

Manual técnico



Modbus: RTU

Características Principales

Modbus RTU es un protocolo abierto, en serie (RS-232 o RS-485) derivado de la arquitectura Master / Slave. Es un protocolo ampliamente aceptado debido a su facilidad de uso y fiabilidad. Modbus RTU se usa ampliamente en Building Management Systems (BMS) y Industrial Automation Systems (IAS). Esta amplia aceptación se debe en gran parte a la facilidad de uso de MODBUS RTU.

Los mensajes MODBUS RTU son un CRC simple de 16 bits (suma de comprobación redundante cíclica). La simplicidad de estos mensajes es garantizar la fiabilidad. Debido a esta simplicidad, la estructura de registro MODBUS RTU de 16 bits se puede usar para empaquetar puntos flotantes, tablas, texto ASCII, colas y otros datos no relacionados.

Modbus: TCP

Características Principales

MODBUS TCP / IP es una variante de la familia MODBUS de protocolos de comunicación sencillos, de proveedor neutral, destinados a la supervisión y control de equipos de automatización. Específicamente, cubre el uso de la mensajería MODBUS en un entorno 'Intranet' o 'Internet' utilizando los protocolos TCP / IP. El uso más común de los protocolos en este momento es para la conexión Ethernet de PLC, módulos de E / S y 'puertas de enlace' a otros buses de campo simples o redes de E / S.

El protocolo MODBUS TCP / IP se está publicando como un estándar de automatización ('de facto'). Como MODBUS ya es ampliamente conocido, debería haber poca información en este documento que no pudiera obtenerse en otra parte. Sin embargo, se ha intentado aclarar qué funciones dentro de MODBUS tienen valor para la interoperabilidad de los equipos de automatización en general, y qué partes son "equipaje" del uso alternativo de MODBUS como protocolo de programación para PLC.

TBEN-PLC Package

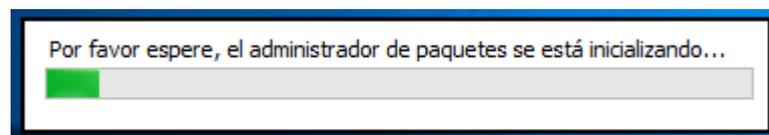
Para poder utilizar el TBEN-PLC de Turck en Codesys como dispositivo programable, deberemos instalar el Package de este el cual podemos descargar desde la web de Turck.

[Enlace de descarga](#)

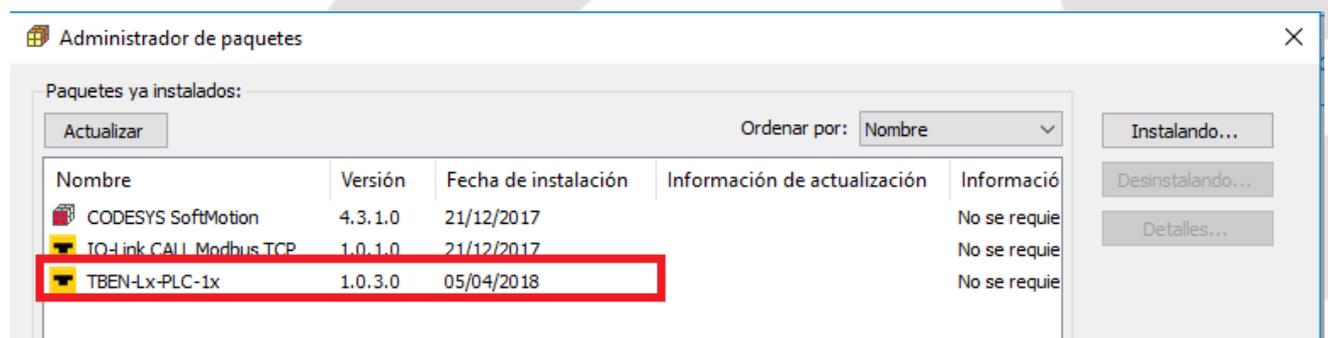
Dependiendo de la versión de Codesys que utilicemos, deberemos descargar uno u otro.

Configuration File	
➔ CODESYS Package TBEN-Lx-PLC Version v1.0.3.0 (for CODESYS v3.5.8.10 or higher)	15010 KB
➔ EDS files	60 KB
➔ GSDML files	11 KB
➔ CODESYS Package TBEN-Lx-PLC Version v1.1.4.0 (for CODESYS v3.5.12.10 or higher)	35840 KB

Una vez descargado el archivo, haremos doble clic sobre él y empezará la instalación.



Seleccionaremos la opción de instalación completa y esperaremos a que finalice, cuando haya terminado, deberá verse reflejado de la siguiente forma:

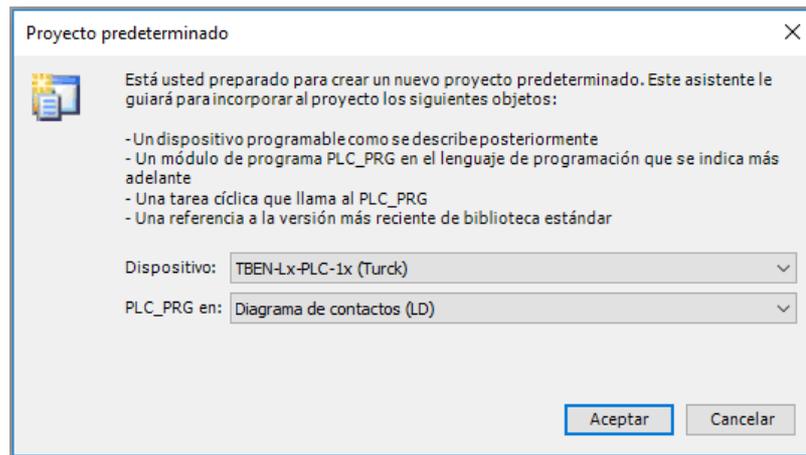


TBEN-PLC: Maestro Modbus RTU

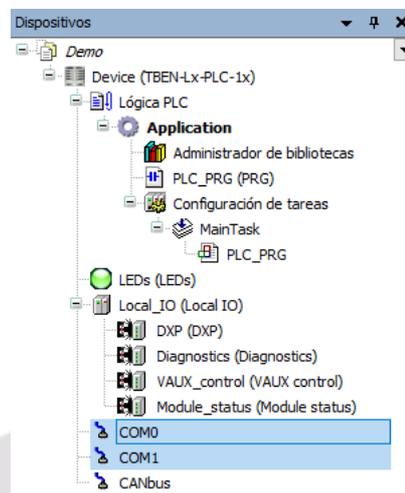
En este primer ejemplo, comunicaremos una [BL20-E-GW-RS-MB/ET](#) con el TBEN-L5-PLC-10, para que este sea el maestro Modbus RTU de la comunicación.

Recibiremos la información de la tarjeta de 16 entradas ([BL20-E-16DI-24VDC-P](#)) que conectaremos a la BL20 para procesarlas y verlas en Codesys.

Una vez instalado este package, podremos abrir Codesys para crear el proyecto con el TBEN-PLC como dispositivo programable.

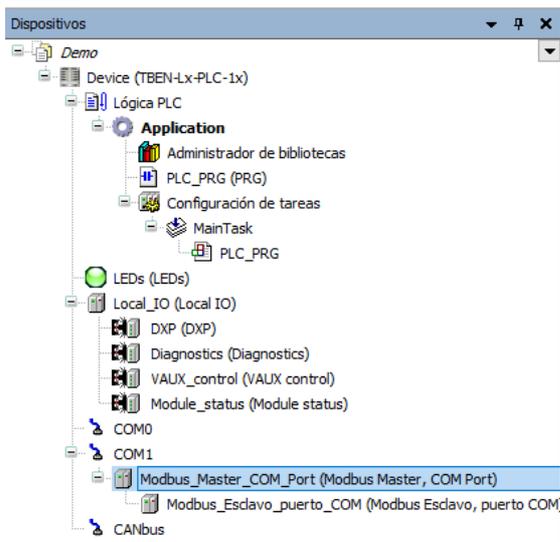


Y en la vista de Dispositivos, veremos como en el PCL tenemos los dos puertos serie (COM0 y COM1)



Será dentro de estos COM donde deberemos añadir los dispositivos que conectemos por Modbus RTU (RS232 o RS485) para configurar sus E/S.

En nuestro ejemplo, dentro del COM1, agregaremos la estructura de Maestro/Esclavo de Modbus RTU.



Ahora, con la BL20-RTU configurada por [Pactware](#) (Software gratuito de Turck) o dip-switch, haremos doble clic sobre COM1 en Codesys para configurar los parámetros de este puerto del TBEN.

Es de vital importancia que los parámetros de la BL20 coincidan con los del TBEN-PLC.

Parámetro	Tipo	Valor	Valor predeterminado	Unidad	Descripción
Baudrate	Enumeration of UDINT	9600 Bit/s	9600 Bit/s		Baudrate of the serial port.
DataBits	Enumeration of USINT	8 bit	8 bit		Number of data bits
StopBits	Enumeration of USINT	1 bit	1 bit		Number of stop bits
Termination active	Enumeration of USINT	no	no		Termination of the serial port (only in RS485 mode)
Biasing active	Enumeration of USINT	no	no		Biasing of the serial port (only in RS485 mode)
Power supply VALUX1	Enumeration of USINT	off	off		Power supply VALUX1
Swap A/B Line	Enumeration of USINT	no	no		Swap A and B lines of the serial port (only in RS485 mode)
Operation mode	Enumeration of USINT	RS232	RS232		Medium of the serial port.
Parity	Enumeration of USINT	none	none		Parity for messages on the serial port.

Para estar seguros de la configuración de la BL20, podemos utilizar el [manual de esta](#), o revisar la configuración con Pactware mediante el USB de la BL20.

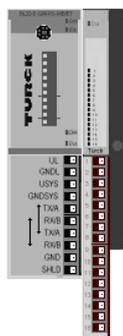
Name	Value
DTM (offline)	
Device identification	
Device short name	
Comment	
Settings	
Force Mode activation	with confirmation
Write reference list	yes
Notation	
Channel name	
Device	
special device properties	
Bus address	
Software revision	
Parameters	
Address (DIP switch)	0 (I/O-ASSISTANT mode)
Bit rate (DIP switch)	9.6 kbit/s
Interface (DIP switch)	RS485
Terminating resistor (DIP switch)	off
Address (I/O-ASSISTANT)	1
Bit rate (I/O-ASSISTANT)	9.6 kbit/s
Electrical interface (I/O-ASSISTANT)	RS485
Terminating resistor (I/O-ASSISTANT)	off
Parity	even
Mode	RTU
Modbus watchdog time	0

Para continuar con la configuración en Codesys, deberemos hacer doble clic sobre el dispositivo Modbus Esclavo que tenemos al final de los dispositivos de COM1, que es donde conectaremos la BL20-E-GW-RS-MB/ET.

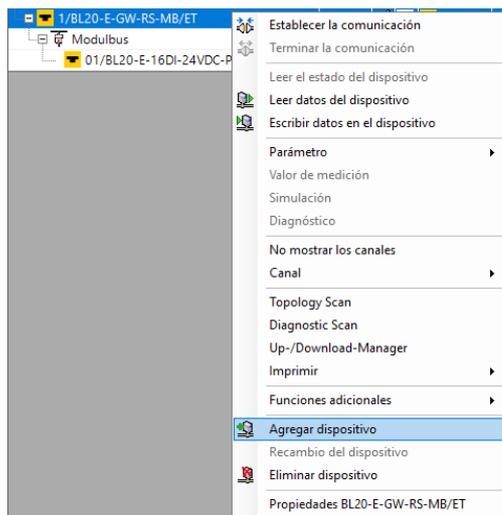


Una vez con la dirección asignada, falta declarar las variables E/S que comunicaremos con la BL20, esta información, deberemos extraerla desde el [Pactware](#).

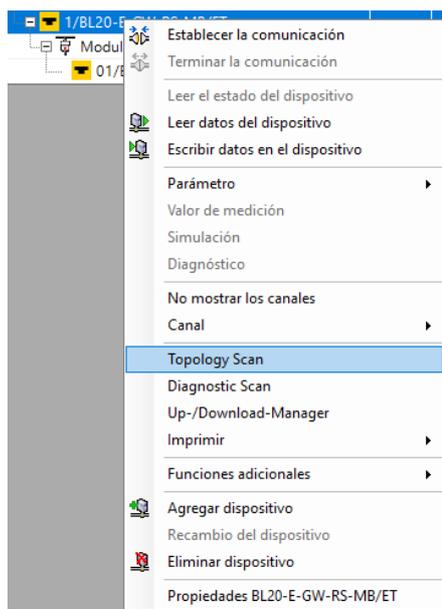
Configurando la BL20 con todas las tarjetas que acoplemos a la cabecera, deberemos extraer el “station report” para tener conocimiento de la cantidad de información que asociaremos en Codesys.



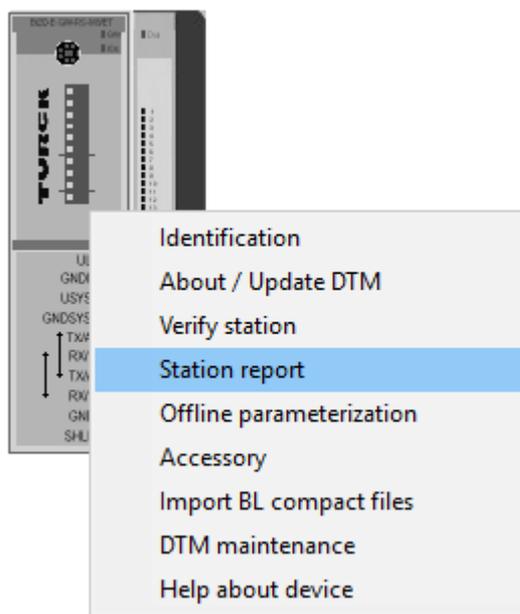
Para realizar dicha configuración, lo podemos hacer manualmente, añadiendo los módulos que acoplemos



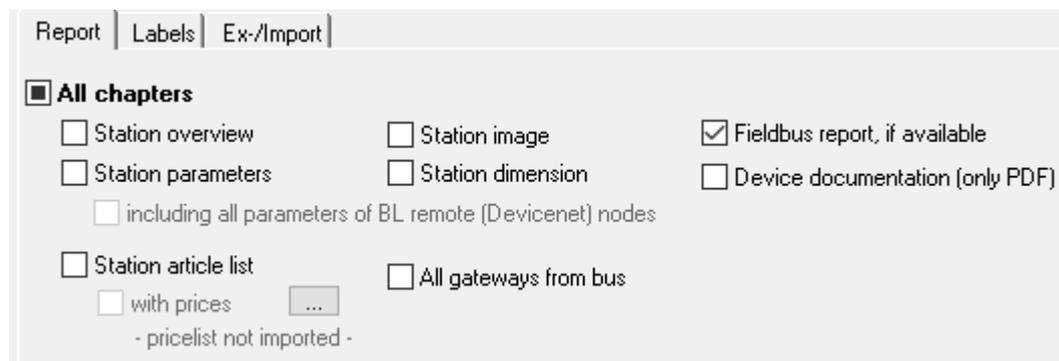
O mediante la función “Topology Scan”, la cual nos exigirá tener conectada la BL20 mediante el USB que podemos ver fácilmente en la cabecera.



Para extraer el “station report”, el cual nos dará información de los Bits, Bytes y Words que tendrá nuestra cabecera, junto con las tarjetas E/S que añadamos, haremos clic derecho sobre la cabecera RTU.



En la ventana que se abra, podremos seleccionar las opciones que queramos, aunque deberemos fijarnos en “Fieldbus report”



Este Fieldbus Report, nos dirá qué cantidad de Entradas y Salidas (físicas, diagnóstico, etc.) tendrá cada componente del Slot de la BL20, tanto cabecera como tarjetas E/S.

Al tener conectada únicamente una tarjeta [BL20-E-16DI-24VDC-P](#), obtenemos la siguiente información:

Station address: 1

Adr./Slot	Name	TAG	Data Size In	Data Size Out
0*	BL20-E-GW-RS-MB/ET	1/BL20-E-GW-RS-MB/ET	16 bit	0 bit
1	BL20-E-16DI-24VDC-P	01/BL20-E-16DI-24VDC-P	16 bit	0 bit
	Local I/O data incl. status/control		2 Words	0 Words
	Summarized diagnostics		1 Word	0 Words
Total size for in/out data rounded on full words			3 Words	0 Words

*For detailed information about status/control word see online help.

1.2. I/O map for input data

Register		Bit position															
Hex	Dec	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	0000	01.15	01.14	01.13	01.12	01.11	01.10	01.09	01.08	01.07	01.06	01.05	01.04	01.03	01.02	01.01	01.00
*0x0001	0001	GW.15	GW.14	GW.13	GW.12	GW.11	GW.10	GW.09	GW.08	GW.07	GW.06	GW.05	GW.04	GW.03	GW.02	GW.01	GW.00
**0x0002	0002	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	M00

Description: 1.Column=Register address, n. Column=Modul number.bitposition

*) GW: gateway status-/diagnostics bits

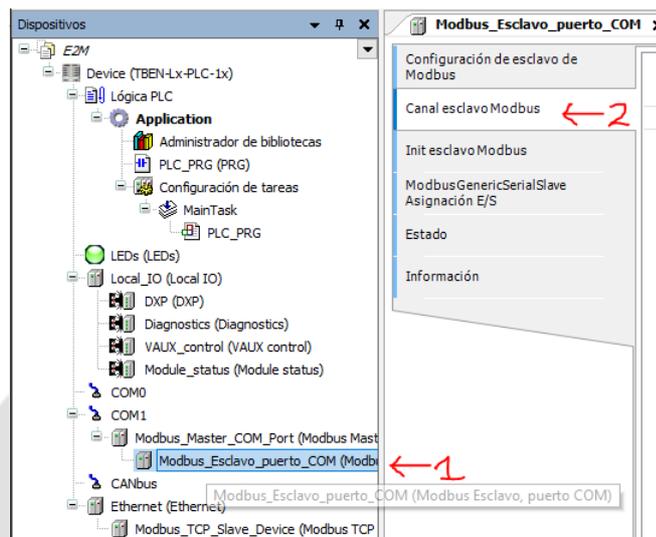
**) M: module diagnostics (1 bit for each module)

Process input data: 3 Words

- Slot 0: Cabecera BL20: Bits reservados (puede ampliar información en el manual del dispositivo)
- Slot 1: Tarjeta 16 Entradas Digitales de 24 Volteos DC.

Estos dos Slots (0 y 1) suman un total de 2Words, una para cada Slot, se añade, además una Word extra de diagnósticos por cada tarjeta de E/S que agreguemos a la cabecera.

Con estas longitudes de variables, crearemos los canales de comunicación. Hay que tener en cuenta, que los diagnósticos del módulo no se añaden a la configuración en Codesys, para ello, abriremos el esclavo modbus RTU de Codesys y el menú de Canales de esclavos.



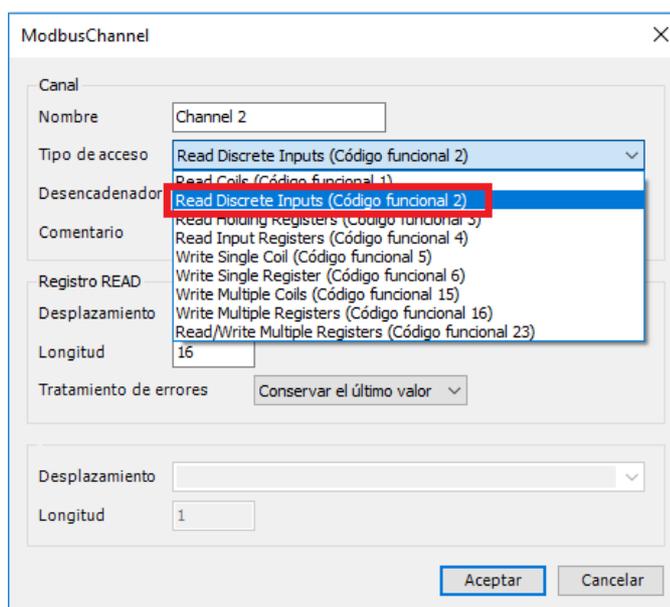
En la ventana que se abrirá, deberemos crear los canales de comunicación necesarios que, según el “Station Report” que hemos visto anteriormente, son dos (uno por cada Word) si no tenemos en cuenta la Word destinada a diagnósticos.

Deberemos empezar agregando un canal con los botones que veremos debajo a la derecha.

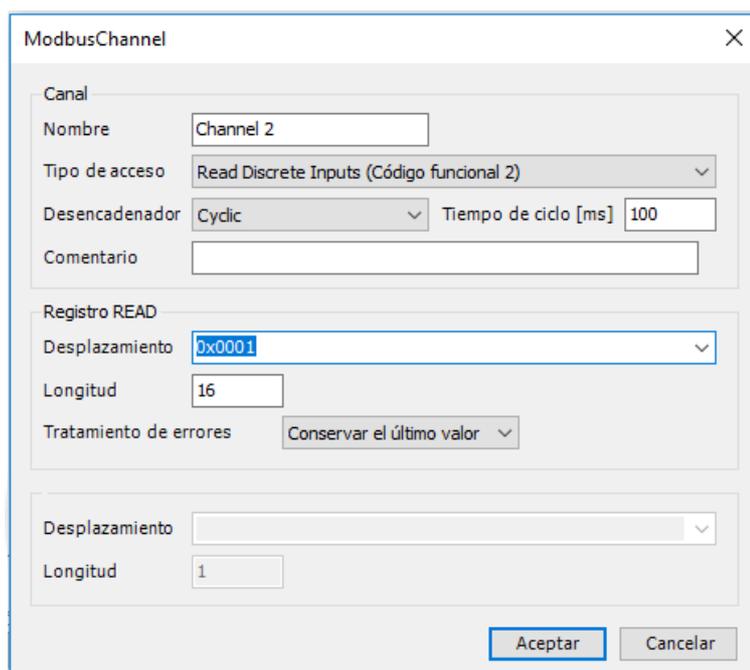


En la ventana que se abra, será donde configuraremos los parámetros del canal de comunicación, al ser el primero, no tendrá desplazamiento inicial (offset) y la longitud, al tratarse de una Word será de 16 bits.

El FC que soporta este tipo de comunicación será el tipo 2.

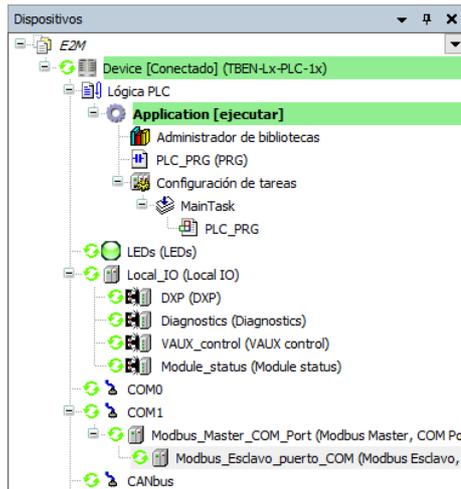


Añadiremos tantos canales como variables tengamos, añadiendo +1 el desplazamiento (0x0001)



Nombre	Tipo de acceso	Desencadenador	Desplazamiento READ	Longitud	Tratamiento de errores	Desplazamiento WRITE	Longitud	Comentario
0 Channel 0	Read Discrete Inputs (Código funcional 02)	Cyclic, t#100ms	16#0000	16	Conservar el último valor			
1 Channel 1	Read Discrete Inputs (Código funcional 02)	Cyclic, t#100ms	16#0001	16	Conservar el último valor			

Ahora sí que podremos descargar la configuración del proyecto y, si hemos seguido y configurado debidamente la guía, tendremos la comunicación funcionando.

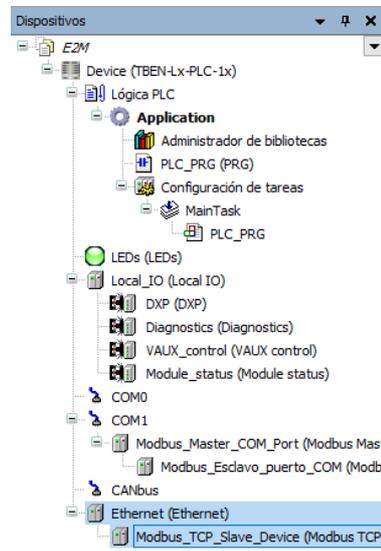


Variable	Asignación	Canal	Dirección	Tipo	Valor actual	Valor nuevo	Unidad	Descripción
in2		Channel 0	%IB28	ARRAY [0..1] OF BYTE				Read Discrete Inputs
in3		Channel 1	%IB30	ARRAY [0..1] OF BYTE				Read Discrete Inputs
in4		Channel 1[0]	%IB30	BYTE	2			Read Discrete Inputs
in5		BT0	%IX30.0	BOOL	FALSE			0x0001
in6		BT1	%IX30.1	BOOL	TRUE			0x0002
in7		BT2	%IX30.2	BOOL	FALSE			0x0003
in8		BT3	%IX30.3	BOOL	FALSE			0x0004
in9		BT4	%IX30.4	BOOL	FALSE			0x0005
in10		BT5	%IX30.5	BOOL	FALSE			0x0006
in11		BT6	%IX30.6	BOOL	FALSE			0x0007
in12		BT7	%IX30.7	BOOL	FALSE			0x0008
in13		Channel 1[1]	%IB31	BYTE	0			Read Discrete Inputs

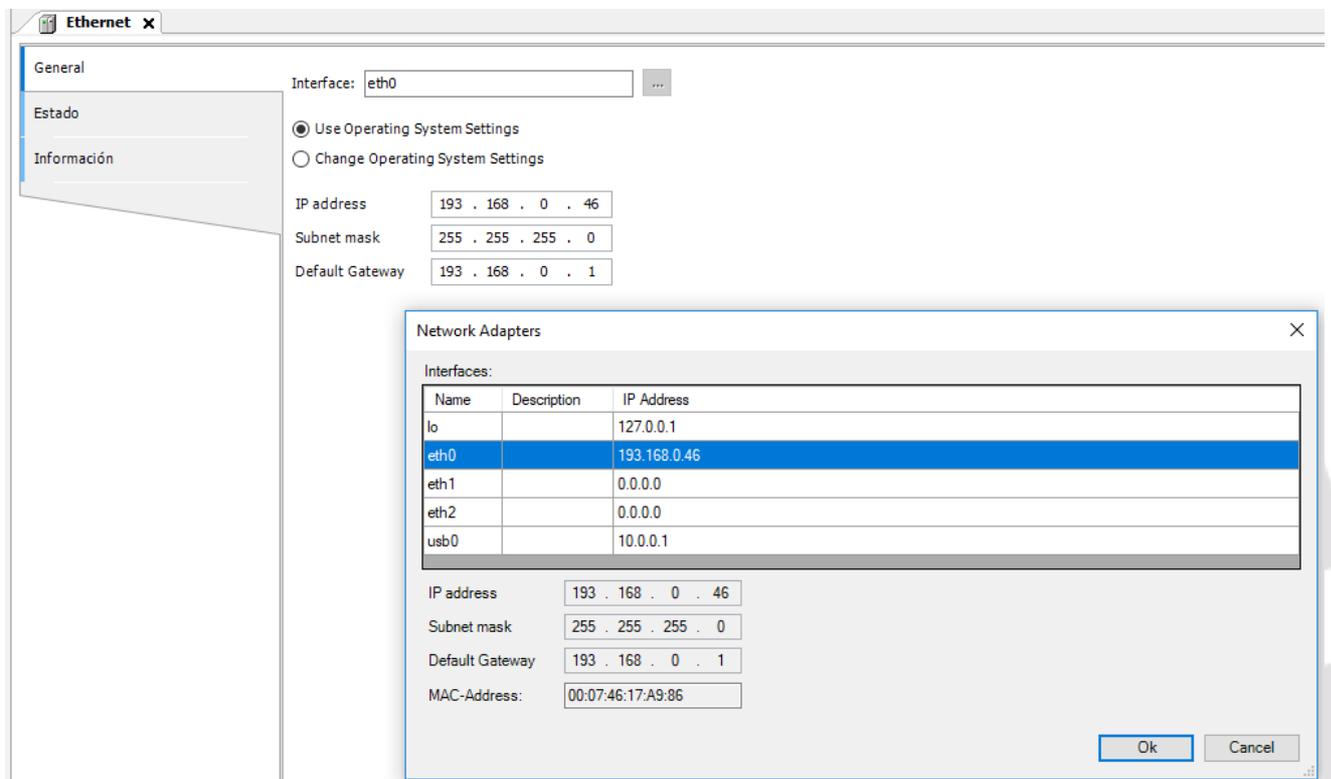
TBEN-PLC: Esclavo Modbus TCP

En este segundo ejemplo, comunicaremos mediante Modbus TCP el dispositivo TBEN-PLC y la pantalla IX T7A de Beijer, donde el PLC será el esclavo de la comunicación y el panel HMI será el maestro.

En primer lugar, desde Codesys deberemos generar la estructura de **esclavo Modbus TCP** de la comunicación.

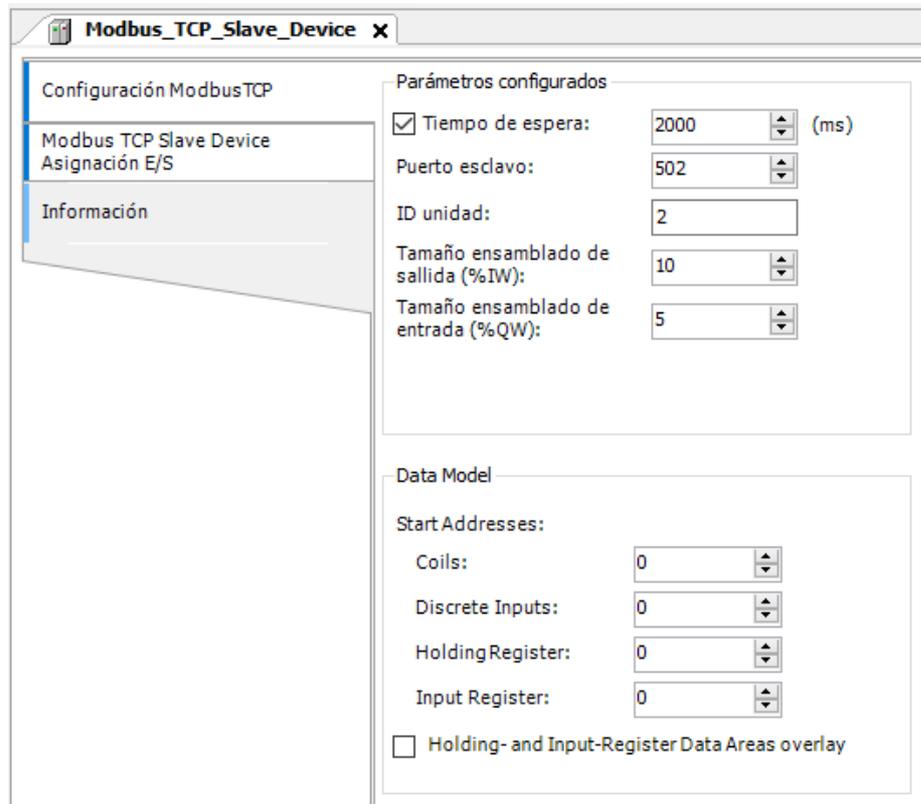


Haciendo doble clic sobre el dispositivo “Ethernet (Ethernet)”, pasaremos a configurar la interfaz que conecte el TBEN-PLC con la pantalla, para ello únicamente deberemos seleccionar la interfaz con la IP que hayamos asignado al configurar el módulo.

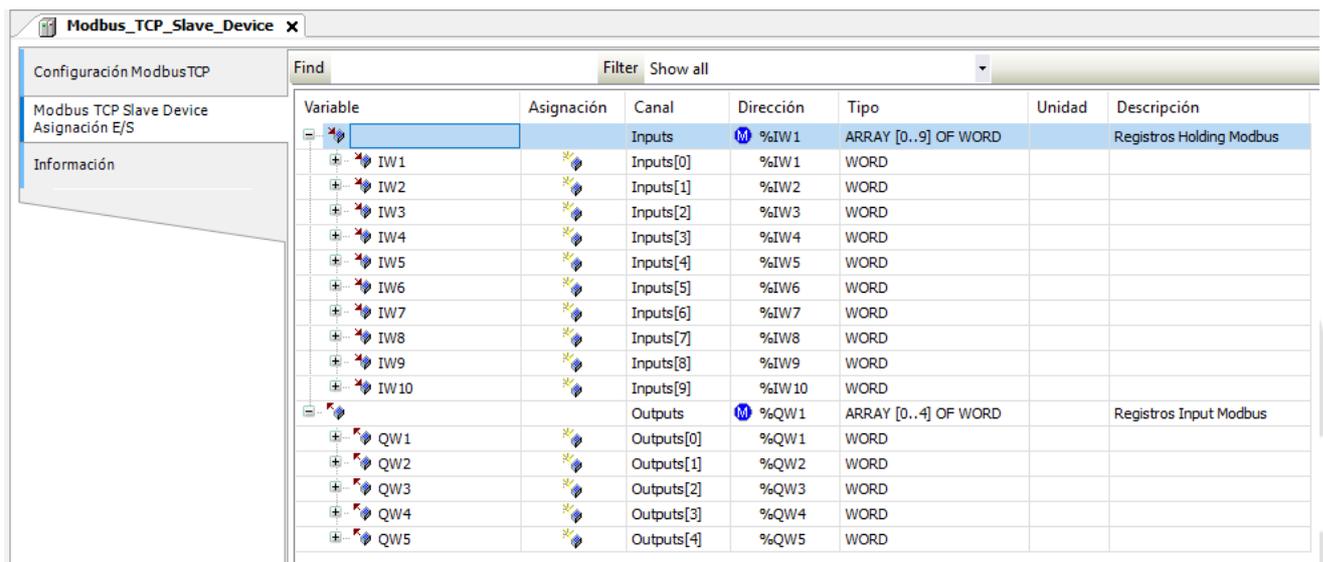


Seguidamente, haremos doble clic sobre “Modbus_TCP_Slave_Device” y pasaremos a configurar el ID de unidad y la longitud/cantidad de E/S que comunicaremos entre PLC y HMI.

El direccionamiento de E/S que genere y podamos Modificar en Codesys, no influirá en los valores de comunicación sobre IX Developer que veremos a continuación.



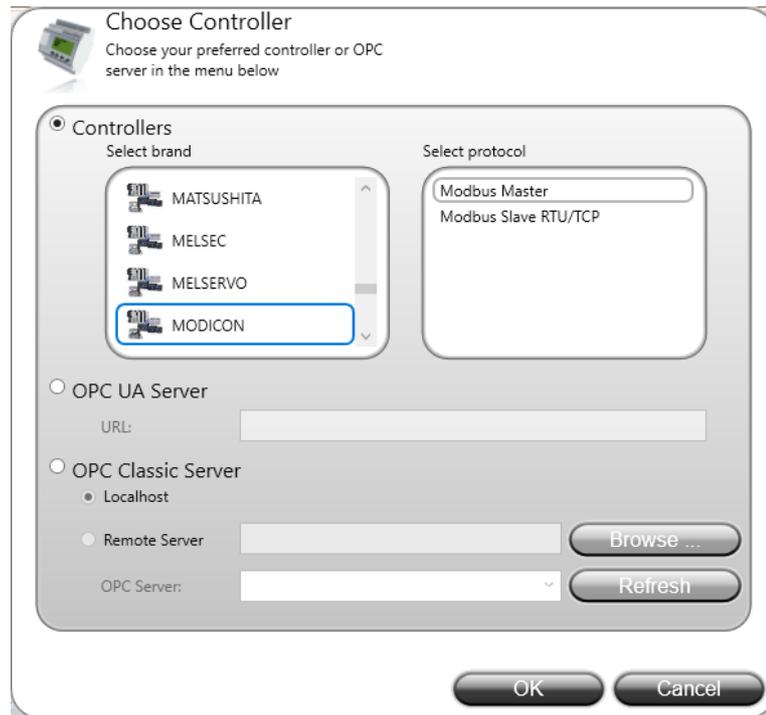
En el menú izquierdo, accederemos a “Modbus TCP Slave Device Asignación E/S” para modificar el direccionamiento y el “tag” asignado a estas E/S que hemos definido anteriormente.



- M: Modificadas.

HMI Beijer: Maestro Modbus TCP

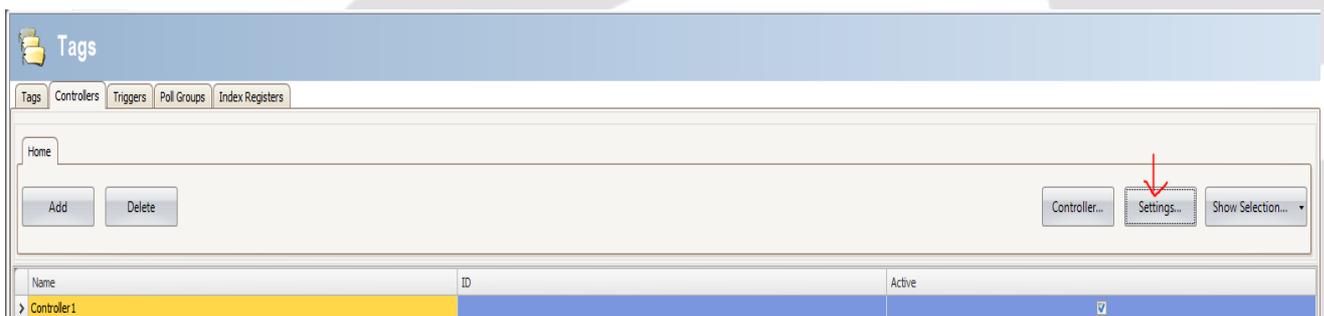
Al crear el proyecto en IX Developer, seleccionaremos el Driver de MODICON, Modbus Master.



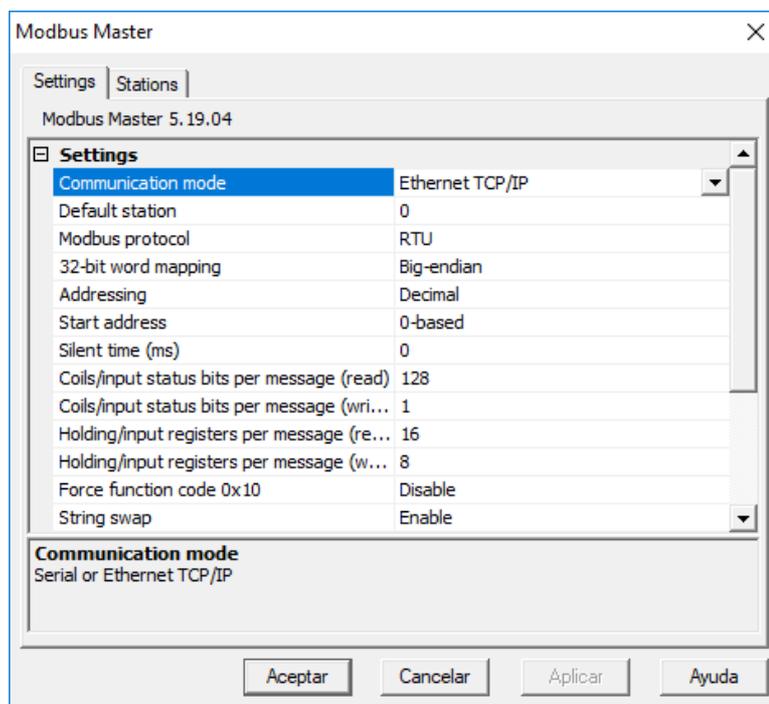
Para configurarlo, entraremos en Tags, dentro del menú de Functions.



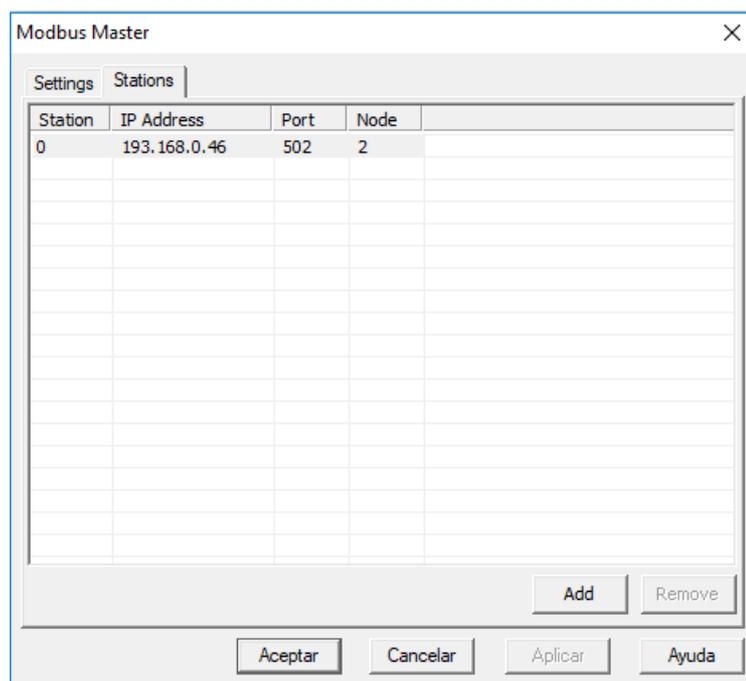
Dentro de Tags, iremos a la pestaña de Controladores y sobre el controlador1, que será el Modbus Master, haremos clic sobre el botón de Settings, para editar los parámetros de comunicación.



Deberemos configurar el método de comunicación que utilizará el Driver, puede ser Serie o EtherNet/IP.



En el menú Stations, asignaremos la dirección IP del TBEN-PLC que será con la cual comunique el HMI.



Direccionamiento de las variables en IX Developer.

Volviendo al direccionamiento de variables que enviará y recibirá el TBEN-PLC, las cuales hemos configurado desde Codesys, veremos qué tipo de variables son y en función de esto, asignaremos las comunicaciones en IX Developer.

Variable	Asignación	Canal	Dirección	Tipo	Unidad	Descripción
		Inputs	%IW1	ARRAY [0..9] OF WORD		Registros Holding Modbus
		Outputs	%QW1	ARRAY [0..4] OF WORD		Registros Input Modbus

1. Registros Holding Modbus (IW en Codesys)
2. Registros Input Modbus (QW en Codesys)

Registros Holding Modbus en IX Developer se inician en las direcciones: 40000

Registros Input Modbus en IX Developer se inician en las direcciones: 30000

Teniendo nociones de este direccionamiento, deberemos asignar de forma incremental el valor a los Tags que vayamos a utilizar, respetando el número total de IW y QW que hayamos definido en Codesys.

Controllers	
Data Type	Controller 1
INT 16	40006
INT 16	40007
INT 16	40008
INT 16	40009
INT 16	40005
INT 16	40001
INT 16	40002
INT 16	40003
INT 16	40004
INT 16	
INT 16	30000
INT 16	30001
INT 16	30002