MP ELECTRONICS

PROTOCOLO MODBUS

PARA PANTALLAS LED

Manual Técnico

Revisión del documento 2.0

Compatible con pantallas LED con Firmware 15.8 o superior.

Modbus

Fecha: 21/02/2023

Nombre de fichero: Protocolo Modbus para pantallas LED. *Revisión 2.0*

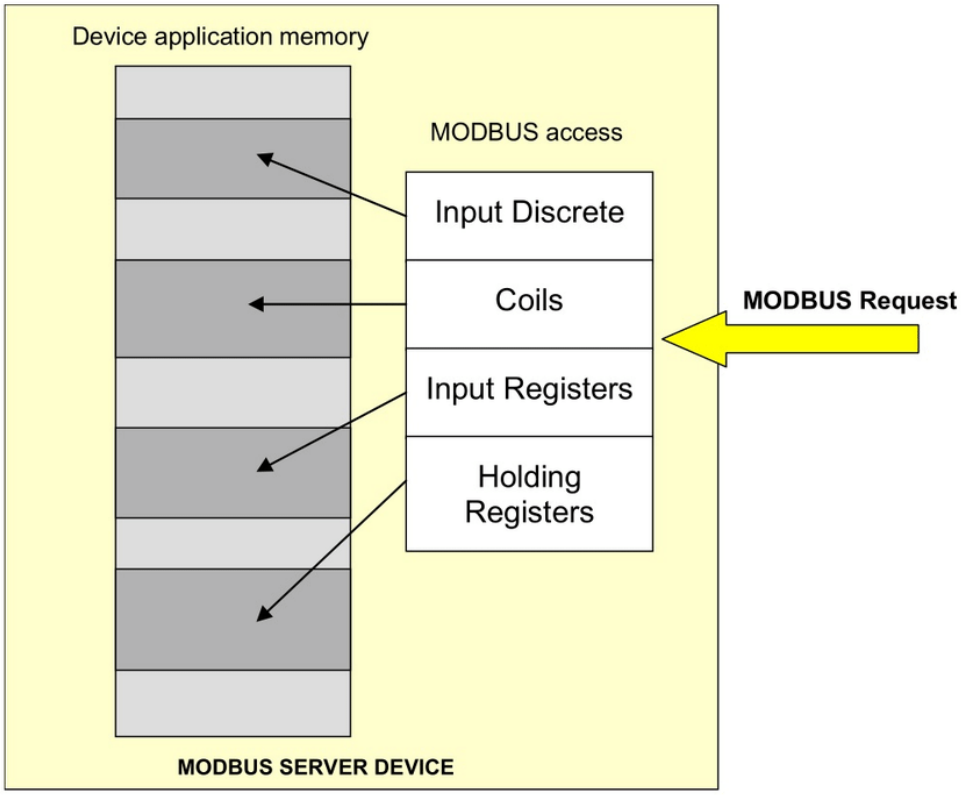
TABLA DE CONTENIDO

1. DESCRIPCIÓN

Se puede interactuar mediante el protocolo Modbus (tanto en modo RTU como en modo TCP/IP) con las pantallas LED de MP Electronics. Este protocolo es muy utilizado en el entorno industrial y fácilmente adaptable a muchos tipos de instrumentación, como por ejemplo los Controladores Lógicos Programables (PLC).

Su funcionamiento se basa en la lectura y escritura de unidades de memoria del dispositivo esclavo mediante el uso de diferentes órdenes o comandos (*Requests*) estandarizados. Hay cuatro tipos diferentes de unidades de memoria y cada una tiene su utilidad:

- Bits de solo lectura (*Status* de tipo *Discrete input*).
- Bits de lectura/escritura (*Status* de tipo *Coil*).
- Registros de lectura de 16 bits (*Input Registers*).
- Registros de lectura/escritura de 16 bits (*Holding Registers*).



1.1. COMANDOS MODBUS IMPLEMENTADOS

Las pantallas de MP Electronics soportan las siguientes funciones de Modbus como **esclavas**:

(01) Read Coil Status	***Modbus RTU & Modbus TCP/IP***
(02) Read Input Status	***Modbus RTU & Modbus TCP/IP***
(03) Read Holding Registers	***Modbus RTU & Modbus TCP/IP***
(04) Read Input Registers	***Modbus RTU & Modbus TCP/IP***
(05) Write Single Coil	***Modbus RTU & Modbus TCP/IP***
(06) Write Single Holding Register	***Modbus RTU & Modbus TCP/IP***
(15) Write Multiple Coils	***Modbus RTU & Modbus TCP/IP***
(16) Write Multiple Holding Registers	***Modbus RTU & Modbus TCP/IP***

Mientras que soportan las siguientes funciones de Modbus como **máster**:

(02) Read Input Status *	***Modbus RTU***
(03) Read Holding Registers	***Modbus RTU***

** Aunque implementada en el código, no tiene funcionalidad real por el momento.*

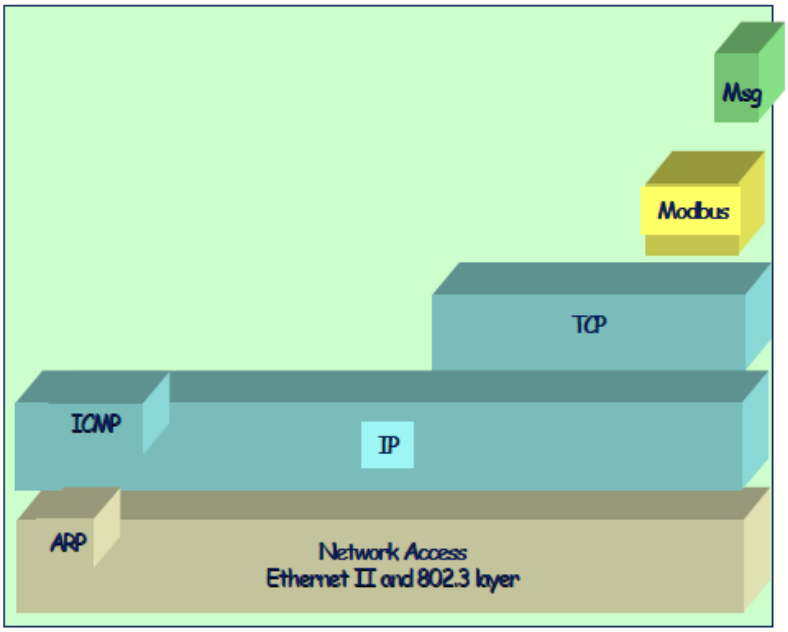
1.2. PROTOCOLO MODBUS-RTU

El protocolo Modbus en modo RTU utiliza los silencios en la línea de transmisión para indicar los inicios y finales del mensaje. Se considera un silencio el tiempo igual o mayor al necesario para transmitir 3,5 bytes. Para cada velocidad de transmisión le corresponde un tiempo de silencio especifico. Una vez realizada la transmisión de un mensaje, no se puede iniciar la transmisión de otro hasta que no haya transcurrido el tiempo de silencio necesario (3,5 veces el tiempo de transmisión de un byte).

Con este protocolo, la pantalla puede trabajar en modo esclavo o máster. Una vez recibido un mensaje con la dirección de la misma, se devolverá un mensaje con el resultado de la transmisión.

1.3. PROTOCOLO MODBUS-TCP/IP

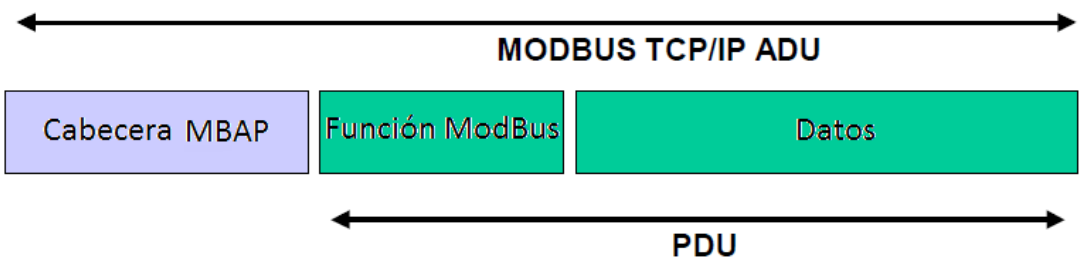
El protocolo Modbus en modo TCP/IP (en adelante Modbus-TCP) es una variante del Protocolo Modbus para comunicaciones sobre redes TCP/IP, realizando las conexiones a través del puerto TCP 502.



Con este protocolo, la pantalla trabaja en modo esclavo para Modbus (Servidor para TCP). Las tramas son igual a las del protocolo Modbus en modo RTU con las siguientes diferencias:

- El campo ID (Dirección del Dispositivo) de la trama Modbus-RTU es substituido por una cabecera llamada Cabecera MBAP, los campos de la cual se detallan en la Tabla 1.
- No hay CRC ni ningún código de control de errores, dado que los protocolos de capas inferiores se ocupan de tal tarea.

La estructura de la trama es la siguiente:



La cabecera MBAP (7 Bytes) contiene los siguientes campos:

Campo	***Bytes***	***Descripción***	***Cliente (Master)***	***Servidor (Slave)***
Transaction Identifier	2	Número de la transacción	Inicializado por Cliente	Reenviado por Servidor
Protocol Identifier	2	0 = Protocolo MODBUS	Inicializado por Cliente	Reenviado por Servidor
Length	2	Numero de Bytes de la trama que siguen a éste (de “Unit Identifier” al final)	Inicializado por Cliente	Inicializado por Servidor
Unit Identifier	1	Siempre a 255 o Unit ID Pantalla	Inicializado por Cliente	Reenviado por Servidor

Tabla 1: Cabecera MBAP del Protocolo Modbus-TCP

Una vez recibido un mensaje con la dirección de la misma, se devolverá un mensaje con el resultado de la transmisión.

2. MAPA COILS/INPUT STATUS/INPUT REGISTERS/HOLDING REGISTERS MODBUS

COILS		
Dirección	Descripción	
0	Orden STOP o ejecución de programa TOPALARM en modo “alarmas”.	
1	Ejecutar programa PRGM1 o ALARM1 en modo “alarmas”.	
2	Ejecutar programa PRGM2 o ALARM2 en modo “alarmas”.	
…	…	
119	Ejecutar programa PRGM119 o ALARM119 en modo “alarmas”.	

INPUT STATUS		
Dirección	Descripción	
0	Estado de ejecución de la pantalla.	

INPUT REGISTERS		
Dirección	Descripción	
0	ID del control de la pantalla.	
1	Versión del firmware. Byte alto.	
2	Versión del firmware. Byte bajo.	
3	Temperatura interna de la pantalla.	

HOLDING REGISTERS				
Dirección interna		Nombre	Descripción	
0x80		Nombre programa	Activar programa por nombre.	
0x100		Ejecutar Script	Ejecución de un Script de programa.	
0x200		Número programa	Activar programa por número.	
0x202		Formato variables	Definición global del formato de las variables.	
0x204		Variable A	Int16 / LSB Int32 / ASCII caracteres 1 y 2.	
0x205		Variable A	MSB Int32 / ASCII caracteres 3 y 4.	
0x206		Variable A	Punto decimal (Int16 & Int32) / ASCII caracteres 5 y 6.	
0x207		Variable A	ASCII caracteres 7 y 8 / Color.	
0x208		Variable B	Int16 / LSB Int32 / ASCII caracteres 1 y 2.	
0x209		Variable B	MSB Int32 / ASCII caracteres 3 y 4.	
0x20A		Variable B	Punto decimal (Int16 & Int32) / ASCII caracteres 5 y 6.	
0x20B		Variable B	ASCII caracteres 7 y 8 / Color.	
…	…	…	…	…
0x268		Variable Z	Int16 / LSB Int32 / ASCII caracteres 1 y 2.	

0x269		Variable Z	MSB Int32 / ASCII caracteres 3 y 4.	
0x26A		Variable Z	Punto decimal (Int16 & Int32) / ASCII caracteres 5 y 6.	
0x26B		Variable Z	ASCII caracteres 7 y 8 / Color.	

3. FUNCIONALIDADES DEL MODBUS CON PANTALLAS

Los distintos modos de funcionamiento de la comunicación Modbus se determinan por el **rol** como maestro o esclavo que ocupa cada parte; las **direcciones** de registros o status que se ven involucradas y el **tipo de acción**, pudiendo ser de lectura o escritura.

3.1. LECTURA DE DATOS INTERNOS

En estos casos la **pantalla funciona como esclava** y el maestro **leerá** datos de ella.

3.1.1. LECTURA DE INPUT STATUS

Orden: ***(02) Read Input Status***

Input Status: **0**

Es posible conocer si la pantalla está ejecutando un programa o no. En caso afirmativo, dicho status bit aparecerá como 1, en caso contrario, como 0.

3.1.2. LECTURA DE INPUT REGISTERS

Orden: ***(04) Read Input Registers***

Registro inicial: **0**

Es posible leer los siguientes registros de la pantalla:

Número de registro	Nombre	Valor
0	Versión de hardware	190 para CONPACK 196 para CONMUL
1	Versión de firmware	Número entero
2	Sub-versión de firmware	Número entero
3	Temperatura interna	En grados Celsius (ºC)

Tabla 2: Inputs Registers habilitados.

3.1.3. LECTURA DE COILS STATUS

Orden: ***(01) Read Coil Status***

Para el funcionamiento de ejecución de programas por *Coils* o alarmas, detallados más adelante, podemos leer los *Coil Status* para comprobación de que se han escrito anteriormente correctamente si se desea.

3.1.4. LECTURA DE HOLDING REGISTERS

Orden: ***(03) Read Holding Registers***

Para el funcionamiento detallado a continuación, podemos leer los *Holding Registers* para comprobación de que se han escrito anteriormente correctamente si se desea.

3.2. EJECUCIÓN DE PROGRAMA PREVIAMENTE GRABADO EN LA PANTALLA

3.2.1. EJECUCIÓN DE PROGRAMA POR NÚMERO USANDO REGISTROS

Orden: ***(06) Write Single Holding Register & (16) Write Multiple Holding Registers***

Registro: **200h**

Número de registros: **1**

En este caso, los programas ejecutados deberán haber sido grabados en la pantalla con el nombre: **PRGM** seguido del número (sin 0s a la izquierda).

Por ejemplo:

- Programa 1: “PRGM1”.
- Programa 3: “PRGM3”.
- Programa 27: “PRGM27”.
- Programa 149: “PRGM149”.

Para detener el programa en ejecución se debe seleccionar el programa número 0.

3.2.2. EJECUCIÓN DE PROGRAMA POR NÚMERO USANDO COILS

Orden: ***(05) Write Single Coil & (15) Write Multiples Coils****

Coil inicial: **0**

Número de Coils: **120**

En este caso, los programas ejecutados también deberán haber sido grabados en la pantalla con el nombre: **PRGM** seguido del número (sin 0s a la izquierda).

Para enviar por pantalla un programa se deberá activar el *Coil* correspondiente al número de programa. Esta funcionalidad está limitada a 119 programas, es decir, hasta el PRGM119.

Así, por ejemplo, para ejecutar por pantalla el programa 57 se deberá escribir un 1 en el *Coil* 57.

Para detener el programa en ejecución se puede escribir un 1 el *Coil* 0 o bien se puede escribir un 0 en el *Coil* previamente estipulado a 1.

No es necesario borrar el *Coil* previamente estipulado a 1 si se quiere ejecutar otro programa, esta operación se realiza de forma automática. Si se ha activado el *Coil* 1, por lo que se muestra por pantalla el PRGM1, y se requiere que se ejecute el PRGM2, se puede activar el *Coil* 2 sin necesidad de poner a 0 el *Coil* 1. Este se pondrá a 0 automáticamente.

Nota:** Solo se puede utilizar la orden ***(15) Write Multiples Coils si se manda un solo *Coil.*

3.2.3. EJECUCIÓN DE PROGRAMA POR NOMBRE

Orden: ***(06) Write Single Holding Register & (16) Write Multiple Holding Registers***

Registro: **80h**

Número de registros: **De 1 a 4.**

En este caso, el campo de datos contiene el nombre del programa que se quiere ejecutar en formato ASCII con un máximo de 8 caracteres y con un valor NULL que marca el final en caso de tener menos.

Por ejemplo, si queremos ejecutar el programa MPTEST gravado en origen en la pantalla, la PDU de Modbus enviada sería:

Func.	Dirección Registro		Num. Registros		Num. Bytes	Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6	Datos Byte 7	Datos Byte 8
10	00	80	00	04	08	4D	50	54	45	53	54	00	00

Ejemplo 1: PDU de MODBUS enviada para ejecutar el programa "MPTEST"

Observando en detalle el campo de datos:

Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6	Datos Byte 7	Datos Byte 8
4D	50	54	45	53	54	00	00
'M'	*'P'*	*'T'*	*'E'*	*'S'*	*'T'*	*NULL*	*NULL*

3.3. EJECUCIÓN DE SCRIPT ENVIADO A LA PANTALLA

Orden: ***(06) Write Single Holding Register & (16) Write Multiple Holding Registers***

Registro: **100h**

Número de registros: **De 1 a 124.**

Esta opción permite enviar el Script de un programa y que éste se ejecute de forma inmediata en la pantalla. Los detalles del Script de DTPM se muestran en el ***Anexo 2***.

Ejemplos:

- Ejecutar un programa que muestre el texto “Hola” en modo Inmediato:

Func.	Dirección Registro		Num. Registros		Num. Bytes	Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6
10	01	00	00	03	06	04	F0	48	6F	6C	61

Ejemplo 2: PDU de Modbus enviada para ejecutar un programa que muestre “Hola” en modo Inmediato

Observando en detalle el campo de datos:

Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6
04	F0	48	6F	6C	61
Modo Inmediato		*'H'*	*'o'*	*'l'*	*'a'*

- Ejecutar un programa que muestre el texto “V:” y la variable A.

Func.	Dirección Registro		Num. Registros		Num. Bytes	Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6	Datos Byte 7	Datos Byte 8
10	01	00	00	04	08	04	F0	56	3A	20	03	AB	41

Ejemplo 3: PDU de Modbus enviada para ejecutar un programa que muestre “V: “ y la variable A

Observando en detalle el campo de datos:

Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6	Datos Byte 7	Datos Byte 8
04	F0	56	3A	20	03	AB	41
Modo Inmediato		*'V'*	*':'*	*' '*	*VAR*		*'A'*

3.4. MODIFICACIÓN DE VARIABLES INTERNAS DE LA PANTALLA

4.1.1. MODIFICACIÓN DEL VALOR

Orden: ***(06) Write Single Holding Register & (16) Write Multiple Holding Registers***

Registro: **202h**

Número de registros: **De 1 a 124.**

Las Pantallas disponen de 26 variables internas que pueden representar números enteros o decimales, así como texto de hasta 8 caracteres.

Las variables se numeran de la letra “A” a la “Z” (sin incluir la 'Ñ').

En el ***Anexo 3*** se detalla los bytes del Script que se debe mandar a la pantalla para que muestre variables.

La dirección inicial, el registro 0202h, determina como serán interpretados los registros de las variables, es decir, si las variables son con o sin signo, de 16 o 32 bits o si por el contrario son alfanuméricas.

Registro *0202h*	**Formato**	**Valor Mínimo**	**Valor Máximo**
0 [1]	Entero 16 bits con signo	-32768	+32767
1	Entero 16 bits sin signo	0	65535
2	Entero 32 bits con signo	-2147483647	+2147483647

1 Por defecto el formato es Entero de 16 bits con signo

3	Entero 32 bits sin signo	0	4294967295
4	Formato Texto	*1 carácter*	*8 caracteres*

Tabla 3: Formato de los valores numéricos en función del Registro 0202h (514)

Cada variable hace uso de 4 registros. La relación de registros de las 26 variables se muestra en la tabla siguiente:

Variable		**Direcciones de Registros**			
		Valor 1	***Valor 2***	***Valor 3***	***Valor 4***
A	***0***	*0204*	*0205*	*0206*	*0207*
B	***1***	*0208*	*0209*	*020A*	*020B*
C	***2***	*020C*	*020D*	*020E*	*020F*
D	***3***	*0210*	*0211*	*0212*	*0213*
E	***4***	*0214*	*0215*	*0216*	*0217*
F	***5***	*0218*	*0219*	*021A*	*021B*
G	***6***	*021C*	*021D*	*021E*	*021F*
H	***7***	*0220*	*0221*	*0222*	*0223*
I	***8***	*0224*	*0225*	*0226*	*0227*
J	***9***	*0228*	*0229*	*022A*	*022B*
K	***10***	*022C*	*022D*	*022E*	*022F*
L	***11***	*0230*	*0231*	*0232*	*0233*
M	***12***	*0234*	*0235*	*0236*	*0237*
N	***13***	*0238*	*0239*	*023A*	*023B*
O	***14***	*023C*	*023D*	*023E*	*023F*
P	***15***	*0240*	*0241*	*0242*	*0243*
Q	***16***	*0244*	*0245*	*0246*	*0247*
R	***17***	*0248*	*0249*	*024A*	*024B*
S	***18***	*024C*	*024D*	*024E*	*024F*
T	***19***	*0250*	*0251*	*0252*	*0253*
U	***20***	*0254*	*0255*	*0256*	*0257*
V	***21***	*0258*	*0259*	*025A*	*025B*
W	***22***	*025C*	*025D*	*025E*	*025F*
X	***23***	*0260*	*0261*	*0262*	*0263*
Y	***24***	*0264*	*0265*	*0266*	*0267*
Z	***25***	*0268*	*0269*	*026A*	*026B*

Tabla 4: Relación de Registros correspondientes a las variables internas de las pantallas.
Valores en Hexadecimal

Donde cada valor corresponde a:

- Valor 1: Word (16 bits) Bajo.
- Valor 2: Word (16 bits) Alto. Se ignora si el formato es 16 bits.
- Valor 3: Posición del punto decimal. Permite activar el punto decimal del valor enviado.

Valor 3	Posición del punto decimal en la variable
0	**00000000000**
1	**0000000000.0**
2	**000000000.00**
…	…
10	**0.0000000000**
> 10	**0.0000000000**

Tabla 5: Posición del punto decimal para cada valor del registro correspondiente a "Valor 3"

- Valor 4: Color de la variable. Se detalla en el apartado 4.1.2.

Ejemplos:

- Asignar el valor 10489 a la variable A.

Func.	Dirección Registro		Num. Registros		Num. Bytes	Valor 1	
10	02	04	00	01	02	28	F9

Ejemplo 4: PDU de Modbus enviada para asignar el valor 10489 a la variable A

- Asignar el valor -10489 a la variable A.

Func.	Dirección Registro		Num. Registros		Num. Bytes	Valor 1	
10	02	04	00	01	02	D7	07

Ejemplo 5: PDU de Modbus enviada para asignar el valor -10489 a la variable A

- Asignar el valor 3,4789 a la variable A.

Func.	Dirección Registro		Num. Registros		Num. Bytes	Formato		Dummy		Variable A Valor 1		Variable A Valor 2	
10	02	02	00	05	0A	00	01	00	00	87	E5	00	00
Variable A Valor 3													
00	04												

Ejemplo 6: PDU de Modbus enviada para asignar el valor 3,4789 a la variable A.

- Asignar el texto "1234JKR" a la variable A y el texto "*AB-12-YZ*" a la variable B.

Func.	Dirección Registro		Num. Registros		Num. Bytes	Formato		Dummy		Variable A Valor 1		Variable A Valor 2	
10	02	02	00	0A	14	00	04	00	00	31	32	33	34
Variable A Valor 3		Variable A Valor 4		Variable B Valor 1		Variable B Valor 2		Variable B Valor 3		Variable B Valor 4			
4A	4B	52	00	41	42	2D	31	32	2D	59	5A		

Ejemplo 7: PDU de Modbus enviada para asignar el texto "1234JKR" y "AB-12-YZ" a la variable A y B.

4.1.2. MODIFICACIÓN DEL COLOR

El color de la variable se define en el Script del programa. Aun así, es posible forzar solo para la variable un color diferente al del Script mediante el segundo byte del "Valor 4" de la Tabla 4, que le llamaremos color intrínseco de la variable en adelante.

Tiene preferencia el color intrínseco de la variable al color estipulado en el Script del programa.

Por defecto, este color intrínseco de la variable es 0, con lo que la variable se representará con el color del Script.

Este color intrínseco de la variable se almacena en la memoria no volátil, al igual que el propio valor de la variable. Así entonces, una vez modificado el color intrínseco de la variable esta siempre lucirá de ese color le mandes el Script que le mandes. Se debe volver a estipular a 0 el color intrínseco de la variable para que vuelva a obedecer a los colores mandados por el Script.

La tabla de valores para el color intrínseco es la siguiente:

0	Color por defecto del Script
1	Rojo
2	Verde
3	Ámbar
4	Azul
5	Magenta
6	Cían
7	Blanco

Tabla 6: Relación de colores.
Valores en Hexadecimal

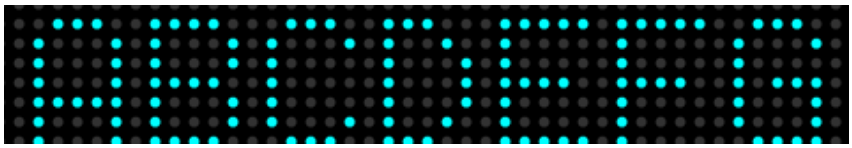
En efecto, al usar la prestación para seleccionar el color se sacrifica un byte de las variables alfanuméricas, aunque solamente si se desea trabajar modificando el color de las variables alfanuméricas. Si no se desea modificar el color se puede trabajar con las variables alfanuméricas de 8 caracteres.

Ejemplo:

Func. (hex)	Dirección Registro		Num. Registros		Num. Bytes	Datos Byte 1	Datos Byte 2	Datos Byte 3	Datos Byte 4	Datos Byte 5	Datos Byte 6	Datos Byte 7	Datos Byte 8
10	02	04	00	04	08	41	42	43	44	45	46	47	06

Ejemplo 8: PDU de Modbus enviada para asignar el texto "ABCDEFG" y el color cían a la variable A.

Obtendremos:



3.5. ACTIVACIÓN DE SALIDAS DÍGITALES

Orden: ***(06) Write Single Holding Register & (16) Write Multiple Holding Registers***

Registro: **1FBh**

Número de registros: **5.**

Esta funcionalidad sirve para activar o desactivar el periférico opcional de dos salidas de relés.

0x1FB: Salida

Indica la salida digital que se activará. Puede ser la 1 o la 2.

0x1FC: Modo

Puede ser (0) Siempre apagado, (1) Siempre encendido o (2) Secuencial.

0x1FD: Número de iteraciones

Si el modo es secuencial, indica el número de ciclos de tiempo activa y tiempo inactivo que realizará.

0x1FE: Tiempo activa

Si el modo es secuencial, indica el tiempo en cuartos de segundo que la salida digital permanecerá activa en cada iteración.

0x1FF: Tiempo inactiva

Si el modo es secuencial, indica el tiempo en cuartos de segundo que la salida digital permanecerá inactiva en cada iteración.

3.6. EJECUCIÓN DE PROGRAMAS DE ALARMAS

Orden: ***(05) Write Single Coil & (15) Write Multiples Coils****

Coil inicial: **0**

Número de Coils: **120**

Este modo de trabajo sirve para reproducir programas de alarma en función de las alarmas, *Coils*, que se activen. La principal diferencia con el modo de "Ejecución de programa por número usando Coil" del apartado 2.2.2 es que es posible ejecutar más de un programa de forma secuencial y cíclica. Nótese que ambos modos comparten *Coils,* con lo que son modos de trabajo excluyentes.

En este caso, los programas ejecutados deberán haber sido grabados en la pantalla con el nombre: **ALARM** seguido del número (**sin** 0s a la izquierda, por ejemplo "ALARM3" o "ALARM 34") hasta llegar al programa 119. Se aconseja guardarlos por orden de prioridad, siendo la ALARM1 la más importante y la ALARM119 la menos importante. También se dispone de una alarma prioritaria por encima de todas, llamada **TOPALARM**.

Así entonces, el *Coil* 0 corresponde al programa TOPALARM, mientras que el *Coil* 1 al programa ALARM1, el *Coil* 2 a la ALARM2 y así sucesivamente.

También podemos añadir un programa **DEFAULT,** que puede contener un mensaje del estilo de "SIN INCIDENCIAS" o incluso puede estar vacío simulando que la pantalla está detenida, para mostrarse por defecto cuando no haya ninguna alarma activada.

Finalmente, este modo también permite mandar un FASTEXEC. A continuación, se detalla el modo de funcionamiento general:

TOPALARM está activo

- Solo se reproduce TOPALARM hasta que la correspondiente alarma se apaga.
- Ignora la orden FASTEXEC.

Alguna ALARMXX está activa

- Se reproducen de manera consecutiva las alarmas activas.
- Es posible mandar un FASTEXEC, aunque solamente se reproducirá una sola vez, ya el panel seguirá ejecutando las ALARMXX que esten activas.

Ninguna alarma está activa

- Se reproduce el programa DEFAULT hasta que se active alguna alarma.
- En caso de enviarse un FASTEXEC, este se reproducirá hasta que se active alguna alarma.

Es necesario activar este modo de trabajo en la configuración del panel en el apartado Modbus a través del MP Tools. Por defecto esta desactivado de fábrica.

Además, también podrá configurar con el MPTools la forma en la que aparecen las nuevas alarmas:

- Continuación: La reproducción de alarmas sigue el mismo orden que tenía. Cuando le llegue, de forma natural, el turno a la alarma que recientemente se ha activado, esta se reproducirá.

- Orden de prioridad: En cuanto una nueva alarma se active, las alarmas volverán a empezar a reproducirse de la más prioritaria a la menos.

- Última activa: En cuanto una alarma nueva se active, las alarmas empezarán a reproducirse comenzando por la que acaba de activarse y, seguidamente, se seguirán reproduciendo por orden de prioridad. **Nota:** Este modo solo funciona si se trabaja con la orden ***(05) Write Single Coil*** , es decir de *Coil* en *Coil*.

Este modo de ejecución secuencial de programas tiene la particularidad de que cuando se mande un STOP a la pantalla para, por ejemplo, guardar programas nuevos, no se ejecutará ninguna alarma nueva, se activen o no, hasta que se le mande la orden de ejecutar un FASTEXEC o ejecutar un programa guardado en la memoria o se apague y se vuelva a encender la pantalla.

Por último, este modo no permite ejecutar programas previamente guardados en la memoria de la pantalla además de las alarmas o el FASTEXEC. Si ejecuta un programa guardado en la memoria se ejecutará una vez y saltará automáticamente al programa DEFAULT.

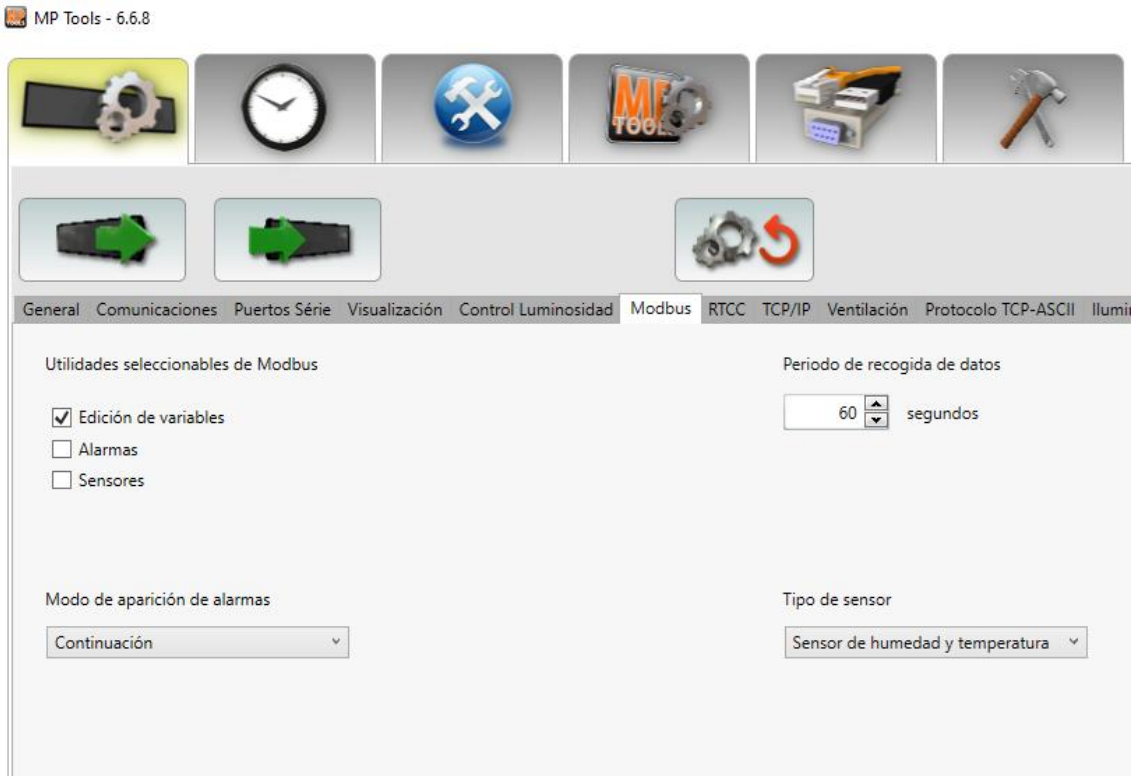
3.7. COMUNICACIÓN CON SENSORES MODBUS

En esta modalidad la pantalla funciona como máster. De esta forma pedirá continuamente al sensor la información de temperatura o de temperatura y humedad.

La información aparecerá directamente en las variables W (temperatura) y X (humedad) con un detalle en pantalla de un decimal.

La frecuencia con la que pide los datos será por defecto de 60 segundos, pero podrá modificarse a través de la configuración pudiendo tener valores entre 4 segundos y 2 minutos.

Finalmente, cabe tener en cuenta que por defecto esta prestación estará inhabilitada para evitar que consuma recursos. Para habilitarla deberá seleccionar la opción “Sensores” del parámetro “Utilidades seleccionables de Modbus”.



Detalle de la pestaña Modbus en MP Tools

ANEXO 1. CONFIGURACIÓN POR DEFECTO DE LAS PANTALLAS

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

PARÁMETRO DE CONFIGURACIÓN	Valor por defecto
Dirección del Dispositivo	1
Dirección *LocalCast* del Dispositivo	0
Puerto Serie	
Puerto Serie: Bauds	9600
Puerto Serie: Data Bits	8
Puerto Serie: Stop Bits	1
Puerto Serie: Paridad	Sin Paridad
TCP / IP	
Dirección IP	192.168.1.100
Máscara de Red	255.255.255.0
Puerto TCP para Modbus	502
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
Modbus	
Utilidades seleccionables de Modbus	Edición de variables
Modo de aparición de alarmas	Continuación
Periodo de recogida de datos	60 segundos
Tipo de sensor	Sensor de temperatura

Tabla 7: Configuración por defecto de las pantallas de MP Electronics

ANEXO 2. SCRIPT DTPM

El Script son los datos que se le manda a la pantalla para que represente la información deseada. Este Script está formado principalmente por Tokens y por texto a mostrar. Si queremos representar en color amarillo el texto “HOLA”, el Token seria la información que estipula el color amarillo mientras que el texto a mostrar sería el “HOLA”. Los Tokens están formados por dos bytes, siendo el primer byte el Pretoken y el segundo byte el Token.

Cada Script puede contener muchos Tokens que se ejecutan secuencialmente, y pueden ser de 4 tipos diferentes: DATO (parámetros de ejecución), MODO (los modos de aparición de los textos en la pantalla) 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 la pantalla con un MODO).

Se aconseja que se coloquen los códigos DATO, MODO, TIEMPO y EFECTO, por este orden.

Los programas deben terminar siempre 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**	***n***	**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	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

DATO	DESCRIPCION	*n*	Pretoken	Token
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** indica si la línea es simple (1) o doble (2).	1 – Líneas Totales	0x03	0xC7
Sincronismo	Todas las líneas o ventanas se ejecutan de forma paralela.		0x03	0xC9
No sincronismo	Todas las líneas o ventanas se ejecutan de forma secuencial.		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** >	Establecemos 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* ***AUTOMÁTICO***	0x03	0xD0

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 ***Hora : Minutos***	0x01	0xA7
Temperatura (**ºC**)	4 caracteres Temperatura y los símbolos ‘ º ’ 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

EFECTO	DESCRIPCION	Pretoken	Token
Monoespaciado	Todos los caracteres ocupan los mismos pixeles de anchura	0x02	0x20
Espaciado variable	Anchura de carácter variable	0x02	0x21
Parpadeo < **n** >	Parpadear n veces cuando el display termine de procesar la ventana	0x02	0xB0
Borrar	Borrar el texto escrito cuando el display termine de procesar la ventana	0x02	0xB2

ANEXO 3. REPRESENTACIÓN DE VARIABLES EN PANTALLAS

Para representar las variables haremos uso del Pretoken DATO (03h) seguido del Token VAR (ABh) y después el nombre de una de las 26 variables [A,Z].

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.

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 000000001
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
		1.000000000

En caso de apagar, o desconectar la corriente, las variables conservan su valor y color por defecto, aunque es posible configurar el equipo para que siempre se inicialicen a 0 y con el color por defecto. Las variables no se alterarán si se para o vuelve a empezar la ejecución de un Script.

ANEXO 4. EJEMPLOS PDU DE MODBUS RTU Y MODBUS TCP

Recordemos que la PDU de Modbus es la Unidad de Datos del Protocolo, y que ésta se encapsula en la trama del protocolo, cuyo formato depende del modo Modbus (RTU o TCP/IP).

Por ejemplo, en caso de trabajar en modo Modbus RTU, para el caso del ejemplo 4 y dirección de la pantalla 01, la trama sería la siguiente:

ID	**PDU**												**CRC**	
ID[2]	Func.	Dirección Registro		Num. Registros		Num. Bytes	Valor 1		Valor 2		Valor 3		CRC	
01	10	02	04	00	03	06	28	F9	00	00	00	00	36	D1

Ejemplo 9: Trama de Modbus-RTU enviada para asignar el valor 10489 a la variable A de la pantalla 01.

En caso de trabajar en modo Modbus-TCP, para el caso del ejemplo 5, la trama sería la siguiente:

MBAP Header							**PDU**											
TID		Protocol ID		Length		Unit ID	Func	Dirección Registro		Num. Registros		Num. Bytes	Valor 1		Valor 2		Valor 3	
00	00	00	00	00	0D	FF	10	02	04	00	03	06	28	F9	00	00	00	00

Ejemplo 10: Trama de Modbus-TCP enviada para asignar el valor 10489 a la variable A de la pantalla.

[2] La Dirección MODBUS de la Pantalla coincide con su ID del Protocolo DTP (1 a 247)

REVISIONES

Revisión 1.0 – Documento inicial

Revisión 2.0 – Funcionalidades añadidas:

- Lectura de Input Registers
- Lectura de Input Status bits
- Envío de programa por número usando Coils.
- Explicación del uso de Modbus para el cambio de color de las variables.
- Reproducción de alarmas
- Activación de salidas digitales.
- Lectura como máster de sensores de temperatura y humedad.