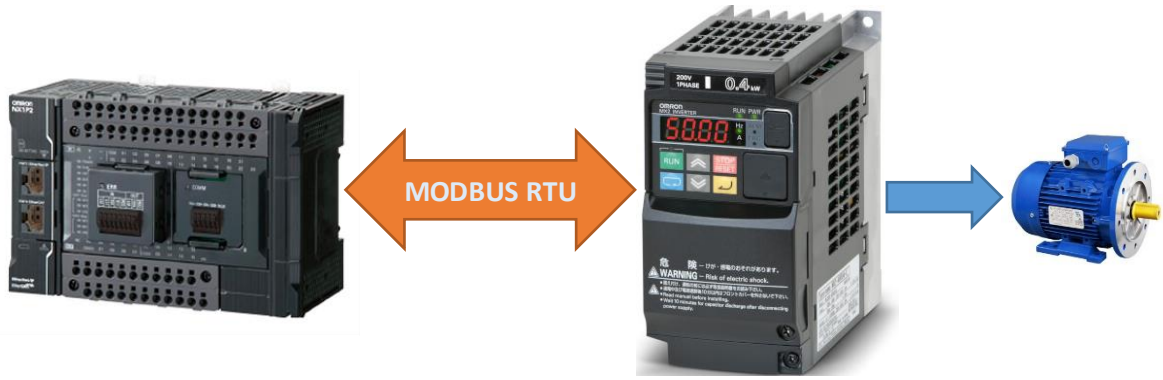
Tema 9. MODBUS RTU NX1P2 (MASTER) Variador MX2 (SLAVE)

1.- Introducción

Seguramente, ya sabrás que un **variador de frecuencia**, no es más que un controlador de **motores industriales**, que se utilizan para determinar el **sentido**, la **frecuencia**, la **aceleración** y multitud de parámetros más, en los motores trifásicos industriales.



Probablemente, también hayas realizado el control de un motor trifásico mediante un variador de frecuencia, utilizando como elemento inteligente, un **autómata programable**.

En este tema, vamos a realizar, este tipo de control, pero utilizando un conexionado **RS485**, entre el PLC y el variador, y **MODBUS RTU**, como protocolo de comunicación.

En este caso, el PLC actuará como **maestro** (cliente) MODBUS, y el variador, funcionará como **esclavo** (servidor) MODBUS. El PLC podría controlar hasta 31 inversores diferentes, cada uno de ellos, con un motor trifásico diferente.

2.- Cableado

Usaremos el NX1W-CIF-11 para la comunicación RS-485, en la **puerta 2** del PLC NX1P2.

El **NX1W-CIF11** dispone de unos conmutadores tipo **switch** para configurar una serie de parámetros referentes a la comunicación. En el documento ***Programación y Comunicaciones.pdf***, (Visto en el tema 5 Comunicación NO-PROTOCOL) en la página 4-24, podemos encontrar una tabla, con la función de cada uno de los anteriores switch:

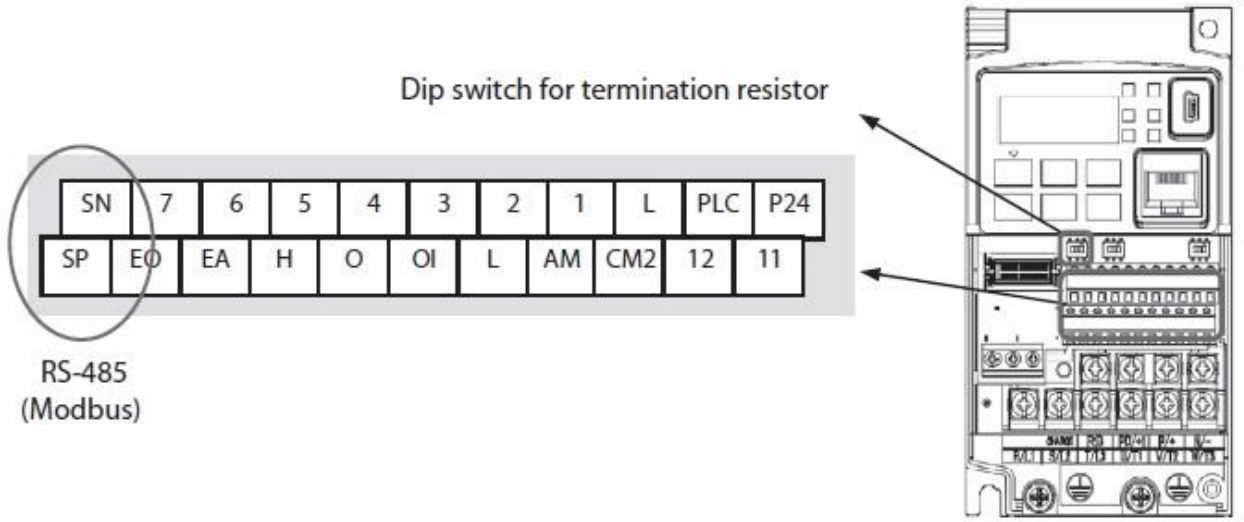
CIF11		CIF12		Setting	Setting description
SW	No.	SW	No.		
SW1	1	SW1	1	ON	With terminating resistance
	2		2	ON	Two-wire type
	3		3	ON	Two-wire type
	4		4	OFF	(Not used)
	5	SW2	1	ON	With RS control for receive data
	6		2	ON	With RS control for send data

Número SW	Función	Valor utilizado
1	Activa la resistencia limitadora de final de BUS	SI el PLC es el último elemento del BUS, a ON. Si no es el último elemento, a OFF.
2	Comunicación 2 hilos	Si usamos RS485 a ON. Si usamos RS422 a OFF.
3	Comunicación 2 hilos	Si usamos RS485 a ON. Si usamos RS422 a OFF.
4	(sin usar)	(sin usar)
5	Utilización de la señal de control RS, en la recepción de la comunicación.	Si la utilizamos a ON. Si no la utilizamos a OFF.
6	Utilización de la señal de control RS, en el envío de la comunicación.	Si la utilizamos a ON. Si no la utilizamos a OFF.

En nuestro caso, el PLC será el **último** elemento del bus, usaremos comunicación de **2** hilos **RS485** y activaremos el control RS, tanto para el envío como para la lectura.

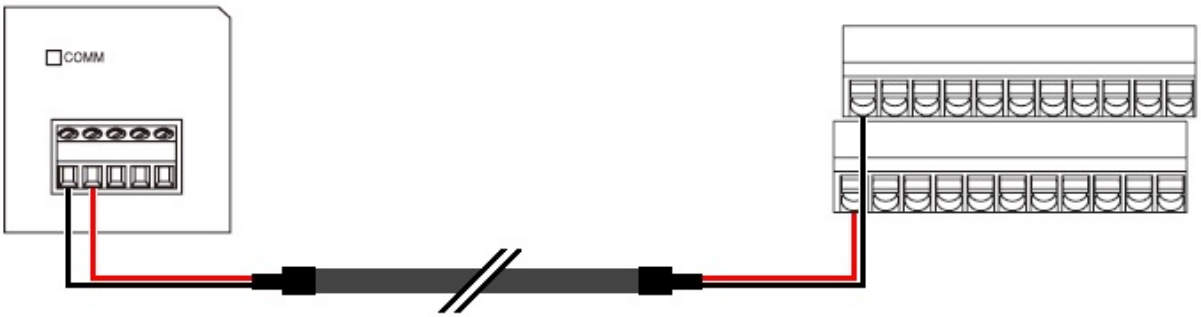
Número SW	Valor utilizado
1	**ON**
2	**ON**
3	**ON**
4	**OFF**
5	**ON**
6	**ON**

El variador MX2, ya dispone de conexiones para la comunicación RS485, tal y como se aprecia en la siguiente figura:



De la misma manera que hemos hecho con el PLC, activaremos también en el MX2, la resistencia de fin de línea, en el caso concreto de que este sea el último elemento del bus.

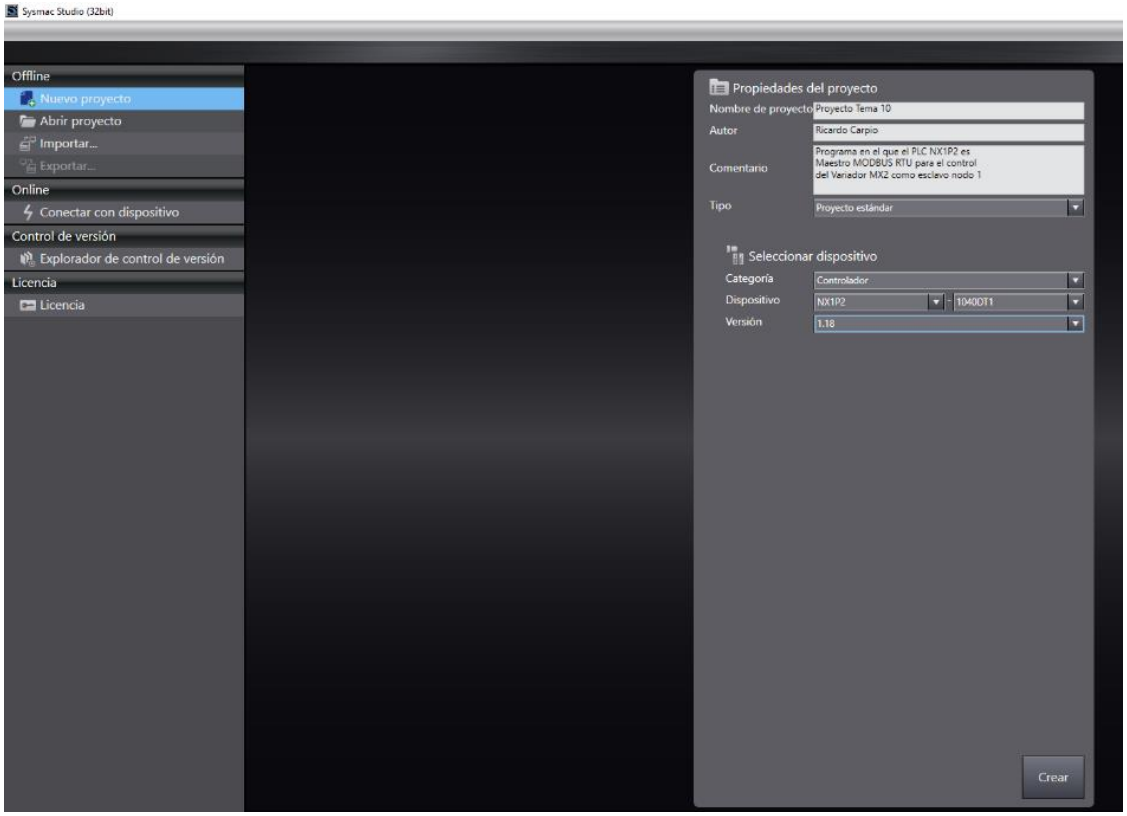
Los terminales que vamos a utilizar para la comunicación RS485 son los terminales SP y SN.



Conectaremos el terminal del CIF11, **RDA- con el terminal SN** del MX2 y el terminal **RDA+ con el SP**, tal y como puedes apreciar en la anterior figura.

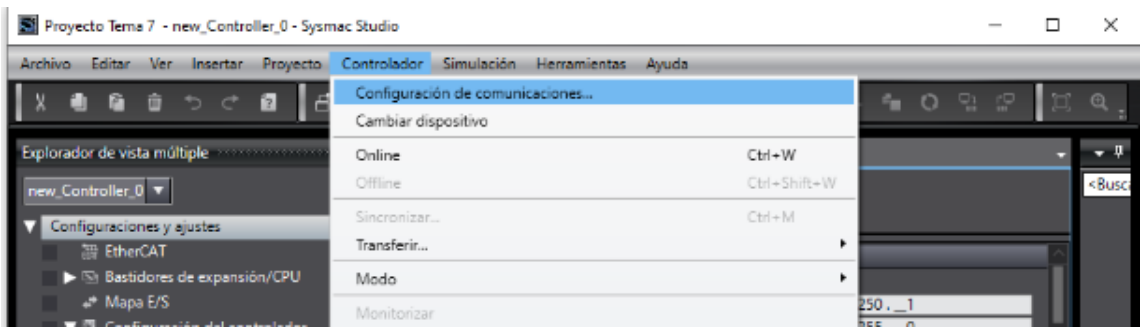
3.- Creación del proyecto en SYSMAC STUDIO

Una vez esté todo perfectamente cableado y configurados los switchs del módulo NX1W-CIF11, crearemos un nuevo proyecto en el SYSMAC STUDIO.

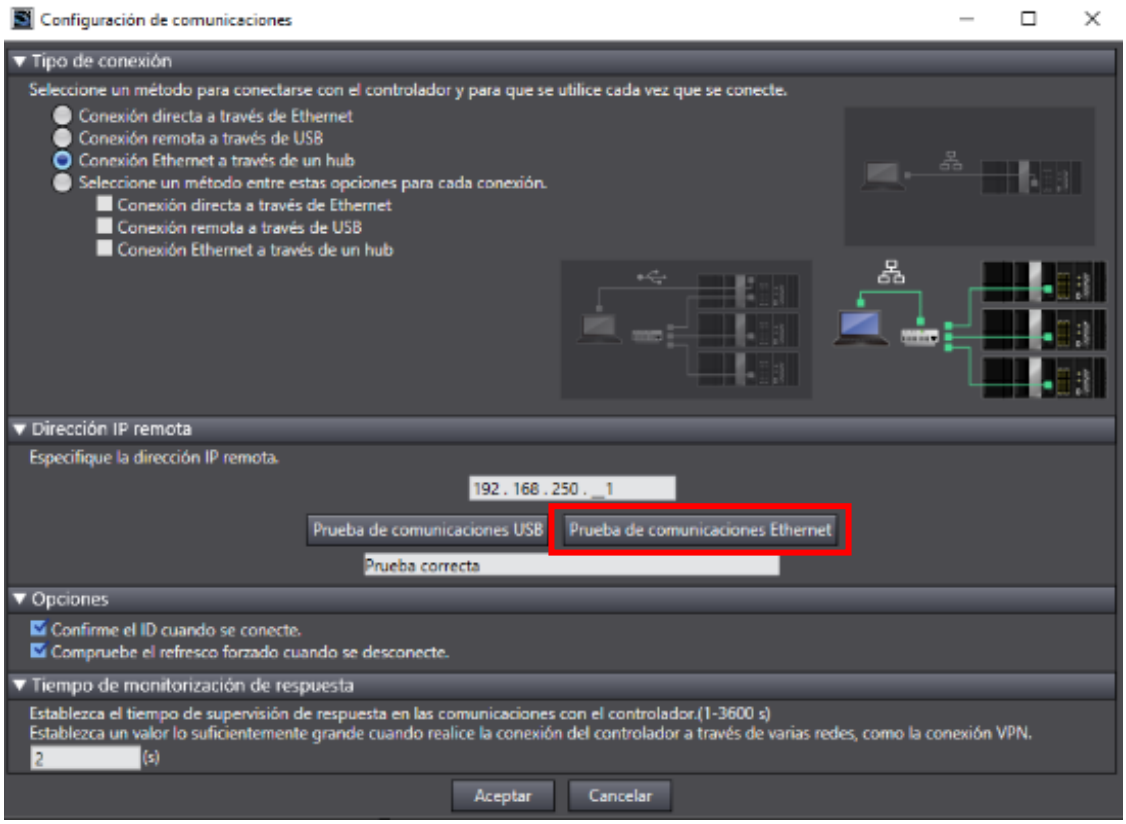


Como por defecto, el PLC tiene la dirección ip 192.168.250.001, pondremos en nuestro ordenador, una ip del mismo rango. Por ejemplo pondremos la 192.168.250.002.

Accederemos al menú ***Controlador*** > ***Configuración de comunicaciones***:

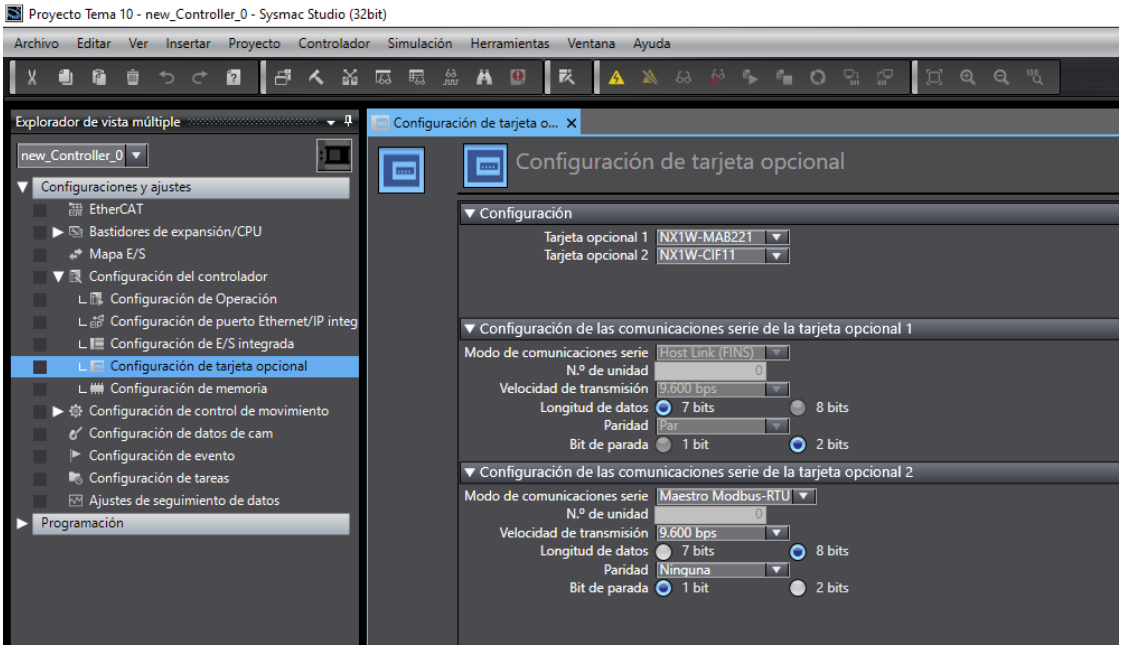


Estableceremos el modo de comunicación a través de un HUB y pondremos la ip del PLC que por defecto es la 192.168.250.001:



Si todo ha ido bien, haremos click en el botón Prueba de comunicaciones Ethernet y nos debe aparecer un mensaje de **prueba correcta**.

Ahora vamos a configurar las tarjetas opcionales que tengamos en el PLC. En nuestro caso tendremos en el slot 1 la **NX1W-MAB221** y en el slot 2, la **NX1W-CIF11**:

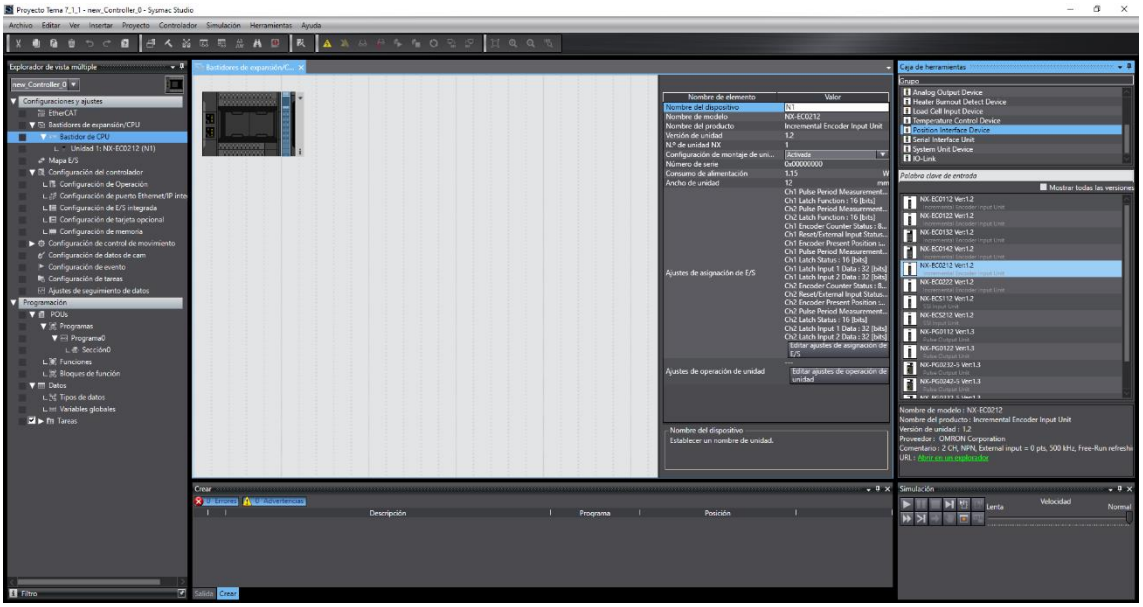


Como vemos en la anterior imagen, configuraremos el NX1W-CIF11 como sigue:

Nombre del Campo	Valor seleccionado
Modo de Comunicación Serie	Maestro Modbus-RTU
Número de Unidad	No requerido
Velocidad de transmisión	9600 bps
Longitud de datos	8
Paridad	Ninguna
Bit de parada	1 bit

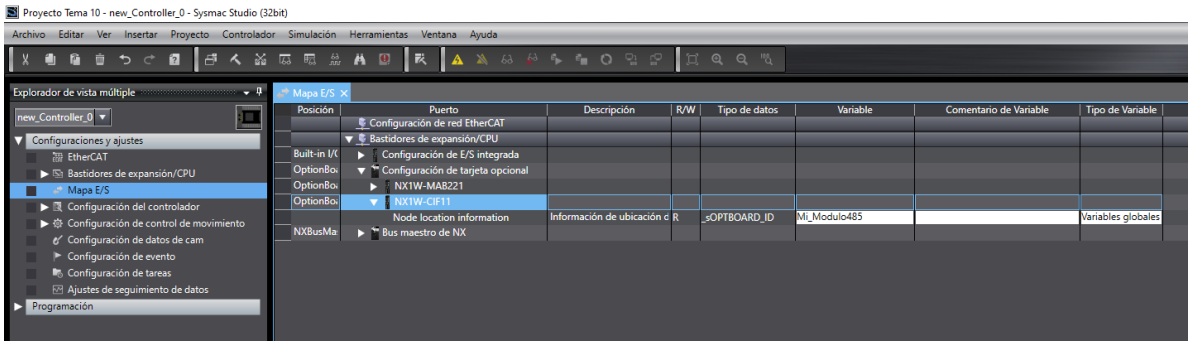
Los anteriores parámetros de la comunicación serie, deben ser idénticos en todos los elementos que introduzcamos en el bus.

También tenemos en nuestro PLC, en el bastidor de la CPU, una unidad de entrada para encoder incremental. Es la ***NX-EC0212***. Solo tendremos que acceder al menú ***Bastidor de CPU***, seleccionar el grupo ***Position Interface Device***, seleccionar el dispositivo ***NX-EC0212*** y arrastrarlo hasta la CPU. Se colocará entre la CPU y el terminal de finalización que llevan todos los PLC de esta gama.



Con esto ya tenemos configurado en SYSMAC STUDIO, nuestro PLC, tal y como lo tenemos en la realidad.

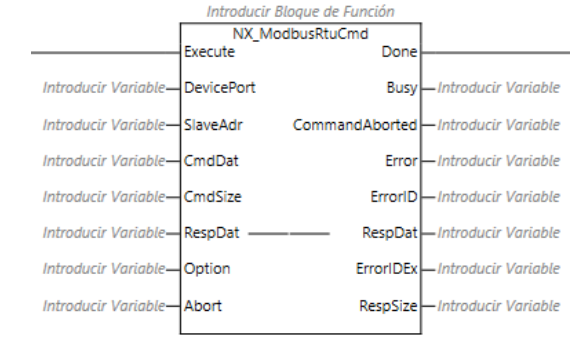
Ahora crearemos una variable de dispositivo en el módulo NX1W-CIF11 en el apartado Mapa E/S que llamaremos **Mi_Modulo485**.

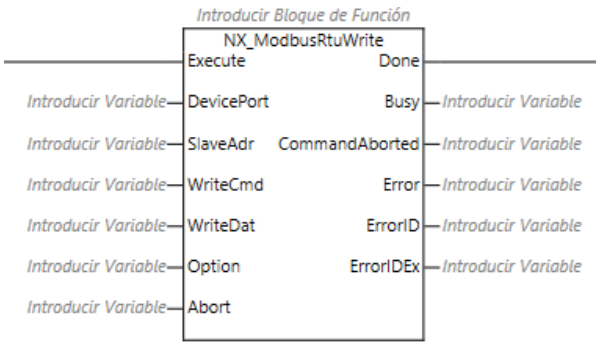


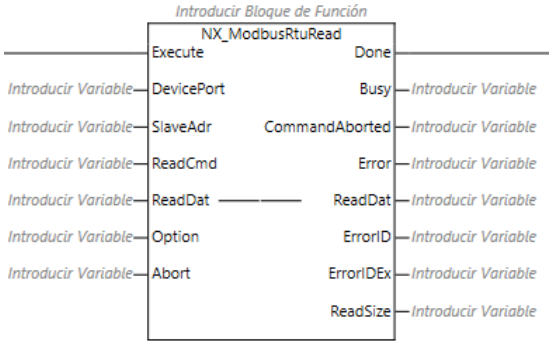
Podremos ahora conectarnos y transferir al PLC el proyecto realizado para comprobar que no hay NINGÚN error en el PLC.

4.- Instrucciones utilizadas en la comunicación MODBUS RTU.

SYSMAC STUDIO nos ofrece 3 funciones para el control de esclavos MODBUS RTU desde el PLC. Estas Funciones son:







1.-NX_ModbusRTUCmd: Mediante este bloque de función podremos enviar tanto tramas de lectura como de escritura a cualquiera de los esclavos Modbus que tengamos en la red.

2.-NX_ModbusRTUWrite: Mediante este bloque de función podremos enviar solo tramas de escritura a cualquiera de los esclavos Modbus que tengamos en la red

3.-NX_ModbusRTURead: Mediante este bloque de función podremos enviar solo tramas de lectura a cualquiera de los esclavos Modbus que tengamos en la red.

Como podemos comprobar, los tres bloques comparten la mayoría de entradas y salidas. Explicaremos a continuación, los parámetros de entrada y salida de los bloques **NX_ModbusRTUWrite** y **NX_ModbusRTURead**.

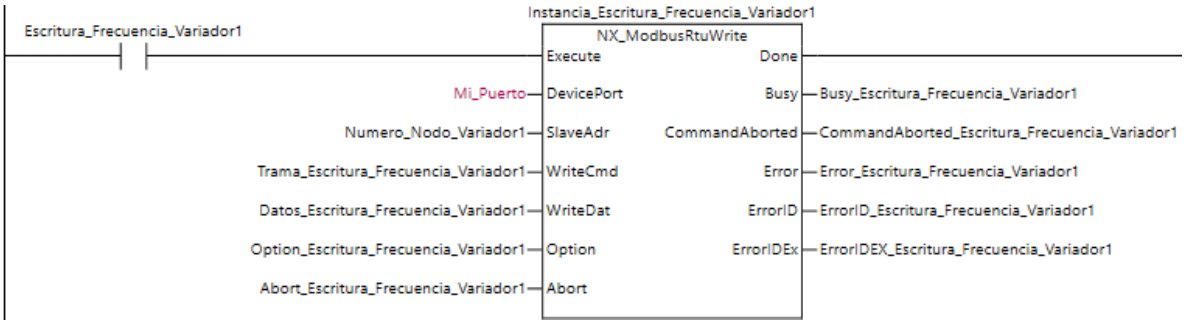
NX_ModbusRTUWrite

Lectura_Frecuencia_V...or1... | Escritura_Frecuencia...or1 -...

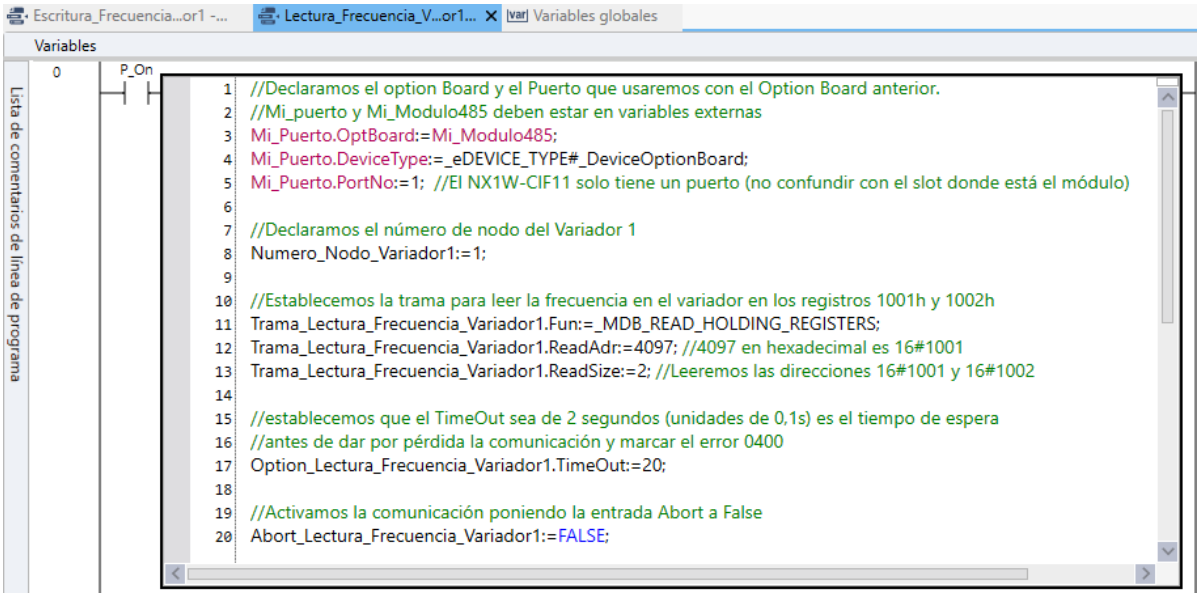
Variables

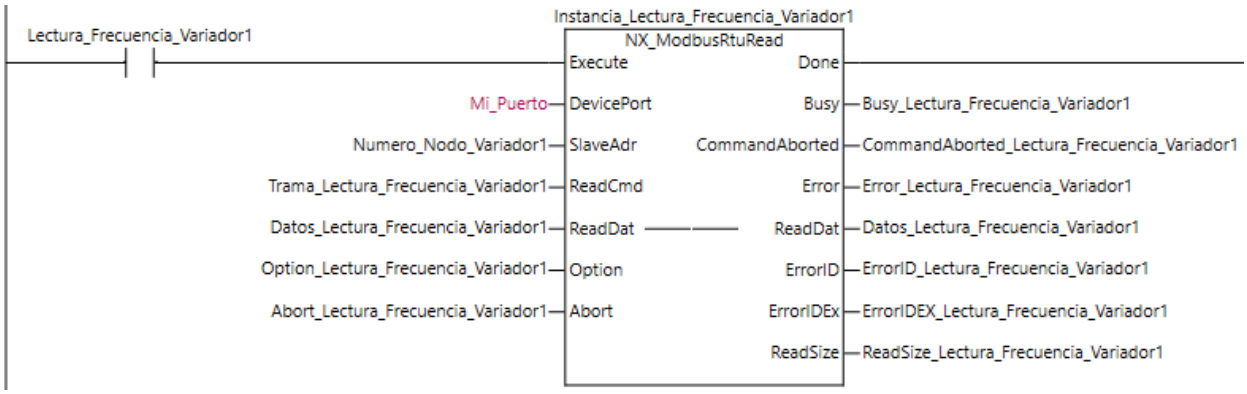
Lista de comentarios de línea de programa

```
1  //Declaramos el option Board y el Puerto que usaremos con el Option Board anterior.
2  //Mi_puerto y Mi_Modulo485 deben estar en variables externas
3  Mi_Puerto.OptBoard:=Mi_Modulo485;
4  Mi_Puerto.DeviceType:=_eDEVICE_TYPE#_DeviceOptionBoard;
5  Mi_Puerto.PortNo:=1;  //El NX1W-CIF11 solo tiene un puerto (no confundir con el slot donde está el módulo)
6
7  //Declaramos el número de nodo del Variador 1
8  Numero_Nodo_Variador1:=1;
9
10 //Establecemos la trama para escribir la frecuencia en el variador en los registros 0001h y 0002h
11 Trama_Escritura_Frecuencia_Variador1.Fun:=_MDB_WRITE_MULTIPLE_REGISTERS;
12 //las direcciones para escribir la frecuencia, son las 16#0001 y 16#0002 pero como vamos a escribir
13 //en ellas, pondremos una unidad menos. Escribiremos en las 16#0000 y 16#0001
14 Trama_Escritura_Frecuencia_Variador1.WriteAdr:=0; //direcciones 16#0000 y 16#0001 para escribir la frecuencia
15 Trama_Escritura_Frecuencia_Variador1.WriteSize:=2;
16
17 //La frecuencia la escribimos en hexadecimal y en unidades de 0,01Hz. 5000 equivale a 50 Hz.
18 Datos_Escritura_Frecuencia_Variador1[0]:=0;
19 Datos_Escritura_Frecuencia_Variador1[1]:=16#1388; //5000 en decimal que son 16#1388 (50 Hz)
20
21 //establecemos que el TimeOut sea de 2 segundos (unidades de 0,1s) es el tiempo de espera
22 //antes de dar por pérdida la comunicación y marcar el error 0400
23 Option_Escritura_Frecuencia_Variador1.TimeOut:=20;
24
25 //Activamos la comunicación poniendo la entrada Abort a False
26 Abort_Escritura_Frecuencia_Variador1:=FALSE;
27
```

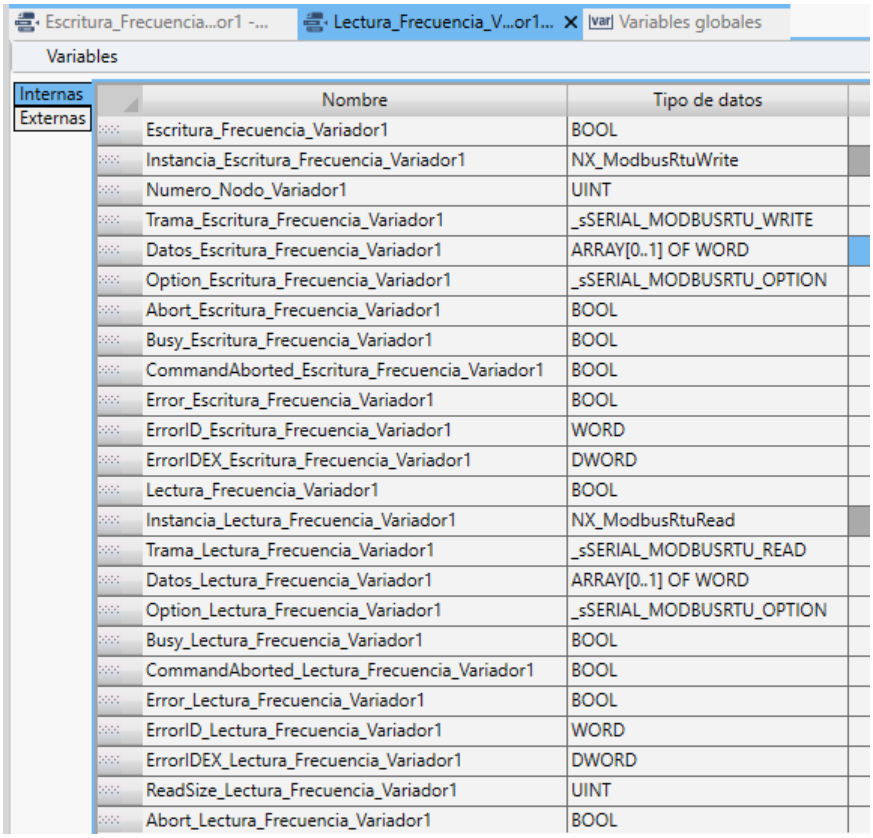


NX_ModbusRTURead

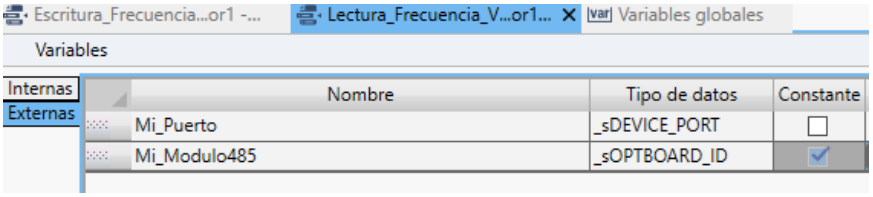




Para las dos secciones anteriores de programa las variables que se usan son del tipo que se aprecia en la siguiente captura:

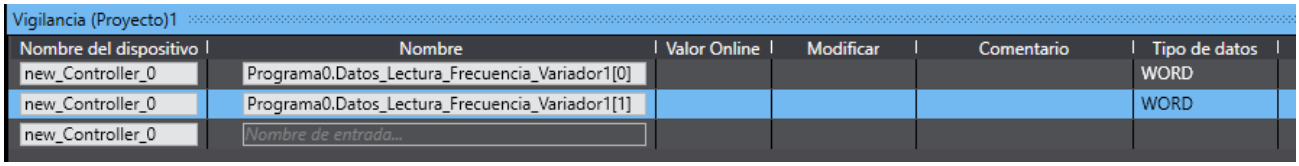


Nombre	Tipo de datos
Escritura_Frecuencia_Variador1	BOOL
Instancia_Escritura_Frecuencia_Variador1	NX_ModbusRtuWrite
Numero_Nodo_Variador1	UINT
Trama_Escritura_Frecuencia_Variador1	_sSERIAL_MODBUSRTU_WRITE
Datos_Escritura_Frecuencia_Variador1	ARRAY[0..1] OF WORD
Option_Escritura_Frecuencia_Variador1	_sSERIAL_MODBUSRTU_OPTION
Abort_Escritura_Frecuencia_Variador1	BOOL
Busy_Escritura_Frecuencia_Variador1	BOOL
CommandAborted_Escritura_Frecuencia_Variador1	BOOL
Error_Escritura_Frecuencia_Variador1	BOOL
ErrorID_Escritura_Frecuencia_Variador1	WORD
ErrorIDEX_Escritura_Frecuencia_Variador1	DWORD
Lectura_Frecuencia_Variador1	BOOL
Instancia_Lectura_Frecuencia_Variador1	NX_ModbusRtuRead
Trama_Lectura_Frecuencia_Variador1	_sSERIAL_MODBUSRTU_READ
Datos_Lectura_Frecuencia_Variador1	ARRAY[0..1] OF WORD
Option_Lectura_Frecuencia_Variador1	_sSERIAL_MODBUSRTU_OPTION
Busy_Lectura_Frecuencia_Variador1	BOOL
CommandAborted_Lectura_Frecuencia_Variador1	BOOL
Error_Lectura_Frecuencia_Variador1	BOOL
ErrorID_Lectura_Frecuencia_Variador1	WORD
ErrorIDEX_Lectura_Frecuencia_Variador1	DWORD
ReadSize_Lectura_Frecuencia_Variador1	UINT
Abort_Lectura_Frecuencia_Variador1	BOOL

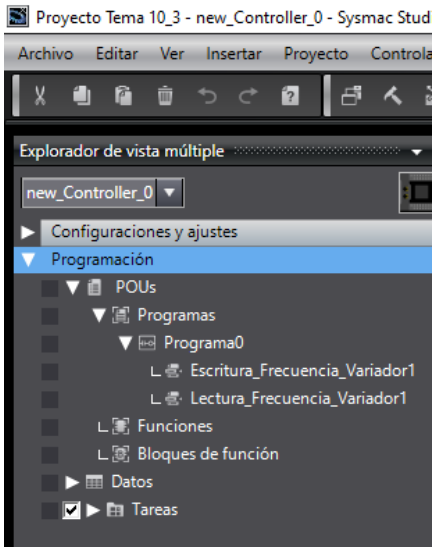


Nombre	Tipo de datos	Constante
Mi_Puerto	_sDEVICE_PORT	☐
Mi_Modulo485	_sOPTBOARD_ID	☑

Podemos utilizar una ventana de vigilancia para visualizar el contenido del Array de Words Datos_Lectura_Frecuencia_Variador1 como en la siguiente captura:



Recordemos ser ordenados en la programación y crear una sección diferente para cada trama Modbus que tenga que enviar nuestro programa.



5.- Tabla de Errores

ErrorID	Détail
0419	Type de la variable ReadDat inadaptée ou adresse CIF incorrecte
0C00	La carte série n'est pas copnfiguré dans le mode correspondant à l'instruction
0C03	Buffer de réception plein
0C04	Plusieurs instruction de communication ont été exécutées simultanément
0C05	Erreur de parité
0C06	Erreur de trame
0C07	Vitesse de réception excessive
0C08	Erreur CRC
0C0B	TimeOut
0C0C	Instruction exécutée sur un port indéterminé
0C10	Code d'erreur retourné par l'exsclave consigné dans la variable ErrorIDEx
0C11	Réponse inattendue reçu de l'esclave
0400	Format/quantité incorrect dans la requête de lecture/écriture Modbus
0415	Erreur dans la configuration du CIF (exécutez une comparaison)

6.- Apéndice de Actividades

5.1.- Busca en la documentación del variador las direcciones de escritura de frecuencia y comprueba que efectivamente son las 0001h y 0002h.

5.2.- Busca en la documentación del variador las direcciones de lectura de frecuencia y comprueba que efectivamente son las 1001h y 1002h.

5.3.- Busca en la ayuda de las instrucciones utilizadas, las unidades del parámetro TimeOut.

5.4.- Creación de las secciones oportunas para poner en marcha el motor, en un sentido u otro y pararlo.

5.5.-Creación de las secciones oportunas para enviar las tramas para escribir una rampa de aceleración y una rampa de desaceleración.

5.6.- Integrar en el sistema una pantalla HMI kinco y diseñar una interface gráfica para realizar todas las acciones del programa incluidas las de los apartados 5.4 y 5.5.

7.- Apéndice de Estrategia para exportar a HMI

Escritura Frecuencia

1.- Bit que ejecuta: Escritura_Frecuencia_Variador1

2.- Variable donde escribir la frecuencia: Datos_Escritura_Frecuencia_Variador1

La frecuencia la escribimos en hexadecimal y en unidades de 0,01Hz. 5000 equivale a 50 Hz. 5000 en decimal que son 16#1388 (50 Hz)

Datos_Escritura_Frecuencia_Variador1[0]: =0;

Datos_Escritura_Frecuencia_Variador1[1]: =16#1388;

Lectura Frecuencia

1.- Bit que ejecuta: Lectura_Frecuencia_Variador1

2.- Variable donde leer la frecuencia: Datos_Lectura_Frecuencia_Variador1

La frecuencia la leeremos en la componente 0 del Array y en unidades de centésimas de Hz. Por ejemplo, si Variador1 está a 20 Hz

Datos_Lectura_Frecuencia_Variador1[0]: =2000;

Datos_ Lectura _Frecuencia_Variador1[1]: =0;

Escritura bit comando de operación

1.- Bit que ejecuta: Escritura_ComandoOperacion_Variador1

2.- Variable donde escribir el bit: Datos_Escritura_ComandoOperacion_Variador1

Si queremos Que el motor esté en marcha, escribimos en la anterior variable un BOOL TRUE. Si queremos que el motor se pare, escribimos un FALSE.

Escritura bit comando de sentido

1.- Bit que ejecuta: Escritura_ComandoSentido_Variador1

2.- Variable donde escribir el bit: Datos_Escritura_ComandoSentido_Variador1

Si queremos Que el motor gire en un sentido, escribimos en la anterior variable un BOOL TRUE. Si queremos que el motor gire en sentido contrario, escribimos un FALSE.