

MANUAL

Funcionamiento comunicaciones MODBUS
con centrales LYON y CLVR

MODBUS communication Operation with
LYON and CLVR Control Panels

INDICE GENERAL / GENERAL INDEX

> CASTELLANO: Idioma / language

	<u>Pág</u>
1. Configuración MODBUS	3
2. Implementación MODBUS.....	4
2.1. Protocolo MODBUS (Modo RTU)	4
2.2. Solicitud	4
2.3. Respuesta	4
2.4. Modo RTU	5
2.5. Conexionado al sistema digital	5
2.6. Código de función 03. READ HOLDING REGISTERS	6
2.7. Código de función 04. READ INPUT REGISTERS	6
2.8. Código de función 06. PRESET SINGLE REGISTER	7
2.9. Códigos de excepción o error	8
2.10. Mapa de direcciones de la central LYON	9
2.11. Mapa de direcciones de la central CLVR	11

> ENGLISH: Idioma / language

1. MODBUS Configuration	13
2. MODBUS Implementation	14
2.1. Modbus protocol (RTU mode)	14
2.2. Application	14
2.3. Response	14
2.4. RTU Mode	15
2.5. Connection to digital system	15
2.6. Function Code 03. READ HOLDING REGISTERS	16
2.7. Function Code 04. READ INPUT REGISTERS	16
2.8. Function Code 06. PRESET SINGLE REGISTER	17
2.9. Exception or error codes	18
2.10. Address Map for LYON Control Panel	19
2.11. Address Map for CLVR Control Panel	21

OCTUBRE / OCTOBER 2014

1 Configuración MODBUS

1.1 En centrales analógicas LYON

Las centrales digitales de detección de incendios COFEM con protocolo MODBUS se comunican con el exterior mediante un bus estándar RS485. Esta conexión viene indicada en el interior de la central como "CONPC" en la tarjeta de Back Panel. El indicativo muestra los tres bornes, GND, - y +. Según diversas nomenclaturas de RS485, el negativo "-" se le puede denominar como "B" y el positivo "+" como "A". GND no requiere ser conectado, pero sirve para crear apantallamiento.

Para que una central digital pueda comunicar con un master se debe activar la comunicación y seleccionar el número de esclavo.

Para activar la comunicación siga los siguientes pasos:

- 1.- Pulsar la tecla MENÚ.
- 2.- Introducir el código de acceso a teclado. (Nº 27)
- 3.- Ir a la opción CONFIGURACIÓN SISTEMA (4)
- 4.- Introducir el código de acceso a sistema. (Nº 9000)
- 5.- Ir a la opción COMUNICACIONES (9)
- 6.- Ir a la opción ACTIVAR (1)

El número de esclavo MODBUS define la identificación propia de la central dentro de la red MODBUS. Por defecto este valor es 1 pero se puede variar en el submenú de COMUNICACIONES del menú SISTEMA de la central.

Los pasos a seguir para acceder a este menú son:

- 1.- Pulsar la tecla MENÚ.
- 2.- Introducir el código de acceso a teclado (Nº 27)
- 3.- Ir a la opción CONFIGURACIÓN SISTEMA (4)
- 4.- Introducir el código de acceso a sistema (Nº 9000)
- 5.- Ir a la opción COMUNICACIONES (9)
- 6.- Ir a la opción PROGRAMAR NÚMERO CENTRAL (5)
- 7.- Introducir el número de esclavo MODBUS que tendrá esta Central. Cada central conectada a la misma red MODBUS deberá tener un número de Central diferente.

Los parámetros de comunicación del puerto son:

VELOCIDAD: 9600 baudios.
BITS DE DATOS : 8 bits.
BITS DE PARADA : 2 bit.
PARIDAD : Sin paridad.

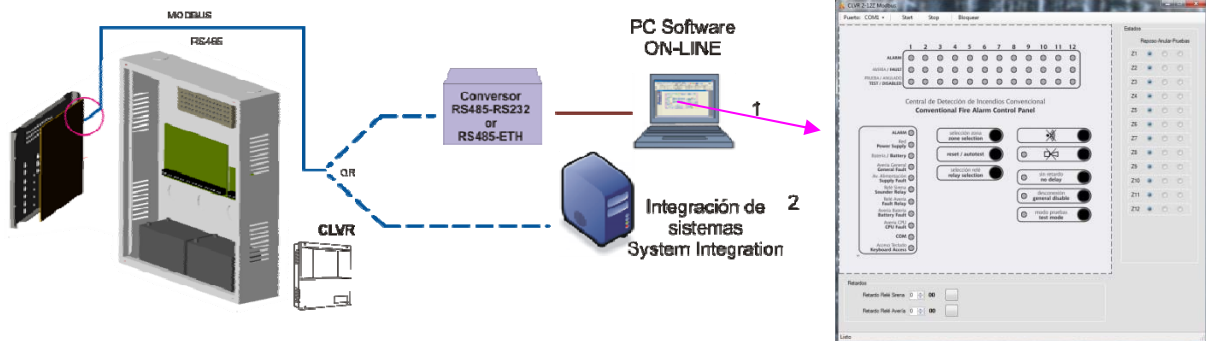
Para comprobar la correcta configuración, enviar, por ejemplo, una orden de reset a la central: enviar un valor 01h con una orden 06h a la posición 00A0h de la central.



1.2 En centrales convencionales CLVR

La central CLVR con MODBUS incorpora protocolo de comunicaciones MODBUS mediante línea estándar RS485.

Dicha funcionalidad, permite a la central comunicarse con equipos exteriores con el fin de permitir la integración del sistema de detección y alarma de incendios con otros sistemas del establecimiento o edificio (ver ejemplo en la figura opción 2).



Comunicación MODBUS

2 Implementación MODBUS

2.1 Protocolo MODBUS (Modo RTU)

El protocolo MODBUS aplica el principio *Maestro-Esclavo*, es decir, en la red tenemos un controlador que actúa como Maestro y es quien *solicita* e inicia siempre las comunicaciones y los periféricos que actúan como esclavos, no tomando nunca la iniciativa y *respondiendo* a las solicitudes del Maestro (siempre que sea posible).

2.2 Solicitud

El código de función en el mensaje de solicitud comunica al periférico direccionado el tipo de acción que debe realizar. Los Bytes de datos contienen información adicional que el esclavo necesita para llevar a cabo la acción solicitada. Por ejemplo, el código de función 03 solicitará al esclavo la lectura de registros y la respuesta por parte de éste de estos valores. Los campos de datos contendrán información acerca del primer registro a leer y del número de registros. Finalmente el Check-Error provee un método para que el esclavo pueda comprobar la integridad del mensaje recibido.

2.3 Respuesta

Si el esclavo genera una respuesta normal, el código de función en la respuesta es el mismo que en la solicitud. Los bytes de datos contendrán la información solicitada al esclavo. Si ocurre un error o la acción solicitada al esclavo no es posible, éste enviará un código de función modificado, para indicar que la respuesta es una respuesta de error y los bytes de datos contendrán información acerca del tipo de error producido. Finalmente el Check-Error permitirá comprobar al Master la integridad del mensaje recibido como respuesta del esclavo.

2.4 Modo RTU

Se utiliza el modo RTU (Remote Terminal Unit). Cada 8-bit byte en un mensaje contiene dos 4-bits caracteres hexadecimales. La ventaja de este sistema, con respecto al modo ASCII, es que permite una mayor densidad de transmisión de información para una misma velocidad. En este modo el mensaje debe enviarse en un flujo continuo.

El formato por cada byte en modo RTU es:

<i>Código:</i>	8-bit binary, hexadecimal 2 caracteres hexadecimales contenidos en cada campo de 8-bit del mensaje.
<i>Bits por Byte:</i>	1 bit de start 8 bits de datos, el bit menos significativo primero 1 bit para paridad par/impar; sin bit para no paridad 1 bit de stop si se usa paridad; 2 bits de stop si no se usa paridad
<i>Campo CHECK-ERROR:</i>	Tipo CRC (Cyclical Redundancy Check)

2.5 Conexión al sistema digital

El campo de CRC esta formado por dos bytes, conteniendo un valor binario de 16-bit. El CRC es calculado por el equipo que va a transmitir y lo incluye al final del mensaje. El equipo que recibe el mensaje separa la parte final del mensaje, correspondiente al CRC, y calcula el CRC de lo que le ha quedado de mensaje comparando luego el resultado con el CRC que había separado del mensaje recibido. Si ambos son iguales procesa el mensaje sino lo desecha.

A continuación se explica una forma o procedimiento para el cálculo del CRC.

- Cargar un registro de 16-bit con FFFF_{HEX} (todo 1's). Llamar a este registro CRC.
- Hacer una OR Exclusiva con los primeros 8-bit del mensaje con el byte de menor peso del registro CRC de 16-bit, poner el resultado en el registro CRC.
- Desplazar el registro CRC un bit a la derecha (hacia el bit menos significativo LSB), rellenar con un cero la posición de la izquierda (bit más significativo MSB), extraer el bit menos significativo (LSB) y examinarlo.

Ejemplo: Partimos de un CRC = 10011011, hacemos el desplazamiento y tenemos _10011011, rellenamos por la izquierda con cero y tenemos 010011011, finalmente extraemos el bit de menor peso y nos queda CRC= 01001101 y LSB=1

- Si el LSB extraído en el paso anterior era 0, repetir el paso 3.
Si el LSB extraído en el paso anterior era 1, hacer una OR Exclusiva del registro CRC con el valor polinomial A001_{HEX} (1010 0000 0000 0001).
- Repetir Pasos 3 y 4 hasta que se hallan completado 8 desplazamientos. Cuando se halla conseguido esto un 8-bit byte habrá sido completamente procesado.
- Repetir Pasos 2 hasta el 5 para el siguiente 8-bit byte del mensaje. Continúe haciendo esto hasta que todos los bytes hallan sido procesados.
- Finalmente en el registro CRC tendremos el valor de CRC.

2.6 Código de función 03. READ HOLDING REGISTERS

Leer el contenido binario de los registros de almacenamiento del esclavo.

Solicitud. El mensaje de solicitud especifica, la dirección del periférico que queremos leer, el código de la función (03 en este caso), la dirección del registro inicial, la cantidad de registros a leer y finalmente el CRC calculado. Los registros son direccionados a partir del cero, es decir, los registros 1-16 son direccionados como 0-15.

Ejemplo: Solicitar al periférico 64 los contenidos de los registros de las direcciones 0001, 0002 y 0003 (En estos registros los valores, almacenados como ejemplo, son 1964, 8520 y 112 respectivamente).

Formato de Solicitud:

La dirección del periférico en Hexadecimal es: 64_{DEC} = 40_{HEX}
 La dirección del registro inicial en Hexadecimal es: 01_{DEC} = 00 01_{HEX}
 El número de registros en Hexadecimal es: 03_{DEC} = 00 03_{HEX}
 El CRC calculado es: 1A 5B_{HEX}

Dirección de Periférico	Código de Función	Dirección Registro Inicial HI	Dirección Registro Inicial LO	Número de Registros HI	Número de Registros LO	Check-Error LO	Check-Error HI
40	03	00	01	00	03	5B	1A

Respuesta. El mensaje de respuesta especifica, la dirección del periférico que responde, el código de la función (03 en este caso), el número de bytes de datos respondidos (que serán el doble que le número de registros solicitados), los bytes de respuesta y finalmente el CRC calculado. Los registros de datos, en la respuesta, están empaquetados en 2 bytes por registro, con el contenido binario justificado a la derecha para cada byte. Para cada registro, el primer byte contiene los bits de mayor orden y el segundo los de menor.

Formato de Respuesta:

La dirección del periférico en Hexadecimal es: 64_{DEC} = 40_{HEX}
 El numero de bytes respondidos en Hexadecimal es: 06_{DEC} = 06_{HEX}
 El registro1 en Hexadecimal es: 1964_{DEC} = 07 AC_{HEX}
 El registro2 en Hexadecimal es: 8520_{DEC} = 21 48_{HEX}
 El registro3 en Hexadecimal es: 0112_{DEC} = 00 70_{HEX}
 El CRC calculado es: 07 47_{HEX}

Dirección de Periférico	Código de Función	Número de Bytes de Respuesta	Registro1 Byte HI	Registro1 Byte LO	Registro2 Byte HI	Registro2 Byte LO	Registro3 Byte HI	Registro3 Byte LO	Check-Error LO	Check-Error HI
40	03	06	07	AC	21	48	00	70	47	07

2.7 Código de función 04. READ INPUT REGISTERS

Leer el contenido binario de los registros de entrada del esclavo.

Solicitud. El mensaje de solicitud especifica, la dirección del periférico que queremos leer, el código de la función (04 en este caso), la dirección del registro inicial, la cantidad de registros a leer y finalmente el CRC calculado. Los registros son direccionados a partir del cero, es decir, los registros 1-16 son direccionados como 0-15.

Ejemplo: Solicitar al periférico 225 el contenido del registro de la dirección 0001.(En este registro el valor, almacenado como ejemplo, es 5555).

Formato de Solicitud:

La dirección del periférico en Hexadecimal es: 225_{DEC} = E1_{HEX}
 La dirección del registro inicial en Hexadecimal es: 01_{DEC} = 00 01_{HEX}
 El número de registros en Hexadecimal es: 01_{DEC} = 00 01_{HEX}
 El CRC calculado es: 6A 76_{HEX}

Dirección de Periférico	Código de Función	Dirección Registro Inicial HI	Dirección Registro Inicial LO	Número de Registros HI	Número de Registros LO	Check-Error LO	Check-Error HI
E1	04	00	01	00	01	76	6A

Respuesta. El mensaje de respuesta especifica, la dirección del periférico que responde, el código de la función (04 en este caso), el número de bytes de datos respondidos (que serán el doble que le número de registros solicitados), los bytes de respuesta y finalmente el CRC calculado. Los registros de datos, en la respuesta, están empaquetados en 2 bytes por registro, con el contenido binario justificado a la derecha para cada byte. Para cada registro, el primer byte contiene los bits de mayor orden y el segundo los de menor.

Formato de Respuesta:

La dirección del periférico en Hexadecimal es: 225_{DEC} = E1_{HEX}
 El numero de bytes respondidos en Hexadecimal es: 02_{DEC} = 02_{HEX}
 El registro1 en Hexadecimal es: 5555_{DEC} = 15 B3_{HEX}
 El CRC calculado es: 77 C3_{HEX}

Dirección de Periférico	Código de Función	Número de Bytes de Respuesta	Registro1 Byte HI	Registro1 Byte LO	Check-Error LO	Check-Error HI
E1	04	02	15	B3	C3	77

2.8 Código de función 06. PRESET SINGLE REGISTER

Predefine el contenido binario de un registro de almacenamiento del esclavo.

Solicitud. El mensaje de solicitud especifica, la dirección del periférico que modificar, el código de la función (06 en este caso), la dirección del registro a modificar, el valor a predefinir y finalmente el CRC calculado. Los registros son direccionados a partir del cero, es decir, el registro 1 es direccionado como 0.

Ejemplo: Solicitar al periférico 23 que predefina el registro de la dirección 14 con el valor 1234.

Formato de Solicitud:

La dirección del periférico en Hexadecimal es: 23_{DEC} = 17_{HEX}
 La dirección del registro a predefinir en Hexadecimal es: 14_{DEC} = 00 0E_{HEX}
 El valor de predefinir en Hexadecimal es: 1234_{DEC} = 04 D2_{HEX}
 El CRC calculado es: 68 62_{HEX}

Dirección de Periférico	Código de Función	Dirección Registro HI	Dirección Registro LO	Valor a Predefinir HI	Valor a Predefinir LO	Check-Error LO	Check-Error HI
17	06	00	0E	04	D2	62	68

Respuesta. El mensaje de respuesta es un eco de la solicitud, es decir, especifica la dirección del periférico que responde, el código de la función (06 en este caso), la dirección del registro a modificado, el valor a predefinido y finalmente el CRC calculado.

Formato de Respuesta:

La dirección del periférico en Hexadecimal es:

23_{DEC} = 17_{HEX}

La dirección del registro a Predefinir en Hexadecimal es:

14_{DEC} = 00 0E_{HEX}

El valor de Predefinir en Hexadecimal es:

1234_{DEC} = 04 D2_{HEX}

El CRC calculado es:

68 62_{HEX}

Dirección de Periférico	Código de Función	Dirección Registro HI	Dirección Registro LO	Valor a Predefinir HI	Valor a Predefinir LO	Check-Error LO	Check-Error HI
17	06	00	0E	04	D2	62	68

2.9 Códigos de excepción o error

Cuando el Master envía una solicitud a un esclavo puede ocurrir:

- Si el esclavo recibe la solicitud sin errores de comunicación, y puede procesar dicha instrucción, este devuelve una respuesta normal.
- Si el esclavo no recibe la solicitud, debido a un error de comunicación, no responde. El programa del Master procesará una condición de TimeOut (Tiempo de espera agotado) y dará por mala la solicitud.
- Si el esclavo recibe la solicitud pero detecta un error de comunicación (fallo en la comprobación del CRC), no responde. El programa del Master procesará una condición de TimeOut (Tiempo de espera agotado) y dará por mala la solicitud.
- Si el esclavo recibe la solicitud sin error de comunicación, pero no puede llevarla a cabo (por ejemplo, la solicitud es leer un registro que no está disponible en este periférico), el esclavo devuelve una **Respuesta de Excepción** informando al Master de la naturaleza del error.

Las respuestas de excepción tienen dos campos que se diferencian de una respuesta normal.

Campo de Código de Función. En una respuesta normal, el esclavo repite el código de función de la solicitud original en el campo de código de función de la respuesta poniendo a 1 el MSB. Con el código de función de esta manera el Master sabe que se ha producido un error de excepción y puede examinar el campo de dato para saber que tipo de error se ha producido.

Campo de Dato. En una respuesta normal, el esclavo devolverá datos en el campo de datos (la información que le requirió el Master). En una respuesta de excepción, el esclavo devuelve un código de excepción en el campo de datos. Esta define la causa que ha dado el error en el esclavo.

Los códigos de excepción están estandarizados y son:

CODIGO	NOMBRE	SIGNIFICADO
01	Función ilegal	La función recibida no la puede llevar a cabo el Esclavo.
02	Dirección de Dato ilegal	La dirección del dato solicitado no es una dirección disponible en el Esclavo
03	Valor de Dato ilegal	El valor contenido en el campo de dato no es un valor admitido por el esclavo
04	Fallo en el Esclavo	Un error irrecuperable se ha producido mientras el esclavo estaba haciendo la acción solicitada
05	Reconocimiento	El esclavo ha recibido la solicitud y la esta procesando, pero requiere mucho tiempo de procesado. Esta respuesta es dada para evitar un error de TimeOut en el Master.
06	Esclavo ocupado	Esclavo ocupado realizando algún proceso. El Master debe retransmitir este mensaje más tarde.
07	No Reconocimiento	El esclavo no puede realizar la función solicitada por el Master.
08	Error de Paridad	El esclavo intenta leer memoria extendida, pero detecta un error de paridad.

2.10 Mapa de direcciones de la central LYON

MSB	LSB	NºBYTES	Área	Valor	Descripción
Dirección		5632			
00h	00h	1	Estado central	XXh	Estado prioritario de la central
00h	10h	1	Estado Periféricos	00h	Normal
				21h	Avería repetidor no comunica
				22h	Avería central no comunica
00h	11h+NºPerif	35+15+1	Periférico X (Central:35;Repetidor:15)	00h	Normal
				21h	Avería repetidor no comunica
				22h	Avería central no comunica
00h	50h	1	Estado Controladoras de Bucle	00h	Normal
				23h	Avería controladora no comunica
00h	51h+NºCTRL	5	Controladora de Bucle X (De 1 a 5 y 11)	00h	Normal
				23h	Avería controladora no comunica
00h	60h	1	Estado Fuente Alimentación	00h	Normal
				31h	Avería cargador
				32h	Avería sin batería.
				33h	Avería batería descargada.
				34h	Avería batería sobrecargada.
				35h	Avería red baja.
				36h	Avería red alta.
				37h	Avería sin red.
				38h	Avería 30V
00h	61h	1	Estado Batería	00h	Normal
				32h	Avería sin batería.
				33h	Avería batería descargada.
				34h	Avería batería sobrecargada.
00h	70h	1	Estado Relés Generales	00h	Normal
				41h	Avería línea Abierta.
				42h	Avería línea Cruzada.
				91h	Disparo.
00h	71h	1	Relé General 1	00h	Normal
				41h	Avería línea Abierta.
				42h	Avería línea Cruzada.
				91h	Disparo.
00h	72h	1	Relé General 2	00h	Normal
				41h	Avería línea Abierta.
				42h	Avería línea Cruzada.
				91h	Disparo.
15h	00h	1	Estado Relés de Bucle	00h	Normal
				91h	Disparo.
15h	01h+Nº Relé	200	Relé Lógico X (Max:200)	00h	Normal
				91h	Disparo.
00h	80h+Nº Bucle	20	Bucle X (Del bucle 1 al bucle 20)	00h	Normal
				11h	Punto en pruebas
				12h	Punto Anulado
				81h	Avería suciedad
				82h	Avería bucle abierto
				83h	Avería consumo
				80h	Avería Sensor
				A0h	Alarma sensor
				XXh	Incidencias de puntos en bucle o de los propios bucles.
00h	A0h	1	Reset	01h	Activado
00h	A1h	1	Tecla Sin Retardo	01h	Activado

00h	A2h	1	Tecla Paro Sirenas	01h	Activado
00h	A3h	1	Tecla Evacuación	01h	Activado
00h	A4h	1	Tecla Modo Manual	01h	Activado
01h+Nº Bucle	01h+Nº Punto	20*255=5100	Punto X de Bucle Y (Max: Y= 20 X=255)	00h	Normal
				11h	Punto en pruebas
				12h	Punto Anulado
				84h	Avería punto no contesta
				85h	Avería Kmay sucio
				86h	Avería kmay línea cruzada
				87h	Avería Kmay línea abierta
				88h	Avería Mstay línea cruzada
				89h	Avería Mstay línea abierta
				8Ah	Avería técnica
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				8Dh	Avería relé no activo
				8Eh	Avería 24V
				A1h	Alarma intrusión
				A2h	Alarma LPG
				A3h	Alarma CO
				A4h	Alarma técnica
				A5h	Alarma detector esclavo
				A6h	Alarma temperatura
				A7h	Alarma humo
				A8h	Alarma pulsador esclavo
				A9h	Alarma pulsador

2.11 Mapa de direcciones de la central CLVR

MSB	LSB	NºBYTES	Área	Valor	Descripción
Dirección		275			
00h	00h	1	Estado central	XXh	Estado prioritario de la central
00h	01h	1	Retardo Rele Sirena General	XX	Valor del retardo en segundos
00h	02h	1	Retardo Rele Avería General	XX	Valor del retardo en segundos
00h	03h	1	Retardo Rele 2	XX	Valor del retardo en segundos
00h	04h	1	Retardo Rearme Extinción	XX	Valor del retardo en segundos
00h	60h	1	Estado Fuente Alimentación	00h	Normal
				31h	Avería cargador
				33h	Avería batería descargada.
				37h	Avería sin red.
				38h	Avería 30V
00h	61h	1	Estado Batería	00h	Normal
				33h	Avería batería descargada.
00h	70h	1	Estado Relés Generales	00h	Normal
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				62h	Disparo.
00h	71h	1	Relé Sirena General 1	00h	Normal
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				62h	Disparo.
00h	72h	1	Relé Sirena General 2	00h	Normal
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				91h	Disparo.
00h	73h	1	Relé Avería General	00h	Normal
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				91h	Disparo.
00h	A0h	1	Tecla Reset	01h	Activado
00h	A1h	1	Tecla Sin Retardo	01h	Activado
00h	A2h	1	Tecla Paro Sirena	01h	Activado
00h	A3h	1	Activar Extinción	01h	Activado
00h	A4h	1	Aborto Extinción	01h	Activado
00h	A5h	1	Tecla Zumbador	01h	Activado
01h	01h*Nº Zona	12	Zona X (Max: X=12)	00h	Normal
				11h	Punto en pruebas
				12h	Punto Anulado
				13h	Alarma Detector Prueba
				14h	Alarma Pulsador Prueba
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				A7h	Alarma humo
				A9h	Alarma pulsador
01h	10h	1	Estado Relés	00h	Normal
				12h	Anulado Relé
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				8Eh	Avería 24V
				90h	Prealarma
				61h	Disparo.
				92h	Aborto

01h	11h*Nº Relé	2	Relé X (Max:2)	00h	Normal
				12h	Anulado relé
				8Bh	Avería línea Abierta.
				8Ch	Avería línea Cruzada.
				8Eh	Avería 24V
				90h	Prealarma
				91h	Disparo.
				92h	Aborto

NOTA:

COFEM S.A. se reserva el derecho a realizar cambios debido a errores tipográficos, impresiones de la información actual o mejoras de programas y/o equipo en cualquier momento y sin previo aviso.

1 MODBUS Configuration

1.1 LYON Control Panel

COFEM digital fire detection control panels with MODBUS, communicates with external through a standard RS485 bus. This connection is indicated inside of the control panel as "CONPC" in the Back Panel card. The label shows the three terminals, GND, - and +. According to different nomenclatures of RS485, the negative "-" can be referred to as "B" and the positive "+" as "A". GND need not be connected, but it is used to develop the screening.

In order to communicate the control panel with a master, it must be activated the communication and select the number of slave.

To enable communication follow these steps:

1. - Press the MENU key.
2. - Enter the access code keypad. (No. 27)
3. - Go to the SYSTEM CONFIGURATION option (4)
4. - Enter the access code system. (No. 9000)
5. - Go to the option COMMUNICATIONS (9)
6. - Go to the Enable (1)

The MODBUS slave number, defines the own control panel identification, within the MODBUS network. By default this value is 1 but may be varied in the submenu COMMUNICATIONS inside SYSTEM menu of the Control panel.

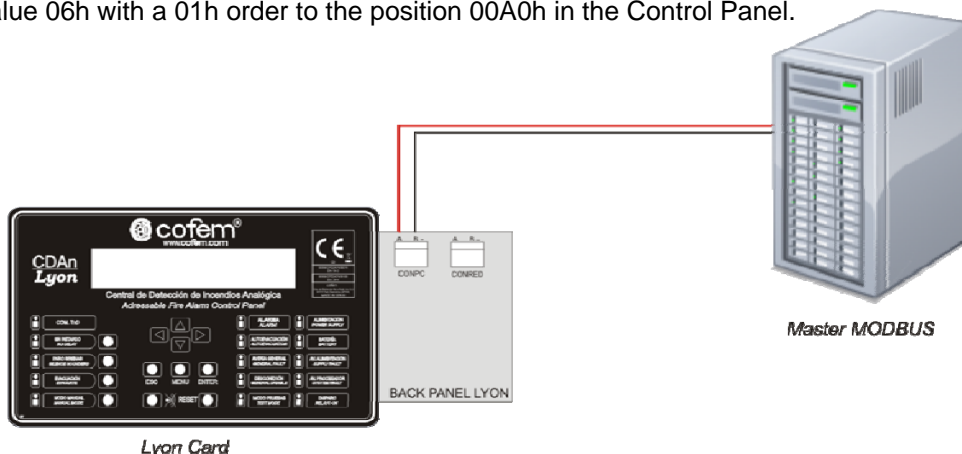
The steps to access this menu are:

1. - Press the MENU key.
2. - Enter the access code keypad (No. 27)
3. - Go to the SYSTEM SETUP option (4)
4. - Enter the access code system (No. 9000)
5. - Go to the option COMMUNICATIONS (9)
6. - Go to the SET NUMBER CONTROL PANEL (5)
7. - Enter the number of MODBUS slave for this Control Panel. Each Control Panel connected to the same network MODBUS must have a different Control Panel number.

The port communication parameters are:

SPEED: 9600 baud.
Data bits: 8 bits.
Stop bit: 2 bit.
Parity: None.

To verify the correct configuration, send, for example, a reset command to the control unit: send a value 06h with a 01h order to the position 00A0h in the Control Panel.



1.2 En centrales convencionales CLVR

CLVR Control panel incorporates MODBUS protocol communication under RS485 standard line.

This functionality allows control panel communication with external equipments for fire detection and fire alarm integration into other building systems (example in figure option 2).

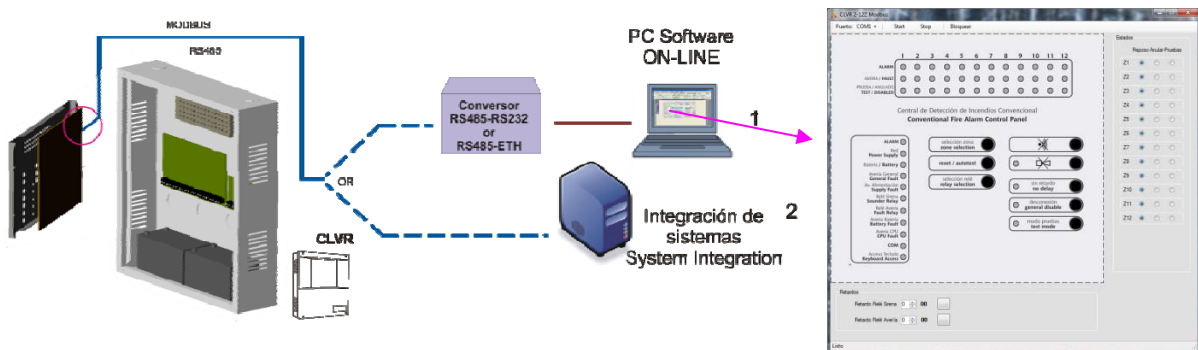


Figura 13 MODBUS Communication

2 MODBUS Implementation

2.1 MODBUS Protocol (RTU Mode)

The MODBUS protocol applies the master-slave principle, that is, we have a network Master controller that acts as It invites and initiates communications and always peripherals that act as slaves, never taking the initiative and responding to Master's requests (where possible).

2.2 Request

The function code in the request message, informs the addressed peripheral type action to do. The data bytes contain additional information that the slave needs to perform the requested action. For example, the request function code 03 reading the slave records and the response by the latter of these values. Fields data contain information about the first record to be read and the number of records. Finally the Check-Error provides a method for the slave to verify the integrity of the received message.

2.3 Response

If the slave generates a normal response, the function code in the response is the same in the application. The data bytes contain the requested information to the slave. If it occurs an error or the slave requested action is not possible, it will send a feature code modified to indicate that the response is an error response and data bytes contain information about the type of error that occurred. Finally Check-Error allows the Master to check the integrity of the message received in response from the slave.

2.4 RTU Mode

Mode is used RTU (Remote Terminal Unit). Each 8-bit byte in a message contains two 4 - bit hexadecimal characters. The advantage of this system with respect to the ASCII mode, is that enables greater information transmission for the same speed. In this mode, the message is sent in a continuous stream.

The format for each byte in RTU mode is:

<i>Code:</i>	8-bit binary, hexadecimal 2 hexadecimal character contained in each field 8-bit message
<i>Bits per Byte:</i>	1 start bit 8 data bits, least significant bit first 1 bit for even / odd parity, no parity bit for not 1 stop bit if parity is used; 2 stop bits if no uses parity
<i>CHECK-ERROR field:</i>	Type CRC (Cyclical Redundancy Check)

2.5 Digital System connection

The CRC field consists of two bytes, containing a binary value of 16-bit. The CRC is calculated by the transmitting equipment and includes it the end of the message. The equipment that receives the message separates the end part of the message, for the CRC, and calculates the CRC what message has been then comparing the result with the CRC had separate the received message. If both are the same processes the message but discards it.

Above it is described one procedure to calculate the CRC.

- Load a 16-bit with FFFF-HEX (all 1's). Call this register: CRC.
- Make an exclusive OR on the first 8-bit byte of the message with the lowest Weight of the record of 16-bit CRC, put the result in the CRC register.
- Displace the CRC register one bit to the right (toward the least significant bit LSB) fill with a zero the left position (MSB), bit removal least significant bit (LSB) and examine it.

Example : We start with a CRC = 10011011, displacement and we have _10011011, Fill the left with zero and 010,011,011 have finally extract the least significant bit and we have CRC = 01001101 and LSB = 1

- If the LSB extracted in the previous step was 0, repeat step 3.
If the LSB extracted in the previous step was 1, make a Exclusive OR the CRC register with the polynomial value A001HEX (1010 0000 0000 0001).
- Repeat Steps 3 and 4 until eight shifts are completed. When achieved this is an 8-bit byte will have been completely processed.
- Repeat Steps 2 to 5 for the next 8-bit byte of the message. Continue this until all bytes are been processed.
- Finally on record have CRC CRC value.

2.6 Function Code 03. READ HOLDING REGISTERS

Reading the binary content of the slave storage registers.

Request. The request message specifies the address of the peripheral you want to read, the function code (03 in this case), the direction of the initial registration, the number of records to read and finally the calculated CRC. The registers are addressed from zero, ie registers 1-16 are addressed as 0-15.

Example: To request the contents of 64 peripheral registers addresses 0001, 0002 and 0003 (values in these records, stored as an example, are 1964, 8520 and 112 respectively).

Request Form:

The peripheral address in hexadecimal is: 64_{DEC} = 40_{HEX}
 The initial register address in hexadecimal is 01_{DEC} = 00 01_{HEX}
 The number of records in hexadecimal is: 03_{DEC} = 00 03_{HEX}
 The CRC is calculated: 1A 5B_{HEX}

Address Of Peripheral	Code Function	Address Register Initial HI	Address Register LO Initial	Number Records HI	Number Records LO	Check-Error LO	Check-Error HI
40	03	00	01	00	03	5B	1A

Response. The response message specifies the address of the peripheral responds, the function code (03 in this case), the number of data bytes answered (that will be the double that number of records requested), the bytes of the CRC response and finally calculated. Data records, in response, are packaged in 2 bytes per record, with the binary contents right justified for each byte. For each record, the first byte contains the high order bits and the second the least.

Response format:

The peripheral address in hexadecimal is: 64_{DEC} = 40_{HEX}
 The number of bytes in hexadecimal answered is 06_{DEC} = 06_{HEX}
 The register1 in hexadecimal is: 1964_{DEC} = 07 AC_{HEX}
 The register2 in hexadecimal is: 8520_{DEC} = 21 48_{HEX}
 The registration3 in hexadecimal is: 0112_{DEC} = 00 70_{HEX}
 The CRC is calculated: 07 47_{HEX}

Address of Peripheral	Code of Function	Number Bytes Reply	Log1 HI Byte	Log1 LO Byte	Register2 HI Byte	Register2 LO Byte	Record3 HI Byte	Record3 LO Byte	Check-Error LO	Check-Error HI
40	03	06	07	AC	21	48	00	70	47	07

2.7 Function Code 04. READ INPUT REGISTERS

Read the binary contents of input registers from the slave.

Application. The request message specifies the address of the peripheral you want to read, the function code (04 in this case), the direction of the initial registration, the number of records to read and finally the calculated CRC. The registers are addressed from zero, ie registers 1-16 are addressed as 0-15.

Example: To request the peripheral 225 in the register of the address 0001. (In this recording the value stored as an example, 5555).

Application Form:

The peripheral address in hexadecimal is: 225_{DEC} = E1_{HEX}
 The initial register address in hexadecimal is 01_{DEC} = 00 01_{HEX}
 The number of records in hexadecimal is 01_{DEC} = 00 01_{HEX}
 The CRC is calculated: 6A 76_{HEX}

Address of Peripheral	Code Function	Address Register Initial HI	Address Register LO Initial	Number Records HI	Number Records LO	Check-Error LO	Check-Error HI
E1	04	00	01	00	01	76	6A

Response. The response message specifies the address of the peripheral responds, the function code (04 in this case), the number of data bytes answered (that will be the double that number of records requested), the bytes of the CRC response and finally calculated. Data records, in response, are packaged in 2 bytes per record, with the binary contents right justified for each byte. For each record, the first byte contains the high order bits and the second the least.

Response format:

The peripheral address in hexadecimal is: 225_{DEC} = E1_{HEX}
 The number of bytes in hexadecimal answered is: 02_{DEC} = 02_{HEX}
 The register1 in hexadecimal is: 5555_{DEC} = 15 B3_{HEX}
 The CRC is calculated: 77 C3_{HEX}

Address of Peripheral	Code of Function	Number Bytes Reply	Log1 HI Byte	Log1 LO Byte	Check-Error LO	Check-Error HI
E1	04	02	15	B3	C3	77

2.8 Function Code 06. PRESET SINGLE REGISTER

Predefines the binary content of a storage register of the slave.

Application. The request message specifies the address of the peripheral modify the code function (06 in this case), the address of the register to modify the preset value and finally the calculated CRC. The registers are addressed from zero, ie register 1 is addressed as 0. Example: To request the peripheral 23 to predefine register address 14 with the value 1234.

Application Form:

The peripheral address in hexadecimal is: 23_{DEC} = 17_{HEX}
 The register address to predefine in hexadecimal is: 14_{DEC} = 00 0E_{HEX}
 The preset value in hexadecimal is: 1234_{DEC} = 04 D2_{HEX}
 The CRC is calculated: 68 62_{HEX}

Address of Peripheral	Code Function	Address Register HI	Address Register LO	Value Preset HI	Value Prest LO	Check-Error LO	Check-Error HI
17	06	00	0E	04	D2	62	68

Response. The response message is an echo of the request, ie the address specified the peripheral response, the function code (06 in this case), the address of the register to modified, the default value and finally the calculated CRC.

Response format:

The peripheral address in hexadecimal is:

23_{DEC} = 17_{HEX}

The register address to predefine in hexadecimal is:

14_{DEC} = 00 0E_{HEX}

Preset value in hexadecimal is:

1234_{DEC} = 04 D2_{HEX}

The CRC is calculated:

68 62_{HEX}

Address of Peripheral	Code Function	Address Register HI	Address Register LO	Value Preset HI	Value Prest LO	Check-Error LO	Check-Error HI
17	06	00	0E	04	D2	62	68

2.9 Exception or error codes

When the Master sends a request to a slave can occur/ may happen:

- If the slave receives the request without a communication error, and can process the instruction, it returns a normal response.
- If the slave does not receive the request due to a communication error, no response. The Master program will process a timeout condition (timed out) and will be by the application.
- If the slave receives the request without a communication error, but can not carry it out (For example, the request is to read a record that is not available in this device), the slave returns an exception response informing the master of the nature of error.

The exception responses have two fields that differ from a normal response.

Function code field. In a normal response, the slave echoes the code function of the original request in the function code field of the response setting to 1 the MSB. With the function code in this way the Master knows that an error has occurred exception and can examine the data field to see what kind of error occurred.

Data field. In a normal response, the slave returns data in the data field (The information we required the Master). In an exception response, the slave returns an exception code in the data field. This defines the cause that gave the error slave.

Exception codes are standardized and are:

CODE	NAME	MEANING
01	Illegal Function	The function received can't be performed by the Slave.
02	Illegal Data Address	The data address requested is not available at the address Slave
03	Illegal Data Value	The value contained in the data field is not a value supported by the slave
04	Fault in Slave	An unrecoverable error occurred while the slave was doing the requested action
05	Recognition	The slave has received the request and is processing it, but requires much processing time. This response is given to avoid TimeOut error in the Master.
06	Slave Busy	Slave busy doing some process. The master must retransmit this message later.
07	No Recognition	The slave can't perform the function requested by the Master.
08	Parity Error	The slave tries to read extended memory, but detected a failure parity.

2.10 Map address for LYON Control Panel.

MSB	LSB	Nº BYTES	Area	Value	Description
Address		5632			
00h	00h	1	Status Control Panel	XXh	Control Panel priority status
00h	10h	1	Status Peripherals	00h	Normal
				21h	Fault repeater doesn't communicate
				22h	Fault Control Panel doesn't communicate
00h	11h+NºPerif	35+15+1	Peripheral X (Control Panel:35;Repeater:15)	00h	Normal
				21h	Fault Repeater doesn't communicate
				22h	Fault Control Panel doesn't communicate
00h	50h	1	Status Loop Controller	00h	Normal
				23h	Fault Loop Controller doesn't communicate
00h	51h+NºCTRL	5	Loop Controller X (From 1 to 5 and 11)	00h	Normal
				23h	Fault Loop Controller doesn't communicate
00h	60h	1	Status External Power Supply	00h	Normal
				31h	Fault charger
				32h	Fault witout battery.
				33h	Fault discharged battery.
				34h	Fault overcharged battery.
				35h	Fault Low power source net.
				36h	Fault High power source net.
				37h	Fault witout power source net.
				38h	Fault 30V
00h	61h	1	Status Battery	00h	Normal
				32h	Fault witout battery.
				33h	Fault discharged battery.
				34h	Fault overcharged battery.
00h	70h	1	Status General Relays	00h	Normal
				41h	Fault Open line.
				42h	Fault crossed line.
				91h	Trigged.
00h	71h	1	General Relay 1	00h	Normal
				41h	Fault Open line.
				42h	Fault crossed line.
				91h	Trigged.
00h	72h	1	General Relay 2	00h	Normal
				41h	Fault Open line.
				42h	Fault crossed line.
				91h	Trigged.
15h	00h	1	Status Loop Relays	00h	Normal
				91h	Trigged.
15h	01h+Nº Relé	200	Logical Relay X (Max:200)	00h	Normal
				91h	Trigged.
00h	80h+Loop Nº	20	Loop X (From Loop 1 to loop 20)	00h	Normal
				11h	Element in Test Mode
				12h	Element diabled
				81h	Fault dirty
				82h	Fault open loop
				83h	Fault Consumption
				80h	Fault sensor

				A0h	Alarm sensor
				XXh	Loop elements incidencies or Loop incidents.
00h	A0h	1	Reset	01h	Triggered
00h	A1h	1	Without Delay key	01h	Triggered
00h	A2h	1	Stop Sounder key	01h	Triggered
00h	A3h	1	Evacuation Key	01h	Triggered
00h	A4h	1	Manual Mode Key	01h	Triggered
01h+Loop Nº	01h+Element Nº	20*255=5100	Element X in Y Loop (Max: Y= 20 X=255)	00h	Normal
				11h	Element in Test Mode
				12h	Element disabled
				84h	Fault Element doesn't answer
				85h	Fault Kmay dirty
				86h	Fault kmay crossed linea
				87h	Fault Kmay open line
				88h	Fault Mstay crossed line
				89h	Fault Mstay open line
				8Ah	Technical fault
				8Bh	Fault open line.
				8Ch	Fault crossed line.
				8Dh	Fault Relay no active.
				8Eh	Fault 24V
				A1h	Intrusion Alarm
				A2h	LPG Alarm
				A3h	CO Alarm
				A4h	Technical Alarm
				A5h	Slave detector Alarm
				A6h	Temperature Alarm
				A7h	Smoke alarm
				A8h	Slave Manual Call Point Alarm
				A9h	Manual Call Point Alarm

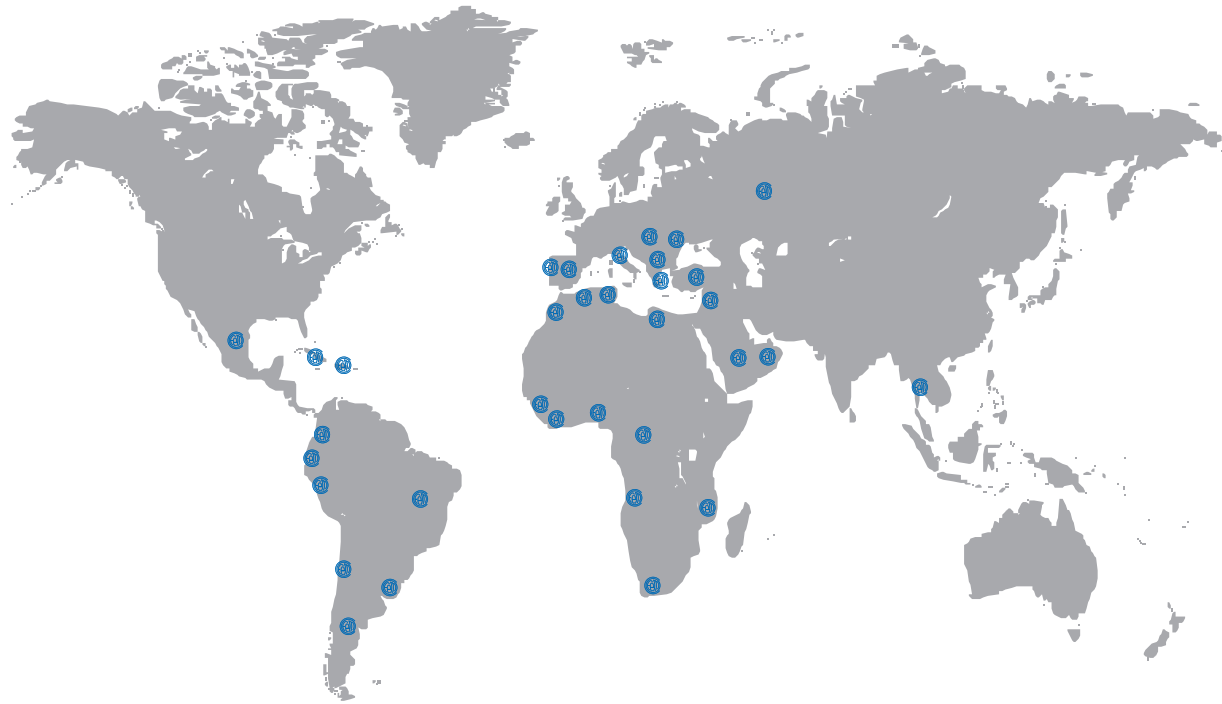
2.11 Map address for CLVR Control Panel.

MSB	LSB	Nº BYTES	Area	Value	Description
Address		275			
00h	00h	1	Status Control Panel	XXh	Control Panel priority status
00h	01h	1	General Relay Sounder Delay	XX	Value delay in seconds
00h	02h	1	General Fault Relay Delay	XX	Value delay in seconds
00h	03h	1	Realy 2 Delay	XX	Value delay in seconds
00h	04h	1	Extinguishing Delay Time for Reset	XX	Value delay in seconds
00h	60h	1	Power Supply Status	00h	Normal
				31h	Fault Charger
				33h	Fault discharged battery
				37h	Fault without power source net.
				38h	Fault 30V
00h	61h	1	Status Battery	00h	Normal
				33h	Faul Discharged Battery.
00h	70h	1	Status General relays	00h	Normal
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				62h	Triggered.
00h	71h	1	General Relay 1	00h	Normal
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				62h	Triggered.
00h	72h	1	General Relay 2	00h	Normal
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				91h	Triggered.
00h	73h	1	General Fault Relay	00h	Normal
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				91h	Triggered.
00h	A0h	1	Reset Key	01h	Triggered
00h	A1h	1	Without Delay key	01h	Triggered
00h	A2h	1	Stop Sounder key	01h	Triggered
00h	A3h	1	Extinguishing Activation	01h	Triggered
00h	A4h	1	Extinguishing Stop	01h	Triggered
00h	A5h	1	Tecla Zumbador	01h	Buzzer
01h	01h*Nº Zona	12	Zone X (Max: X=12)	00h	Normal
				11h	Zone in Test
				12h	Zone disabled
				13h	Alarm Detector in Test
				14h	Alarm Manual Call Point in Test
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				A7h	Smoke Alarm
				A9h	Manual Call Point Alarm
01h	10h	1	Status Relays	00h	Normal
				12h	Disabled Relay
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				8Eh	Fault 24V
				90h	Pre-alarm
				61h	Triggered.
				92h	Stopped

01h	11h*Nº Relé	2	Relay X (Max:2)	00h	Normal
				12h	Disabled Relay
				8Bh	Fault Open line.
				8Ch	Fault Crossed line.
				8Eh	Fault 24V
				90h	Pre-alarm
				91h	Trigged.
				92h	Sttoped

NOTE:

COFEM S.A. reserves the right to make changes due to typographical errors, prints information, or improvements to programs and / or equipment at any time and without notice.



FABRICANTE DE PRODUCTOS CONTRA INCENDIOS - FIRE PROTECTION MANUFACTURER
FABRICANTS DE PRODUITS CONTRE INCENDIES
Ctra. de Molins de Rei a Rubí, Km. 8,4 - 08191 RUBÍ (Barcelona) SPAIN

Comercial <i>Commercial</i>	+34 935 862 690	comercial@cofem.com cofem@cofem.com
Ventas <i>Sales</i>	+34 935 862 690	ventas1@cofem.com ventas2@cofem.com
Atención Técnico Comercial <i>Technical & Commercial Service</i>	+34 902 448 811	sat1@cofem.com sat2@cofem.com
S.A.T. <i>T.A.S</i>	+34 935 862 692	tecnic@cofem.com
Compras <i>Purchase</i>	+34 935 862 693	compras@cofem.com
Export <i>Export</i>	+34 935 862 694	export@cofem.com
Fax pedidos <i>Orders Fax</i>	+34 902 338 811	
Fax general <i>Fax</i>	+34 936 999 261	