

**COMUNICACIÓN CON LOS  
VISUALIZADORES  
MEDIANTE  
PROCOLO DTPM.**

# ÍNDICE GENERAL

1. DESCRIPCIÓN

2. PROTOCOLO DTPM

2.1. Protocolo DTPM.

2.2. Órdenes Protocolo DTPM.

2.3. Opciones de Visualización.

2.4. Variables Alfanuméricas.

2.5. Test con el Software MP Tools.

Anexo 1. Configuración por defecto de las Pantallas

Anexo 2. Códigos de Error Protocolo DTPM

Anexo 3. Estructura de datos *DEVICE\_USER\_SETTINGS*

## 1. DESCRIPCIÓN

Con las Pantallas de la serie DMM, se puede interactuar de varias formas, según el modelo del que se trate.

El Protocolo DTPM, es el Protocolo Nativo de estos dispositivos

En este documento, se detalla la comunicación con el Protocolo DTPM a través de comunicación por puerto Serie, USB o ETHERNET (TCP/IP) y las Órdenes y opciones necesarias para interactuar con dichas pantallas.

## 2. PROTOCOLO DTPM

### 2.1. Protocolo DTPM.

#### 2.1.1. Descripción del Protocolo.

El Protocolo DTPM es el Protocolo Nativo de los dispositivos DMM.

El protocolo de transmisión comprende un byte de sincronismo, la longitud del paquete, el nº de dispositivo, la orden que se pretende dar al dispositivo, los datos (si procede) y el checksum. Cada paquete se lanza vía USB, RS232/RS485 o ETHERNET y si se recibe correctamente, el dispositivo devuelve una contestación.

El paquete sólo debe reenviarse si se han producido errores en la transmisión (retornados por el dispositivo) o bien, no se ha recibido el ACK.

La conectividad entre el Display (Esclavo) y el dispositivo que realice la función de Maestro, puede ser por varios puertos de comunicación, tal y como se ilustra en la siguiente tabla:

<b><i>Puerto</i></b>	<b><i>Conectividad Protocolo DTPM</i></b>
<b>RS232</b>	SI
<b>RS485</b>	SI
<b>USB</b>	USB Device – Funcion CDC (VCP)
<b>ETHERNET (TCP)</b>	Hay que abrir el Socket del Puerto 53 <sup>1</sup> con la IP del Dispositivo

*Tabla 1: Opciones de conectividad para el DTPM*

<sup>1</sup>El puerto TCP puede modificarse en la configuración del dispositivo.

## 2.1.2. Estructura del Protocolo.

Cada paquete de DTPM está formado por 3 bloques fundamentales:

- Header o Cabecera
- Datos (opcionalmente)
- Checksum

En la siguiente figura se puede apreciar de una forma gráfica su composición.



Figura 1: Composición del DTPM

## 2.1.3. Campos de los paquetes.

La siguiente tabla muestra los campos de los paquetes del protocolo DTPM a transmitir.

	# BYTE	VALOR	NOMBRE	DESCRIPCIÓN
HEADER	1	0x16	SYN	Byte de sincronismo
	2	LEN1	LEN	LSB numero bytes paquete
	3	LEN2	LEN	MSB numero bytes paquete
	4	ID	ID	Dirección dispositivo destino
	5	OD	OD	Código de la orden a ejecutar
DATOS	6	DATO[0]	Primer byte de datos	Primer byte de datos (si hay datos)
	...	...	...	...
	6+n <sup>2</sup>	DATO[n-1]	Ultimo byte de datos	Ultimo byte de datos (si hay datos)
CHECKSUM	5+n+1	MSB Checksum	LSB del Checksum	
	5+n+2	LSB Checksum	MSB del Checksum	

Figura 1: Composición del DTPM

<sup>2</sup>Donde n=número de bytes de datos. En caso de n=0, el LSB del Checksum quedará en el #BYTE 6 y el MSB del Checksum quedará en el #BYTE 7

- **LEN:** Indica el número de bytes del paquete (todos los bytes entre SYN y CHECKSUM, éstos inclusive).
- **ID:** Indica el número de identificación del dispositivo.
- **OD:** Orden (1 Byte). Indica el código de la orden a ejecutar

El campo DATOS debe colocarse solo en determinados tipos de paquete. Los otros campos son obligatorios y se colocan en todos los tipos de paquete.

#### 2.1.4. Checksum.

Al final del paquete, se debe enviar el **Checksum** para verificar que no ha habido errores durante la transmisión. El **Checksum** es la suma de todos los bytes del paquete menos los dos de checksum, enviando primero el byte de menor peso (LSB).

Si este **Checksum** no corresponde, el destinatario descartará el paquete, y el origen, al terminar el tiempo de espera para el ACK, volverá a enviar el mismo paquete.

### 2.1.5. Respuesta del dispositivo.

Si el ID del paquete es **0xFF**<sup>3</sup>, todos los dispositivos lo admitirán y ejecutarán la orden correspondiente, pero no devolverán respuesta. En caso que el ID del paquete coincida con la dirección **LocalCast**<sup>4</sup> de uno o varios dispositivos, éstos admitirán el paquete sin devolver respuesta, pero el resto de los dispositivos lo ignorarán.

La respuesta general del dispositivo consiste en 2 bytes:

	# BYTE	VALOR	NOMBRE	DESCRIPCIÓN
	1	0x06	ACK	ACK
	2	ERROR_CODE	CÓDIGO ERROR	

Tabla 3: Respuesta del dispositivo en el Protocolo DTPM

Si la orden ha llegado correctamente, se recibirá ésta confirmación. En ella hay un código de validación fijo (ACK) y un código de error del dispositivo. Éste normalmente será 0 (lo que significa que la operación se ha realizado correctamente), pero puede darse el caso que se envíe una orden ilegal en determinado momento. El valor de los distintos códigos de error se muestra en el Anexo 2.

La orden CHECKSUM (0x07) devuelve el byte bajo del último Checksum recibido correctamente en el lugar del código de error.

La orden GET NUM PACKET (0x21) devuelve el número del último paquete **SEND** recibido correctamente por el dispositivo. El contador se pone a 0 si el dispositivo recibe un paquete **EDIT** o **EDITFILE** y se incrementa al validar un paquete **SEND**.

La orden GET BAT LEVEL (0x96) devuelve un byte<sup>5</sup> que indica el nivel de la batería del dispositivo en el lugar del código de error.

<sup>3</sup> Dirección BROADCAST del Protocolo DTPM.

<sup>4</sup> Parámetro de la Configuración del Dispositivo.

<sup>5</sup> Ver descripción de la orden GET BAT LEVEL.

### 2.1.6. Errores en Transmisión.

En caso de transmisión UNICAST (no cuando se envía en modo BROADCAST o LOCALCAST), el paquete sólo debe reenviarse si se han producido errores en la transmisión (retornados por la pantalla) o bien, la pantalla no ha devuelto ACK. En caso de no recibir el ACK se debe seguir el siguiente procedimiento para determinar si es necesario reenviar el paquete:

- 1) Enviar la orden **GET NUM PACKET** después de 100ms del fallo de comunicación.
- 2) La pantalla enviará el número del último paquete recibido correctamente. Si no se recibe, se debe esperar durante 3 segundos.
- 3) Si todavía así no se recibe, se debe volver a empezar por 1) hasta haberlo probado 3 veces.

Si no se ha recibido, se puede afirmar que la comunicación se ha interrumpido. En caso contrario se debe seguir.

- 4) Enviar la orden **CHECKSUM** después de 100ms.
- 5) La pantalla enviará el Checksum del último paquete recibido correctamente. Si no se recibe, se debe esperar durante 3 segundos
- 6) Si todavía así no se recibe, se debe volver a empezar por 4) hasta haberlo probado 3 veces.

Si no se ha recibido, se puede afirmar que la comunicación se ha interrumpido. En caso contrario se debe seguir.

Si tanto el Checksum coincide con el del último paquete como el número del paquete coincide con el enviado, se puede afirmar que el display ha recibido y procesado correctamente el paquete y por tanto, es posible continuar la transmisión con el siguiente.

En caso contrario se debe seguir:

- 7) Enviar la orden **GET NUM PACKET** después de 100ms.
- 8) La pantalla enviará el número del último paquete recibido correctamente. Si no se recibe se debe esperar durante 3 segundos.
- 9) Si todavía así no se recibe, se debe volver a empezar por 7) hasta haberlo probado 3 veces.

Si tanto el Checksum coincide con el del último paquete como el número del paquete coincide con el enviado, se puede afirmar que el primer **GET NUM PACKET** recibió datos incorrectos por la razón que fuera y que el display ha recibido y procesado correctamente el paquete y por tanto, se puede continuar la transmisión con el siguiente.

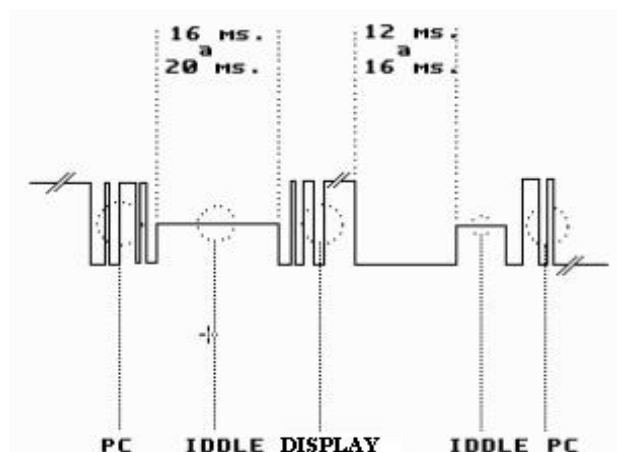
En caso contrario, se concluye que el paquete NO ha sido recibido correctamente por el display y por tanto debe reenviarse.

**El reenvío de información sin los debidos controles puede provocar que la información en un determinado programa quede duplicada provocando una ejecución no deseada por el usuario.**



### 2.1.7. Comunicación por Bus RS485.

Éste sistema se basa en la transmisión de datos mediante tensiones diferenciales. Está concebido para que se "cuelguen" en una misma línea de comunicaciones hasta 32 transmisores/receptores RS485. El sistema consiste en que solamente haya un transmisor activo en un momento dado; los demás se mantienen en recepción, quedando en estado de alta impedancia. Para que no haya conflicto de líneas, un receptor se puede convertir en transmisor sólo si no hay ningún otro elemento transmisor. En el gráfico adjunto se muestra como debe establecerse la comunicación entre dos elementos de una línea RS485.



Primeramente aparece la información transmitida por el DTE; se puede observar que cuando ésta termina, el driver del DTE se pone en recepción, quedando en alta impedancia o IDLE LINE.

Al cabo de entre 16 y 20 milisegundos, el display activará su driver de transmisión, y empezará a transmitir si la recepción ha sido correcta (NOTA: en caso de órdenes que conlleven una gran cantidad de tiempo al ejecutarlas por la pantalla, como las relativas a la edición de archivos, este tiempo puede retrasarse hasta 3 segundos).

Cuando haya terminado, la pantalla se mantiene en modo transmisión durante 12 a 16 milisegundos. Al cabo de ese tiempo, se vuelve a poner en recepción, pudiendo activarse el driver del DTE para empezar a transmitir de nuevo otro paquete. Si no se cumplen estos tiempos, probablemente habrá algún conflicto de línea con los consiguientes errores de transmisión.

## 2.2. Órdenes del Protocolo DTPM.

A continuación, se detallan las distintas órdenes del Protocolo, una breve descripción y algún ejemplo de su uso. Si no se indica lo contrario, todos los ejemplos se realizan en dispositivos con ID 1. La notación numérica es en hexadecimal, salvo se indique lo contrario.

### **RESET RAM (0x01)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Se reinicializa el dispositivo y se borra el contenido de la memoria no volátil. Al volver a arrancar no se ejecutará ningún programa.

**Ésta orden no borra los archivos del directorio de la pantalla.**

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 01 1F 00

Respuesta de Dispositivo: 06 00

### **RESTART (0x02)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Se reinicializa el dispositivo sin borrar los datos guardados en memoria no volátil.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 02 20 00

Respuesta de Dispositivo: 06 00

### **STOP (0x03)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Con ésta orden se detendrá el programa en ejecución.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 03 21 00

Respuesta de Dispositivo: 06 00

### **STOP AND CLEAR (0xA1)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Con ésta orden se detendrá el programa en ejecución y se borrará el contenido de la pantalla.

### **CHECKSUM (0x07)**

**Datos:** SIN DATOS

**Respuesta:** ACK + byte bajo del último Checksum recibido correctamente.

**Descripción:** Devuelve el byte bajo del Checksum del último paquete recibido correctamente, salvo los paquetes CHECKSUM y GET NUM PACKET.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 07 25 00

Respuesta de Dispositivo: 06 XX

Siendo XX el byte bajo del Checksum (byte menos significativo) del último paquete recibido correctamente.

### **SET TIME (0x0A)**

**Datos:** 6 Bytes para indicar la Hora y Fecha:

*Año (1 Byte): 0 - 99*

*Mes (1 Byte): 1 - 12*

*Día (1 Byte): 1 - 31*

*Hora (1 Byte): 0 - 23*

*Minutos (1 Byte): 0 - 59*

*Segundos (1 Byte): 0 - 59*

**Respuesta:** ACK.

**Descripción:** Envía la hora y fecha al Reloj interno de la pantalla

El formato es 1 byte de año (00-99), 1 byte de número de mes (1-12), 1 byte de número del día (1-31), 1 byte para la hora (0-23), 1 byte para los minutos (0-59) y 1 byte para los segundos (0-59).

**Ejemplo:**

Hora 13:40:00 Fecha 02/03/2014

Enviado a Dispositivo: 16 0D 00 01 0A 0E 03 02 0D 28 00 76 00

Respuesta de Dispositivo: 06 00

### **GET TIME (0x0B)**

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** 6 Bytes para indicar la Hora y Fecha:

*Año (1 Byte): 0 - 99*

*Mes (1 Byte): 1 - 12*

*Día (1 Byte): 1 - 31*

*Hora (1 Byte): 0 - 23*

*Minutos (1 Byte): 0 - 59*

*Segundos (1 Byte): 0 - 59*

**Descripción:** Solicita la hora y fecha del Reloj interno de la pantalla, la cual se lo envía mediante un paquete SEND al ordenador (ID *0xFE*).

El formato es 1 byte de año (00-99), 1 byte de número de mes (1-12), 1 byte de número del día (1-31), 1 byte para la hora (0-23), 1 byte para los minutos (0-59) y 1 byte para los segundos (0-59).

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 0B 29 00

Respuesta de Dispositivo: 06 00

16 0D 00 FE 0C 0E 03 02 0D 28 13 88 01

Con lo que determina que el Reloj de la pantalla tiene Hora *13:40:19* y Fecha *02/03/2014*

### **SEND (0x0C)**

**Datos:** *Distinto para cada caso.*

**Respuesta:** ACK.

**Descripción:** Se usa esta orden tanto si el PC envía algo a la pantalla como viceversa. Este paquete puede contener cualquier tipo de datos, tanto programas, como Fuentes o Gráficos. El contenido de los datos del paquete es distinto para cada caso.

## GETVER (0x12)

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** Estructura GETVER (6 Bytes):

Versión de Software del Dispositivo (1 Byte): 10 - 254

Versión de Hardware (1 Byte): 1 - 255

Número de Columnas (2 Bytes).

NO USADO (1 Byte).

Número de Líneas (1 Byte).

**Descripción:** Este paquete solicita información sobre el Hardware y Software al display mediante una estructura que es enviada al ordenador (ID *0xFE*) en un paquete SEND. Un detalle que diferencia la estructura de GERVER EXT respecto a la de GETVER es que esta última es de dimensión fija (6 Bytes).

### **Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 12 30 00

Respuesta de Dispositivo<sup>7</sup>: 06 00

16 0D 00 FE 0C 2E C4 60 00 01 06 86 02

Con lo que se determina:

- Versión de Software del Dispositivo (1 Byte): **4.6**
- Versión de Hardware (1 Byte): **196**
- Número de Columnas (2 Bytes): **96**
- NO USADO (1 Byte): **1**
- Número de Líneas (1 Byte): **6**

## NEXEC (0x1F)

**Datos:** El nombre del programa a ejecutar en ASCII (máximo 8 caracteres).

**Respuesta:** ACK.

**Descripción:** Ejecuta el contenido de un programa enviado anteriormente y que se encuentra en la memoria del display.

**Ejemplo:** Si por ejemplo, se desea ejecutar el programa MPTTEST incluido dentro de la memoria del display al salir de fábrica.

Enviado a Dispositivo: 16 0D 00 01 1F 4D 50 54 45 53 54 20 02

Respuesta de Dispositivo: 06 00

Si el programa no se encuentra grabado en la pantalla, la respuesta sería: 06 01

<sup>7</sup>Ejemplo para un caso determinado. Los valores varían según producto

### GET NUM PACKET (0x21)

**Datos:** SIN DATOS

**Respuesta:** ACK + número del último paquete **SEND** recibido correctamente.

**Descripción:** En lugar del código de error, devuelve el número del último paquete **SEND** recibido correctamente por el dispositivo. El contador se pone a 0 si el dispositivo recibe un paquete **EDIT** o **EDITFILE** y se incrementa al validar un paquete **SEND**.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 21 3F 00

Respuesta de Dispositivo: 06 XX

Siendo XX el número del último paquete SEND recibido correctamente.

### FASTEXEC (0x27)

**Datos:** Programa para ser ejecutado por el Display (Máximo 1000 Bytes)

**Respuesta:** ACK.

**Descripción:** Ésta orden es la más usada para interactuar con el display. Éste paquete debe contener en su interior un programa constituido por los códigos de efecto y textos que se quieren mostrar al público.

Se utiliza este paquete para enviar información a la pantalla, es decir, dentro del paquete FASTEXEC se incluye un programa que puede estar constituido por todos los Tokens y Pretokens descritos en el apartado VISUALIZACION, así como con los textos que se desee mostrar en pantalla.

El paquete FASTEXEC sólo debe usarse cuando el contenido del mismo sea inferior a **1000 bytes** porque en algunos dispositivos el tamaño del buffer de recepción es muy limitado y no sería suficiente para recibirlo sin perder bytes. En caso de suceder esto, se puede bajar la velocidad de transmisión o usar los paquetes **EDITFILE**, **SEND** y **NEXEC**, que son mucho más flexibles y permiten la transmisión de programas de grandes tamaños.

Las distintas opciones de visualización se detallan en el apartado referente a Opciones de Visualización.

**Ejemplo:**

Programa que muestra el texto MP en modo Correr (Scroll) en la línea 1 de la pantalla.

Enviado a Dispositivo: 16 10 00 01 27 03 C7 31 2C 31 04 E0 4D 50 27 03

Respuesta de Dispositivo: 06 00

## **PUTPORT (0x28)**

**Datos:** 5 Bytes:

Número de Puerto (1 Byte): 1 - n

Acción a realizar (1 Byte): 0: Desactivar; 1: Activar; 2: Ejecutar Secuencia

Número de Repeticiones de la secuencia ON/OFF (1 Byte)

Tiempo en ON (1 Byte): en cuartos de segundos

Tiempo en OFF (1 Byte): en cuartos de segundos

**Respuesta:** ACK.

**Descripción:** Esta orden permite el control de los Puertos Digitales (RELES) de la pantalla a través del protocolo de comunicación.

El campo *Acción a realizar* (Byte 2) puede tener los siguientes valores:

0: DESACTIVAR

1: ACTIVAR

2: Ejecutar Secuencia

10: **MODO TEST** - DESACTIVAR<sup>8</sup>

11: **MODO TEST** - ACTIVAR<sup>9</sup>

**Ejemplo:**

Se desea activar el Relé conectado al Puerto Digital 1 de la pantalla

Enviado a Dispositivo: 16 0C 00 01 28 01 01 00 00 00 4D 00

Respuesta de Dispositivo: 06 00

## **PUTVARS (0x2E)**

**Descripción:** Envía Variables a la memoria de la pantalla.

[Ver apartado 2.4 referente a Variables Alfanuméricas.](#)

## **GETVARS (0x2F)**

**Descripción:** Solicita las Variables de la memoria de la pantalla.

[Ver apartado 2.4 referente a Variables Alfanuméricas.](#)

<sup>8</sup> El puerto se DESACTIVA independientemente de la configuración. Permanece en este estado hasta que se reinicia.

<sup>9</sup> El puerto se ACTIVA independientemente de la configuración. Permanece en este estado hasta que se reinicia.

## GETVER EXT (0x30)

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** Estructura GETVER EXT (16 Bytes o más):

Versión de Software del Dispositivo (1 Byte): 10 - 254

Versión de Hardware (1 Byte): 1 - 255

Número de Columnas (2 Bytes).

NO USADO (1 Byte).

Número de Líneas (1 Byte).

Versión de FONTS (1 Byte).

Versión de MPBASIC (1 Byte).

Versión de PROGRAMAS (1 Byte).

... Futura Ampliación ...

**Descripción:** Este paquete solicita información sobre el Hardware y Software al display mediante una estructura que es enviada al ordenador (ID 0xFE) en un paquete SEND. Un detalle que diferencia la estructura de GERVER respecto a la de GETVER EXT, es que esta última es de dimensión variable y en el futuro se irá ampliando con nuevos datos de hardware y software del display, por tanto, el programa que la reciba en el PC no debe esperar un paquete de 16 bytes sino que debe contemplar que la longitud del mismo sea variable.

### **Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 30 4E 00

Respuesta de Dispositivo<sup>10</sup>: 06 00

16 39 00 FE 0C 2E C4 60 00 01 06 1E 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C4 00 2A 43 46 47 20 31 00 00 00 00 00 45 00 00 00 00 00 00 00 00 00 00 00 38 05

Con lo que se determina:

- Versión de Software del Dispositivo (1 Byte): 4.6
- Versión de Hardware (1 Byte): 196
- Número de Columnas (2 Bytes): 96
- NO USADO (1 Byte): 1
- Número de Líneas (1 Byte): 6
- Versión de FONTS (1 Byte): 3.0
- Versión de MPBASIC (1 Byte): 2.0
- Versión de PROGRAMAS (1 Byte): 0

<sup>10</sup> Ejemplo para un caso determinado. Los valores varían según producto.



### **TEST PIXELS (0x3C)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Este paquete inicia un test de barras y columnas automático en el display. El test permanece hasta que se le envíe un paquete STOP o se reinicie el display.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 3C 5A 00

Respuesta de Dispositivo: 06 00

### **GET SETTINGS (0x59)**

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** 36 Bytes: Estructura de tipo DEVICE\_USER\_SETTINGS

**Descripción:** Devuelve una estructura con parámetros de configuración de la pantalla relativos a la visualización. [Ver detalles en ANEXO 3.](#)

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 59 77 00

Respuesta de Dispositivo: 06 00

16 2B 00 FE 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 05 0F 05 64  
00 1E 40 01 01 01 00 28 00 05 78 00 01 F0 01 32 00 00 F2 03

### **PUT SETTINGS (0x5A)**

**Datos:** 36 Bytes: Estructura de tipo DEVICE\_USER\_SETTINGS

**Respuesta:** ACK

**Descripción:** Envía a la pantalla una estructura con parámetros de configuración de la pantalla relativos a la visualización. [Ver detalles en ANEXO 3.](#)

**Ejemplo:**

Enviado a Dispositivo: 16 2B 00 01 5A 49 4E 54 38 39 33 32 00 00 00 00 00 00 05 0F 05  
64 00 1E 40 01 01 01 00 28 00 05 78 00 01 F0 01 32 00 00 04 05

Respuesta de Dispositivo: 06 00

*Luego de recibir ACK (y con un retraso de unos 2 segundos) la pantalla se reiniciará para que se apliquen los cambios realizados en la configuración.*

### GET LUM INPUT (0x69)

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** 2 Bytes:

*Lectura de la sonda de luz ambiente (1 Byte): 0 a 100 %  
NO USADO (1 Byte).*

**Descripción:** Pide la lectura de la sonda de luz ambiental del dispositivo. Éste devuelve un paquete con un buffer de 2 bytes mediante la orden SEND: la lectura de la sonda se encuentra en el primer byte (0 - 100).

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 69 87 00  
Respuesta de Dispositivo: 06 00  
16 09 00 FE 0C 41 00 6A 01

Con lo que se determina que la sonda detecta luz ambiente al 65% del máximo.

### GET TEMP INT (0x88)

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** 2 Bytes:

*Temperatura interna del dispositivo (1 Byte con signo): -128 a +127°  
NO USADO (1 Byte).*

**Descripción:** Pide la temperatura interna del dispositivo. Éste devuelve un paquete con un buffer de 2 bytes mediante la orden SEND: la temperatura se encuentra en el primer byte (byte con signo).

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 88 A6 00  
Respuesta de Dispositivo: 06 00  
16 09 00 FE 0C 1B 00 44 01

Con lo que se determina que la temperatura interna del dispositivo es de 27°

### **GET BAT LEVEL (0x96)**

**Datos:** SIN DATOS

**Respuesta:** ACK + CÓDIGO.

Significado del CÓDIGO:

0x25: Batería baja o descargada.

0x30: Batería correcta.

Cualquier otro valor: Orden no soportada por el dispositivo.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 96 B4 00

Respuesta de Dispositivo: 06 25

Con lo que se determina que la batería del dispositivo está descargada.

### **FILE SYSTEM RESET (0x9E)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Formatea el directorio del dispositivo borrando todos los archivos del mismo. Esta tarea puede tardar hasta 1 minuto según el modelo.

**ATENCIÓN:** Utilizar con precaución ya que una vez realizado no se podrán recuperar los archivos (incluidos los archivos de Fuentes y Gráficos). Para poder volver a usarlos habrá que volver a cargarlos en el dispositivo.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 9E BC 00

Respuesta de Dispositivo: 06 00

*La respuesta puede tardar hasta un minuto en recibirse. Luego la pantalla se reiniciará.*

### **RESET CONFIG (0xA0)**

**Datos:** SIN DATOS

**Respuesta:** ACK.

**Descripción:** Restaura la Configuración de Fábrica. El dispositivo se comporta igual que cuando salió de fábrica.

**ATENCIÓN:** Utilizar con precaución ya que una vez restaurada la configuración original no se podrán recuperar los valores que habían sido modificados con anterioridad, por lo que habrá que volver a configurar el dispositivo.

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 A0 BE 00

Respuesta de Dispositivo: 06 00

*Luego de recibir ACK (y con un retraso de unos 2 segundos) la pantalla se reiniciará para que se apliquen los cambios realizados en la configuración.*

## **SET TEMP SETTINGS (0xC1)**

**Datos:** 1 Byte:

*Offset Temperatura Exterior (1 Byte con signo): -12.0° a +12.0°*

**Respuesta:** ACK.

**Descripción:** Pone un nuevo Offset a la Temperatura Exterior de la pantalla (en décimas de grado con signo).

**Ejemplo:**

Se desea corregir la lectura del sensor de temperatura exterior con un offset de -1.5°.

Enviado a Dispositivo: 16 08 00 01 C1 F1 D1 01

Respuesta de Dispositivo: 06 00

## **N GET TEMP (0xC2)**

**Datos:** SIN DATOS

**Respuesta:** ACK + **PAQUETE SEND**

**Datos PAQUETE SEND:** 4 Bytes:

*Temperatura externa (2 Bytes con signo): -20.0 a +120.0°*

*Offset (1 Byte con signo): -12.0° a +12.0°*

*NO USADO (1 Byte).*

**Descripción:** Pide la temperatura exterior y offset al dispositivo. Éste devuelve un paquete con un buffer de 2 bytes mediante la orden SEND: la temperatura se encuentra en los 2 primeros bytes (el valor representa décimas de grado con signo), y el offset en el tercer byte (el valor representa décimas de grado con signo).

**Ejemplo:**

Enviado a Dispositivo: 16 07 00 01 C2 E0 00

Respuesta de Dispositivo: 06 00

16 0B 00 FE 0C 0B 01 F1 00 28 02

Con lo que se determina que la temperatura exterior es de 26.7° y el offset es de -1.5°

## 2.3. Opciones de Visualización.

Para mandar programas a la Pantalla para visualizarlos, podemos usar el paquete **FASTEXEC**. Esta orden permite en la zona (datos) del paquete de comunicaciones ubicar un script o programa que la pantalla ejecutará. Cuando se ejecuta un programa en el display, éste va decodificando cada código constituido por 2 bytes (1 byte Pretoken y 1 byte Token), seguido de un texto. Cuando es interpretado un código, el público puede observar cambios en la información del display.

Cada programa puede contener muchos códigos que se ejecutan secuencialmente, y pueden ser de 4 tipos diferentes: DATO (parámetros de ejecución), MODO (los modos de aparición de los textos en el display), TIEMPO (datos variables de tiempo y temperatura que se pueden añadir a los textos) y EFECTO (alteran el contenido de los datos ya mostrados en el display con un MODO).

Se aconseja que se coloquen los códigos DATO, MODO, TIEMPO y EFECTO por este orden. Los programas terminan con un byte NULL (0x00).

Los códigos de carácter van del 0 al 255 y se corresponden con la codificación **Windows-1252** (Extensión de **ISO-8859-1**) aunque solo son visibles a partir del espacio en blanco (0x20).

DATO	DESCRIPCION	<i>n</i>	Pretoken	Token
Página	Nueva página del Script		0x03	0x20
Blink	Parpadea el texto que se encuentre entre 2 Blink		0x03	0xA0
Color < n >	A partir de aquí se cambia el color del texto	0 - 7	0x03	0xA1
Gráfico < n >	Aparece un gráfico	0 - 99	0x03	0xA4
Variable < n >	Varias cifras formateadas que se pueden actualizar <sup>11</sup>	A - Z	0x03	0xAB
Grosor < n >	Cada columna ocupará n columnas de grosor	1 - 8	0x03	0xC0
Tipo de letra < n >	Tipos de letra (según modelo)	0 - 99	0x03	0xC1
Velocidad Modo < n >	Velocidad del correr	1 - 99	0x03	0xC4
Espera Modo < n >	Tiempo de espera entre modos (en cuartos de segundo)	4 - 99	0x03	0xC5
Línea < n , h >	Nos situamos en la línea n de la pantalla. El texto que viene a continuación se situará en la línea correspondiente. El campo h debe tener siempre valor 1 <sup>12</sup> .	1 - Líneas Totales	0x03	0xC7
Sincronismo	Al principio de grupo de líneas (ejecución en paralelo)		0x03	0xC9
No sincronismo	Final de un grupo de líneas (ejecución en paralelo)		0x03	0xCA
Idioma < n >	Cambia el idioma de los tiempos		0x03	0xCB
Día evento < x >	Fecha destino para el evento		0x03	0xCC
Alineación < n >	Alineación del texto en la línea o ventana	0: Centro 1 : Izquierda 2 : Derecha	0x03	0xCD
Brillo < n >	Se establece el brillo del display como un porcentaje respecto al brillo máximo. Valor 0 es AUTOMATICO: Brillo en función de luz ambiente (solo en caso de disponer de sensor de luz)	1 - 100 % 0 para <b>AUTOMÁTICO</b>	0x03	0xD0

<sup>11</sup> Ver apartado 2.4

<sup>12</sup> Futura ampliación

MODO	DESCRIPCION	Pretoken	Token
Entrada Izquierda	El texto entra por la izquierda	0x04	0xD0
Correr	El texto se desplaza de derecha a izquierda	0x04	0xE0
Sube	El texto sube	0x04	0xE5
Baja	El texto baja	0x04	0xE6
Inmediato	El texto aparece inmediatamente	0x04	0xF0

TIEMPO	DESCRIPCION	Pretoken	Token
Año	Dos cifras que indican el año	0x01	0x96
Número de mes	Dos cifras que indican el mes	0x01	0x97
Mes	Nombre del mes	0x01	0x98
Número de día	Dos cifras que indican el día	0x01	0x99
Día	Nombre del día	0x01	0x9A
Horas	2 cifras que indican la hora del día	0x01	0x9B
Minutos	2 cifras que indican el minuto del día	0x01	0x9C
Segundos	2 cifras que indican los segundos de la hora	0x01	0x9D
Temperatura	Temperatura	0x01	0x9F
Diferencia: días	Días que faltan para la fecha del evento	0x01	0xA4
Diferencia: Semanas	Semanas que faltan para la fecha del evento	0x01	0xA5
Diferencia: Meses	Meses que faltan para la fecha del evento	0x01	0xA6
Hora (HH:MM)	5 caracteres <b>Hora : Minutos</b>	0x01	0xA7
Temperatura (°C)	4 caracteres Temperatura y los símbolos '°' y 'C'	0x01	0xA8
Día Abreviado	Día de la semana abreviado	0x01	0xA9
Mes Abreviado	Nombre del mes abreviado	0x01	0xAA
Diferencia: Minutos	Horas que faltan para la fecha del evento	0x01	0xAC
Diferencia: Segundos	Segundos que faltan para la fecha del evento	0x01	0xAD

## 2.4. Variables Alfanuméricas.

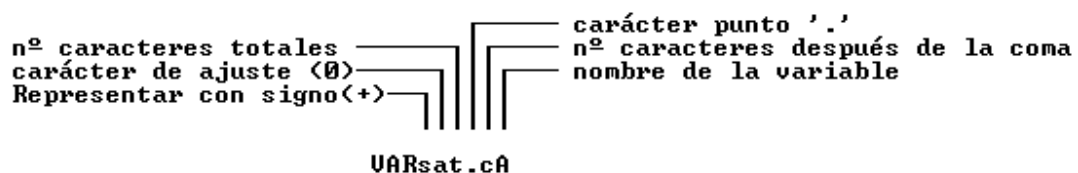
Las pantallas guardan en su memoria no volátil un conjunto de 26 variables alfanuméricas. Cada una de estas variables puede ser:

- Un *string* de 8 caracteres como máximo.
- Un número real en coma flotante con el formato IEEE de 64 bits.

Para leer o modificar el valor de las variables se usan las órdenes GETVARS y PUTVARS del protocolo. Los detalles de estas órdenes se detallan posteriormente.

Para representar las variables hace falta insertar en el Script el **DATO VAR** y después el nombre de una de las 26 variables [**A,Z**] (en mayúscula y sin incluir la Ñ) sin ningún espacio en medio. Las variables se ponen a 0 si se les asigna este valor o cuando se inicializa la pantalla. En caso de apagar, o desconectar la corriente, las variables conservarán su valor mientras dure la batería interna. Tampoco se alterarán si se para o vuelve a empezar la ejecución del script. Las variables tienen una precisión de 16 dígitos, es decir, se pueden mostrar 16 dígitos (sumando los de antes y después de la coma) sin pérdida de precisión, en caso que se muestren más de 16, los dígitos menos significativos diferirán del valor real. Esto último es inherente al formato de coma flotante IEEE de 64 bits.

Como el formateo por defecto de una variable es con 6 dígitos después de la coma, puede ser incómodo representar números enteros con tantos decimales. Se puede formatear la variable añadiendo el número de dígitos totales y el número de dígitos después de la coma, de la forma:



Ejemplos:

**VAR6.2B** Si la variable vale por ejemplo 1, el resultado será: `__1.00`  
(los dos underscores representan espacios)

**VAR9.0** `_____1`

**VAR09.0** Se puede ajustar a la izquierda con ceros en vez de espacios, colocando un cero  
`00000001`

**VAR+9.0** También se puede indicar la representación con signo `_____+1`

**VAR-9.0** Poniendo el signo menos se ajusta por la izquierda `1_____`

**VAR9B** Puede ponerse solo el ajuste de antes de la coma `_____1`

**VAR.9B** También puede ponerse solo el ajuste de después de la coma `1.00000000`

En principio, una variable a la que se le ha asignado un número, puede ser substituida por un *string* mediante un paquete PUTVARS. Si el formateo es correcto, no tiene por qué haber corrimiento.

Por ejemplo: a la variable X se le ha asignado el número 456342 mediante un paquete PUTVARS. **INMEDIATO VAR8.0Z** mostrará el número \_\_456342 (con dos espacios de justificación a la izquierda que suman los 8 caracteres). Si mediante otro paquete PUTVARS asignamos el *string* 'PARO' a la variable Z, en la pantalla se mostrará \_\_\_\_PARO (con 4 espacios de justificación a la izquierda que suman los 8 caracteres) con lo que no habrá variaciones de justificación y tanto el número como el *string* se mostrarán en el mismo lugar de la pantalla. Se debe ir con mucho cuidado si se usa este sistema ya que si se envía un paquete PUTVARS, que por ejemplo decremente en 1 un *string* (acción ilegal), el *string* será considerado como el número 0, con lo que el resultado será el número -1.

Los números se redondean a partir de la primera cifra visible; es decir, si internamente la variable vale 3.141592 y nosotros mostramos cuatro decimales (EX: **VAR.4X**), como el primer número invisible (el 9) es superior a 4, entonces se redondea el 5 a 6, mostrándose en pantalla 3.1416.

### RESTRICCIONES

- Es ilegal representar números de más de 16 caracteres sin contar los símbolos (como la coma, etc.) ya que habría pérdida de precisión.
- Es ilegal formatear la variable para más de 22 caracteres, contando todos los símbolos que haya, el signo positivo, la coma, etc.
- Si se colocan más de 8 caracteres de formateo (entre el DATO VAR y la variable), éstos son despreciados, en la pantalla no se muestra la variable, sino el símbolo '---' (tres guiones), también se muestran los caracteres que sobran del formateo.

### CONSEJOS

Como que se trata de mostrar una información en coma flotante de alta precisión en la pantalla, cuanto más variables se escriban, más tiempo tardará la pantalla en completar la información, como resultado de lo cual, puede darse el caso que se retrasen algunos modos de aparición o se produzcan brusquedades. Procure utilizar siempre el modo INMEDIATO o cualquier otro que deje la información estática en pantalla. Evite utilizar HORMIN o BLINK, ya que el carácter que parpadea puede quedar estático más tiempo de lo normal.

En caso que sea imprescindible mostrar HORMIN o cualquier otra información sensible al tiempo de proceso, considere la opción de mostrar alguna de las variables en formato *string*, es decir, calculando los caracteres que han de aparecer en pantalla antes de ser enviados en un paquete PUTVARS.

### ERRORES

Cuando se usan tantas opciones, es fácil que se produzcan errores. Si la pantalla no reconoce la variable que debe mostrarse, entonces aparecerán tres guiones '---' en el sitio donde debería verse la variable. También podría ser que hayan más de 8 caracteres de formateo. Vuelva a editar el programa corrigiendo el problema y ejecútelo.

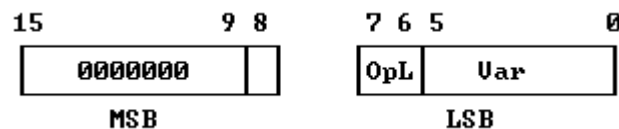


## Orden PUTVARS (0x2E)

Pone variables en la memoria de la pantalla para que se puedan mostrar con el DATO 'VAR'. El paquete se puede enviar en cualquier momento, esté la pantalla en ejecución o no, y está formado por un número de 1 a 26 estructuras del tipo PUTVARS, que contienen 10 bytes cada una. Aunque en la siguiente explicación los bytes se describan de la forma habitual (el más significativo a la izquierda), hay que tener en cuenta que para enviarlos a la pantalla se transmite primero el byte menos significativo (LSB).

### **bytes [0,1]**

- 6 bits      Número de variable [0,25] (26 variables, sin la 'Ñ').    0='A', 1='B',..., 25='Z'
- 3 bits      Operación a realizar con el valor en coma flotante o string de caracteres.  
              0 = Asignación de *string* de caracteres  
              1 = Asignación del valor en coma flotante  
              2 = Sumar el valor a la var y asignar a la var (var = var + valor)  
              3 = Restar el valor a la var y asignar a la var (var = var - valor)
- 7 bits      Ha de ser 0



### **bytes [2-9]**

- valor      Valor en coma flotante a asignar u operar con la variable. El número tiene el formato IEEE de 64 bits. O bien *string* de caracteres terminado en 0 si es menor de 8 (opcional).

## RESTRICCIONES

Aunque el sistema es muy versátil, no está permitida la mezcla de números y strings. Así se considerará ilegal y no se responderá del resultado en los siguientes casos:

- Sumar o restar un número a un string almacenado en la pantalla, en estos casos el string se considera un 0, por lo que por ejemplo si se incrementa en 1 un string, el resultado será el número 1.
- Los strings no pueden sobrepasar los 8 bytes, aunque pueden ser más cortos, poniendo un 0 (cero hexadecimal) al final del mismo, aunque no es necesario, pero puede servir para formatear la salida en la pantalla.
- Es ilegal enviar el paquete sin ninguna estructura.
- Es ilegal enviar un paquete con cualquier variable repetida (varias estructuras que se refieran a la misma variable).

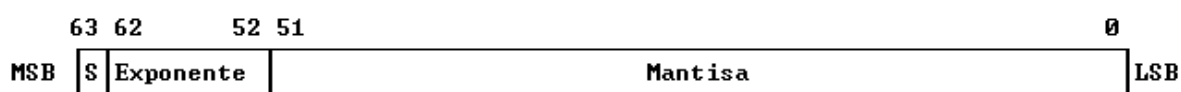
## CONSEJOS

Para aumentar el número de transmisiones por segundo, actualice sólo las variables imprescindibles, con ello reducirá el tamaño de los paquetes PUTVARS y evitará que se retrase mucho el ACK debido al tiempo de proceso en coma flotante de 64 bits. Pueden inicializarse todas las variables de una vez al principio y luego, actualizar variable a variable.

## NÚMEROS EN COMA FLOTANTE DE 64 BITS

Los números de coma flotante con el formato IEEE de 64 bits, permiten una resolución de 16 dígitos (sumando los de antes y después de la coma) sin pérdida de precisión, y están formados por una mantisa que comprende los 52 bits menos significativos, los siguientes 11 son el exponente y el último al signo. Representándose un número como:

$$N^{\circ} = -1^S * 2^{(E-1023)} * 1.MANTISA$$



El número 0 se representa con 8 bytes a 0.

Para calcular el valor de un número en coma flotante de 64 bits como los que hay en los ejemplos de este documento, a menudo sucede que el número de dígitos que soportan las calculadoras hexadecimales es menor de lo que es necesario, por lo que una forma alternativa de calcularlo sería:

Ahora bien, otra forma de representar el número en coma flotante es:

$$N = -1^S * 2^{(E-1023)} * (1 + MANTISA * 2^{-52})$$

Supongamos que enviamos a la pantalla el número 00 00 00 00 74 5D 40 41, que ya está en el formato de salida, es decir, primero el byte menos significativo. El exponente con el signo (en este caso positivo) lo forman los 3 nibbles más altos: 414H. Basándonos en que el valor 1023 es 3FFH,  $e^{(E-1023)}$  da un resultado de  $2^{(414H-3FF)} = 2^{21}$ .

La mantisa tendrá el valor 05D740000000H, lo que es igual a decir  $5D74H * 2^{32}$ . Por tanto:

$$N = 2^{21} * (1 + 5D74H * 2^{32} * 2^{-52})$$

$$N = 2^{21} + 2^{21} * 5D74H * 2^{32} * 2^{-52}$$

$$N = 2^{21} + 5D74H * 2^1$$

$$N = 2145000$$

Para calcular un número en coma flotante a partir de un entero se debe hacer la mantisa cuanto más grande, mejor y luego ajustar el exponente.

## TRANSMISION

Hay que tener en cuenta que siempre se transmite primero el byte menos significativo (LSB). Al final de las estructuras hay un byte de control que realmente no se lee, sino que solo se usa para reforzar el sistema de comunicaciones, ya que en caso de que un paquete corto no se recibiera correctamente, el PC debería preguntar a la pantalla el checksum del último paquete correcto recibido, y si fueran diferentes, intentar un reenvío. Si el último paquete con ACK recibido fuera idéntico al que no fue recibido (cosa que suele suceder), el PC, al pedir el último checksum, comprobaría que es igual al del paquete no recibido, y por lo tanto consideraría que el paquete fue recibido correctamente pero no así el ACK, con lo que no se produciría el reenvío aun cuando el paquete fue recibido incorrectamente por la pantalla.

Este efecto no tiene importancia en operaciones de asignación, pero sí en todas las otras.

Para evitar estos casos se ha previsto que al final de la/s la estructura/s haya un byte cuyo valor debería ser cambiado en cada nueva transmisión. Se recomienda un contador que se incremente cada vez en 15H respecto el valor que tenía en la última transmisión.

Con el byte de control cambiando de un paquete al siguiente, ya no habría posibilidad que se confundieran los checksum de dos paquetes idénticos y consecutivos.

Por otra parte es imprescindible tener en cuenta la recepción del ACK y el sistema de reenvío descrito en el DTP (ver el manual del DTP), ya que en caso de una mala recepción del ACK, podría suceder que una misma operación se realizara tantas veces como reenvíos haya en una pantalla, mientras que debería haberse hecho una sola vez.

### EJEMPLO:

En el siguiente ejemplo, se asigna:

1. El *string* 'PRODUCTO' en la variable A
2. El número 1 sumado a la variable B
3. El número 2145000 en la variable C
4. El número 13406.25 en la variable D
5. El número 1 restado a la variable E
6. Se ponen las demás variables a 0 (no es necesario ya que está permitido enviar desde 1 a 26 estructuras, pero sirve para el ejemplo)

```
16 0C 01 01 2E 00 00 50 52 4F 44 55 43 54 4F 81 00 00 00 00 00 00 00 00 F0 3F
42 00 00 00 00 00 74 5D 40 41 43 00 00 00 00 00 20 2F CA 40 C4 00 00 00 00
00 00 00 F0 3F 45 00 00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00 00
47 00 00 00 00 00 00 00 00 00 48 00 00 00 00 00 00 00 00 00 49 00 00 00
00 00 00 00 00 4A 00 00 00 00 00 00 00 00 00 4B 00 00 00 00 00 00 00 00
4C 00 00 00 00 00 00 00 00 00 4D 00 00 00 00 00 00 00 00 00 4E 00 00 00
00 00 00 00 00 4F 00 00 00 00 00 00 00 00 00 50 00 00 00 00 00 00 00 00
51 00 00 00 00 00 00 00 00 00 52 00 00 00 00 00 00 00 00 00 53 00 00 00
00 00 00 00 00 54 00 00 00 00 00 00 00 00 00 55 00 00 00 00 00 00 00 00
56 00 00 00 00 00 00 00 00 00 57 00 00 00 00 00 00 00 00 00 58 00 00 00
00 00 00 00 00 59 00 00 00 00 00 00 00 00 00 15 25 10
```

En detalle:

16	SYN
0C 01	Longitud (0x010C)
01	Nº pantalla
2E	PUTVAR
00 00	Variable A string
50 52 4F 44 55 43 54 4F	'PRODUCTO'
81 00	Variable B suma
00 00 00 00 00 00 F0 3F	1
42 00	Variable C asignación
00 00 00 00 74 5D 40 41	2145000
43 00	Variable D asignación
00 00 00 00 20 2F CA 40	13406.25
C4 00	Variable E resta
00 00 00 00 00 00 F0 3F	1
45 00 00 00 00 00 00 00 00 00	Variable F asignación 0
46 00 00 00 00 00 00 00 00 00	Variable G asignación 0
47 00 00 00 00 00 00 00 00 00	Variable H asignación 0
48 00 00 00 00 00 00 00 00 00	Variable I asignación 0
49 00 00 00 00 00 00 00 00 00	Variable J asignación 0
4A 00 00 00 00 00 00 00 00 00	Variable K asignación 0
4B 00 00 00 00 00 00 00 00 00	Variable L asignación 0
4C 00 00 00 00 00 00 00 00 00	Variable M asignación 0
4D 00 00 00 00 00 00 00 00 00	Variable N asignación 0
4E 00 00 00 00 00 00 00 00 00	Variable O asignación 0
4F 00 00 00 00 00 00 00 00 00	Variable P asignación 0
50 00 00 00 00 00 00 00 00 00	Variable Q asignación 0
51 00 00 00 00 00 00 00 00 00	Variable R asignación 0
52 00 00 00 00 00 00 00 00 00	Variable S asignación 0
53 00 00 00 00 00 00 00 00 00	Variable T asignación 0
54 00 00 00 00 00 00 00 00 00	Variable U asignación 0
55 00 00 00 00 00 00 00 00 00	Variable V asignación 0
56 00 00 00 00 00 00 00 00 00	Variable W asignación 0
57 00 00 00 00 00 00 00 00 00	Variable X asignación 0
58 00 00 00 00 00 00 00 00 00	Variable Y asignación 0
59 00 00 00 00 00 00 00 00 00	Variable Z asignación 0
15	Control transmisión
25 10	Checksum 0x1025

## **Orden GETVARS (0x2F)**

Pide variables de la memoria de la pantalla que le son enviadas mediante un paquete **SEND** que contiene 26 estructuras, las cuales corresponden a las variables ordenadas de la 'A' a la 'Z'. Cada una de ellas tiene 10 bytes con el siguiente formato:

### ***bytes [0,1]***

1 bit        Indica si es *string*.

15 bits     Reservado

### ***bytes [2-9]***

Valor        Valor que tiene la variable. Puede ser un número con el formato IEEE de 64 bits o bien string de 8 caracteres o menos, que puede no terminar en 0.

### **Ejemplo:**

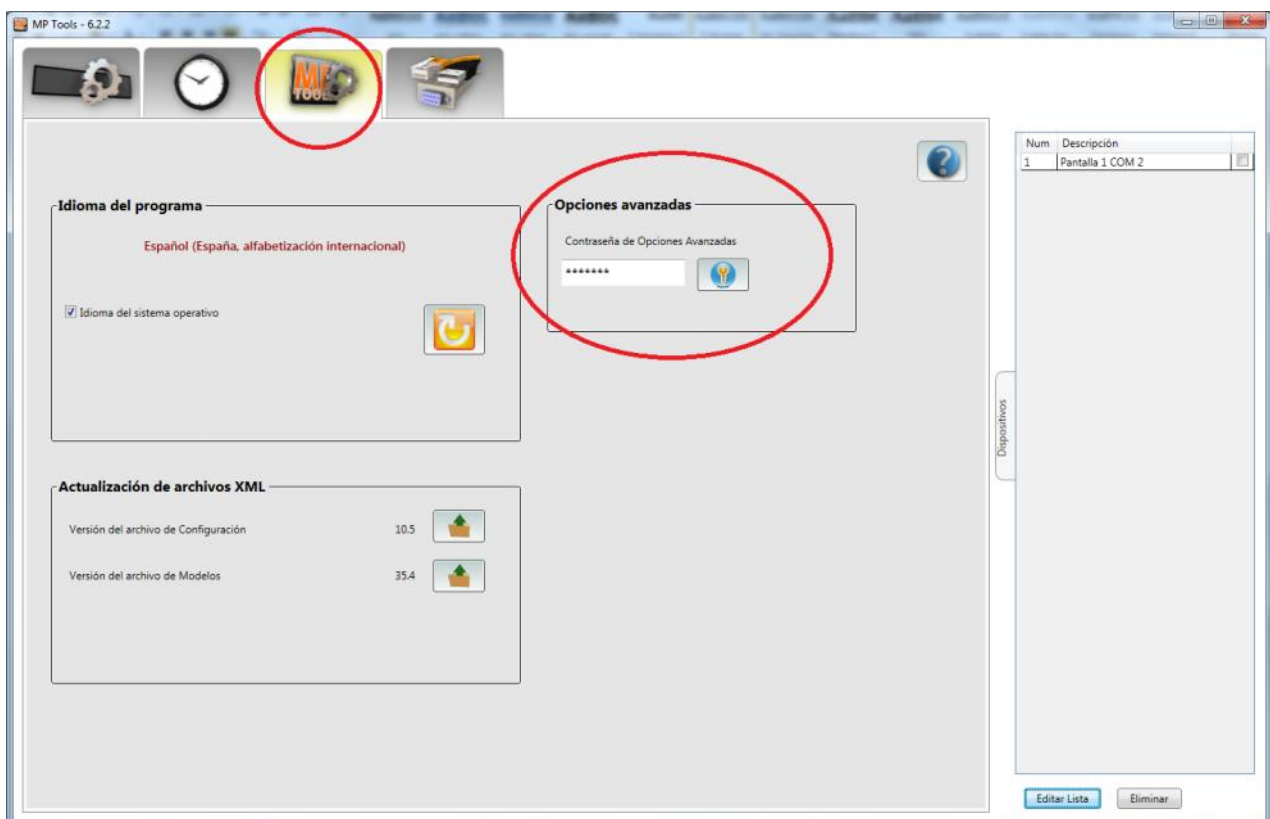
Enviado a Dispositivo: 16 07 00 01 2F 4D 00

## 2.5. Test con el Software MP Tools.

El Software MP Tools, permite poder enviar directamente órdenes a los dispositivos conectados y observar en detalle, tanto las tramas enviadas como las respuestas del dispositivo.

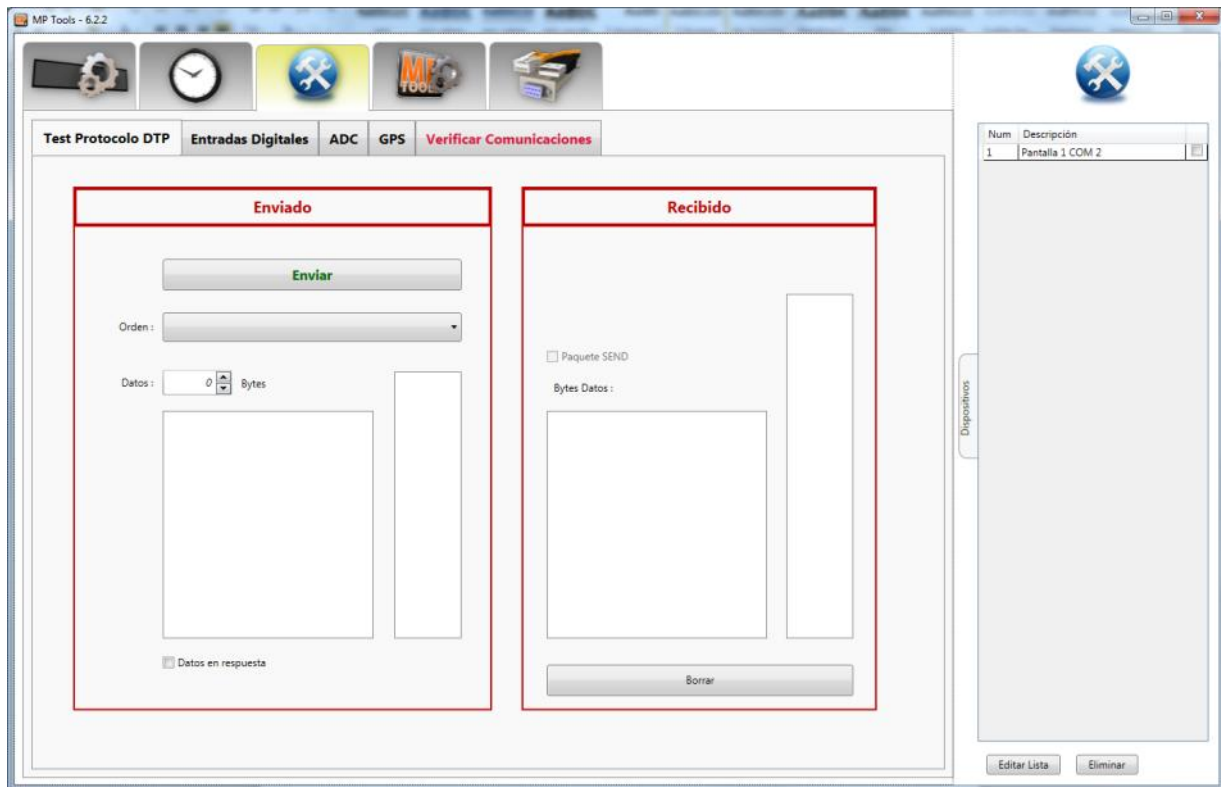
Para detalles sobre el procedimiento de instalación y configuración del software, referirse al manual específico del mismo.

Para trabajar con dicha opción se deben desbloquear las **Opciones Avanzadas** del programa, en caso que no esté habilitada ésta opción. Para ello en la pestaña **Configuración de la aplicación**, y en la casilla correspondiente, introducir la contraseña de **Opciones Avanzadas** y pulsar el botón de **Validar Contraseña**. La contraseña es **INT8932**.



Una vez realizada esta tarea, el programa guarda en memoria la opción para el usuario, de forma que no tendremos que volver a realizar este procedimiento al arrancar el programa de nuevo.

Una vez habilitadas las **Opciones Avanzadas**, se puede ver la pestaña homónima en el menú general del programa. Una vez en ella, situarse en la primera sub-pestaña (**Test Protocolo DTP**).



En la figura pueden verse dos zonas: **Enviado** y **Recibido**.

En el sector **Enviado** es donde debe seleccionarse la Orden que se quiere enviar al dispositivo mediante el Combo-box.

No hay que indicar la dirección del dispositivo, ni la longitud del paquete ni el Checksum, ya que de esto se encarga MP Tools.

Es posible enviar la Orden sin datos o añadir datos al paquete indicando el número de bytes y luego dando valor a los mismos. Es posible introducir el valor de los bytes en Decimal, Hexadecimal o ASCII.

También es posible indicar si se espera un paquete SEND luego de recibir ACK.

Test Protocolo DTP	Entradas Digitales	ADC	GPS	Verificar Comunicaciones
--------------------	--------------------	-----	-----	--------------------------

### Enviado

Orden: 0x12 - GETVER

Datos:  Bytes

0x16  
0x07  
0x00  
0x01  
0x12  
0x30  
0x00

Datos en respuesta

### Recibido

Paquete SEND

Bytes Datos: 6

0	46	.	2E
1	196	?	C4
2	96	-	60
3	0	□	00
4	1		01
5	6		06

0x06  
0x00  
0x16  
0x0D  
0x00  
0xFE  
0xDC  
0x2E  
0xC4  
0x60  
0x00  
0x01  
0x06  
0x86  
0x02

Al pulsar en el botón **Enviar**, se transmitirá el paquete y el cuadro de la derecha del sector **Enviado** mostrará la trama completa enviada (con cabecera, datos y checksum).

Si no ha habido problemas en la comunicación y se ha recibido respuesta del dispositivo, ésta se mostrará en el sector **Recibido**. Aquí también se puede observar con detalle la trama completa recibida.

Con el botón **Borrar** del sector **Recibido** borramos las tramas enviada y recibida y podemos realizar otro test.



### Enviado

Enviar

Orden : 0x12 - GETVER

Datos :  Bytes

0x16  
 0x07  
 0x00  
 0x01  
 0x12  
 0x30  
 0x00

Datos en respuesta

### Recibido

ACK

Paquete SEND

Bytes Datos : 6

0	46	.	2E
1	196	?	C4
2	96	.	60
3	0	□	00
4	1		01
5	6		06

0x06  
 0x00  
 0x16  
 0x0D  
 0x00  
 0xFE  
 0x0C  
 0x2E  
 0xC4  
 0x60  
 0x00  
 0x01  
 0x06  
 0x86  
 0x02

Borrar

## Anexo 1. Configuración por defecto de las Pantallas

Las pantallas de MP Electronics, en los parámetros que hacen referencia a las comunicaciones, tienen al salir de fábrica la siguiente configuración.

PARÁMETRO DE CONFIGURACIÓN	Valor por defecto
Dirección del Dispositivo	1
Dirección <i>Local/Cast</i> del Dispositivo	0
<b><i>Puerto Serie</i></b>	
Puerto Serie: Bauds	9600
Puerto Serie: Data Bits	8
Puerto Serie: Stop Bits	1
Puerto Serie: Paridad	Sin Paridad
<b><i>TCP / IP</i></b>	
Dirección IP	192.168.1.100
Máscara de Red	255.255.255.0
Puerto TCP	53
Puerta de Enlace	192.168.1.1
Servidor DNS Primario	192.168.1.100
Servidor DNS Secundario	192.168.1.100
Direccionamiento IP Dinámico. Cliente DHCP Habilitado	NO

Tabla 4: Configuración por defecto de las pantallas

## Anexo 2. Códigos de error Protocolo DTPM

Código		Significado
0	0x00	OK (operación efectuada correctamente)
1	0x01	Programa no encontrado
2	0x02	No se puede Editar (se está ejecutando un programa)
3	0x03	Archivo no encontrado
5	0x05	Programa seleccionado está actualmente en ejecución
6	0x06	Operación no permitida en ejecución
7	0x07	Orden desconocida
8	0x08	Archivo sin contenido (no se puede ejecutar)
9	0x09	Password Incorrecto
10	0x0A	Nombre demasiado largo
11	0x0B	Formato de Hora/Fecha incorrecto
12	0x0C	Formato de EDITFILE incorrecto
13	0x0D	Archivo mayor que espacio libre en directorio
14	0x0E	Archivo mayor que tamaño máximo
25	0x19	Datos no válidos
28	0x1C	Dispositivo ocupado
49	0x31	Puerto Digital no disponible
62	0x3E	No se ha podido crear MBR (Sistema de Archivos)
63	0x3F	No se ha podido formatear el directorio
64	0x40	Directorio vacío
65	0x41	Sistema de archivos no habilitado
66	0x42	No se pudo borrar el archivo
67	0x43	No se pudo inicializar (Borrar) el Sistema de Archivos
68	0x44	Longitud solicitada demasiado grande
69	0x45	Error Sistema de Archivos (General)
70	0x46	Tamaño de archivo superior al espacio reservado
73	0x49	Archivo Corrompido/Dañado
74	0x4A	FASTEXEC no disponible

Tabla 5: Códigos de Error

### Anexo 3. Estructura de datos *DEVICE\_USER\_SETTINGS*

Esta estructura de datos aparece en las órdenes GET\_SETTINGS y PUT\_SETTINGS.

BYTE	DESCRIPCIÓN	VALORES POSIBLES
1	<b>CONTRASEÑA</b> Al recibir los datos de la pantalla con la orden GET_SETTINGS, este campo tiene todos los Bytes a 0.  Al enviar nuevos valores de configuración con la orden PUT_SETTINGS, debe tener el valor "INT8932" en ASCII y con los bytes sobrantes a NULL (0x00). En caso contrario, la pantalla devolverá Error de Password Incorrecto (0x09).	'I' (0x49)
2		'N' (0x4E)
3		'T' (0x54)
4		'8' (0x38)
5		'9' (0x39)
6		'3' (0x33)
7		'2' (0x32)
8		0x00
9		0x00
10		0x00
11		0x00
12	Idioma textos de ayuda	0 - 10
13	Reset al POWER-ON	0 - 1
14	No borrar con STOP	0: Borrar ; 1: No Borrar
15	Tiempo entre modos de aparición en pantalla.	1 - 200 (cuartos de segundo)
16	Velocidad de los efectos (excepto el "Inmediato")	1 - 100 (píxeles por segundo)
17	En caso de Luminosidad Automática, indica el % mínimo de luminosidad de los LEDs que mostrará el display independientemente de la lectura del sensor	1 - 100 %
18	En caso de Luminosidad Automática, indica el % máximo de luminosidad que captará el sensor	1 - 100 %
19	Porcentaje de la luminosidad. (Automática o manual)	0 para AUTOMÁTICA 1 a 100 % para valor fijo
20	Filtro de la Sonda de Luminosidad	0 - 255 segundos
21	RTCC: Horario de Verano	0 para DESACTIVADO 64 para ACTIVADO
22	RTCC: Tipo Horario de Verano	1 : EUROPEO
23	RTCC: Huso horario	-12 a +14 horas respecto a UTC
24	Aviso por nivel bajo de batería	0 - 1
25	RTCC: SoftTrimm	-7 a +8 ppm
26	Temperatura de activación del ventilador	-120°C a +120°C
27	Modo anti condensación	0 - 1
28	Tiempo de activación del ventilador (Modo anti condensación)	0 - 255 minutos
29	Período en que se activa el ventilador (Modo anti condensación)	0 - 1440 minutos
30		
31	Caracteres de ancho variable	0 - 1
32	Modo (Efecto) de aparición de los textos en la pantalla	240 Inmediato; 224 Correr
33	Control Iluminación Cabecera	0 - 1
34	Nivel de activación de la Iluminación Cabecera	5 - 85 %
35	<i>(futura ampliación)</i>	
36	<i>(futura ampliación)</i>	

Estructura *DEVICE\_USER\_SETTINGS*

Para usar las órdenes GET\_SETTINGS y PUT\_SETTINGS hay que seguir el siguiente procedimiento:

1. Enviar a la pantalla la orden GET\_SETTINGS y recibir el paquete con la estructura de datos con los valores de la configuración actual de la pantalla.
2. Modificar en esta estructura solamente los campos que deseemos, manteniendo inalterados los valores de los demás.
3. Modificar los 11 primeros bytes para que tengan el valor de la contraseña indicada. 'I' 'N' 'T' '8' '9' '3' '2' y los 4 restantes bytes con valor 0.
4. Enviar a la pantalla la orden PUT\_SETTINGS con la estructura original con las modificaciones descritas en los apartados 2 y 3.
5. Luego de recibir ACK en caso de operación realizada correctamente, esperar unos 2 segundos a que la pantalla se reinicie y se apliquen los cambios realizados en la configuración.

## REVISIONES

Revisión 1.0 – Documento inicial

Revisión 1.1 – Variables Alfanuméricas.

Revisión 1.2 – Orden *STOP AND CLEAR*.

Revisión 1.3 – Modificaciones en orden PUTPORT.

Revisión 1.4 – Ordenes GET\_SETTINGS y PUT\_SETTINGS.