHPScanAddrCPU := 1674; // High priority scan Command Base Address
ZBAddrCP := 62; // Zones Byte Base Address
MSL1BAddrCP := 544; // Modules Status Loop1 Byte Base Address
MSL2BAddrCP := 800; // Modules Status Loop2 Byte Base Address
MSL3BAddrCP := 1056; // Modules Status Loop3 Byte Base Address
MSL4BAddrCP := 1312; // Modules Status Loop4 Byte Base Address
AckZBAddrCP := 1568; // ZonesSTS Acknowledgements Base Address
xHPScanAddrCP := 1632; // High priority scan Command Base Address
END_CONST

//=====
VAR
FBID : WORD;
// Panel type
ZBAddr : INT; // Zones Byte Base Address
MSL1BAddr : INT; // Modules Status Loop1 Byte Base Address
MSL2BAddr : INT; // Modules Status Loop2 Byte Base Address
MSL3BAddr : INT; // Modules Status Loop3 Byte Base Address
MSL4BAddr : INT; // Modules Status Loop4 Byte Base Address
AckZBAddr : INT; // ZonesSTS Acknowledgements Base Address
xHPScanAddr : INT; // High priority scan Command Base Address
xZonSTS : ZTZoneSTS; // Zone Status
xModSTS : WORD; // Module Status
xBitModSTS AT xModSTS : SBits16inWORD; // Module as Bits
xHMISTS : WORD; // HMI Status
xBitHMISTS AT xHMISTS : SBits16inWORD; // HMI Status as Bits
xMZonAlarm : BOOL; // Save Zone in Alarm
xMZonPreAlm : BOOL; // Save Zone in PreAlarm
xMZonFault : BOOL; // Save Zone in Fault
xAckZone : // Zone Status Acknowledgement
STRUCT
    HB : BYTE := BYTE#00;
    PreAlarm : BOOL; //Zone in PreAlarm
    Alarm : BOOL; //Zone in Alarm
    Fault : BOOL; //Zone in Fault
    X3 : BOOL;
    X4 : BOOL;
    X5 : BOOL;
    X6 : BOOL;
    X7 : BOOL;
END_STRUCT;

CMP_ID : DWORD; // Area code
xMsgCond1 : BOOL; // Condition MSG1
xMsgCond2 : BOOL; // Condition MSG2
xMsgCond3 : BOOL; // Condition MSG3

```

```

xMsgTrig1      : BOOL;          // Trigger MSG1
xMsgTrig2      : BOOL;          // Trigger MSG2
xMsgTrig3      : BOOL;          // Trigger MSG3
xMsgArch1      : INT;           // MSG1 Status
xMsgArch2      : INT;           // MSG2 Status
xMsgArch3      : INT;           // MSG3 Status
AuxValue:      // Msg Aux Value
  STRUCT
    Zone        : INT;          // Msg Zone
    DBNum       : WORD;         // Msg DB num
  END_STRUCT;
xRetSts        : BOOL;          // Msg Status
xMTimeActiv    : BOOL;          // Mem Activ Time flag
xHPDWrd        : DWORD;         // High Priority Scan Word
ArrxHPDWrd AT xHPDWrd : ARRAY[1..32] OF BOOL;
END_VAR
//=====
VAR_TEMP
xAux_Err       : INT;           // Auxiliar Return Function Error
xAddr          : DINT;
xFMsgLock      : BOOL;         // MsgLock by Features Bits
xMsg1Zone      : BOOL;         // MsgLock1 by Zone Alarm Status
xMsg2Zone      : BOOL;         // MsgLock2 by Zone PreAlarm Status
xMsg3Zone      : BOOL;         // MsgLock3 by Zone Fault Status

// Structures to copy data areas
PPanel         : ANY;          // Pointer to Panel Block
PPan_STR AT PPanel : PointSTR; // Pointer Structure
PLocal        : ANY;          // Pointer to Local Var
PLoc_STR AT PLocal : PointSTR; // Pointer Structure

// Variable for Operator Permissions
xdwOS_PerLog   : DWORD; // Temporary variable for the value of OS_PerLog
strxOS_PerLog AT xdwOS_PerLog :
  STRUCT
    Bit0        : BOOL; //1 = Operator can Reset Activated Cycle Totalizer
    Bit1        : BOOL; //1 = Operator can Lock Messages
    Bit2        : BOOL; //Not used
    Bit3        : BOOL; //Not used
    Bit4        : BOOL; //Not used
    Bit5        : BOOL; //Not used
    Bit6        : BOOL; //Not used
    Bit7        : BOOL; //Not used
    Bit8        : BOOL; //Not used
    Bit9        : BOOL; //Not used
    Bit10       : BOOL; //Not used
    Bit11       : BOOL; //Not used
    Bit12       : BOOL; //Not used
    Bit13       : BOOL; //Not used
    Bit14       : BOOL; //Not used
    Bit15       : BOOL; //Not used
    Bit16       : BOOL; //Not used
    Bit17       : BOOL; //Not used
    Bit18       : BOOL; //Not used
    Bit19       : BOOL; //Not used
    Bit20       : BOOL; //Not used
    Bit21       : BOOL; //Not used
    Bit22       : BOOL; //Not used
    Bit23       : BOOL; //Not used
    Bit24       : BOOL; //Not used
    Bit25       : BOOL; //Not used
    Bit26       : BOOL; //Not used
    Bit27       : BOOL; //Not used
    Bit28       : BOOL; //Not used
    Bit29       : BOOL; //Not used
    Bit30       : BOOL; //Not used
    Bit31       : BOOL; //Not used
  END_STRUCT;
ArrxOS_PerLog AT xdwOS_PerLog : ARRAY[0..3] OF BYTE;

// Variable for status
dwStatus       : DWORD;
ArrBitStatus AT dwStatus : SBits32inDWORD;

// Variable for Maintenance Time/Date
dtCPU_Time     : DATE_AND_TIME; // Actual CPU time
dtCPU_Time_st AT dtCPU_Time :
  STRUCT
    b0 : BYTE;
    b1 : BYTE;
    b2 : BYTE;
    b3 : BYTE;
    b4 : BYTE;
    b5 : BYTE;
    b6 : BYTE;
    b7 : BYTE;
  END_STRUCT;

```

```

SnREAD_CLK          : INT;    // Error information READ_CLK;
cDate               : DINT;   // Current Date
cTime              : DINT;   // Current Time
TimeActiv          : BOOL;   // Activate Time flag
END_VAR

// CODE =====
BEGIN
// Narrow ranges
IF Mod_Id>127 THEN Mod_Id :=127; END_IF;
IF Mod_Id<1 THEN Mod_Id :=1; END_IF;
IF Loop_Id>4 THEN Loop_Id :=4; END_IF;
IF Loop_Id<1 THEN Loop_Id :=1; END_IF;

//Panel Type to knows base addresses
PPanel              := Zone_STS; // Pointer to Panel Block
PPan_STR.TypeCode  := BYTE#04; // Work with WORDS
PPan_STR.Quantity  := WORD#01; // 1 Word of Data
PPan_STR.PArea     := DW#16#84000000; // First Word of the DB
PLocal             := FBID;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ZoneX_STS
IF FBID=WORD#01 THEN // ZITONCP
    ZBAddr          := ZBAddrCP;
    MSL1BAddr       := MSL1BAddrCP;

    MSL2BAddr       := MSL2BAddrCP;
    MSL3BAddr       := MSL3BAddrCP;
    MSL4BAddr       := MSL4BAddrCP;
    AckZBAddr       := AckZBAddrCP;
    xHPScanAddr     := xHPScanAddrCP;
ELSE
    ZBAddr          := ZBAddrCPU;
    MSL1BAddr       := MSL1BAddrCPU;
    MSL2BAddr       := MSL2BAddrCPU;
    MSL3BAddr       := MSL3BAddrCPU;
    MSL4BAddr       := MSL4BAddrCPU;
    AckZBAddr       := AckZBAddrCPU;
    xHPScanAddr     := xHPScanAddrCPU;
END_IF;

//Zone outputs
PPanel              := Zone_STS; // Pointer to Panel Block
PPan_STR.TypeCode  := BYTE#02; // Work with BYTES
PPan_STR.Quantity  := WORD#01; // 1 Byte of Data
PLocal             := xZonSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ZoneX_STS
Zone_Dis           := xZonSTS.Disabled; // Zone Disabled
Zone_Test          := xZonSTS.Test; // Zone Test
Zone_Num :=1+((DWORD_TO_INT(PPan_STR.PArea)-ZBAddr*8)/16); // Calculate Zone Number

CMP_ID := WORD_TO_DWORD(PLoc_STR.DB_Num); // DBNum for Messages
AuxValue.Zone := Zone_Num;
AuxValue.DBNum := PLoc_STR.DB_Num;

//Module Outputs
PPan_STR.TypeCode := BYTE#04; // Work with WORDS
PPan_STR.Quantity := WORD#01; // 1 Word of Data
CASE (Loop_Id) OF
    1: xAddr := (MSL1BAddr+2*(Mod_Id-1))*8; // Addr in bits for the mod reg
    2: xAddr := (MSL2BAddr+2*(Mod_Id-1))*8; // Addr in bits for the mod reg
    3: xAddr := (MSL3BAddr+2*(Mod_Id-1))*8; // Addr in bits for the mod reg
    4: xAddr := (MSL4BAddr+2*(Mod_Id-1))*8; // Addr in bits for the mod reg
END_CASE;
PPan_STR.PArea := DINT_TO_DWORD(xAddr) OR DW#16#84000000;
PLocal := xModSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ModuleSTS[x]
PreAlarm := xBitModSTS.X0;
Alarm := xBitModSTS.X1;
Fault := xBitModSTS.X2;
Test := xBitModSTS.X3;
Disabled := xBitModSTS.X4;
OOS := xBitModSTS.X15;

//High priority scan
IF (NOT (Feature.Bit13) AND xZonSTS.PreAlarm AND NOT (xMzonPreAlm)) OR
(NOT (Feature.Bit14) AND xZonSTS.Alarm AND NOT (xMzonAlarm)) OR
(NOT (Feature.Bit15) AND xZonSTS.Fault AND NOT (xMzonFault)) THEN
    // Bit14 indicates to the Panel that the module is connected to the affected Zone
    xBitModSTS.X14 := 1;
    xAux_Err :=BLKMOV(SRCBLK := PLocal, DSTBLK := PPanel); // Write ModuleSTS[x] setting b14

    PPan_STR.TypeCode := BYTE#06; // Work with DWORD
    PPan_STR.Quantity := WORD#01; // 1 DWORD of Data
    xAddr := xHPScanAddr*8; // Addr in bits
    PPan_STR.PArea := DINT_TO_DWORD(xAddr) OR DW#16#84000000;
    PLocal := xHPDWrd;
    xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve HPScanCMD from panel

```

```

    ArrxHPDWrd[Zone_Num]:=TRUE; // Set Zone Bit
    xAux_Err :=BLKMÖV(SRCBLK := PLocal, DSTBLK := PPanel); // Write HPScanCMD to panel
END_IF;
xMZonPreAlm := xZonSTS.PreAlarm;
xMZonAlarm := xZonSTS.Alarm;
xMZonFault := xZonSTS.Fault;

//OS Permissives
ArrOS_PermOut[3] := ArrOS_Perm[0];
ArrOS_PermOut[2] := ArrOS_Perm[1];
ArrOS_PermOut[1] := ArrOS_Perm[2];
ArrOS_PermOut[0] := ArrOS_Perm[3];
IF OOS THEN
    OS_PermLog := DW#16#30 AND OS_PermOut; //MsgLock & Reset Totalizer Mask
ELSE
    xdwOS_PermLog := dwOS_Perm;
    IF Activ_Cyc=0 THEN strxOS_PermLog.Bit4 := false; //Activated Cycle Tot Permission
    ELSE strxOS_PermLog.Bit4 := OS_Perm.Bit4; END_IF;
    strxOS_PermLog.Bit5 := OS_Perm.Bit5; //Lock Messages Permission
    //OS_PermLog
    ArrOS_PermLog[3] := ArrxOS_PermLog[0];
    ArrOS_PermLog[2] := ArrxOS_PermLog[1];
    ArrOS_PermLog[1] := ArrxOS_PermLog[2];
    ArrOS_PermLog[0] := ArrxOS_PermLog[3];
END_IF;

//Status1 DWord
dwStatus := DWORD#0;
ArrBitStatus.X0 := PreAlarm;
ArrBitStatus.X1 := Alarm;
ArrBitStatus.X2 := Fault;
ArrBitStatus.X3 := Test;
ArrBitStatus.X4 := Disabled;
ArrBitStatus.X5 := OOS;
ArrBitStatus.X6 := Zone_Test;
ArrBitStatus.X7 := Zone_Dis;
ArrBitStatus.X8 := MsgLock;
ArrBitStatus.X9 := MsgSup1;
ArrBitStatus.X10 := MsgSup2;
ArrBitStatus.X11 := MsgSup3;
ArrBitStatus.X12 := xMsgCond1;
ArrBitStatus.X13 := xMsgCond2;
ArrBitStatus.X14 := xMsgCond3;
Status1 := dwStatus;
//Alarms Managment
xFMsgLock := (Disabled AND Feature.Bit0) OR //Lock Messages on Module Disabled
              (Zone_Dis AND Feature.Bit1) OR //Lock Messages on Zone Disabled
              (Test AND Feature.Bit2) OR //Lock Messages on Module Test
              (Zone_Test AND Feature.Bit3); //Lock Messages on Zone Test

xMsg1Zone := NOT(xZonSTS.Alarm) AND Feature.Bit4; //Lock Alarm Messages when Zone is out of Alarm
xMsg2Zone := NOT(xZonSTS.PreAlarm) AND Feature.Bit5; //Lock PreAlarm Mess when Zone is out PreAlarm
xMsg3Zone := NOT(xZonSTS.Fault) AND Feature.Bit6; //Lock Fault Mess when Zone is out Fault
xMsgCond1 := Alarm AND NOT(MsgSup1 OR MsgLock OR xFMsgLock OR xMsg1Zone);
xMsgCond2 := PreAlarm AND NOT(MsgSup2 OR MsgLock OR xFMsgLock OR xMsg2Zone);
xMsgCond3 := Fault AND NOT(MsgSup3 OR MsgLock OR xFMsgLock OR xMsg3Zone);
//Activate MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
    MsgSTS1:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond1
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_1
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xMsgArch1:=ALARM_SC(
    EV_ID := MSG_EVID_1
    ,STATE := xRetSts
    ,Q_STATE := MsgACK1);
//PreAlarm MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
    MsgSTS2:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond2
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_2
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xMsgArch2:=ALARM_SC(
    EV_ID := MSG_EVID_2
    ,STATE := xRetSts
    ,Q_STATE := MsgACK2);
//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN

```

```

MsgSTS3:=INT_TO_WORD(ALARM_DQ(
    SIG      := xMsgCond3
    ,ID      := W#16#EEEE
    ,EV_ID   := MSG_EVID_3
    ,CMP_ID  := CMP_ID
    ,SD      := AuxVValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xMsgArch3:=ALARM_SC(
    EV_ID    := MSG_EVID_3
    ,STATE   := xRetSts
    ,Q_STATE := MsgACK3);
//Zone status Acknowledgement
PPan_STR.TypeCode := BYTE#02;           // Work with BYTES
PPan_STR.Quantity := WORD#02;          // 2 BYTES of Data
xAddr := (AckZBAddr+2*(Zone_Num-1))*8; // Addr in bits

PPan_STR.PArea   := DINT_TO_DWORD(xAddr) OR DW#16#84000000;
PLocal          := xAckZone;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve AckZonesSTS[x]

xAckZone.Alarm   := xAckZone.Alarm OR xMsgCond1;
xAckZone.PreAlarm := xAckZone.PreAlarm OR xMsgCond2;
xAckZone.Fault   := xAckZone.Fault OR xMsgCond3;
xAux_Err :=BLKMOV(SRCBLK := PLocal, DSTBLK := PPanel); // Write AckZonesSTS[x]
//Maintenance Managment
TimeActiv := NOT(Zone_Test) AND NOT(Test) AND Alarm;
IF (TimeActiv AND NOT(xMTimeActiv)) THEN
    SnREAD_CLK := READ_CLK(CDT:=dtCPU_Time);
    cDate      := (BCD_TO_DINT(dtCPU_Time_st.b0)+10);
    cDate      := (cDate*365)+(cDate/4);
    CASE (BCD_TO_INT(dtCPU_Time_st.b1)) OF
        1 : cDate:=cDate+0;
        2 : cDate:=cDate+31;
        3 : cDate:=cDate+59;
        4 : cDate:=cDate+90;
        5 : cDate:=cDate+120;
        6 : cDate:=cDate+151;
        7 : cDate:=cDate+181;
        8 : cDate:=cDate+212;
        9 : cDate:=cDate+243;
       10 : cDate:=cDate+273;
       11 : cDate:=cDate+304;
       12 : cDate:=cDate+334;
    END_CASE;
    cDate := cDate+BCD_TO_DINT(dtCPU_Time_st.b2);
    cTime := BCD_TO_DINT(dtCPU_Time_st.b3)*3600000;
    cTime := cTime+BCD_TO_DINT(dtCPU_Time_st.b4)*60000;
    cTime := cTime+BCD_TO_DINT(dtCPU_Time_st.b5)*1000;
    IF TimeActiv AND NOT(xMTimeActiv) THEN
        Activ_D := cDate;
        Activ_T := cTime;
        Activ_Cyc := Activ_Cyc+1;
    END_IF;
END_IF;
xMTimeActiv := TimeActiv; // Time Activ updated
IF RstActCycOp THEN // Reset Activate Totalizer
    Activ_Cyc := 0;
    RstActCycOp := false;
END_IF;

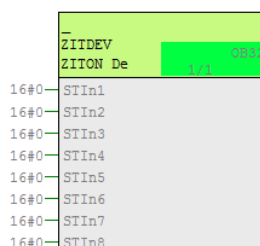
//HMINAV
//Status
xBithMISTS.X0 := Alarm;
xBithMISTS.X1 := PreAlarm;
xBithMISTS.X2 := Fault;
xBithMISTS.X3 := false;
//Msg activated
xBithMISTS.X4 := xMsgCond1;
xBithMISTS.X5 := xMsgCond2;
xBithMISTS.X6 := xMsgCond3;
xBithMISTS.X7 := false;
//Msg Acknowled
xBithMISTS.X8 := MsgACK1;
xBithMISTS.X9 := MsgACK2;
xBithMISTS.X10 := MsgACK3;
xBithMISTS.X11 := false;
//Msg in Archive (true if msg in Archive)
xBithMISTS.X12 := (xMsgArch1=0);
xBithMISTS.X13 := (xMsgArch2=0);
xBithMISTS.X14 := (xMsgArch3=0);
xBithMISTS.X15 := false;
HMINAV.Status := xHMISTS;
HMINAV.DBNum := WORD_TO_INT(AuxVValue.DBNum);
HMINAV.Btype := 11; //ZITMOD code=11

END_FUNCTION_BLOCK

```

2.18 FB556: ZITDEV

- Representación Gráfica



- Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
STIn1	DWord	0.0	DW#16#0	Connect to Module Status 1
STIn2	DWord	4.0	DW#16#0	Connect to Module Status 1
STIn3	DWord	8.0	DW#16#0	Connect to Module Status 1
STIn4	DWord	12.0	DW#16#0	Connect to Module Status 1
STIn5	DWord	16.0	DW#16#0	Connect to Module Status 1
STIn6	DWord	20.0	DW#16#0	Connect to Module Status 1
STIn7	DWord	24.0	DW#16#0	Connect to Module Status 1
STIn8	DWord	28.0	DW#16#0	Connect to Module Status 1
STIn9	DWord	32.0	DW#16#0	Connect to Module Status 1
STIn10	DWord	36.0	DW#16#0	Connect to Module Status 1
STIn11	DWord	40.0	DW#16#0	Connect to Module Status 1
STIn12	DWord	44.0	DW#16#0	Connect to Module Status 1
STIn13	DWord	48.0	DW#16#0	Connect to Module Status 1
STIn14	DWord	52.0	DW#16#0	Connect to Module Status 1
STIn15	DWord	56.0	DW#16#0	Connect to Module Status 1
STIn16	DWord	60.0	DW#16#0	Connect to Module Status 1
OS_Perm	Struct	64.0		Operator Permissions
Bit0	Bool	64.0	TRUE	Reserved
Bit1	Bool	64.1	TRUE	Reserved
Bit2	Bool	64.2	TRUE	Reserved
Bit3	Bool	64.3	TRUE	Reserved
Bit4	Bool	64.4	TRUE	Reserved
Bit5	Bool	64.5	TRUE	1 = Operator can Lock Messages
Bit6	Bool	64.6	TRUE	Reserved
Bit7	Bool	64.7	TRUE	Reserved
Bit8	Bool	65.0	TRUE	Reserved
Bit9	Bool	65.1	TRUE	Reserved
Bit10	Bool	65.2	TRUE	Reserved
Bit11	Bool	65.3	TRUE	Reserved
Bit12	Bool	65.4	TRUE	Reserved
Bit13	Bool	65.5	TRUE	Reserved
Bit14	Bool	65.6	TRUE	Reserved
Bit15	Bool	65.7	TRUE	Reserved
Bit16	Bool	66.0	TRUE	Reserved
Bit17	Bool	66.1	TRUE	Reserved
Bit18	Bool	66.2	TRUE	Reserved
Bit19	Bool	66.3	TRUE	Reserved
Bit20	Bool	66.4	TRUE	Reserved
Bit21	Bool	66.5	TRUE	Reserved
Bit22	Bool	66.6	TRUE	Reserved
Bit23	Bool	66.7	TRUE	Reserved
Bit24	Bool	67.0	TRUE	Reserved
Bit25	Bool	67.1	TRUE	Reserved
Bit26	Bool	67.2	TRUE	Reserved
Bit27	Bool	67.3	TRUE	Reserved
Bit28	Bool	67.4	TRUE	Reserved
Bit29	Bool	67.5	TRUE	Reserved
Bit30	Bool	67.6	TRUE	Reserved
Bit31	Bool	67.7	TRUE	Reserved
Feature	Struct	68.0		Status of various features
Bit0	Bool	68.0	FALSE	1 = Lock Messages for In1
Bit1	Bool	68.1	FALSE	1 = Lock Messages for In2
Bit2	Bool	68.2	FALSE	1 = Lock Messages for In3
Bit3	Bool	68.3	FALSE	1 = Lock Messages for In4
Bit4	Bool	68.4	FALSE	1 = Lock Messages for In5
Bit5	Bool	68.5	FALSE	1 = Lock Messages for In6
Bit6	Bool	68.6	FALSE	1 = Lock Messages for In7
Bit7	Bool	68.7	FALSE	1 = Lock Messages for In8

Name	Data Type	Address	Initial Value	Comment
Bit8	Bool	69.0	FALSE	1 = Lock Messages for In9
Bit9	Bool	69.1	FALSE	1 = Lock Messages for In10
Bit10	Bool	69.2	FALSE	1 = Lock Messages for In11
Bit11	Bool	69.3	FALSE	1 = Lock Messages for In12
Bit12	Bool	69.4	FALSE	1 = Lock Messages for In13
Bit13	Bool	69.5	FALSE	1 = Lock Messages for In14
Bit14	Bool	69.6	FALSE	1 = Lock Messages for In15
Bit15	Bool	69.7	FALSE	1 = Lock Messages for In16
MSG_EVID_1	DWord	70.0	DW#16#0	Message 1 ID (Activate)
MSG_EVID_2	DWord	74.0	DW#16#0	Message 2 ID (PreAlarm)
MSG_EVID_3	DWord	78.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	82.0	FALSE	Suppress Message 1
MsgSup2	Bool	82.1	FALSE	Suppress Message 2
MsgSup3	Bool	82.2	FALSE	Suppress Message 3
MsgLock	Bool	82.3	FALSE	Message Lock
OUT				
OS_PermOut	DWord	84.0	DW#16#FFFFFFF	Operator permission: output for OS
OS_PermLog	DWord	88.0	DW#16#FFFFFFF	Operator permission: output for OS
Status1	DWord	92.0	DW#16#0	Status1 Word
MsgSTS1	Word	96.0	W#16#0	Message 1 Status
MsgACK1	Bool	98.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	100.0	W#16#0	Message 2 Status
MsgACK2	Bool	102.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	104.0	W#16#0	Message 3 Status
MsgACK3	Bool	106.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:= 'y' ;
Scl_GenerateReferenceData := 'y' ;
Scl_S7ServerActive:= 'y' ;
Scl_CreateObjectCode:= 'y' ;
Scl_OptimizeObjectCode:= 'y' ;
Scl_MonitorArrayLimits:= 'y' ;
Scl_CreateDebugInfo := 'y' ;
Scl_SetOKFlag:= 'n' ;
Scl_SetMaximumStringLength:= '254'
}
// SBits32inDWORD UDT100
TYPE SBits32inDWORD
STRUCT
// highest Byte in accu BYTE0 in memory
X24 : BOOL; // 0.0
X25 : BOOL;
X26 : BOOL;
X27 : BOOL;
X28 : BOOL;
X29 : BOOL;
X30 : BOOL;
X31 : BOOL; // 0.7
// second highest Byte in accu BYTE1 in memory
X16 : BOOL; // 1.0
X17 : BOOL;
X18 : BOOL;
X19 : BOOL;
X20 : BOOL;
X21 : BOOL;
X22 : BOOL;
X23 : BOOL; // 1.7
// third highest Byte in accu BYTE2 in memory
X8 : BOOL; // 2.0
X9 : BOOL;
X10 : BOOL;
X11 : BOOL;
X12 : BOOL;
X13 : BOOL;
X14 : BOOL;
X15 : BOOL; // 2.7
// lowest Byte in accu BYTE3 in memory
X0 : BOOL; // 3.0
X1 : BOOL;
X2 : BOOL;
X3 : BOOL;
X4 : BOOL;
X5 : BOOL;
X6 : BOOL;
X7 : BOOL; // 3.7
END_STRUCT
END_TYPE

// Pointer Struct UDT104
TYPE PointSTR
STRUCT
ID : BYTE; // Always B#16#10
TypeCode : BYTE; // Pointer Type
Quantity : WORD; // Quantity of Data
DB_Num : WORD; // Data Block Number
PArea : DWORD; // Data Area Pointer
END_STRUCT;
END_TYPE

FUNCTION_BLOCK FB556
TITLE = 'ZITON Device'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'ZITDEV'
FAMILY: 'ZITON'
KNOW_HOW_PROTECT
{PO_Count := '1';
S7_m_c := 'true';
S7_alarm_ui := '1'}

//=====
VAR_INPUT
STIn1 {
S7_m_c := 'true';
S7_edit := 'para';
BLK_Jump := '1'} : DWORD; //Connect to Module Status1
BSTIn1 AT STIn1 : SBits32inDWORD;
STIn2 {
S7_m_c := 'true';
S7_edit := 'para';
BLK_Jump := '1'} : DWORD; //Connect to Module Status1

```

```

BSTIn2 AT STIn2          : SBits32inDWORD;
STIn3 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1'}      : DWORD;          //Connect to Module Status1
BSTIn3 AT STIn3          : SBits32inDWORD;
STIn4 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1'}      : DWORD;          //Connect to Module Status1
BSTIn4 AT STIn4          : SBits32inDWORD;
STIn5 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1'}      : DWORD;          //Connect to Module Status1
BSTIn5 AT STIn5          : SBits32inDWORD;
STIn6 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1'}      : DWORD;          //Connect to Module Status1
BSTIn6 AT STIn6          : SBits32inDWORD;
STIn7 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1'}      : DWORD;          //Connect to Module Status1
BSTIn7 AT STIn7          : SBits32inDWORD;

STIn8 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1'}      : DWORD;          //Connect to Module Status1
BSTIn8 AT STIn8          : SBits32inDWORD;
STIn9 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn9 AT STIn9          : SBits32inDWORD;
STIn10 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn10 AT STIn10        : SBits32inDWORD;
STIn11 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn11 AT STIn11        : SBits32inDWORD;
STIn12 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn12 AT STIn12        : SBits32inDWORD;
STIn13 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn13 AT STIn13        : SBits32inDWORD;
STIn14 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn14 AT STIn14        : SBits32inDWORD;
STIn15 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn15 AT STIn15        : SBits32inDWORD;
STIn16 {
  S7_m_c    :='true';
  S7_edit   :='para';
  BLK_Jump  := '1';
  S7_visible :='false'} : DWORD;          //Connect to Module Status1
BSTIn16 AT STIn16        : SBits32inDWORD;
//Operator Permissions
OS_Perm {
  S7_visible :='false'}   :
  STRUCT
  Bit0      : BOOL := 1;//Reserved
  Bit1      : BOOL := 1;//Reserved
  Bit2      : BOOL := 1;//Reserved

```

```

Bit3   : BOOL := 1; //Reserved
Bit4   : BOOL := 1; //Reserved
Bit5   : BOOL := 1; //1 = Operator can Lock Messages
Bit6   : BOOL := 1; //Reserved
Bit7   : BOOL := 1; //Reserved
Bit8   : BOOL := 1; //Reserved
Bit9   : BOOL := 1; //Reserved
Bit10  : BOOL := 1; //Reserved
Bit11  : BOOL := 1; //Reserved
Bit12  : BOOL := 1; //Reserved
Bit13  : BOOL := 1; //Reserved
Bit14  : BOOL := 1; //Reserved
Bit15  : BOOL := 1; //Reserved
Bit16  : BOOL := 1; //Reserved
Bit17  : BOOL := 1; //Reserved
Bit18  : BOOL := 1; //Reserved
Bit19  : BOOL := 1; //Reserved
Bit20  : BOOL := 1; //Reserved
Bit21  : BOOL := 1; //Reserved
Bit22  : BOOL := 1; //Reserved
Bit23  : BOOL := 1; //Reserved
Bit24  : BOOL := 1; //Reserved
Bit25  : BOOL := 1; //Reserved
Bit26  : BOOL := 1; //Reserved
Bit27  : BOOL := 1; //Reserved
Bit28  : BOOL := 1; //Reserved
Bit29  : BOOL := 1; //Reserved
Bit30  : BOOL := 1; //Reserved
Bit31  : BOOL := 1; //Reserved
END_STRUCT;
//Operator Permissions
dwOS_Perm AT OS_Perm : DWORD;
ArrOS_Perm AT OS_Perm : ARRAY[0..3] OF BYTE;
//FB Features
Feature {
  S7_visible:='false':
  STRUCT
    Bit0   : BOOL; //1 = Lock Messages for In1
    Bit1   : BOOL; //1 = Lock Messages for In2
    Bit2   : BOOL; //1 = Lock Messages for In3
    Bit3   : BOOL; //1 = Lock Messages for In4
    Bit4   : BOOL; //1 = Lock Messages for In5
    Bit5   : BOOL; //1 = Lock Messages for In6
    Bit6   : BOOL; //1 = Lock Messages for In7
    Bit7   : BOOL; //1 = Lock Messages for In8
    Bit8   : BOOL; //1 = Lock Messages for In9
    Bit9   : BOOL; //1 = Lock Messages for In10
    Bit10  : BOOL; //1 = Lock Messages for In11
    Bit11  : BOOL; //1 = Lock Messages for In12
    Bit12  : BOOL; //1 = Lock Messages for In13
    Bit13  : BOOL; //1 = Lock Messages for In14
    Bit14  : BOOL; //1 = Lock Messages for In15
    Bit15  : BOOL; //1 = Lock Messages for In16
  END_STRUCT;
// Status of various features
//Messages Configuration
MSG_EVID_1 {
  S7_visible := 'false';

  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s' : DWORD:=0; //Message 1 ID (Activate)
MSG_EVID_2 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s' : DWORD:=0; //Message 2 ID (PreAlarm)
MSG_EVID_3 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s' : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
  S7_visible := 'false';
  S7_m_c := 'true';
  S7_string_1 := 'activ';
  S7_string_0 := 'inactiv' : BOOL; //Suppress Message 1
MsgSup2 {
  S7_visible := 'false';
  S7_m_c := 'true';
  S7_string_0 := 'inactiv';
  S7_string_1 := 'activ' : BOOL; //Suppress Message 2
MsgSup3 {
  S7_visible := 'false';

```

```

S7_m_c      := 'true';
S7_string_0 := 'inactiv';
S7_string_1 := 'activ' } : BOOL; // Suppress Message 3
MsgLock {
  Op_Level   := '3';
  S7_m_c     := 'true';
  S7_visible := 'false';
  S7_link    := 'false';
  S7_string_0 := 'inactiv';
  S7_string_1 := 'activ' } : BOOL; // Message Lock
END_VAR

//=====
VAR_OUTPUT
OS_PermOut {
  S7_visible := 'false';
  S7_m_c := 'true' } : DWORD := 16#FFFFFFFF; // Operator permission: output for OS
ArrOS_PermOut AT OS_PermOut : ARRAY[0..3] OF BYTE;
OS_PermLog {
  S7_visible := 'false';
  S7_m_c := 'true' } : DWORD := 16#FFFFFFFF; // Operator permission: output for OS
ArrOS_PermLog AT OS_PermLog : ARRAY[0..3] OF BYTE;
Status1 {
  S7_visible := 'false';
  S7_m_c := 'true' } : DWORD; // Status1 Word
//Messages Configuration
MsgSTS1 {
  S7_visible := 'false' } : WORD; // Message 1 Status
MsgACK1 {
  S7_visible := 'false' } : BOOL; // Message 1 Acknowledge
MsgSTS2 {
  S7_visible := 'false' } : WORD; // Message 2 Status
MsgACK2 {
  S7_visible := 'false' } : BOOL; // Message 2 Acknowledge
MsgSTS3 {
  S7_visible := 'false' } : WORD; // Message 3 Status
MsgACK3 {
  S7_visible := 'false' } : BOOL; // Message 3 Acknowledge
END_VAR

//=====
VAR
// Variable for Messages
CMP_ID : DWORD; // Area code
xMsgCond1 : BOOL; // Condition MSG1
xMsgCond2 : BOOL; // Condition MSG2
xMsgCond3 : BOOL; // Condition MSG3
xMsgTrig1 : BOOL; // Trigger MSG1
xMsgTrig2 : BOOL; // Trigger MSG2
xMsgTrig3 : BOOL; // Trigger MSG3
xRetVal : INT;
xRetSts : BOOL; // Msg Status
AuxValue:
  STRUCT
    Value : INT; // Msg Aux Values
  END_STRUCT;
END_VAR

//=====
VAR_TEMP
// Variable for Operator Permissions
xdwOS_PermLog : DWORD; // Temporary variable for the value of OS_PermLog
strxOS_PermLog AT xdwOS_PermLog :
  STRUCT
    Bit0 : BOOL; //Not used
    Bit1 : BOOL; //Not used
    Bit2 : BOOL; //Not used
    Bit3 : BOOL; //Not used
    Bit4 : BOOL; //Not used
    Bit5 : BOOL; //1 = Operator can Lock Messages
    Bit6 : BOOL; //Not used
    Bit7 : BOOL; //Not used
    Bit8 : BOOL; //Not used
    Bit9 : BOOL; //Not used
    Bit10 : BOOL; //Not used
    Bit11 : BOOL; //Not used
    Bit12 : BOOL; //Not used
    Bit13 : BOOL; //Not used
    Bit14 : BOOL; //Not used
    Bit15 : BOOL; //Not used
    Bit16 : BOOL; //Not used
    Bit17 : BOOL; //Not used
    Bit18 : BOOL; //Not used
    Bit19 : BOOL; //Not used
    Bit20 : BOOL; //Not used
    Bit21 : BOOL; //Not used
    Bit22 : BOOL; //Not used
  END_STRUCT;

```

```

    Bit23      : BOOL; //Not used
    Bit24      : BOOL; //Not used
    Bit25      : BOOL; //Not used
    Bit26      : BOOL; //Not used
    Bit27      : BOOL; //Not used
    Bit28      : BOOL; //Not used
    Bit29      : BOOL; //Not used
    Bit30      : BOOL; //Not used
    Bit31      : BOOL; //Not used
END_STRUCT;
ArrxOS_PermLog AT xdwOS_PermLog : ARRAY[0..3] OF BYTE;

// Variable for status
dwStatus      : DWORD;
ArrBitStatus AT dwStatus      : SBits32inDWORD;

//Local pointer
PLocal        : ANY; // Pointer to Local Var
PLoc_STR AT PLocal : PointSTR; // Pointer Structure
END_VAR

//=====
BEGIN
//OS Permissives
ArrOS_PermOut[3] := ArrOS_Perm[0];
ArrOS_PermOut[2] := ArrOS_Perm[1];
ArrOS_PermOut[1] := ArrOS_Perm[2];
ArrOS_PermOut[0] := ArrOS_Perm[3];
xdwOS_PermLog := dwOS_Perm;
strxOS_PermLog.Bit5 := OS_Perm.Bit5; //Lock Messages Permission

//OS_PermLog
ArrOS_PermLog[3] := ArrxOS_PermLog[0];
ArrOS_PermLog[2] := ArrxOS_PermLog[1];
ArrOS_PermLog[1] := ArrxOS_PermLog[2];
ArrOS_PermLog[0] := ArrxOS_PermLog[3];

//Status1 DWord
//| X0=PreAlarm | X1=Alarm | X2=Fault | X3=Test |
//| X4=Disabled | X5=OOS | X6=Zone_Test | X7=Zone_Dis |
//| X8=MsgLock | X9=MsgSup1 | X10=MsgSup2 | X11=MsgSup3 |
//| X12=xMsgCond1 | X13=xMsgCond2 | X14=xMsgCond3 | X15=LocMsgLck |
dwStatus := DWORD#0;
ArrBitStatus.X0 := BSTIn1.X0 OR BSTIn2.X0 OR BSTIn3.X0 OR BSTIn4.X0 OR BSTIn5.X0 OR BSTIn6.X0 OR
BSTIn7.X0 OR BSTIn8.X0 OR BSTIn9.X0 OR BSTIn10.X0 OR BSTIn11.X0 OR BSTIn12.X0 OR
BSTIn13.X0 OR BSTIn14.X0 OR BSTIn15.X0 OR BSTIn16.X0;
ArrBitStatus.X1 := BSTIn1.X1 OR BSTIn2.X1 OR BSTIn3.X1 OR BSTIn4.X1 OR BSTIn5.X1 OR BSTIn6.X1 OR
BSTIn7.X1 OR BSTIn8.X1 OR BSTIn9.X1 OR BSTIn10.X1 OR BSTIn11.X1 OR BSTIn12.X1 OR
BSTIn13.X1 OR BSTIn14.X1 OR BSTIn15.X1 OR BSTIn16.X1;
ArrBitStatus.X2 := BSTIn1.X2 OR BSTIn2.X2 OR BSTIn3.X2 OR BSTIn4.X2 OR BSTIn5.X2 OR BSTIn6.X2 OR
BSTIn7.X2 OR BSTIn8.X2 OR BSTIn9.X2 OR BSTIn10.X2 OR BSTIn11.X2 OR BSTIn12.X2 OR
BSTIn13.X2 OR BSTIn14.X2 OR BSTIn15.X2 OR BSTIn16.X2;
ArrBitStatus.X3 := BSTIn1.X3 OR BSTIn2.X3 OR BSTIn3.X3 OR BSTIn4.X3 OR BSTIn5.X3 OR BSTIn6.X3 OR
BSTIn7.X3 OR BSTIn8.X3 OR BSTIn9.X3 OR BSTIn10.X3 OR BSTIn11.X3 OR BSTIn12.X3 OR
BSTIn13.X3 OR BSTIn14.X3 OR BSTIn15.X3 OR BSTIn16.X3;
ArrBitStatus.X4 := BSTIn1.X4 OR BSTIn2.X4 OR BSTIn3.X4 OR BSTIn4.X4 OR BSTIn5.X4 OR BSTIn6.X4 OR
BSTIn7.X4 OR BSTIn8.X4 OR BSTIn9.X4 OR BSTIn10.X4 OR BSTIn11.X4 OR BSTIn12.X4 OR
BSTIn13.X4 OR BSTIn14.X4 OR BSTIn15.X4 OR BSTIn16.X4;
ArrBitStatus.X5 := BSTIn1.X5 OR BSTIn2.X5 OR BSTIn3.X5 OR BSTIn4.X5 OR BSTIn5.X5 OR BSTIn6.X5 OR
BSTIn7.X5 OR BSTIn8.X5 OR BSTIn9.X5 OR BSTIn10.X5 OR BSTIn11.X5 OR BSTIn12.X5 OR
BSTIn13.X5 OR BSTIn14.X5 OR BSTIn15.X5 OR BSTIn16.X5;
ArrBitStatus.X6 := BSTIn1.X6 OR BSTIn2.X6 OR BSTIn3.X6 OR BSTIn4.X6 OR BSTIn5.X6 OR BSTIn6.X6 OR
BSTIn7.X6 OR BSTIn8.X6 OR BSTIn9.X6 OR BSTIn10.X6 OR BSTIn11.X6 OR BSTIn12.X6 OR
BSTIn13.X6 OR BSTIn14.X6 OR BSTIn15.X6 OR BSTIn16.X6;
ArrBitStatus.X7 := BSTIn1.X7 OR BSTIn2.X7 OR BSTIn3.X7 OR BSTIn4.X7 OR BSTIn5.X7 OR BSTIn6.X7 OR
BSTIn7.X7 OR BSTIn8.X7 OR BSTIn9.X7 OR BSTIn10.X7 OR BSTIn11.X7 OR BSTIn12.X7 OR
BSTIn13.X7 OR BSTIn14.X7 OR BSTIn15.X7 OR BSTIn16.X7;
ArrBitStatus.X8 := BSTIn1.X8 OR BSTIn2.X8 OR BSTIn3.X8 OR BSTIn4.X8 OR BSTIn5.X8 OR BSTIn6.X8 OR
BSTIn7.X8 OR BSTIn8.X8 OR BSTIn9.X8 OR BSTIn10.X8 OR BSTIn11.X8 OR BSTIn12.X8 OR
BSTIn13.X8 OR BSTIn14.X8 OR BSTIn15.X8 OR BSTIn16.X8 OR MsgLock;
ArrBitStatus.X9 := BSTIn1.X9 OR BSTIn2.X9 OR BSTIn3.X9 OR BSTIn4.X9 OR BSTIn5.X9 OR BSTIn6.X9 OR
BSTIn7.X9 OR BSTIn8.X9 OR BSTIn9.X9 OR BSTIn10.X9 OR BSTIn11.X9 OR BSTIn12.X9 OR
BSTIn13.X9 OR BSTIn14.X9 OR BSTIn15.X9 OR BSTIn16.X9 OR MsgSup1;
ArrBitStatus.X10 := BSTIn1.X10 OR BSTIn2.X10 OR BSTIn3.X10 OR BSTIn4.X10 OR BSTIn5.X10 OR
BSTIn6.X10 OR BSTIn7.X10 OR BSTIn8.X10 OR BSTIn9.X10 OR BSTIn10.X10 OR
BSTIn11.X10 OR BSTIn12.X10 OR BSTIn13.X10 OR BSTIn14.X10 OR BSTIn15.X10 OR
BSTIn16.X10 OR MsgSup2;
ArrBitStatus.X11 := BSTIn1.X11 OR BSTIn2.X11 OR BSTIn3.X11 OR BSTIn4.X11 OR BSTIn5.X11 OR
BSTIn6.X11 OR BSTIn7.X11 OR BSTIn8.X11 OR BSTIn9.X11 OR BSTIn10.X11 OR
BSTIn11.X11 OR BSTIn12.X11 OR BSTIn13.X11 OR BSTIn14.X11 OR BSTIn15.X11 OR
BSTIn16.X11 OR MsgSup3;
ArrBitStatus.X12 := (BSTIn1.X12 AND NOT (Feature.Bit0) ) OR (BSTIn2.X12 AND NOT (Feature.Bit1) ) OR
(BSTIn3.X12 AND NOT (Feature.Bit2) ) OR (BSTIn4.X12 AND NOT (Feature.Bit3) ) OR
(BSTIn5.X12 AND NOT (Feature.Bit4) ) OR (BSTIn6.X12 AND NOT (Feature.Bit5) ) OR
(BSTIn7.X12 AND NOT (Feature.Bit6) ) OR (BSTIn8.X12 AND NOT (Feature.Bit7) ) OR
(BSTIn9.X12 AND NOT (Feature.Bit8) ) OR (BSTIn10.X12 AND NOT (Feature.Bit9) ) OR
(BSTIn11.X12 AND NOT (Feature.Bit10) ) OR (BSTIn12.X12 AND NOT (Feature.Bit11) ) OR

```

```

        (BSTIn13.X12 AND NOT (Feature.Bit12)) OR (BSTIn14.X12 AND NOT (Feature.Bit13)) OR
        (BSTIn15.X12 AND NOT (Feature.Bit14)) OR (BSTIn15.X12 AND NOT (Feature.Bit15));
ArrBitStatus.X13 := (BSTIn1.X13 AND NOT (Feature.Bit0) ) OR (BSTIn2.X13 AND NOT (Feature.Bit1) ) OR
        (BSTIn3.X13 AND NOT (Feature.Bit2) ) OR (BSTIn4.X13 AND NOT (Feature.Bit3) ) OR
        (BSTIn5.X13 AND NOT (Feature.Bit4) ) OR (BSTIn6.X13 AND NOT (Feature.Bit5) ) OR
        (BSTIn7.X13 AND NOT (Feature.Bit6) ) OR (BSTIn8.X13 AND NOT (Feature.Bit7) ) OR
        (BSTIn9.X13 AND NOT (Feature.Bit8) ) OR (BSTIn10.X13 AND NOT (Feature.Bit9) ) OR
        (BSTIn11.X13 AND NOT (Feature.Bit10)) OR (BSTIn12.X13 AND NOT (Feature.Bit11)) OR
        (BSTIn13.X13 AND NOT (Feature.Bit12)) OR (BSTIn14.X13 AND NOT (Feature.Bit13)) OR
        (BSTIn15.X13 AND NOT (Feature.Bit14)) OR (BSTIn15.X13 AND NOT (Feature.Bit15));
ArrBitStatus.X14 := (BSTIn1.X14 AND NOT (Feature.Bit0) ) OR (BSTIn2.X14 AND NOT (Feature.Bit1) ) OR
        (BSTIn3.X14 AND NOT (Feature.Bit2) ) OR (BSTIn4.X14 AND NOT (Feature.Bit3) ) OR
        (BSTIn5.X14 AND NOT (Feature.Bit4) ) OR (BSTIn6.X14 AND NOT (Feature.Bit5) ) OR
        (BSTIn7.X14 AND NOT (Feature.Bit6) ) OR (BSTIn8.X14 AND NOT (Feature.Bit7) ) OR
        (BSTIn9.X14 AND NOT (Feature.Bit8) ) OR (BSTIn10.X14 AND NOT (Feature.Bit9) ) OR
        (BSTIn11.X14 AND NOT (Feature.Bit10)) OR (BSTIn12.X14 AND NOT (Feature.Bit11)) OR
        (BSTIn13.X14 AND NOT (Feature.Bit12)) OR (BSTIn14.X14 AND NOT (Feature.Bit13)) OR
        (BSTIn15.X14 AND NOT (Feature.Bit14)) OR (BSTIn15.X14 AND NOT (Feature.Bit15));
ArrBitStatus.X15 := MsgLock;
Status1          := dwStatus;

//Alarms Managment
PLocal := CMP_ID;
CMP_ID := WORD_TO_DWORD(PLoc_STR.DB_Num); // DBNum for Messages
xMsgCond1 := ArrBitStatus.X12 AND NOT (MsgSup1 OR MsgLock);
xMsgCond2 := ArrBitStatus.X13 AND NOT (MsgSup2 OR MsgLock);
xMsgCond3 := ArrBitStatus.X14 AND NOT (MsgSup3 OR MsgLock);

//Activate MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
    MsgSTS1:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond1
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_1
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig1:=xMsgCond1;
xRetVal:=ALARM_SC(
    EV_ID := MSG_EVID_1
    ,STATE := xRetSts
    ,Q_STATE := MsgACK1);

//PreAlarm MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
    MsgSTS2:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond2
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_2
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xRetVal:=ALARM_SC(
    EV_ID := MSG_EVID_2
    ,STATE := xRetSts
    ,Q_STATE := MsgACK2);

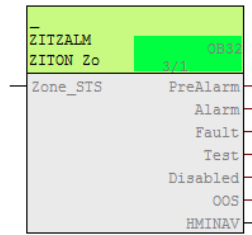
//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
    MsgSTS3:=INT_TO_WORD(ALARM_DQ(
        SIG := xMsgCond3
        ,ID := W#16#EEEE
        ,EV_ID := MSG_EVID_3
        ,CMP_ID := CMP_ID
        ,SD := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xRetVal:=ALARM_SC(
    EV_ID := MSG_EVID_3
    ,STATE := xRetSts
    ,Q_STATE := MsgACK3);

END_FUNCTION_BLOCK

```

2.19 FB557: ZITALM

- Representación Gráfica



- Parámetros

Name	Data Type	Address	Initial Value	Comment
IN				
Zone_STS	Any	0.0		Connect to ZITONP2 ZoneX_STS
MSG_EVID_1	DWord	10.0	DW#16#0	Message 1 ID (Alarm)
MSG_EVID_2	DWord	14.0	DW#16#0	Message 2 ID (PreAlarm)
MSG_EVID_3	DWord	18.0	DW#16#0	Message 2 ID (Fault)
MsgSup1	Bool	22.0	FALSE	Suppress Message 1
MsgSup2	Bool	22.1	FALSE	Suppress Message 2
MsgSup3	Bool	22.2	FALSE	Suppress Message 3
MsgLock	Bool	22.3	FALSE	Message Lock
OUT				
PreAlarm	Bool	24.0	FALSE	Module PreAlarm
Alarm	Bool	24.1	FALSE	Module Activate
Fault	Bool	24.2	FALSE	Module Fault
Test	Bool	24.3	FALSE	Module in Test
Disabled	Bool	24.4	FALSE	Module Disabled
Zone_Num	Int	26.0	0	Zone Number
OOS	Bool	28.0	FALSE	Out of Service
HMINAV	Struct	30.0		HMI Panel Interface
Status	Word	30.0	W#16#0	Status Word
DBNum	Int	32.0	0	DB Number
Btype	Int	34.0	0	Block Type
MsgSTS1	Word	36.0	W#16#0	Message 1 Status
MsgACK1	Bool	38.0	FALSE	Message 1 Acknowledge
MsgSTS2	Word	40.0	W#16#0	Message 2 Status
MsgACK2	Bool	42.0	FALSE	Message 2 Acknowledge
MsgSTS3	Word	44.0	W#16#0	Message 3 Status
MsgACK3	Bool	46.0	FALSE	Message 3 Acknowledge

```

{
Scl_ResetOptions ;
Scl_OverwriteBlocks:=      'y' ;
Scl_GenerateReferenceData := 'y' ;
Scl_S7ServerActive:=      'y' ;
Scl_CreateObjectCode:=    'y' ;
Scl_OptimizeObjectCode:=  'y' ;
Scl_MonitorArrayLimits:=  'y' ;
Scl_CreateDebugInfo :=    'y' ;
Scl_SetOKFlag:=           'n' ;
Scl_SetMaximumStringLength:= '254'
}
// Pointer Struct UDT104
TYPE PointSTR
STRUCT
  ID      : BYTE;      // Always B#16#10
  TypeCode : BYTE;    // Pointer Type
  Quantity : WORD;    // Quantity of Data
  DB_Num   : WORD;    // Data Block Number
  PArea   : DWORD;   // Data Area Pointer
END_STRUCT;
END_TYPE

//Zone Status UDT105
TYPE ZTZoneSTS
STRUCT
  PreAlarm : BOOL;    //Zone in PreAlarm
  Alarm    : BOOL;    //Zone in Alarm
  Fault    : BOOL;    //Zone in Fault
  Test     : BOOL;    //Zone in Test
  Disabled : BOOL;    //Zone Disabled
END_STRUCT
END_TYPE

// HMINAV Link Struct UDT106
TYPE HMINAVSTR
STRUCT
  Status : WORD;    // Status Word
  DBNum  : INT;    // DB Number
  Btype  : INT;    // Block Type
END_STRUCT;
END_TYPE

//=====
FUNCTION_BLOCK FB557
TITLE = 'ZITON Zone Alarm'
VERSION: '09.00'
AUTHOR: 'AdvLibFP'
NAME: 'ZITZALM'
FAMILY: 'ZITON'
KNOW_HOW_PROTECT
{PO_Count := '1';
 S7_m_c := 'true';
 S7_alarm_ui := '1'}

//=====
VAR_INPUT
Zone_STS : ANY;      //Connect to ZITONP2 ZoneX_STS
//Messages Configuration
MSG_EVID_1 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s'} : DWORD:=0; //Message 1 ID (Alarm)
MSG_EVID_2 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s'} : DWORD:=0; //Message 2 ID (PreAlarm)
MSG_EVID_3 {
  S7_visible := 'false';
  S7_link := 'false';
  S7_param := 'false';
  S7_server := 'alarm_archiv';
  S7_a_type := 'alarm_s'} : DWORD:=0; //Message 2 ID (Fault)
MsgSup1 {
  S7_visible := 'false';
  S7_string_1 := 'activ';
  S7_string_0 := 'inactiv'} : BOOL; //Suppress Message 1
MsgSup2 {
  S7_visible := 'false';
  S7_string_0 := 'inactiv';
  S7_string_1 := 'activ'} : BOOL; //Suppress Message 2
MsgSup3 {
  S7_visible := 'false';

```

```

    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ' } : BOOL; //Suppress Message 3
MsgLock {
    S7_visible := 'false';
    S7_string_0 := 'inactiv';
    S7_string_1 := 'activ' } : BOOL; //Message Lock
END_VAR

//=====
VAR_OUTPUT
PreAlarm {
    S7_dynamic := 'true';
    S7_string_1 := 'PreAlarm' } : BOOL; //Zone PreAlarm
Alarm {
    S7_dynamic := 'true';
    S7_string_1 := 'Alarm' } : BOOL; //Zone Activate
Fault {
    S7_dynamic := 'true';
    S7_string_1 := 'Fault' } : BOOL; //Zone Fault
Test {
    S7_dynamic := 'true';
    S7_string_1 := 'Test' } : BOOL; //Zone in Test
Disabled {
    S7_dynamic := 'true';
    S7_string_1 := 'Disabled' } : BOOL; //Zone Disabled
Zone_Num {
    S7_m_c := 'true';

    S7_visible := 'false';
    S7_edit := 'para' } : INT; //Zone Number
OOS {
    S7_dynamic := 'true';
    S7_string_1 := 'OOS' } : BOOL; //Out of Service
HMINAV : HMINAVSTR; //HMI Panel Interface
//Messages Configuration
MsgSTS1 {
    S7_visible := 'false' } : WORD; //Message 1 Status
MsgACK1 {
    S7_visible := 'false' } : BOOL; //Message 1 Acknowledge
MsgSTS2 {
    S7_visible := 'false' } : WORD; //Message 2 Status
MsgACK2 {
    S7_visible := 'false' } : BOOL; //Message 2 Acknowledge
MsgSTS3 {
    S7_visible := 'false' } : WORD; //Message 3 Status
MsgACK3 {
    S7_visible := 'false' } : BOOL; //Message 3 Acknowledge
END_VAR

//=====
CONST
ZBAddrCPU := 72; // Zones Byte Base Address
AckZBAddrCPU := 1610; // ZonesSTS Acknowledgements Base Address
ZBAddrCP := 62; // Zones Byte Base Address
AckZBAddrCP := 1568; // ZonesSTS Acknowledgements Base Address
END_CONST

//=====
VAR
FBID : WORD; // Panel type
ZBAddr : INT; // Zones Byte Base Address
AckZBAddr : INT; // ZonesSTS Acknowledgements Base Address
xZonSTS : ZTZoneSTS; // Zone Status
xAckZone : // Zone Status Acknowledgement
STRUCT
    X8 : BOOL;
    X9 : BOOL;
    X10 : BOOL;
    X11 : BOOL;
    X12 : BOOL;
    X13 : BOOL;
    X14 : BOOL;
    OOS : BOOL; //Out of service
    PreAlarm : BOOL; //Zone ACK for PreAlarm
    Alarm : BOOL; //Zone ACK for Alarm
    Fault : BOOL; //Zone ACK for Fault
    X3 : BOOL;
    X4 : BOOL;
    X5 : BOOL;
    X6 : BOOL;
    X7 : BOOL;
END_STRUCT;
xHMISTS : WORD; // HMI Status
xBithMISTS AT xHMISTS : SBits16inWORD; // HMI Status as Bits

CMP_ID : DWORD; // Area code
xMsgCond1 : BOOL; // Condition MSG1

```

```

xMsgCond2      : BOOL;          // Condition MSG2
xMsgCond3      : BOOL;          // Condition MSG3
xMsgTrig1      : BOOL;          // Trigger MSG1
xMsgTrig2      : BOOL;          // Trigger MSG2
xMsgTrig3      : BOOL;          // Trigger MSG3
xMsgArch1      : INT;           // MSG1 Status
xMsgArch2      : INT;           // MSG2 Status
xMsgArch3      : INT;           // MSG3 Status
AuxValue:      // Msg Aux Value
  STRUCT
    Zone        : INT;           // Msg Zone
    DBNum       : WORD;          // Msg DB num
  END_STRUCT;
xRetSts        : BOOL;          // Msg Status
END_VAR

//=====
VAR_TEMP
xAux_Err       : INT;           // Auxiliar Return Function Error
xAddr          : DINT;
xFMsgLock      : BOOL;          // MsgLock by Features Bits
xMsg1Zone      : BOOL;          // MsgLock1 by Zone Alarm Status
xMsg2Zone      : BOOL;          // MsgLock2 by Zone PreAlarm Status

// Structures to copy data areas
PPanel         : ANY;           // Pointer to Panel Block
PPan_STR AT PPanel : PointSTR;  // Pointer Structure
PLocal        : ANY;           // Pointer to Local Var
PLoc_STR AT PLocal : PointSTR;  // Pointer Structure
END_VAR

// CODE =====
BEGIN
//Panel Type to knows base addresses
PPanel         := Zone_STS;      // Pointer to Panel Block
PPan_STR.TypeCode := BYTE#04;    // Work with WORDS
PPan_STR.Quantity := WORD#01;    // 1 Word of Data
PPan_STR.PArea  := DW#16#8400000; // First Word of the DB
PLocal         := FBID;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ZoneX_STS
IF FBID=WORD#01 THEN // ZITONCP
  ZBAddr      := ZBAddrCP;
  AckZBAddr   := AckZBAddrCP;
ELSE
  ZBAddr      := ZBAddrCPU;
  AckZBAddr   := AckZBAddrCPU;
END_IF;
//Zone outputs
PPanel         := Zone_STS;      // Pointer to Panel Block
PPan_STR.TypeCode := BYTE#02;    // Work with BYTES
PPan_STR.Quantity := WORD#01;    // 1 Byte of Data
PLocal        := xZonSTS;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve ZoneX_STS
Zone_Num :=1+((DWORD_TO_INT(PPan_STR.PArea)-ZBAddr*8)/16); // Calculate Zone Number

//Zone status Acknowledgement
PPan_STR.TypeCode := BYTE#02;    // Work with WORDS
PPan_STR.Quantity := WORD#02;    // 1 Word of DATA
xAddr := (AckZBAddr+2*(Zone_Num-1))*8; // Addr in bits
PPan_STR.PArea  := DINT_TO_DWORD(xAddr) OR DW#16#84000000;
PLocal         := xAckZone;
xAux_Err :=BLKMOV(SRCBLK := PPanel, DSTBLK := PLocal); // Retrieve AckZonesSTS[x]

//update Outputs
PreAlarm := xZonSTS.PreAlarm;
Alarm    := xZonSTS.Alarm;
Fault    := xZonSTS.Fault;
Test     := xZonSTS.Test;
Disabled := xZonSTS.Disabled;
OOS      := xAckZone.OOS;

//Alarms Managment
CMP_ID   := WORD_TO_DWORD(PLoc_STR.DB_Num); // DBNum for Messages
AuxValue.Zone := Zone_Num;
AuxValue.DBNum := PLoc_STR.DB_Num;
xMsgCond1 := Alarm AND NOT(xAckZone.Alarm OR MsgSup1 OR MsgLock);
xMsgCond2 := PreAlarm AND NOT(xAckZone.PreAlarm OR MsgSup2 OR MsgLock);
xMsgCond3 := Fault AND NOT(xAckZone.Fault OR MsgSup3 OR MsgLock);
//Activate MSG
IF (xMsgCond1 <> xMsgTrig1) THEN
  MsgSTS1:=INT_TO_WORD(ALARM_DQ(
    SIG      := xMsgCond1
    ,ID      := W#16#EEEE
    ,EV_ID   := MSG_EVID_1
    ,CMP_ID  := CMP_ID
    ,SD      := AuxValue));
END_IF;

```

```

xMsgTrig1:=xMsgCond1;
xMsgArch1:=ALARM_SC(
    EV_ID    := MSG_EVID_1
    ,STATE   := xRetSts
    ,Q_STATE := MsgACK1);

//PreAlarm MSG
IF (xMsgCond2 <> xMsgTrig2) THEN
    MsgSTS2:=INT_TO_WORD(ALARM_DQ(
        SIG     := xMsgCond2
        ,ID     := W#16#EEEE
        ,EV_ID  := MSG_EVID_2
        ,CMP_ID := CMP_ID
        ,SD     := AuxValue));
END_IF;
xMsgTrig2:=xMsgCond2;
xMsgArch2:=ALARM_SC(
    EV_ID    := MSG_EVID_2
    ,STATE   := xRetSts
    ,Q_STATE := MsgACK2);

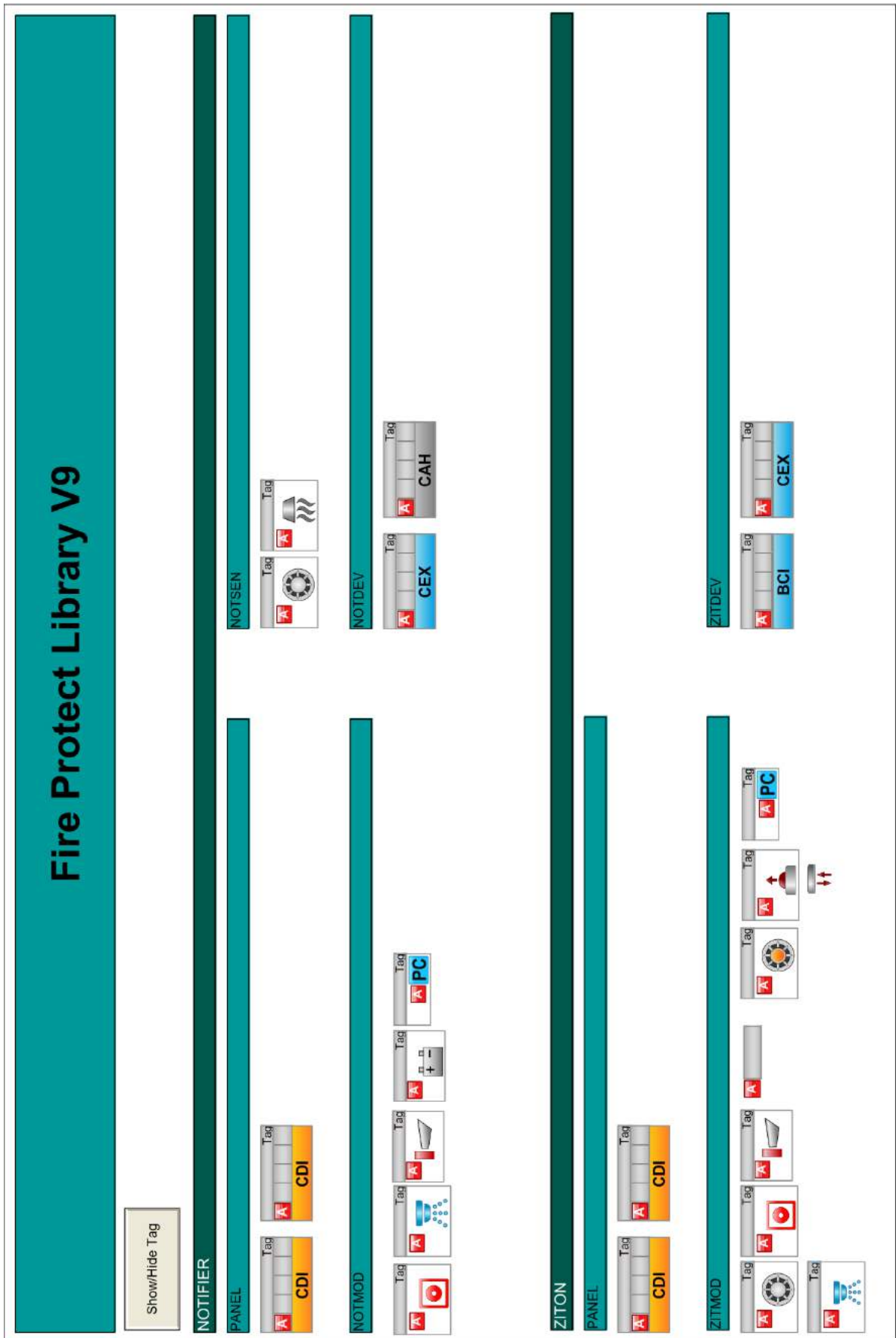
//Fault MSG
IF (xMsgCond3 <> xMsgTrig3) THEN
    MsgSTS3:=INT_TO_WORD(ALARM_DQ(
        SIG     := xMsgCond3
        ,ID     := W#16#EEEE
        ,EV_ID  := MSG_EVID_3
        ,CMP_ID := CMP_ID
        ,SD     := AuxValue));
END_IF;
xMsgTrig3:=xMsgCond3;
xMsgArch3:=ALARM_SC(
    EV_ID    := MSG_EVID_3
    ,STATE   := xRetSts
    ,Q_STATE := MsgACK3);

//HMINAV
//Status
xBithMISTS.X0 := Alarm;
xBithMISTS.X1 := PreAlarm;
xBithMISTS.X2 := Fault;
xBithMISTS.X3 := false;
//Msg activated
xBithMISTS.X4 := xMsgCond1;
xBithMISTS.X5 := xMsgCond2;
xBithMISTS.X6 := xMsgCond3;
xBithMISTS.X7 := false;
//Msg Acknowled
xBithMISTS.X8 := MsgACK1;
xBithMISTS.X9 := MsgACK2;
xBithMISTS.X10 := MsgACK3;
xBithMISTS.X11 := false;
//Msg in Archive (true if msg in Archive)
xBithMISTS.X12 := (xMsgArch1=0);
xBithMISTS.X13 := (xMsgArch2=0);
xBithMISTS.X14 := (xMsgArch3=0);
xBithMISTS.X15 := false;
HMINAV.Status := xHMISTS;
HMINAV.DBNum := WORD_TO_INT(AuxValue.DBNum);
HMINAV.Btype := 13; //ZITZALM code=13

END_FUNCTION_BLOCK

```

2.20 PCS7: Librería Block Icons



Monitorización de un Sistema de Detección de Incendios

PLANOS



UNIVERSITAT ROVIRA I VIRGILI

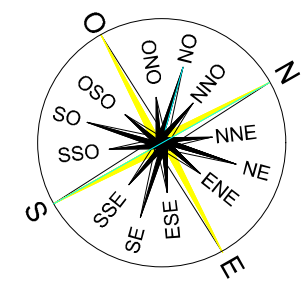
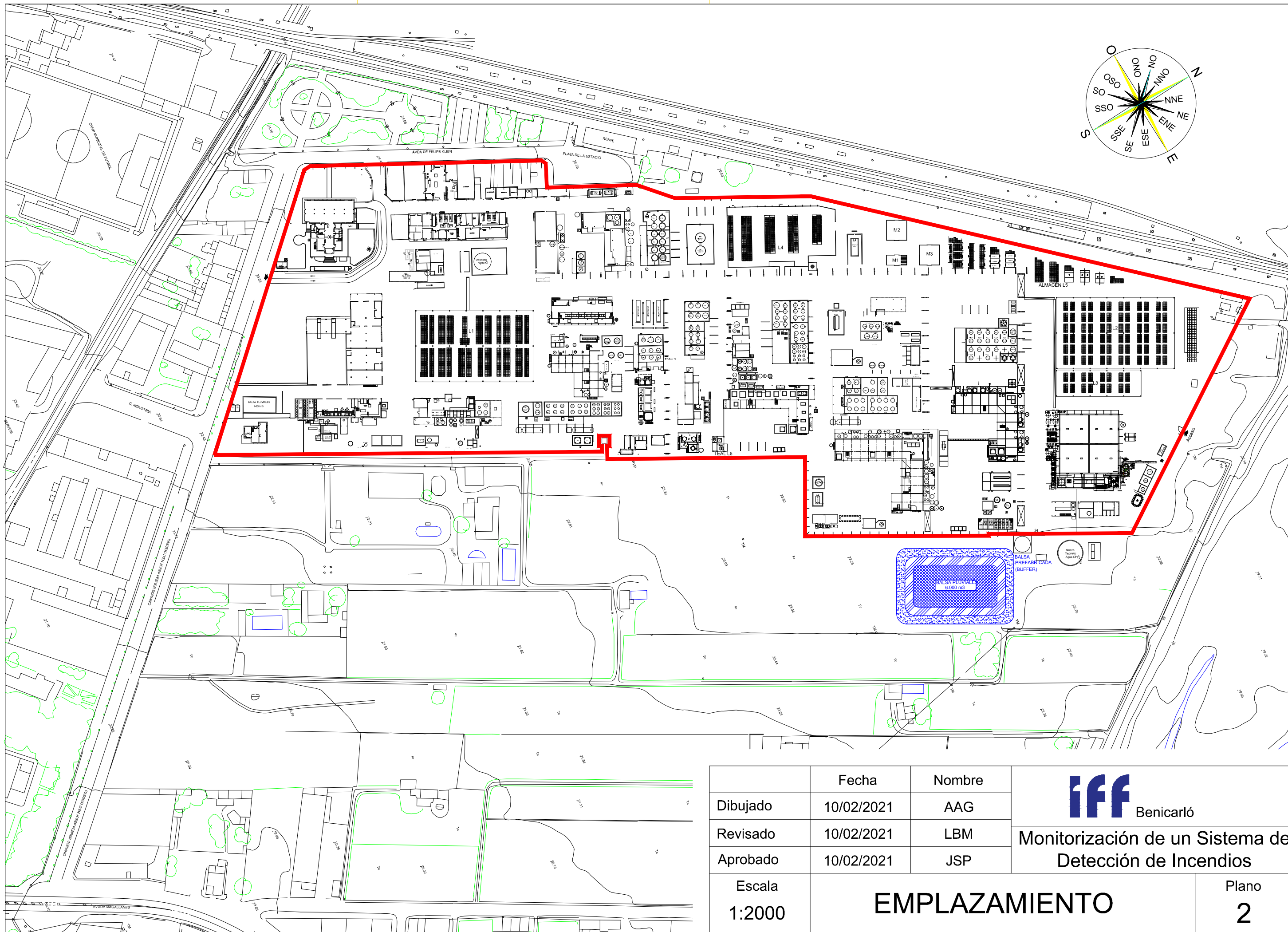
Tarragona
2021


3 Planos

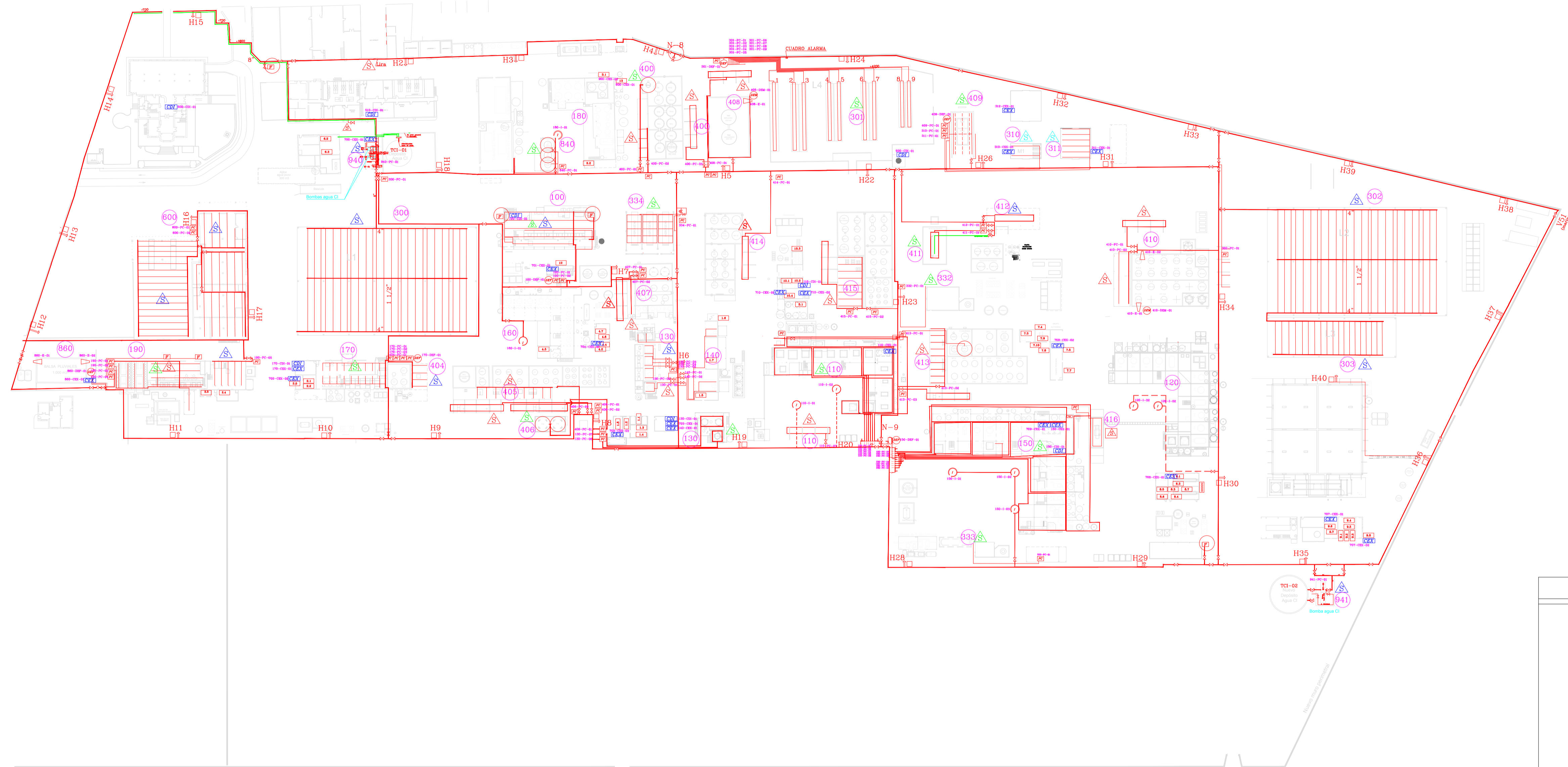
3.1	Situación.....	281
3.2	Emplazamiento.....	282
3.3	Red General Contra Incendios	283
3.4	Detectores Edif. Oficinas Planta Baja	284
3.5	Detectores Edif. Oficinas Primera Planta.....	285
3.6	Detectores Edif. Laboratorios	286
3.7	Detectores Edif. Capability Lab	287
3.8	Detectores Edif. Vestuarios/Comedor	288
3.9	Detectores Varios	289
3.10	Detectores Servidores Edif. Laboratorios.....	290
3.11	Detectores Servidores Plantas.....	291
3.12	Detectores Sala Eléctrica Planta Finos	292
3.13	Distribución CDI.....	293
3.14	Trazado Fibra Óptica	294
3.15	Arquitectura del Sistema de Control	295



	Fecha	Nombre	 IFF Benicarló
Dibujado	10/02/2021	AAG	
Revisado	10/02/2021	LBM	
Aprobado	10/02/2021	JSP	Monitorización de un Sistema de Detección de Incendios
Escala N/A	SITUACION		Plano 1



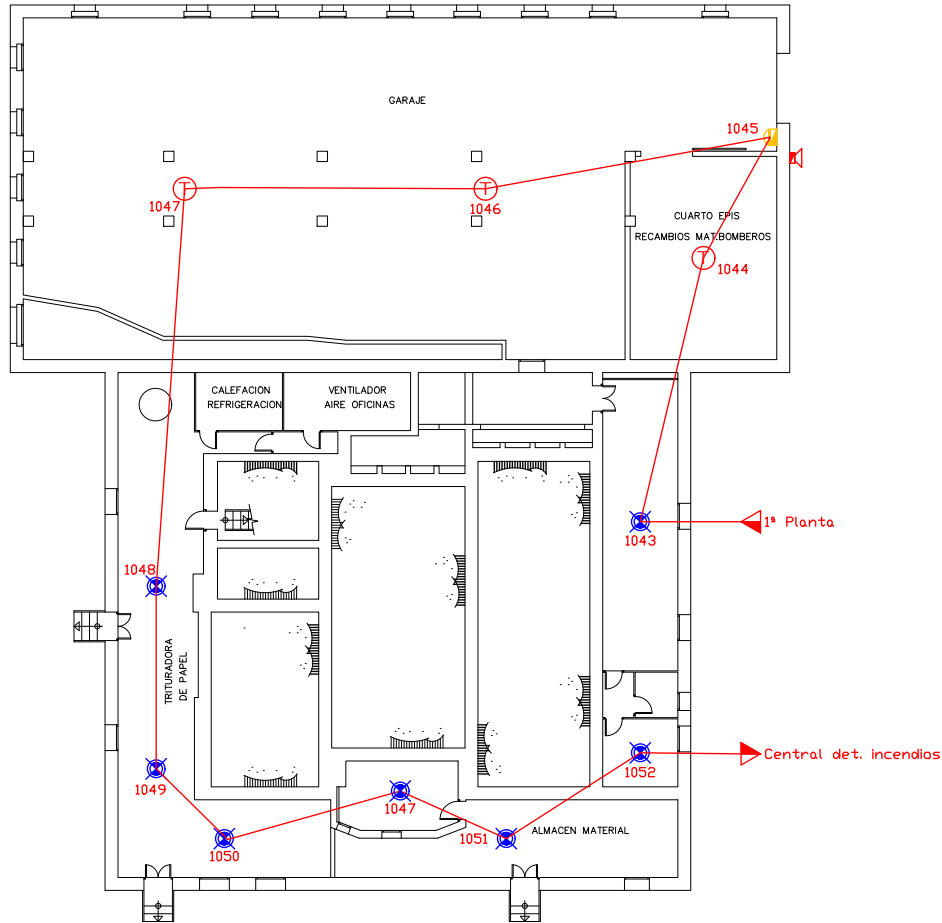
	Fecha	Nombre	 Monitorización de un Sistema de Detección de Incendios
Dibujado	10/02/2021	AAG	
Revisado	10/02/2021	LBM	
Aprobado	10/02/2021	JSP	
Escala 1:2000	EMPLAZAMIENTO		Plano 2



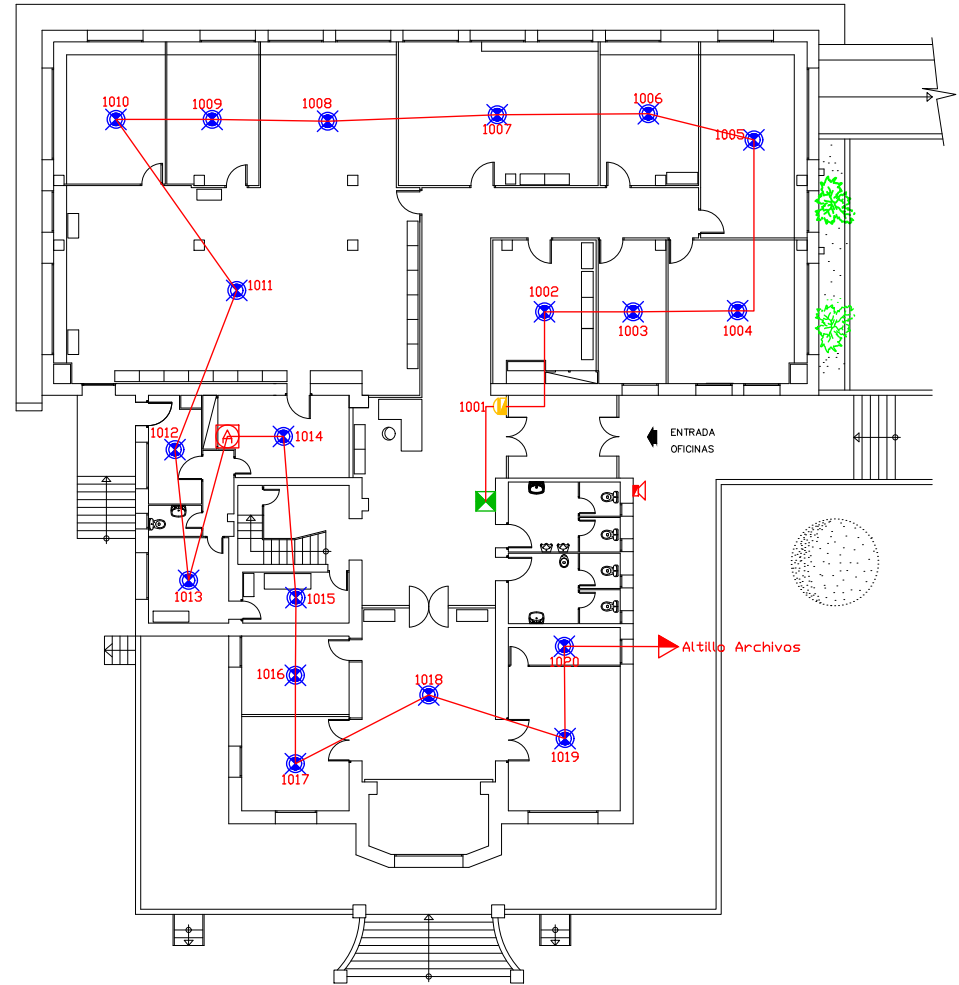
SIMBOLOGÍA	
	CENTRAL DETECCIÓN DE INCENDIOS
	CENTRAL EXTINCIÓN DE INCENDIOS
	NUMERACION SPRINKLERS Y LANZAS
	BOCA DE INCENDIOS EQUIPADA
	LANZA de AGUA - ESPUMA
	PUESTO DE CONTROL
	DEPOSITO DE ESPUMOCENO FIJO
	DEPOSITO DE ESPUMOCENO MOVIL
	RED INCENDIO
	SPRINKLERS EN FUTURA EJECUCION
	CARON de AGUA/ESPUMA de 1800 l/m
	HIDRANTE con 2 TOMAS de 45 mm, 1 de 80 mm y ANCHO de MATERIAL
	NUDO
	SPRAY NOZZLES ACTIV MANUAL de Carbonato Sódico
	SPRAY NOZZLES ACTIVACION MANUAL (Tubería seca)
	SPRINKLERS AUTOMATICOS (Rociadores)
	SPRAY NOZZLES DOBLE VIA (Diluvio con línea pilotada de activación)
	SPRAY NOZZLES ACTIVACION POR SOLENOIDE (Diluvio)
	Botellón CO2 de Reserva
	Instalaciones Detección-Alarma-Extinción CO2

IFF BENICARLÓ, S.L.
Avenida Felipe Klein, nº2 Benicarló (Castellón)

PLANO PLANO DE ABASTECIMIENTO AGUA CI			PLANTA GENERAL	
Dibujado	Fecha	Nombre	FECHA	PLANO
Comprobado	Ago-20	LSM	04-08-20	3
Aprobado			ESCALA	REVISIÓN
			1/1.000	0



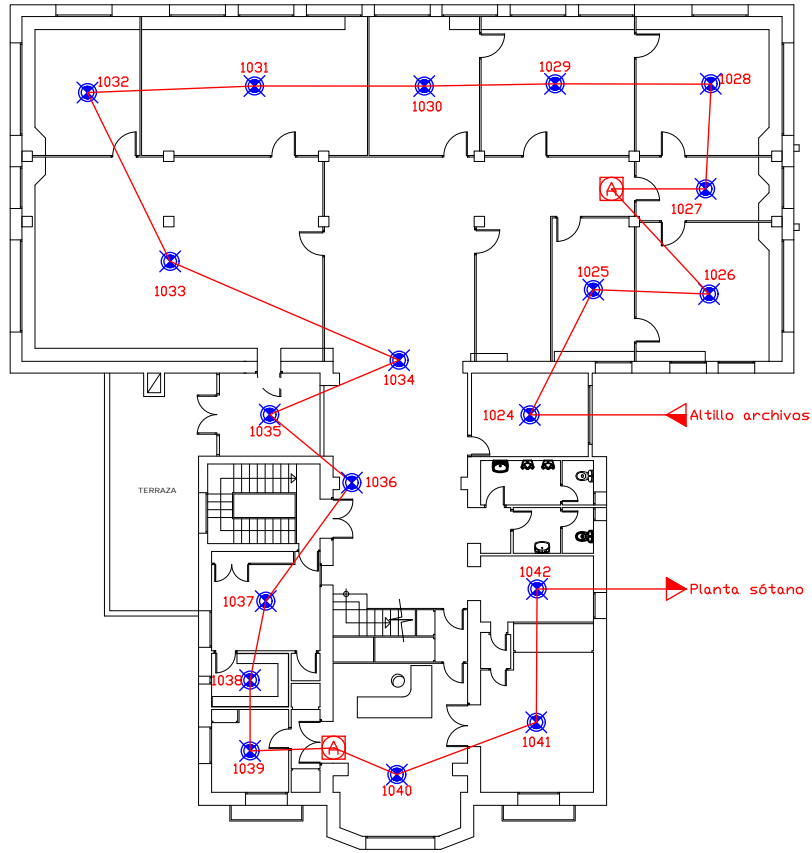
PLANTA SEMISOTANO



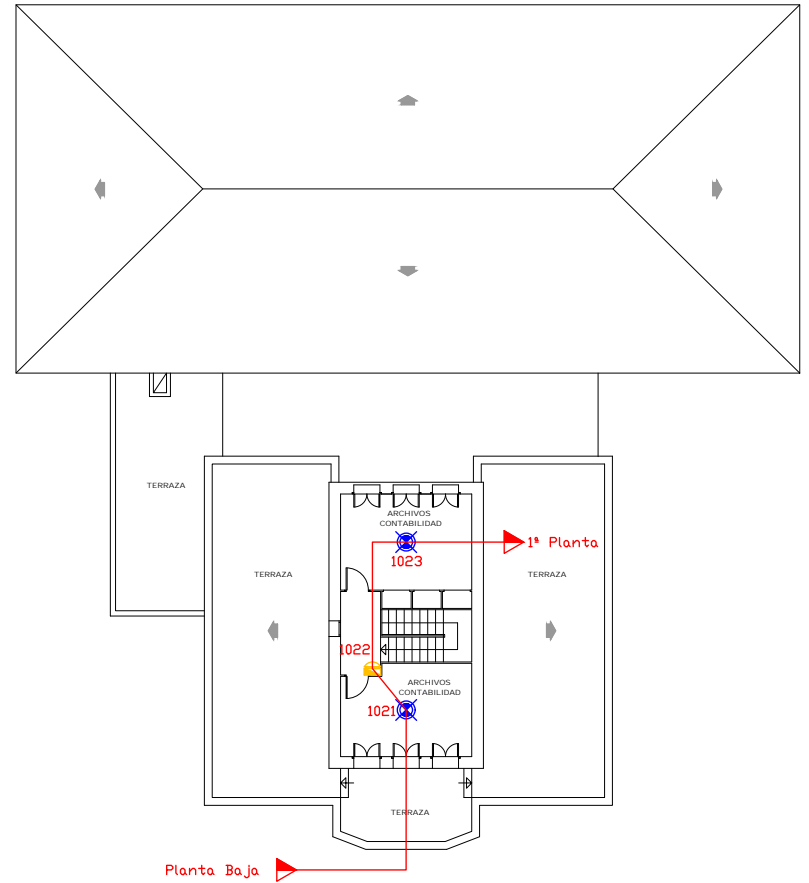
PLANTA BAJA

- Aislador
- Sensor térmico analógico
- Pulsador alarma analógico
- Sensor óptico analógico direccionable
- Sirena incendio roja exterior
- Central analógica direccionable de 1 lazo

FECHA	NOMBRE		BENICARLO, S.A. (CASTELLON) SPAN
DIBUJADO	04/01/14		
COMPROBADO	04/01/14	JAVIER	
Escala	DISTRIBUCION OFICINA PLANTA SEMISOTANO/BAJA		DETECTORES DE HUMO
1/200	DETECTORES DE HUMO		REVISION
			Revisado por:



PLANTA PRIMERA

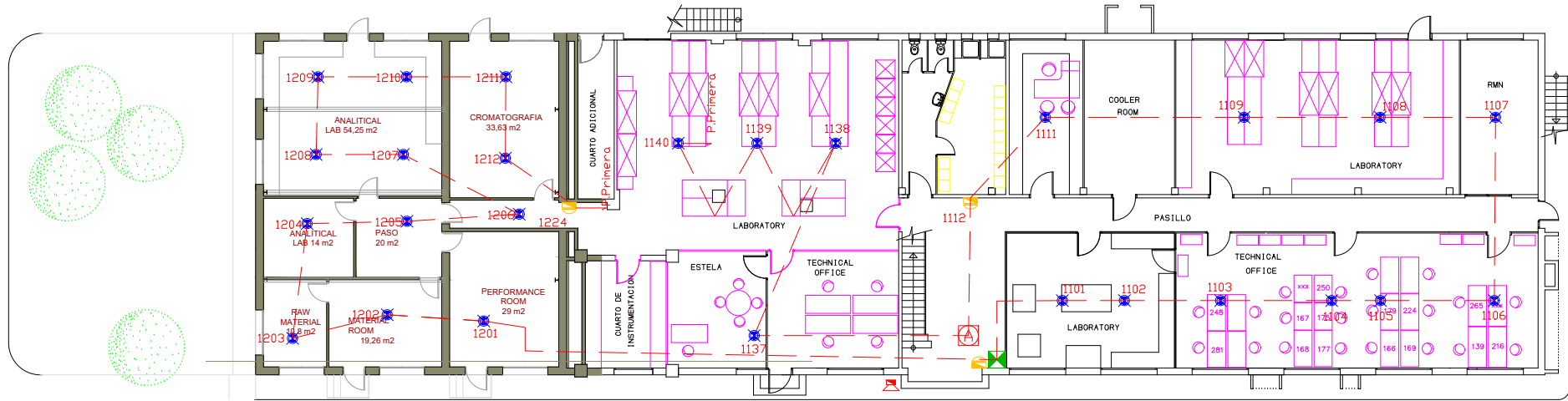
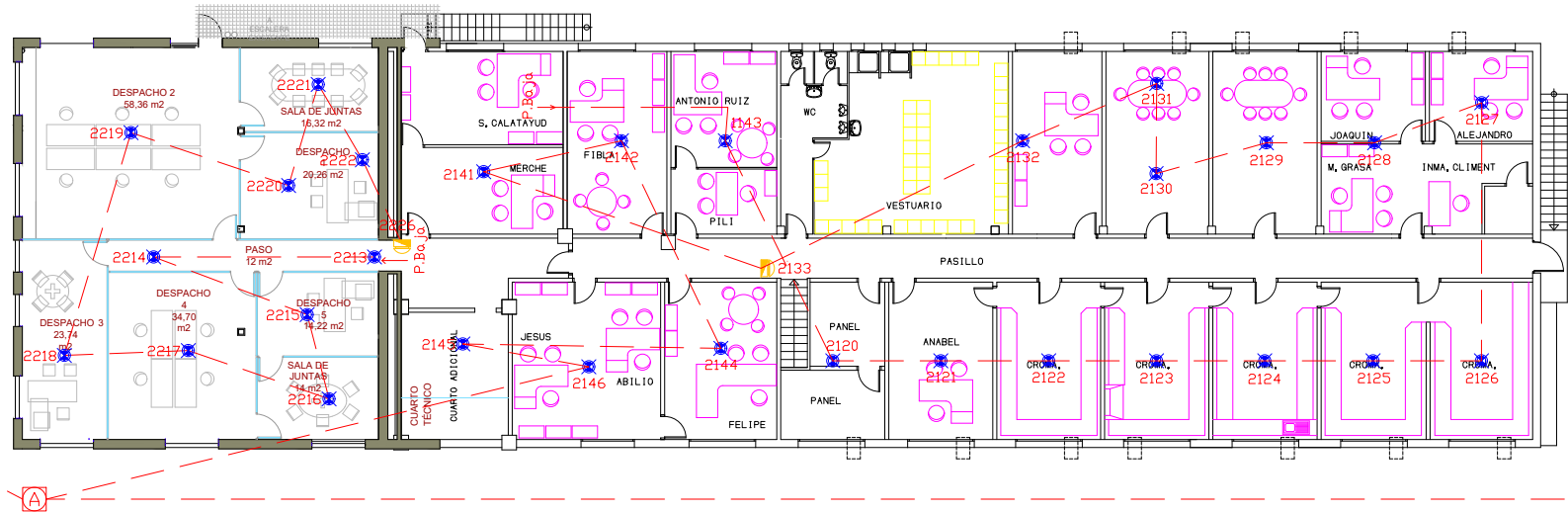


PLANTA ATICO

- A Aislador
- T Sensor térmico analógico
- P Pulsador alarma analógico
- D Sensor óptico analógico direccional
- A Sirena incendio roja exterior
- X Central analógica direccional de 1 lazo

FECHA	NOMBRE	BENICARLO, S.A.
DIBUJADO 04/6/14	AVANZUELA	(CASTELLÓN) SPAIN
COMPROB. 04/6/14	JORDANS	
Escala	DISTRIBUCION DE OFICINAS	DETECTORES DE HUMO
1/200	PLANTA PRIMERA, ATICO	REVISION
	DETECTORES DE HUMO	Revisión por:

LABS-OFFICES - FIRST FLOOR



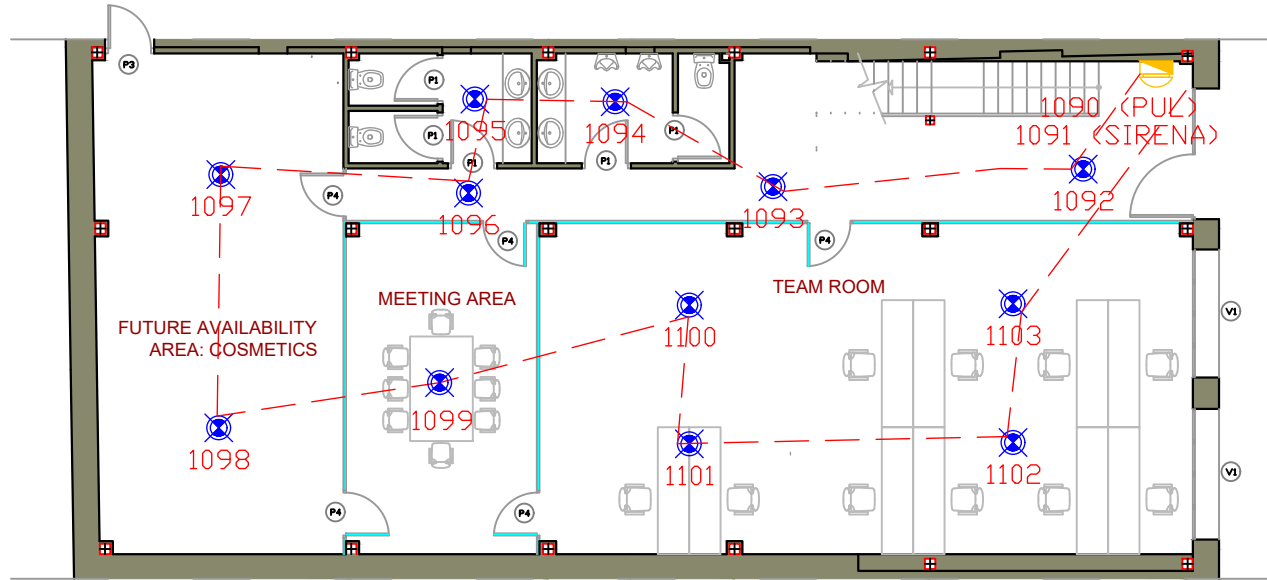
LABS-OFFICES - GROUND FLOOR

XYZZX: Número de la zona; Y: Número de lazo; ZZ: Número del detector.

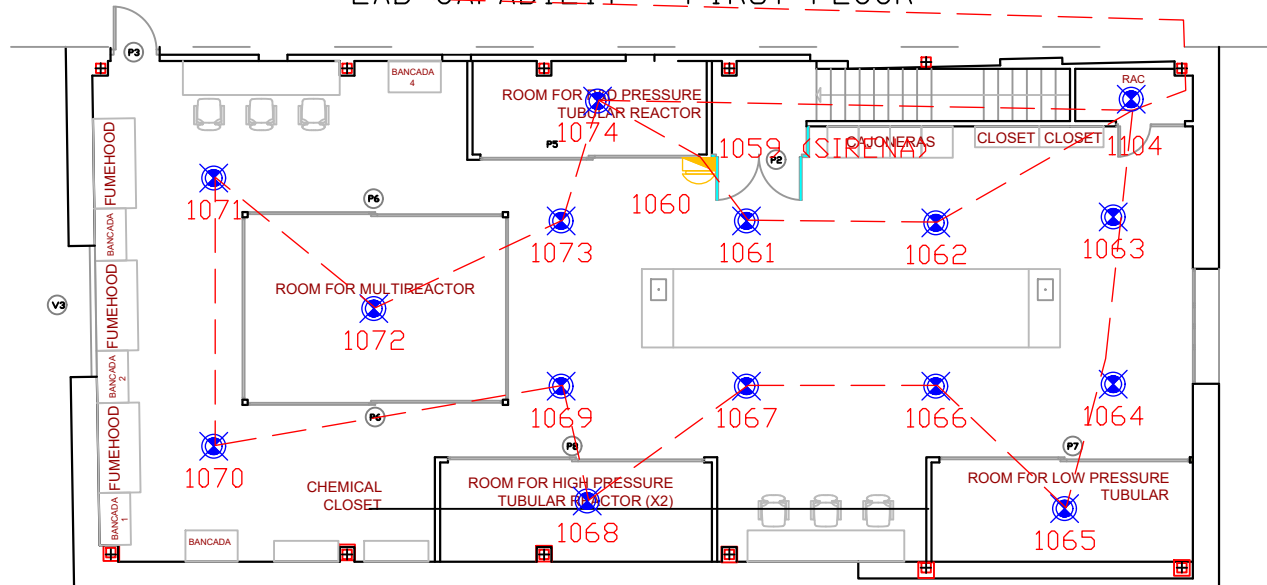
- Aislador
- Sensor térmico analógico
- Pulsador alarma analógico
- Sensor óptico analógico direccional
- Sirena incendio roja exterior
- Central analógica direccional de 1 lazo

FECHA	NOMBRE	
DESEÑADO	24/06/2011	
COMPROB.	24/06/2011	JROJAS
		BENICARLO S.A. (CASTELLÓN) SPAIN
EXPOS	OFICINA DE FABRICA	LABORAT-2-Rev-1
1/100	LABORATORIO - VESTUARIOS	REVISION

LABS CAPABILITY - GROUND FLOOR



LAB CAPABILITY - FIRST FLOOR



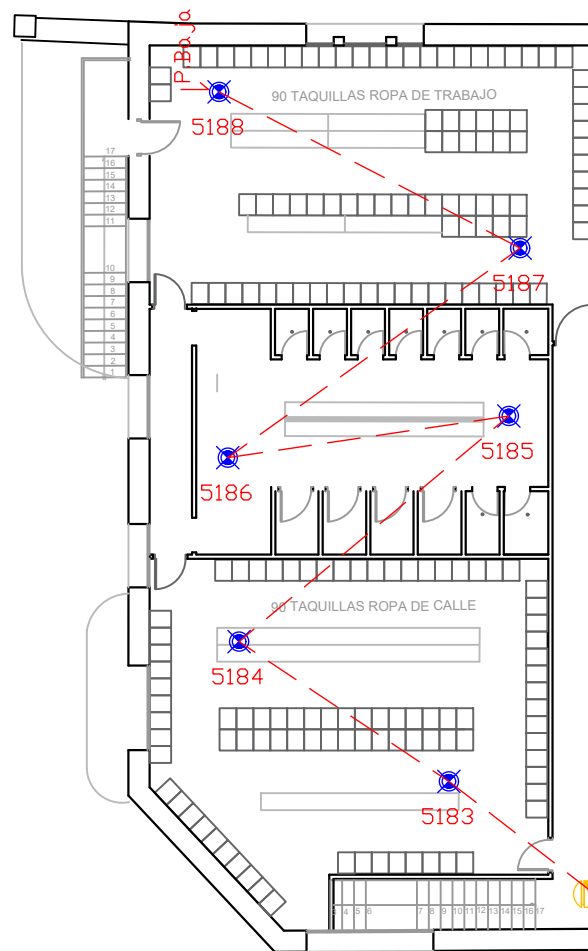
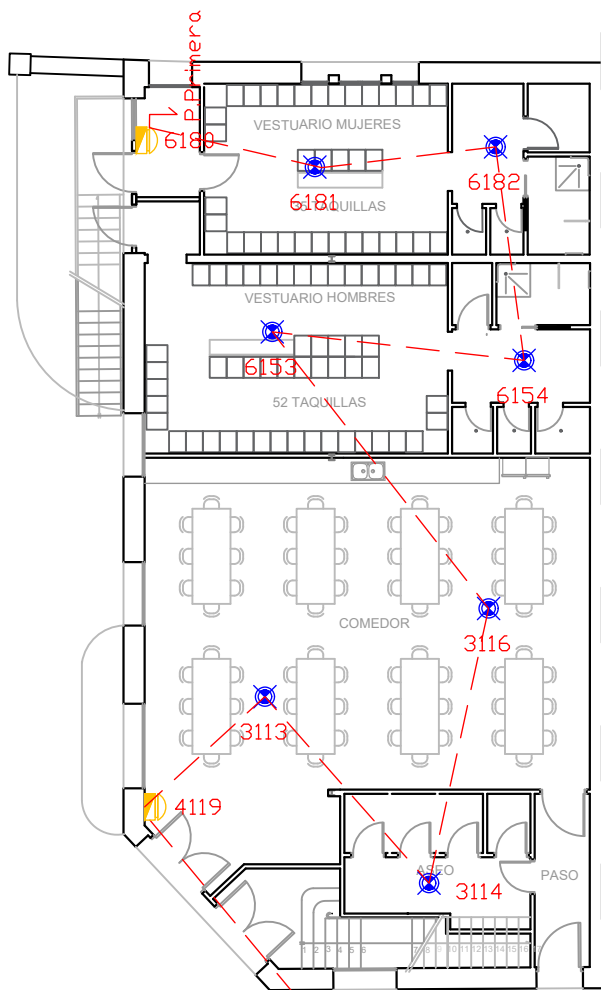
XYZZ

X: Número de la zona
Y: Número de lazo
ZZ: Número del detector.

- Aislador
- Sensor térmico analógico
- Pulsador alarma analógico
- Sensor óptico analógico direccionable
- Sirena incendio raja exterior
- Central analógica direccionable de 1 lazo

FECHA	NOMBRE	BENICARLO, S.A.	
DESEÑO	ELABORADO	(CASTELLÓN) SPAIN	
COMPROBADO	REVISADO		
Modelo	LABORAT-0-Rev-1		
1/100	CAPABILITY LAB	Escala: 1:100	

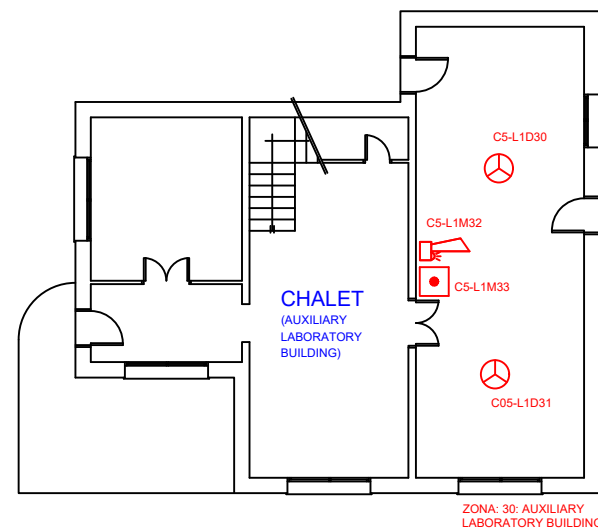
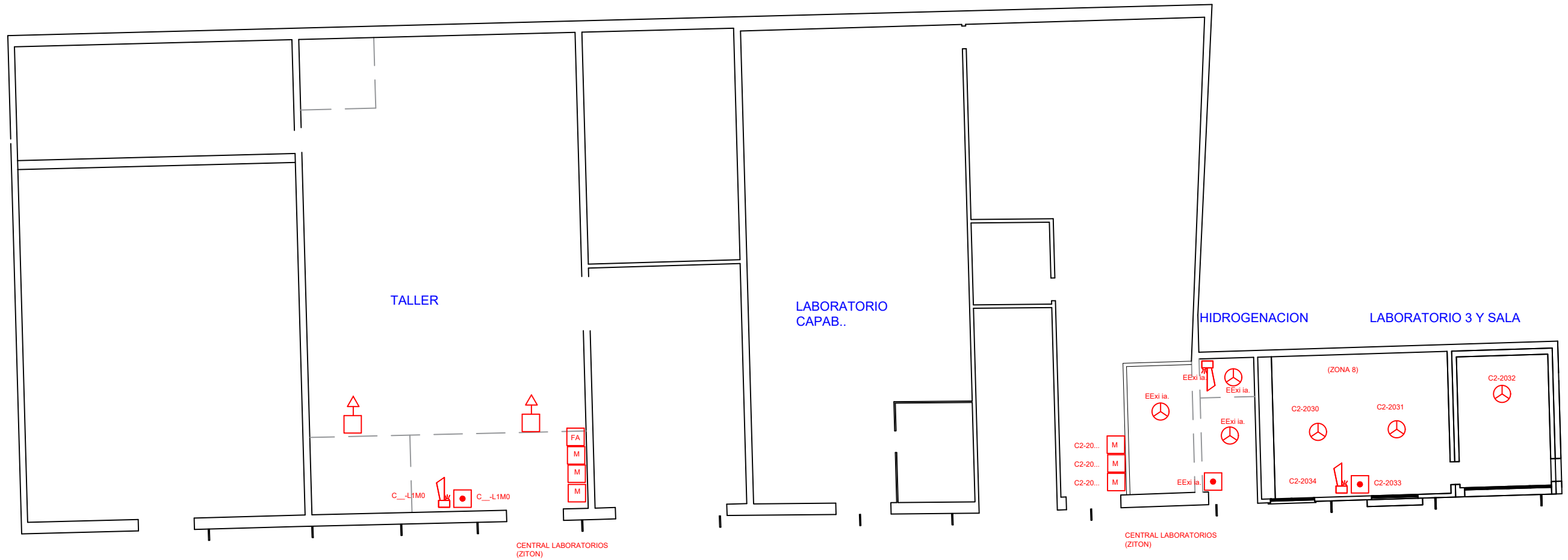
VESTUARIOS Y COMEDOR



XYZZ
 X: Número de la zona
 Y: Número de lazo
 5115 ZZ: Número del detector.

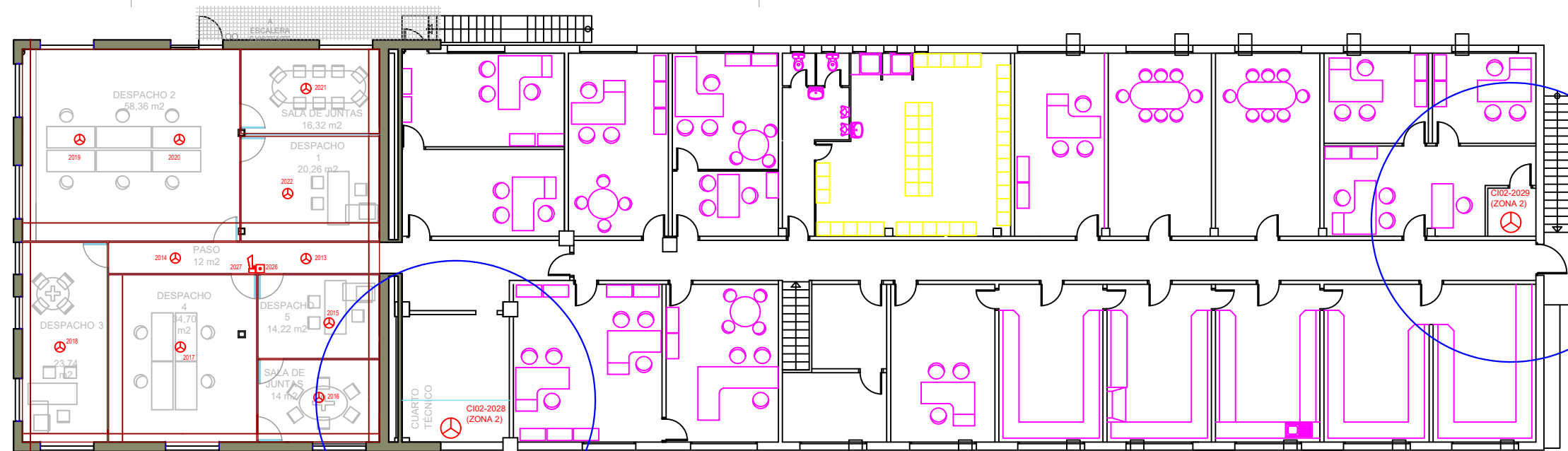
- Aislador
- Sensor térmico analógico
- Pulsador alarma analógico
- Sensor óptico analógico direccional
- Sirena incendio roja exterior
- Central analógica direccional de 1 lazo

FECHA	NOMBRE	
DISEÑADO	BY/AL/MS	VER/MSZ
COPIADO	BY/AL/MS	AR/MS
DIBUJADO POR:		
ESCALA	1/100	VESTUARIOS-COMEDOR
LABORADO POR:		

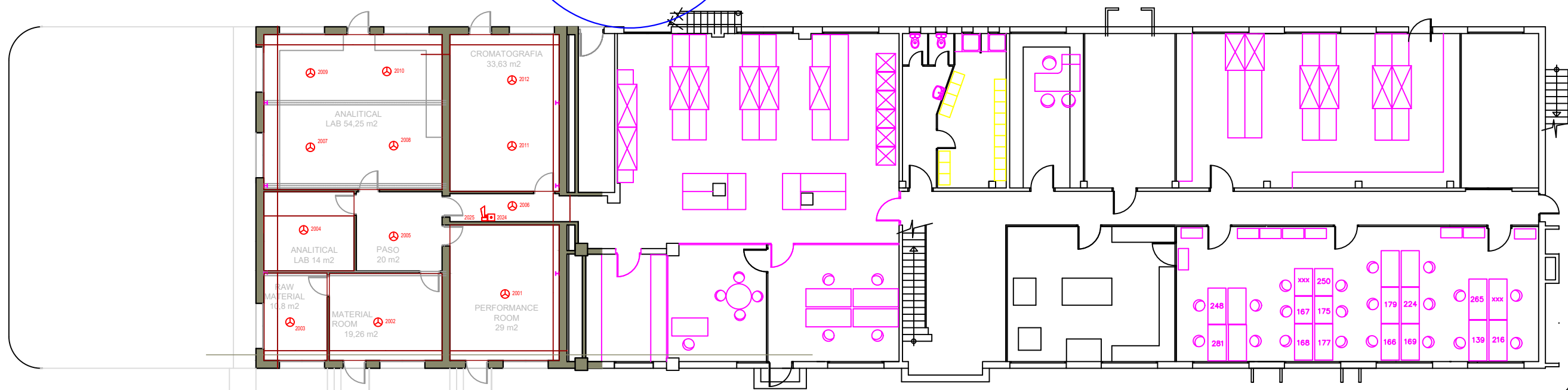


SIMBOLOGÍA	
	Detector óptico de humos
	Detector termovelocimétrico
	Pulsador de Alarma
	Sirena de Alarma óptico-acústica

REVISIÓN	FECHA	DESCRIPCIÓN	DIBUJADO	COMPROBADO	APROBADO
0	11/06/2020	PLANO SISTEMA DE SEGURIDAD	N.B.M.	J.H.R.	
		Securitas Seguridad España, S.A.	Usuario: IFF BENICARLÓ		
Nº de Obra:	120-2701-0473	Situación: AVD. FELIPE KLEIN, 2 BENICARLÓ			
Nº de Plano:	120-2701-DET01	Denominación: SISTEMA DE DETECCIÓN			
Formato:	A3	Plano: TALLER, AUX. LABORATORY BUILDING HIDROGENACION Y LABORATORIO 3			
Plano: 1	De: 1				
Nº de Plano:	Escala:				
1.1	1/200				

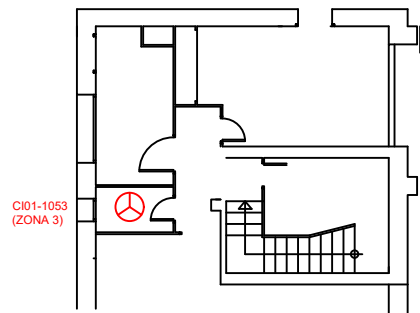


LABS-OFFICES - FIRST FLOOR



SERVIDOR IMPRESORAS Y EHS PL. 1ª

CENTRAL LABORATORIOS (ZITON)

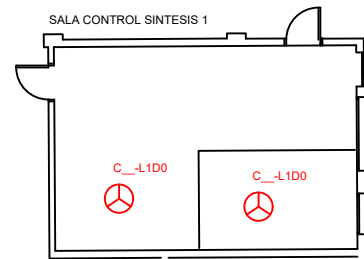


CI01-1053 (ZONA 3)

SERVIDOR OFICINAS PL. 1ª

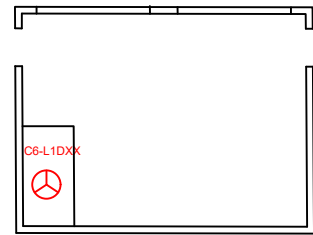
SIMBOLOGÍA	
	Detector óptico de humos
	Detector termovelocimétrico
	Pulsador de Alarma
	Sirena de Alarma óptico-acústica

REVISIÓN	FECHA	DESCRIPCIÓN	DIBUJADO	COMPROBADO	APROBADO																																										
0	11/06/2020	PLANO SISTEMA DE SEGURIDAD	N.B.M.	J.H.R.																																											
<table border="1"> <tr> <td colspan="2">Securitas Seguridad España, S.A.</td> <td colspan="4"> Usuario: IFF BENICARLÓ </td> </tr> <tr> <td>Nº de Obra:</td> <td>120-2701-0487</td> <td colspan="4"> Situación: AVD. FELIPE KLEIN, 2 BENICARLÓ </td> </tr> <tr> <td>Nº de Plano:</td> <td>120-2701-DET01</td> <td colspan="4"> Denominación: SISTEMA DE DETECCIÓN SALAS SERVIDORES </td> </tr> <tr> <td>Formato:</td> <td>A3</td> <td colspan="4"> Plano: SERVIDOR IMPRESORAS Y EHS PL. 1ª OFICINAS (PL. 1ª) </td> </tr> <tr> <td>Plano: 1</td> <td>De: 2</td> <td colspan="4"></td> </tr> <tr> <td>Nº de Plano:</td> <td>Escala:</td> <td colspan="4"></td> </tr> <tr> <td>1.1</td> <td>1/200</td> <td colspan="4"></td> </tr> </table>						Securitas Seguridad España, S.A.		Usuario: IFF BENICARLÓ				Nº de Obra:	120-2701-0487	Situación: AVD. FELIPE KLEIN, 2 BENICARLÓ				Nº de Plano:	120-2701-DET01	Denominación: SISTEMA DE DETECCIÓN SALAS SERVIDORES				Formato:	A3	Plano: SERVIDOR IMPRESORAS Y EHS PL. 1ª OFICINAS (PL. 1ª)				Plano: 1	De: 2					Nº de Plano:	Escala:					1.1	1/200				
Securitas Seguridad España, S.A.		Usuario: IFF BENICARLÓ																																													
Nº de Obra:	120-2701-0487	Situación: AVD. FELIPE KLEIN, 2 BENICARLÓ																																													
Nº de Plano:	120-2701-DET01	Denominación: SISTEMA DE DETECCIÓN SALAS SERVIDORES																																													
Formato:	A3	Plano: SERVIDOR IMPRESORAS Y EHS PL. 1ª OFICINAS (PL. 1ª)																																													
Plano: 1	De: 2																																														
Nº de Plano:	Escala:																																														
1.1	1/200																																														



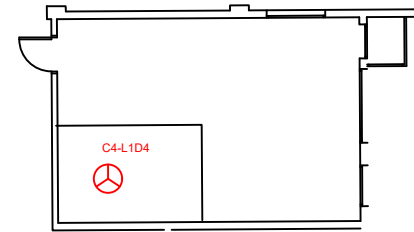
CENTRAL SÍNTESIS 1 EN SALA ELECTRICA (NOTIFIER)

SERVIDOR SÍNTESIS



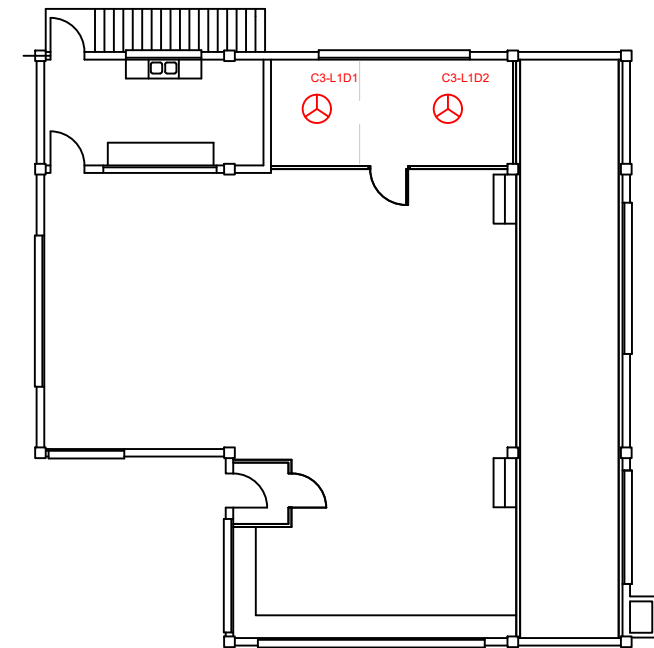
CENTRAL SÍNTESIS 1 EN SALA ELECTRICA (NOTIFIER)

SERVIDOR HIDROGENACION



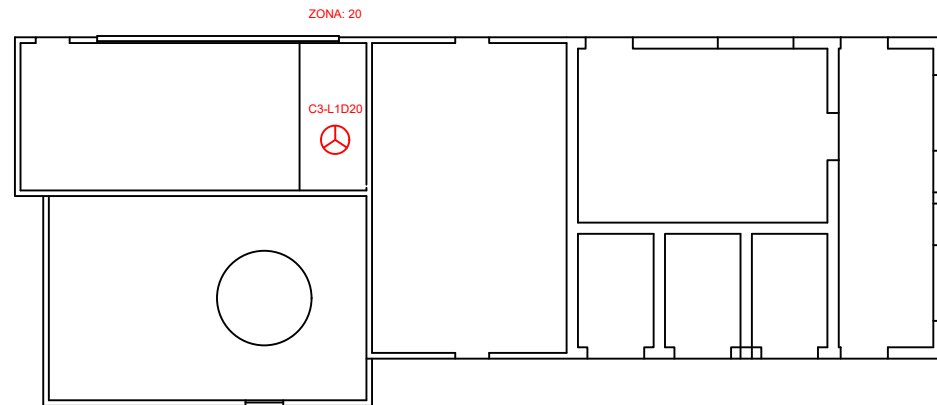
CENTRAL FINOS (NOTIFIER)

SERVIDOR FINOS



CENTRAL ESPECIALIDADES (NOTIFIER)

SERVIDOR ESPECIALIDADES



CENTRAL ESPECIALIDADES (NOTIFIER)

SERVIDOR DEPURADORA

SIMBOLOGÍA	
	Detector óptico de humos
	Detector termovelocimétrico
	Pulsador de Alarma
	Sirena de Alarma óptico-acústica

REVISIÓN	FECHA	DESCRIPCIÓN	DIBUJADO	COMPROBADO	APROBADO
0	11/06/2020	PLANO SISTEMA DE SEGURIDAD	N.B.M.	J.H.R.	
		Usuario: IFF BENICARLÓ			
Nº de Obra:	120-2701-0487	Situación: AVD. FELIPE KLEIN, 2 BENICARLÓ			
Nº de Plano:	120-2701-DET02	Denominación: SISTEMA DE DETECCIÓN SALAS SERVIDORES			
Formato:	A3	Plano: DEPURADORA, SÍNTESIS I, HIDROGENACION H820, FINOS, ESPECIALIDADES			
Plano: 2	De: 2				
Nº de Plano:	Escala:				
1.1	1/200				

DIRECCIONES EQUIPOS ANALOGICOS			
EQUIPO	TIPO	DIRECCION	FUNCION/DESCRIPCION
SENSOR (VIEW)	ENTRADA	S01	DETECTOR Z1
SENSOR (VIEW)	ENTRADA	S02	DETECTOR Z2
DETECTOR OPTICO	ENTRADA	S03	DETECTOR AMBIENTE SALA
H701	SALIDA	M01	ACTIVACION Z-1
H701	SALIDA	M02	ACTIVACION Z-2
H721	ENTRADA	M03	AVERIA F.A.
H721	ENTRADA	M04	AVERIA EQUIPO ASPIRACION
H721	SALIDA	M05	RESET EQUIPO ASPIRACION
SIRENA	SALIDA	M06	SIRENA SALA CONTROL
ITAC	ENTRADA	M10	PREACTIVO
ITAC	ENTRADA	M11	ACTIVO
ITAC	ENTRADA	M12	MODO ESPERA
ITAC	ENTRADA	M13	MODO MANUAL
ITAC	ENTRADA	M14	MODO ANULADO
ITAC	ENTRADA	M15	EXTINCION
ITAC	ENTRADA	M16	AVERIA GENERAL
H710	ENTRADA	M20	EQUIPO ASPIRACION



CILINDRO DE 67 LT CARGADO CON 50Kg DE CO2



LETTERO EXTINCION DISPARADA PAN 4 NOTIFIER



DETECTOR OPTICO ANALOGICO NF71 -OPT + BASE B 501 , NOTIFIER



PULSADOR DE PARO



PULSADOR DE DISPARO



PULSADOR DE INHIBICION



PULSADOR DE ALARMA ANALOGICO ,NOTIFIER



SA SIRENA OPTICO-ACUSTICA ANALOGICA NOTIFIER



SE SIRENA CONVENCIONAL OPTICO ACUSTICA,PRE-ALARMA EXTINCION, NOTIFIER



DETECTOR DE ASPIRACION VESDA , CON DISPLAY MODELO VEP-A10-P (1 CANAL/4 TUBOS)



CENTRAL DE EXTINCION RP1rSUPRA , NOTIFIER



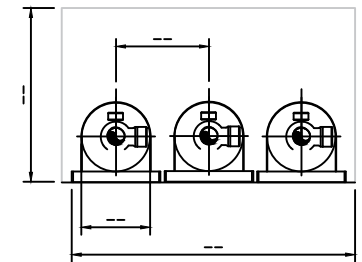
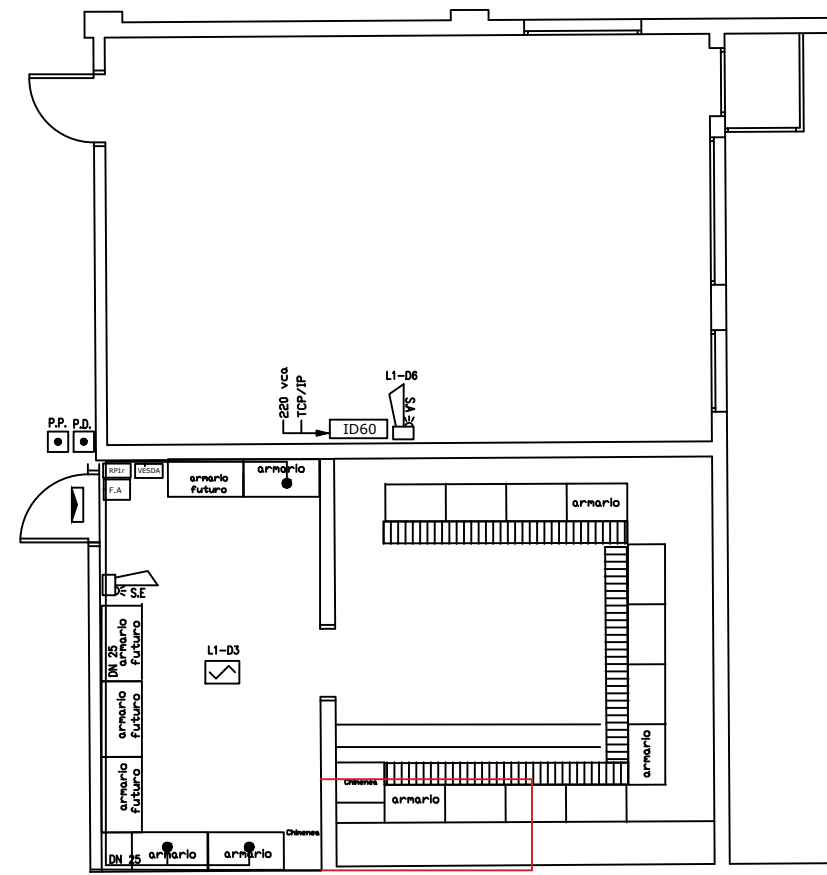
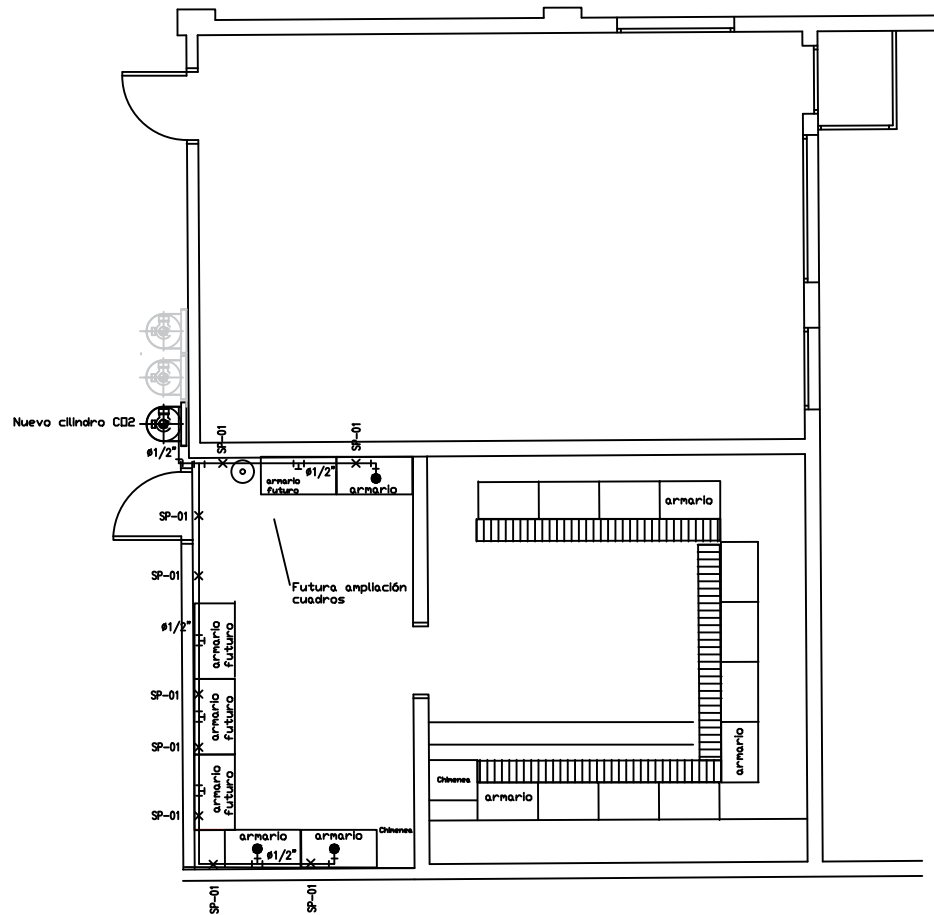
FUENTE DE ALIMENTACION 24 v 2, 5 Ah , NOTIFIER H1SP25



CENTRAL DE INCENDIOS ANALOGICA ID 60 , 1 LAZO, NOTIFIER

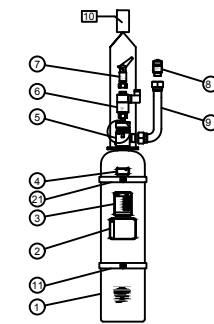


EXTINTOR DE 5 KGS DE CO2 , EFCANCA 89 B



CASETA EXISTENTE

DETALLE CASETA CILINDROS DE CO2 SIN ESCALA



ALZADO

Nº	DESIGNACION/DENOMINACION
1	Cilindro extintor/Extintor cilindro a) P/total/más b) Reserva/Reserva
2	Pageina cilindra instrucciones/instrucciones cilindro cilindro
3	Pageina cilindra manual CO2/CO2 manual cilindro cilindro
4	Pageina cilindra manual P/VI manual cilindro cilindro
5	Valvula de cilindro/Cylinder valve
6	Cabezal de disparo cilindro/assembly/As-ensamblaje cilindro cilindro
7	Cabezal de disparo manual/Manual extintor base
8	Valvula de retencion/Check valve
9	Llave de descarga/Discharge base
10	Pinza montado
11	Soportes para cilindro

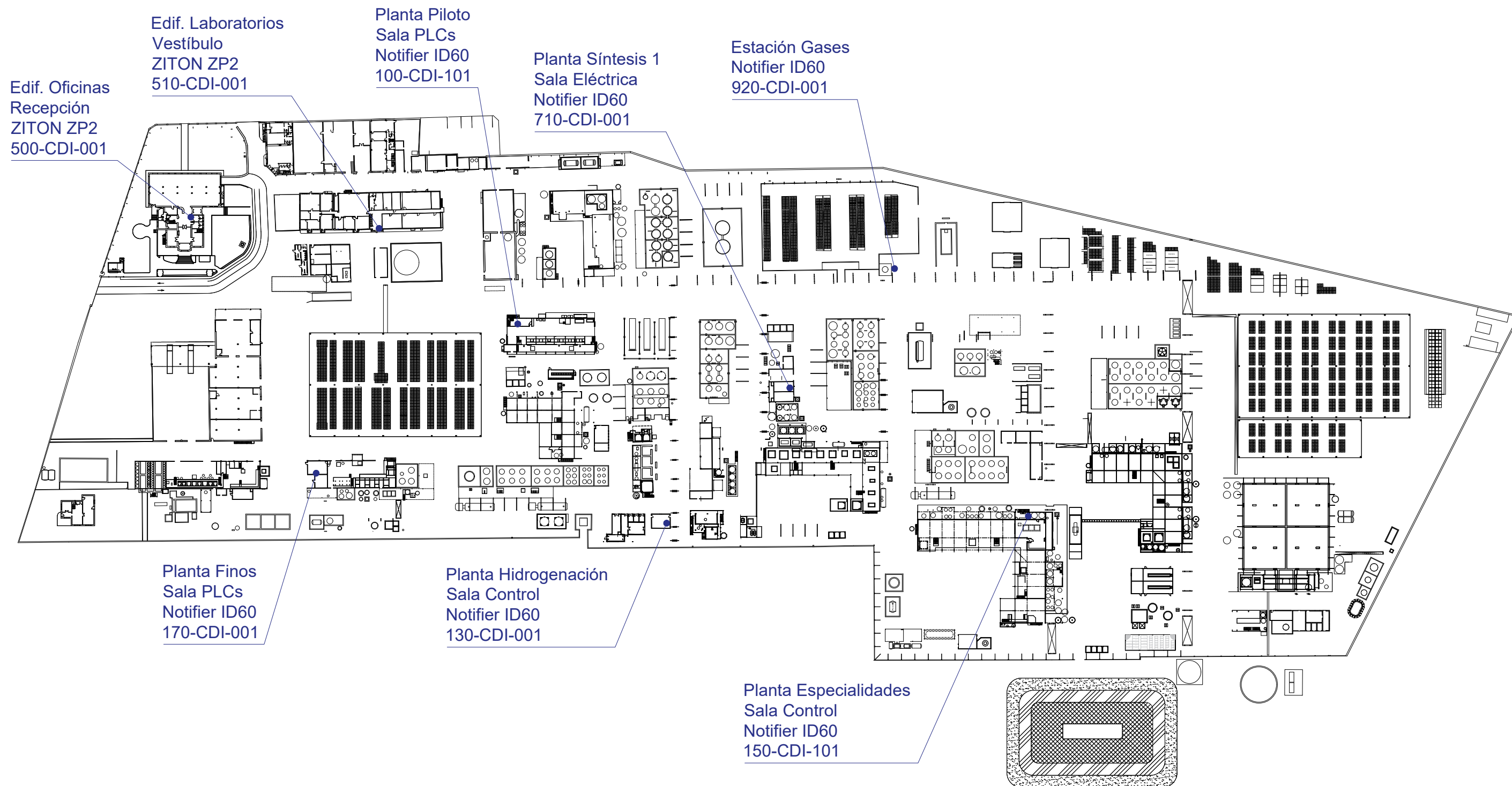


PLANTA

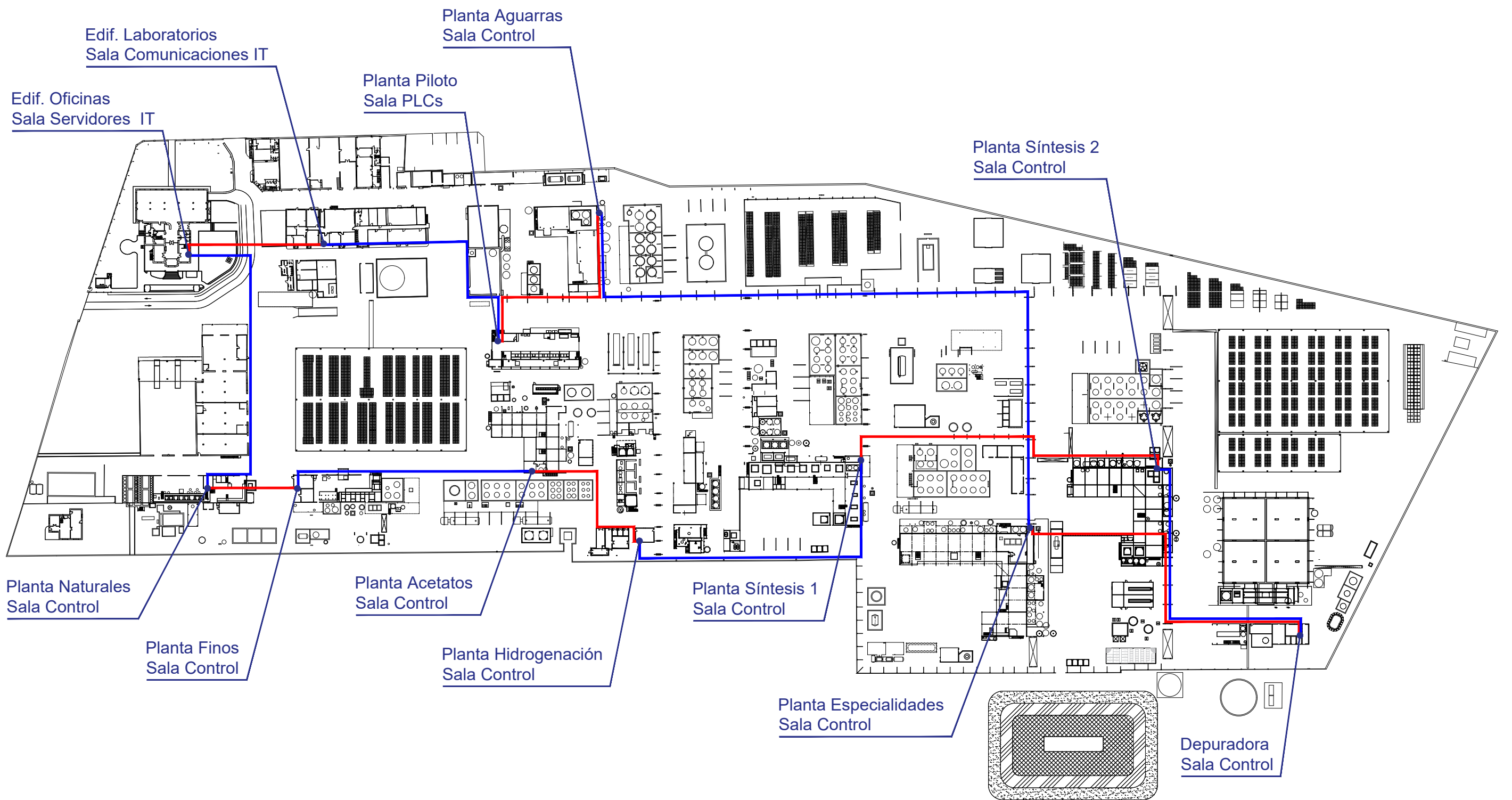
DETALLE CILINDRO SIN ESCALA


REVISION	FECHA	DESCRIPCION	DELLADO	COMPROBADO	APROBADO
1	31/07/2018	AS-BUILT	J.M.R.	F.M.G.	
0	22/05/2018	PARA APROBACION	J.M.R.	F.M.G.	

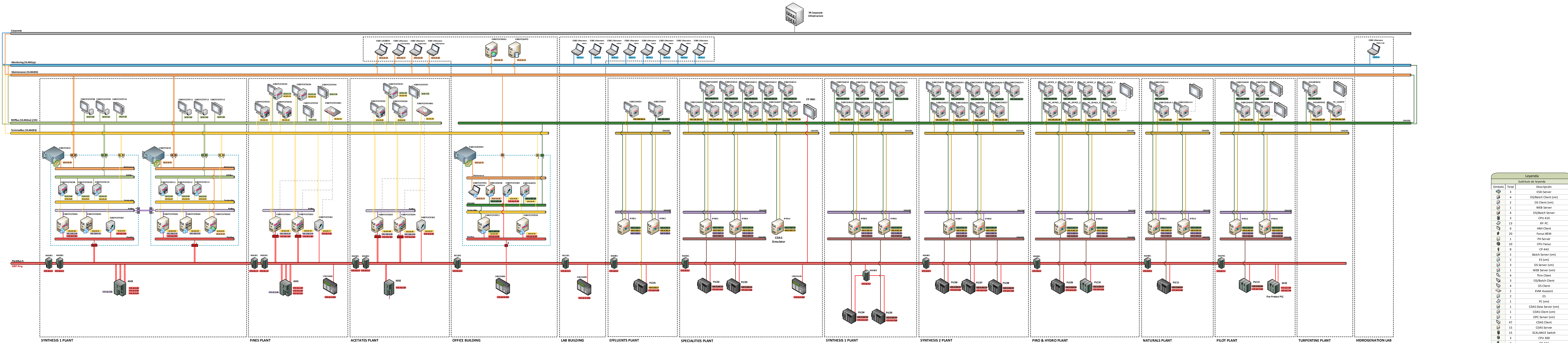
		Usuario: IFF BENICARLÓ	
Nº de Plano: 118-2701-0373	Situación: AV. FELIPE KLEIN, 2 BENICARLÓ		
Nº de obra 118-2701-373-001	Denominación: SISTEMA EXTINCION INCENDIOS CUADROS EN SALA ELECTRICA		
Formato: A1	Plano: 1 De: 2		
Nº de Plano: Escala: 1.1 S/E	Plano: PLANTA FINOS		



	Fecha	Nombre	 Monitorización de un Sistema de Detección de Incendios
Dibujado	10/02/2021	AAG	
Revisado	10/02/2021	LBM	
Aprobado	10/02/2021	JSP	
Escala N/A	Distribución CDIs		Plano 3



	Fecha	Nombre	 iff Benicarló
Dibujado	10/02/2021	AAG	
Revisado	10/02/2021	LBM	
Aprobado	10/02/2021	JSP	Monitorización de un Sistema de Detección de Incendios
Escala N/A	Trazado Fibra Óptica		Plano 4



Legenda		
Subtitulo de leyenda		
Simbolo	Total	Descripcion
[Server Icon]	3	ESXi Server
[Client Icon]	4	OS/Batch Client (vm)
[Client Icon]	2	OS Client (vm)
[Client Icon]	1	PH Server
[Server Icon]	4	OS/Batch Server
[Client Icon]	3	CPU 410
[Client Icon]	13	IFF PC
[Client Icon]	6	HMI Client
[Client Icon]	20	Fanuc BEM
[Client Icon]	1	PH Server
[Client Icon]	10	CPU Fanuc
[Client Icon]	8	CP-443
[Client Icon]	2	Batch Server (vm)
[Client Icon]	2	ES (vm)
[Client Icon]	3	OS Server (vm)
[Client Icon]	1	WEB Server (vm)
[Client Icon]	4	Thin Client
[Client Icon]	3	OS/Batch Client
[Client Icon]	4	OS Client
[Client Icon]	2	KVM Account
[Client Icon]	2	ES
[Client Icon]	1	PC (vm)
[Client Icon]	1	CDAS Data Server (vm)
[Client Icon]	1	CDAS Client (vm)
[Client Icon]	1	OPC Server (vm)
[Client Icon]	47	CDAS Client
[Client Icon]	15	CDAS Server
[Client Icon]	15	SCALANCE Switch
[Client Icon]	3	CPU 300
[Client Icon]	3	CP-343
[Client Icon]	4	Fire Alarm Panel

Monitorización de un Sistema de Detección de Incendios

PLIEGO DE CONDICIONES



UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2021

4 Pliego de Condiciones

4.1	Condiciones Técnicas.....	298
4.1.1	<i>Condiciones de la Instalación Eléctrica.....</i>	298
4.1.2	<i>Condiciones de la Instalación del PLC.....</i>	298
4.1.3	<i>Condiciones del Cableado.....</i>	299
4.1.4	<i>Condiciones de Alimentación a 24V.....</i>	299
4.2	Condiciones Facultativas.....	299
4.2.1	<i>Responsabilidades del Contratista.....</i>	299
4.2.2	<i>Responsabilidades del Contratante.....</i>	299
4.3	Condiciones Económicas.....	299
4.3.1	<i>Precios.....</i>	299
4.3.2	<i>Revisión de Precios.....</i>	300
4.3.3	<i>Penalizaciones.....</i>	300
4.4	Condiciones Legales.....	300
4.4.1	<i>Contrato.....</i>	300
4.4.2	<i>Rescisión de Contrato.....</i>	300

4 Pliego de Condiciones

Este documento recoge los detalles a cumplir por todas las partes intervinientes en el proyecto. Se abarcará diferenciando cuatro apartados: condiciones técnicas, condiciones facultativas, condiciones económicas y condiciones legales.

- Condiciones Técnicas: incluyen las características de los materiales y los procedimientos a emplear a lo largo de la realización del trabajo.
- Condiciones Facultativas: recogen los derechos y obligaciones de las partes.
- Condiciones Económicas: se presentan las formas de pago e indemnizaciones.
- Condiciones Legales: hacen referencia al perfil de contratista, la forma de adjudicación, el tipo de contrato, la obligatoriedad de suscripción de seguros de responsabilidad civil y otros asuntos relacionados.

4.1 Condiciones Técnicas

4.1.1 Condiciones de la Instalación Eléctrica.

Los elementos que componen el sistema y que estarán conectados a la red eléctrica, se regirán por el Reglamento Electrotécnico de Baja Tensión (RBT) y sus Instrucciones Complementarias. También se deben considerar las siguientes normas UNE y DIN:

ITC-BT-18: instalaciones de puesta a tierra.

ITC-BT-19: instalaciones interiores o receptoras. Prescripciones generales.

ITC-BT-20: instalaciones interiores o receptoras. Sistemas de instalación.

ITC-BT-22: instalaciones interiores o receptoras. Protección contra sobre-intensidades.

ITC-BT-23: instalaciones interiores o receptoras. Protección contra sobretensiones.

ITC-BT-24: instalaciones interiores o receptoras. Protección contra contactos directos e indirectos.

ITC-BT-43: instalación de receptores. Prescripciones generales.

ITC-BT-51: instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad.

UNE 20 514 1M: reglas de seguridad para aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uno general análogo conectado a una red de energía.

4.1.2 Condiciones de la Instalación del PLC

El armario donde se instale el PLC deberá garantizar unas condiciones ambientales adecuadas, correspondientes a las recomendaciones del fabricante:

- Se deberá asegurar unas temperaturas de la envolvente comprendidas entre 10°C y 50°C.
- Deberá asegurarse una correcta sujeción de los elementos para evitar desprendimientos y vibraciones.
- Tendrá que estar protegido contra el polvo y los agentes corrosivos.

4.1.3 Condiciones del Cableado

- Los cables alimentados con tensiones continuas dispondrán de diferente canalización que los de alterna.
- Las canalizaciones para cables de comunicaciones serán independientes del resto y se marcarán con un color distintivo.
- Los colores para los cables de tensión continua serán siempre: rojo para el polo positivo y negro para el polo negativo, sin poder hacer uso de ellos en cables de corriente alterna.

4.1.4 Condiciones de Alimentación a 24V

- Las oscilaciones de la tensión de red respecto al valor nominal de la fuente de alimentación a 24V deben encontrarse dentro del margen de tolerancia admisible.
- Entre la conexión de la salida de la fuente y los elementos finales se instalarán magneto-térmicos que permitan aislar los elementos de forma individual.

4.2 Condiciones Facultativas**4.2.1 Responsabilidades del Contratista**

Debe estar informado y actuar según las leyes y normas que rigen su actividad profesional, así como conocer las especificaciones técnicas y normas de seguridad aplicables a los elementos del proyecto.

Tiene la responsabilidad de comprobar que los elementos utilizados cumplan los requisitos recogidos en el proyecto y es su función realizar las comprobaciones especificadas en el apartado anterior antes de dar el visto bueno.

El trabajo de programación deberá ser ejecutado por personas especialmente preparadas y con conocimientos suficientes para el desarrollo del mismo.

El contratista será responsable del cumplimiento de todas las especificaciones indicadas en la memoria del proyecto, así como de todos aquellos perjuicios que se puedan derivar de una incorrecta programación, no teniendo derecho a recibir pago alguno por el coste derivado de cualquier modificación para el cumplimiento de las especificaciones de la memoria.

4.2.2 Responsabilidades del Contratante

Deberá facilitar todos los elementos e información de que disponga que estén relacionados con el proyecto al Contratista. Debe también entregar por escrito las especificaciones del proyecto que desea que se lleven a cabo.

No podrá reclamar por retrasos en el proyecto causados por motivos ajenos al contratista. Estos motivos deberán de poderse justificar. En el caso de que se produzcan retrasos y no se justifiquen se producirán reducciones en el precio del proyecto según lo establecido en el siguiente apartado.

4.3 Condiciones Económicas**4.3.1 Precios**

El contratista presentará, al formalizarse el contrato, relación de los precios de las unidades de obra que integran el proyecto, los cuales de ser aceptados tendrán valor contractual y se aplicarán a las posibles variaciones que pueda haber. Estos precios unitarios, se entiende que comprenden la ejecución total de la unidad de obra, incluyendo todos los trabajos aún los complementarios y los materiales, así como la parte proporcional de imposición fiscal, las cargas laborales y otros gastos repercutibles.

4.3.2 *Revisión de Precios*

El contratista tiene derecho a revisión de precios si en el momento de la ejecución de la obra estos han variado más de un 3% respecto a lo establecido en el presupuesto.

4.3.3 *Penalizaciones*

Por retraso en los plazos de entrega de las obras, se establecen tablas de penalización cuyas cuantías y demoras se fijan a continuación. Por retrasos en el pago desde la conclusión del proyecto, el contratista tendrá derecho a aplicar recargos sobre el presupuesto total del proyecto

Entre 1 y 7 días	Sin recargo
Entre 8 y 15 días	1%
Entre 16 y 25 días	2%
Entre 26 y 50 días	4%
Entre 51 y 100 días	8%
Entre 101 y 200 días	20%
Entre 201 y 300 días	30%

En caso de impago tras 300 días naturales desde la conclusión del proyecto el Contratista tiene derecho a realizar una demanda ante los tribunales.

Si se producen retrasos en la finalización del proyecto según los plazos establecidos por motivos no justificados, se reducirá el precio en función de lo siguiente:

Entre 1 y 7 días	Sin reducción
Entre 8 y 15 días	2%
Entre 16 y 25 días	3%
Entre 26 y 50 días	5%
Entre 51 y 100 días	15%

En caso de no haber finalizado el proyecto después de 101 según los plazos acordados, siempre que los retrasos no sean por causas justificadas ajenas al Contratista, el contratante tendrá derecho a rescindir el contrato.

4.4 **Condiciones Legales**

4.4.1 *Contrato*

El contrato recogerá el precio final del proyecto recogido en el presupuesto junto con las cláusulas vistas en el apartado anterior en caso de retrasos del pago. Se realizará por escrito y deberá ser firmado por todas las partes implicadas. También recogerá todas las cláusulas que se negocien entre las partes.

4.4.2 *Rescisión de Contrato*

Para la posible rescisión del contrato se deberá de dar una de las siguientes condiciones:
 - Acuerdo entre ambas partes para rescindir el contrato - Modificación de los aspectos fundamentales del proyecto acordados - Incumplimiento de los plazos establecidos - Impago - Actuaciones con mala fe Por supuesto, la decisión de rescindir el contrato si se da una de estas condiciones la tendrá la parte perjudicada.

Monitorización de un Sistema de Detección de Incendios

PRESUPUESTO



UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2021

5 Presupuesto

5.1	Justificación de Precios.....	303
5.1.1	<i>Mano de obra.....</i>	303
5.1.2	<i>Materiales.....</i>	303
5.1.3	<i>Maquinaria.....</i>	304
5.1.4	<i>Licencias de Software.....</i>	305
5.2	Mediciones	306
5.2.1	<i>Instalación Infraestructura de red.....</i>	306
5.2.2	<i>Instalación Switches</i>	307
5.2.3	<i>Instalación PLC.....</i>	307
5.2.4	<i>Instalación Servidor OS.....</i>	308
5.2.5	<i>Estudio y Desarrollo del Programa</i>	308
5.3	Capítulos	309
5.3.1	<i>Instalación Infraestructura de red.....</i>	309
5.3.2	<i>Instalación Switches</i>	310
5.3.3	<i>Instalación PLC.....</i>	311
5.3.4	<i>Instalación Servidor OS.....</i>	312
5.3.5	<i>Estudio y Desarrollo del Programa</i>	313
5.4	Resumen del Presupuesto.....	314

5 Presupuesto

5.1 Justificación de Precios

5.1.1 Mano de obra

Código	Concepto	Ud.	Importe (€)
MO1001	Oficial 1ª Electricista	H	22,60
MO1002	Ayudante Electricista	H	18,70
MO1003	Técnico Programador	H	45,00

5.1.2 Materiales

Código	Concepto	Ud.	Importe (€)
MT1001	Cable Fibra Óptica 08x10 NEXO (DP). Cable de 8 fibras Multimodo OM3 con grado GC	ML	0,60
MT1002	Cable U/FTP CAT6 EXTERIOR PE DK NEGRO	ML	0,67
MT1003	Conector RJ45 macho CAT6 con guía PK100	UD	0,36
MT1004	Caja 3M Caja Mural VF45 12 Fibras.	UD	22,45
MT1005	Empalme Universal Fibra Óptica	UD	5,52
MT1006	PIGTAIL Fibra Óptica Multimodo 62.5	UD	3,80
MT1007	Pasamuros (Raccord) ST Simple Multimodo	UD	1,00
MT1008	Latiguillo FO Multimodo ST/ST 62,5/125, 1M	UD	10,83
MT1009	Latiguillo FO Multimodo ST/LC OM1 62,5/125, 5M	UD	14,10
MT1010	Patch Panel 24P Fibra Óptica vacía	UD	40,65
MT1011	Cassette Union 12 FO Apil Para Patch Panel	UD	6,38
MT1012	030008 Canaleta DLP 20x12,5-2,10mm PVC	ML	0,59
MT1013	LEGRAND 30017 Canaleta DLP 32x20-2,10mm PVC	ML	1,37
MT1014	Collarín 102x7,7x376 NEGRO 2273-0	UD	0,14
MT1015	Collarín 50x7,7x213 NEGRO 2271-0	UD	0,06
MT1016	Brida uso Ext.PA 6.6W 45x4,8x188 NEGRO	UD	0,02
MT1017	Envase-Soporte Autorroscable EM-7	UD	0,13
MT1018	Cartucho 6,8/11M- ROJO	UD	0,22
MT1019	Clavo X-EM6H-11-9 FP8 P/ACERO X-EM6	UD	0,34

Código	Concepto	Ud.	Importe (€)
MT1020	Taco SX 6 d.6	UD	0,03
MT1021	Tornillo ABC SPAX 4X30	UD	0,02
MT1022	Cinta TEMFLEX 1300 19mmx25m PVC NG.0,13mm	UD	0,64
MT1023	Rollo Cinta DYMO REF.18764/18483/16959 12mmx7m	UD	4,05
MT1024	Cable Flexible Libre Halógenos 1,5 mm ² EXZHELLENT CPR 750V General Cable	ML	0,20
MT1025	Automático Magneto-térmico iC60N 10A 2P SCHNEIDER	UD	12,56
MT2001	6ES7390-1AE80-0AA0. SIMATIC S7-300, Perfil soporte longitud =480mm	UD	26,71
MT2002	6ES7307-1EA01-0AA0. SIMATIC S7-300, PS 307 de 5 A, fuente de alimentación estabilizada, entrada: AC 120/230 V, salida: DC 24 V/5 A.	UD	117,68
MT2003	6ES7317-2EK14-0AB0. SIMATIC CPU 317-2 PN/DP, Módulo central con 1 MByte. Memoria principal, Interfaz 1: MPI/DP 12 Mbits/s, Interfaz 2: Ethernet Profinet, con Switch de 2 Puertos.	UD	3.758,16
MT2004	6ES7953-8LM32-0AA0. SIMATIC S7, Micro Memory Card para S7-300/C7/ET. 200, 3, 3 V Nflash, 4 Mbytes.	UD	303,18
MT2005	6GK7343-1EX30-0XE0. SIMATIC NET CP343-1, procesador de comunicaciones para la conexión de SIMATIC S7-300 a Ethernet Industrial vía ISO y TCP/IP, controlador PN IO o dispositivo PROFINET-IO, switch de 2 puertos ERTEC200 integrado, comunicación S7, FETCH/WRITE, SEND/RCV con y sin RFC1006, multicast DHCP, sincronización de CPU vía NTC, diagnóstico inicialización vía LAN, 2 x conexiones RJ45 para LAN a 10/100 Mbit/s.	UD	1.120,57
MT3001	6GK5992-1AL00-8AA0. Accesorios para SCALANCE X; Transceptor enchufable SFP992-1; 1 puerto LC a 1000 Mbits/s, óptico; multimodo, vidrio, hasta máx. 750 m.	UD	157,34
MT3002	6GK5206-2BS00-2AC2. SCALANCE XC206-2SFP Layer 2 IE Switch gestionable; 6 puertos RJ45 10/100 Mbits/s; 2 puertos SFP 100/1000 Mbits/s; 1 puerto de consola; LED de diagnóstico; Alimentación redundante; rango de temperatura de -40 °C a +70 °C; montaje: perfil DIN/soporte S7/pared Funciones de redundancia Office características (RSTP, VLAN, ...); dispositivo PROFINET IO; conforme con Ethernet/IP; caja de conector C;	UD	666,22

5.1.3 Maquinaria

Código	Concepto	Ud.	Importe (€)
MQ1001	Plataforma Elevadora HAULOTTE Gasoil x día	D	148,00

5.1.4 Licencias de Software

Código	Concepto	Ud.	Importe (€)
LS1001	6ES7658-2BA58-0YA0. SIMATIC PCS7, Software, OS software Server v9.0 (PO 100) Lic. Indiv. p. 1 instalación SW RT, clave de lic. en lapiz usb, clase a, 5 idiomas (al, in, fr, it, es), ejecutable bajo win 7 ultimate win 10 enterprise ltsb para más informaciones ver PCS7 v9.0	UD	9.916,19
LS1002	6ES7658-2XB00-0XB0. SIMATIC PCS7, software RT license OS (PO 1000) licencia Indiv. p. 1 instalación SW RT, sin sw, sin docum. clave licencia en stick usb.	UD	2.643,87
LS1003	6ES7650-1CD58-2YB5. SIMATIC PCS7, Software, BCE V9.0 lic. flotante P. 1 usuario SW RT, sin SW, sin docum. clave de lic. en lapiz usb, clase a, 3 idiomas (al, in, fr), ejecutable bajo win 7 ultimate win 10 enterprise ltsb server 2012 R2, Server 2016 Standard.	UD	809,69

5.2 Mediciones

5.2.1 Instalación Infraestructura de red

Código	Concepto	Ud.	Cantidad
MT1001	Cable Fibra Óptica 08x10 NEXO (DP). Cable de 8 fibras Multimodo OM3 con grado GC	ML	3.575,00
MT1002	Cable U/FTP CAT6 EXTERIOR PE DK NEGRO	ML	190,00
MT1003	Conector RJ45 macho CAT6 con guía PK100	UD	30,00
MT1004	Caja 3M Caja Mural VF45 12 Fibras.	UD	18,00
MT1005	Empalme Universal Fibra Óptica	UD	200,00
MT1006	PIGTAIL Fibra Óptica Multimodo 62.5	UD	200,00
MT1007	Pasamuros (Raccord) ST Simple Multimodo	UD	200,00
MT1008	Latiguillo FO Multimodo ST/ST 62,5/125, 1M	UD	20,00
MT1009	Latiguillo FO Multimodo ST/LC OM1 62,5/125, 5M	UD	30,00
MT1010	Patch Panel 24P Fibra Óptica vacía	UD	3,00
MT1011	Cassette Union 12 FO Apil Para Patch Panel	UD	24,00
MT1012	030008 Canaleta DLP 20x12,5-2,10mm PVC	ML	12,00
MT1013	LEGRAND 30017 Canaleta DLP 32x20-2,10mm PVC	ML	12,00
MT1014	Collarín 102x7,7x376 NEGRO 2273-0	UD	750,00
MT1015	Collarín 50x7,7x213 NEGRO 2271-0	UD	800,00
MT1016	Brida uso Ext.PA 6.6W 45x4,8x188 NEGRO	UD	350,00
MT1017	Envase-Soporte Autorroscable EM-7	UD	250,00
MT1018	Cartucho 6,8/11M- ROJO	UD	150,00
MT1019	Clavo X-EM6H-11-9 FP8 P/ACERO X-EM6	UD	150,00
MT1020	Taco SX 6 d.6	UD	150,00
MT1021	Tornillo ABC SPAX 4X30	UD	150,00
MT1022	Cinta TEMFLEX 1300 19mmx25m PVC NG.0,13mm	UD	7,00
MT1023	Rollo Cinta DYMO REF.18764/18483/16959 12mmx7m	UD	14,00

5.2.2 Instalación Switches

Código	Concepto	Ud.	Cantidad
MT1024	Cable Flexible Libre Halógenos 1,5 mm ² EXZHELLENT CPR 750V General Cable	ML	20,00
MT1025	Automático Magneto-térmico iC60N 10A 2P SCHNEIDER	UD	14,00
MT3001	6GK5992-1AL00-8AA0. Accesorios para SCALANCE X; Transceptor enchufable SFP992-1; 1 puerto LC a 1000 Mbits/s, óptico; multimodo, vidrio, hasta máx. 750 m	UD	28,00
MT3002	6GK5206-2BS00-2AC2. SCALANCE XC206-2SFP Layer 2 IE Switch gestionable; 6 puertos RJ45 10/100 Mbits/s; 2 puertos SFP 100/1000 Mbits/s; 1 puerto de consola; LED de diagnóstico; Alimentación redundante; rango de temperatura de -40 °C a +70 °C; montaje: perfil DIN/soporte S7/pared Funciones de redundancia Office características (RSTP, VLAN, ...); dispositivo PROFINET IO; conforme con Ethernet/IP; caja de conector C	UD	14,00

5.2.3 Instalación PLC

Código	Concepto	Ud.	Cantidad
MT1024	Cable Flexible Libre Halógenos 1,5 mm ² EXZHELLENT CPR 750V General Cable	ML	2,00
MT1025	Automático Magneto-térmico iC60N 10A 2P SCHNEIDER	UD	1,00
MT2001	6ES7390-1AE80-0AA0. SIMATIC S7-300, Perfil soporte longitud =480mm	UD	1,00
MT2002	6ES7307-1EA01-0AA0. SIMATIC S7-300, PS 307 de 5 A, fuente de alimentación estabilizada, entrada: AC 120/230 V, salida: DC 24 V/5 A.	UD	1,00
MT2003	6ES7317-2EK14-0AB0. SIMATIC CPU 317-2 PN/DP, Módulo central con 1 MByte. Memoria principal, Interfaz 1: MPI/DP 12 Mbits/s, Interfaz 2: Ethernet Profinet, con Switch de 2 Puertos.	UD	1,00
MT2004	6ES7953-8LM32-0AA0. SIMATIC S7, Micro Memory Card para S7-300/C7/ET. 200, 3, 3 V Nflash, 4 Mbytes.	UD	1,00
MT2005	6GK7343-1EX30-0XE0. SIMATIC NET CP343-1, procesador de comunicaciones para la conexión de SIMATIC S7-300 a Ethernet Industrial vía ISO y TCP/IP, controlador PN IO o dispositivo PROFINET-IO, switch de 2 puertos ERTEC200 integrado, comunicación S7, FETCH/WRITE, SEND/RCV con y sin RFC1006, multicast DHCP, sincronización de CPU vía NTC, diagnóstico inicialización vía LAN, 2 x conexiones RJ45 para LAN a 10/100 Mbit/s.	UD	1,00

5.2.4 Instalación Servidor OS

Código	Concepto	Ud.	Cantidad
LS1001	6ES7658-2BA58-0YA0. SIMATIC PCS7, Software, OS software Server v9.0 (PO 100) Lic. Indiv. p. 1 instalación SW RT, clave de lic. en lapiz usb, clase a, 5 idiomas (al, in, fr, it, es), ejecutable bajo win 7 ultimate win 10 enterprise ltsb para más informaciones ver PCS7 v9.0	UD	1,00
LS1002	6ES7658-2XB00-0XB0. SIMATIC PCS7, software RT license OS (PO 1000) licencia Indiv. p. 1 instalación SW RT, sin sw, sin docum. clave licencia en stick usb.	UD	1,00
LS1003	6ES7650-1CD58-2YB5. SIMATIC PCS7, Software, BCE V9.0 lic. flotante P. 1 usuario SW RT, sin SW, sin docum. clave de lic. en lapiz usb, clase a, 3 idiomas (al, in, fr), ejecutable bajo win 7 ultimate win 10 enterprise ltsb server 2012 R2, Server 2016 Standard	UD	1,00

5.2.5 Estudio y Desarrollo del Programa

Código	Concepto	Ud.	Cantidad
SW1001	Estudio y planificación	H	80,00
SW1002	Desarrollo de librería PLC (FBs) y pruebas	H	280,00
SW1003	Desarrollo de librería Gráfica (Block Icons y Faceplates) y pruebas	H	160,00
SW1004	Programación PLC	H	80,00
SW1005	Comisionado y puesta en marcha	H	40,00

5.3 Capítulos

5.3.1 Instalación Infraestructura de red

C.01		Instalación Infraestructura de red				14.287,91
Mano de obra:						6.425,00
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MO1001	Oficial 1ª Electricista	H	185,00	22,60	4181,00	
MO1002	Ayudante Electricista	H	120,00	18,70	2244,00	
Subtotal:					6425,00	
Materiales:						6.034,43
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MT1001	Cable Fibra Óptica 08x10 NEXO (DP).Cable de 8 fibras Multimodo OM3 con grado GC	ML	3.575,00	0,60	2.145,00	
MT1002	Cable U/FTP CAT6 EXTERIOR PE DK NEGRO	ML	190,00	0,67	127,30	
MT1003	Conector RJ45 macho CAT6 con guía PK100	UD	30,00	0,36	10,80	
MT1004	Caja 3M Caja Mural VF45 12 Fibras.	UD	18,00	22,45	404,10	
MT1005	Empalme Universal Fibra Óptica	UD	200,00	5,52	1104,00	
MT1006	PIGTAIL Fibra Óptica Multimodo 62.5	UD	200,00	3,80	760,00	
MT1007	Pasamuros (Raccord) ST Simple Multimodo	UD	200,00	1,00	200,00	
MT1008	Latiguillo FO Multimodo ST/ST 62,5/125, 1M	UD	20,00	10,83	216,60	
MT1009	Latiguillo FO Multimodo ST/LC OM1 62,5/125, 5M	UD	30,00	14,10	423,00	
MT1010	Patch Panel 24P Fibra Óptica vacía	UD	3,00	40,65	121,95	
MT1011	Cassette Union 12 FO Apil Para Patch Panel	UD	24,00	6,38	153,12	
MT1012	030008 Canaleta DLP 20x12,5-2,10mm PVC	ML	12,00	0,59	7,08	
MT1013	LEGRAND 30017 Canaleta DLP 32x20-2,10mm PVC	ML	12,00	1,37	16,44	
MT1014	Collarín 102x7,7x376 NEGRO 2273-0	UD	750,00	0,14	105,00	
MT1015	Collarín 50x7,7x213 NEGRO 2271-0	UD	800,00	0,06	48,00	
MT1016	Brida uso Ext.PA 6.6W 45x4,8x188 NEGRO	UD	350,00	0,02	7,00	
MT1017	Envase-Soporte Autorroscable EM-7	UD	250,00	0,13	32,50	
MT1018	Cartucho 6,8/11M- ROJO	UD	150,00	0,22	33,00	
MT1019	Clavo X-EM6H-11-9 FP8 P/ACERO X-EM6	UD	150,00	0,34	51,00	
MT1020	Taco SX 6 d.6	UD	150,00	0,03	4,50	
MT1021	Tornillo ABC SPAX 4X30	UD	150,00	0,02	3,00	
MT1022	Cinta TEMFLEX 1300 19mmx25m PVC NG.0,13mm	UD	7,00	0,64	4,48	
MT1023	Rollo Cinta DYMO REF.18764/18483/16959 12mmx7m	UD	14,00	4,04	56,56	
Subtotal:					6034,43	
Maquinaria:						1.480,00
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MQ1001	Plataforma Elevadora HAULOTTE Gasoil	D	10,00	148,00	1480,00	
Subtotal:					1480,00	
Costes Directos:					13.939,43	
Gastos Indirectos (2,5%):					348,48	
Coste Ejecución:					14.287,91	

5.3.2 *Instalación Switches*

C.02		Instalación Switches y Pruebas				16.568,55
Mano de obra:						2.252,00
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MO1001	Oficial 1ª Electricista	H	20,00	22,60	452,00	
MO1003	Técnico Programador	H	40,00	45,00	1.800,00	
Subtotal:					2.252,00	
Materiales:						13.912,44
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MT1024	Cable Flexible Libre Halógenos 1,5 mm2 EXZHELLENT CPR 750V General Cable	ML	20,00	0,20	4,00	
MT1025	Automático Magneto-térmico iC60N 10A 2P SCHNEIDER	UD	14,00	12,56	175,84	
MT3001	6GK5992-1AL00-8AA0. Accesorios para SCALANCE X; Transceptor enchufable SFP992-1; 1 puerto LC a 1000 Mbits/s, óptico; multimodo, vidrio, hasta máx. 750 m	UD	28,00	157,34	4.405,52	
MT3002	6GK5206-2BS00-2AC2. SCALANCE XC206-2SFP Layer 2 IE Switch gestionable; 6 puertos RJ45 10/100 Mbits/s; 2 puertos SFP 100/1000 Mbits/s; 1 puerto de consola; LED de diagnóstico; Alimentación redundante; rango de temperatura de -40 °C a +70 °C; montaje: perfil DIN/soporte S7/pared Funciones de redundancia Office características (RSTP, VLAN, ...); dispositivo PROFINET IO; conforme con Ethernet/IP; caja de conector C	UD	14,00	666,22	9.327,08	
Subtotal:					13.912,44	
					Costes Directos:	16.164,44
					Gastos Indirectos (2,5%):	404,11
					Coste Ejecución:	16.568,55

5.3.3 Instalación PLC

C.03		Instalación PLC y Pruebas				6.520,98
Mano de obra:						855,80
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MO1001	Oficial 1ª Electricista	H	8,00	22,60	180,80	
MO1003	Técnico Programador	H	15,00	45,00	675,00	
Subtotal:					855,80	
Materiales:						5.506,14
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MT1024	Cable Flexible Libre Halógenos 1,5 mm2 EXZHELLENT CPR 750V General Cable	ML	2,00	0,20	4,00	
MT1025	Automático Magneto-térmico iC60N 10A 2P SCHNEIDER	UD	1,00	12,56	12,56	
MT2001	6ES7390-1AE80-0AA0. SIMATIC S7-300, Perfil soporte longitud =480mm	UD	1,00	26,71	26,71	
MT2002	6ES7307-1EA01-0AA0. SIMATIC S7-300, PS 307 de 5 A, fuente de alimentación estabilizada, entrada: AC 120/230 V, salida: DC 24 V/5 A.	UD	1,00	117,68	117,68	
MT2003	6ES7317-2EK14-0AB0. SIMATIC CPU 317-2 PN/DP, Módulo central con 1 MByte. Memoria principal, Interfaz 1: MPI/DP 12 Mbits/s, Interfaz 2: Ethernet Profinet, con Switch de 2 Puertos.	UD	1,00	3.758,16	3.758,16	
MT2004	6ES7953-8LM32-0AA0. SIMATIC S7, Micro Memory Card para S7-300/C7/ET. 200, 3, 3 V Nflash, 4 Mbytes.	UD	1,00	303,18	303,18	
MT2005	6GK7343-1EX30-0XE0. SIMATIC NET CP343-1, procesador de comunicaciones para la conexión de SIMATIC S7-300 a Ethernet Industrial vía ISO y TCP/IP, controlador PN IO o dispositivo PROFINET-IO, switch de 2 puertos ERTEC200 integrado, comunicación S7, FETCH/WRITE, SEND/RCV con y sin RFC1006, multicast DHCP, sincronización de CPU vía NTC, diagnóstico inicialización vía LAN, 2 x conexiones RJ45 para LAN a 10/100 Mbit/s.	UD	1,00	1.120,57	1.120,57	
Subtotal:					5.506,14	
					Costes Directos:	6.361,94
					Gastos Indirectos (2,5%):	159,05
					Coste Ejecución:	6.520,98

5.3.4 Instalación Servidor OS

C.04		Instalación Servidor OS y Pruebas				15.179,99
Mano de obra:						1.440,00
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
MO1003	Técnico Programador	H	32,00	45,00	1.440,00	
Subtotal:					1.440,00	
Materiales:						13.369,75
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
LS1001	6ES7658-2BA58-0YA0. SIMATIC PCS7, Software, OS software Server v9.0 (PO 100) Lic. Indiv. p. 1 instalación SW RT, clave de lic. en lapiz usb, clase a, 5 idiomas (al, in, fr, it, es), ejecutable bajo win 7 ultimate win 10 enterprise ltsb para más informaciones ver PCS7 v9.0	UD	1,00	9.916,19	9.916,19	
LS1002	6ES7658-2XB00-0XB0. SIMATIC PCS7, software RT license OS (PO 1000) licencia Indiv. p. 1 instalación SW RT, sin sw, sin docum. clave licencia en stick usb.	UD	1,00	2.643,87	2.643,87	
LS1003	6ES7650-1CD58-2YB5. SIMATIC PCS7, Software, BCE V9.0 lic. flotante P. 1 usuario SW RT, sin SW, sin docum. clave de lic. en lapiz usb, clase a, 3 idiomas (al, in, fr), ejecutable bajo win 7 ultimate win 10 enterprise ltsb server 2012 R2, Server 2016 Standard.	UD	1,00	809,69	809,69	
Subtotal:					13.369,75	
Costes Directos:						14.809,75
Gastos Indirectos (2,5%):						370,24
Coste Ejecución:						15.179,99

5.3.5 Estudio y Desarrollo del Programa

C.05		Estudio y Desarrollo del Programa				29.520,00
Mano de obra:						28.800,00
Código	Concepto	Ud.	Cantidad	Precio	Parcial	
SW1001	Estudio y planificación	H	80,00	45,00	3.600,00	
SW1002	Desarrollo de librería PLC (FBs) y pruebas	H	280,00	45,00	12.600,00	
SW1003	Desarrollo de librería Gráfica (Block Icons y Faceplates) y pruebas	H	160,00	45,00	7.200,00	
SW1004	Programación PLC	H	80,00	45,00	3.600,00	
SW1005	Comisionado y puesta en marcha	H	40,00	45,00	1.800,00	
Subtotal:					28.800,00	
Costes Directos:					28.800,00	
Gastos Indirectos (2,5%):					720,00	
Coste Ejecución:					29.520,00	

5.4 Resumen del Presupuesto

Capítulo	Resumen	Importe (€)
C.01	Instalación Infraestructura de red	14.287,91
C.02	Instalación Switches y Pruebas	16.568,55
C.03	Instalación PLC y Pruebas	6.520,98
C.04	Instalación Servidor OS y Pruebas	15.179,99
C.05	Estudio y Desarrollo del Programa	29.520,00
Presupuesto de ejecución del material:		82.077,43
Gastos generales (13%):		10.670,06
Beneficio industrial (6%):		4.924,64
Presupuesto de ejecución por contrata:		97.672,13
I.V.A. (21%):		20.511,14
Presupuesto Total:		118.183,27

El presupuesto total del proyecto asciende a la cantidad de **CIENTO DIECIOCHO MIL CIENTO OCHENTA Y TRES EUROS CON VEINTISIETE CÉNTIMOS DE EURO.**

Benicarló, junio de 2021



Alfonso Arranz García