

Capítulo 7 PROTOCOLO DE TRANSMISIÓN X-10

En primer lugar, aclaremos los siguientes términos:

- Cuando usamos “X-10”, con guión, nos estamos refiriendo al protocolo, no a la compañía.
- Cuando usamos “X10”, sin guión, nos referimos a la compañía.

Existe un gran malentendido con estos términos, ya que la compañía X-10 cambió su nombre por X10. No se saben a ciencia cierta los motivos, pero es curioso que al buscar en Internet el nombre X-10, con guión, es lo mismo que buscar “X 10”, entre comillas. Por tanto, cuando intentabas buscar información relacionada con la compañía X-10, involuntariamente estábamos haciendo referencia a cualquier sitio Web que contuviera una X y un 10, seguidos. Esto ocasionaba resultados poco deseables.

La información ofrecida a continuación fue originalmente extraída de la página Web de X-10, correspondiente al documento del 25 de diciembre de 1996.

Sin embargo, dado que el protocolo X-10 es un estándar de facto, son muchas las versiones que circulan por la red. Esta versión ha sido modificada en varias ocasiones, recogidas con detalle en el documento alojado en <http://heyu.tanj.com/docs/>.

Por otra parte, ha sido añadida información acerca del protocolo extendido gracias a las contribuciones de varios artículos técnicos que podemos encontrar en <http://www.hometoys.com>.

7.1 CÓDIGO DE CASA Y CÓDIGO DE DISPOSITIVO

Los distintos códigos asociados a los códigos de casa y de los códigos de unidad se muestran en la siguiente tabla:

CÓDIGO DE CASA (HOUSECODE)	CÓDIGO DE DISPOSITIVO (DEVICE CODE)	VALOR BINARIO	VALOR HEXADECIMAL
A	1	0110	6
B	2	1110	E
C	3	0010	2
D	4	1010	A
E	5	0001	1
F	6	1001	9
G	7	0101	5
H	8	1101	D
I	9	0111	7
J	10	1111	F
K	11	0011	3
L	12	1011	B
M	13	0000	0
N	14	1000	8
O	15	0100	4
P	16	1100	C

TABLA III: Códigos de casa y dispositivo

7.2 CÓDIGOS DE FUNCIÓN

FUNCIÓN	VALOR BINARIO	VALOR HEXADECIMAL
All Units Off (Todas las unidades apagadas)	0000	0
All Lights On (Todas las luces encendidas)	0001	1
On (Encender)	0010	2
Off (Apagar)	0011	3
Dim (Atenuar intensidad)	0100	4
Bright (Aumentar intensidad)	0101	5
All lights off (Apagar todas las luces)[1]	0110	6
Extended code (Código Extendido) [2]	0111	7
Hail request (Petición de Saludo) [3]	1000	8
Hail Acknowledge(Aceptación de Saludo)	1001	9
Pre-set Dim(1) (Atenuación Preestablecida) [4]	1010	A
Pre-set Dim(2)	1011	B
Extended Data Transfer (Datos Extendidos (analógico)) [5]	1100	C
Status On (Estado=on)	1101	D
Status Off(Estado=off)	1110	E
Status Request (Petición de Estado) [6]	1111	F

TABLA IV: Códigos de función

[1] El comando “All Lights On” fue implementado en los módulos X-10 desde el principio, sin embargo “All Lights Off” apareció bastante más tarde. Por eso los módulos de lámpara más antiguos y los interruptores de pared no soportan este comando. Pero por ejemplo, el módulo de lámpara **LM14A** sí que responde a esta función.

[2] Ver nota en protocolo extendido, apartado 7.5

[3] La Petición de Saludo se transmite para ver si existen otros transmisores X-10 dentro del rango de escucha. Esto permite al O.E.M. (Original Equipment Manufacturer) asignar un Código de Casa diferente si se recibe un mensaje de Aceptación de Saludo.

[4] En una instrucción de Atenuación Preestablecida (1 0 1 x 1), el Bit D8 representa el Bit más significativo del nivel. H1, H2, H4 y H8, representan los Bits menos significativos. Estas órdenes no están aceptadas por la compañía X-10, sin embargo se siguen usando actualmente. Una explicación más detallada la encontraremos en el punto 7.4.

[5] El código de Datos Extendidos se sigue de bytes que pueden representar información analógica (después de una conversión A/D). No debe haber separación entre los bytes de datos, ni entre el código de datos extendidos y datos reales. El primer byte se puede utilizar para indicar cuántos bytes de información le seguirán. Si se dejan separaciones entre los bytes de información, estos códigos podrán ser recibidos por Módulos X-10 causando un funcionamiento erróneo.

7.3 PARÁMETROS PUERTO SERIE

Los parámetros para las comunicaciones entre la interfaz serie y el ordenador son las siguientes:

Tasa de baudios	4800 bps
Paridad	Ninguna
Bits de datos	8
Bits de stop	1

TABLA V: Parámetros del puerto serie

El CM11 se une con una conexión RJ11 en un extremo y con un conector de 9 pins tipo D con el puerto serie del ordenador.

SEÑAL	CONECTOR DB9	CONECTOR RJ11
SIN	Pin 2	Pin 1
SOUT	Pin 3	Pin 3
GND	Pin 5	Pin 4
RI	Pin 9	Pin 2

TABLA VI: Conexión del CM11

Donde

- SIN** Serial input to PC (Entrada serie al PC)
- SOUT** Serial Output from PC (Salida del PC por el puerto serie)
- GND** Signal Ground (Señal de tierra)
- RI** Ring Signal (Señal de entrada al PC)

7.4 TRANSMISIÓN X-10 ESTÁNDAR PC-INTERFAZ

Una transmisión X-10 desde el ordenador al interfaz serie típicamente consiste en el envío de un Código de Casa seguido de un Código de Unidad o un código de Función.

Un pequeño inciso; la palabra estándar no nos debe llevar a confusiones, ya que no se refiere a que sea una transmisión estandarizada por algún miembro regulador, sino que se refiere a que es una transmisión básica, en contra de las transmisiones extendidas.

El formato de estas transmisiones es:

NÚMERO DE BYTES	PC	INTERFAZ
2 bytes	Header:Code	
1byte		Cchecksum
1byte	Acknowledge	
1byte		Interfaz preparada para recibir

TABLA VII: Transmisión estándar PC-Interfaz

Este es el formato típico de todas las comunicaciones entre el PC y el interfaz, exceptuando la primera transmisión desde el PC.

- **HEADER: CODE**

Es una combinación de 2 bytes, donde el primero de ellos corresponde al siguiente formato:

BYTE HEADER

7 6 5 4 3	2	1	0
Number of dims	1	F/A	E/S

Donde:

- NUMBER OF DIMS es un valor entre 0 y 22 identificando el número de dims a transmitir. (22 es equivalente al 100%)
- BIT 2 está siempre al valor 1 para asegurarnos de que la interfaz mantenga el sincronismo.
- F/A indica si el siguiente bit se refiere a una función (valor 1) o a una dirección (valor 0).
- E/S indica si el siguiente byte es una transmisión extendida (1) o una transmisión estándar (0).

BYTE CODE

Según sea una dirección o una función, podemos interpretar la información de este byte de dos maneras distintas:

7 6 5 4	3 2 1 0
Housecode (Código de casa)	Device code (Código de dispositivo)
Housecode (Código de casa)	Función

Recordemos que la función sólo opera en las unidades direccionadas con el mismo código de casa. Una vez que un dispositivo ha sido direccionado, dicho dispositivo responderá a cualquier orden recibida por la red.

- **INTERFACE CHECKSUM AND PC ACKNOWLEDGE (SUMA DE VERIFICACIÓN Y ASENTIMIENTO DEL PC)**

Cuando la interfaz recibe una transmisión procedente del PC, sumará todos los bytes, y luego devolverá un byte de checksum, es decir de suma de verificación. Si el byte de checksum es correcto, el PC debería devolver a la interfaz el valor 0x00 para indicar que la transmisión está realizándose correctamente y debe continuar.

- **INTERFAZ PREPARADA PARA RECIBIR (READY)**

Una vez que la transmisión de X-10 se ha establecido (lo cual puede llevar bastante tiempo en el caso de las funciones Dim y Bright), la interfaz mandará 0x55 al PC para indicar que está preparado para recibir (ready state).

- **PRESET DIM**

Dedicamos un capítulo especial a estas funciones ya que son causa de gran confusión por parte de los usuarios.

El protocolo original X-10 estaba formado por 16 comandos (listados en orden binario):

All Units Off, All Lights On, On, Off, Dim, Bright, All Lights Off, Extended Code, Hail Request, Hail Acknowledge, Preset Dim (1), Preset Dim (2), Unused, Status On, Status Off, y Status Request.

Posteriormente, la compañía X10 decidió modificar dicha lista. Fueron cuatro los comandos que cambiaron, dos de ellos fueron de particular interés, ya que eran comandos que los propios miembros de X10 no utilizaron. Esto no quiere decir que estos comandos no fueran usados nunca, ya que otras compañías sí que lo hicieron. Dichos comandos eran “Preset Dim (1)” y “Preset Dim (2)”.

Desde que X10 estableció el protocolo, la propia compañía tomó la decisión de abandonar esos dos comandos. Sin embargo, debido al hecho de que el protocolo X-10 no está regulado (fue publicado pero las patentes expiraron y no hay ningún miembro regulador que establezca el orden), estos antiguos códigos siguen siendo usados por muchos productos de otras compañías, contrariamente a los deseos de X10. (Por ejemplo, Switchlinc y PCs)

La idea de la orden “PRESET DIM” era de que un transmisor X-10 (un controlador, por ejemplo) enviara una dirección y un comando que provocara que el receptor se estableciera directamente en uno de los 32 diferentes niveles, sin importar el estado en que estuviera. Así que “Preset Dim” se puede asociar al nombre “Direct Dim”.

Pero, ¿qué fue lo que impulsó a abandonar dichas funciones? Uno de los principales motivos fue que difería en formato con el resto de las

órdenes. Al contrario que los demás comandos X-10, la orden “PRESET DIM” no incluía el código de casa en los bytes enviados a la línea eléctrica. El formato de la orden consistía en que los 4 primeros bits –que en cualquier orden sería el código de casa-, era el nivel de dim.

Por ello, este método nunca fue usado por la compañía X10 y ha sido eliminado de las especificaciones de su protocolo, aunque existen todavía empresas que aceptan esos comandos.

- **DIM O BRIGHT**

Existen dos maneras de considerar estas funciones, la absoluta y la relativa. La relativa se refiere a que la cantidad que tú indicas que varíe la intensidad de tu dispositivo es relativa al valor actual que dicho dispositivo posee. Por ejemplo, en el caso de que una luz esté al 80%, y le mandamos un comando relativo de dim al 20%, el resultado será que la lámpara se situará aproximadamente al 60%.

En el caso del dim absoluto, el valor del dim indica el valor exacto al que debe situarse la intensidad de la lámpara en cuestión. En el mismo caso anterior, si la lámpara estaba al 80% y el valor del dim absoluto es del 20%, el resultado final será que la lámpara estará al 20% de su intensidad.

Pero dado que muchos de los dispositivos X-10 no soportan el caso del dim absoluto, es a menudo necesario enviar un comando que establezca la luz a un nivel determinado y posteriormente envíe el nivel de dim deseado. Por ejemplo, enviar varios comandos de bright hasta que la intensidad se establezca al 100% antes de enviar los comandos de dim.

En la orden básica de dim, el valor de dim seleccionado es relativo, mientras que en la macro es posible optar por el dim absoluto o relativo, según prefiera el usuario.

- **EJEMPLO DE TRANSMISIÓN**

PC	INTERFAZ	DESCRIPCIÓN
0x04,0x06		Dirección A1
	0x6a	Checksum ((0x04+0x66))&0xff)
0x00		PC preparado para transmitir
	0x55	Interfaz preparada
0x04,0x6e		Dirección A2
	0x72	Checksum ((0x04+0x6e))&0xff)
0x00		PC preparado para transmitir
	0x55	Interfaz preparada

0x86,0x64		Función: A Dim 16/22+100%
	0xe0	Checksum incorrecto
0x86,0x64		Retransmisión de la función
	0xea	Checksum ((0x86+0x64)&0xff)
0x00		PC preparado para transmitir
	0x55	Interfaz preparada

TABLA VIII: Transmisión estándar PC-Interfaz

Este ejemplo nos muestra una transmisión en la que hacemos referencia a las direcciones de dos módulos, A1 y A2, y luego envía la función Dim para atenuar las luces un 72%.

Hemos de notar que el direccionamiento múltiple no puede hacerse con varios códigos de casa, por ejemplo, la orden A1 B2 DIM 72% no será válida para atenuar las lámparas A1 y B2, sino que sólo se reflejará la acción en la unidad B2.

7.5 STATUS

Los comandos relacionados con el estado de los dispositivos son "status", "status on", y "status off". Estos comandos solamente son soportados por los módulos bidireccionales, como por ejemplo el módulo de lámpara LM14A. El módulo estándar de lámpara, LM465, no responde a la petición de estado.

7.6 TRANSMISIÓN EXTENDIDA X-10

Con la publicación del protocolo de código extendido en <ftp://ftp.x10.com/pub/manuals/xtc798.doc>, la compañía X10 publicó que habría tres diferentes "extended codes":

- Extended code 1 (para datos o control)
- Extended code 2 (for meter read -lectura de contador- & dsm, Distributed Shared Memory)
- Extended code 3 (para mensajes de seguridad)

Al establecerse las especificaciones del protocolo extendido, la lista de los antiguos comandos se vio modificada un poco, en concreto los comandos siguientes:

- Extended Code (0111) Ahora denominado “Extended Code 1”, para datos y control.
- Preset Dim 1 (1010) Ahora diseñado como “Extended Code 3”, para mensajes de seguridad.
- Preset Dim 2 (1011) Sin utilizar.
- Extended Data (1100) Denominado “Extended Code 2” (meter read y dsm).

En el apartado anterior ya hablamos de las razones por las que los comandos de preset dim desaparecieron. Veamos ahora las razones por las que a los otros dos comandos les sucedió lo mismo.

En primer lugar, hagamos referencia al antiguo comando “**Extended Data**”. La documentación de X-10 estableció que este comando fuera seguido por 8 bytes que podían representar datos analógicos. Asimismo, establecía que el primer byte podía usarse para indicar cuántos bytes de datos se transmitirían a continuación.

Esto significaba que, dependiendo de la situación, la trama básica de datos (11periodos), podría ser continuada por un byte más (8 ciclos de la onda senoidal eléctrica), o que ese byte indicara cuántos bytes más seguirían en la transmisión.

Esto implicaba que podrían añadirse a la transmisión, desde 8 hasta 2056 ciclos más, con lo que se convertiría en una trama de datos demasiado larga. Y tomaría 34 segundos o más en enviarla (considerando la transmisión una sola vez).

Esto también suponía un problema para los repetidores en la línea eléctrica, ya que si la trama tiene una longitud variable, es imposible diseñar en la práctica un repetidor adecuado, pues el repetidor no sabe cuándo tiene que comenzar la a repetir la transmisión.

Así que incluso aunque “Extended data” ha dado paso a “**Extended Code 2**”, este problema continúa, ya que la trama de datos no tiene una longitud fija, depende del mensaje en cuestión.

En el caso del antiguo comando “**Extended Code**”, surgía el mismo problema, ya que el número de bytes de la trama de datos estaba indefinido. Afortunadamente, esta orden se convirtió en “**Extended Code 1**”, que sí que posee una trama de datos de longitud determinada, 31 ciclos (62bits) que describimos a continuación:

- Start code=4 bits
- Housecode=4bits
- Extended code 1= 5bits (01111)

- Unit code (device code)=4bits
- Data=8bits
- Command=8bits

“Extended code 3” ha sido asignado a la seguridad, sin embargo todavía no ha sido asignado su formato ni su trama.

Resumiendo, cuando nos refiramos a Extended Code, en realidad nos estamos refiriendo al “Extended Code 1” Para más información, visite el documento X-10 XTC797.

La transmisión extendida es simplemente una extensión del protocolo para permitir el envío de dos bytes adicionales. Veamos en concreto el formato de los bytes:

NÚMERO DE BYTES	PC	INTERFAZ
5bytes		Header:Function:Unitcode:Data:Command
1byte		Checksum
1byte	acknowledge	
1byte		Interfaz preparada para recibir

TABLA IX: Formato de una transmisión extendida

La cabecera (header) de una transmisión extendida es la siguiente:

BIT	7	6	5	4	3	2	1	0
VALOR(0x07)	0	0	0	0	0	1	1	1

- Los bits 7-3 son siempre cero ya que el nivel de dim no es aplicable en las transmisiones extendidas.
- El bit 2 debe ser uno como en todas las cabeceras procedentes del PC (para sincronización).
- El bit 1 está fijado a uno, ya que la transmisión extendida es siempre una función.
- El bit 0 está fijado a 1 para indicar precisamente que es una transmisión extendida (bit E/S=1).

El byte de función es el siguiente:

BIT	7	6	5	4	3	2	1	0
VALOR	HOUSECODE				0	1	1	1

De nuevo, el código de casa debe ser el mismo en cualquier módulo de dirección previo, y para transmisiones extendidas el código de función debe ser 0111.

Finalmente, los bytes de datos y de comandos pueden tomar cualquier valor entre 0x00 y 0xff. Y en este caso el checksum se corresponde con la siguiente expresión:

$$\text{Checksum} = (\text{header} + \text{code} + \text{data} + \text{commanmd}) \& 0xff$$

7.7 RECEPCIÓN EN X-10

Siempre que la interfaz comience a recibir datos desde la línea de corriente (power line), inmediatamente mandará la señal 'serial ring' (RI) para iniciar el proceso de avisar al PC. Una vez que la recepción de datos se ha completado, la interfaz comenzará a sondear al PC para pasarle su buffer de datos (10 bytes como máximo). Si el PC no responde, el búfer de datos de la interfaz desbordará y los nuevos datos que lleguen no serán guardados dentro del buffer.

- **Señal de sondeo de la interfaz (poll signal)**

Para sondear al PC, la interfaz estará continuamente enviando los siguientes bytes:

BIT	7	6	5	4	3	2	1	0
VALOR	0	1	0	1	1	0	1	0

Los bits corresponden al valor **0x5a**. Dicha señal será repetida una vez cada segundo hasta que el PC responda.

- **Respuesta del PC a la señal de sondeo (poll)**

Para finalizar el sondeo por parte de la interfaz al PC e iniciar la transferencia de datos, el PC deberá mandar un asentimiento (ack) a la señal 'poll'. Este asentimiento consta de un byte con los siguientes valores:

BIT	7	6	5	4	3	2	1	0
VALOR	1	1	0	0	0	0	1	1

Dichos bits corresponden al valor en hexadecimal **0xc3**. Hay que tener en cuenta que el bit #2 de la transmisión del PC está a cero, indicando que este byte no puede ser el comienzo de una transmisión por parte del PC.

- **Buffer de datos de recepción del interfaz serie**

El buffer consiste en un conjunto de 10 bytes definidos como sigue:

BYTE	FUNCIÓN
0	Upload Buffer Size (Tamaño del búffer de datos a recibir)
1	Function/Address Mask (Máscara de Función/Dirección)
2	Byte de datos #0
3	Byte de datos #1
4	Byte de datos #2
5	Byte de datos #3
6	Byte de datos #4
7	Byte de datos #5
8	Byte de datos #6
9	Byte de datos #7

TABLA X: Buffer de datos de recepción

El número de bytes que la interfaz recibirá está especificado en el byte 0 de la transmisión, es decir, no se recibirán los 10 bytes por defecto, sino que dependerá de cada caso en cuestión.

El byte 'Function/Address Mask' indica cuáles de los siguientes 8 bytes deben ser interpretados como una función (en cuyo caso el bit correspondiente a la posición de ese byte estará a 1) o como una dirección (en cuyo caso estará a 0).

Los bytes de datos están en el mismo formato que las transmisiones X-10 estudiadas anteriormente (por ejemplo, en un byte encontraremos Código de Casa: Código de Dispositivo o Código de Casa: Código de Función).

Nótese que una vez que el buffer de datos de la interfaz ha sido transferido, no hay ningún asentimiento por parte del PC. Sin embargo esto no supone un problema ya que la recepción en el buffer es simplemente para informar al PC de un estado de la red externo, no se trata de controlar ningún dispositivo, en cuyo caso sí que podría haber problemas.

- **Dim o bright**

Cuando en el buffer de datos recibido por el PC se hace referencia a una orden de DIM o BRIGHT (atenuación o aumento de la intensidad), el PC esperará un byte que le indique el nivel de intensidad, que será un valor de 0 a 22 que estará contenido en un byte. Un módulo X-10 tiene 210 posibles valores discretos de brillo, y por tanto este byte será equivalente a un cambio en el brillo de $n/210*100\%$, donde n es el dato proporcionado en el byte.

- **Código extendido**

El código extendido es procesado de una manera similar a las funciones DIM y BRIGHT, excepto que en el caso del código extendido, el PC esperará 2 bytes, que son los bytes de Datos y de Comando.

Veámoslo con un ejemplo.

PC	INTERFAZ	DESCRIPCIÓN
	0x5a	Señal de sondeo por parte de la interfaz(poll)
0xc3		Respuesta del PC ('PC Ready?')
	0x05	5bytes serán enviados a continuación
	0x04	xxxx x100 → Los bytes 0 y 1 son direcciones, y el byte 2 es función.
	0xe9	B6 Código de casa→B Código de dispositivo→6
	0xe5	B7
	0xe5	B Bright Código de casa→B Función→Bright
	0x58	58/210*100%

TABLA XI: Ejemplo de código extendido

En este ejemplo, primeramente la interfaz requiere la atención del PC para luego indicarle que va a tener lugar una transmisión de 5 bytes, donde los bytes 0 y 1 son direcciones y el byte 2 es una función bright y por tanto, el byte siguiente es el valor del nivel de intensidad.

7.8 FAST MACRO DOWNLOAD (CM10)

Esta sección es solamente aplicable al dispositivo CM10, que precedió al CM11. La interfaz del CM10 posee un buffer de 42 bytes que contiene los códigos de la macro. Debido a un número tan pequeño de bytes, el código de la macro está comprimido agrupando las funciones similares.

Hay que tener en cuenta que cualquier error en los códigos de función puede resultar en introducir a la interfaz en un bucle infinito y bloquearla, así que hemos de asegurarnos que el código es correcto antes de la transmisión.

A diferencia del CM11, el CM10 no posee compartimento alguno donde poder insertarle pilas, por lo que si la interfaz detecta que ha sufrido una situación de corte de la corriente eléctrica, avisará al PC mandándole una señal de sondeo para indicarle que la macro debe ser refrescada.

- **Macro download poll code (CM10)**

Esta señal es principalmente para el CM10, y se refiere a una señal para la descarga de la macro. En ese caso la interfaz enviará continuamente con el objeto de sondear al PC:

POLL(sondeo)	7	6	5	4	3	2	1	0
VALOR 0xa5	1	0	1	0	0	1	0	1

Esta señal será repetida una vez por segundo hasta que el PC responda.

Es cierto que este código de sondeo (0xa5) es enviado por el CM11 cuando las pilas situadas en el mismo detectan una situación de pérdida de corriente, pero la señal se refiere a una directiva de configuración del reloj más que a una señal para descargar una macro. Por ello, en el caso del CM11, la interfaz espera a que el PC le responda con una actualización del reloj (0x9b, ver sección 9).

- **Respuesta del PC al macro download poll code (CM10)**

Para parar la señal de sondeo, el PC debe responder con lo siguiente:

RESPUESTA DEL PC	7	6	5	4	3	2	1	0
VALOR 0xfb	1	1	1	1	1	0	1	1

Una vez que se ha transmitido esto, la macro debes ser inmediatamente transferida. En este punto, la interfaz esperará hasta que los 42 bytes de la macro sean recibidos antes de que ninguna transmisión X-10 pueda tener lugar.

Para resetear el CM10 y borrar el contenido de la interfaz, debemos mandar el byte 0xfb seguido de 42 bytes 0x00.

- **MACRO CODE (CM10)**

El código de la macro está dividido en macros individuales, y grupos funcionales dentro de la macro. El único límite al número de macros y al número de grupos son los bytes disponibles.

Cada macro comienza con un byte de inicio que indica el Código de Casa y el Código de dispositivo que serán los que indiquen al CM10 que comience la macro.

Después del byte de inicio está la longitud de la macro, y el trigger de la función (por ejemplo, las funciones On y Off). El tamaño está determinado en los 7 bits menos significativos, y la función inicial está indicada en el bit más significativo. Si dicho bit está a 1, el trigger de la función es 'On'; y si el bit está a 0, el trigger funcional está a 'Off'.

Como hemos mencionado previamente, la macro está dividida en grupos funcionales, y cada grupo tiene un byte que indica la longitud del grupo, antes de que la macro en sí esté definida. Este byte que indica la longitud es exclusivo del código de la función.

El grupo entonces estará compuesto por un código de casa común (un nibble), seguido de un número determinado de códigos de dispositivo (cada uno de ellos ocupará un nibble), y finalmente un código de función (un nibble).

Todos los bytes sin usar deben tomar el valor 0x00.

- **FUNCIONES DIM Y BRIGHT EN LA MACRO**

Si la función es dim o bright, el siguiente byte indicará el valor -hasta 22 posibles niveles- que deberá ser atenuado o aumentada la intensidad del dispositivo en cuestión. Si el bit más significativo está a uno, la interfaz enviará suficientes comandos de bright hasta que las lámparas asociadas estén al 100%, y luego atenuar la lámpara al valor especificado, con lo que estaríamos en el caso de dim o bright absoluto.

- **CÓDIGO EXTENDIDO EN MACROS.**

El código extendido no puede ser agrupado como cualquier otra función; consecuentemente el grupo del código extendido será definido como sigue:

BYTE	DESCRIPCIÓN
0x01	Longitud del grupo
0xa7	Código de Casa D (1010=D), Función de código extendido (extended code)
0x03	Código de dispositivo 11 (0011=11)
0xff	Byte de datos 0xff
0x55	Byte de comando 0x55

TABLA XII: Código extendido en macro (CP10)

➤ CHECKSUM

Una vez que la macro ha sido enviada a la interfaz, ésta calculará un byte de checksum sumando los 42 bytes del código de la macro (sin incluir el byte de comienzo de descarga de la macro).

Si el valor es incorrecto, el PC deberá iniciar de nuevo la descarga de la macro transmitiendo el byte de inicio de la descarga de la macro.

➤ EJEMPLO

PC	INTERFAZ	DESCRIPCIÓN
	0xa5	Power-fail, macro poll
0xfb		Byte de inicio de la descarga de la macro
0x26		Iniciador C1
0x0a		Trigger Funcional OFF, longitud de la macro:10 bytes
0x04		Longitud del grupo: 4 nibbles
0x66		Macro Housecode A, dispositivo 1
0x2e		Dispositivos 2 y 3
0x04		Dim
0x0b		Dim $11/22*100\%=50\%$
0x02		Longitud del grupo: 2 nibbles
0x6a		Macro Housecode, A, dispositivo 4
0x02		Función:On
0x26		Iniciador C1
0x8c		Trigger Funcional On, Longitud de la macro 12 bytes
0x02		Longitud del grupo: 2 nibbles
0x66		Macro Housecode A, dispositivo 1
0x02		Función On
0x03		Longitud del grupo: 3 nibbles
0x6e		Macro housecode A, dispositivo 2

0x42		Dispositivo 3, función Dim (0100)
0x06		Dim 6/22*100%=27%
0x02		Logitud del grupo: 2 nibbles
0x6a		Macro housecode A, dispositivo 4
0x03		Función Off
0x00....		20 bytes a 0x00
	0x91	Checksum de la maro: 0x91
0x00		Checksum correcto
	0x55	Interfaz lista

TABLA XIII: Ejemplo de macro en CP10

7.9 CÓDIGO DE LA EEPROM (CM11 Y CP10)

Generalmente, cuando enviamos una orden X-10 del tipo A1 ON, esta orden hace efecto en el dispositivo A1, encendiéndolo. Sin embargo, el controlador interfaz CM11 puede ser programado para que al interceptar un determinado comando responda generando un conjunto de órdenes denominadas macro. Por ejemplo, al recibir la orden A1 ON, el CM11 podría encender las luces de la escalera y de las habitaciones, y luego, veinte minutos más tarde, atenuar la intensidad de las luces un 50% y apagar las luces de la escalera. Finalmente, apagar todas las luces treinta minutos más tarde.

Asimismo existe otra posibilidad, en la que el usuario programa una macro para que sea ejecutada en la fecha y hora que el usuario decida. Esta opción se denomina programación de timers, donde a cada timer se le asocia una o varias órdenes.

El código de la EEPROM para el CM11 y el CP10 contiene tanto los timers descargados de la interfaz (que indican la hora a la que deben realizarse las acciones de la macro) como iniciadores de cada macro.

En otras palabras, tanto los timers como los iniciadores apuntan a su macro correspondiente. La tabla de timers se chequea cada minuto para ver si alguna de las entradas debería iniciar la ejecución de una macro. Por otra parte, los iniciadores de la macro son chequeados en cualquier momento que una señal X10 sea detectada en la línea eléctrica.

Podemos dividir la EEPROM del CM11 en varias partes:

- *Macro Offset* (Offset de los iniciadores de la macro) (2bytes)
- *Timer Initiators* (Iniciadores de los timers)

- *Timer end delimiter* (0xff) (Delimitador tabla de timer)
- *Macro Initiators* (Iniciadores de la macro) (Comienzan en el byte indicado por el macro offset)
- *Initiators end delimiter* (0xff,0xff) (Delimitador tabla de iniciadores)
- *Macro Data* (Datos de la macro)

- **MACRO OFFSET**

Los iniciadores de la macro sólo son procesados cuando se recibe un dato por la línea eléctrica, y se compara ambos valores y en el caso de que coincidan se produce la ejecución de la macro. Es por ello necesaria la existencia de un offset, ya que hay que indicarle al CM11 el rango de elementos dentro de la tabla con los que tiene que hacer la comparación. Los primeros 2 bytes de la EEPROM contienen un offset para la tabla de iniciadores de la macro, que indican el byte de la EEPROM a partir del cual se sitúan dichos iniciadores.

- **INICIADOR DE TIEMPO (TIMER INITIATOR)**

Los timers se encuentran a partir de la dirección 0x0002 en la EEPROM y están delimitados por 0xff al final de la tabla. El offset de cada timer es relativo al valor del timer que lo precede.

Cada entrada de un timer son 9 bytes que contiene los siguientes datos:

RANGO DE BITS	DESCRIPCIÓN
71	Reservados
70-64	Máscara del día (bit1=domingo, bit7=sábado)
63-56	Día del año de inicio(bits 0 a 7)
55-48	Día del año de fin (bits 0 a 7)
47-44	Event start time x 120 minutes
43-40	Event stop time x 120 minutes
39	Día del año de inicio (bit 8)
38-32	Event start time (0 to 120 minutes, bits 0 to 6)
31	Día del año de fin (bit 8)
30-24	Event stop time (0 to 120 minutes, bits 0 to 6)
23-20	Start event macro pointer (bits 8 to 11)
19-16	Stop event macro pointer (bits 8 to 11)
15-8	Start event macro pointer (bits 0 to 7)
7-0	Stop event macro pointer (bits 0 to 7)

TABLA XIV: Timer iniciador

El día del año, tanto de comienzo como de fin se representa con 9bits, con rango de 0-365.

Para que una macro comience su ejecución debe coincidir el día de la semana y el día del año del timer de la macro con la fecha actual.

Si el bit de modo de seguridad está activado, el CM11 añadirá un valor de tiempo comprendido entre 0 y 60 minutos al tiempo del evento.

El puntero de evento de la macro tiene la dirección de la macro que será ejecutada cuando dicho evento sea ejecutado. Existen tanto un puntero al evento a ejecutar a la hora de inicio como otro puntero al evento a ejecutar a la hora seleccionada de finalización.

Si el usuario no desea programar ningún evento asociado al timer de finalización, lo que debe hacer es poner a uno todos los bits de la hora de stop, de modo que la hora de stop nunca es alcanzada. Por otra parte, si las horas de inicio y de fin de la macro coinciden, el CM11 opta por desechar el evento de finalización, sólo se ejecutará el de inicio.

- **INICIADOR DE LA MACRO**

Los iniciadores de la macro (3 bytes) están configurados tal como sigue:

RANGO DE BITS	DESCRIPCIÓN
23-20	Código de casa iniciador
19-16	Código de dispositivo iniciador
15	función iniciadora ('1'= on, '0'=off)
14-12	Reservados
11-0	Puntero a la macro (bits 0 a 11) Macro pointer

TABLA XV: Iniciadores de la macro

La tabla de los iniciadores de la macro termina con dos bytes 0xff.

- **DATOS DE LA MACRO**

Los datos de la macro comienzan con un offset del timer **TIMER OFFSET**, en un rango desde 0 (espera nula) hasta 240 (4 horas), valor relativo al valor del timer anterior. Posteriormente al timer offset se encuentra el número de elementos dentro de la macro (desde 1 a 255). Y finalmente se encuentran los elementos de la macro en sí mismos.

Es decir, podemos hacer una división en paquetes de datos, que estarán formados por:

Paquete= retraso : número de elementos : elementos de la macro(datos)

- o El campo **número de elementos** se utiliza para la compresión de datos, evitando el duplicado de funciones asociadas al mismo código de casa. De este modo podemos ahorrar algunos bytes de EEPROM, algo muy positivo teniendo en cuenta el tamaño de la memoria.

Veamos un ejemplo:

A1 ON; B1 ON;A2 ON;B3 ON ---> A1,2 ON; B1,3 ON

Pero en el caso de que programemos una macro como esta:

A1 ON; A1 OFF; A1 ON; A1 OFF;A1 ON; A1 OFF...

De modo que lo que esperemos sea encender y apagar la luz A1 repetidas veces, no podemos comprimir la macro, ya que el resultado final es:

A1 ON, OFF

Con lo que la luz se encendería y apagaría la luz una sola vez.

Las órdenes como DIM o BRIGHT sólo son agrupadas si tienen el mismo valor de dim.

- o Los **elementos de la macro** están configurados como sigue:

RANGO DE BITS	DESCRIPCIÓN
23-20	Comando de código de casa
19-16	Comando de función
15-0	X10 format device bitmap (2bytes) Mapa de bits del dispositivo en formato X10

TABLA XVI: Instrucción básica (3bytes)

La tabla de mapeo de los dispositivos X-10 es la siguiente:

High byte	10	2	8	16	12	4	6	14
Low byte	9	1	7	15	11	3	5	13

Cada bit se corresponde con un dispositivo, por ejemplo, 0x0001 se corresponde con el dispositivo 13.

RANGO DE BITS	DESCRIPCIÓN
31-28	Comando de código de casa
27-24	Comando de función
23-8	X10 format device bitmap
7	Brighten first ('1) or simply dim ('0)
6-5	Reservados
4-0	Valor de dim (del 0 al 22)

TABLA XVII: Instrucciones bright o dim (4bytes)

Si el bit 7 está activado, la intensidad de la lámpara se establece al 100% y posteriormente se efectúa la operación de dim.

RANGO DE BITS	DESCRIPCIÓN
47-44	Comando de código de casa
43-40	Comando de función
39-24	X10 format device bitmap
23-0	Datos (Código extendido)

TABLA XVIII: Instrucciones de datos extendidos

La tabla de arriba debería haberse titulado “Instrucciones de Código Extendido”, en lugar de “Instrucciones de Datos Extendidos”.

La instrucción de “Transferencia de Datos Extendidos” (0x0c) es sólo un elemento de 3 bytes de la macro básica.

Los elementos de la macro para la Instrucción de Código Extendido son programados de la siguiente manera:

RANGO DE BITS	DESCRIPCIÓN
47-44	Comando de código de casa
43-40	Comando de función (0x7)
39-24	X10 format device bitmap
23-16	Código de dispositivo(en el nibble menos significativo)
15-8	Byte de datos
7-0	Extended type command function

TABLA XIX: Instrucciones de código extendido

Los comandos de Código Extendido tienen significado en módulos tales como el módulo de lámpara LM14A two-way lamp. Para una

descripción más detallada de de las funciones para el código extendido, visitar el documento de X-10 *XTC798.doc*.

Establecer el bitmap de los dispositivos a cero suprimiría la transmisión de byte de dirección Código de Casa|Código de dispositivo en aquellas instrucciones en las que este byte es superfluo, como por ejemplo, en la instrucción "All Lights On" (todas las luces encendidas) y en la mayoría de los comandos de Código Extendido. Sin embargo, ActiveHome establece el bitmap a 0x0001, que corresponde al dispositivo 13.

- **MACROS ENCADENADAS**

Si en la memoria EEPROM una macro viene inmediatamente seguida por una o más macros con un offset de tiempo no nulo, la ejecución de la primera macro activará las segunda, y así sucesivamente, respetando los retrasos, que vienen referidos al offset de tiempo anterior.

Para evitar este comportamiento, y para no arriesgarnos a que se produzca un fortuito encadenamiento, la solución pasa por insertar un byte 0x00 entre diferentes macros, de modo que aunque estemos consumiendo parte de la preciada memoria de la EEPROM, es un precio a pagar para evitar la ejecución de órdenes sin sentido.

- **TRANSFERENCIA DE DATOS A LA EEPROM**

Los datos de la EEPROM son transferidos a la interfaz en bloques de 19 bytes. El primer byte es el comando de iniciación de la transferencia de la macro (0xfb). Le siguen 2 bytes que contienen la dirección de la EEPROM en la que se deben almacenar los datos. A la dirección de la EEPROM le siguen los 16 bytes de datos. Por otra parte, el CM11 usa los dos primeros bytes de la EEPROM para indicar el offset de los iniciadores de la macro.

Una vez que la interfaz ha recibido los datos de la EEPROM, devolverá un checksum. Si el checksum es correcto, el PC asentirá (0x00), y después de que los datos hayan sido programados en la EEPROM, la interfaz devolverá una señal de que 'Interfaz Ready' (0x55) indicando que ya está disponible para procesar nuevas peticiones del PC.

- **EJEMPLO**

PC	INTERFAZ	DESCRIPCIÓN
0xfb		Byte de inicio de la transferencia (Primer bloque de datos)
0x00 0x00		Dirección de la EEPROM 0x0000 (byte menos significativo) (byte más significativo)
0x00 0x0c		Offset para los iniciadores de la macro 0x000c Comienza en el byte 12
0x3e		Máscara del día de la semana cuando se ejecuta el evento 0111110 (.LMXJV.)
0x00		[0..7] Día del año de inicio
0x6d		[0..7] Día del año final
0x49		Minuto de comienzo del evento de inicio x120 minutos Minuto de comienzo del evento de fin x120 minutos
0x00		[bit 8] Día del año de inicio [0..6] Hora del evento de inicio
0x80		[bit 8] Día del año final [0..6] Hora del evento de fin
0x00		[bits 8..11] Puntero al evento de inicio de la macro [bits 8..11] Puntero al evento de fin de la macro
0x1d		[0..7] Puntero al evento de inicio de la macro
0x22		[0..7] Puntero al evento de fin de la macro
		EN RESUMEN: Día de comienzo: 0x000 (1 de enero) Día de fin: 0x16d (día 365=31 de diciembre) Hora de inicio: 4x120 minutos= 08:00 Hora de fin: 9x120 minutos= 18:00 Puntero del evento de inicio de la macro: 0x01d (byte29) Puntero del evento de fin de la macro: 0x022 (byte34)
0xff		Delimitador de la zona de timers
0x6a		Código de casa y código de dispositivo iniciador de la macro (A4)
0x80		[4..7] Función iniciadora (ON) [0..3] Corresponden a los Bits [8..11] del puntero de la macro
0x11		Corresponde a los bits [0..7] del puntero de la Macro En total, (0x011) (Apunta al byte17)
0xff		
	0xb8	Checksum procedente de la interfaz
0x00		Checksum correcto
	0x55	Programación completa
0xfb		Segundo bloque de datos

0x00 0x10		Dirección de inicio de la EEPROM 0x0010 (EN EL byte 16)
0xff		(BYTE16)Delimitador de la tabla de iniciadores de la macro
0x00		Macro correspondiente a la orden A4 ON Instante de la macro 0 (offset 0 respecto al inicio de la macro)
0x01		1 elemento en la macro
0x64		Código de casa A, función DIM
0x00 0x40		Bitmap: dispositivo #1
0x0b		Dim level 11/22=50%
0x0f		Macro:delayed by 15 minutes (MACRO ENCADENADA)
0x01		1 element
0x64		Housecode a, function dim
0x00 0x40		Bitmap:device #1
0x80		Nivel de bright hasta el 100%
0x00		BYTE 29. Evento de inicio del timer Macro:Instant
0x01		1 elemento
0x62		Código de casa A, función On
	0x56	Checksum procedente de la interfaz
0x00		Checksum correcto
	0x55	Programación completa
0xfb		Tercer bloque de datos
0x00 0x20		Dirección de inicio de la EEPROM (byte 32)
0x00 0x04		Bitmap:device #3
0x00		BYTE 34. Evento de fin de timer Macro:instant(offset respecto al instante de inicio de la macro)
0x01		1elemento
0x63		Código de Casa A,función OFF
0x00 0x04		Bitmap:device #3
0x00		Envío de ceros para el resto de la cadena de datos
0x00		

0x00		
	0x8c	Checksum procedente de la interfaz
0x00		Checksum correcto
	0x55	Programación completa. Interfaz lista

TABLA XX: Ejemplo de macro CM11

De cada bloque de 19 bytes enviado sólo 2 bytes se desperdician, por lo que se guardan 16bytes en la macro. Si la macro es de 1024bytes, caben 64 bloques enviados.

Las direcciones de la Eeprom de inicio de cada bloque son de 16 en 16 bytes.

- **BORRADO DE LA EEPROM**

Para borrar el contenido de la Eeprom debemos establecer los siguientes valores en la memoria:

Número de byte	Contenido
1	0x00
2	0x03(macro offset)
3	0xff (fin de tabla de timers)
4	0xff (fin de tabla de iniciadores)
5	0xff

TABLA XXI: Borrado del contenido de la Eeprom

El resto de los bytes los establecemos a 0xff.

- **RECEPCIÓN DE DIRECCIONES DE LA EEPROM TRAS LA EJECUCIÓN DE UNA MACRO**

Cuando la interfaz recibe una señal de iniciación de macro, o cuando se procesa un evento del timer, inmediatamente se lleva a cabo una transmisión asíncrona de la dirección de la Eeprom a partir de la cual están alojadas las órdenes asociadas al evento que se ejecuta , que será subsecuentemente procesada por el CM11.

El comando de instrucciones es el siguiente:

0x5b	Transmisión de la dirección de la EEPROM
0xhh	Byte alto de la dirección de la macro de la EEPROM (*)
0xll	Byte bajo de la dirección de la macro de la EEPROM

TABLA XXII: Dirección de la Eeprom enviada por el CM11

(*) El bit 7 de este byte siempre está a 1. Los bits 4-6, en el caso de que iniciador de la macro son una réplica de los bits 12-14 reservados (ver sección 6.8); o en el caso de que la transmisión se deba a un timer, dichos bits son 0. Solamente los bits 0-1 de este byte son parte de la dirección de la Eeprom.

Esta transmisión se efectúa de una sola vez, y no requiere asentimiento por parte de la interfaz, ya que puede no estar conectada al PC.

• **EJEMPLO MACRO DECODER**

Este es un ejemplo de una macro traducida con el programa Eeprom Decoder, al que se hace referencia en el apartado del ACTIVEHOME.

A1 Off: 001d
Delay 0:00, 1 element
A3 Brighten first, then Dim 000%
A1 On: 0023
Delay 0:00, 1 element
A2 Brighten first, then Dim 000%
Delay 0:25, 1 element
A2 Brighten first, then Dim 50%
S.....S Start 348, Stop 365, Time On 6:30, Time Off 17:30 (Security)
Time On Macro:
Delay 0:00, 1 element
A2 Brighten first, then Dim 100%
Time Off Macro:
Delay 0:00, 1 element
A2 Off

TABLA XXIII: Macro traducida con el programa Eeprom Decoder

En este código observamos dos macros y un evento programado:

MACRO #1

- A1 Off: 001d
- Delay 0:00, 1 element
- A3 Brighten first, then Dim 000%

Esta macro se inicia cuando el usuario envía la señal A1 OFF a la línea eléctrica, y los datos de la macro están almacenados en la dirección 001d de la Eeprom. Los datos consisten en una función y una

dirección, y el evento es aumentar la intensidad del dispositivo A3 hasta un 100%.

MACRO #2

- A1 On: 0023
- Delay 0:00, 1 element
- A2 Brighten first, then Dim 000%
- Delay 0:25, 1 element
- A2 Brighten first, then Dim 050%

Esta macro se inicia cuando la señal A1 ON es detectada en la línea eléctrica, y los datos son almacenados a partir de la dirección 0023 de la EEPROM. Los datos consisten en 2 elementos, el primero de ellos instantáneo y el segundo con un retraso de 25 minutos. La acción instantánea a realizar será aumentar la intensidad de A2 hasta un 100%, y luego, 25 minutos más tarde, deberá atenuarse A2 hasta el 50%.

EVENTO PROGRAMADO #1

- S.....S Start 0, Stop 365, Time On 6:30, Time Off 17:30 (Security)
- Time On Macro:
 - Delay 0:00, 1 element
 - A2 Brighten first, then Dim 000
- Time Off Macro:
 - Delay 0:00, 1 element
 - A2 Off

Este evento sólo ocurrirá los sábados y domingos a lo largo del año. Es un evento de seguridad, y por lo tanto habrá una variación de +/- 30 minutos alrededor de la hora programada.

A las 6:30 am +/- 30 minutos, la lámpara A2 será iluminada hasta un 100%, y a las 5:30 pm +/- 30 minutos, A2 será apagada.

7.10 DESHABILITAR LA ALARMA DEL PUERTO SERIE

Puede ser necesario, para evitar el 'trouble-shooting' (localización y resolución de problemas), deshabilitar la señal de Serial Ring (RI), es decir, la alarma del puerto serie, aunque por el contrario las macros contenidas

dentro del ordenador no podrán operar, ni tampoco el PC será capaz de comprobar el estado del sistema.

El protocolo siguiente permitirá habilitar y deshabilitar la señal de alarma del puerto serie (serial ring, RI)

PC	INTERFAZ	DESCRIPCIÓN
0xeb		Habilita la señal de alarma (ring)
	0xeb	Checksum
0x00		Checksum correcto
	0x55	Interfaz preparada

TABLA XXIV: Habilitar la alarma puerto serie

PC	INTERFAZ	DESCRIPCIÓN
0xdb		Deshabilita la señal de alarma (ring)
	0xdb	Checksum
0x00		Checksum correcto
	0x55	Interfaz preparada

TABLA XXV: Deshabilitar la alarma puerto serie

El valor por defecto de la señal de alarma (serial ring RI) después de un reset es habilitada.

7.11 PROGRAMAR LA HORA DE LA INTERFAZ

El PC puede establecer el reloj de la interfaz con una transmisión no solicitada por dicha interfaz en cualquier momento que lo desee.

Asimismo, en el caso de que la interfaz detecte la ausencia de señal eléctrica en la línea de comunicación, esperará a que el PC vuelva a estar disponible y en ese momento solicitará al PC la hora actual.

Para el CM11, desde la interfaz, la instrucción para solicitar la hora es **0xa5**. El PC debe responder con la siguiente transmisión (7 bytes):

RANGO DE BITS	DESCRIPCIÓN
55-48	Cabecera (0x9b) (byte 0)
47-40	Hora actual (segundos) (byte 1)
39-32	Hora actual (minutos, en un rango desde 0 a 119) (byte 2)
31-24	Hora actual (horas/2, en un rango desde 0 a 11) (byte 3)
23-16	Bits [0..7]del día del año (byte 4)
15	Día del año (bit 8)

14-8	Máscara del día (DLMXJVS, bit 8= Domingo)
7-4	Monitored Código de casa
3	Reservado
2	Battery timer clear flag
1	Monitored status clear flag
0	Timer purge flag

TABLA XXIV: Programación de la hora de la interfaz

- **Monitored status clear flag.** Si establecemos este bit a uno, inicializaremos los registros de estado que el CM11 mantiene de los dispositivos direccionados y encendidos, y el valor de dim de cada uno. Esta opción debe contemplarse cuando queramos cambiar el código de casa monitorizado por el CM11.
- **Battery timer clear flag.** Cuando situamos a uno este flag, el contador de las baterías se sitúa a 0xffff, y a partir de ahí se decrementa cada minuto que el CM11 detecte ausencia de línea eléctrica. Asimismo, después de una situación de 10 minutos en ausencia de corriente, se produce una situación conocida como reset, y también entonces el contador de la batería se sitúa a 0xffff.

Otra posibilidad que tenemos cuando recibimos por parte de la interfaz la instrucción 0xa5 es enviar solamente la cabecera 0x9b seguida de un tiempo de retardo indeterminado -de alrededor de 10 milisegundos-, con lo que sería suficiente para que el CM11 cesara de sondear al PC.

7.12 SOLICITUD DEL ESTADO DE LA INTERFAZ

El PC puede solicitar el estado actual de la interfaz en cualquier momento mediante la instrucción **0x8b**.

Dicha petición es inmediatamente seguida por:

RANGO DE BITS	DESCRIPCIÓN
111-96	Temporizador de la batería (Battery timer)(establecido a 0xffff después de una señal de reset) (2byte)
95-88	Hora actual(segundos) (1byte)
87-80	Hora actual (minutos, en un rango desde 0 a 119)(1byte)
79-72	Hora actual (horas/2, en un rango desde 0 a 11)
71-63	Día del año
62-56	Máscara del día (DLMXJVS, bit 56=Domingo)
55-52	Monitored Código de Casa
51-48	Revisión del Firmware (en un nivel de 0-15)

47-32	Actuales dispositivos monitorizados (bytes 8-9)
31-16	Estado de los dispositivos monitorizados (On/Off) (bytes 10-11)
15-0	Nivel de DIM de los dispositivos monitorizados (bytes 12-13)

TABLA XXVII: Respuesta del CM11 ante solicitud de estado de la interfaz

Los últimos 3 bytes corresponden a un mapeo de los dispositivos en dos bytes, como vimos en la sección de la Eeprom, y que mostramos a continuación:

High byte	10	2	8	16	12	4	6	14
Low byte	9	1	7	15	11	3	5	13

El byte bajo se recibe en primer lugar, y posteriormente el byte alto.

