MICROCOM
Simply More

MANUAL DE MICROPLC-II

Manual de programación en lenguaje MICROPLC-II

REV 25.01

MICRO PLC II

Advertencia

Este sistema ha sido desarrollado para ser instalado por profesionales, no por usuarios finales. En caso de duda ante cualquier aspecto técnico, por favor, consulte con nuestros expertos.

Nuestro esfuerzo de innovación tanto en software como en hardware es permanente. Sin embargo, a pesar de poner gran atención en documentar nuestros productos adecuadamente, podrían encontrarse, por error, discrepancias entre el producto y algunas de sus especificaciones. De esta forma, ante cualquier duda u observación, póngase en contacto con nosotros en la siguiente dirección de correo electrónico: microcom@microcom.es.

Las comunicaciones basadas en la red GSM son extraordinariamente fiables. No obstante, se desaconseja utilizar nuestro equipo en sistemas críticos si no se ha previsto algún tipo de redundancia relativa a la red de comunicaciones, ya que, excepcionalmente, puede quedar fuera de servicio.

"Apoyo vital": Esta unidad no está diseñada para su utilización en sistemas de los que dependa la vida humana. Es decir, en dispositivos cuyo mal funcionamiento ponga en riesgo la vida humana.

Nuestra responsabilidad en relación con el equipo se limitará a su reparación o restitución en los términos establecidos en la garantía.

Todos los derechos reservados. Ninguna parte de esta documentación deberá ser reproducida, almacenada en un sistema de recuperación o transmitida por cualquier medio (electrónico, mecánico, fotocopia, grabación o cualquier otro) sin el permiso previo por escrito de Microcom Sistemas Modulares, S.L.

A pesar de todas las precauciones que se han tomado en la preparación de esta documentación, el editor y el autor no asumen responsabilidad alguna por errores u omisiones. Tampoco se asume responsabilidad por los daños resultantes del uso de la información contenida en este documento. La información contenida en este documento está sujeta a cambios sin previo aviso y no representa un compromiso por parte de Microcom Sistemas Modulares, S.L.

El software descrito en este documento se suministra bajo un acuerdo de no divulgación. Este software puede ser utilizado o copiado de acuerdo con los términos de estos acuerdos.

© 2003-2024 Microcom Sistemas Modulares, S.L. Todos los derechos reservados.

Microcom Sistemas Modulares, S.L.
C/Gorostiaga, 53 • Irún
GUIPÚZCOA • 20305
Teléfono: +34 943 639 724 • Fax +34 943 017 800
microcom@microcom.es
https://www.microcom360.com

ÍNDICE

CONTENIDO DEL DOCUMENTO 5
VERSIONES Y COMPATIBILIDAD 5
CUADROS INFORMATIVOS 5
1 - DESCRIPCIÓN GENERAL 6
1.1 Acceso 7
1.2 Presentación de la interfaz 7
1.3 Panel de control MICROPLC 9
2 - ESTADOS OPERATIVOS RUN Y STOP 10
2.1 Stop 10
2.2 Run 10
2.3 Comandos asociados 10
3 - VERIFICACIÓN 11
3.1 Formato de mensaje de error 11
3.2 Descripción de mensajes de error 11
4 - MONITOR DE VARIABLES 13
5 - ESTRUCTURA DE CÓDIGO Y SINTAXIS 14
5.1 Sintaxis 14
6 - INSTRUCCIONES 15
6.1 Instrucción IF 15
6.2 Instrucción REM 15
6.3 Instrucción ACT 16
7 - OPERADORES 17
7.1 Operadores de asignación 17
7.2 Operadores aritméticos 17
7.3 Constantes matemáticas 17
7.4 Otros operadores y funciones 17
7.5 Funciones de comparación 18
7.6 Operadores de comparación 18
7.7 Operadores lógicos 18
7.8 Operadores de estado 20
8 - SECCIÓN DE INICIALIZACIÓN #INIT Y #END_INIT 21
8.1 Asignar un alias a los identificadores (opcional) 21

9 - VARIABLES 22
9.1 Declaración de variables 22
9.2 Asignar valores a las variables 23
9.3 Manejo de bits 24
9.4 Manejo de medias palabras 25
9.5 Salvar Y recuperar variables en memoria persistente 26
10 - VARIABLES DE FECHA Y HORA 27
11 - VARIABLES DE ENTRADAS Y SALIDAS 28
11.1 Listado de identificadores 28
11.2 Consultar y configurar COORDENADAS GEOGRÁFICAS 32
11.3 Consultar y configurar alarmas (ALM_CCX.PROP) 32
12 - BLOQUES TEMPORIZADORES 36
12.1 Temporizador acumulador de tiempo (RTO) 37
12.2 Temporizador de conexión (TON) 38
12.3 Temporizador de desconexión (TOF) 40
12.4 Temporizador de pulso (TP) 42
12.5 Temporizador semanal (TW) 43
12.6 Temporizadores cíclicos (CT) 44
13 - MÓDULO DE ENVÍO Y RECEPCIÓN DE SMS 46
13.1 Envío de SMS 46
13.2 Recepción de SMS 47
13.3 Declaración de teléfonos autorizados 47
14 - MÓDULO DE COMUNICACIÓN MODBUS 48
14.1 IMPORTACION IDENTIFICADOR SERVIDOR MODBUS-TCP 49
15 - MÓDULO DE CONTROL DE BOMBAS (PUMP) 51
16 - MÓDULO PID 56
16.1 Ajuste del módulo PID 58
17 - EJEMPLOS DE PROGRAMACIÓN 61
17.1 Termostato 61
17.2 Interruptor crepuscular o astronómico 63
17.3 Automatismo de arranque de bombas 65
17.4 Control de bombas en modo principal/apoyo 71
17.5 Control caudal/presión por PID 79
17.6 Configuración del mapa de memoria MODBUS en modo esclavo 83

CONTENIDO DEL DOCUMENTO

Este manual describe el lenguaje para programación de automatismos MicroPLC-II disponible para los dispositivos de la serie Hermes.

A lo largo de este documento se asume que el lector está familiarizado con la herramienta de configuración MicroConf y la configuración general de estos dispositivos.

Podrá encontrar estos manuales en la sección de descargas de la página web de Microcom. http://www.microcom360.com/start

En nuestro canal de YouTube tiene a su disposición un conjunto de video tutoriales sobre la configuración de los dispositivos y otros videos de interés.

VERSIONES Y COMPATIBILIDAD

La información mostrada en este documento corresponde con la versión del software de configuración MicroConf y firmware de los Hermes que se indican a continuación:

Elemento	Versión
Software Microconf	v.9.3.3
Firmware	v.9.38

CUADROS INFORMATIVOS

A lo largo del manual se utilizan los siguientes cuadros informativos:

	Nota: Se emplean para resaltar información de especial importancia o interés.
	Atención: Se emplean para describir las condiciones, prácticas o procedimientos que se deben seguir para realizar un correcto uso del hardware o del software.
	Ejemplo: Se emplean para mostrar casos prácticos y de esa forma, ayudar a entender mejor el texto descrito en el apartado.

1 - DESCRIPCIÓN GENERAL

MicroPLC-II es un lenguaje de programación de automatismos que destaca por su equilibrio entre facilidad de aprendizaje y potencia. Inspirado en los lenguajes clásicos de automatismos, MicroPLC-II simplifica la sintaxis y ofrece módulos lógicos nativos que facilitan la implementación de soluciones para los problemas más comunes, como la gestión de bombeos, comunicación entre estaciones y otros.

Una característica clave de MicroPLC-II es su ciclo de escaneo fijo de 1 segundo, independientemente de la complejidad y extensión del programa.

Ejemplo de SCRIPT para activación automática de una bomba según el nivel de un depósito:

```
L1   #INIT
L2   REM EJEMPLO SECCION DE CONFIGURACION
L3   AI0 : NIVEL_DEPOSITO
L4   DO0 : BOMBA
L5   #END_INIT

L6   REM EJEMPLO BUCLE DE PROGRAMA PRINCIPAL
L7   IF NIVEL_DEPOSITO >= 3.5 ; BOMBA = 0
L8   IF NIVEL_DEPOSITO <= 1.5 ; BOMBA = 1
```

Línea	Descripción
L1	Palabra clave para dar comienzo a la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Comentario para describir el programa. Esta línea no se ejecuta.
L3	Asigna el alias “NIVEL_DEPOSITO” a la entrada analógica 0 “AI0”. En la entrada analógica 0 se ha conectado y configurado un sensor que mide el nivel de agua de un depósito.
L4	Asigna el alias “BOMBA” a la salida digital 0 “DO0”. El relé de la salida digital 0 controla el encendido y apagado de un motor de bombeo de agua que rellena el depósito de agua.
L5	Palabra clave para finalizar la sección de inicialización.
L6	Comentario para describir el programa. Esta línea no se ejecuta.
L7	Si el valor en “NIVEL_DEPOSITO” es superior o igual a 3,5 metros se desactiva “BOMBA”.
L8	Si el valor en “NIVEL_DEPOSITO” es inferior a 1,5 metros se activa “BOMBA”.

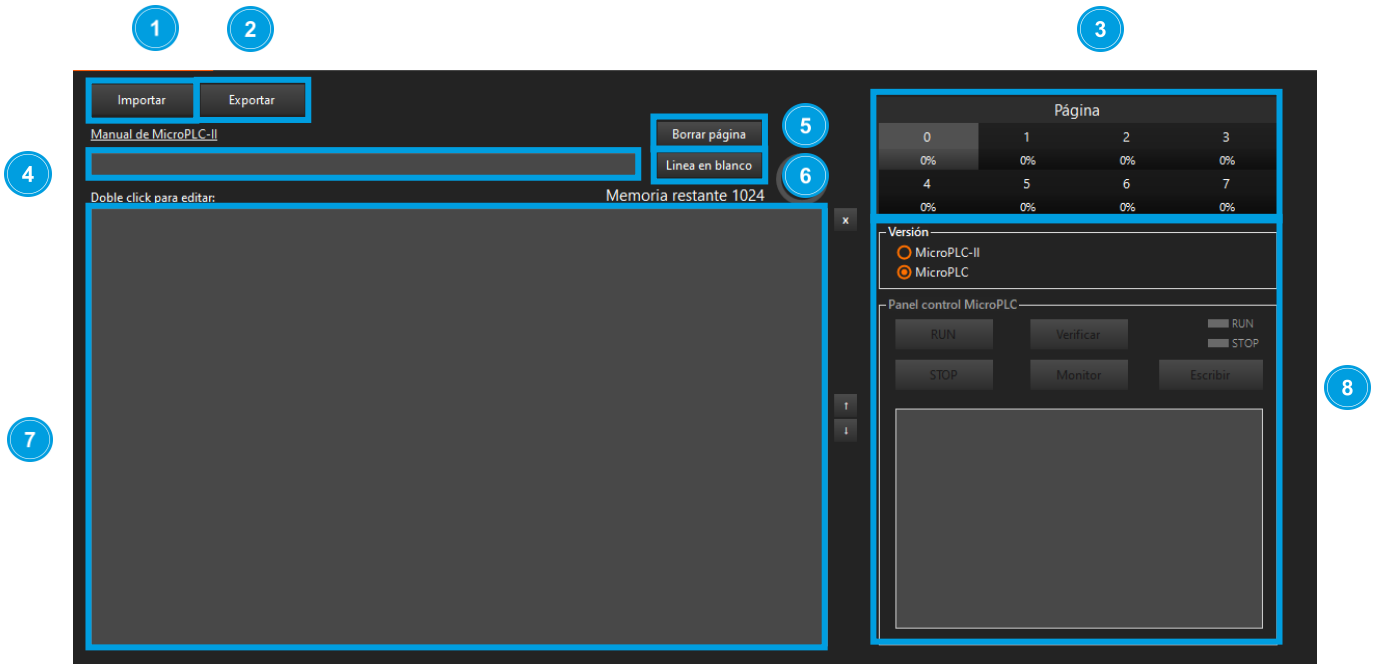
1.1 ACCESO

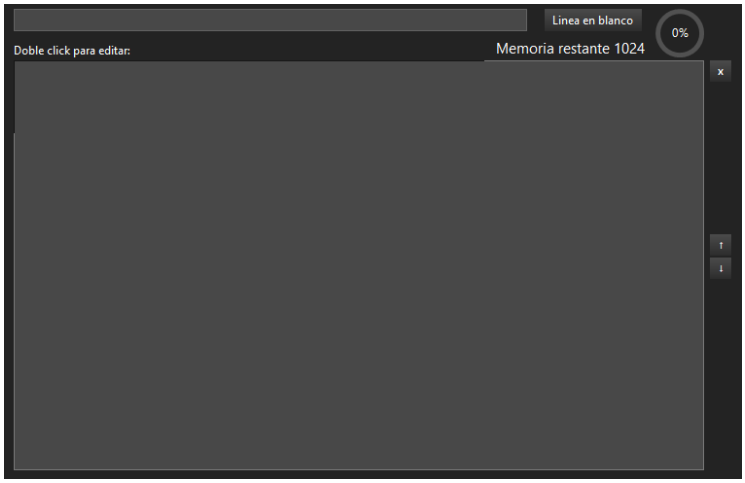
Menú → General → MicroPLC

La versión de firmware 9 (y posteriores) y la versión de MICROCONF 9 (y posteriores) integran MicroPLC-II.

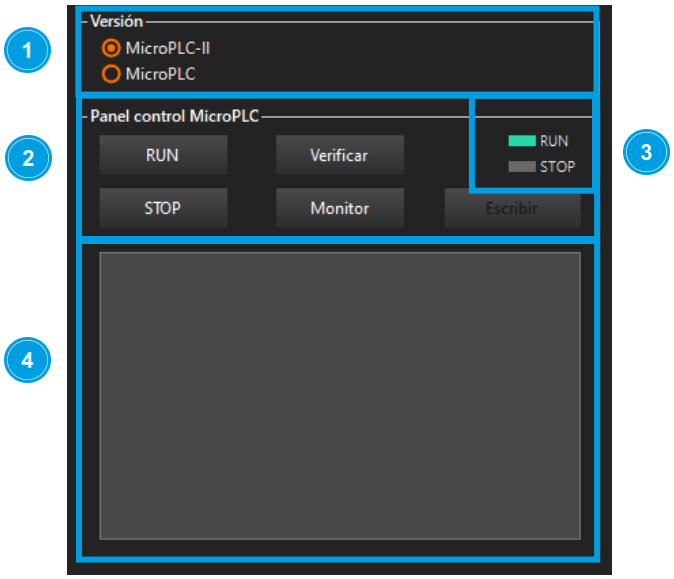
Los equipos compatibles con MicroPLC-II también son retro compatibles con el lenguaje MicroPLC.

1.2 PRESENTACIÓN DE LA INTERFAZ



Ítem	Campo	Descripción
1	Importar	Permite importar un fichero de texto plano (.TXT) con líneas de código en lenguaje MICROPLC-II. Este fichero debe incluir todas las páginas.
2	Exportar	Permite exportar todas las líneas de código MICROPLC-II a un fichero de texto plano (.TXT). Este fichero incluye la información de todas las páginas.
3	Selector de páginas	Permite navegar entre las páginas y conocer el porcentaje de utilización de cada página.
4	Caja entrada código	Línea de programa. Pulsar el botón "ENTER" para añadir el código al script de la página.
5	Borrar página	Permite borrar el código de la página seleccionada
6	Línea en blanco	Permite añadir una línea en blanco al código
7	Panel de código	Líneas de código o script.  Parte superior: Cantidad de memoria libre. Parte derecha: Botones para cambiar el orden de las líneas de código y eliminarlas.
8	Panel control MicroPLC	Ver el siguiente apartado.

1.3 PANEL DE CONTROL MICROPLC



Ítem	Campo	Descripción
1	Selector de lenguaje	Lenguaje de programación a utilizar. MicroPLC-II: Versión disponible a partir de la versión 9 de firmware. Descrita en este manual. MicroPLC: Versión disponible en versiones de firmware anteriores, que se mantiene por retro compatibilidad.
2	Botones	Botones habilitados solamente para el lenguaje MICROPLC-II: RUN: Establece el modo de ejecución. Ver Apartado 2.1. STOP: Establece el modo seguro. El código no se ejecuta. Ver Apartado 2.2. Verificar: Botón de verificación. Revisa el programa y busca errores en la sintaxis. Ver Apartado 3. Es necesarios que el sistema esté en STOP para poder verificar el Script. Monitor: Abre la ventana del monitor de variables. Ver Apartado 4. Escribir: Escribe el código de programación en el equipo.
3	Estado operativo	Indica el estado en el que se encuentra el equipo.
4	Terminal	Área de mensajes. Muestra trazas y detalles de errores. Al hacer clic en un mensaje de error, se resalta en el panel de código la línea en la que se encuentra el error. Ver Apartado 3.

2 - ESTADOS OPERATIVOS RUN Y STOP

Indica si está ejecutando el código programado en MICROPLC II.

2.1 STOP

Estado seguro. El equipo entra en este estado manualmente, pulsando el botón "STOP" o automáticamente, si detecta un fallo de código.

> El equipo no está ejecutando el programa y todas las salidas permanecen desactivadas.
>
> Este estado permite revisar la programación y la instalación sin peligro, así como cargar nuevos programas.

2.2 RUN

Estado de ejecución. El equipo entra automáticamente en este modo tras el encendido o cuando se pulsa el botón "RUN".

El equipo:

- Ejecutará las líneas que se encuentren dentro de la sección de inicialización (#INIT …#END_INIT) de todas las páginas.
- Tras ello, se ejecutará, cíclicamente, el código de todas las páginas, de izquierda a derecha y de arriba abajo (una vez por segundo), leyendo las entradas, ejecutando todas las condiciones del programa y activando las salidas necesarias en todo momento.

2.3 COMANDOS ASOCIADOS

Los siguientes comandos están disponibles para gestión de la funcionalidad MicroPLC-II:

Comando	Descripción
PLCRUN	Establece el modo RUN.
PLCSTOP	Establece el modo STOP.

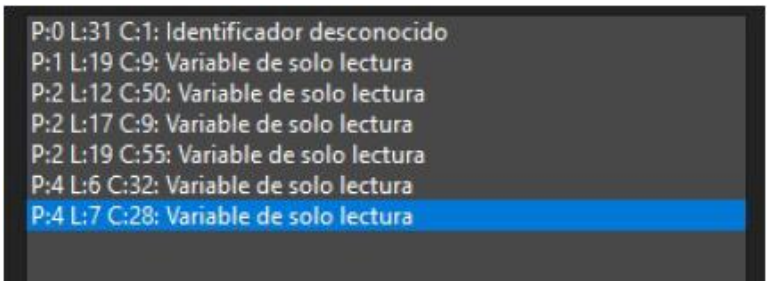
3 - VERIFICACIÓN

El software de configuración MICROCONF integra un revisor de código que devuelve los siguientes mensajes de error.

3.1 FORMATO DE MENSAJE DE ERROR

P:*00* L:*00* C:*00*: *Mensaje de error*

Donde:
- P:xx: xx indica la página de programación.
- L:xx: xx indica la línea.
- C:xx: xx indica la columna.
- Mensaje de error: Ver siguiente apartado para más información.



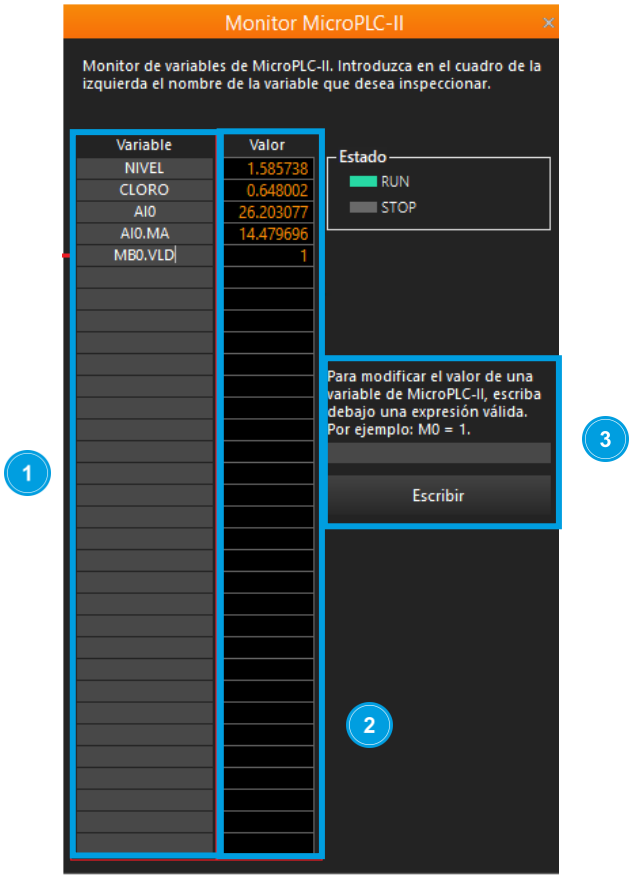
3.2 DESCRIPCIÓN DE MENSAJES DE ERROR

Mensaje de error	Descripción
Nombre de variable o alias ya existe	Se ha encontrado una declaración de nombre de variable o alias ya declarado previamente.
Índice fuera de rango	El índice está fuera de rango; por ejemplo, F32 es un índice fuera de rango, pues los flags cubren desde F0 hasta F31.
Nombre de variable o alias no válido	Se ha encontrado una declaración de nombre de variable o alias no válida. Las razones más comunes son el empleo de un nombre reservado o la presencia de caracteres no válidos.
Propiedad inválida	El objeto no tiene ninguna propiedad con el nombre indicado. Por ejemplo: DI0.Q. 'Q' no es una propiedad válida de DIx.
Sin memoria para la variable	No queda espacio en memoria para la variable declarada.
Variable de solo lectura	Se ha encontrado una asignación a una variable de solo lectura. Por ejemplo, asignar 'DI0 = 1'. 'DI0' retorna el valor de la entrada digital 0; por lo tanto, no se puede escribir.
Error sintáctico	Se encontró un error sintáctico.

Identificador desconocido	Se ha encontrado una operación con una variable o alias no declarado.
Redefinición de variable	Un nombre de variable o alias se ha declarado más de una vez. Ejemplo: BOOL : BOYA_BAJO INT : BOYA_BAJO
Parámetro no encontrado	Este error se produce cuando se intenta hacer referencia a una variable o parámetro que no existe en el dispositivo.
Propiedad no permitida en definición de alias	Se intentó crear un alias incluyendo un modificador de propiedad (por ejemplo, AI0.MA). La sintaxis válida para declarar un alias es únicamente el nombre de la señal, sin incluir ninguna propiedad.

4 - MONITOR DE VARIABLES

El monitor de variables permite la inspección y modificación de variables de MicroPLC-II en tiempo real para propósitos de depuración del código.



Ítem	Campo	Descripción
1	Variable	Nombre de las variables a inspeccionar. Tanto las variables declaradas dentro de MICROPLC-II como las variables internas del sistema son válidas.
2	Valor	Valor de la variable. El carácter ? indica que la variable no se reconoce.
3	Escribir	Modificación de variables en tiempo real.

5 - ESTRUCTURA DE CÓDIGO Y SINTAXIS

Cada línea de código puede contener una o más instrucciones, separadas por un punto y coma (";"). La ejecución de las instrucciones es secuencial: el código se ejecuta de principio a fin en cada línea, a menos que se encuentre con una condición "IF" no satisfecha. En tal caso, la ejecución se interrumpe y se continúa en la siguiente línea.

El código se ejecuta una vez por segundo. Primero se ejecuta la página 0, luego la página 1, y así sucesivamente hasta llegar a la última página.

5.1 SINTAXIS

Un código correcto en MICROPLC-II respeta las siguientes reglas:

- Un comando termina al llegar al final de la línea o con un punto y coma ";"
- El separador decimal es un punto.
  Ejemplo:

```
NIVEL = 3.25
```

- Los nombres de las variables pueden contener letras latinas, números y guiones bajos.
  Ejemplo:

```
AI0 : NIVEL_DEPOSITO_2
```

- Los nombres de las variables no pueden comenzar con un dígito.
- Se permite dejar espacios en blanco entre instrucciones y operadores, y saltos de línea entre líneas de código.

```
L1    #INIT
L2    REM EJEMPLO SECCION DE CONFIGURACION
L3    AI0 : NIVEL_DEPOSITO
L4    DO0 : BOMBA
L5    #END_INIT

L6    REM EJEMPLO BUCLE DE PROGRAMA PRINCIPAL
L7    IF NIVEL_DEPOSITO >= 3.5 ; BOMBA = 0
L8    IF NIVEL_DEPOSITO <= 1.5 ; BOMBA = 1
```

6 - INSTRUCCIONES

6.1 INSTRUCCIÓN IF

En caso de cumplir la condición indicada, se ejecutan las instrucciones posteriores. De lo contrario, pasa a la siguiente línea.

SINTAXIS

```
IF condición; instrucciones...
```

Donde:
- Condición: Expresión cuyo resultado se evaluará para condicionar la ejecución. Cualquier resultado diferente de 0 se considera VERDADERO.
- Instrucciones: Una o varias instrucciones que siguen a IF y se ejecutan si condición se evalúa como VERDADERO.

Ejemplo:

```
IF NIVEL < 3.25 ; FALTA_AGUA = 1
```

6.2 INSTRUCCIÓN REM

Se utiliza para incluir un comentario en el código.

SINTAXIS

```
REM comentario
```

Donde:
- Comentario: Texto libre que no se evaluará.

Ejemplo:

```
REM DECLARAR VARIABLES
```

6.3 INSTRUCCIÓN ACT

Instrucción para ejecución de acciones desde MicroPLC-II.

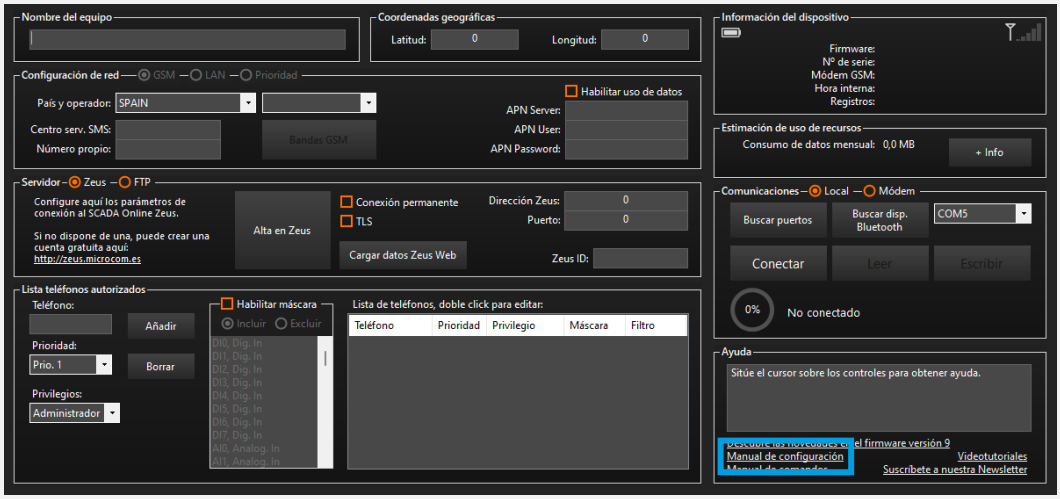
SINTAXIS

```
ACT acciones
```

Donde:

- Acciones: Lista de hasta 8 acciones separadas por comas.

Consultar las acciones disponibles en el apéndice A del manual del software de configuración que está disponible en www.microcom360.com.O en el menú general del software de configuración.



Ejemplo:

```
ACT 79
ACT 79,53,75,65,57,111,98,99
```

7 - OPERADORES

7.1 OPERADORES DE ASIGNACIÓN

	Descripción	Ejemplo
:	Se utiliza en la declaración de variables, de alias y otros objetos.	INT : TEMPERATURA_MAX
=	Asigna el valor del término de la derecha al operador de la izquierda.	INT : TEMPERATURA_MAX = 27

7.2 OPERADORES ARITMÉTICOS

	Descripción		Descripción
+	Suma	^	Potenciación
-	Resta	\	Módulo. Entero de una división
*	Multiplicación	SQRT()	Raíz cuadrada
/	División		

7.3 CONSTANTES MATEMÁTICAS

	Descripción		Descripción
TRUE	Equivale a un valor lógico 1	PI	Número PI
FALSE	Equivale a un valor lógico 0	E	El número "e", número de Euler

7.4 OTROS OPERADORES Y FUNCIONES

	Descripción		Descripción
SIN( )	Seno	ASIN( )	Arco seno
COS( )	Coseno	ACOS( )	Arco coseno
TAN( )	Tangente	ATAN( )	Arco tangente
EXP( )	Exponenciación	LN( )	Logaritmo neperiano
ABS( )	Valor absoluto	LOG( )	Logaritmo decimal
INT( )	Devuelve la parte entera de un valor	FRAC( )	Devuelve la parte decimal de un valor
RAND	Generador de número aleatorios. Rango: 0…1		

7.5 FUNCIONES DE COMPARACIÓN

	Descripción
MIN(n1,n2)	Compara dos valores (n1 y n2) y devuelve el de menor valor.
MAX(n1,n2)	Compara dos valores (n1 y n2) y devuelve el de mayor valor.

7.6 OPERADORES DE COMPARACIÓN

Descripción		Ejemplo (*)	Resultado ejemplo
>	Mayor que	FLAG = NIVEL > 3	Si "NIVEL" es mayor que 3, "FLAG" obtendrá el valor 1. Si "NIVEL" es menor que 3 obtendrá el valor 0.
<	Menor que	FLAG = NIVEL < 4	Si "NIVEL" es menor que 4, "FLAG" obtendrá el valor 1. Si "NIVEL" es mayor que 4 obtendrá el valor 0.
=	Igual que	FLAG = NIVEL = 2	Si "NIVEL" es igual a 2, "FLAG" obtendrá el valor 1. Si "NIVEL" es distinto a 2 obtendrá el valor 0.
>=	Mayor o igual que	FLAG = NIVEL >= 3	Si "NIVEL" es mayor o igual que 3, "FLAG" obtendrá el valor 1. Si "NIVEL" es menor que 3 obtendrá el valor 0.
<=	Menor o igual que	FLAG = NIVEL <= 4	Si "NIVEL" es menor o igual que 4, "FLAG" obtendrá el valor 1. Si "NIVEL" es mayor que 4 obtendrá el valor 0.
<>	Distinto de	FLAG = NIVEL <> 5	Si "NIVEL" es distinto a 5, "FLAG" obtendrá el valor 1. Si "NIVEL" es igual a 5 obtendrá el valor 0.

(*) En los ejemplos se considera que la variable FLAG es de tipo BOOL, y la variable NIVEL es de tipo REAL.

7.7 OPERADORES LÓGICOS

	Descripción	Ejemplo (*)	Resultado ejemplo
NOT	Invierte el resultado de la expresión que sigue inmediatamente.	DO0 = NOT(NIVEL>3.25)	Si "NIVEL" es mayor que 3.25, DO0 obtendrá el valor 0. Si "NIVEL" es menor o igual que 3.25, DO0 obtendrá el valor 1.
AND	AND lógico	FLAG = REBOSE AND BOMBA_ON	FLAG solo será verdadero cuando REBOSE sea verdadero y BOMBA_ON sea verdadero.
OR	OR lógico	FLAG = REBOSE OR BOMBA_ON	FLAG será verdadero si REBOSE o BOMBA_ON son verdaderos.

	Descripción	Ejemplo (*)	Resultado ejemplo
&	AND binario	A = B & C	Si B=0101 y C=1100, A = 0100 0101 & 1100 -------- 0100
|	OR inclusivo binario	A = B | C	Si B=0101 y C=1100, A = 0100 0101 | 1100 --------- 1101
~	Complemento a uno. Cada bit que es 1 en el operando es 0 en el resultado y viceversa.	A = ~B	Si B contiene el valor 39 (00100111 en binario), A contendrá el valor 216 (11011000 en binario).
>>n	Desplazamiento de bit 'n' posiciones a la derecha.	TEMP = NIVEL >> 2	Si NIVEL contiene el valor 8 (1000 en binario), TEMP contendrá el valor 2 (0010 en binario).
<<n	Desplazamiento de bit 'n' posiciones a la izquierda.	TEMP = NIVEL << 1	Si NIVEL contiene el valor 2 (0010 en binario), TEMP contendrá el valor 4 (0100 en binario).

(*) En los ejemplos se considera que la variable "FLAG", "REBOSE" y "BOMBA_ON" son de tipo BOOL. La variable "NIVEL" y "TEMP" son de tipo REAL.

7.8 OPERADORES DE ESTADO

	Descripción
PWR.VIN	Devuelve la tensión de alimentación externa.
PWR.VBAT	Devuelve el valor de la tensión de la batería interna de mayor voltaje. Compatible con todos los modelos de Nemos.
PWR.VBAT1	Devuelve el valor de la tensión de la batería interna conectada al puerto 1. Compatible con los Nemos N200 y N200+.
PWR.VBAT2	Devuelve el valor de la tensión de la batería interna conectada al puerto 2. Compatible con los Nemos N200 y N200+.
PWR.FAIL	Devuelve 1 si el suministro eléctrico ha fallado y el equipo se está alimentado de su batería interna. Compatible con los Hermes LC2+ y TCR210.
PWR.BL	(Battery Life) Devuelve un valor entre 0 y 2 (incluidos), que representa el estado de la batería interna en un código de colores, siendo: 0=Verde (OK), 1=Naranja (inferior a 3 meses), 2=Rojo (Agotada). En caso de disponer de dos packs de baterías, muestra el de la que tiene mejor estado. Compatible con todos los modelos de Nemos.
GSM	RSSI o intensidad de campo señal GSM. El equipo devuelve un valor numérico entre 1 y 32. El mínimo recomendable es de 8. Ecuación de conversión a dBm: dBm = -113 + N * 2 (donde N es el valor devuelto).
STAT.MBCE	Devuelve un 1 si error de comunicación MODBUS.
STAT.EXCE	Devuelve un 1 si error de comunicación con módulos de expansión.
STAT.PBCE	Devuelve un 1 si error de comunicación con sondas digitales Microcom STDV01/STDV02.
TEMP	Temperatura interna del dispositivo. Compatible con todos los modelos de Nemos.

Nota: Si bien MicroPLC-II solo está disponible en dispositivos tipo Hermes, se han incluido los operadores de estado correspondientes a dispositivos Nemos por coherencia con el manual de configuración.

8 - SECCIÓN DE INICIALIZACIÓN #INIT Y #END_INIT

#INIT y #END_INIT son palabras clave utilizadas para iniciar y cerrar la sección de inicialización en el código. Puede haber una sección de inicialización por página. Esta sección se utiliza para declarar variables, bloques temporizadores, asignación de alias, entre otros elementos. Ambas palabras clave se mostrarán en color naranja. Incluir la sección de inicialización en el código es opcional.

La sección de inicialización se ejecuta solo una vez en las siguientes circunstancias: al aplicar alimentación al dispositivo, al pasar del estado STOP al estado RUN (ver Apartado 2.) o al cargar una nueva configuración o script.

El orden de ejecución es el siguiente: primero se ejecuta la sección de inicialización de la página 0, luego la de la página 1, y así sucesivamente hasta la última página. Tras esto, comienza la ejecución del código regular a una frecuencia de una vez por segundo.

8.1 ASIGNAR UN ALIAS A LOS IDENTIFICADORES (OPCIONAL)

A las interfaces exteriores, registros matemáticos y flags se les puede asignar un "alias" mediante el símbolo ":".

Los alias no soportan acentos y otros caracteres especiales, como la letra "Ñ".

Ejemplo de asignación de alias en la sección de inicialización.

```
L1  #INIT
L2  INT : TEMPERATURA
L3  DI0 : BOMBA_ON
L4  TON : TEMPO, PT = 60
L5  #END_INIT
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declara "TEMPERATURA" como una variable de tipo entero con signo de 32 bits.
L3	Asigna el alias BOMBA_ON a la entrada digital 0 "DI0".
L4	Declara TEMPO como un temporizador de conexión. Configura el temporizador para que expire a los60 segundos.
L5	Palabra clave para finalizar la sección de inicialización.

9 - VARIABLES

MicroPLC-II permite definir hasta 64 variables de tipo BOOL y hasta 64 variables adicionales, combinando tipos REAL o INT. Las variables son volátiles, lo que significa que su valor se pierde si se interrumpe el flujo eléctrico.

Tipo	Descripción	Rango
BOOL	Variable de tipo booleana.	0 … 1
INT	Variable tipo entero con signo de 32 bits.	-2.147.483.648 … +2.147.483.647
REAL	Variable tipo coma flotante de precisión simple.	$1{,}2x10^{-38}$… $3{,}4x10^{38}$

9.1 DECLARACIÓN DE VARIABLES

Para declarar las variables internas se utiliza el símbolo ":". Opcionalmente, se puede asignar un valor inicial a la variable en la misma línea de código utilizando el símbolo "=". Si no se especifica un valor de inicialización, la variable se inicializa a 0. Las variables se deben declarar dentro de la sección de inicialización.

Los nombres de la variables no soportan acentos y otros caracteres especiales, como la letra "Ñ".

Ejemplo de declaración de variables.

```
L1 #INIT
L2 BOOL : BOMBAON
L3 BOOL : FALLO = TRUE
L4 INT : CONTADOR
L5 INT : AUTO = 2
L6 REAL : NIVELMAX
L7 REAL : NIVEL = 5.5
L8 #END_INIT
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declara "BOMBAON" como una variable de tipo booleana. Al no indicar nada más, la variable se inicializa a 0.
L3	Declara "FALLO" como una variable de tipo booleana y la inicializa a 1 (TRUE).

Línea	Descripción
L4	Declara "CONTADOR" como una variable de tipo entero con signo de 32 bits. Al no indicar nada más, la variable se inicializa a 0.
L5	Declara "AUTO" como una variable de tipo entero con signo de 32 bits y la inicializa a 2.
L6	Declara "NIVELMAX" como una variable de tipo coma flotante de precisión simple. Al no indicar nada más, la variable se inicializa a 0.
L7	Declara "NIVEL" como una variable de tipo coma flotante de precisión simple y la inicializa a 5,5.
L8	Palabra clave para finalizar la sección de inicialización.

9.2 ASIGNAR VALORES A LAS VARIABLES

Para modificar las variables declaradas se utiliza el símbolo "=".

Ejemplo de asignación de un valor a una variable interna.

```
L1  #INIT
L2  INT : TEMPERATURA_MAX
L3  INT : CUENTAHORAS
L4  #END_INIT

L5  TEMPERATURA_MAX = 27
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declara "TEMPERATURA_MAX" como una variable de tipo entero con signo de 32 bits.
L3	Declara "CUENTAHORAS" como una variable de tipo entero con signo de 32 bits.
L4	Palabra clave para finalizar la sección de inicialización.
L5	Iguala a 27 el valor de la variable "TEMPERATURA_MAX".

9.3 MANEJO DE BITS

Las variables de tipo INT se pueden manipular a nivel de bit mediante la propiedad ".Bx", donde "x" representa el número del bit a modificar. En este contexto, el bit 0 es el menos significativo y el bit 31 es el más significativo.

La propiedad ".Bx" permite tanto la lectura como la escritura.

Crear una variable que contenga el estado de canales de entradas digitales, MODBUS y expansiones.

```
L1  #INIT
L2  INT : ESTADO_ENTRADAS
L3  #END_INIT
L4  ESTADO_ENTRADAS.B0 = DI0
L5  ESTADO_ENTRADAS.B1 = DI1
L6  ESTADO_ENTRADAS.B2 = DI2
L7  ESTADO_ENTRADAS.B3 = DI3
L8  ESTADO_ENTRADAS.B4 = MB0
L9  ESTADO_ENTRADAS.B5 = MB1
L10 ESTADO_ENTRADAS.B6 = EXP0
L11 ESTADO_ENTRADAS.B7 = EXP1
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declarar "ESTADO_ENTRADAS" como una variable de tipo entero con signo de 32 bits.
L3	Palabra clave para finalizar la sección de inicialización.
L4	Iguala el bit número 0 de "ESTADO_ENTRADAS" con el valor de la entrada digital 0.
L5	Iguala el bit número 1 de "ESTADO_ENTRADAS" con el valor de la entrada digital 1.
L6	Iguala el bit número 2 de "ESTADO_ENTRADAS" con el valor de la entrada digital 2.
L7	Iguala el bit número 3 de "ESTADO_ENTRADAS" con el valor de la entrada digital 3.
L8	Iguala el bit número 4 de "ESTADO_ENTRADAS" con el valor del canal MODBUS 0.
L9	Iguala el bit número 5 de "ESTADO_ENTRADAS" con el valor del canal MODBUS 1.
L10	Iguala el bit número 6 de "ESTADO_ENTRADAS" con el valor de canal de la expansión 0.
L11	Iguala el bit número 7 de "ESTADO_ENTRADAS" con el valor de canal de la expansión 1.

9.4 MANEJO DE MEDIAS PALABRAS

Las variables de tipo INT y REAL se pueden manipular a nivel de media palabra mediante las propiedades ".H" y ".L". La propiedad ".H" direcciona los 16 bits más significativos, mientras que la propiedad ".L" selecciona los 16 bits menos significativos.

Las propiedades ".H" y ".L" permiten tanto la lectura como la escritura.

Esta funcionalidad resulta particularmente útil para la definición del mapa MODBUS en modo esclavo.

Las variables de tipo INT, a través de registros de usuario se leen como un entero de 32 bits, mientras que las variables de tipo REAL se leen como numero de coma flotante de precisión simple (FLOAT).

Mapear el valor de la entrada analógica 1 en la dirección 30001 como numero de coma flotante de precisión simple en modo "Big endian".

```
L1  #INIT
L2  REAL : NIVEL
L3  #END_INIT
L4  NIVEL = AI0
L5  MBIR0 = NIVEL.L
L6  MBIR1 = NIVEL.H
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declarar "NIVEL" como una variable de tipo entero con signo de 32 bits.
L3	Palabra clave para finalizar la sección de inicialización.
L4	Guarda en "NIVEL" el valor de la entrada analógica 0 "AI0"
L5	Guarda los 16 bit de menor peso de la variable "NIVEL" en el registro de usuario 0 para que sean accesibles desde la dirección 30001
L6	Guarda los 16 bit de mayor peso de la variable "NIVEL" en el registro de usuario 1 para que sean accesibles desde la dirección 30002

Para ver otro ejemplo de implementación Ver apartado 16.4.

9.5 SALVAR Y RECUPERAR VARIABLES EN MEMORIA PERSISTENTE

Las variables de tipo BOOL, INT y REAL son volátiles y su valor se pierde si se interrumpe el flujo eléctrico. Para hacer persistente el valor de estas variables, se puede salvar y recuperar su valor en FLAGS o REGISTROS MATEMÁTICOS, los cuales son registros de memoria no volátiles.

Configuración de un contador de arranques guardando el valor del contador en una memoria no volátil.

Se incrementa el valor del contador cada vez que hay un nuevo arranque y se guarda el valor en una memoria no volátil para no perder la información en caso de fallo de suministro eléctrico.

```
L1  #INIT
L2  INT : CONTADOR
L3  INT : PRE_BOMBA
L4  CONTADOR = M0
L5  EXP16 : BOMBA
L6  #END_INIT
L7  IF BOMBA = TRUE AND BOMBA <> PRE_BOMBA ; CONTADOR = CONTADOR + 1
L8  PRE_BOMBA = BOMBA
L9  M0 = CONTADOR
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declarar “CONTADOR” como una variable de tipo entero con signo de 32 bits.
L3	Declarar “PRE_BOMBA” como una variable de tipo entero con signo de 32 bits.
L4	Inicializa la variable “CONTADOR” con el valor de “M0”. La primera vez que se ejecute el programa, “M0” será igual a 0, pero si el programa se reanuda tras un apagado, esté tendrá el valor asignado en L9.
L5	Asigna el alias “BOMBA” al canal de expansión 16.
L6	Palabra clave para finalizar la sección de inicialización.
L7	Si “BOMBA” está activado y su valor es distinto del valor de “PRE_BOMBA”, significa que hay un nuevo arranque, y sumamos 1 a la variable “CONTADOR”.
L8	Guarda en “PRE_BOMBA” el valor de “BOMBA” para la siguiente comprobación.
L9	Guarda en el registro matemático 0 “M0” el valor de “CONTADOR”. Guardamos el valor en una memoria no volátil para no perderlo en caso de fallo de alimentación. En caso de fallo de red el valor se recupera en L4.

Para ver otro ejemplo de implementación, Ver el apartado 16.3.

10 - VARIABLES DE FECHA Y HORA

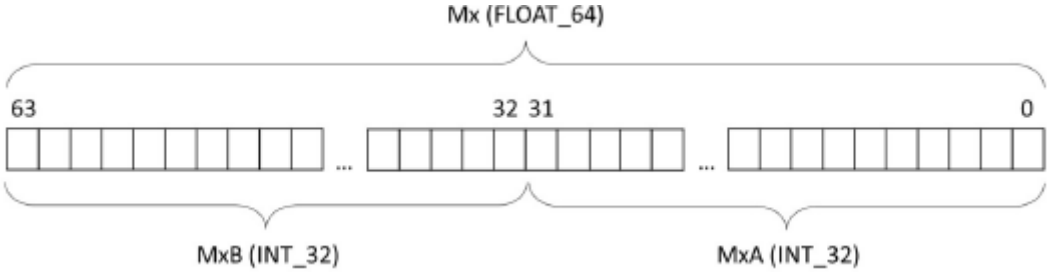
Variable	Descripción
NOW.Y	Devuelve un número de cuatro dígitos que representa el año en curso.
NOW.MO	Devuelve un número comprendido entre 1 y 12 (incluidos), que representa el mes del año en curso.
NOW.D	Devuelve un número comprendido entre 1 y 31 (incluidos), que representa el día del mes en curso.
NOW.WD	Devuelve un número comprendido entre 1 y 7 (incluidos), que representa el día de la semana: Lunes=1, Martes=2, Miércoles=3, Jueves=4, Viernes=5, Sábado=6, Domingo=7
NOW.YD	Devuelve un número comprendido entre 1 y 366 (incluidos), que representa el día del año en curso.
NOW.HHMM	Devuelve la hora del día en curso en formato 24H, donde: HH: devuelve un número comprendido entre 0 y 23 (incluidos) MM: devuelve un número comprendido entre 0 y 59 (incluidos)
NOW.SOD	(Seconds Of Day). Devuelve un número comprendido entre 0 y 86399 (incluidos), que representan los segundos transcurridos del día en curso.
SUMMER	Devuelve un número comprendido entre 0 y 1 (incluidos) que indica el cambio horario: 0 para invierno, 1 para verano.
SUNRISE	Devuelve un número comprendido entre 0 y 86399 (incluidos), que representa la hora de la salida de sol del día en curso en segundos. Para utilizar este operador matemático, las coordenadas geográficas deben estar informadas.
SUNSET	Devuelve un número comprendido entre 0 y 86399 (incluidos), que representa la hora de la puesta de sol del día en curso en segundos. Para utilizar este operador matemático, las coordenadas geográficas deben estar informadas.

11 - VARIABLES DE ENTRADAS Y SALIDAS

Cada canal, así como los registros matemáticos y los flags, tiene su identificador asociado.

11.1 LISTADO DE IDENTIFICADORES

En todos los identificadores la "x" se debe sustituir por el número de la entrada, de la salida o del canal.

Identificador	Descripción
Mx	Valor del registro matemático x. Un registro matemático es una variable de 64 bits de tipo coma flotante y precisión doble. Son variables no volátiles y sus valores no se pierden si se interrumpe el flujo eléctrico. Los equipos compatibles con MICROPLC-II disponen de 32 registros matemáticos.
MxA MxB	Cada registro matemático de 64 bits y tipo coma flotante de precisión doble se puede utilizar como dos registros matemáticos de 32 bits y tipo entero con signo. Siendo MxA, los 32 bits de menor peso y MxB los 32 bits de mayor peso. De esta forma se duplica el número de registros matemáticos disponibles.  Ejemplo: M0A = M0A + 1 M0B = M0B + 1
Fx	Estado del flag x.

Identificador	Descripción
FLAG	Palabra que contiene el estado de los 32 FLAGs del equipo. En representación binaria el bit menos significativo corresponde con FLAG 0 y el bit más significativo corresponde con el FLAG 31.

Flag activo	Decimal	Binario
Ninguno	0	0000 0000 0000 0000 0000 0000 0000 0000
0	1	0000 0000 0000 0000 0000 0000 0000 0001
1	2	0000 0000 0000 0000 0000 0000 0000 0010
2	4	0000 0000 0000 0000 0000 0000 0000 0100
…	…	…
29	536870912	0010 0000 0000 0000 0000 0000 0000 0000
30	1073741824	0100 0000 0000 0000 0000 0000 0000 0000
31	2147483648	1000 0000 0000 0000 0000 0000 0000 0000

Ejemplo:

Si están activados el flag 1 (activado el segundo bit) y el flag 4 (activado el quinto bit), la palabra FLAG devuelve un 18: FLAG=18 (10010 binario).

DIx	Estado de la entrada digital x. Estado eléctrico que no considera la configuración NA/NC de dicha entrada.
DIx.AC	Tiempo acumulado en segundos que lleva la entrada digital x activada (cuenta horas).
DIx.T	Tiempo de activación en segundos desde la última activación que lleva la entrada digital x.
DIx.H	Verdadero si la entrada digital x está en estado activo según la configuración NA/NC de dicha entrada.
DIx.L	Verdadero si la entrada digital x está en estado inactivo según la configuración NA/NC de dicha entrada.
CNTx	Valor crudo sin conversión del contador x.
CNTx.H	16 bits de más peso del valor crudo sin conversión del contador x.
CNTx.L	16 bits de menos peso del valor crudo sin conversión del contador x.
CNTx.M3	Valor convertido a metros cúbicos del contador x.

<table>
<tr><th>Identificador</th><th>Descripción</th></tr>
<tr><td>DAYFLOW(x)</td><td>Valor en metros cúbicos del volumen acumulado en 24h del caudalímetro x. Hora de inicio 00:00 y hora final 23:59.</td></tr>
<tr><td>FRx</td><td>Valor del caudal x</td></tr>
<tr><td>FRx.IM</td><td>Valor instantáneo del caudal x.</td></tr>
<tr><td>DIP</td><td>Estado del puerto de entradas digitales. Permite leer el estado de varias entradas digitales a la vez sumando sus valores decimales. En representación binaria, el bit menos significativo es la entrada digital 0 y el bit más significativo es la entada digital 7.
<table>
<tr><th>Entrada digital activa</th><th>Decimal</th><th>Binario</th></tr>
<tr><td>Ninguna</td><td>0</td><td>0000 0000</td></tr>
<tr><td>0</td><td>1</td><td>0000 0001</td></tr>
<tr><td>1</td><td>2</td><td>0000 0010</td></tr>
<tr><td>2</td><td>4</td><td>0000 0100</td></tr>
<tr><td>3</td><td>8</td><td>0000 1000</td></tr>
<tr><td>4</td><td>16</td><td>0001 0000</td></tr>
<tr><td>5</td><td>32</td><td>0010 0000</td></tr>
<tr><td>6</td><td>64</td><td>0100 0000</td></tr>
<tr><td>7</td><td>128</td><td>1000 0000</td></tr>
</table>
Ejemplo: Si las entradas digitales 1 (2 decimal) y 4 (16 decimal) están activadas, el identificador DIP devuelve un 18: DIP=18 (0001 0010 en binario).</td></tr>
<tr><td>AIx</td><td>Valor en unidad de ingeniería de la entrada analógica x.</td></tr>
<tr><td>AIx.MA</td><td>Valor en mA de la entrada analógica x.</td></tr>
<tr><td>AIx.VLD</td><td>Flag de validez de la entrada analógica x. Devolverá 1 si la corriente en el bucle es superior a 3,8mA.</td></tr>
<tr><td>DOx</td><td>Estado de la salida digital x.</td></tr>
</table>

<table>
<tr><th>Identificador</th><th>Descripción</th></tr>
<tr><td>DOP</td><td>Estado del puerto de las salidas digitales. Devuelve el estado de varias salidas digitales a la vez sumando sus valores decimales. En binario, el bit menos significativo es la salida digital 0 y el bit más significativo es la salida digital 7.
<table>
<tr><th>Salida digital activa</th><th>Decimal</th><th>Binario</th></tr>
<tr><td>Ninguna</td><td>0</td><td>0000 0000</td></tr>
<tr><td>0</td><td>1</td><td>0000 0001</td></tr>
<tr><td>1</td><td>2</td><td>0000 0010</td></tr>
<tr><td>2</td><td>4</td><td>0000 0100</td></tr>
<tr><td>3</td><td>8</td><td>0000 1000</td></tr>
<tr><td>4</td><td>16</td><td>0001 0000</td></tr>
<tr><td>5</td><td>32</td><td>0010 0000</td></tr>
<tr><td>6</td><td>64</td><td>0100 0000</td></tr>
<tr><td>7</td><td>128</td><td>1000 0000</td></tr>
</table>
Ejemplo: Si las salidas digitales 2 (4 decimal) y 3 (8 decimal) están activadas, el operador DOP devuelve un 12: DOP=12 (0000 1100 en binario).</td></tr>
<tr><td>AOx</td><td>Valor de la salida analógica x.</td></tr>
<tr><td>AOx.MA</td><td>Valor en miliamperios de la corriente en el bucle de la salida analógica x.</td></tr>
<tr><td>MBx</td><td>Valor del canal MODBUS x.</td></tr>
<tr><td>MBx.VLD</td><td>Flag de validez del canal MODBUS x. Devolverá 1 si la lectura se realiza con éxito.</td></tr>
<tr><td>MBIRx</td><td>Valor del MODBUS INPUT REG x. Ver apartado 16.4</td></tr>
<tr><td>EXPx</td><td>Valor de la expansión x.</td></tr>
<tr><td>EXPx.T</td><td>Tiempo en segundos transcurrido desde la última activación del canal de expansión x en modo entrada digital.</td></tr>
<tr><td>EXPx.H</td><td>Verdadero si la expansión x en modo entrada digital está en estado activo según la configuración NA/NC de dicha entrada. Es equivalente a EXPx y se proporciona para mantener la coherencia con las entradas digitales propias del módulo maestro (DIx.H).</td></tr>
<tr><td>EXPx.L</td><td>Verdadero si la expansión x en modo entrada digital está en estado inactivo según la configuración NA/NC de dicha entrada.</td></tr>
<tr><td>EXPx.VLD</td><td>Flag de validez de la expansión x. Devolverá 1 si la lectura se realiza con éxito.
En expansiones analógicas en modo corriente adicionalmente el valor en el bucle debe ser superior a 3,8mA para considerar la señal válida.</td></tr>
<tr><td>PBx</td><td>Valor de la sonda 1-wire x.</td></tr>
</table>

Identificador	Descripción
PBx.VLD	Flag de validez de la sonda 1-wire x. Devolverá 1 si la lectura se realiza con éxito.
ALM_CCx.PROP	Consultar y configurar alarmas. Donde CCx es la abreviatura del canal correspondiente y PROP es la propiedad que se quiere modificar.

11.2 CONSULTAR Y CONFIGURAR COORDENADAS GEOGRÁFICAS

Identificador	Descripción
POS.LAT	Latitud en grados decimales.
POS.LON	Longitud en grados decimales.

11.3 CONSULTAR Y CONFIGURAR ALARMAS (ALM_CCX.PROP)

Identificador	Descripción
ALM_DIx	Alarma de la entrada digital x.
ALM_AIx	Alarma de entrada analógica x.
ALM_FRx	Alarma del caudalímetro x.
ALM_MBx	Alarma del canal MODBUS x.
ALM_PBx	Alarma de la sonda MICROCOM (STDV01 y STDV02) x.
ALM_EXPx	Alarma del canal de expansión x.
ALM_MCx	Alarma de canal matemático x.
ALM_Fx	Alarma del flag x.

Consultar y modificar alarmas de canales con valores digitales.

Identificador	Propiedad	Descripción	Tipo
ALM_DIx ALM_MBx ALM_EXPx ALM_Fx	PH	Lectura y escritura. Persistencia. Tiempo en segundos que tiene que estar la señal digital en el estado activo para que se dispare la alarma. Persistencia máxima: 65535 segundos.	INT
	PL	Lectura y escritura. Rearme. Tiempo en segundos para que se rearme (vuelva a estar activa) la alarma tras el disparo. Tiempo máximo rearme: 65535 segundos.	INT

Ejemplo de modificación de alarma de un canal con valores digitales.

```
L1   ALM_DI2.PH = 10
L2   ALM_DI2.PL = 15
```

Línea	Descripción
L1	Configura la persistencia de la alarma de la entrada digital 2 configuración 0 a 10 segundos.
L2	Configura el rearme de la alarma de la entrada digital 2 configuración 0 a 15 segundos.

Consultar y modificar alarmas de canales con valores analógicos.

Identificador	Propiedad	Descripción	Tipo
ALM_AIx ALM_FRx ALM_MBx ALM_PBx ALM_EXPx ALM_MCx	MAX	Lectura y escritura. Umbral máximo. Valor por encima del cual se dispara la alarma.	REAL
	MIN	Lectura y escritura. Umbral mínimo. Valor por debajo del cual se dispara la alarma.	REAL
	HYS	Lectura y escritura. Histéresis o “Banda muerta”. Diferencia entre el umbral de alarma y de reposición.	INT
	PER	Lectura y escritura. Persistencia. Tiempo que debe estar la señal fuera de rango para que se dispare la alarma. Persistencia máxima: 65535 ciclos de lectura (Se expresa en segundos, salvo que se indique lo contrario).	INT

Ejemplo de modificación de alarma de un canal con valores analógico.

```
L1  ALM_AI3.MAX = 3.75
L2  ALM_AI3.MIN = 1.25
L3  ALM_AI3.HYS = 0.5
L4  ALM_AI3.PER = 60
L5  M0 = ALM_AI3.MAX
L6  M1 = ALM_AI3.MIN
```

Línea	Descripción
L1	Configura el umbral máximo en la alarma de la entrada analógica 3 configuración 0 a 3.75.
L2	Configura el umbral mínimo en la alarma de la entrada analógica 3 configuración 0 a 1.25.
L3	Configura la histéresis en la alarma de la entrada analógica 3 configuración 0 a 0.5.
L4	Configura la persistencia la alarma de la entrada analógica 3 configuración 0 a 60 segundos
L5	Guarda en el registro matemático 0 el valor del umbral máximo en la alarma de la entrada analógica 3 configuración 0.
L6	Guarda en el registro matemático 1 el valor del umbral mínimo en la alarma de la entrada analógica 3 configuración 0.

Consultar y modificar alarmas de las diferentes configuraciones (opcional).

Cada interfaz puede tener hasta 4 configuraciones de alarmas independientes, identificadas como 0, 1, 2 y 3. Se accede a cada una de ellas especificando el número de la alarma en la propiedad correspondiente. Por ejemplo, "MAX1" corresponde a la configuración 1. Si no se especifica el número de configuración de la alarma, se considera que se está configurando la alarma número 0.

Ejemplo de modificación de las configuraciones de alarma 0 y 1 de la entrada analógica 3.

```
L1  ALM_AI3.MAX = 3.75
L2  ALM_AI3.MIN = 1.25
L3  ALM_AI3.HYS = 0.5
L4  ALM_AI3.PER = 60

L5  ALM_AI3.MAX1 = 4.15
L6  ALM_AI3.MIN1 = 1.5
L7  ALM_AI3.HYS1 = 0.75
L8  ALM_AI3.PER1 = 300
```

Línea	Descripción
L1	Configura el umbral máximo en la alarma de la entrada analógica 3 configuración 0 a 3.75.
L2	Configura el umbral mínimo en la alarma de la entrada analógica 3 configuración 0 a 1.25.
L3	Configura la histéresis en la alarma de la entrada analógica 3 configuración 0 a 0.5.
L4	Configura la persistencia la alarma de la entrada analógica 3 configuración 0 a 60 segundos
L5	Configura el umbral máximo en la alarma de la entrada analógica 3 configuración 1 a 4.15
L6	Configura el umbral mínimo en la alarma de la entrada analógica 3 configuración 1 a 1.5.
L7	Configura la histéresis en la alarma de la entrada analógica 3 configuración 1 a 0.75.
L8	Configura la persistencia la alarma de la entrada analógica 3 configuración 1 a 300 segudos.

12 - BLOQUES TEMPORIZADORES

MicroPLC-II permite definir hasta 24 bloques temporizadores. Se utiliza el símbolo ":" para asignar un alias al temporizador.

Los alias no soportan acentos y otros caracteres especiales, como la letra "Ñ".

Tipos de bloques temporizadores soportados:

	Descripción
RTO	Temporizador acumulador
TON	Temporizador de conexión
TOF	Temporizador de desconexión
TP	Temporizador de pulso
TW	Temporizador semanal
CT	Temporizador Cíclico

En la misma línea que se declara un bloque temporizador, se permite configurar sus propiedades separándolas con comas ",".

Ejemplo:

Declaración en diferentes líneas:

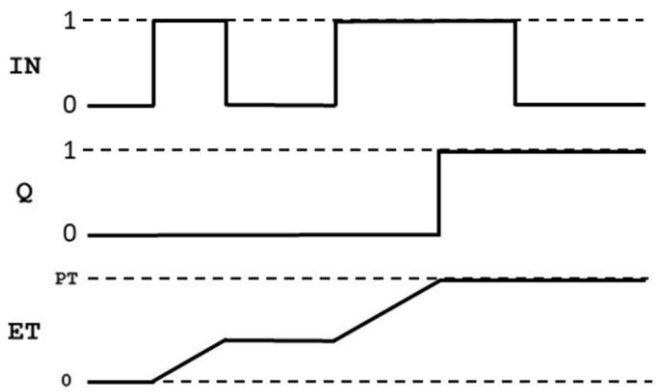
```
L1  #INIT
L2  TON : TEMPORIZADOR
L2  TEMPORIZADOR.PT = 10
L4  #END_INIT
```

Declaración en una misma línea:

```
L1  #INIT
L2   TON : TEMPORIZADOR, PT=10
L3  #END_INIT
```

12.1 TEMPORIZADOR ACUMULADOR DE TIEMPO (RTO)

Mientras la entrada esté activa, se incrementa un contador interno a razón de una vez por segundo. Al desactivar la entrada, el contador interno no se reinicia y continuará el conteo en la próxima activación. Al alcanzar un tiempo máximo configurable, el temporizador activará la salida.



	Propiedad	Descripción	Tipo
RTO	IN	Entrada. Mientras esté a 1, incrementará el contador ET a razón de una vez por segundo. Al recibir un 0, detiene el incremento del contador ET pero no lo resetea.	BOOL
	PT	Entrada. Umbral de conexión. Configura el periodo del temporizador. Rango de valores: 0 … 16777215.	INT
	R	Entrada. Al cambiar de 0 a 1, se iguala a 0 el valor del ET y se iguala a 0 la salida Q.	BOOL
	ET	Salida. Devuelve el tiempo acumulado expresado en segundos.	INT
	Q	Salida. Cambia de 0 a 1 cuando ET se iguala con PT.	BOOL

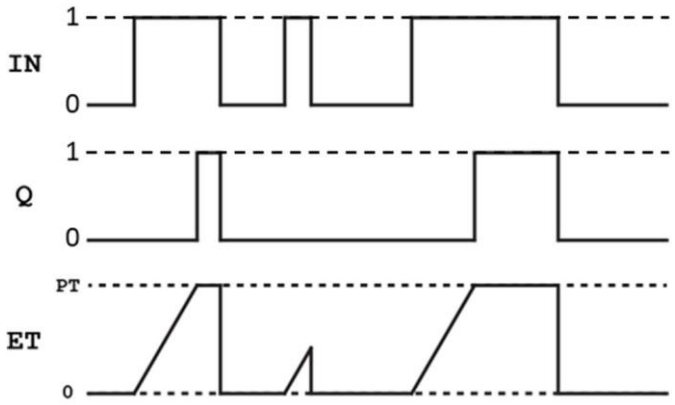
Registrar el tiempo de encendido de un grupo de luminarias y alertar tras alcanzar las 4500 horas de uso para su sustitución.

```
L1  #INIT
L2  DI0 : LUCES_ON
L3  DI1 : RESET
L4  DO0 : ALARMA
L5  M0 : TIEMPO
L6  RTO: TEMPORIZADOR, PT = 4500 * 3600
L7  TEMPORIZADOR.ET=TIEMPO
L8  #END_INIT
L9  TEMPORIZADOR.IN = LUCES_ON
L10 TEMPORIZADOR.R = RESET
L11 TIEMPO = TEMPORIZADOR.ET
L12 ALARMA = TEMPORIZADOR.Q
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script
L2	Asigna el alias "LUCES_ON" a la entrada digital 0. Esta entrada estará vinculada al interruptor de encendido de las luminarias.
L3	Asigna el alias "RESET" a la entrada digital 1.
L4	Asigna el alias "ALARMA" a la salida digital 0.
L5	Asigna el alias "TIEMPO" al registro matemático 0 "M0". Se utiliza un registro matemático en lugar de una variable para que este valor no se pierda en caso de fallo en el suministro eléctrico.
L6	Declara "TEMPORIZADOR" como un temporizador acumulador y configura el umbral de la conexión (PT) a 16200000 segundos (4500 horas).
L7	Inicializa el tiempo acumulado el temporizador "TEMPORIZADOR.ET" con el valor de la variable "TEIMPO". Relacionado con la L11
L8	Palabra clave para finalizar la sección de inicialización.
L9	La entrada IN de "TEMPORIZADOR" se activa con la señal "LUCES_ON".
L10	La entrada R de "TEMPORIZADOR" se activa con la señal "RESET".
L11	La variable "TIEMPO" se iguala con la salida ET de "TEMPORIZADOR".
L12	La señal "ALARMA" se iguala con la salida Q de "TEMPORIZADOR".

12.2 TEMPORIZADOR DE CONEXIÓN (TON)

Se activa la salida tras un tiempo parametrizable.



	Propiedad	Descripción	Tipo
TON	IN	Entrada. Al cambiar de 0 a 1 se inicia el tiempo para activar la salida Q tras el retardo de tiempo PT configurado.	BOOL

Propiedad	Descripción	Tipo
PT	Entrada. Umbral de conexión. Configura el periodo del temporizador. Rango de valores: 0 … 16777215.	INT
R	Entrada. Al cambiar de 0 a 1, se iguala a 0 el valor del ET y se iguala a 0 la salida Q.	BOOL
ET	Salida. Devuelve el tiempo que lleva activada la entrada IN expresado en segundos. Este tiempo se iguala al 0 cuando la entrada IN cambia de 1 a 0 y si la entrada R cambia de 0 a 1.	INT
Q	Salida. Cambia de 0 a 1 cuando el tiempo de activación transcurrido alcanza el tiempo PT mientras la entrada IN sea igual a 1.	BOOL

Activar una salida digital tras una persistencia de 30 segundos con respecto a la entrada digital.

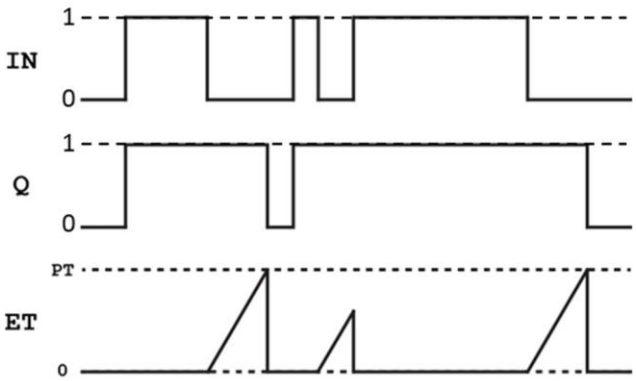
```
L1   #INIT
L2   DI0 : FALLORED
L3   DI1 : RESET
L4   DO0 : ALARMA
L5   INT : CONTADOR
L6   TON : TEMPORIZADOR, PT = 30
L7   #END_INIT
L8   TEMPORIZADOR.IN = FALLORED
L9   TEMPORIZADOR.R = RESET
L10  CONTADOR = TEMPORIZADOR.ET
L11  ALARMA = TEMPORIZADOR.Q
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script
L2	Asigna el alias “FALLORED” a la entrada digital 0.
L3	Asigna el alias “RESET” a la entrada digital 1.
L4	Asigna el alias “ALARMA” a la salida digital 0.
L5	Declara “CONTADOR” como una variable de tipo INT.
L6	Declara “TEMPORIZADOR” como un temporizador de conexión y configura el tiempo de retardo de conexión (PT) a 30 segundos.
L7	Palabra clave para finalizar la sección de inicialización.
L8	La entrada IN de “TEMPORIZADOR” se activa con la señal “FALLORED”.
L9	La entrada R de “TEMPORIZADOR” se activa con la señal “RESET”.

Línea	Descripción
L10	La variable “CONTADOR” se iguala con la salida ET de “TEMPORIZADOR”.
L11	La señal “ALARMA” se iguala con la salida Q de “TEMPORIZADOR”.

12.3 TEMPORIZADOR DE DESCONEXIÓN (TOF)

Se desactiva la salida tras un tiempo parametrizable.



	Propiedad	Descripción	Tipo
TOFF	IN	Entrada. Al cambiar de 0 a 1 la salida Q se iguala a 1. Al cambiar de 1 a 0 se inicia el tiempo para cambiar de 1 a 0 la salida Q. El tiempo de retardo se configura en la propiedad PT.	BOOL
	PT	Entrada. Umbral de conexión. Configura el periodo del temporizador. Rango de valores: 0 … 16777215.	INT
	R	Entrada. Al cambiar de 0 a 1, se iguala a 0 el valor del ET y se iguala a 0 la salida Q.	BOOL
	ET	Salida. Devuelve el tiempo de desconexión. Este tiempo se expresa en segundos e indica cuanto lleva la entrada IN igualada a 0 tras estar igualada a 1. Este contador se iguala a 0 tras igualarse con PT o al cambiar de 0 a 1 la entrada R.	INT
	Q	Salida. Cambia de 0 a 1 al detectar en la entrada IN un cambio de 0 a 1. Cambia de 1 a 0 cuando el tiempo de desconexión alcanza el tiempo PT.	BOOL

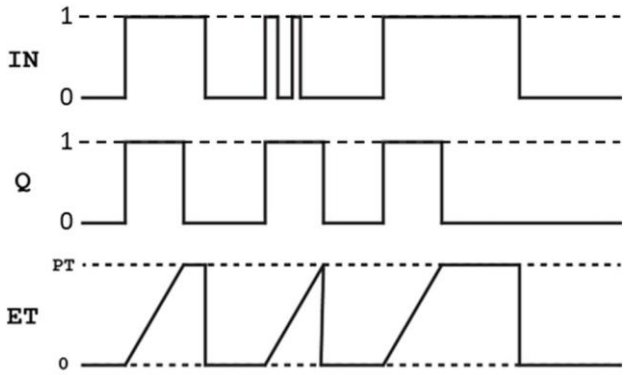
Desactivar una salida digital 45 segundos después de la desactivación de una entrada digital.

```
L1  #INIT
L2  DI0 : MARCHA
L3  DI1 : RESET
L4  DO0 : MOTOR
L5  INT : CONTADOR
L6  TOF : TEMPORIZADOR, PT = 45
L7  #END_INIT
L8  TEMPORIZADOR.IN = MARCHA
L9  TEMPORIZADOR.R = RESET
L10 CONTADOR = TEMPORIZADOR.ET
L11 MOTOR = TEMPORIZADOR.Q
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script
L2	Asigna el alias “MARCHA” a la entrada digital 0.
L3	Asigna el alias “RESET” a la entrada digital 1.
L4	Asigna el alias “MOTOR” a la salida digital 0.
L5	Declara “CONTADOR” como una variable de tipo INT.
L6	Declara “TEMPORIZADOR” como un temporizador de desconexión y configura el tiempo de retardo de desconexión (PT) a 45 segundos.
L7	Palabra clave para finalizar la sección de inicialización.
L8	La entrada IN de “TEMPORIZADOR” se activa con la señal “MARCHA”.
L9	La entrada R de “TEMPORIZADOR” se activa con la señal “RESET”.
L10	La variable “CONTADOR” se iguala con la salida ET de “TEMPORIZADOR”.
L11	La señal “MOTOR” se iguala con la salida Q de “TEMPORIZADOR”.

12.4 TEMPORIZADOR DE PULSO (TP)

El temporizador de pulsos genera un pulso en la salida cuando la entrada cambia de 0 a 1. La duración del pulso es configurable.



	Propiedad	Descripción	Tipo
TP	IN	Entrada. Cuando pasa de 0 a 1 se iguala la salida Q a 1 durante el tiempo que se haya programado en PT, una vez transcurrido el tiempo programado el parámetro Q se desactiva. La salida Q permanecerá en 1 durante el tiempo programado, aunque el parámetro IN pase a 0.	BOOL
	PT	Entrada. Ancho de pulso. Rango de valores: 0 … 16777215.	INT
	R	Entrada. Al cambiar de 0 a 1, se iguala a 0 el valor del ET y se iguala a 0 la salida Q.	BOOL
	ET	Salida. Devuelve el tiempo que lleva activada la salida Q expresado en segundos. Deja de contar una vez alcanza el valor PT. Este tiempo se iguala a 0 al igualar a 0 la entrada IN después de haber alcanzado el valor PT y al igualar a 1 la entrada R.	INT
	Q	Salida. Cambia de 0 a 1 cuando se recibe un 1 en la entrada IN. Después de que la salida Q haya estado encendida el tiempo PT, cambiará a 0 automáticamente independientemente del valor en la entrada IN.	BOOL

Al pulsar un botón, genera un pulso de salida de 2 segundos.

```
L1  #INIT
L2  DI0 : BOTON
L3  DO0 : PULSO_2S
L4  TP : TEMPORIZADOR, PT = 2
L5  #END_INIT
L6  TEMPORIZADOR.IN = BOTON
L7  PULSO_2S = TEMPORIZADOR.Q
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script
L2	Asigna el alias "BOTON" a la entrada digital 0.
L3	Asigna el alias "PULSO_2S" a la salida digital 0.
L4	Declara "TEMPORIZADOR" como un temporizador de pulso y configura el tiempo del pulso (PT) a 2 segundos.
L5	Palabra clave para finalizar la sección de inicialización.
L6	La activación del temporizador se controla con "BOTON"
L7	La señal "PULSO_2S" se iguala con la salida Q de "TEMPORIZADOR".

12.5 TEMPORIZADOR SEMANAL (TW)

La salida de este temporizador se controla programando días de la semana y la hora de activación y desactivación.

	Propiedad	Descripción	Tipo
TW	WEEK	Entrada. Selector de días de la semana. Indicar cada uno de los días de la semana por separado, siendo: Lunes=1, Martes=2, Miércoles=3, Jueves=4, Viernes=5, Sábado=6, Domingo=7.	STRING
	ON	Entrada. Hora de activación en formato HHMM, donde HH representa la hora y MM los minutos.	STRING
	OFF	Entrada. Hora de desactivación en formato HHMM, donde HH representa la hora y MM los minutos.	STRING
	Q	Salida. Cambia de 0 a 1 cuando se cumplen las condiciones de activación: día de la semana + hora de activación. Cambia de 1 a 0 cuando se cumplen las condiciones de desactivación: día de la semana + hora de desactivación.	BOOL

Activar una salida digital según programación semanal.

```
L1  #INIT
L2  DO0 : MARCHA_BOMBA
L3  TW : TEMPORIZADOR, WEEK = 12345, ON = 1230, OFF = 1845
L4  #END_INIT
L5  MARCHA_BOMBA = TEMPORIZADOR.Q
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script
L2	Asignar el alias "MARCHA_BOMBA" a la salida digital 0. Este canal controla el arranque de la bomba 0.
L3	Declara "TEMPORIZADOR" como un temporizador semanal y lo configura para que esté habilitado de lunes a viernes de 12:30 a 18:45
L4	Palabra clave para finalizar la sección de inicialización.
L5	La señal "MARCHA_BOMBA" se iguala con la salida Q de "TEMPORIZADOR".

12.6 TEMPORIZADORES CÍCLICOS (CT)

La salida se activa cuando transcurre el tiempo asociado a cada temporizador.

	Descripción
CT.PPM	Se activa cuanto cambia de minuto.
CT.PPH	Se activa cuando cambia de hora.
CT.PPD	Se activa cuando cambia de día.
CT.PPW	Se activa cuando cambia de semana.
CT.PPMO	Se activa cuando cambia de mes.

Calcular el volumen diario de un contador

```
L1  #INIT
L2  CNT0.M3 : TOTALIZADOR
L3  M0 : TOTALIZADOR_PARCIAL
L4  REAL : VOLUMEN_DIARIO
L5  #END_INIT
L6  VOLUMEN_DIARIO = TOTALIZADOR – TOTALIZADOR PARCIAL
L7  IF CT.PPD ; TOTALIZADOR_PARCIAL = TOTALIZADOR ; CT.PPD=0
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias "TOTALIZADOR" al contador en metros cúbicos de la entrada digital 0.
L3	Asigna el alias "TOTALIZADOR_PARCIAL" al registro matemático 0.
L4	Declara "VOLUMEN_DIARIO" como una variable de tipo REAL.
L5	Palabra clave para finalizar la sección de inicialización.
L6	Calcula el volumen diario restando el valor que tenía el totalizador al inicio del día al valor actual del totalizador.
L7	Cuando detecta el cambio de día (CT.PPD=1), guarda un nuevo valor en el totalizador parcial, y vuelve a poder a cero CT.PPD para dejarlo preparado para el próximo día.

13 - MÓDULO DE ENVÍO Y RECEPCIÓN DE SMS

MicroPLC-II implementa de manera nativa la funcionalidad para el intercambio de información entre estaciones mediante SMS, previsto como método de comunicación redundante para la funcionalidad de PUBLICACION/SUSCRIPCION.

Esta funcionalidad permite el intercambio de variables enteras entre estaciones, segmentadas en 4 canales lógicos.

13.1 ENVÍO DE SMS

La instrucción SMSTX provoca el envío de un SMS a otra estación Microcom, tomando como argumentos el teléfono de destino y un número variable de parámetros con los valores a enviar.

SINTAXIS

```
SMSTX teléfono,canal_0,canal_1,canal_2,canal_3
```

Donde:
- Teléfono: Teléfono de destino del SMS.
- Canal_0: Valor entero de 32 bits que se enviara por el canal lógico 0.
- Canal_1: Valor entero de 32 bits que se enviara por el canal lógico 1.
- Canal_2: Valor entero de 32 bits que se enviara por el canal lógico 2.
- Canal_3: Valor entero de 32 bits que se enviara por el canal lógico 3.

La inclusión de cada canal es opcional.

Ejemplos:

Envío del valor 1 en el canal 0 al teléfono +34637885326:

```
SMSTX +34637885326,1
```

Envío del valor 1 en el canal 0 y el valor 100 en el canal 1 al teléfono +34637885326:

```
SMSTX +34637885326,1,100
```

Envío del valor 200 en el canal 3 al teléfono +34637885326:

```
SMSTX +34637885326,,,,200
```

Envío del valor de la variable MODO en el canal 2 al teléfono +34637885326:

```
SMSTX +34637885326,,MODO
```

MODO es una variable de tipo INT declarada previamente.

13.2 RECEPCIÓN DE SMS

La variable SMSRX permite la lectura de la información recibida mediante SMS.

	Propiedad	Descripción	Tipo
SMSRX	CHx	Lectura/escritura. Valor recibido en el canal x. El valor inicial al reset del sistema es 0.	INT

Ejemplo:

Activar salida digital 1 al recibir el valor 3 en el canal 0:

```
IF SMSRX.CH0 = 3 ; DO1 = 1; SMSRX.CH0 = 0
```

Tras activar la salida digital 1, se reinicia el canal 0 a 0. Éste es un método que permite detectar la llegada de un nuevo valor de manera sencilla.

13.3 DECLARACIÓN DE TELÉFONOS AUTORIZADOS

El módulo de recepción de SMS para comunicación entre estaciones sólo acepta SMS provenientes de una lista cerrada de teléfonos autorizados que es independiente de la lista general para envío y recepción de comandos vía SMS (pantalla general de MicroConf). La instrucción SMSAUT permite declarar hasta cuatro teléfonos autorizados.

SINTAXIS:

SMSAUT *lista de teléfonos*

Donde:

- Lista teléfonos: Entre uno y cuatro números de teléfonos separados por comas.

Los números de teléfono deben expresarse en formato internacional salvo que se trate de números corporativos cortos

Ejemplo:

```
SMSAUT +34637885326 , 3005
```

14 - MÓDULO DE COMUNICACIÓN MODBUS

MicroPLC-II incorpora de manera nativa un módulo para la transmisión de datos mediante MODBUS-RTU o MODBUS-TCP. Se pueden definir hasta 48 instancias de este módulo, identificado como MBOUT.

<table>
<tr><th></th><th>Propiedad</th><th>Descripción</th><th>Tipo</th></tr>
<tr><td rowspan="5">MBOUT</td><td>SL</td><td>Dirección del esclavo MODBUS al que se desea escribir.
Implica la operación en modo MODBUS-RTU.</td><td>INT</td></tr>
<tr><td>SRV</td><td>Identificador del servidor MODBUS-TCP al que se desea escribir
Obtenido mediante la instrucción IMPORT MBSRV.
Implica la operación en modo MODBUS-TCP.</td><td>STRING</td></tr>
<tr><td>RA</td><td>Dirección de registro que se desea escribir según notación MODBUS estándar:
4xxxx: Escritura en holding register.
0xxxx: Escritura en coil.</td><td>INT</td></tr>
<tr><td>MODE</td><td>Tipo de variable a escribir:
<table>
<tr><th>Modo</th><th>Descripción</th></tr>
<tr><td>SET.MBI16</td><td>Entero de 16 bits.</td></tr>
<tr><td>SET.MBI32</td><td>Entero de 32 bits. Big endian.</td></tr>
<tr><td>SET.MBI32LE</td><td>Entero de 32bits. Litle endian.</td></tr>
<tr><td>SET.MBFLT</td><td>Float de 32 bits. Big endian.</td></tr>
<tr><td>SET.MBFLTLE</td><td>Float de 32bits. Litle endian.</td></tr>
<tr><td>SET.MBCOIL</td><td>Booleano.</td></tr>
</table></td><td>STRING</td></tr>
<tr><td>VLD</td><td>Flag de validez. Verdadero si el valor se transmitió con éxito.</td><td>BOOL</td></tr>
</table>

14.1 IMPORTACION IDENTIFICADOR SERVIDOR MODBUS-TCP

Para la importación de los identificadores de servidores MODBUS-TCP se dispone de la instrucción IMPORT MBSRV:

SINTAXIS:

```
IMPORT MBSRV : servidor
```

Donde:

- Servidor: Nombre del servidor a importar según la declaración en la sección MODBUS del MicroConf.

Ejemplo:

```
IMPORT MBSRV : PLC1
```

Envío del valor 124 a la dirección 40001 del esclavo 1. El valor se envía como un entero de 32 bits (big endian). Modo MODBUS-RTU.

```
L1  #INIT
L2   MBOUT : D_NIVEL, SL = 1, RA = 40001, MODE = SET.MBI32
L3  #END_INIT

L4    D_NIVEL=124
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias "D_NIVEL" a una variable de escritura MODBUS configurada para escribir en el esclavo 1, dirección 40001 un valor de 32 bits big endian.
L3	Palabra clave para finalizar la sección de inicialización.
L4	Asigna el valor 124 al canal de escritura MODBUS.

Envío del valor de la entrada analógica 0 al servidor MODBUS-TCP con nombre PLC1. El valor se envía como un numero de coma flotante (big endian). Modo MODBUS-TCP.

```
L1  #INIT
L2   IMPORT MBSRV : PLC1
L3   MBOUT : D_NIVEL, SRV = PLC1, RA = 40001, MODE = SET.MBFLT
L4  #END_INIT

L5    D_NIVEL=AI0
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Importa el identificador del servidor MODBUS-TCP con nombre "PLC1".
L3	Asigna el alias "D_NIVEL" a una variable de escritura MODBUS configurada para escribir en el servidor PLC1, dirección 40001 un valor de coma flotante.
L4	Palabra clave para finalizar la sección de inicialización.
L5	Asigna el valor en la entrada analógica 0 al canal de escritura MODBUS.

15 - MÓDULO DE CONTROL DE BOMBAS (PUMP)

MicroPLC-II integra un módulo lógico para el control de hasta 6 bombas. Este módulo gestiona la alternancia, modo de funcionamiento, el conteo de horas, el contador de arranques, las entradas de inhibición de bomba, las entradas de confirmación de marcha y otros aspectos necesarios en la gestión de un automatismo de bombeo.

Para adaptarse a diferentes arquitecturas de instalación, el módulo PUMP ofrece dos esquemas de operación: el primero, de alternancia simple, distribuye equitativamente el trabajo entre todas las bombas; el segundo, en modo principal/apoyo, permite designar unas unidades como "principales" y otras como "apoyo", estableciendo prioridades de servicio entre ellas. Gracias a esta flexibilidad, la solución se ajusta prácticamente a cualquier necesidad de automatización de bombeo.

Por otro lado, con el fin de mantener la compatibilidad con versiones anteriores de firmware, el módulo arranca siempre en modo de prioridad simple —es decir, todas las bombas actúan como auxiliares con prioridad 3— y, en el momento en que se configura al menos una bomba como principal, cambia automáticamente al modo principal/auxiliar.

En la misma línea que se declara el módulo de control de bombas, se permite configurar sus propiedades.

Ejemplo:

Declaración en diferentes líneas:

```
L1  #INIT
L2  PUMP.NUM = 2
L3  PUMP.REQ = 2
L4  PUMP.DON = 5
L5  #END_INIT
```

Declaración en una misma línea:

```
L1  #INIT
L2  PUMP.NUM = 2, REQ = 2, DON =5
L3  #END_INIT
```

	Propiedad.	Descripción	Tipo
PUMP	NUM	Entrada. Configurar el número total de bombas en el sistema. Rango de valores: 2 … 6.	INT
	REQ	Entrada. Indica el número de bombas que se desean encendidas actualmente.	INT
	AUTO	Entrada. Habilita el modo de automático. Se pondrán en marcha las bombas que indique la propiedad REQ teniendo en cuenta la alternancia.	BOOL
	HAND	Entrada. Habilita el modo manual. Permite activar manualmente las bombas utilizando la propiedad MANx.	BOOL
	OFF	Entrada. Activa el modo "paro" del sistema. Todas las bombas se mantienen en paro.	BOOL

Propiedad.	Descripción	Tipo
MANx	Entrada. Activa manualmente la salida indicada cuando PUMP.HAND está a 1.	BOOL
LEADx	Entrada. Configura la bomba x como prioritaria o de apoyo. Rango de valores: 1 → Prioritaria 0 → Apoyo	BOOL
PRIOx	Entrada. Configura la prioridad de la bomba x. Rango de valores: 0 → Deshabilitada 1 → Prioridad máxima 2 → Prioridad media 3 → Prioridad baja	INT
DON	Entrada. Retardo de encendido expresado en segundos. Tras una operación de encendido de bombas no se ejecutará ninguna otra operación (encendido o apagado) durante el tiempo indicado. Rango de valores: 0 … 255.	INT
DOFF	Entrada. Retardo de apagado expresado en segundos. Tras una operación de apagado de bombas no se ejecutará ninguna otra operación (encendido o apagado) durante el tiempo indicado. Rango de valores: 0 … 255.	INT
PRTO	Entrada. Tiempo máximo para recibir la confirmación de marcha de bombas expresado en segundos. 0 desactiva la función. Si no se recibe la señal de confirmación de marcha dentro del período configurado, la bomba se desactiva, se activa la señal PRFx correspondiente y se enciende la siguiente bomba según el orden de alternancia.	INT
PRCx	Entrada. Confirmación de marcha de la bomba x.	BOOL
PRFx	Salida. Fallo de encendido en bomba x. Indica la condición de fallo de la bomba x cuando no se recibe a tiempo la señal de confirmación de marcha. Una vez activada la salida de fallo de encendido la bomba quedará deshabilitada hasta que se reinicie el fallo utilizando la propiedad Rx.	BOOL
Rx	Entrada. Reinicia el fallo de encendido de la bomba x.	BOOL
DISx	Entrada. Inhabilita la bomba x. Por ejemplo, por estar fuera de servicio.	INT
Qx	Salida. Orden de marcha bomba x.	BOOL
ACx	Salida. Devuelve el tiempo de encendido acumulado en bomba conectada a la salida Qx. Tiempo expresado en segundos.	INT
STx	Salida. Devuelve el número de arranques de la bomba conectada a la salida digital x.	INT

Las memorias del módulo de control de bombas son volátiles. Se recomienda guardar los acumuladores de tiempo y de arranque en registros matemáticos (memorias no volátiles). Ver el siguiente ejemplo.

Llenado de un depósito con alternancia de bombas y cuenta horas.

```
L1   #INIT
L2   EXP20 : B0ON
L3   EXP21 : B1ON
L4   DI0 : P0DIS
L5   DI1 : P1DIS
L6   DI2 : MARCHA_B0
L7   DI3 : MARCHA_B1
L8   AI0 : NIVEL
L9   M8 : P0AC
L10  M9 : P1AC
L11  F0 : RF_B0
L12  F1 : RF_B1
L13  BOOL : BON
L14  REM DOS BOMBAS. FUNCIONAMIENTO CICLICO
L15  PUMP.NUM = 2, DON = 5, DOFF = 5, PRTO=10
L16  PUMP.AC0 = P0AC, AC1 = P1AC
L17  #END_INIT
L18  IF NIVEL < 0.5 ; BON = TRUE
L19  IF NIVEL > 3.2 ; BON = FALSE
L20  PUMP.REQ=BON
L21  REM RESET FALLO DE ARRANQUE
L22  IF RF_B0; PUMP.R0=RF_B0; RF_B0=0
L23  IF RF_B1; PUMP.R1=RF_B1; RF_B1=0
L24  REM E/S MODULO PUMP
L25  B0ON = PUMP.Q0
L26  B1ON = PUMP.Q1
L27  PUMP.PRC0=MARCHA_B0
L28  PUMP.PRC1=MARCHA_B1
L29  PUMP.DIS0 = P0DIS
L30  PUMP.DIS1 = P1DIS
L31  P0AC = PUMP.AC0
L32  P1AC = PUMP.AC1
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias "B0ON" a la expansión 20, "EXP20". Este canal controla el arranque de bomba 0.
L3	Asigna el alias "B1ON" a la expansión 21, "EXP21". Este canal controla el arranque de bomba 1.
L4	Asigna el alias "P0DIS" a la entada digital 0, "DI0". Este canal inhabilita el uso de la bomba 0.
L5	Asigna el alias "P1DIS" a la entada digital 1, "DI1". Este canal inhabilita el uso de la bomba 1.
L6	Asigna el alias "MARCHA_B0" a la entada digital 2, "DI2". Este canal indica la confirmación de marcha de la bomba 0.
L7	Asigna el alias "MARCHA_B1" a la entada digital 3, "DI3". Este canal indica la confirmación de marcha de la bomba 1.
L8	Asigna el alias "NIVEL" a la entada analógica 0, "AI0". Este canal indica los metros de columna de agua del depósito.
L9	Asigna el alias "P0AC" al registro matemático 8, "M8". Esta memoria no volátil guardará el tiempo de encendido acumulado de la bomba 0.
L10	Asigna el alias "P1AC" al registro matemático 9, "M9". Esta memoria no volátil guardará el tiempo de encendido acumulado de la bomba 1.
L11	Asigna el alias "RF_B0" al flag 0, "F0". Este canal indica el reinicio del falo de arranque de la bomba 0.
L12	Asigna el alias "RF_B1" al flag 1, "F1". Este canal indica el reinicio del falo de arranque de la bomba 1.
L13	Declara "BON" como una variable de tipo BOOL.
L14	Comentario para describir el programa. Esta línea no se ejecuta.
L15	Configurar el módulo de control de bombas. Número de bombas a controlar = 2, tiempo funcionamiento continuo máximo = 30 minutos, retardo de apagado y encendido = 5 segundos.
L16	Inicializar el módulo de control de bombas. Igualar "PUMP.AC0", el tiempo de encendido acumulado para la bomba 0 con el valor de "P0AC". Igualar "PUMP.AC1", el tiempo de encendido acumulado para la bomba 1 con el valor de "P1AC". La memoria del bloque temporizador es volátil y se reiniciará en caso de que el equipo pierda el suministro eléctrico. Mediante esta línea de código se evita la pérdida de datos haciendo que cargue los valores guardados en los registros de memorias no volátiles. Relacionado con líneas de código L22 y L23.
L17	Palabra clave para finalizar la sección de inicialización.
L18	Si "NIVEL" es menos que 0,5 metros, la variable "BON" se activa indicando que hace falta agua.
L19	Si "NIVEL" es mayor que 3,2 metros, la variable "BON" se desactiva indicando que no hace falta agua.
L20	"PUMP.REQ" inda las bombas que tienen que estar encendidas en cada momento. Al igualarlo a "BON" activará una bomba cuando el nivel es menor que 0,5 metros y la apagará cuando el nivel es superior a 3,2 metros.

Línea	Descripción
L21	Comentario para describir el programa. Esta línea no se ejecuta.
L22	Si "RF_B0" es igual a 1, se escribe dicho valor en la propiedad que reinicia el fallo de arranque de la bomba 0 "PUMP.R0" y se iguala a cero el valor de "RF_B0". Previsto para que el usuario desde el sinóptico en Zeus pueda restaurar la bomba.
L23	Si "RF_B1" es igual a 1, se escribe dicho valor en la propiedad que reinicia el fallo de arranque de la bomba 0 "PUMP.R1" y se iguala a cero el valor de "RF_B1". Previsto para que el usuario desde el sinóptico en Zeus pueda restaurar la bomba.
L24	Comentario para describir el programa. Esta línea no se ejecuta.
L25	Iguala el estado de la orden de marcha de la bomba 0 "PUMP.Q0" con la salida digital que controla la bomba 0 "B0ON".
L26	Iguala el estado de la orden de marcha de la bomba 1 "PUMP.Q1" con la salida digital que controla la bomba 1 "B1ON".
L27	Iguala el estado del indicador de confirmación de marcha de la bomba 0 "MARCHA_B0" con la entrada del módulo de control correspondiente "PUMP.PRC0".
L28	Iguala el estado del indicador de confirmación de marcha de la bomba 1 "MARCHA_B1" con la entrada del módulo de control correspondiente "PUMP.PRC1".
L29	Iguala el estado del indicador de inhabilitación de la bomba 0 "P0DIS" con la entrada del módulo de control correspondiente "PUMP.DIS0".
L30	Iguala el estado del indicador de inhabilitación de la bomba 1 "P1DIS" con la entrada del módulo de control correspondiente "PUMP.DIS1".
L31	Guarda el tiempo de encendido de la bomba 0, "PUMP.AC0" en "P0AC". Se realiza para que este dato sea persistente y no se pierda en caso de pérdida de suministro eléctrico. Relacionado con línea de código L12.
L32	Guarda el tiempo de encendido de la bomba 1, "PUMP.AC1" en "P1AC". Se realiza para que este dato sea persistente y no se pierda en caso de pérdida de suministro eléctrico. Relacionado con línea de código L12.

16 - MÓDULO PID

El módulo PID de MicroPLC-II está diseñado para ofrecer un control preciso en aplicaciones de regulación continua. Su arquitectura soporta la ejecución de múltiples lazos de control independientes, cada uno con un tiempo de refresco de 1 segundo.

Para cada lazo se pueden ajustar de forma individual las constantes proporcional (Kp), integral (Ti) y derivativa (Td), así como definir límites de salida y rangos de acción, permitiendo adaptar la regulación a las características específicas de bombas, válvulas o cualquier otro actuador.

Además, el sistema incorpora funciones avanzadas de protección y optimización del lazo:

- Anti-windup en la parte integral, para evitar la saturación de la señal de control cuando la salida alcanza sus límites.
- Suavizado de consigna, que permite aplicar filtrado al setpoint y evita cambios bruscos que podrían desestabilizar el proceso.
- Tracking, que permite al controlador seguir un valor de referencia o trayectoria sin provocar transitorios bruscos al cambiar entre modos de control (por ejemplo, de manual a automático) o al alternar entre diferentes lazos PID.
- Selección de modos automático / manual, facilitando intervenciones manuales temporales sin perder los parámetros de regulación.

Por último, el módulo registra de forma continua los valores de proceso, consigna y salida de control, lo que facilita la puesta en marcha, el diagnóstico de comportamiento y la optimización de parámetros. Gracias a esta versatilidad, MicroPLC-II puede integrarse en una amplia variedad de sistemas de bombeo y procesos industriales que requieran un lazo PID robusto y de rápida respuesta.

En la misma línea que se declara el módulo de PID, se permite configurar sus propiedades.

Ejemplo:

Declaración en diferentes líneas:

```
L1  #INIT
L2  PID : REG
L3  REG.KP = 1
L4  REG.TI = 400
L5  REG.TD = 100
L6  RAMP = 0.1
L7  #END_INIT
```

Declaración en una misma línea:

```
L1  #INIT
L2  PID : REG, KP =1, TI =400, TD = 100, RAMP = 0.1
L3  #END_INIT
```

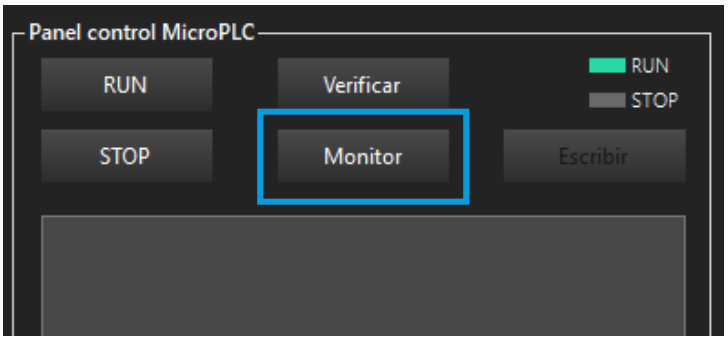
	Propiedad.	Descripción	Valores predef.	Tipo
PID	KP	Entrada. Ganancia proporcional.	0	REAL
	TI	Entrada. Tiempo integral. 0 inhibe. La acción integral corrige errores acumulados a lo largo del tiempo.	0	REAL
	TD	Entrada. Tiempo derivativo. La acción derivativa anticipa la evolución futura del error.	0	REAL
	MIN	Entrada. Permite configurar la salida mínima del PID	0	REAL
	MAX	Entrada. Permite configurar la salida máxima del PID	100	REAL
	DA	Entrada. Acción directa=true → Una acción mayor aumenta el valor de la variable de proceso. Acción directa=false → Una acción mayor disminuye el valor de la variable de proceso.	True	BOOL
	MAN	Entrada. Activa / Inhibe el modo manual. El modo manual provoca que la salida del PID se fije en el valor configurado en la propiedad MO	False	BOOL
	MO	Entrada. Permite configurar el valor de salida del PID cuando se activa el modo manual.	0	REAL
	TRK	Entrada. Activa / Inhibe el modo Tracking. El modo Tracking permite disponer de una realimentación externa para el PID.	False	BOOL
	TV	Entrada. Track value. Valor de realimentación para el modo Tracking	0	REAL
	DFF	Entrada. Factor de filtro derivativo. Atenúa las perturbaciones de alta frecuencia en la señal de error aplicando un **filtro de respuesta infinita** (IIR) al término derivativo.	0	REAL
	RAMP	Entrada. Permite configurar la velocidad de cambio del set point por cada ciclo de 1s.	0	REAL
	SP	Entrada. Set point. Valor objetivo del regulador.	0	REAL
	PV	Entrada. Variable de proceso. Valor de entrada actual.	0	REAL
	OUT	Salida. Variable de control. Valor de salida del PID	0	REAL

16.1 AJUSTE DEL MÓDULO PID

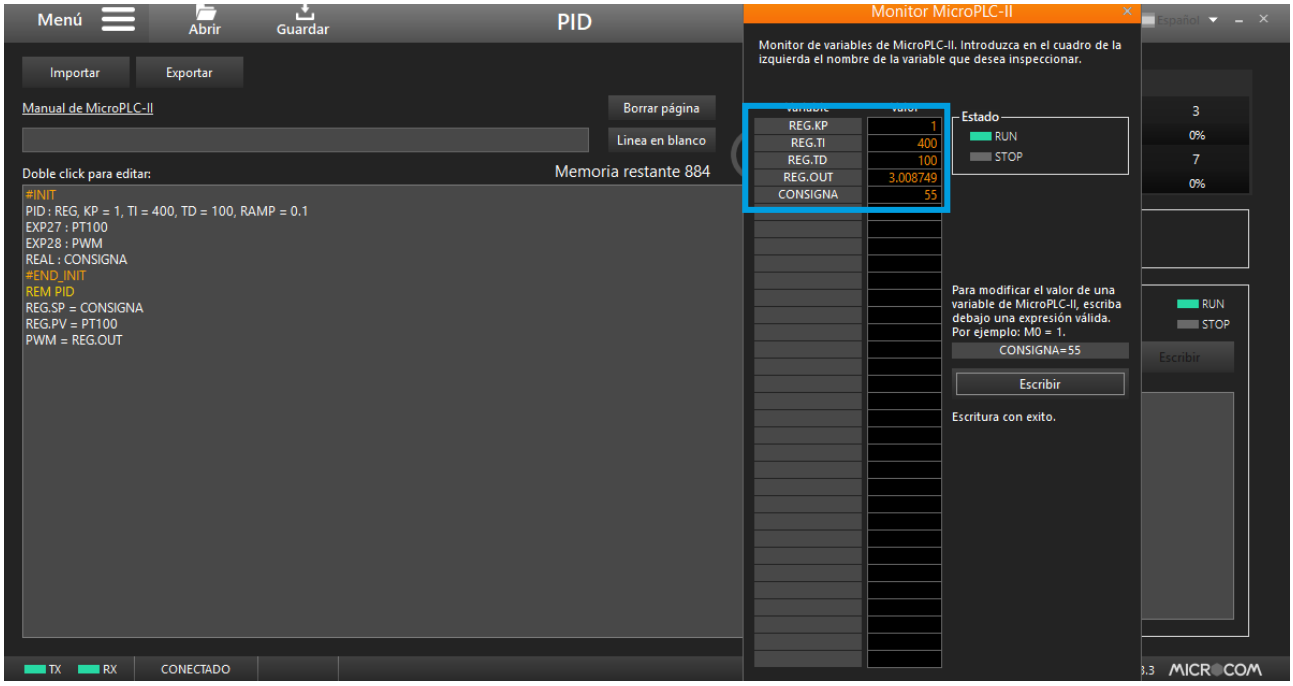
Para un correcto ajuste del módulo PID se recomienda utilizar el monitor del MicroPLC-II. Esta herramienta permite observar en tiempo real los valores del proceso, la consigna y la salida de control generada por el módulo PID. Asimismo, facilita la visualización y configuración precisa de los parámetros individuales del controlador, incluyendo la constante proporcional (Kp), el tiempo integral (Ti) y el tiempo derivativo (Td). Esta capacidad de monitoreo y ajuste permite al usuario optimizar fácilmente el comportamiento del lazo PID en función de las características específicas de cada aplicación.

Para un correcto ajuste se recomienda seguir los siguientes pasos:

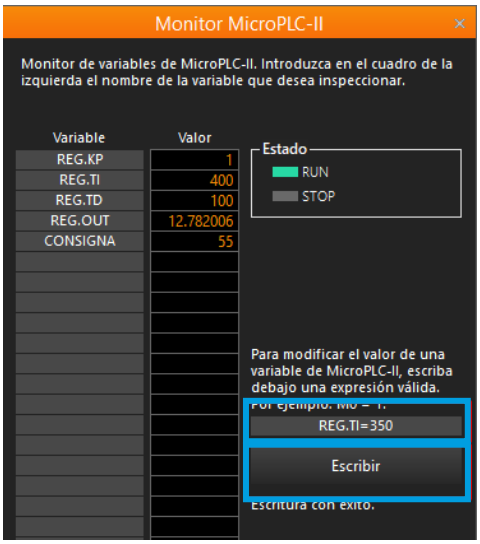
1. Abrir el monitor desde el panel de control de MicroPLC-II



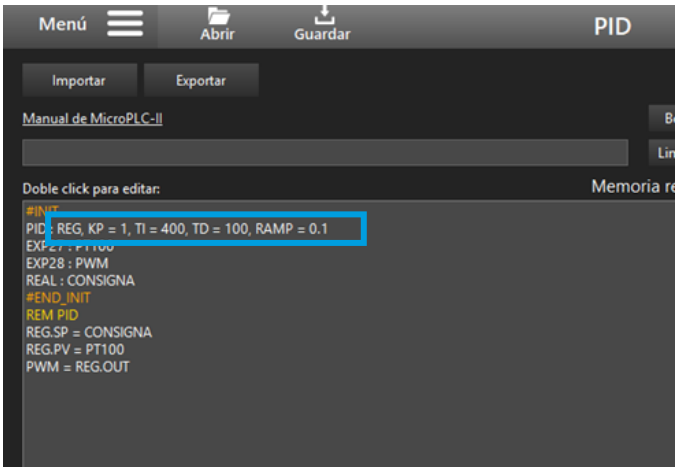
2. Configurar las propiedades del PID de interés para un correcto ajuste del sistema. Usualmente serán la ganancia proporcional, el tiempo de integración, el tiempo derivativo y la salida.



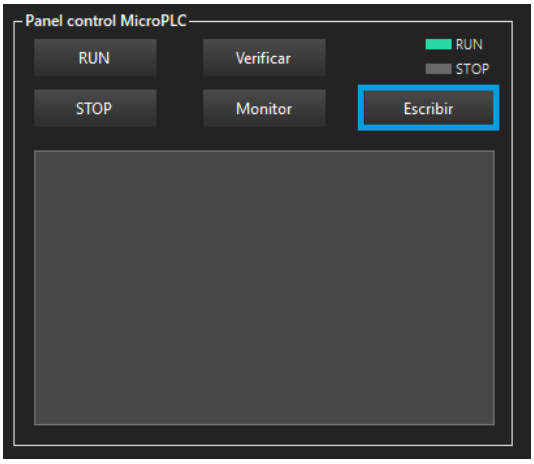
3 Modificar los ajustes de PID hasta conseguir la respuesta deseada en la salida.



4 Una vez ajustado correctamente el PID, escribir los valores correspondientes en las líneas donde se declara el PID para que el sistema arranque con los nuevos valores predefinidos.



5 Escribir el nuevo Script de MicroPLC-II desde el Manel de control para cargar los valores definitivos apara el módulo PID.



Regulación de la temperatura en un horno.

```
L1   #INIT
L2   PID : REG, KP =1, TI = 400, TD = 100, RAMP = 0.1
L3   EXP27 : PT100
L4   EXP28 : PWM
L5   REAL : CONSIGNA
L6   #END_INIT

L7   REM PID
L8   REG.SP = CONSIGNA
L9   REG.PV = PT100
L10  PWM = REG.OUT
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declara un PID con alias "REG", con una ganancia proporcional de 1, un tiempo integral de 400, un tiempo derivativo de 100 y una rampa de 0.1.
L3	Asigna el alias "PT100" a la expansión 27, "EXP27". En este canal se recibe la lectura de la temperatura.
L4	Asigna el alias "PWM" a la expansión 28, "EXP28". Desde esta salía analógica se envía la señal de salida al PWM.
L5	Declara "CONSIGNA" como una variable de tipo REAL.
L6	Palabra clave para finalizar la sección de inicialización.
L7	Comentario para describir el programa. Esta línea no se ejecuta.
L8	Introduce el valor de la variable "CONSIGNA" en la propiedad SP del PID. Esta propiedad es el set point del PID. El valor que tenemos como objetivo de temperatura
L9	Introduce el valor de la entrada "PT100" a la propiedad PV del PID. Esta propiedad es la variable de proceso. La referencia de temperatura real que tenemos del sistema
L10	Asocia la salida del PID a la salida analógica "PWM"

17 - EJEMPLOS DE PROGRAMACIÓN

17.1 TERMOSTATO

Ejemplo de activación automática de un sistema de calefacción dependiendo de la temperatura ambiente, la temperatura de activación o consigna y una histéresis o banda muerta.

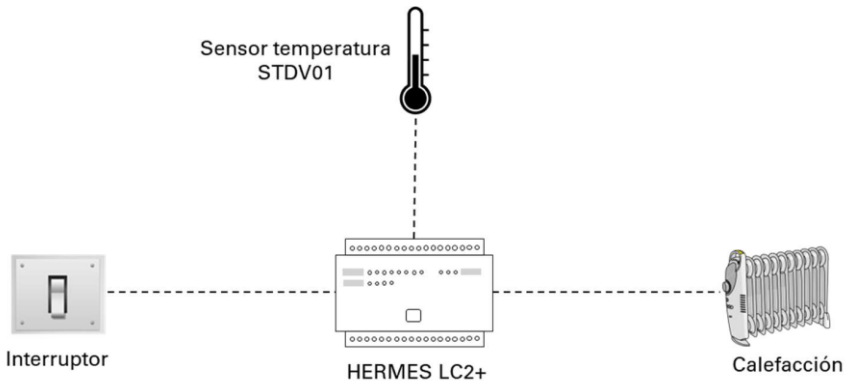
El sistema del ejemplo se compone de los siguientes elementos:

- Microcom Hermes LC2+. Equipo de telecontrol y autómata al que se conectan el resto de los elementos: sonda STDV01, interruptor habilitación encendido automático y el sistema calefacción. Este equipo ejecuta la lógica de programación.

- Sonda STDV01. Sonda de temperatura ambiente compatible con el Hermes LC2+.

- Interruptor para habilitar o deshabilitar manualmente el encendido automático de la caldera. Aunque el equipo de telecontrol considere que se debe encender la caldera, ésta sólo se encenderá si este interruptor está activado.

- Sistema de calefacción. Se pone en marcha al recibir la tensión de alimentación. En este ejemplo, el telecontrol a través de una de sus salidas a relé activará un contactor que, a su vez, permitirá el paso de la tensión de alimentación al sistema de calefacción.



Ejemplo de programación del sistema en lenguaje MICROPLC-II:

```
L1   #INIT
L2   DI0 : TERMO_ON ; REM ENTRADA ENCENDIDO CALEFACCION
L3   DO2 : CALE_ON ; REM SALIDA ENCENDER CALEFACCION
L4   PB0 : TSALA ; REM TEMPERATURA DE LA SALA
L5   REAL : CONSIGNA = 21 ; REM TEMPERATURA DE CONSGINA
L6   REAL : HIST = 0.5 ;  REM HISTERESIS
L7   BOOL : ENCENDER ; REM BANDERA TEMPERATURA INFERIOR A CONSIGNA
L8   #END_INIT
L9    REM TERMOSTATO
L10   IF TSALA > CONSIGNA ; ENCENDER = FALSE
L11   IF TSALA - HIST < CONSIGNA ; ENCENDER = TRUE
L12   CALE_ON = TERMO_ON AND ENCENDER
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias "TERMO_ON" a la entrada digital 0 "DI0". En esta entrada digital se conecta el interruptor para habilitar y deshabilitar manualmente el encendido automático de la caldera. Comentario.
L3	Asigna el alias "CALE_ON" a la salida digital 2 "DO2". Esta salida digital se conecta al relé que activa el sistema de calefacción. Comentario.
L4	Asigna el alias "TSALA" a la sonda 0 "PB0", se conecta una sonda de temperatura ambiente STDV01. Comentario.
L5	Declara "CONSIGNA" como una variable de tipo REAL. Inicializar la variable con el valor 21 (grados Celsius). En esta variable se configura la temperatura máxima a mantener en la sala. Comentario.
L6	Declara "HIST" como una variable de tipo REAL. Inicializar la variable con el valor 0.5 (grados celsius). En esta variable se configura la HISTERÉSIS y, en este caso, y junto con la consigna, sirve para configurar la temperatura de activación de la calefacción. Comentario.
L7	Declarar "ENCENDER" como una variable de tipo BOOL. Se utilizará como indicador de orden de marcha del sistema de calefacción. Esta variable se activa y se desactiva según la lógica de programación y utiliza el valor de la temperatura ambiente "TSALA", temperatura de consigna "CONSIGNA" y valor de histéresis "HISTÉRESIS". Comentario.
L8	Palabra clave para finalizar la sección de inicialización.
L9	Comentario para describir el programa. Esta línea no se ejecuta.
L10	Si la temperatura ambiente "TSALA" es mayor que la temperatura objetivo o de consigna "CONSIGNA", se desactiva la orden de encender sistema de calefacción, "ENCENDER" se iguala a 0. En este ejemplo, la desactivación tendrá lugar cuando la temperatura ambiente sea mayor que 21 grados Celsius.
L11	Si el resultado de la operación temperatura ambiente "TSALA" menos la histéresis "HIST" es menor que la temperatura objetivo o de "CONSIGNA", se activa la orden de encender el sistema de calefacción, "ENCENDER" se iguala a 1. En este ejemplo, la temperatura de activación se cumple cuando "TSALA" sea menor que 20.5 grados Celsius.
L12	Para que la salida digital que controla el encendido de la caldera se active ("CALE_ON" se iguala a 1), se tienen que cumplir las dos condiciones: el interruptor debe estar activo ("ENCENDER" es igual a 1) y la orden de activación por temperatura fuera de consigna está activa ("TERMO_ON" es igual a 1). Si una o ambas condiciones no se cumplen, no se activará la caldera.

17.2 INTERRUPTOR CREPUSCULAR O ASTRONÓMICO

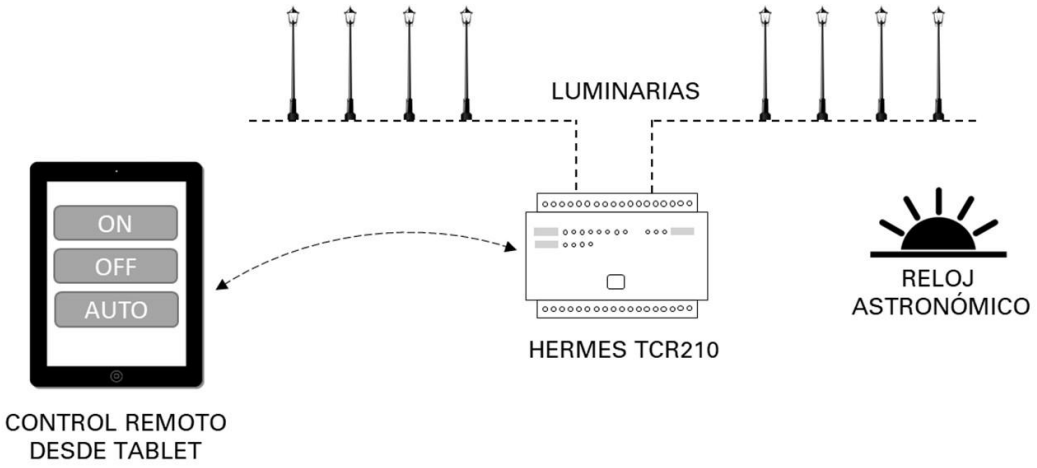
Ejemplo de activación automática y manual de luminarias en negocios y para alumbrado público.

Los interruptores crepusculares o astronómicos son programadores horarios cuyo funcionamiento se basa en el cálculo diario automático de las horas de salida y puesta del sol, teniendo para ello en cuenta las coordenadas del lugar geográfico en el que están siendo instalados. Su aplicación típica es el encendido de la iluminación al ocaso y su posterior apagado al amanecer. El cambio de horario de invierno a verano es automático. El sistema del ejemplo se compone de los siguientes elementos:

- Microcom Hermes TCR210. Equipo de telecontrol y autómata que controla el encendido y apagado del sistema de luminarias. Este equipo ejecuta la lógica de programación.

- Sistema de luminarias. Este sistema se pone en marcha al recibir la tensión de alimentación. En este ejemplo, el telecontrol a través de una de sus salidas activará un contactor que, a su vez, encenderá la iluminación.

- Plataforma ZEUS: Aplicación web y para smartphones que permite monitorizar y controlar remotamente equipos MICROCOM.



Ejemplo de programación de interruptor astronómico en lenguaje MICROPLC-II:

```
L1   #INIT
L2   DO3 : LUZ_ON ; REM SALIDA ENCENDIDO LUMINARIAS
L3   M0 : MODO ; REM 0→OFF 1→ON 2→AUTO
L4   INT : OFF = 0
L5   INT : ON = 1
L6   INT : AUTO = 2
L7   #END_INIT
L8   REM INT. CREPUSCULAR
L9   IF MODO = OFF ; LUZ_ON = FALSE
L10  IF MODO = ON ; LUZ_ON = TRUE
L11  IF MODO = AUTO ; LUZ_ON = NOW.SOD > SUNSET OR NOW.SOD < SUNRISE
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias "LUZ_ON" a la salida digital 3 "DO3". Esta salida está conectada al sistema de encendido de las luminarias. Comentario.
L3	Asigna el alias "MODO" al registro matemático 0 "M0". Indica el modo de funcionamiento de las luminarias. Disponemos de tres modos: "OFF", "ON" y "AUTO"
L4	Declara "OFF" como una variable de tipo INT y lo inicializa al valor 0. Esta variable se utiliza como una constante.
L5	Declara "ON" como una variable de tipo INT y lo inicializa al valor 1. Esta variable se utiliza como una constante.
L6	Declara "AUTO" como una variable de tipo INT y lo inicializa al valor 2. Esta variable se utiliza como una constante.
L7	Palabra clave para finalizar la sección de inicialización.
L8	Comentario para describir el programa. Esta línea no se ejecuta.
L9	Si "MODO" es igual a "OFF", "LUZ_ON" se iguala a 0
L10	Si "MODO" es igual a "ON", "LUZ_ON" se iguala a 1
L11	Si "MODO" es igual a 2, "LUZ_ON" se igualará a 1 si la hora actual (NOW.SOD) es mayor que la hora del anochecer (SUNSET) o menor que la hora del amanecer (SUNRISE). El resto del tiempo será igual a 0. Si el modo automático está activado, la orden de marcha de las luminarias se activa al anochecer y se desactiva al amanecer.

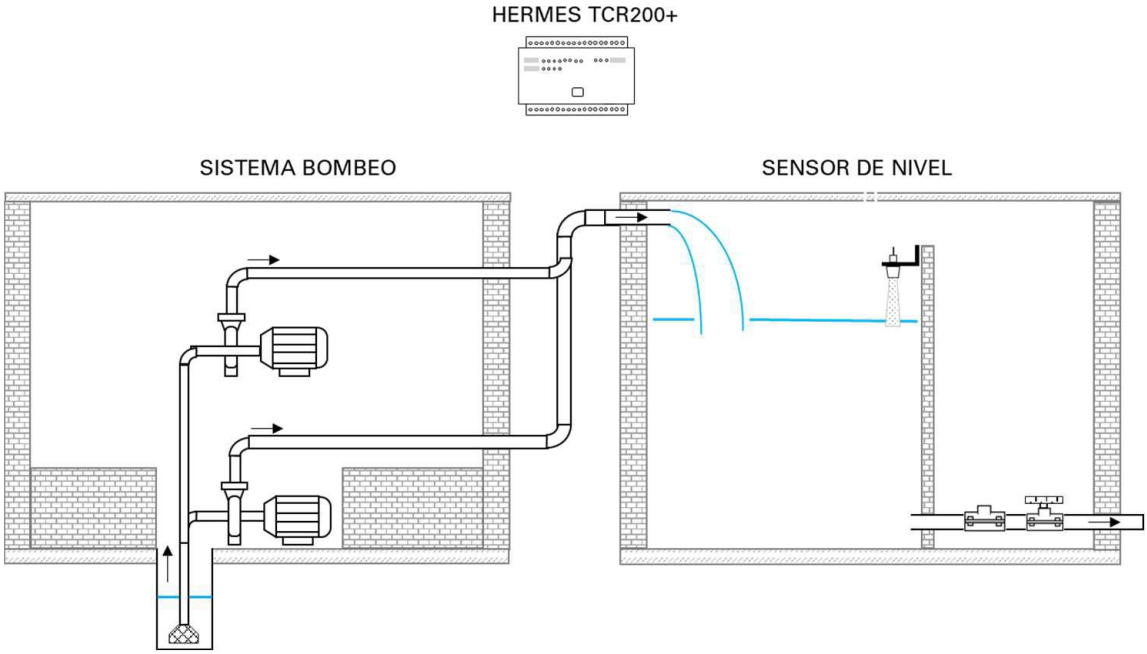
17.3 AUTOMATISMO DE ARRANQUE DE BOMBAS

Ejemplo de arranque automático de una estación de bombeo.

Este ejemplo ilustra una estrategia avanzada de gestión de bombeos que combina múltiples condiciones operando de forma simultánea. El sistema permite la conmutación entre control manual y automático, incorpora alternancia entre bombas para equilibrar el desgaste, y ajusta dinámicamente las consignas de nivel según la franja tarifaria eléctrica. Durante las horas valle, cuando la energía es más económica, el depósito se llena hasta un nivel mínimo de 4 metros (80% de su capacidad). En horas llanas, con una tarifa intermedia, el objetivo se reduce a 2 metros (40%), optimizando el consumo al aprovechar el agua acumulada previamente. En horas punta, la tarifa más costosa, se permite que el nivel descienda hasta 0.5 metros (10%), maximizando el ahorro energético al evitar bombeos durante los periodos de mayor coste.

El sistema del ejemplo se compone de los siguientes elementos:

- Microcom Hermes TCR200+: Equipo de telecontrol y autómata al que se conecta el resto de los elementos: sensor de nivel de agua, protecciones de los bombeos y órdenes de marcha de los bombeos.
- Sonda de nivel: Sonda analógica para medir el nivel del agua del depósito a mantener con agua.
- Sistema de bombeo: Este sistema está compuesto por 2 bombas y se tiene un acceso independiente al control de arranque y del estado de cada bomba.



Esta configuración utiliza 2 páginas de las 8 que tienen disponibles los equipos Hermes.

<u>Página 0:</u>

```
L1  #INIT
L2  DO0 : MARCHA_B0
L3  DO1 : MARCHA_B1
L4  DI0 : FALLO_B0
L5  DI1 : FALLO_B1
L6  DI2 : CONF_B0
L7  DI3 : CONF_B1
L8  M8A : ACU_B0
L9  M8B : ACU_B1
L10 M10 : MODO; REM MODO=0-> PARO, 1-> MANUAL, 2 -> AUTO
L11 AI0 : NIVEL
L12 REAL : NBAJO
L13 REAL : NALTO
L14 BOOL : FALTA_AGUA
L15 INT : PARO=0
L16 INT : MANU=1
L17 INT : AUTO=2
L18 REM DOS BOMBAS. FUNCIONAMIENTO CICLICO
L19 PUMP.NUM=2, DON=5, DOFF=5, PRTO=10, AC0=ACU_B0, AC1=ACU_B1
L20 REM TARIFA PUNTA MAÑANA (LUN-VIE 10:00-14:00)
L21 TW : PUNTAM, WEEK = 12345, ON = 1000, OFF = 1400
L22 REM TARIFA PUNTA TARDE (LUN-VIE 18:00-22:00)
L23 TW : PUNTAT,WEEK = 12345, ON = 1800, OFF = 2200
L24 REM TARIFA LLANO MAÑANA (LUN-VIE 08:00-10:00)
L25 TW : LLANOM,WEEK = 12345, ON = 0800, OFF = 1000
L26 REM TARIFA LLANO TARDE (LUN-VIE 14:00-18:00)
L27 TW : LLANOT,WEEK = 12345, ON = 1400, OFF = 1800
L28 #END_INIT
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias “MARCHA_B0” a la salida digital 0 “DO0”. Salida conectada al sistema de arranque de la bomba 0.
L3	Asigna el alias “MARCHA_B1” a la salida digital 1 “DO1”. Salida conectada al sistema de arranque de la bomba 1.
L4	Asigna el alias “FALLO_B0” a la entrada digital 0 “DI0”. Entrada conectada al indicador de fallo de la bomba 0.
L5	Asigna el alias “FALLO_B1” a la entrada digital 1 “DI1”. Entrada conectada al indicador de fallo de la bomba 1.
L6	Asigna el alias “CONF_B0” a la entrada digital 2 “DI2”. Entrada conectada a la señal de confirmación de marcha de la bomba 0.

Línea	Descripción
L7	Asigna el alias "CONF_B1" a la entrada digital 3 "DI3". Entrada conectada a la señal de confirmación de marcha de la bomba 1.
L8	Asigna el alias "ACU_B0" al registro matemático 8 (32 bits menor peso) "M8A". Almacena el tiempo de encendido de la bomba 0.
L9	Asigna el alias "ACU_B1" al registro matemático 8 (32 bits mayor peso) "M8B". Almacena el tiempo de encendido de la bomba 1.
L10	Asigna el alias "MODO" al registro matemático 10 "M10". Esta variable indica el modo de funcionamiento del sistema.
L11	Asigna el alias "NIVEL" a la entrada analógica 0 "AI0". Entrada conectada al sensor de nivel del depósito.
L12	Declarar "NBAJO" como una variable de tipo REAL. Consigna nivel bajo o de activación.
L13	Declarar "NALTO" como una variable de tipo REAL. Consigna nivel alto o de desactivación.
L14	Declarar "FALTA_AGUA" como una variable de tipo BOOL. Indica petición de agua al bombeo.
L15	Declarar "PARO" como una variable de tipo INT e inicializar a valor a 0. Es un valor constante.
L16	Declarar "MANU" como una variable de tipo INT e inicializar a valor a 1. Es un valor constante.
L17	Declarar "AUTO" como una variable de tipo INT e inicializar a valor a 2. Es un valor constante.
L18	Comentario para describir el programa. Esta línea no se ejecuta.
L19	Configurar el módulo de control de bombas: bombas a controlar 2 (PUMP.NUM=2), retardo de encendido 5 segundos (DON=5), retardo de apagado 5 segundos (DOFF=5), tiempo de confirmación de marcha 10 segundos (PRTO=10), restaurar el valor del tiempo de encendido de la bomba 0 desde "ACU_B0" (AC0=ACU_B0) y restaurar el valor del tiempo de encendido de la bomba 1 desde "ACU_B1" (AC1=ACU_B1).
L20	Comentario para describir el programa. Esta línea no se ejecuta.
L21	Declarar "PUNTAM" como un temporizador semanal. Configurarlo para que esté habilitado de lunes a viernes (WEEK=12345), se active a las 10:00 (ON=1000) y se desactive a las 14:00 (OFF=1400).
L22	Comentario para describir el programa. Esta línea no se ejecuta
L23	Declarar "PUNTAT" como un temporizador semanal. Configurarlo para que esté habilitado de lunes a viernes (WEEK=12345), se active a las 18:00 (ON=1800) y se desactive a las 22:00 (OFF=2200).
L24	Comentario para describir el programa. Esta línea no se ejecuta.
L25	Declarar "LLANOM" como un temporizador semanal. Configurarlo para que esté habilitado de lunes a viernes (WEEK=12345), se active a las 8:00 (ON=0800) y se desactive a las 10:00 (OFF=1000).
L26	Comentario para describir el programa. Esta línea no se ejecuta
L27	Declarar "LLANOT" como un temporizador semanal. Configurarlo para que esté habilitado de lunes a viernes (WEEK=12345), se active a las 14:00 (ON=1400) y se desactive a las 18:00 (OFF=1800).
L28	Palabra clave para finalizar la sección de inicialización.

Página 1

```
L1  REM PARAMETROS LLENADO SEGUN TARIFA
L2  REM VALLE
L3  NBAJO=4 ; NALTO=5
L4  REM LLANO
L5  IF LLANOM.Q OR LLANOT.Q ; NBAJO=2 ; NALTO=3
L6  REM PUNTA
L7  IF PUNTAM.Q OR PUNTAT.Q ; NBAJO=0.5 ; NALTO=1
L8  REM AUTOMATISMO LLENADO
L9  IF MODO<>AUTO AND MODO<>MANU AND MODO<>PARO ; MODO=PARO
L10 IF NIVEL<NBAJO ; FALTA_AGUA=TRUE
L11 IF NIVEL>NALTO ; FALTA_AGUA=FALSE
L12 IF MODO=AUTO ; PUMP.REQ=FALTA_AGUA
L13 IF MODO=MANU ; PUMP.REQ=1
L14 IF MODO=PARO ; PUMP.REQ=0
L15 REM E/S MODULO PUMP
L16 MARCHA_B0=PUMP.Q0
L17 MARCHA_B1=PUMP.Q1
L18 PUMP.DIS0=FALLO_B0
L19 PUMP.DIS1=FALLO_B1
L20 PUMP.PRC0=CONF_B0
L21 PUMP.PRC1=CONF_B1
L22 ACU_B0=PUMP.AC0
L23 ACU_B1=PUMP.AC1
```

Línea	Descripción
L1	Comentario para describir el programa. Esta línea no se ejecuta.
L2	Comentario para describir el programa. Esta línea no se ejecuta.
L3	Iguala “NBAJO” a 4 y “NALTO” a 5. El valor de las consignas de arranque y paro en franja horaria valle, la más barata, se configuran para que el depósito se mantenga con una altura de agua superior a 4 metros e inferior a 5 metros.
L4	Comentario para describir el programa. Esta línea no se ejecuta.
L5	Si los temporizadores “LLANOM” o “LLANOT” están activados, Igualar “NBAJO” a 2 y “NALTO” a 3. El valor de las consignas de arranque y paro en franja horaria llana, precio intermedio, se configuran para que el depósito se mantenga con una altura de agua superior a 2 metros e inferior a 3 metros.
L6	Comentario para describir el programa. Esta línea no se ejecuta.
L7	Si los temporizadores “PUNTAM” o “PUNTAT” están activados, Igualar “NBAJO” a 0.5 y “NALTO” a 1. El valor de las consignas de arranque y paro en franja horaria punta, precio más caro, se configuran para que el depósito se mantenga con una altura de agua superior a 0.5 metros e inferior a 1 metro.
L8	Comentario para describir el programa. Esta línea no se ejecuta.

Línea	Descripción
L9	Si “MODO” es diferente a “AUTO”, “MANU” o “PARO”, “MODO” se iguala a “PARO”. Los modos de funcionamiento indican si opera automáticamente por niveles de consigna o manualmente por órdenes en remoto. Esta línea es un control de seguridad, en caso de tener en la variable MODO un valor diferente a 0, 1 o 2, se considera que hay un fallo y se fuerza el sistema para entrar en modo PARO.
L10	Si el nivel del depósito (“NIVEL”) está por debajo de la consigna de nivel mínimo o de arranque (“NBAJO”), se activa el indicador de falta de agua (“FALTA_AGUA” se iguala a 1 (TRUE)). El estado de este indicador se utiliza posteriormente como condición en la orden a de arranque de las bombas en modo automático (ver línea L12).
L11	Si el nivel del depósito (“NIVEL”) supera la consigna de nivel máximo o de paro (“NALTO”), se desactiva el indicador de falta de agua (“FALTA_AGUA” se iguala a 0 (FALSE)). El estado de este indicador se utiliza posteriormente como condición en la orden a de arranque de las bombas en modo automático (ver línea L12).
L12	Si el modo de funcionamiento (“MODO”) es igual a automático (“AUTO”), la propiedad del módulo que de control de bombas que controla el número de bombas activas (“PUMP.REQ”) se iguala con el valor del indicador “FALTA_AGUA”, de este modo, si el indicador “FALTA_AGUA” está activado, se dará la orden de activar una bomba. Si el indicador “FALTA_AGUA” está desactivado, se dará la orden de no activar ninguna bomba.
L13	Si “MODO” es igual a manual (“MANU”), “PUMP.REQ” tomará el valor 1. Configurar el módulo de control de bombas para que funcione 1 bomba. Mientras esté en este modo una de las bombas permanecerá encendida.
L14	Si “MODO” es igual a paro (“PARO”), “PUMP.REQ” tomará el valor 0. Configurar el módulo de control de bombas para que ninguna bomba funcione, por tanto, el sistema de bombeo permanecerá apagado.
L15	Comentario para describir el programa. Esta línea no se ejecuta.
L16	“MARCHA_B0” se iguala con el valor “PUMP.Q0”. La orden de arranque de la bomba 0 del bloque de control de bombas, controla la salida digital que pone en marcha la bomba 0.
L17	“MARCHA_B1” se iguala con el valor “PUMP.Q1”. La orden de arranque de la bomba 1 del bloque de control de bombas, controla la salida digital que pone en marcha la bomba 1.
L18	“PUMP.DIS0” se iguala con el valor de “FALLO_B0”. Controla la inhabilitación de la bomba 0. Mientras la entrada digital 0 esté activa, la bomba 0 permanecerá inhabilitada, por tanto, el automatismo no intentará ponerla en marcha.
L19	“PUMP.DIS1” se iguala con el valor de “FALLO_B1”. Controla la inhabilitación de la bomba 1. Mientras la entrada digital 1 esté activa, la bomba 1 permanecerá inhabilitada, por tanto, el automatismo no intentará ponerla en marcha.
L20	“PUMP.PRC0” se iguala con el valor de “CONF_B0”. Esto informa al módulo PUMP de la confirmación de marcha de la bomba 0.
L21	“PUMP.PRC1” se iguala con el valor de “CONF_B1”. Esto informa al módulo PUMP de la confirmación de marcha de la bomba 1.
L22	“ACU_B0” se iguala con el valor de “PUMP.AC0”. Guarda en una memoria no volátil el tiempo de encendido o cuenta horas. De este modo, en caso de corte en el suministro eléctrico este valor no se perderá.

Línea	Descripción
L23	“ACU_B1” se iguala con el valor de “PUMP.AC1”. Guarda en una memoria no volátil el tiempo de encendido o cuenta horas. De este modo, en caso de corte en el suministro eléctrico, este valor no se perderá.

17.4 CONTROL DE BOMBAS EN MODO PRINCIPAL/APOYO

Ejemplo de control de bombas de una estación de bombeo de aguas residuales.

Este ejemplo describe un automatismo de control para estaciones de bombeo de aguas residuales basada en un esquema de operación de bombas en modo principal/apoyo. El sistema está compuesto por cuatro bombas, con acceso independiente tanto a su estado como a su control de arranque. La configuración permite definir qué bombas actuarán como principales y cuales como apoyo, así como establecer una prioridad de uso para cada una. El nivel del agua en el depósito se mide mediante una sonda analógica, lo que permite ajustar dinámicamente la operación del sistema según la demanda. A diferencia del ejemplo anterior, este escenario no contempla discriminación horaria, por lo que el criterio de activación se basa exclusivamente en el nivel de agua y la lógica de reparto de carga entre bombas.

El sistema del ejemplo se compone de los siguientes elementos:

- Microcom Hermes M103 Equipo de telecontrol y autómata al que se conecta el resto de los elementos: sensor de nivel de agua, protecciones de los bombeos y órdenes de marcha de los bombeos.
- Sonda de nivel: Sonda analógica para medir el nivel del agua del depósito a mantener con agua.
- Sistema de bombeo: Este sistema está compuesto por 4 bombas y se tiene un acceso independiente al control de arranque y del estado de cada bomba.

<u>Página 0:</u>

```
L1   #INIT
L2    DI0 : FALLO_B0
L3    DI1 : FALLO_B1
L4    DI2 : FALLO_B2
L5    DI3 : FALLO_B3
L6    DI4 : MARCHA_B1
L7    DI5 : MARCHA_B2
L8    DI6 : MARCHA_B3
L9    DI7 : MARCHA_B4
L10   EXP16 : RELE_B0
L11   EXP17 : RELE_B1
L12   EXP18 : RELE_B2
L13   EXP19 : RELE_B3
L14   AI0 : NIVEL
L15   M0 : NALTO
L16   M1 : NBAJO
L17   M2 : NAPOYO
L18   M3 : PRIO_B0
L19   M4 : PRIO_B1
L20   M5 : PRIO_B2
L21   M6 : PRIO_B3
L22   M10 : MODO; REM 0–>PARO, 1–>MARCHA, 2–>AUTO
L23   F0 : PRIN_B0
```

```
L24   F1 : PRIN_B1
L25   F2 : PRIN_B2
L26   F3 : PRIN_B3
L27   F4 : B0MAN
L28   F5 : B1MAN
L29   F6 : B2MAN
L30   F7 : B3MAN
L31   F8 : DIS_MAN_B0
L32   F9 : DIS_MAN_B1
L33   F10 : DIS_MAN_B2
L34   F11 : DIS_MAN_B3
L35   F12: RESET_FB0
L36   F13: RESET_FB1
L37   F14: RESET_FB2
L38   F15: RESET_FB3
L39   BOOL : MARCHA
L40   BOOL : APOYO
L41   INT : PARO=0
L42   INT : MANU=1
L43   INT : AUTO=2
L44   PUMP.NUM=6,DON=5,DOFF=5,PRTO=10
L45  #END_INIT
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias “FALLO_B0” a la entrada digital 0 “DI0”. Entrada conectada al indicador de fallo de la bomba 0.
L3	Asigna el alias “FALLO_B1” a la entrada digital 1 “DI1”. Entrada conectada al indicador de fallo de la bomba 1.
L4	Asigna el alias “FALLO_B2” a la entrada digital 2 “DI2”. Entrada conectada al indicador de fallo de la bomba 2.
L5	Asigna el alias “FALLO_B3” a la entrada digital 3 “DI3”. Entrada conectada al indicador de fallo de la bomba 3.
L6	Asigna el alias “MARCHA_B0” a la entrada digital 4 “DI4”. Entrada conectada a la señal de confirmación de marcha de la bomba 0.
L7	Asigna el alias “MARCHA_B1” a la entrada digital 5 “DI5”. Entrada conectada a la señal de confirmación de marcha de la bomba 1.
L8	Asigna el alias “MARCHA_B2” a la entrada digital 6 “DI6”. Entrada conectada a la señal de confirmación de marcha de la bomba 2
L9	Asigna el alias “MARCHA_B3” a la entrada digital 7 “DI7”. Entrada conectada a la señal de confirmación de marcha de la bomba 3
L10	Asigna el alias “RELE_B0” a la expansión 16 “EXP16”. Salida conectada al sistema de arranque de la bomba 0.

Línea	Descripción
L11	Asigna el alias “RELE_B1” a la expansión 17 “EXP17”. Salida conectada al sistema de arranque de la bomba 1.
L12	Asigna el alias “RELE_B2” a la expansión 18 “EXP18”. Salida conectada al sistema de arranque de la bomba 2.
L13	Asigna el alias “RELE_B3” a la expansión 19 “EXP19”. Salida conectada al sistema de arranque de la bomba 3.
L14	Asigna el alias “NIVEL” a la entrada analógica 0 “AI0”. Entrada conectada al sensor de nivel del depósito.
L15	Asigna el alias “NBAJO” al registro matemático 0 “M0”. Consigna nivel bajo o de activación.
L16	Asigna el alias “NALTO” al registro matemático 1 “M1”. Consigna nivel alto o de desactivación.
L17	Asigna el alias “NAPOYO” al registro matemático 2 “M2”. Consigna de activación/desactivación de bomba de apoyo.
L18	Asigna el alias “PRIO_B0” al registro matemático 3 “M3”. Prioridad de la bomba 0.
L19	Asigna el alias “PRIO_B1” al registro matemático 4 “M4”. Prioridad de la bomba 1.
L20	Asigna el alias “PRIO_B2” al registro matemático 5 “M5”. Prioridad de la bomba 2.
L21	Asigna el alias “PRIO_B3” al registro matemático 6 “M6”. Prioridad de la bomba 3.
L22	Asigna el alias “MODO” al registro matemático 10 “M10”. Esta variable indica el modo de funcionamiento del sistema.
L23	Asigna el alias “PRIN_B0” al flag 0 “F0”. Configura la bomba 0 como prioritaria o de apoyo.
L24	Asigna el alias “PRIN_B1” al flag 1 “F1”. Configura la bomba 1 como prioritaria o de apoyo.
L25	Asigna el alias “PRIN_B2” al flag 2 “F2”. Configura la bomba 2 como prioritaria o de apoyo.
L26	Asigna el alias “PRIN_B3” al flag 3 “F3”. Configura la bomba 3 como prioritaria o de apoyo.
L27	Asigna el alias “B0MAN” al flag 4 “F4”. Necesaria para activar la bomba 0 en el modo manual.
L28	Asigna el alias “B1MAN” al flag 5 “F5”. Necesaria para activar la bomba 1 en el modo manual.
L29	Asigna el alias “B2MAN” al flag 6 “F6”. Necesaria para activar la bomba 2 en el modo manual.
L30	Asigna el alias “B3MAN” al flag 7 “F7”. Necesaria para activar la bomba 3 en el modo manual.
L31	Asigna el alias “DIS_MAN_B0” al flag 8 “F8”. Inhibición manual de la bomba 0.
L32	Asigna el alias “DIS_MAN_B1” al flag 9 “F9”. Inhibición manual de la bomba 1.
L33	Asigna el alias “DIS_MAN_B2” al flag 10 “F10”. Inhibición manual de la bomba 2.
L34	Asigna el alias “DIS_MAN_B3” al flag 11 “F11”. Inhibición manual de la bomba 3.
L35	Asigna el alias “RESET_FB0” al flag 12 “F12”. Reset del fallo de arranque de la bomba 0.
L36	Asigna el alias “RESET_FB1” al flag 13 “F13”. Reset del fallo de arranque de la bomba 1.

Línea	Descripción
L37	Asigna el alias “RESET_FB2” al flag 14 “F14”. Reset del fallo de arranque de la bomba 2.
L38	Asigna el alias “RESET_FB3” al flag 15 “F15”. Reset del fallo de arranque de la bomba 3.
L39	Declarar “PIDEAGUA” como una variable de tipo BOOL. Se activa cuando hay petición de agua.
L40	Declarar “PIDEAPOYO” como una variable de tipo BOOL. Se activa cuando hay petición bomba de apoyo.
L41	Declarar “PARO” como una variable de tipo INT e inicializar a valor a 0. Es un valor constante.
L42	Declarar “MANU” como una variable de tipo INT e inicializar a valor a 1. Es un valor constante.
L43	Declarar “AUTO” como una variable de tipo INT e inicializar a valor a 2. Es un valor constante.
L44	Configura el módulo de control de bombas: bombas a controlar 4 (PUMP.NUM=4), retardo de encendido 5 segundos (DON=5), retardo de apagado 5 segundos (DOFF=5), tiempo de confirmación de marcha 10 segundos (PRTO=10)
L45	Palabra clave para finalizar la sección de inicialización.

Página 1:

```
L1   REM DEFINO PRINCIPALES/APOYO
L2   PUMP.LEAD0=PRIN_B0
L3   PUMP.LEAD1=PRIN_B1
L4   PUMP.LEAD2=PRIN_B2
L5   PUMP.LEAD3=PRIN_B3
L6   REM DEFINO PRIORIDADES
L7   PUMP.PRIO0=PRIO_B0
L8   PUMP.PRIO1=PRIO_B1
L9   PUMP.PRIO2=PRIO_B2
L10  PUMP.PRIO3=PRIO_B3
L11  REM RESET FALLOS DE ARRANQUE
L12  IF RESET_FB0; PRF0=0; RESET_FB0=0
L13  IF RESET_FB1; PRF1=0; RESET_FB1=0
L14  IF RESET_FB2; PRF2=0; RESET_FB2=0
L15  IF RESET_FB3; PRF3=0; RESET_FB3=0
L16  REM PETICIÓN DE AGUA
L17  IF NIVEL>NALTO ; MARCHA=TRUE
L18  IF NIVEL<NBAJO ; MARCHA=FALSE
L19  IF NIVEL>NAPOYO; APOYO=TRUE
L20  IF NIVEL<NALTO; APOYO=FALSE
L21  REM MODO DE FUNCIONAMIENTO
L22  IF MODO<>PARO AND MODO<>MANU AND MODO<>AUTO ; MODO=PARO
L23  PUMP.HAND=(MODO=MANU) ; PUMP.AUTO=(MODO=AUTO) ; PUMP.OFF=(MODO=PARO)
L24  REM ARRANQUE/PARO BOMBAS
L25  PUMP.REQ=MARCHA + APOYO
L26  REM E/S MODULO PUMP
L27  RELE_B0=PUMP.Q0
L28  RELE_B1=PUMP.Q1
L29  RELE_B2=PUMP.Q2
L30  RELE_B3=PUMP.Q3
L31  PUMP.MAN0=B0MAN
L32  PUMP.MAN1=B1MAN
L33  PUMP.MAN2=B2MAN
L34  PUMP.MAN3=B3MAN
L35  PUMP.PRC0=MARCHA_B0
L36  PUMP.PRC1=MARCHA_B1
L37  PUMP.PRC2=MARCHA_B2
L38  PUMP.PRC3=MARCHA_B3
L39  PUMP.DIS0=FALLO_B0 OR DIS_MAN_B0
L40  PUMP.DIS1=FALLO_B1 OR DIS_MAN_B1
L41  PUMP.DIS2=FALLO_B2 OR DIS_MAN_B2
L42  PUMP.DIS3=FALLO_B3 OR DIS_MAN_B3
```

Línea	Descripción
L1	Comentario para describir el programa. Esta línea no se ejecuta.
L2	Asigna el valor de "PRIN_B0" a la propiedad PUMP.LEAD0 del módulo de control de bombas. Esta propiedad determina si la bomba 0 es principal o de apoyo.

Línea	Descripción
L3	Asigna el valor de "PRIN_B1" a la propiedad PUMP.LEAD1 del módulo de control de bombas. Esta propiedad determina si la bomba 1 es principal o de apoyo.
L4	Asigna el valor de "PRIN_B2" a la propiedad PUMP.LEAD2 del módulo de control de bombas. Esta propiedad determina si la bomba 2 es principal o de apoyo.
L5	Asigna el valor de "PRIN_B3" a la propiedad PUMP.LEAD3 del módulo de control de bombas. Esta propiedad determina si la bomba 3 es principal o de apoyo.
L6	Comentario para describir el programa. Esta línea no se ejecuta.
L7	Asigna el valor de "PRIO_B0" a la propiedad PUMP.PRIO0 del módulo de control de bombas. Esta propiedad determina la prioridad de la bomba 0.
L8	Asigna el valor de "PRIO_B1" a la propiedad PUMP.PRIO1 del módulo de control de bombas. Esta propiedad determina la prioridad de la bomba 1.
L9	Asigna el valor de "PRIO_B2" a la propiedad PUMP.PRIO2 del módulo de control de bombas. Esta propiedad determina la prioridad de la bomba 2.
L10	Asigna el valor de "PRIO_B3" a la propiedad PUMP.PRIO3 del módulo de control de bombas. Esta propiedad determina la prioridad de la bomba 3.
L11	Comentario para describir el programa. Esta línea no se ejecuta.
L12	Si se activa al orden de reset del fallo arranque de la bomba 0 "RESET_FB0" se desactiva el fallo y se desactiva de nuevo la orden para permitir activar el fallo de nuevo.
L13	Si se activa al orden de reset del fallo arranque de la bomba 1 "RESET_FB1" se desactiva el fallo y se desactiva de nuevo la orden para permitir activar el fallo de nuevo.
L14	Si se activa al orden de reset del fallo arranque de la bomba 2 "RESET_FB2" se desactiva el fallo y se desactiva de nuevo la orden para permitir activar el fallo de nuevo.
L15	Si se activa al orden de reset del fallo arranque de la bomba 3 "RESET_FB3" se desactiva el fallo y se desactiva de nuevo la orden para permitir activar el fallo de nuevo.
L11	Comentario para describir el programa. Esta línea no se ejecuta.
L12	Si el nivel del depósito ("NIVEL") está por encima de la consigna de nivel máximo o de arranque ("NALTO"), se activa el indicador de marcha ("MARCHA" se iguala a 1 (TRUE)). El estado de este indicador se utiliza posteriormente como condición en la orden a de arranque de las bombas en modo automático (ver línea L20).
L13	Si el nivel del depósito ("NIVEL") está por debajo de la consigna de nivel mínimo o de paro ("NBAJO"), se desactiva el indicador de marcha ("MARCHA" se iguala a 0 (FLASE)). El estado de este indicador se utiliza posteriormente como condición en la orden a de arranque de las bombas en modo automático (ver línea L20).
L14	Si el nivel del depósito ("NIVEL") está por encima de la consigna de nivel de apoyo ("NAPOYO"), se activa el indicador de apoyo ("APOYO" se iguala a 1 (TRUE)). El estado de este indicador se utiliza posteriormente como condición en la orden a de arranque de las bombas en modo automático (ver línea L20).
L15	Si el nivel del depósito ("NIVEL") está por debajo de la consigna de nivel alto ("NALTO"), se desactiva el indicador de apoyo ("APOYO" se iguala a 0 (FLASE)). El estado de este indicador se utiliza

Línea	Descripción
	posteriormente como condición en la orden a de arranque de las bombas en modo automático (ver línea L20).
L16	Comentario para describir el programa. Esta línea no se ejecuta.
L17	Si "MODO" es diferente a "AUTO", "MANU" o "PARO", "MODO" se iguala a "PARO". Los modos de funcionamiento indican si opera automáticamente por niveles de consigna o manualmente por órdenes en remoto. Esta línea es un control de seguridad, en caso de tener en la variable MODO un valor diferente a 0, 1 o 2, se considera que hay un fallo y se fuerza el sistema para entrar en modo PARO.
L18	Se activa la propiedad PUMP.HAND, PUMP.AUTO o PUMP.OFF dependiendo del valor que tenga el registro "MODO"
L19	Comentario para describir el programa. Esta línea no se ejecuta.
L20	La propiedad del módulo que de control de bombas que controla el número de bombas activas ("PUMP.REQ") se iguala a la suma de "PIDEAGUA" y "PIDEAPOYO", indicando cuantas bombas tienen que estar arrancadas.
L21	Comentario para describir el programa. Esta línea no se ejecuta.
L22	"RELE_B0" se iguala con el valor "PUMP.Q0". La orden de arranque de la bomba 0 del bloque de control de bombas, controla la salida digital que pone en marcha la bomba 0.
L23	"RELE_B1" se iguala con el valor "PUMP.Q1". La orden de arranque de la bomba 1 del bloque de control de bombas, controla la salida digital que pone en marcha la bomba 1.
L24	"RELE_B2" se iguala con el valor "PUMP.Q2". La orden de arranque de la bomba 2 del bloque de control de bombas, controla la salida digital que pone en marcha la bomba 2.
L25	"RELE_B3" se iguala con el valor "PUMP.Q3". La orden de arranque de la bomba 3 del bloque de control de bombas, controla la salida digital que pone en marcha la bomba 3.
L26	"PUMP.MAN0" se iguala con el valor "B0MAN". La orden de arranque de la bomba 0 del bloque de control de bombas, cuando está activado el modo manual.
L27	"PUMP.MAN1" se iguala con el valor "B1MAN". La orden de arranque de la bomba 1 del bloque de control de bombas, cuando está activado el modo manual.
L28	"PUMP.MAN2" se iguala con el valor "B2MAN". La orden de arranque de la bomba 2 del bloque de control de bombas, cuando está activado el modo manual.
L29	"PUMP.MAN3" se iguala con el valor "B3MAN". La orden de arranque de la bomba 3 del bloque de control de bombas, cuando está activado el modo manual.
L30	"PUMP.PRC0" se iguala con el valor "MARCHA_B0". La confirmación de marcha de la bomba 0 del bloque de control de bombas.
L31	"PUMP.PRC1" se iguala con el valor "MARCHA_B1". La confirmación de marcha de la bomba 1 del bloque de control de bombas.
L32	"PUMP.PRC2" se iguala con el valor "MARCHA_B2". La confirmación de marcha de la bomba 2 del bloque de control de bombas.

Línea	Descripción
L33	“PUMP.PRC3” se iguala con el valor “MARCHA_B3”. La confirmación de marcha de la bomba 3 del bloque de control de bombas.
L34	Si se activa el indicador de inhabilitación automática de la bomba 0 “FALLO_B0” o de inhabilitación manual “FALLOM_B0, se activa la entrada del módulo de control correspondiente “PUMP.DIS0”, para inhibir la bomba.
L35	Si se activa el indicador de inhabilitación automática de la bomba 1 “FALLO_B1” o de inhabilitación manual “FALLOM_B1, se activa la entrada del módulo de control correspondiente “PUMP.DIS1”, para inhibir la bomba.
L36	Si se activa el indicador de inhabilitación automática de la bomba 2 “FALLO_B2” o de inhabilitación manual “FALLOM_B2, se activa la entrada del módulo de control correspondiente “PUMP.DIS2”, para inhibir la bomba.
L37	Si se activa el indicador de inhabilitación automática de la bomba 3 “FALLO_B3” o de inhabilitación manual “FALLOM_B3, se activa la entrada del módulo de control correspondiente “PUMP.DIS3”, para inhibir la bomba.

17.5 CONTROL CAUDAL / PRESIÓN POR PID

Control de caudal/presión de una estación de bombeo de agua potable

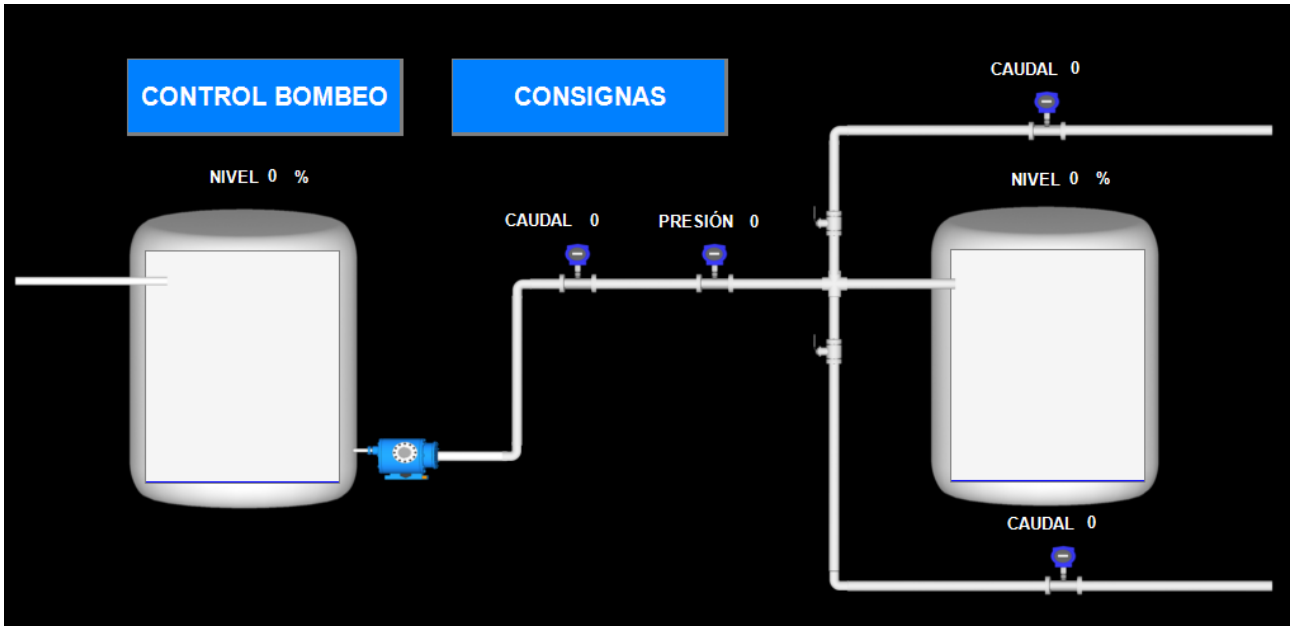
Este ejemplo muestra la aplicación del módulo PID en un grupo de presión donde se requiere mantener constante la presión de salida, sin exceder un caudal máximo predefinido.

Para ello se implementan dos controladores PID que operan en paralelo: uno dedicado a la regulación de la presión y otro a la limitación del caudal. El sistema evalúa dinámicamente cuál de los dos lazos debe tomar el control en función de las condiciones de operación:

- Cuando el caudal se mantiene dentro del umbral permitido, el PID de presión regula con normalidad.
- Si el caudal alcance el límite establecido, se cede el control al PID de caudal, que actúa reduciendo indirectamente la presión para restringir el caudal.

El controlador que no tiene el control activo permanece en modo tracking, siguiendo la salida del PID dominante para evitar transitorios al retomar el control.

Esta estrategia permite una regulación precisa de la presión sin comprometer los límites hidráulicos del sistema.



Página 0:

```
L1   #INIT
L2   AI0 : PRESION
L3   AI1 : CAUDAL
L4   M0 : OBJETIVO_PRESION
L5   M1 : OBJETIVO_CAUDAL
L6   M2 : MIN_ARRANQUE
L7   M3 : MODO; REM 0 –> PARO  2 –> AUTO
L8   BOOL : PRES_CAU; REM 0 –> DOMINIA PRESION  1 –> DOMINA CAUDAL
L9   INT : PARO=0
L10  INT : AUTO=2
L11  REAL : CONSIGNA
L12  EXP16 : BOMBA
L13  PID : CONTROL_P,KP=2,TI=10,TD=0,RAMP=0.1
L14  PID : CONTROL_C,KP=2,TI=10,TD=0,RAMP=0.1
L15  #END_INIT
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Asigna el alias "PRESION" a la entrada analógica 0 "AI0". Entrada conectada al transductor de presión.
L3	Asigna el alias "CAUDAL" a la entrada analógica 1 "AI1". Entrada conectada a la salida analógica del caudalímetro.
L4	Asigna el alias "OBJETIVO_PRESION" al registro matemático 0 "M0". Registro de memoria en el que se guarda el objetivo de presión.
L5	Asigna el alias "OBJETIVO_CAUDAL" al registro matemático 1 "M1". Registro de memoria en el que se guarda el objetivo de caudal.
L6	Asigna el alias "MIN_ARRANQUE" al registro matemático 2 "M2". Registro de memoria en el que se guarda el valor mínimo de consigna para el arranque de la bomba.
L7	Asigna el alias "MODO" al registro matemático 3 "M3". Esta variable indica el modo de funcionamiento del sistema.
L8	Declarar "PRES_CAU" como una variable de tipo BOOL. Indica si está predominando la regulación por caudal o por presión.
L9	Declarar "PARO" como una variable de tipo INT e inicializar a valor a 0. Es un valor constante.
L10	Declarar "AUTO" como una variable de tipo INT e inicializar a valor a 2. Es un valor constante.
L11	Declarar "CONSIGNA" como una variable de tipo REAL.
L12	Asigna el alias "BOMBA" A la salida analógica configurada en la expansión 16 "EXP16"
L13	Asigna el alias "CONTROL_P" a un módulo de PID con una ganancia proporcional de 2, un tiempo integral de 10, un tiempo derivativo de 0, una rampa de 0.1 y una salida máxima de 200.

Línea	Descripción
L14	Asigna el alias "CONTROL_P" a un módulo de PID con una ganancia proporcional de 2, un tiempo integral de 10, un tiempo derivativo de 0, una rampa de 0.1 y una salida máxima de 200.
L15	Palabra clave para finalizar la sección de inicialización.

Página 1:

```
L1   REM PID PRESION
L2   CONTROL_P.SP=OBJETIVO_PRESION
L3   CONTROL_P.PV=PRESION
L4   REM PID CAUDAL
L5   CONTROL_C.SP=OBJETIVO_CAUDAL
L6   CONTROL_C.PV=CAUDAL
L7   REM MARCHA/PARO POR PID
L8   IF MODO<>PARO AND MODO<>AUTO ; MODO=PARO
L9   IF MODO=PARO; BOMBA=0
L10  IF MODO=AUTO AND CONSIGNA>MIN_ARRANQUE; BOMBA=CONSIGNA
L11  REM PID DOMINANTE
L12  IF CAUDAL>=OBJETIVO_CAUDAL;PRES_CAU=1
L13  IF CAUDAL<(OBJETIVO_CAUDAL-HISTERESIS);PRES_CAU=0
L14  IF PRES_CAU ; CONTROL_P.TRK=1 ; CONTROL_P.TV=CONTROL_C.OUT ; CONTROL_C.TRK=0 ;
     CONSIGNA=CONTROL_C.OUT
L15  IF PRES_CAU=0 ; CONTROL_C.TRK=1 ; CONTROL_C.TV=CONTROL_P.OUT ; CONTROL_P.TRK=0 ;
     CONSIGNA=CONTROL_P.OUT
```

Línea	Descripción
L1	Comentario para describir el programa. Esta línea no se ejecuta.
L2	Asigna el valor de "OBJETIVO_PRESION" al set point del PID "CONTROL_P
L3	Asigna el valor de "PRESION" a la variable de proceso del PID "CONTROL_P
L4	Comentario para describir el programa. Esta línea no se ejecuta.
L5	Asigna el valor de "OBJETIVO_CAUDAL" al set point del PID "CONTROL_C
L6	Asigna el valor de "CAUDAL" a la variable de proceso del PID "CONTROL_C
L7	Comentario para describir el programa. Esta línea no se ejecuta.
L8	Si "MODO" es diferente a "AUTO" o "PARO", "MODO" se iguala a "PARO". Esta línea es un control de seguridad, en caso de tener en la variable MODO un valor diferente a 0 o 2, se considera que hay un fallo y se fuerza el sistema para entrar en modo PARO.
L9	Si "MODO" es igual a "PARO", se apaga la bomba.
L10	Si "MODO" es igual a "AUTO" y la "CONSIGNA" es mayor que "MIN_ARRANQUE" se cada el valor de consigna desde la expansión de salida analógica "BOMBA"

Línea	Descripción
L11	Comentario para describir el programa. Esta línea no se ejecuta.
L12	Si el caudal actual supera el objetico de caudal, el PID del caudal toma el mando “PRES_CAU=1”
L13	Si el caudal actual es inferior al objetico de caudal menos la histéresis configurada, el PID de presión toma el mando “PRES_CAU=0”
L14	Si predomina el control por caudal “PRES_CAU=1”, se activa el modo tracking del PID “CONTROL_P”, se introduce la salida del PID “CONTROL_C” en la entrada track value del PID “CONTROL_P”, se desactiva el modo tracking del PID “CONTROL_C” y se establece como consigna la salida del PID “CONTROL_C”
L15	Si predomina el control por caudal “PRES_CAU=0”, se activa el modo tracking del PID “CONTROL_C”, se introduce la salida del PID “CONTROL_P” en la entrada track value del PID “CONTROL_C”, se desactiva el modo tracking del PID “CONTROL_P” y se establece como consigna la salida del PID “CONTROL_P”

17.6 CONFIGURACIÓN DEL MAPA DE MEMORIA MODBUS EN MODO ESCLAVO

Ejemplo de configuración del Mapa de memoria.

- En la dirección 30001 se mapea el valor de la entrada analógica 0 como un numero de coma flotante de precisión simple de 32 bits en modo "Big endian". Para ello se emplean las propiedades H y L de la variable R.
- En la dirección 30003 se mapea el valor de la entrada analógica 1 como un numero de coma flotante de precisión simple de 32 bits en modo "Big endian". Para ello se emplean las propiedades H y L de la variable R.
- En la dirección 30005 se mapea el valor del contador totalizador 0 como un entero de 32 bits en modo "Big endian". Para ello se emplean las propiedades H y L de la variable I.
- En la dirección 30007 se mapea una palabra de estado que contiene en los bits 0 y 1 el estado del flag de validez de las entradas analógicas 0 y 1. Para ello se emplean las propiedades Bx de la variable I.
- En la dirección 30008 se mapea el estado de las entradas digitales como un entero de 16 bits.

```
L1   #INIT
L2   REAL : R
L3   INT : I
L4   #END_INIT
L5   R = AI0
L6   MBIR0 = R.L
L7   MBIR1 = R.H
L8   R = AI1
L9   MBIR2 = R.L
L10  MBIR3 = R.H
L11  I = CNT0
L12  MBIR4 = I.L
L13  MBIR5 = I.H
L14  I = 0
L15  I.B0 = AI0.VLD
L16  I.B1 = AI1.VLD
L17  MBIR6 = I
L18  MBIR7 = DIP
```

Línea	Descripción
L1	Palabra clave para iniciar la sección de inicialización. La sección de inicialización se ejecuta una única vez al iniciar la ejecución del script.
L2	Declara "R" como una variable de tipo REAL.
L3	Declara "I" como una variable de tipo INT.
L4	Palabra clave para finalización de la sección de inicialización.

Línea	Descripción
L5	Carga en la variable “R” el valor de la entrada analógica 0.
L6	Carga en el MBIR0 (MODBUS Input register 0) los 16 bits de menos peso la variable “R”.
L7	Carga en el MBIR1 (MODBUS Input register 1) los 16 bits de más peso la variable “R”.
L8	Carga en la variable “R” el valor de la entrada analógica 1.
L9	Carga en el MBIR2 (MODBUS Input register 2) los 16 bits de menos peso la variable “R”.
L10	Carga en el MBIR3 (MODBUS Input register 3) los 16 bits de más peso la variable “R”.
L11	Carga en la variable “I” el valor del contador totalizador 0.
L12	Carga en el MBIR4 (MODBUS Input register 4) los 16 bits de menos peso la variable “I”.
L13	Carga en el MBIR5 (MODBUS Input register 5) los 16 bits de más peso la variable “I”.
L14	Inicializa la variable “I” a 0.
L15	Carga en el bit 0 de la variable “I” el valor del bit de validez de la entrada analógica 0.
L16	Carga en el bit 1 de la variable “I” el valor del bit de validez de la entrada analógica 1.
L17	Carga en el MBIR6 (MODBUS Input register 6) el valor de la variable “I”.
L18	Carga en el MBIR7 (MODBUS Input register 7) el estado del puerto de entradas digitales.