

MANUAL

Alarma Cámaras Frigoríficas

© 2025 Ibercomp S.A.



INDICE

1 INTRODUCCIÓN.....	3
2 MONTAJE Y CONEXIONADO.....	6
3 FRONTAL DEL EQUIPO.....	10
4 DESCRIPCIÓN DEL PROTOCOLO DE COMUNICACIONES.....	11
5 SWITCHES DE CONFIGURACIÓN.....	15
6 REGISTROS.....	16
7 DESENSAMBLADO DEL MÓDULO.....	20
8 CONSEJOS SOBRE INSTALACIÓN RS485.....	21
8.1 Control de flujo mediante RTS.....	21
8.2 Terminación de las líneas.....	22
8.3 Polarización de las líneas.....	23
8.4 Utilización adecuada del cable.....	23
8.5 Instalación adecuada del cable.....	24
8.6 Protección contra sobre-tensiones.....	24
9 CONDICIONES DE GARANTÍA.....	26
10 CÁLCULO CRC.....	27
C/C++.....	27
JAVA.....	28
VB .net.....	32

1 INTRODUCCIÓN

Agradecemos sinceramente su confianza al elegir nuestros productos. Estamos comprometidos en ofrecerle soluciones de alta calidad que satisfagan sus necesidades y superen sus expectativas.

Este manual describe detalladamente el funcionamiento de nuestro sistema de alarmas para cámaras frigoríficas con comunicación Modbus. El equipo está diseñado para emitir alarmas sonoras y visuales en caso de que la puerta de la cámara permanezca abierta, garantizando así el mantenimiento de condiciones óptimas.

Además, el sistema incluye una sonda NTC para medir la temperatura interna de la cámara y, opcionalmente, puede utilizar una sonda AM2302 para monitorizar la temperatura y la humedad del entorno exterior.

La funcionalidad de comunicación remota permite integrar este sistema con un PLC (Controlador Lógico Programable) o un ordenador. Esto posibilita registrar y almacenar datos del estado de la cámara y las alarmas generadas, facilitando un análisis posterior. De esta manera, se pueden generar informes personalizados que cumplan con los requisitos legales y ayuden a optimizar el control de sus instalaciones.

Características generales:

Parámetro	Especificación
Construcción	Caja estanca de 90×90 mm, diseño compacto y resistente, ideal para montaje sobre la puerta de la cámara frigorífica.
Alimentación	12-24V CC.
Alerta Visual	LED frontal de alta intensidad, se activa automáticamente al abrir la puerta.
Alerta Sonora	Zumbador de 105 dB con pitidos intermitentes tras un tiempo programable (5 minutos por defecto).
Sensores Integrados	- Sensor magnético: Detecta el estado de la puerta (abierta o cerrada). - Sonda de temperatura NTC 10K: Precisión para medir la temperatura interna (-40 a 80°C). - Sensor opcional AM2302: Monitoreo de temperatura y humedad interna (-40 a 80°C y 0 a 99.9%).
Configuración Programable	Tiempo de alerta ajustable para adaptar el intervalo antes de que el zumbador y el LED se activen.
Comunicación	Puerto Modbus para configuración y monitoreo remoto.
Aplicaciones	Supervisión de puertas y temperaturas en cámaras frigoríficas para garantizar condiciones óptimas.

Es habitual que en distintos establecimientos la cámara quede parcialmente abierta o que haya algún fallo en la maquinaria que rompa la cadena de frío.

El equipo tiene las siguientes aplicaciones:

- **Almacenamiento de alimentos en hostelería y restauración:**

Supervisión de la temperatura en cámaras frigoríficas de restaurantes, hoteles, cafeterías y servicios de catering para garantizar la correcta conservación de los alimentos.

- **Supermercados y tiendas de alimentación:**

Control de cámaras refrigeradas utilizadas para almacenar productos perecederos como carnes, pescados, lácteos y congelados.

- **Almacenes logísticos de cadena de frío:**

Vigilancia de centros de distribución de alimentos o productos sensibles a la temperatura, evitando posibles fallos en la cadena de frío.

- **Industrias alimentarias:**

Monitorización de cámaras frigoríficas en fábricas que procesan, envasan o almacenan productos alimenticios.

- **Farmacias y laboratorios:**

Supervisión de cámaras donde se almacenan medicamentos, vacunas o muestras biológicas que requieren condiciones específicas de temperatura y humedad.

- **Hospitales y centros sanitarios:**

Uso en cámaras de almacenamiento para bancos de sangre, medicamentos o productos médicos sensibles a la temperatura.

- **Industria química:**

Vigilancia de cámaras refrigeradas donde se almacenan productos químicos que necesitan condiciones específicas de temperatura para su conservación.

- **Sector agrícola:**

Control de cámaras frigoríficas destinadas a la conservación de frutas, verduras y flores.

- **Cámaras de maduración de carne:**

Supervisión de las condiciones internas en cámaras diseñadas para la maduración de carnes, ayudando a mantener las temperaturas óptimas para el proceso.

- **Vinotecas y bodegas:**

Monitorización de la temperatura y humedad en cámaras para la conservación de vinos y otros productos sensibles.

- **Espacios de almacenamiento en grandes superficies:**

Vigilancia de zonas refrigeradas en centros comerciales y grandes almacenes para mantener productos perecederos en buen estado.

- **Centros educativos y residencias:**

Uso en cocinas y áreas de almacenamiento de alimentos para asegurar una correcta conservación.

- **Zonas de investigación y desarrollo:**

Monitorización de cámaras experimentales para pruebas de productos que requieren un control preciso de temperatura y humedad.

Los datos medidos pueden ser leídos remotamente usando un bus RS485 con protocolo modbus RTU. El RS485, un bus de comunicación económico y de trazado simple, permite la transmisión de datos a velocidades relativamente altas con distancias garantizadas de hasta 1.300 metros sin necesidad de repetidores, mostrando una gran inmunidad a interferencias.

El protocolo ModBus RTU, utilizado en esta familia de dispositivos, es abierto, estándar y universal, respaldado por numerosos entornos de desarrollo, autómatas, microcontroladores y SCADAs. Se ajusta a los estándares publicados por ModBus.org.

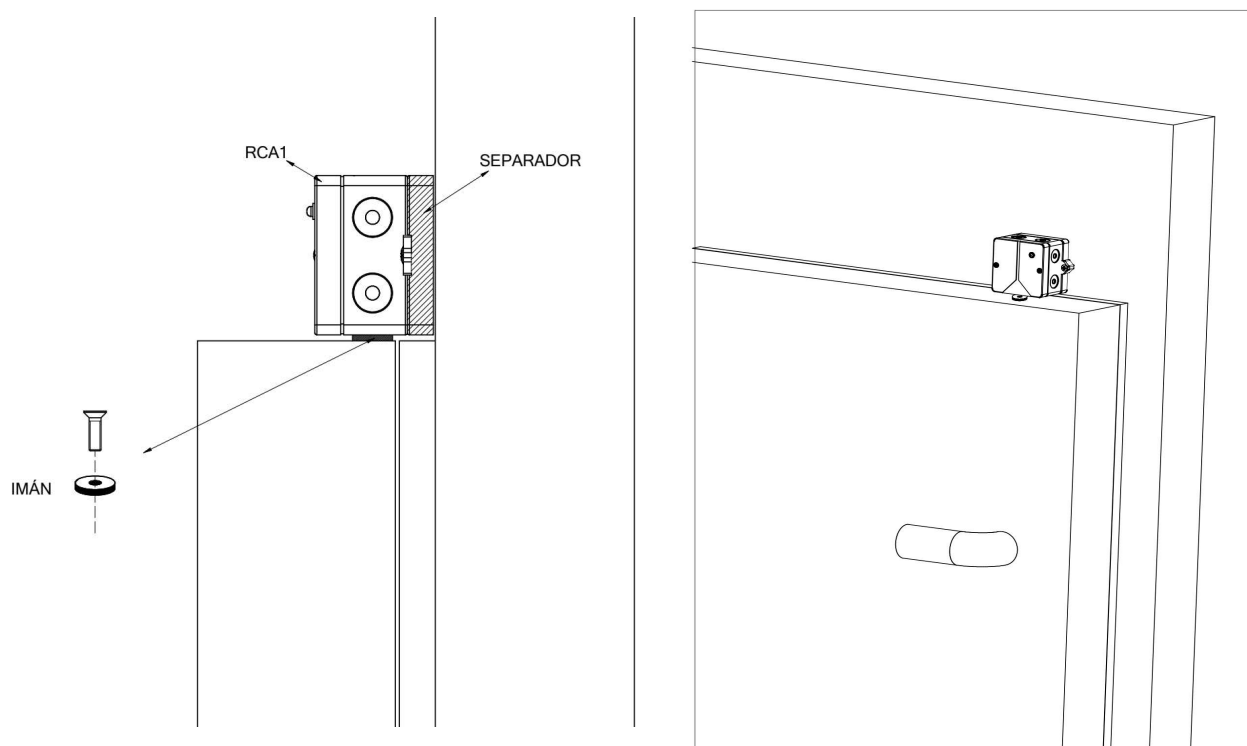
El bus de comunicación está debidamente protegido contra sobretensiones y sobre corrientes estáticas mediante dispositivos como TBU, TVS y GDT, ofreciendo una protección de nivel 4.

	Test	Normas
ESD	15 kV	IEC 61000-4-2 electrostatic discharge.
EFT	2 kV	IEC 61000-4-4 electrical fast transients.
Surge	6 kV	IEC 61000-4-5 surge immunity.

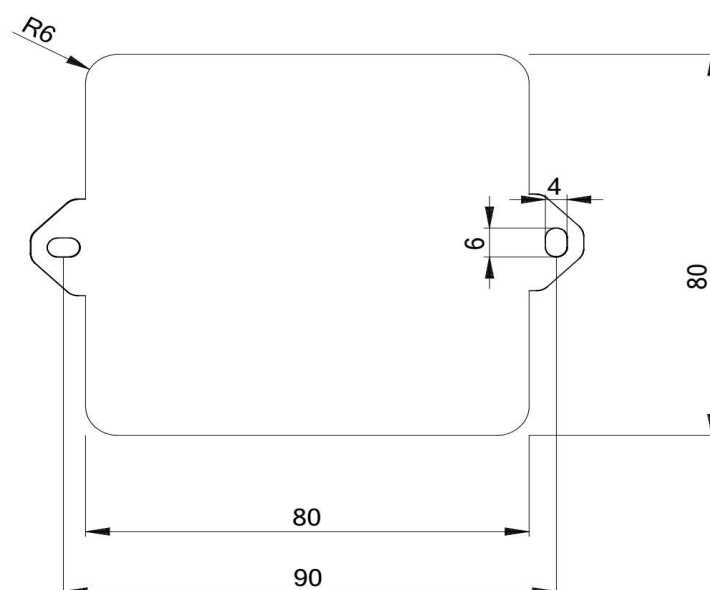
El equipo está integrado en una caja IP65 tamaño 80x80x50 mm que dispone de orejas para atornillar a bastidor o conducto.

2 MONTAJE Y CONEXIONADO

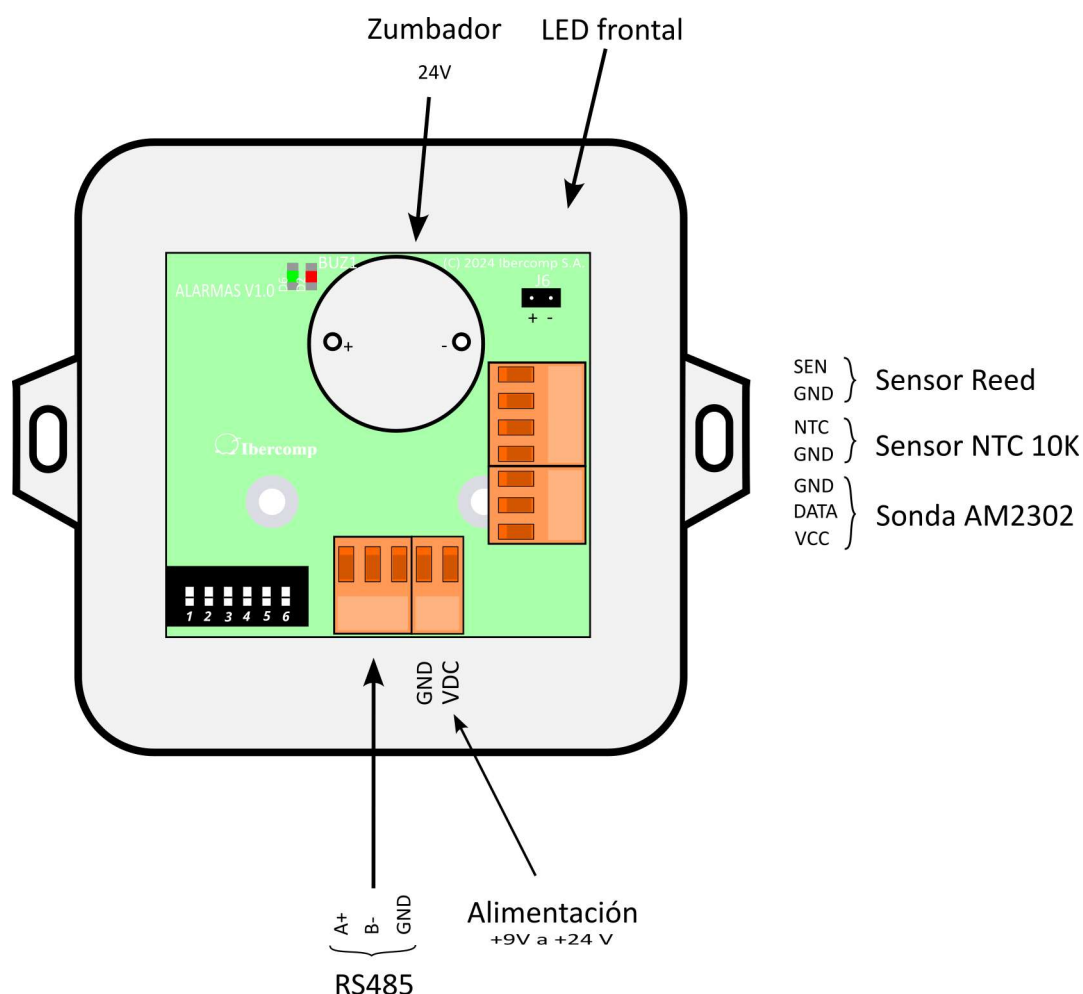
El equipo idealmente se puede montar sobre la puerta, atornillando sobre esta un imán de neodimio de 20 mm con un tornillo autoperforante. El imán no es necesario que esté perfectamente centrado. Si la distancia fuese grande se pueden apilar varios imanes.



En algunas instalaciones será necesario construir un separador con las siguiente medidas:



Abriendo la tapa del equipo, con cuidado de no tirar de los cables del LED y del zumbador, podrá ver en el interior el siguiente conexionado:



La alimentación de unos 10mA deberá ser una tensión continua entre 10V y 30V, siendo la más común 24V que es la que utilizan todos nuestros equipos industriales. Idealmente recomendamos nuestras fuentes para carril DIN.

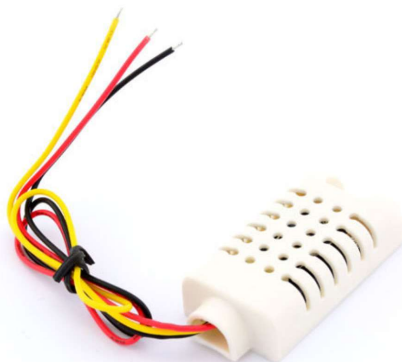
El LED frontal irá en su conector, cable rojo positivo, negro negativo.

En el conector de BUZ1, se puede montar un zumbador piezoeléctrico, el que nosotros incluimos puede dar hasta 105dB, pero lo alimentamos a 5V. Si desea tener un mayor volumen se puede cortar el cable rojo y conectarlo directamente a la alimentación VDC.

El sensor reed está integrado en la caja, es simplemente un contacto que está cerrado cuando la puerta de la cámara está cerrada. Se puede montar si se desea cualquier otra alternativa.

La sonda NTC de 10K a montar es la sonda referencia TEWA NTC TT02-10KC3-1D-1000-20-5 Water Proof. Otras sondas NTC de 10K pueden añadir errores a la medida, errores que mediante programa pueden ser corregidos. La sonda no debería estar a más de 10m del equipo.

Opcionalmente podemos incluir una sonda complementaria de temperatura/humedad tipo AM2302. Esta sonda se puede montar fuera de la cámara o dentro. Fuera podrá servir para ver el efecto que tiene la puerta sobre la temperatura/humedad exterior. Esta sonda hay que evitar que esté a más de 5 metros del equipo.

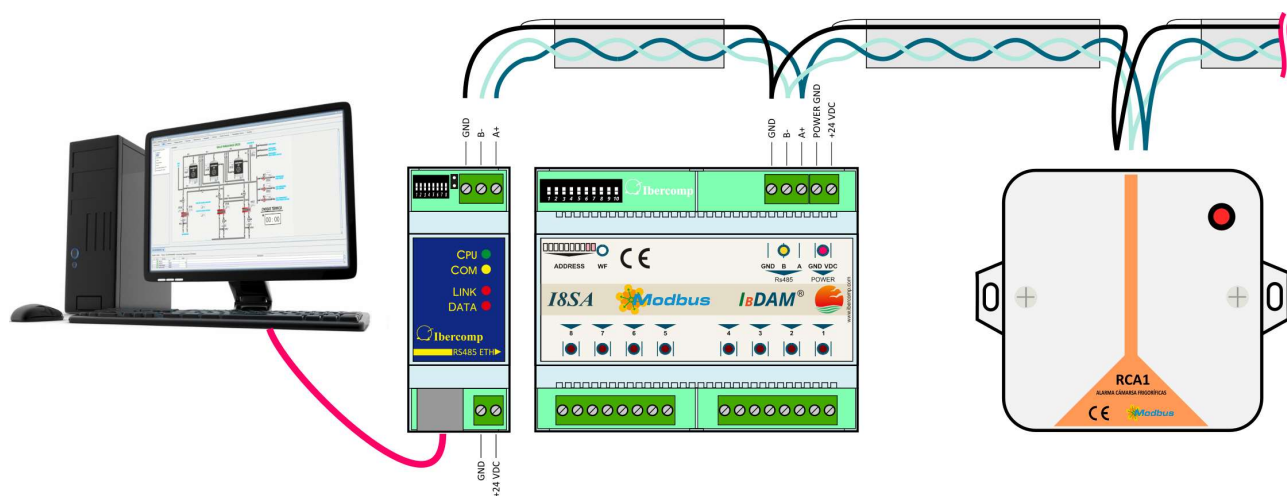


El conector RS485 contiene las líneas necesarias para poder comunicar remotamente desde un ordenador PC o bien desde un micro controlador o autómatas programables. En ambos casos las líneas RS485 se conectan bajo las siguientes reglas:

- a) Todas las señales A deben ir conectadas entre sí, del mismo modo deben ir las señales B. Es necesario unir adecuadamente todas las tierras (para más información lea el capítulo "Consejos sobre instalación RS485").
- b) Las líneas RS485 deben tener una longitud máxima de 1300 metros entre el primer equipo y el último (aproximadamente 4000 pies).
- c) Las líneas RS485 idealmente no deben tener bifurcaciones.

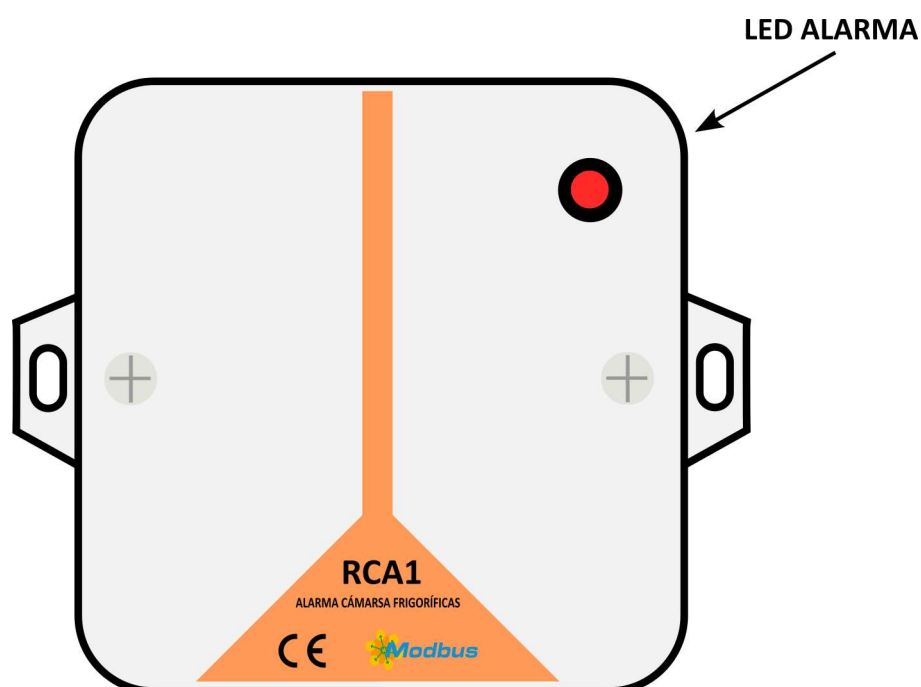
OJO: Debe tener presente que nosotros utilizamos la nomenclatura utilizada por los fabricantes de transceptores, es decir A tiene la señal positiva y B la señal negativa. En algunos PLC utilizan la nomenclatura contraria. A nuestro A siempre debe llegar el positivo y a B el negativo, independientemente de como lo denominen en el PLC.

En la siguiente ilustración mostramos una línea donde se utiliza nuestro conversor Ethernet a RS485 para acceder a varios equipos entre los que hay una alarma de cámara frigorífica RCA1.



3 FRONTAL DEL EQUIPO

El frontal dispone de un indicador LED perfectamente visible. Tienen el siguiente significado:



LED ALARMA

Este LED de alto brillo rojo, se activa cuando la puerta está abierta, y se pone a parpadear de forma intermitente cuando la puerta está abierta más de un tiempo predefinido..

4 DESCRIPCIÓN DEL PROTOCOLO DE COMUNICACIONES

Sobre la red física 485 es necesario implementar algunas normas de comunicación que permitan acceder a unos u otros dispositivos. Estas normas se denominan “protocolo de comunicación”.

Es difícil definir el mejor protocolo y prueba de ello es la diversidad de protocolos que coexisten en el mercado, siendo los más conocidos ModBus, PROFIBUS, CHIPBUS, etc... Todos tienen alguna ventaja frente a sus competidores y la elección de uno u otro dependerá de las exigencias de la aplicación.

Dentro de un protocolo de comunicación los datos son transmitidos en paquetes de información, siendo lo habitual que dispongan de una cabecera seguido de unos datos y una cola.

La mayoría de los protocolos son complejos y suelen utilizar paquetes de datos demasiado grandes para transmitir poca información.

Nosotros hemos optado por dotar a esta familia de módulos con el protocolo ModBus RTU, ajustándonos como se mencionó antes a las especificaciones dadas por ModBus.org.

Los motivos de la elección de este protocolo son:

- Disponer de un protocolo ampliamente estandarizado que sea utilizado por programas Scada, autómatas,... ya existentes.
- Un protocolo que sea sencillo de implementar y sea abierto, es decir, sin complicaciones ni secretos ni “royalties”.
- Que pueda utilizar una red física RS485.

Los dispositivos ModBus disponen en su interior de registros que pueden asemejarse a variables. Un registro puede ser físico, por ejemplo el estado de un relé, o lógico, por ejemplo el valor de un temporizador.

Se definen cuatro tipos de registros:

Discrete input

Se trata de registros de 1 bit de solo lectura. Normalmente están asociados a entradas digitales.

Coil

Registros de 1 bit de salida. Pueden leerse y escribirse. Normalmente están asociados a entradas digitales.

Input Register

Registro de 16 bits de lectura. No pueden escribirse y su función es servir de soporte para entradas analógicas.

Holding Register

Son registros de 16 bits de salida, que pueden leerse y escribirse. Normalmente están asociados a entradas analógicas, aunque su significado es definido por el fabricante del módulo.

En nuestro caso, como indicamos antes, los 16 primeros “holding registers” son comunes a todos los equipos y sirven para identificar la tarjeta, así como para definir algunos comportamientos.

En nuestros módulos los cuatro tipos de registros tienen mapas distintos, es decir, el valor del “holding register” 7 no coincide con el del “input register” 7. Son dos registros distintos.

El protocolo ModBus está orientado a mensajes y es maestro-esclavo. Es decir, en la red RS485 sólo hay un único dispositivo que hace de maestro y numerosos equipos que hacen de esclavo.

No puede haber más de un equipo maestro al mismo tiempo porque el protocolo no sabe nada de resolver colisiones. A nivel lógico utilizando nuestro servicio ModBus sí se podría implementar un sistema donde varias aplicaciones master tipo SCADA pudiesen acceder al mismo tiempo a un módulo. Eso se escapa de este manual.

El maestro, normalmente un ordenador, autómatas o micro controlador, mantiene un diálogo imperativo con los esclavos. Para ello les envía mensajes y ellos los interpretan, obedecen y responden. Cada vez que se pregunta a un esclavo, este responde.

Dentro de ModBus hay dos tipos de mensajes: los mensajes ASCII y los binarios. Estos últimos son también conocidos como RTU y son los que implementan nuestros equipos.

Los mensajes tienen la siguiente forma:

Dirección	Cod. Función	D A T O S	CRC
-----------	--------------	-----------	-----

El maestro envía un mensaje y el esclavo devuelve otro mensaje, respetando el mismo formato. El significado de cada una de las partes es el siguiente:

Dirección	Es la dirección del dispositivo esclavo al que ha sido dirigido el mensaje. Cuando el esclavo responde indica su origen en este campo. Debe tener presente que en ModBus son válidas solamente las direcciones 1 a 247. Las direcciones 248 a 255 están reservadas y no deben ser utilizadas. Todos los equipos responden a la dirección 254, pero esta solo debe ser empleada si hay un único equipo en la red, pues todas las respuestas van a colisionar.
Cod. Función	Se trata del código de función, es decir, lo que se espera que haga el módulo esclavo. El código de función, como se describirá más adelante, sirve básicamente para decirle al esclavo que se desea leer o escribir sobre un registro. En la respuesta este código de función puede también contener una indicación de que se ha producido un error.
Datos	Es un número de bytes que contiene datos adicionales al código de la función, por ejemplo, si se desea escribir sobre relés, los datos deberán contener el estado deseado para los relés.
CRC	Es un valor que permite saber si el contenido del mensaje ha sido alterado o no por el camino. Se trata de una función polinómica de 16 bits que se muestra al final de este manual.

Para poder diferenciar el inicio y el final de un paquete, entre estos se intercala un tiempo en el que ningún equipo transmite nada. Este tiempo debe ser de al menos 3,5 caracteres, es decir, 35 bits.

Si está trabajando a 9600 baudios, este tiempo será de 35/9600 baudios, es decir, unos 3,6ms. Con la finalidad de evitar que los tiempos sean excesivamente cortos cuando aumenta la velocidad se pone como tiempo mínimo 1,7ms.

Para evitar confusiones, entre caracteres de un mismo mensaje no debe haber más de 1,5 caracteres. Para evitar tiempos muy grandes, se pone un tiempo máximo de 759us para velocidades superiores a 19200 baudios.

En la práctica, cumplir este requisito es sencillo. El programador únicamente debe asegurarse de que los caracteres de un mensaje se transmiten seguidos, sin pausas ni interrupciones. Para ello la solución es cargar todo el mensaje en una variable o arreglo y luego enviar el conjunto hacia el puerto RS485.

Por el contrario, si se envía primero el byte de dirección, el código de función, luego los datos y finalmente se calcula el CRC, es garantía de que no funcionará. Posiblemente calcular el

CRC requiera un tiempo y posiblemente entre las muchas instrucciones el sistema operativo realice un cambio de contexto (asignar la CPU a otro proceso/programa).

El protocolo ModBus tiene un gran número de códigos de función, no siendo necesarios todos. Nosotros no los hemos implementados todos. Los que están implementados en nuestras tarjetas son los siguientes:

0x01 Read Coils

Permite leer el estado de los “coils”.

0x02 Read Discrete Inputs

Permite leer el estado de los “discrete inputs”.

0x03 Read Holding Registers

Permite leer los registros de 16 bits del equipo.

0x04 Read Input Register

Permite leer los registros de 16 bits del equipo.

0x05 Write Single Coil

Permite definir el estado de un “coil”.

0x06 Write Single Holding Register

Permite definir el estado de un “holding register”.

0x0F Write Multiple Coils

Permite mediante una transacción escribir varios/todos los relés.

0x10 Write Multiple Holding Register

Permite mediante una transacción escribir varios/todos los registros .

5 SWITCHES DE CONFIGURACIÓN

En el circuito del equipo WS1 hay un array de 6 switches que sirven para poder configurar el id modbus y la velocidad en baudios:



Switches	Descripción
1 a 5	Selecciona en binario la dirección de la tarjeta dentro de la red, siendo válidas las tarjetas 1 a 31.
6	10/9 Velocidad baudios. 0 9600 baudios 1 19200 baudios

Si se pone todos los switches 1, al darle corriente adquirirá los valores de fábrica.

Al darle corriente tardará unos segundos, y cuando esté reprogramado ambos LEDs de la placa harán un parpadeo rápido.

Una vez reiniciados deberá poner la dirección deseada, quitar corriente y volverla a conectar. Si pone todos los switches a OFF cogerá los valores indicados por software.

En esta tabla un estado 1 es sinónimo de un estado ON y un estado 0 es sinónimo de un estado OFF.

Si se ponen los switches a ON una vez arrancada la placa, el LED rojo empezará a parpadear, con lo que podrá accederse a la placa utilizando la dirección 247. Con el programa Modbus ID podrá modificar la velocidad y modbus id de usuario.

6 REGISTROS

TABLA DISCRETE INPUTS

--	--	--

Este equipo carece de "discrete inputs".

TABLA COILS

nº	Tipo	Descripción
0		Si está a 1 tiene el acceso a algunos holding registros bloqueados. Si está a 0 la escritura a los registros se desbloquean.
1	RO	Se pone a 1 para indicar que todos los switches están a ON. En este modo la placa responderá a la dirección 247, pero debe bajarse al menos un switch, de lo contrario si se corta la corriente y vuelve se reiniciará a fábrica y quedará bloqueado.
2	RO	Se pone a 1 para indicar que hay un error en la sonda de temperatura, es decir que o no está, está cortocircuitada o está fuera del rango -30°C a 140°C. OJO: Puede medir correctamente entre -40°C y -30°C aunque el error esté activado.
3	RO	Se pone a 1 para indicar que no detecta sonda AM2302.
4	RO	Es el estado del sensor de puerta, 1 indica que esta está abierta y 0 que está cerrada.

TABLA REGISTER INPUTS

--	--	--

Este equipo carece de "register inputs".

TABLA HOLDING REGISTERS

Los módulos disponen de unos registros tipo "holding register" que permiten conocer algunas características del módulo. Algunos de estos registros de configuración son preservados en una memoria E2PROM.

Los primeros dieciséis registros son los "holding registers" comunes a todas las placas. Permiten conocer de qué tipo de módulo se trata, versión de firmware, número de registros, ...

nº	Tipo	Descripción
0	EE	Número de arranques del módulo. Cada vez que el módulo arranca o es reiniciado se incrementa el valor de este registro en una unidad. Con ello

nº	Tipo	Descripción
		se puede conocer si un módulo se está reiniciando de forma anómala.
1	RO	Versión del equipo. La versión viene codificada como versión*10 + subversión. La versión 1.0 se representa como 10.
2	RO	Código del módulo, en este caso: 224 RCA1
3	EE	Velocidad en baudios de usuario, es decir la que tiene el equipo si se arranca con todos los switches puestos a cero. Por defecto 9600 baudios Se pueden poner los valores estándares como por ejemplo 4800, 9600, 19200, ..., 57600. Por ejemplo si pone 19200 baudios, el sistema calculará la velocidad real más cercana que puede tener el procesador, en este caso 19230. Una diferencia pequeña en la velocidad no tiene importancia. Para poderlo modificar el coil[0] debe tener valor 0 y el holding register 15 debe tener un valor 3.
4	EE	R E S E R V A D O
5	EE	Dirección modbus que tiene el equipo si se arranca con los switches puestos a cero. Por defecto id=1. Para poderlo modificar el coil[0] debe tener valor 0 y el holding register 15 debe tener un valor 3.
6	RO	Número de registros tipo "discrete input"
7	RO	Número de registros tipo "coil"
8	RO	Número de registros tipo "input register"
9	RO	Número de "holding registers". Debe tener presente que los 16 primeros siempre serán comunes a todos los módulos y que los registros propios del módulo comienzan en el registro 16.
10	EE	R E S E R V A D O
11	EE	R E S E R V A D O
12	RO	Muestra el estado de los switches. Como todas las placas responden a la dirección 254, conectando una a la línea se puede comprobar si los switches funcionan correctamente. También se puede utilizar en conjunción del registro 2 y 5 para deducir si el módulo está instalado en el lugar correcto.
13		R E S E R V A D O
14		R E S E R V A D O
15	MM	Escribiendo secuencialmente los valores adecuados se puede reiniciar el módulo remotamente. Es equivalente a todos los efectos a desconectarlo de la corriente y volverlo a conectar.

nº	Tipo	Descripción
		Antes de empezar deberá leer el registro 15 y asegurarse que tiene un valor 0. Si no lo tuviese escriba cualquier valor que no fuese 0x55 o 0xAA (85 o 170). Una vez escrito vuelva a comprobar. Para iniciar la secuencia habrá que escribir el valor 0x55 (85), luego escribir un 0xAA (170) y finalmente un 0xAA (85). Si se lee el registro se obtendrá el valor de reinicio de la máquina. 0, 1 y 2. La tercera escritura no será respondida dando un fallo de comunicaciones. Sabrá que el equipo se ha reiniciado cuando se incremente el registro 0 en una unidad. Si teniendo el registro un valor 0 se escribe el valor 0xA5A5 (42405) este registro tomará valor 3 y se permitirá modificar los registros 3 y 5 (si además el coil[0]=0).
16	RO	Temperatura de la sonda NTC en décimas de grado. Es un número entero de 16 bits con signo complemento a 2.
17	RO	Valor en Ohmios medido en la entrada. Puede ser útil si se monta otro tipo de sonda distinto. El equipo que hace la lectura en ohmios deberá calcular la temperatura real.
18	RO	Valor del conversor analógico digital. Hay que tener presente que da un valor analógico obtenido de un divisor resistivo donde la resistencia de arriba es de 10K 1% y la de abajo la sonda. De forma un poco más compleja pero más precisa se puede inferir a partir de este valor la temperatura real.
19	EE	En algunas instalaciones puede ser que la temperatura medida por la sonda no coincida exactamente con la temperatura real de la cámara. Esto puede deberse por ejemplo a la ubicación de la sonda. Se puede añadir un offset que suba o baje toda la pendiente, o que cambie su inclinación. Este registro contiene el offset en décimas de grado cuando la temperatura es 0°C. Para poder modificar este registro es necesario que el coil[0] tenga un valor 0.
20	EE	Este registro contiene el offset cuando la temperatura es 100°C. Al igual que el registro anterior es un número de 16 bits con signo complemento a 2. Si ambos registros de offset se les suma o resta la misma cantidad la pendiente entera se desplazará manteniendo su inclinación.

nº	Tipo	Descripción
		Para poder modificar este registro es necesario que el coil[0] tenga un valor 0.
21	RO	Temperatura de la sonda AM2302 en décimas de grado. Número de 16 bits con signo complemento a 2. Solo será válida la medida si no hay error en la lectura.
22	RO	Humedad de la sonda AM2302 x 10 en %. El valor va desde 0 a 999. Solo será válida la medida si no hay error en la lectura.
23	RO	Parte alta de un contador de 32 bits que nos indica cuantos segundos lleva la puerta abierta.
24	RO	Parte baja de un contador de 32 bits que nos indica cuantos segundos lleva la puerta a abierta.
25	EE	Tiempo que ha de estar la puerta abierta en segundos para que salte la alarma.

LEYENDA:

- EE Indica que son registros persistentes que se almacenan en la memoria E2PROM. Estos registros pueden ser rescritos pero debe tener presente que después de 1.000.000 de escrituras su contenido puede no ser correcto.
- MM Indica que el registro es almacenado en memoria. Si se corta la corriente se pierde. Puede escribir el registro tantas veces como sea necesario.
- RO Indica que se trata de un registro de solo lectura. Las escrituras sobre este registro serán ignoradas.

7 DESENSAMBLADO DEL MÓDULO

Para abrir el equipo, debe extraer los tornillos que hay en la tapa y retirarla con cuidado para no pegar un tirón a los cables del LED frontal y del zumbador. Una vez abierto tendrá acceso a los switches de configuración descritos en el apartado 5.

También tendrá acceso a las distintas regletas de conexionado.



8 CONSEJOS SOBRE INSTALACIÓN RS485

Montar una red RS485 no es una tarea delicada ni complicada y menos cuando se trata de redes a baja velocidad. En la práctica no es nada exigente ni en cuanto a tipología ni en cuanto a tipo de cable.

Ahora bien, todos los transceptores RS485, incluyendo los de los equipos de la competencia, son muy susceptibles a sobretensiones, de modo que si no se tiene cuidado en algunos puntos la instalación además de no funcionar se romperá con casi toda seguridad.

Los transceptores utilizados son 65HVD3082, que soportan 128 nodos y son robustos frente a descargas estáticas. Este integrado se puede sustituir por otros compatibles con otras características.

Estos integrados transceptores están siempre en zócalo.

En una instalación mal realizada bastará la caída de un rayo a unos centenares de metros para que quede dañada y fuera de servicio.

También es necesario entender el concepto de transmisión “Half Duplex” y que si son necesarios dos hilos no es que uno sea la transmisión y el otro la recepción.

A continuación le describimos algunas ideas o recetas que pueden serle útiles para realizar una buena instalación.

8.1 Control de flujo mediante RTS

El protocolo RS485 es “half duplex”, lo cual quiere decir que se utilizan las mismas líneas para la transmisión y recepción de datos. De modo que además de transmitir o recibir datos es necesario realizar un control de flujo.

En sistemas donde no se tiene en cuenta o no se gestiona correctamente es típico que ocurra lo siguiente:

- a) Combinaciones de tarjetas que son incompatibles entre sí, por lo que hay que estar probando cuales pueden mezclarse o no.
- b) El sistema funciona correctamente encima de la mesa del taller o del laboratorio pero en la instalación real da problemas.

Si no gestionamos la línea RTS, es probable que por casualidad todos los equipos están en modo transmisión, transmitiendo una señal 1. Si alguno transmite una señal 0 es posible en

distancias muy cortas, que los demás equipos reciban este 0. En cualquier caso es necesario que se gestione correctamente esta línea.

Generalmente en los ordenadores que disponen de puertos RS485 (PCs industriales) este control de flujo se realiza mediante la línea RTS. Esta debe ser puesta a nivel alto antes de iniciar la transmisión de información y puesta de nuevo a nivel bajo para esperar la recepción de datos.

Esta tarea parece simple, pero no lo es por lo siguiente: cuando envía desde un programa un carácter hacia el puerto serie, en realidad se lo está enviando a un buffer que controla el sistema operativo, de modo que de antemano no conoce ni cuando comienza la transmisión ni cuando termina.

Existen varias soluciones a este problema, la primera solución es dotar al sistema operativo de un “driver” que controle la línea RTS a bajo nivel o bien, si el lenguaje lo dispone, habilitar este protocolo de transmisión. No debe confundirse con el control de flujo RTS/CTS, no tiene nada que ver y no sirve para gestionar una RED RS485.

La segunda solución es utilizar un puerto serie RS232c y conectar sobre él un conversor RS232c $\leftrightarrow$ RS485 con control de flujo automático. Aquí puede surgir un segundo problema; existen algunos conversores que añaden un retardo a este control de flujo. Si este retardo es mayor que 3,5 caracteres tenemos la posibilidad de perder el inicio del paquete de respuesta.

Normalmente ningún conversor tiene retardos tan largos, pero si por el camino se insertan repetidores y distintos convertidores RS485-fibra puede llegar a tenerse dicho problema.

Debe tener presente que a diferencia de un puerto serie “full duplex” (RS232 ó RS422) en un puerto “half duplex” (RS485) se recibe siempre un Eco de los datos transmitidos. De modo que al preguntar algo a una tarjeta periférica se recibe la pregunta seguida de la respuesta.

Existen en el mercado algunos conversores RS232c $\leftrightarrow$ RS485 que disponen de una opción de cancelación de Eco, pero es mejor que el software utilice este Eco para comprobar que la línea funciona correctamente. Por ejemplo un corto entre las líneas A y B haría que no se produjese ese ECO y en un sistema “MultiMaster” podría servir para detectar la colisión de mensajes (aunque no es una buena idea en ModBus).

8.2 Terminación de las líneas

En algunos casos es necesario añadir en los extremos de las líneas RS485 unas resistencias terminadoras para adaptar la impedancia de la línea a las impedancias de los transceptores. Cuando las impedancias no coinciden, la señal transmitida no es absorbida adecuadamente por lo que parte de ella se refleja de nuevo en la línea.

Añadiéndolas eliminamos este reflejo pero aumentamos la complejidad de la instalación y aumentamos el consumo de los transceptores. Cuando utilizamos la línea RS485 a baja velocidad,

caso de nuestras tarjetas, no adquirimos ninguna ventaja al instalar dichas resistencias ya que el reflejo no afecta a las comunicaciones.

Si utiliza un conversor RS232c $\leftrightarrow$ RS485 que se alimenta del propio puerto serie no debe utilizar este tipo de terminaciones. Si va a utilizar resistencias terminadoras debe instalar solo una en cada extremo de la línea y calcularlas para que la impedancia de esta sea 120 Ohmios.

En el caso particular de realizar una instalación con nuestras tarjetas no recomendamos la instalación de dichas resistencias, aunque por experiencia puede conectarlas y puede servirle para verificar un buen funcionamiento.

Si lo hace y el sistema funciona sin fallos es que todos los equipos funcionan correctamente, en caso contrario algún transceptor o la línea tienen algún problema. Normalmente sin las resistencias las comunicaciones funcionarán.

8.3 Polarización de las líneas

Cuando en la línea RS485 ningún equipo está transmitiendo, está en alta impedancia, esto es, no hay ninguna tensión presente. En este estado es posible que se induzca fácilmente corriente que puede llegar a confundirse con la recepción de un dato.

Si bien en casi ningún caso es necesario, puede añadir unas resistencias que conecten la línea B a +5V y la línea A a GND de forma que entre A y B haya por lo menos 200 mV. El valor de estas resistencias depende del número de equipos instalados y de la instalación. Con ello asegurará en los momentos de silencio un estado definido.

Estas resistencias pueden ser montadas en cualquier punto de la red, o incluso las puede repartir. Si estas resistencias son demasiado grandes (o no están montadas) baja la inmunidad contra el ruido. Por el contrario si las resistencias son demasiado pequeñas a los transceptores les cuesta más transmitir (=menos distancia y menos equipos conectados).

Algunos conversores RS232c $\leftrightarrow$ RS485 no admiten este tipo de resistencias.

8.4 Utilización adecuada del cable

En una instalación RS485 se requiere un mínimo de 3 líneas, 2 para la transmisión/recepción de datos y una para unificar las tierras (GND). Es importante utilizar el cable adecuado. Para la transmisión de datos se debe utilizar un cable trenzado preferiblemente apantallado.

No unir las tierra de los diferentes equipos es garantía de que con el tiempo surgirán problemas, incluso en tarjetas o equipos en los que el transceptor RS485 esté optoacoplado y el fabricante indique que con dos líneas es suficiente.

Aunque funcione, no es recomendable utilizar el apantallamiento del cable para unir las tierras. Recomendamos utilizar un segundo par para este fin y si lo desea puede utilizar un tercer par para la alimentación (+24 DC) de los equipos.

El motivo es que la resistividad del apantallado del cable es más alta que los hilos de datos y por tanto pueden aparecer diferencias de tensión importantes entre las líneas de datos y tierra de dos equipos distantes.

Recomendamos por tanto la utilización de un cable apantallado con 3 pares trenzados, un par de datos, un par para tierras y un par de alimentación. La pantalla debe estar conectada a una buena tierra en un solo punto.

Si el presupuesto no es un problema es mejor utilizar un cable con aislantes de polietileno en vez de PVC ya que disponen de menor atenuación de señal.

Un cable apropiado es el cable Ethernet STP categoría 5, es barato, sencillo de localizar y supera con creces las especificaciones RS485.

8.5 Instalación adecuada del cable

Una instalación RS485 debe trazarse en línea, esto es, debe tener un inicio y un final. A lo largo de la línea se pueden ir conectando equipos. El trozo de cable que une el equipo con la línea debe ser muy corto.

No es correcto realizar una instalación en la que la línea se divida en varias (configuración en estrella), ya que puede haber reflejos desfasados que líen la comunicación.

La longitud máxima que puede tener la línea es de 1.300 metros de extremo a extremo. En la práctica, aun no siendo recomendable, utilizando velocidades bajas (ej: 9600 baudios) funciona una instalación con topología en estrella y longitudes superiores a los 1.300 metros.

Nosotros hemos llegado a ver funcionando sin problemas una instalación con 12Km de cable RS485, 9600 baudios, sin repetidores y con una topología no conforme a ninguna recomendación. No nos parece correcto, pero en ese caso concreto, inesperadamente funciona.

8.6 Protección contra sobre-tensiones

Como ya indicamos antes, los transceptores RS485 son muy delicados y sensibles ante sobretensiones y por lo general su deterioro es el responsable de casi todas las averías que se producen en instalaciones RS485.

Estas sobretensiones en la mayor parte de los casos tienen su origen en una instalación mal diseñada como por ejemplo pasar las líneas RS485 con cable inadecuado junto con cables de alta tensión. En otras ocasiones no podrán evitarse debido a un presupuesto que impida separar las líneas o bien se deberán a fenómenos tales como la caída próxima de un rayo.

Nuestros equipos incluyen unos varistores de protección, que serán una protección suficiente para instalaciones que vayan a realizarse en el interior de edificios o que estén dotadas de una buena toma tierra.

Pero en instalaciones con longitud de cable grande donde no haya toma de tierra, que unan dos edificios distantes o bien que estén en pleno campo, de no tomar ninguna protección tenemos la seguridad de que a la primera de cambio se romperán la mitad de transceptores quedando el sistema inoperativo.

Podemos tomar varias medidas:

Asegurarnos de disponer de una buena tierra que esté unificada en toda la instalación, aunque ello suponga montar un cable independiente con una sección generosa.

Instalar un protector contra sobretensiones con varios niveles de protección en cascada (fusibles, traszorb, descargadores de Gas y varistores). Este debe ser instalado en mitad de la línea, por ejemplo en una instalación que une dos edificios debería montarse a la entrada y salida de los mismos.

Instalar en algunos puntos medios de la red repetidores optoaislados.

Intercalar en la líneas fusibles rápidos de 100 mA. Es preferible que se fundan estos a que se fundan los transceptores.

9 CONDICIONES DE GARANTÍA

Nosotros hemos puesto el máximo interés y esmero en el diseño de esta nueva familia de tarjetas RS485 y están concebidas para funcionar todos los días 24 horas sin que se manifiesten problemas.

Ibercomp SA se compromete a dar garantía contra todo defecto de fabricación de 3 años desde la fecha de factura de las alarmas de cámaras frigoríficas.

Queda exenta de garantía cualquier rotura que se haya realizado físicamente, es decir, resultado directo de una mala instalación, alimentación a una tensión incorrecta o sobretensiones inducidas a través de la red eléctrica o líneas de datos.

La garantía incluye todos los elementos que hayan de sustituirse y la mano de obra necesaria para realizar dicha reparación. Si el equipo no tuviese arreglo bajo garantía se sustituirá por otro nuevo idéntico o modelo superior sustitutivo.

La garantía no incluye los gastos de transportista hacia nuestros talleres.

Ibercomp SA, como empresa fabricante del equipo, no se hace responsable de:

- 1) Cualquier daño o perjuicio que pueda ocasionar el equipo como consecuencia de un fallo de su funcionamiento o pérdida de datos.
- 2) Los fallos o defectos que pudieran hallarse en el software o ejemplos asociados al mismo. Estos están en continuo desarrollo y podrán actualizarse de forma gratuita siempre que haya una nueva versión disponible.

El software o ejemplos que suministramos son gratuitos, aunque Ibercomp posee sus derechos. Usted es libre de utilizarlo, utilizar el de terceros o bien desarrollar su propio software.

Ibercomp S.A. garantiza la aceptación de la devolución del equipo hasta 30 días después de la fecha de adquisición con la condición de que el equipo y accesorios así como su embalaje se encuentren en su estado original.

Si la adquisición fue realizada a través de tienda o distribuidor, la devolución del paquete deberá realizarse a través de la misma.


```

0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6,
0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE,
0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA,
0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,
0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62,
0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE,
0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,
0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76,
0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A,
0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86,
0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40
};

```

JAVA

```

package com.ibercomp.ModBus;

/**
 * Paquete de ModBus. En la nomenclatura ModBus
 * esta clase representa un mensaje RTU.
 * @author A.Miguel - (C) 2009 Ibercomp SA
 * @version 1.00
 */
public class ModBusPacket {

    private int uchCRCHi = 0xFF ;    //Byte alto del CRC
    private int uchCRCLo = 0xFF ;    //Byte bajo del CRC

    private byte [] trama=null;
    private int lenght=0;

    /**
     * Construye un nuevo paquete.
     */
    public ModBusPacket() {
        trama=new byte[256];    //La longitud máxima de una trama es de 256
    }

    /**
     * Obtiene la trama del paquete.
     * @return El resultado es un byte[] de 256 elementos. <BR/>
     * Los elementos pueden tener valores de -128 a 127, y solo son válidos
     * los indicados por getLenght
     */
    public byte[] getTrama() {
        return trama;
    }
}

```

```

/**
 * Indica la dirección del esclavo ModBus asociado a este paquete.
 * @param addr
 */
public void setAddress(int addr) {
    trama[0]=toByte(addr);
}

/**
 * Obtiene la dirección del esclavo ModBus asociado a este paquete.
 * @return Entero conteniendo la dirección.
 */
public int getAddress() {
    return toInt(trama[0]);
}

/**
 * Indica el código de función asociado al paquete.
 * @param function
 */
public void setFunction(int function) {
    trama[1]=toByte(function);
}

/**
 * Obtiene el código de función asociado al paquete.
 * So este código es negativo, quiere decir que este paquete
 * es un código de error ModBus.
 * @return Entero con la función ModBus.
 */
public int getFunction() {
    return toInt(trama[1]);
}

/**
 * Permite escribir un elemento de la trama.
 * Sirve para crear el paquete a bajo nivel.
 * @param address
 * @param value
 */
public void setElement(int address, int value) {
    trama[address]=toByte(value);
}

/**
 * Permite leer un elemento de la trama.
 * Sirve para interpretar a bajo nivel la trama.
 * @param address
 * @return Un entero conteniendo una dirección del módulo esclavo.
 */
public int getElement(int address) {
    return toInt(trama[address]);
}

/**
 * Define la longitud del paquete.
 * Esta longitud incluye direccion, funcion, datos y crc.
 * @param l
 */
public void setLenght(int l) {
    lenght=l;
}

/**
 * Obtiene la longitud del paquete.
 * @return Entero con la longitud del paquete.
 */
public int getLenght() {
    return lenght;
}

/**
 * Calcula el CRC del paquete.
 */

```

```

public void calculateCRC() {
    CRC16(trama,lenght-2);
    trama[lenght-2]=toByte(uchCRCLo);
    trama[lenght-1]=toByte(uchCRCHi);
}

/**
 * Nos permite conocer si el CRC del paquete es correcto
 * @return Devuelve true si el paquete es correcto,
 * false en caso contrario.
 */
public boolean isCRCOK() {
    if(CRC16(trama,lenght)==0) {
        return true;
    }
    else {
        return false;
    }
}

/**
 * Obtiene el contenido del CRC
 * @return Devuelve un entero con el CRC
 */
public int getCRC() {
    return uchCRCHi*256+uchCRCLo;
}

/**
 * Convierte un entero de 0 a 255 en un byte con signo
 * @param i Valor entero a codificar como byte, de 0 a 255
 * @return Devuelve byte codificado de -128 a 127
 */
static private byte toByte(int i) {
    byte resultado;
    if (i<128) {
        resultado=(byte)i;
    }
    else if (i<256){
        resultado=(byte) (i-256);
    }
    else {
        resultado=0;
    }
    return resultado;
}

/**
 * Convierte un byte a un entero.
 * @param b
 * @return
 */
static private int toInt(byte b) {
    int resultado;

    if (b>=0) {
        resultado=b;
    }
    else {
        resultado=256+b;
    }
    return resultado;
}

//Tabla CRC bytes altos

private final static int [] auchCRCHi = {
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,

```

```

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
} ;

//Tabla CRC bytes bajos
private final static int [] auchCRCLo = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
};

/*
 * Inicializa CRC
 */
private void CRCInit() {
    uchCRCHi = 0xFF ; //Byte alto del CRC
    uchCRCLo = 0xFF ; //Byte bajo del CRC
}

/*
 * Añade un byte al calculo de crc
 */
private void CRC16l(int c) {
    int uIndex ; //El indice en la tabla CRC

    uIndex = uchCRCLo ^ c ; //Calcula el indice del CRC
    uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex] ; //Byte bajo del CRC
    uchCRCHi = auchCRCLo[uIndex] ; //Byte alto del CRC
}

/**
 * Obtiene el resultado del CRC
 * @return
 */
private int obtieneCRC() {
    return uchCRCHi * 256 | uchCRCLo ;
}

//
// Calcula el CRC del ModBus
//
// puchMsg <- Puntero al mensaje a calcular CRC
// usDataLen <- Numero de bytes a calcular
//
// CRC16 -> el crc calculado
//
private int CRC16 ( byte [] puchMsg, int usDataLen ) {
    int x ;

    CRCInit();

    //Muestrea todo el mensaje

```

;

References

```
Module modulo_crc_ModBus

Public crc_table() As Byte =
{&H0, &H1, &H81, &H40, &H1, &H0, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H0, &HC1, &H81, &H40, &H1, &HC0, &H80, &H41,
&H1, &HC0, &H80, &H41, &H0, &HC1, &H81, &H40,
&H0, &HC0, &HC1, &H1, &HC3, &H3, &H2, &HC2,
&HC6, &H6, &H7, &HC7, &H5, &HC5, &HC4, &H4,
&HCC, &HC, &HD, &HCD, &HF, &HCF, &HCE, &HE,
&HA, &HCA, &HCB, &HB, &HC9, &H9, &H8, &HC8,
&HD8, &H18, &H19, &HD9, &H1B, &HDB, &HDA, &H1A,
&H1E, &HDE, &HDF, &H1F, &HDD, &H1D, &H1C, &HDC,
&H14, &HD4, &HD5, &H15, &HD7, &H17, &H16, &HD6,
&HD2, &H12, &H13, &HD3, &H11, &HD1, &HD0, &H10,
&HF0, &H30, &H31, &HF1, &H33, &HF3, &HF2, &H32,
&H36, &HF6, &HF7, &H37, &HF5, &H35, &H34, &HF4,
&H3C, &HFC, &HFD, &H3D, &HFF, &H3F, &H3E, &HFE,
&HFA, &H3A, &H3B, &HFB, &H39, &HF9, &HF8, &H38,
&H28, &HE8, &HE9, &H29, &HEB, &H2B, &H2A, &HEA,
&HEE, &H2E, &H2F, &HEF, &H2D, &HED, &HEC, &H2C,
&HE4, &H24, &H25, &HE5, &H27, &HE7, &HE6, &H26,
&H22, &HE2, &HE3, &H23, &HE1, &H21, &H20, &HE0,
&HA0, &H60, &H61, &HA1, &H63, &HA3, &HA2, &H62,
&H66, &HA6, &HA7, &H67, &HA5, &H65, &H64, &HA4,
&H6C, &HAC, &HAD, &H6D, &HAF, &H6F, &H6E, &HAE,
&HAA, &H6A, &H6B, &HAB, &H69, &HA9, &HA8, &H68,
&H78, &HB8, &HB9, &H79, &HBB, &H7B, &H7A, &HBA,
&HBE, &H7E, &H7F, &HBF, &H7D, &HBD, &HBC, &H7C,
&HB4, &H74, &H75, &HB5, &H77, &HB7, &HB6, &H76,
```

```

&H72, &HB2, &HB3, &H73, &HB1, &H71, &H70, &HB0, _
&H50, &H90, &H91, &H51, &H93, &H53, &H52, &H92, _
&H96, &H56, &H57, &H97, &H55, &H95, &H94, &H54, _
&H9C, &H5C, &H5D, &H9D, &H5F, &H9F, &H9E, &H5E, _
&H5A, &H9A, &H9B, &H5B, &H99, &H59, &H58, &H98, _
&H88, &H48, &H49, &H89, &H4B, &H8B, &H8A, &H4A, _
&H4E, &H8E, &H8F, &H4F, &H8D, &H4D, &H4C, &H8C, _
&H44, &H84, &H85, &H45, &H87, &H47, &H46, &H86, _
&H82, &H42, &H43, &H83, &H41, &H81, &H80, &H40}

```

```
Public Function crc_16(ByVal ModBusframe() As Byte, ByVal length As Integer) As Integer
```

```

Dim i As Integer
Dim index As Integer
Dim crc_Low As Integer = &HFF
Dim crc_High As Integer = &HFF

For i = 0 To length - 1
    index = (crc_High Xor CByte(ModBusframe(i) And 255))
    crc_High = (crc_Low Xor CByte(crc_table(index) And 255))
    crc_Low = CByte(crc_table(index + 256) And 255)
Next

```

```

Return ((crc_High * 256) + crc_Low)
End Function

```

```

Public Sub main()
    'paquete a enviar
    Dim d1() As Byte

    'Añadir CRC
    Dim crc As Integer = crc_16(d1, d1.Length)

    ReDim Preserve d1(UBound(d1) + 2)
    d1(UBound(d1) - 1) = crc \ 256
    d1(UBound(d1)) = crc Mod 256
End Sub

```

```
End Module
```