

Tema 8. MODBUS PC (MASTER) Variador MX2 (SLAVE), Regulador de temperatura E5CN (SLAVE) i PLC CP1H/L (SLAVE)

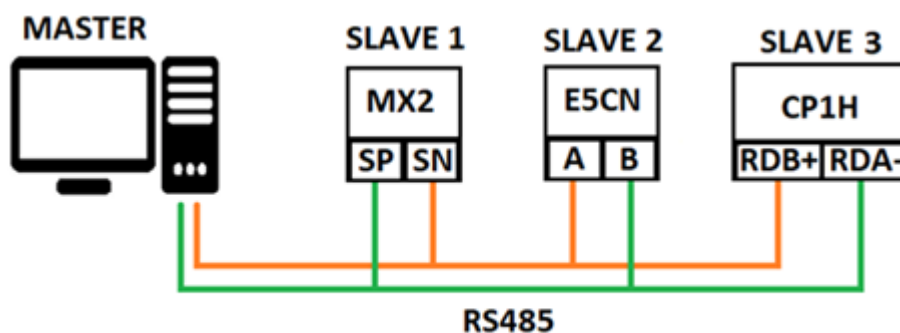
1.- Introducció

En este tema vamos a crear una red MODBUS, en la que tendremos los siguientes elementos:

- 1.- PC Windows 10 Aplicación de Visual Studio .NET **Master**
- 2.- Variador de frecuencia MX2 OMRON **Esclavo 01**
- 3.- Regulador de temperatura E5CN OMRON **Esclavo 02**
- 4.- PLC CP1H (o CP1L) OMRON **Esclavo 03**



Como capa física usaremos RS485. Los dos elementos que no disponen de conexión de RS485 (PC i PLC), usaremos el conversor pertinente.



2.- Cableado Completo del sistema

Cablear un elemento y comprobar envíos y respuestas, seguidamente cablear el siguiente elemento. Si no hay comunicación en los dos sentidos, intercambiar los cables de comunicación A por B y B por A.

3.- Configuración Variador MX2 para MODBUS

Antes de configurar el variador para su utilización en MODBUS, se recomienda como medida general, restaurar los valores de fábrica
(Página 286 Manual de Usuario)

<https://www.youtube.com/watch?v=CybJJKmgwWc>



Param	Descripción	Ajuste
B084	Selección de modo (Parámetros o Histórico)	00: Desactivado 01: Borra el histórico de fallos 02: Inicializa todos los parámetros 03: Borra el histórico de fallos e inicializa todos los parámetros 04: Borra el histórico de fallos, inicializa todos los parámetros y el programa EzSQ
B085	Selección de datos iniciales	00: Japón y EEUU 01: Europa
B094	Configuración de datos objetivo de la inicialización	00: Todos los parámetros 01: Todos los parámetros excepto los terminales de entrada/salida y las comunicaciones 02: Sólo los parámetros registrados en Uxxx 03: Todos los parámetros excepto los registrados en Uxxx y b037
B180	Activación de inicialización	01: Activación

Nota: Los parámetros resaltados en rojo muestran la configuración para el restablecimiento completo del variador a valores de fábrica.


Para configurar el Variador para comunicaciones MODBUS
(Página 303 del manual de usuario) (Versión gorda del manual impreso que tenemos por el aula)


<https://www.youtube.com/watch?v=xMEf7U98xZQ>

Resulta imposible la comunicación con la tarjeta Ethercat 3G3AX-MX2_ECT insertada en los variadores MX2. Para que el Variador acepte las tramas MODBUS y responda correctamente, hay que quitar la anterior tarjeta opcional Ethercat.

4.- Configuración Regulador E5CN para MODBUS



Primero pulsamos el botón  durante más de 3 segundos y luego 1 pulsación de menos de un segundo para pasar al Nivel de configuración de comunicaciones:

- El parámetro PSEL lo ponemos a CWF (CompoWay/F) para poder habilitar los apartados donde configuraremos los Bits de datos y Bits de parada. Seleccionamos la opción con las teclas de las flechas arriba y abajo. Una vez seleccionada la opción deseada, presionamos el botón  y quedará guardado el valor en el parámetro y nos mostrará el siguiente parámetro.



- El parámetro **U-No** nos indica el número de unidad de comunicaciones. En la imagen lo ponen a 1. En nuestro caso, pondremos el valor **2**.



- El parámetro **bPS** es la velocidad de transmisión la cual la configuraremos a **9600** bps.



- El parámetro **LEN** es la cantidad de Bits de datos que vamos a enviar, en este caso lo ponemos a **8**.



- El parámetro **Sbct** es la cantidad de Bits de parada, en este caso lo ponemos a **1**.




- El parámetro **PRtY** es la paridad de comunicaciones, en este caso lo ponemos en paridad **NONE** (Sin Paridad).





- El parámetro **SdWt** es el retardo de envío, este lo dejamos por defecto en **20ms**.



Una vez terminado de configurar estos parámetros volvemos al parámetro **PSEL** y lo configuramos en **Mod** para que sea una comunicación en **Modbus**.

Salimos al menú raíz haciendo una pulsación de tres segundos a la tecla . Reiniciamos el regulador, quitándole y poniéndole la alimentación para que los parámetros sean actualizados en el software del regulador.

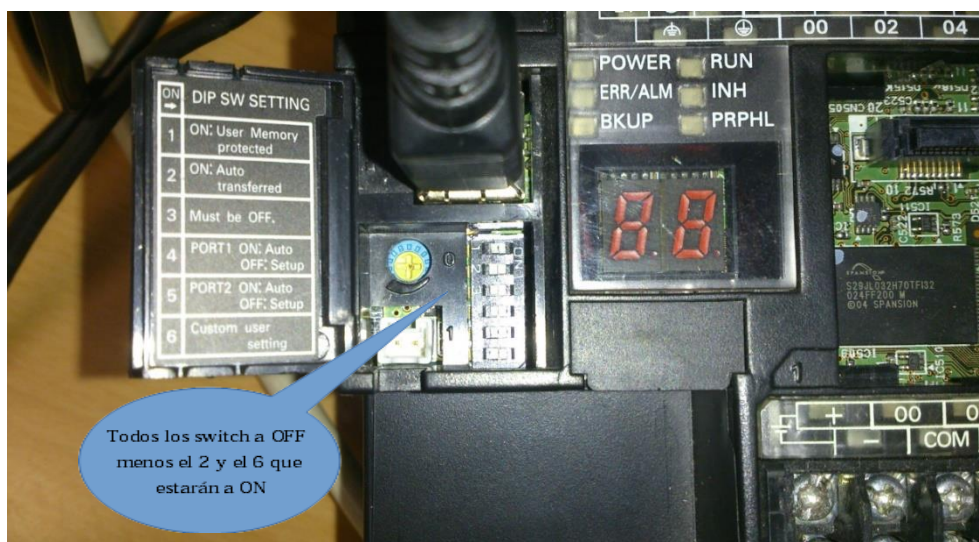


Para **cuando necesitemos escribir en la memoria** del regulador por comunicaciones (no es necesario para leer solo la temperatura, pero si para poner el regulador en modo RUN o STOP), tendremos que poner el parámetro **CMWt** a **ON**. Para acceder a este parámetro, desde el menú raíz que aparece después de reiniciar el regulador, presionar la tecla  menos de 1 segundo para acceder al nivel de ajuste (**LADJ**) y después la tecla . Observaremos el parámetro a modificar si es necesario (lo tenemos que dejar en ON)



5.- Configuración CP1H como esclavo MODBUS RTU

Para usar un PLC **CP1H** (o un CP1L) de **OMRON** como **esclavo Modbus**, debemos importar el **bloque de función** correspondiente que nos suministra **OMRON**, junto a su inestimable, documento de ayuda en pdf. El Bloque de función se llama **ModbusSlave.cxf**. La configuración del puerto será en RS-232C y deberemos usar 8 bits de datos. El switch que determina que se use o no la configuración por software del puerto, deberá estar en la posición correcta.

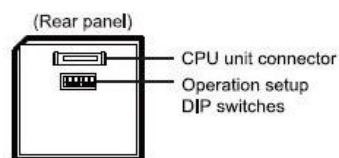


Switch	Selección	Descripción	Aplicación	Por defecto
SW1	ON	UM protegida	Proteger el programa contra sobrescritura no deseada	OFF
	OFF	UM no protegida		
SW2	ON	Transferencia automática de cassette de memoria a CPU	Permite realizar el volcado del programa, memoria de datos y setup, del cassette de memoria a la CPU	OFF
	OFF	Datos no transferidos		
SW3	---	No utilizar	---	OFF
SW4	ON	Toolbus	Establece unos parámetros de comunicación fijos para el slot 1 (Toolbus)	OFF
	OFF	Según "PLC Setup"		
SW5	ON	Toolbus	Establece unos parámetros de comunicación fijos para el slot 2 (Toolbus)	OFF
	OFF	Según "PLC Setup"		
SW6	ON	A395.12 a ON	Muy útil para realizar chequeo de programa sin necesidad de cablear una entrada física	OFF
	OFF	A395.12 a OFF		



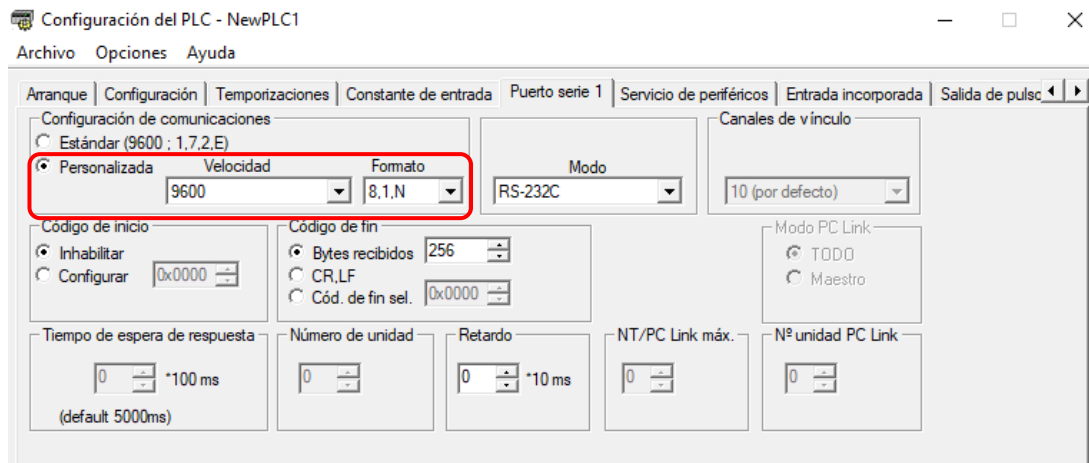
Todos lo Switch a ON menos el 4 y 5

1	<input type="checkbox"/>	ON
2	<input type="checkbox"/>	
3	<input type="checkbox"/>	
4	<input type="checkbox"/>	
5	<input type="checkbox"/>	
6	<input type="checkbox"/>	



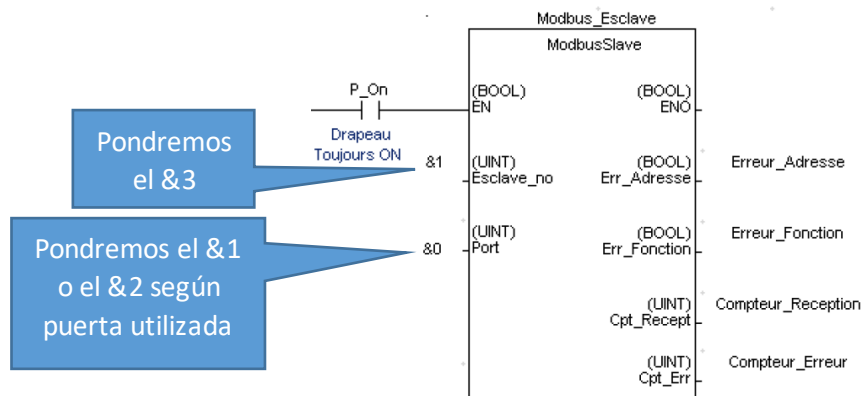
No.	Setting	ON/OFF	Content
1	Presence of terminating resistance	ON	Terminating resistance present
2	2/4-wire selection	ON	2-wire type
3	2/4-wire selection	ON	2-wire type
4	-	OFF	Always OFF
5	RS control for RD	ON	Enabled
6	SD control for RD	ON	Enabled

La configuración del setup del PLC CP1H/L será (la parte dinámica según cada caso):



Para la programación usamos un bloque de función.

El bloque de función tiene 3 parámetros de entrada y 4 parámetros de salida.



El **UINT Esclave_no** es el identificador de esclavo de MODBUS y el **UINT Port** es la puerta donde está alojado el CIF11. Con un CP1H, podremos utilizar 0 o 1. En un CP1L, solo podremos utilizar el 0.

Parámetros de Entrada

- 1.- **EN** (BOOL): Se trata del *enable* del bloque. Podemos utilizar un Contacto de siempre ON.
- 2.- **Esclave_no** (UIN): Es el número de esclavo. Puede ser desde 1 a 247.
- 3.- **Port** (UINT): Pondremos &1 o &2 para elegir la puerta opcional por el que comunicamos.

Parámetros de Salida

- 1.- **ENO** (BOOL): No se utiliza.
- 2.- **Err_Adresse** (BOOL): Se activa cuando se ha producido un error de dirección ya sea de lectura o de escritura en el esclavo.
- 3.- **Err_Fonction** (BOOL): Se activa cuando se ha producido un error de función no soportada por el esclavo.
- 4.- **Cpt_Recept** (UINT): 0000 a FFFF. Contador de recepción de tramas con el CRC16 correcto.
- 5.- **Cpt_Err** (UINT): 0000 a FFFF. Contador de recepción de tramas con el CRC16 correcto pero con dirección o función incorrecto.

Las funciones soportadas por el esclavo son las siguientes:

Code	Function	Name in MODBUS
0x01	I/O memory area (CIO) Read bits	Read Coils
0x02	I/O memory area (CIO) Read bits	Read Discrete Inputs
0x03	I/O memory area (DM) Read Multiple Registers	Read Holding Registers
0x04	I/O memory area (CIO) Read Multiple Registers	Read Input Registers
0x05	I/O memory area Write Single Coil (CIO)	Write Single Coil
0x06	I/O memory area (DM) Write Single Register	Write Single Register
0x08	Echo back test	Diagnostic
0x0F	***** NOT SUPPORTED *****	Write Multiple Coils
0x10	I/O memory area (DM) Write Multiple Registers	Write Multiple Registers

6.- Tramas MODBUS para el control del Variador MX2

Para controlar el Variador MX2, necesitamos solamente, unas direcciones de memoria: (Página 298 Manual de Usuario Variador MX2)

- a) **0001 Dirección del bit de Operación.** En este bit, vamos a escribir **FF00** para que el motor se ponga en **RUN** y **0000**, para que se ponga en **STOP**. Para escribir sobre el bit 0000, podemos usar la función **05** de MODBUS (*Write Single Coil, Escribir en bit de salida*). La trama para que se ponga en RUN y en STOP podría ser:

RUN	01 05	00 00	FF 00 CRC0 CRC1	En la trama pongo una dirección menos
STOP	01 05	00 00	00 00 CRC0 CRC1	

- b) **0002 Dirección del bit de Sentido de Giro.** En este bit, vamos a escribir **FF00** para que **gire en un sentido** y **0000**, para que gire en **sentido contrario**. Para escribir sobre el bit 0001, podemos usar la función **05** de MODBUS (*Write Single Coil, Escribir en bit de salida*). La trama para que determinar un sentido o el sentido contrario podría ser:

→ **01 05 00 01 FF 00 CRC0 CRC1**
 ← **01 05 00 01 00 00 CRC0 CRC1**

En la trama
pongo una
dirección menos

Es decir, podremos enviar una trama para que se ponga en RUN o en STOP e independientemente a esta trama, otra trama para determinar en qué sentido deberá girar el motor.

Pero tenemos la posibilidad de enviar una única trama para escribir en varios bits de salida consecutivos. Se trata de la función **0F** (**página 290 manual de usuario hay un ejemplo**). Las tramas para poner el motor a girar en un sentido o en girar en el sentido contrario, serían:

RUN → **01 0F 00 00 00 02 02 01 00 CRC0 CRC1**
 RUN ← **01 0F 00 00 00 02 02 03 00 CRC0 CRC1**

Ver direcciones
Página 301 del
manual de usuario

- c) **Registro 0001 y registro 0002**, son los registros donde pondremos la frecuencia a la que se pondrá a girar el motor. Por ejemplo, la trama para poner a 10 Hz (1000 en hexadecimal es 03 E8) la frecuencia del variador es:

01 10 00 00 00 02 04 00 00 03 E8 CRC0 CRC1

En la trama
pongo una
dirección menos

7.- Tramas MODBUS para el control del Regulador E5CN

Trabajaremos solo con dos registros. La temperatura que mide nuestro regulador a través de su sonda (termopar por ejemplo) se guarda en dos registros, el 0000 y el 0001. Para leer esta temperatura, usaremos la función 03 de Modbus. Si el regulador está configurado como nodo 02, la trama para leer la temperatura será:

Leer Temperatura

→ **02 03 00 00 00 02 CRC0 CRC1**

Para activar el modo RUN o desactivarlo, usaremos la función 06 de Modbus. Escribiendo el parámetro 0100 en el registro 0000 para activar el modo RUN y el parámetro 0101 para desactivar el modo RUN (aparecerá el mensaje STOP en la pantalla del regulador).

Activar Modo RUN

→ **02 06 00 00 01 00 CRC0 CRC1**

Desactivar Modo RUN

→ **02 06 00 00 01 01 CRC0 CRC1**

8.- Tramas MODBUS para el control del CP1H/L

En el punto 5 de este documento, vimos cuales son las funciones soportadas por el PLC CP1H/L. Podemos usar la que necesitemos en cada situación. Solo tenemos que seguir la sintaxis correcta para cada función, según vimos en el tema 9.

A modo de ejemplo usaremos la función **03** (Read DM Data Memory) y **06** (Write 1 Word in DM área). Estas funciones nos permitirán leer un DM o escribir en un DM de nuestro PLC esclavo.

Leer DM 1000

→ **03 03 03 E8 00 01 CRC0 CRC1**

(1000 en hexadecimal es 03 E8)

Escribir CACA en DM01000

→ **03 06 03 E8 CA CA CRC0 CRC1**

O también podemos usar la función **05** (Write 1 bit in CIO area) para activar o desactivar la salida 100.03

Activar 100.03

→ **03 05 06 43 FF 00 CRC0 CRC1**

Desactivar 100.03

→ **03 05 06 43 00 00 CRC0 CRC1**

En la anterior función, los bits de la zona CIO, se enumeran desde el bit 0.00 que sería el bit 0 en una trama MODBUS. Es decir, Si queremos acceder al bit 2.02 de la zona CIO, haríamos $(2 \times 16) + 2 = 34$ que en hexadecimal sería 0x22. En la trama pondríamos la parte baja de la dirección, 00 y la parte alta de la dirección sería 22. Tenemos en cuenta que en cada registro tiene 16 bits.

En nuestro Ejemplo, el bit 100.03 se direccionará $(100 \times 16) + 3 = 1603 = 0x643$. Parte baja 06 y parte alta 43.

Una vez direccionado el bit, 00FF lo activa (ON) y 0000 lo desactiva (OFF).

Podemos usar también la función **04** (Read CIO) para leer uno o varios bits de la zona CIO. Por ejemplo, si queremos leer el bit 1.04 y los sucesivos 18 bits siguientes (un total de 19 bits), deberemos especificar la dirección del primer bit a leer, que según el criterio

explicado antes, sería $(1 \times 16) + 4 = 20 = 0 \times 14$. Parte alta 00 y parte baja 14.

En las respuestas de estas lecturas, aparecerá un byte por cada 8 bits leídos. Así tendríamos un byte para los estados del bit 20 al 27. Otro para los estados del bit 28 al 35 y otro para los bits 36, 37 y 38 (se completan con ceros el byte que no esté completo en la respuesta)

La trama para leer la entrada 0.5 de nuestro PLC sería:

Leer Entrada 0.05

→ **03 04 00 05 00 01 CRC0 CRC1**

La dirección se ha calculado de la siguiente manera:
 $(0 \times 16) + 5 = 5 = 0 \times 5$. Siguiendo el criterio explicado anteriormente para el formato de las respuestas, las tramas recibidas en función del estado de la entrada leída serán:

Respuesta si 0.05 es ON

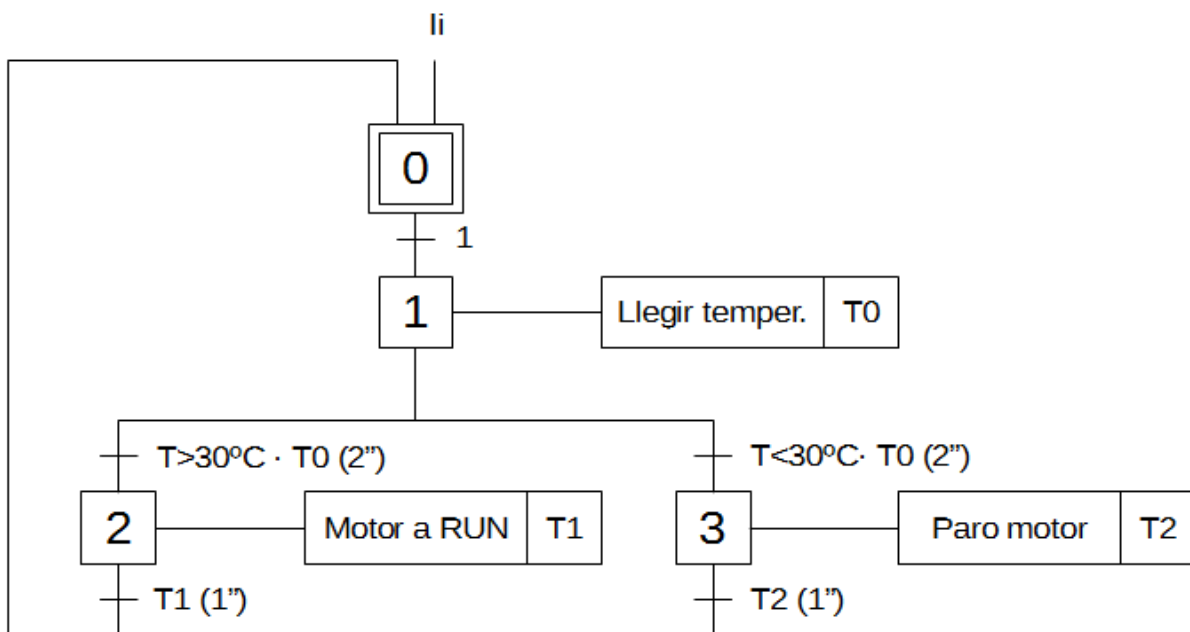
→ **03 04 01 01 CRC0 CRC1**

Respuesta si 0.05 es OFF

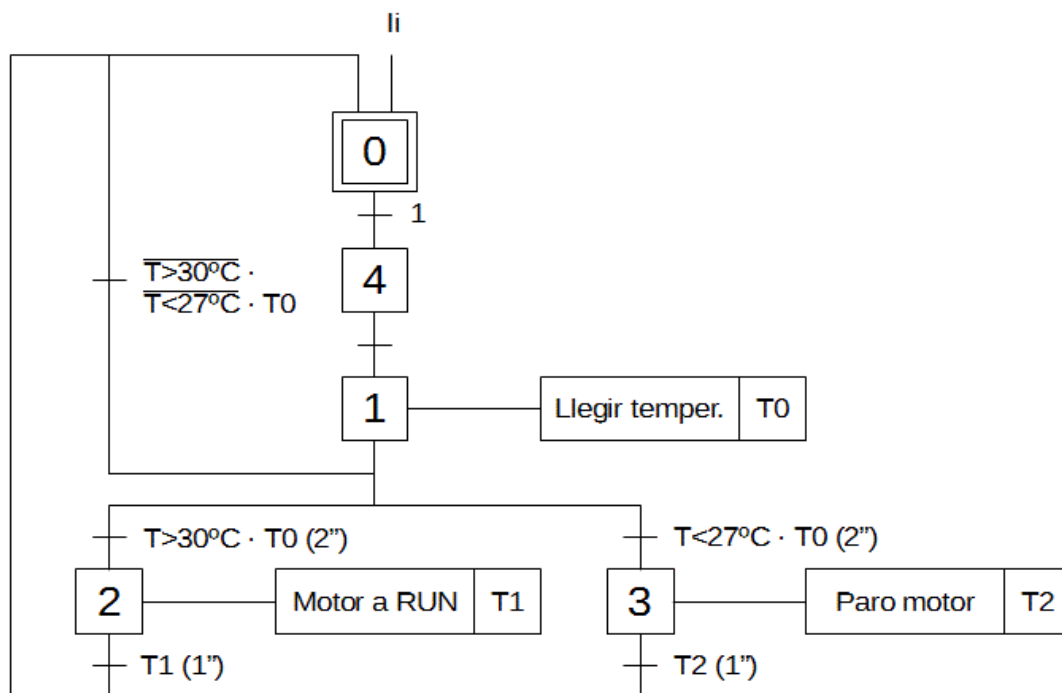
→ **03 04 01 00 CRC0 CRC1**

Aplicación: Control del motor en función de la temperatura leída.

A continuación vamos a mostrar el graficet, correspondiente al caso en el que, según la temperatura del regulador, encenderemos o apagaremos el motor. En este caso lo hemos realizado sin la histéresis, es decir, si la temperatura supera los 30 °C, el motor se pondrá en marcha, una vez baje de los 30 °C, el motor se parará. Este es el graficet del programa de control de temperatura SIN histéresis:



Ahora toca mostrar el segundo caso, que es el mismo control de temperatura pero ahora añadiéndole la histéresis. En esta segunda parte, cuando la temperatura supere los 30 °C igual que en el otro caso, el motor empezará a girar (simulando un ventilador), lo que cambia es que aquí, cuando la temperatura baje de 27 °C, es cuando el motor parará de girar, no antes. Por tanto tenemos que una vez se sobrepase los 30 °C, el motor estará en marcha hasta que la temperatura baje de los 27 °C. A continuación puedes ver el graficet del planteamiento descrito:



Consideraciones finales

- El PLC NX1P2 no acepta la función de esclavo de MODBUS RTU, solo puede ser Maestro en una red de este tipo. Sección 4-1 Machine Automation Controller NX-series NX1P2 CPU Unit
The function to use **NX1P2** CPU Units as **Modbus-RTU slaves** is not supported
- Posible ampliación de la red con el uso del visualizador FEMA como esclavo MODBUS RTU.
- Para modificar las rampas de aceleración y desaceleración del variador, buscar las direcciones de los registros implicados y realizar una trama como la de escribir la frecuencia (1103 y 1104). El tiempo en las rampas se introduce en centésimas de segundo (0,01 s). Por ejemplo si queremos poner 10s >1000 que en hexadecimal es 03E8.