

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

Desarrollo e implementación de una red de datos
basada en Modbus y Ethernet para autómatas
industriales

Autor: Marina Hernández Tinoco

Tutor: Daniel Limón Marruedo

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Desarrollo e implementación de una red de datos basada en Modbus y Ethernet para autómatas industriales

Autor:

Marina Hernández Tinoco

Tutor:

Daniel Limón Marruedo

Profesor titular

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Proyecto Fin de Grado: **Desarrollo e implementación de una red de datos basada en Modbus y Ethernet para autómatas industriales**

Autor: Marina Hernández Tinoco

Tutor: Daniel Limón Marruedo

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

*A mi familia,
mis amigos,
mis maestros.*

Agradecimientos

En primer lugar agradecer a Daniel Limón la oportunidad que me ha brindado para realizar este proyecto y aprender de él.

A mis padres y a mi hermano, por darme la oportunidad de estudiar esta carrera y ofrecerme todo su apoyo para que pudiera conseguir mis objetivos.

A mis amigas, las que me acompañan desde el inicio de esta aventura y las que he tenido el placer de conocer a lo largo de esta etapa, con personas como ellas todo se hace más fácil.

A Sergio, gracias a su apoyo incondicional, tantas horas invertidas juntos en la Escuela y de aguantarnos te hace merecedor de aparecer aquí.

A Juan Carlos, por sus ánimos y por acompañarme en este camino.

Marina Hernández Tinoco

Sevilla, 2016

Resumen

El proyecto consiste en el desarrollo y la implementación de una red de datos basada en Modbus y Ethernet para el control en tiempo real mediante autómatas industriales

En él se estudian los protocolos utilizados, las opciones de comunicaciones que ofrece el autómata Modicon M340, la configuración y como se realiza una puesta en marcha.

Concluye con el desarrollo de una aplicación práctica en el que se puede validar lo estudiado mediante el control en tiempo real de dos plantas emuladas en el ITS-PLC que cooperan entre sí para poder realizar el proceso industrial.

Agradecimientos	ix
Resumen	xi
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvi
1 Introducción	1
1.1 <i>Objetivo</i>	1
1.2 <i>Alcance</i>	1
2 Entorno de trabajo	3
2.1 <i>Hardware</i>	3
2.1.1 PLC	3
2.2 <i>Software</i>	4
2.2.1 Unity Pro	4
2.2.2 Simulador ITS	5
3 Comunicaciones PLC vía Modbus	7
3.1 <i>Protocolo de comunicaciones</i>	7
3.2 <i>Funcionamiento y elementos</i>	8
3.2.1 Estructura de la trama Modbus	9
3.3 <i>Cableado</i>	13
3.3.1 Comunicación serie	13
3.3.2 Normas de comunicación en series	13
3.3.3 Estándar RS-485	14
3.3.4 Puerto serie del PLC M340.	16
3.4 <i>Configuración del PLC</i>	17
3.5 <i>Tiempo de intercambio de una palabra</i>	20
4 Comunicaciones PLC vía Ethernet	21
4.1 <i>Modbus sobre Ethernet</i>	22
4.2 <i>¿Por qué combinar Modbus con Ethernet?</i>	24
4.2.1 Ventajas	24
4.3 <i>Cableado</i>	25
4.3.1 Topología de la red	25
4.3.2 Tipos de cables.	27
4.3.3 Puerto Ethernet del PLC	30
4.4 <i>Configuración del PL</i>	31
5 Programación del PLC	33
5.1 <i>Distribución de la memoria</i>	33
5.2 <i>Funciones</i>	35
5.2.1 Gestión de direcciones	37
5.2.2 Lectura de valores	38

5.2.3	Escritura de valores	39
5.2.4	Función DATA_EXCH	41
5.3	<i>Tabla de gestión</i>	42
5.4	<i>Acceso a las características de un canal de E/S</i>	44
5.4.1	Acceso a las características del protocolo Modbus	44
5.4.2	Acceso a las características del protocolo Ethernet	46
5.5	<i>Información de depuración de comunicación</i>	46
5.5.1	Depurar comunicación Modbus	46
5.5.2	Depurar comunicación Ethernet	47
5.6	<i>Ejemplo comunicación Modbus</i>	48
5.7	<i>Ejemplo comunicación Ethernet</i>	51
6	Implementación de una red industrial para laboratorio	53
6.1	<i>Objetivo de la red industrial</i>	53
6.1.1	Sistemas industriales de control	53
6.1.2	La pirámide CIM	54
6.2	<i>Planteamiento de la red industrial en el laboratorio</i>	56
6.2.1	Escenario	57
6.3	<i>Diseño del problema</i>	58
6.3.1	Planta clasificadora	58
6.3.2	Planta mezcladora de pintura	59
6.3.3	Visión general del escenario	60
6.4	<i>Desarrollo de la solución.</i>	62
6.4.1	Declaración de variables	62
6.5	<i>Especificaciones de cableado del laboratorio</i>	65
6.5.1	RS-485. Polarización y terminación de la línea Modbus	65
6.5.2	Montaje cableado Ethernet.	67
7	Conclusiones y líneas futuras de trabajo	69
7.1	<i>Conclusiones</i>	69
7.2	<i>Líneas futuras de trabajo</i>	69
8	Bibliografía	71
9	ANEXO I: Funciones Modbus	73
9.1	<i>Direcciones</i>	74
9.2	<i>Función 1 o 2:</i>	75
9.3	<i>Función 3 o 4</i>	75
9.4	<i>Función 5</i>	76
9.5	<i>Función 6</i>	77
9.6	<i>Función 7</i>	77
9.7	<i>Función 15</i>	78
9.8	<i>Función 16</i>	78
9.9	<i>Función 20</i>	79
9.10	<i>Función 21</i>	80
10	ANEXO II: Norma IEEE 802	83
10.1	802.1D	85
10.2	802.2	85
10.3	802.3	85
11	ANEXO III: RFC 826	87

ÍNDICE DE TABLAS

Tabla 3-1 Funciones Modbus de lectura.	11
Tabla 3-2 Funciones Modbus de escritura.	11
Tabla 3-3 Trama Query.	12
Tabla 3-4 Trama Response.	12
Tabla 3-5 Tiempo intercambio de una palabra en Modbus.	20
Tabla 4-1 Velocidad y tiempo en una red Ethernet.	26
Tabla 4-2 Código errores puerto Ethernet.	30
Tabla 5-1 Código de Funciones Modbus equivalentes en UnityPro.	35
Tabla 5-2 Parámetros función ADDM (Modbus).	37
Tabla 5-3 Parámetros de la función READ_VAR.	38
Tabla 5-4 Parámetros de la función WRITE_VAR.	40
Tabla 5-5 Parámetros función DATA_EXCH.	41
Tabla 5-6 Tabla de los valores del informe de operación.	43
Tabla 5-7 Tabla de informe de operación I.	43
Tabla 5-8 Tabla de informe de operación II.	44
Tabla 6-1 Sensores planta clasificadora.	62
Tabla 6-2 Actuadores planta clasificadora.	63
Tabla 6-3 Sensores planta procesamiento por lotes.	64
Tabla 6-4 Actuadores planta procesamiento por lotes.	64
Tabla 10-1 Relación de normas IEEE.	84

ÍNDICE DE FIGURAS

Figura 2-1. PLC Modicon M340.	4
Figura 2-2 Intercambio de información entre los sistemas, la tarjeta DAQ y el PLC.	6
Figura 2-3 Conexiones eléctricas entre la tarjeta DAQ y el PLC	6
Figura 3-1 Interacción Maestro – Esclavo	8
Figura 3-2 Tipos de direcciones Modbus.	8
Figura 3-3 Modo unicast.	9
Figura 3-4 Modo broadcast.	9
Figura 3-5 Trama Modbus.	10
Figura 3-6 Bit de secuencia modo RTU con paridad.	13
Figura 3-7 Diagrama de bloques con bucle de tensión.	14
Figura 3-8 Interfaz RS485 de 16 Puerto de Alto Rendimiento PCI Express.	15
Figura 3-9 Puerto serie en el PLC M340.	16
Figura 3-10 Configuración del bastidor.	17
Figura 3-11 Configuración del puerto serie.	17
Figura 3-12 Configuración del cliente Modbus.	19
Figura 4-1 Paquete Modbus TCP.	23
Figura 4-2 Trama Modbus TCP/IP.	24
Figura 4-3 Topología en bus para la red Ethernet.	25
Figura 4-4 Topología en estrella para la red Ethernet.	26
Figura 4-5 Cable par trenzado sin apantallar.	28
Figura 4-6 Código de colores conector RJ45.	28
Figura 4-7 Cable STP. Par trenzado apantallado.	29
Figura 4-8 Cable FTP. Par trenzado con pantalla global.	29
Figura 4-9 Puerto RJ45 Ethernet.	30
Figura 4-10 Crear una nueva red Ethernet	31
Figura 4-11 Pantalla configuración IP.	32
Figura 4-12 Configuración del puerto Ethernet.	32
Figura 5-1 Opciones visualizar memoria P34 2020.	33
Figura 5-2 Memoria principal.	34
Figura 5-3 Declarar variables.	34
Figura 5-4 Pantalla para permitir matrices dinámicas.	36
Figura 5-5 Sintaxis de la función ADDM (Modbus).	37

Figura 5-6 Función ADDM (Ethernet).	37
Figura 5-7 Sintaxis de la función READ_VAR.	38
Figura 5-8 Función READ_VAR (Ethernet).	39
Figura 5-9 Sintaxis de la función WRITE_VAR.	39
Figura 5-10 Función WRITE_VAR (Ethernet).	40
Figura 5-11 Sintaxis de la función DATA_EXCH	41
Figura 5-12 Función WRITE_VAR (Ethernet).	42
Figura 5-13 Figura de los campos de la variable port en Modbus.	45
Figura 5-14 Contadores para depuración Modbus.	46
Figura 5-15 Pantalla depuración comunicación Ethernet.	47
Figura 5-16 Depuración Ethernet, tráfico de mensajes.	48
Figura 5-17 Variables básicas en el maestro para comunicación Modbus.	50
Figura 5-18 Programación Maestro en una comunicación Modbus.	51
Figura 5-19 Programación PLC en una comunicación Ethernet.	52
Figura 6-1 Pirámide CIM de comunicación.	55
Figura 6-2 Planta clasificadora de cajas.	57
Figura 6-3 Planta mezcladora de pintura.	57
Figura 6-4 Escenario de la aplicación práctica.	58
Figura 6-5 Esquemático de la planta transportadora.	59
Figura 6-6 Esquemático procesamiento por lotes.	60
Figura 6-7 Diagrama de secuencia.	61
Figura 6-8 Asignar valor ITS en la memoria principal.	65
Figura 6-9 Función INT_TO_BOOL.	65
Figura 6-10 Conector RJ-45.	66
Figura 6-11 Cableado de los interfaces serie con R485.	67
Figura 6-12 Montaje cable bus Modbus RS-485.	67
Figura 10-1 Capas del Modelo OSI.	83
Figura 11-1 ARP Request.	87
Figura 11-2 ARP Reply.	87

1 INTRODUCCIÓN

El presente proyecto surge de la necesidad de controlar los procesos a tiempo real en las plantas industriales aprovechando las herramientas de comunicación de las que disponemos. A día de hoy es primordial en la industria poder revisar la productividad y minimizar los errores en cualquier tipo de sistema gestionado por autómatas programables.

Las comunicaciones industriales son aquellas que permiten el flujo de información del controlador a los diferentes dispositivos a lo largo del proceso de producción: detectores, actuadores, sensores entre otros.

Disponemos de distintos protocolos para establecerlas pudiendo detectar anomalías y cooperar en la resolución del problema, la posibilidad de utilizar sistemas SCADA y controlarlas de forma gráfica.

1.1 Objetivo

El objetivo del proyecto es el desarrollo e implementación de una red de datos basada en Modbus y Ethernet para el control en tiempo real mediante autómatas industriales.

Se ha desarrollado en el laboratorio de robótica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Sevilla.

Para concluir se realiza una aplicación práctica de lo estudiado donde podremos observar cómo llevar a cabo el control a tiempo real en un mismo sistema con diferentes autómatas interactuando entre ellos.

1.2 Alcance

El alcance del proyecto consiste en el diseño e implementación de una red de comunicaciones para 12 PLCs de los que dispone el laboratorio y su validación mediante el control en tiempo real de una planta emulada en ITS-PLC.

Se analizará las posibilidades de seguir ampliando dicho Proyecto ya que como se mencionó anteriormente poder comunicar los autómatas abre el campo de desarrollo.

2 ENTORNO DE TRABAJO

El presente proyecto ha sido desarrollado en el Laboratorio de Robótica del Departamento de Ingeniería de Sistemas y Automática.

Aprovechando los recursos que nos ofrecía hemos montado una red de comunicaciones entre los 12 autómatas de los que disponemos. A continuación, detallaremos el hardware y software utilizados.

2.1 Hardware

Nuestro escenario está compuesto por 12 PLCs de la marca Schneider. Dispuestos en 4 filas con 3 autómatas cada una.

2.1.1 PLC

2.1.1.1 Introducción

PLC, Controlador Lógico Programable, es una computadora utilizada en la ingeniería automática, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas.

A diferencia de las computadoras de propósito general, el PLC está diseñado para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto. Los programas para el control de funcionamiento de la máquina se suelen almacenar en baterías, copia de seguridad o en memorias no volátiles. Un PLC es un ejemplo de un sistema de tiempo real, donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no producirá el resultado deseado.

2.1.1.2 Modelo

En este Proyecto hemos trabajado con el modelo Modicon M340 de Schneider mostrado en la Figura 2-2.

Compuesto por los siguientes módulos conectados de izquierda a derecha:

- BMX CPS 2000: módulo de alimentación diseñados para alimentar cada rack BMX y sus módulos.
- BMX P34 2020: procesador con puertos USB, serie, Ethernet y tarjeta de memoria.
- BMX DDM 3202K: módulo de adquisición de datos del ITS¹.
- BMX AMI 0410: módulo de entradas analógicas.
- BMX DDM 16022: módulo mixto de entradas/salidas estáticas.

¹ ITS: Sistema de capacitación interactiva para PLCs.

La numeración de la Figura 2-2 hace referencia a:

1. Puerto de protección de la tarjeta de memoria.
2. Puerto USB.
3. Puerto Ethernet.
4. Puerto serie.



Figura 2-1. PLC Modicon M340.

A lo largo de la memoria se explicará cómo comunicar los PLCs sacando el mayor aprovechamiento de estos.

2.2 Software

Para el desarrollo de nuestro proyecto hemos utilizado el software propio de los PLCs y un simulador para poder realizar un ejemplo práctico de todo lo estudiado.

2.2.1 Unity Pro

Unity Pro es el software de programación, depuración y operación común para las Modicon.

Para crear el programa de usuario proporciona los siguientes lenguajes de programación:

- Lenguaje de contactos (LD)
- Lenguaje de bloques funcionales (FBD)
- Lista de instrucciones (IL)
- Literal estructurado (ST)
- Diagrama funcional en secuencia (SFC)



gamas

Todos estos lenguajes de programación pueden utilizarse juntos en el mismo proyecto. [1]

El programa nos ofrece grandes posibilidades como:

- Creación de variables desde los módulos de entrada/salida.
- Configurar dirección IP del PLC.
- Editar datos.
- Crear variables.
- Biblioteca de funciones y librerías.

En la bibliografía podrá acceder a toda la configuración detalladamente, cuando hagamos uso de ellas se especificará como realizar lo necesario para nuestra aplicación.

2.2.2 Simulador ITS

ITS PLC Professional Edition – Sistema de Capacitación Interactiva para PLCs – es una herramienta educativa y de capacitación para la programación de PLCs.

Con base en la tecnología informática más reciente, ITS PLC hace que la capacitación sobre PLCs resulte fácil y amena

El resultado, es un ambiente virtual simulado que ofrece sistemas de capacitación altamente realistas, ofreciendo avanzados gráficos tridimensionales en tiempo real y una interactividad total, además los ambientes virtuales, nos evitan riesgo de lesiones para el personal, como daños a los equipos involucrados.

ITS PLC ofrece problemas con niveles incrementales de dificultad, que permite pasar a ejercicios más avanzados. [2]

2.2.2.1 Funcionamiento

ITS PLC ofrece cinco sistemas virtuales con fines educativos y de capacitación para la programación de PLCs.

Cada sistema es una simulación visual de un sistema industrial que incluye actuadores y sensores virtuales, de modo que su estado pueda ser detectado por el PLC.

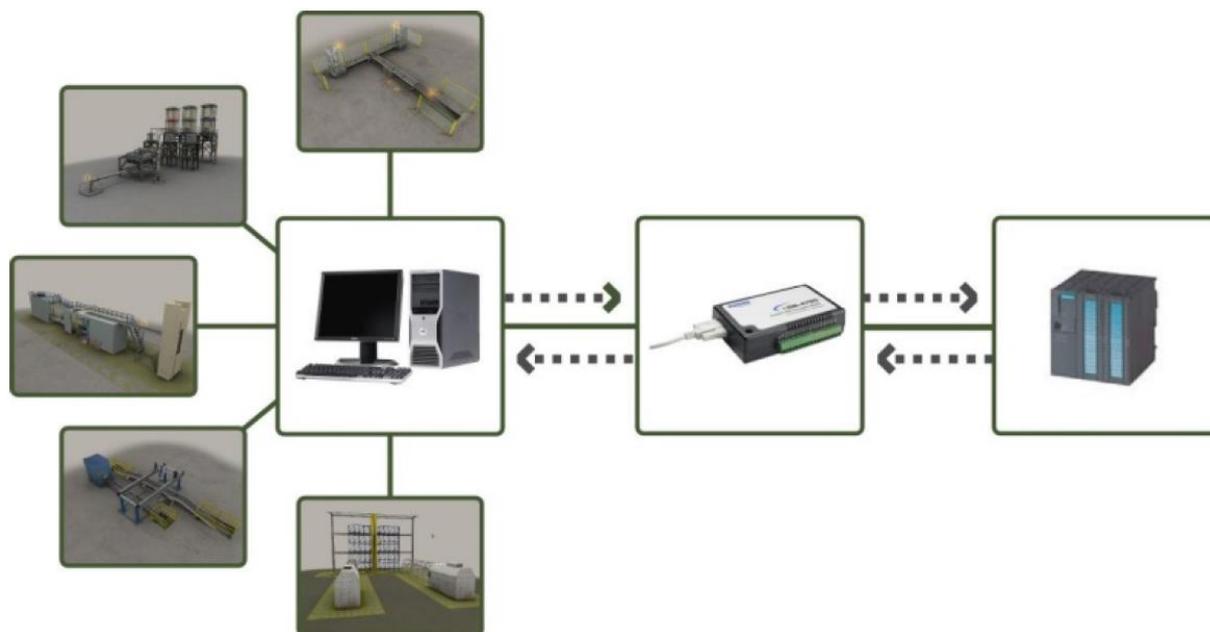


Figura 2-2 Intercambio de información entre los sistemas, la tarjeta DAQ y el PLC.

El objetivo es programar el PLC de modo que controle cada sistema virtual como si se tratara de un sistema real.

Como podemos ver en la Figura 2-3 la información es intercambiada entre el PLC y el sistema virtual a través de una tarjeta de adquisición de datos (DAQ) con 32 canales de E/S aislados y una interfaz USB.

En la Figura 2-4 se observan las conexiones de la tarjeta DAQ y el PLC.



Figura 2-3 Conexiones eléctricas entre la tarjeta DAQ y el PLC

3 COMUNICACIONES PLC VÍA MODBUS

Modbus es un protocolo de comunicación serie de nivel 7 del modelo OSI². Desarrollado y publicado por Modicon en 1979. En su origen el uso de Modbus estaba orientado exclusivamente al mundo de los controladores lógicos programables o PLCs de Modicon. [3]

A día de hoy, el protocolo Modbus es el protocolo de comunicaciones más común utilizado en entornos industriales, sistemas de telecontrol y monitorización. Lo que implica de forma implícita que: tanto a nivel local como a nivel de red, en su versión TCP/IP, seguirá siendo uno de los protocolos de referencia en las llamadas Smart Grids³, redes de sensores, telecontrol y un largo etc. de sistemas de información que ya empiezan a asomar la cabeza en nuestro día a día.

3.1 Protocolo de comunicaciones

El objeto del protocolo Modbus es bien sencillo: La transmisión de información entre distintos equipos electrónicos conectados a un mismo bus. Existiendo en dicho bus un solo dispositivo maestro (Maestro) y varios equipos esclavos (Esclavos) conectados.

En su origen estaba orientado a una conectividad a través de líneas serie como pueden ser RS-232 o RS-485, pero con el paso del tiempo han aparecido variantes como Modbus TCP, que permite el encapsulamiento de Modbus serie en tramas Ethernet TCP/IP de forma sencilla. Esto sucede porque desde un punto de vista de la torre OSI, el protocolo Modbus se ubica en la capa de aplicación.

El hecho de que se haya extendido su uso hasta convertirse en el protocolo más estandarizado en el sector industrial se debe a varias razones diferenciales respecto a otros protocolos:

- La ventaja fundamental de Modbus es que puede funcionar sobre prácticamente todos los medios de comunicación, incluyendo cables de par trenzado, sin cable, fibra óptica, Ethernet, módems telefónicos, teléfonos celulares y microondas, lo cual implicaría que una conexión Modbus podría implantarse fácilmente en instalaciones de vieja construcción.
- El estándar Modbus es público, lo que permite a los fabricantes desarrollar dispositivos tanto Maestro como Esclavo sin gastos aplicados al protocolo. Este hecho facilita el acceso a la información y estructura del protocolo que, además, es muy básica pero funcional para su objetivo.
- Desde un punto de vista técnico, su implementación es muy sencilla y en consecuencia el tiempo de desarrollo se acorta considerablemente respecto a otros protocolos en los que se complica la estructura de las tramas y con ello el acceso a los datos que no están almacenados en estructuras complejas.
- La transmisión de información no está comprometida a ningún tipo de datos. Lo que implica cierta flexibilidad a la hora del intercambio de información. Por ejemplo, si se transmite un dato de 16bits de información su representación no está sujeta a ninguna restricción, por lo que puede tratarse de un dato tipo Word con signo, un entero sin signo de 16bits o la parte alta de una representación tipo Float de 32bits, etc... La representación del valor vendrá definida por la especificación que el fabricante dé del dispositivo, lo que permite la representación de un amplio

² Modelo de interconexión de sistemas abiertos (Open System Interconnection) es un modelo de referencia para los protocolos de la red de arquitectura en capas, creado en el año 1980 por la Organización Internacional de Normalización.

³ Redes inteligentes.

rango de valores.

Modbus permite el control de una red de dispositivos. También se usa para la conexión de un ordenador de supervisión con una unidad remota (RTU) en sistemas de supervisión adquisición de datos (SCADA). Existen versiones del protocolo Modbus para puerto serie y Ethernet (Modbus/TCP). [4]

3.2 Funcionamiento y elementos

El funcionamiento tiene una base muy sencilla: el Maestro pregunta y los Esclavos responden o actúan en función de lo que este diga.

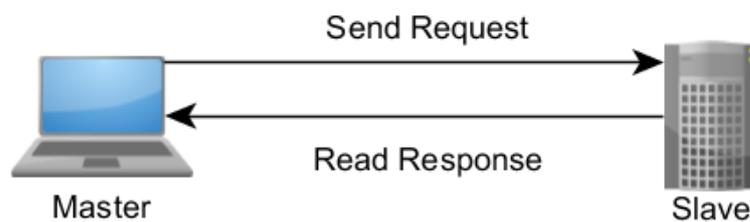


Figura 3-1 Interacción Maestro – Esclavo

Un dispositivo conectado al bus ejerce de maestro solicitando información del resto de dispositivos conectados que ejercen como esclavos y son quienes suministran la información al primero. Según el estándar Modbus y dada su implementación, en una red Modbus habrá un Maestro y hasta un máximo de 247 dispositivos Esclavos. Esta limitación está determinada por el simple hecho de que en una trama Modbus la dirección del esclavo se representa con un solo Byte, existiendo algunas direcciones reservadas para propósitos específicos como broadcast, etc. [5]

0	From 1 to 247	From 248 to 255
Broadcast address	Slave individual addresses	Reserved

Figura 3-2 Tipos de direcciones Modbus.

La dirección 0 se reserva como la dirección de difusión. Todos los nodos esclavos deben reconocer esta dirección.

El nodo Modbus maestro no tiene ninguna dirección específica, solamente los nodos esclavos deben tener una dirección que debe ser única en un bus serie MODBUS.

Dentro de la trama Modbus RTU, la dirección del esclavo corresponde al primer byte. En una red Modbus el Maestro no sólo puede ejercer la función de recopilar información de los esclavos mediante preguntas, sino que puede interactuar con ellos o alterar su estado, pudiendo escribir además de leer información en cualquiera de ellos.

El nodo Maestro emite una petición MODBUS a los nodos Esclavos en dos modos: [6]

- En el modo unicast, el Maestro se dirige a un Esclavo individual. Después de recibir y procesar la petición, el esclavo devuelve una respuesta al maestro.

En este modo, una transacción MODBUS consta de 2 mensajes: la solicitud del maestro y la respuesta del esclavo.

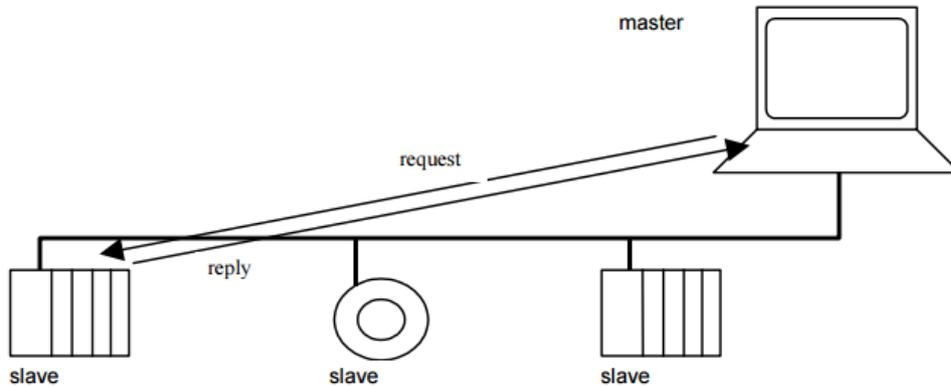


Figura 3-3 Modo unicast.

- En el modo de difusión, el Maestro puede enviar una petición a todos los Esclavos

El Esclavo no envía respuesta al Maestro. Las peticiones de difusión solo pueden ser de escritura. La dirección 0 está reservada para identificar un intercambio de difusión.

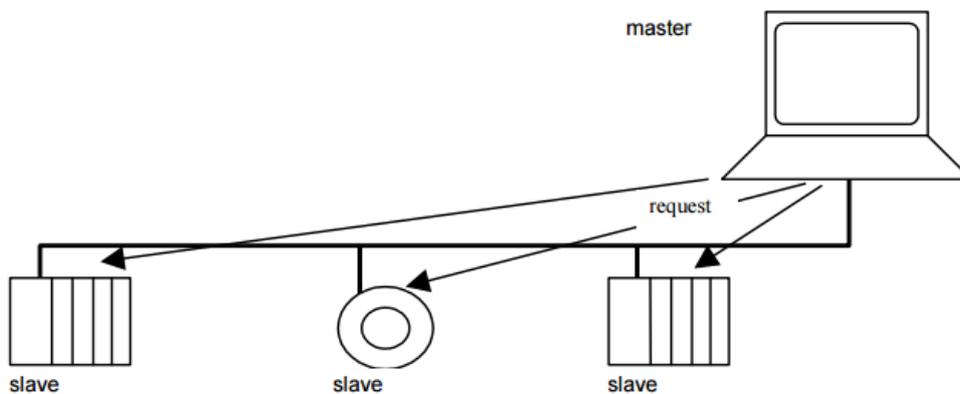


Figura 3-4 Modo broadcast.

3.2.1 Estructura de la trama Modbus

Existen dos variantes, con diferentes representaciones numéricas de los datos y detalles del protocolo ligeramente desiguales.

Modbus RTU es una representación binaria compacta de los datos. Modbus ASCII es una representación legible del protocolo, pero menos eficiente. Ambas implementaciones del protocolo son serie. El formato RTU finaliza la trama con una suma de control de redundancia cíclica (CRC), mientras que el formato ASCII utiliza una suma de control de redundancia longitudinal (LRC). La versión Modbus/TCP es muy semejante al formato RTU, pero estableciendo la transmisión mediante paquetes TCP/IP (puerto del sistema 502,

identificador *asa-appl-PROTO*).⁴

Como en cualquier protocolo de campo destinado al intercambio de información entre un servidor y los dispositivos esclavo, el protocolo Modbus RTU sigue una estructura de trama bien definida por campos. La estructura de trama Modbus es muy sencilla, siendo uno de los motivos de su éxito junto a ser un protocolo abierto y a no estar orientado a conexión como hemos indicado.

Así pues, la estructura básica de una trama Modbus RTU, tanto de lectura como escritura, es la que se muestra en la Figura 3-4. [7]

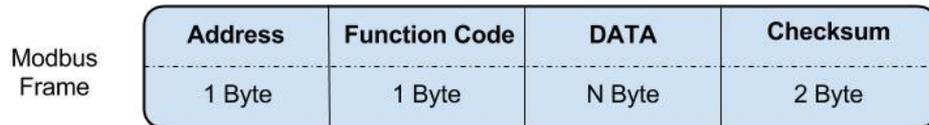


Figura 3-5 Trama Modbus.

➤ Address

Indica la dirección del esclavo, al ser 1 Byte de manera directa limita el número de esclavos que podemos tener conectados de forma correcta al bus serie Modbus. Dado que existen direcciones reservadas para propósitos especiales como el broadcast el valor que puede ir de 1 a 247.

El campo no se ve afectado si se trata de una trama de escritura o lectura.

Cuando el Maestro pregunta al Esclavo este campo contiene la dirección del Esclavo al que va dirigido. Cuando se trata de una trama de respuesta de un esclavo al maestro este campo contiene también la dirección del esclavo indicando quién es el que responde.

Las direcciones a las que se puede acceder de un PLC a otro son las que corresponde al módulo del procesador (BMX P34 2020), la memoria principal.

➤ Función Modbus

Con este campo se especifica que acción requiere el Maestro del Esclavo al que va dirigida la trama. Con el paso de los años se han ido añadiendo más funciones específicas.

En este caso, el valor contenido en este campo sí que puede variar si se trata de una trama Maestro->Esclavo o si por el contrario es Esclavo->Maestro.

El valor de este byte se verá modificado en la trama de respuesta sólo cuando exista algún error en el campo de datos de la trama Modbus enviada por el Maestro. En el caso de existir error el Esclavo responderá con la misma función que en la trama enviada por el Maestro, pero con la máscara 0x80 aplicada.

El código de operación puede tomar cualquier valor comprendido entre el 0 y el 127 (el bit de más peso se reserva para indicar error). Cada código se corresponde con una determinada operación. Algunos de estos códigos se consideran estándar y son aceptados e interpretados por igual por todos los dispositivos que dicen ser compatibles con MODBUS, mientras que otros códigos son implementaciones propias de cada fabricante. Es decir que algunos fabricantes realizan implementaciones propias de estos códigos “no estándar”.

Es también mediante el código de función que el esclavo confirma si la operación se ha ejecutado correctamente o no. Si ha ido bien responde con el mismo código de operación que se le ha enviado, mientras que, si se ha producido algún error, responde también con el mismo código de operación, pero con su bit de más peso a 1 (0x80) y un byte en el campo de datos indicando el código de error que ha tenido lugar. [8]

⁴ Service Name and Transport Protocol Port Number Registry

Por ejemplo: si el máster envía una trama con función 0x03 correcta el Esclavo responderá con el mismo campo y el mismo valor, 0x03; si por el contrario contiene algún error, el Maestro aplicará el valor 0x80 y responderá con una trama con el valor 0x83. [9]

Se describirán las funciones básicas, en el anexo se incluye una descripción más amplia de las funciones.

COD. FUNCION	CÓDIGO HEX.	FUNCIÓN DE LECTURA
1	0X01	Read Coil Status
2	0X02	Read Input Status
3	0X03	Read Holding Register
4	0X04	Read Input Register

Tabla 3-1 Funciones Modbus de lectura.

COD. FUNCIÓN	CÓDIGO HEX.	FUNCIÓN DE ESCRITURA
5	0X05	Force Single Coil
15	0X0F	Force Multiple Coils
6	0X06	Preset Single Register
16	0X10	Preset Multiple Registers

Tabla 3-2 Funciones Modbus de escritura.

➤ Byte de datos

Este campo dependerá tanto en contenido como en longitud de la función que se indique en el campo anterior (Función) así como de si se trata de una trama Maestro-Esclavo o de respuesta Esclavo-Maestro.

El campo de datos de los mensajes enviados desde un maestro a los dispositivos esclavos contiene informaciones que debe utilizar el esclavo para tomar la acción definida por el código de función. El campo de datos puede ser de diversas longitudes, o puede ser inexistente (de longitud cero). En el Ejemplo 1 puede verse como sería la pregunta y la respuesta de una lectura.

➤ CRC – Campo de comprobación de errores

Este campo consta de dos bytes y como en cualquier otro protocolo en el caso de Modbus sirve para la detección de errores en la trama. El CRC (Cyclic Redundancy Check o comprobación de redundancia cíclica) es un código más que frecuente en la detección de errores en redes digitales, sistemas de almacenamiento para la detección de modificación accidental de los datos o en este caso para comprobar la integridad de los datos en su transmisión por buses de campo.

Para el cálculo del CRC se utilizan cada uno de los bytes que conforman la trama. El procedimiento es el siguiente:

- Se envía la trama Modbus con el CRC calculado.
- El receptor del mensaje recibe la trama completa e internamente calcula el CRC con los datos recibidos. Y lo compara con el CRC que le ha llegado.
- Si el código coincide, la trama Modbus es correcta y se prosigue con el funcionamiento normal generando la respuesta pertinente.

Si el código es erróneo, es decir que no coincide el CRC recibido con el CRC no se responderá a la petición de datos por parte del Esclavo, de manera que ocurrirá un Timeout en recepción del Maestro y este deberá entender que el Esclavo no ha recibido la trama correctamente y procederá a un reintento. [10]

Ejemplo 3-1.

Query: El mensaje de petición especifica la bobina, la cantidad de bobinas y desde donde empezar a leer. Esta petición de lectura de la bobina 20 - 56, 37 bobinas en total, del esclavo 3: (Teniendo en cuenta que la dirección de inicio es de 19 o 0x13.)

Field Name	Example (Hex)	RTU 8-Bit Field (Hex)
Header		None
Slave Address	0x03	0x03
Function	0x01	0x01
Starting Address Hi	0x00	0x00
Starting Address Lo	0x13	0x13
No. of Registers Hi	0x00	0x00
No. of Registers Lo	0x25	0x25
Error Check		CRC (16 bits)
Trailer		None
Total Bytes:		8

Tabla 3-3 Trama Query.

Response: El estado de la bobina en el mensaje de respuesta es un bit del campo de datos. El estado se indica como: 1 = ON, 0 = OFF.

El LSB del primer byte de datos contiene la bobina abordado en la consulta.

Por ejemplo, cuando el estado de las bobinas 20 - 27 se muestra ON - ON - OFF - OFF - ON - OFF - ON - OFF, representado como el valor de byte binario 0101 0011 (0x53). Un byte contiene el estado de ocho bobinas. Si la cantidad de la bobina no es un múltiplo de ocho, los bits restantes en el byte de datos final se rellenan con ceros.

La Tabla 3-4 muestra un ejemplo de una respuesta a la consulta que se muestra en la Tabla 3-3.

Field Name	Example (Hex)	RTU 8-Bit Field (Hex)
Header		None
Slave Address	0x03	0x03
Function	0x01	0x01
Byte Count	0x05	0x05
Data 1	0x53	0x53
Data 2	0x6B	0x6B
Data 3	0x01	0x01
Data 4	0xF4	0xF4
Data 5	0x1B	0x1B
Error Check		CRC (16 bits)
Trailer		None
Total Bytes:		10

Tabla 3-4 Trama Response.

3.3 Cableado

3.3.1 Comunicación serie

La comunicación serie consiste en el envío secuencial de un bit a la vez de información entre dos o más dispositivos, cada bit va seguido uno de otro (en serie), esto implica que para la transmisión de datos es necesario un solo cable entre los dos dispositivos y una referencia a tierra.

Cualquier dato se puede enviar en serie, por ejemplo, para enviar el carácter “A” de un dispositivo a otro, siendo el número 65 el correspondiente en ASCII a la “A”, que convertido a unos y ceros (código binario) nos da como resultado el 1000001, entonces para enviar la letra “A” simplemente se envía una trama de bits.

Las características a tener en cuenta en una comunicación serie son:

- Bit de Start y Stop

A la trama de 8 bits se le agregan dos bits más, uno al inicio y otro al final, esto con la finalidad de indicarle al receptor cuando inicia el dato (bit de start) y cuando termina (bit de stop). El bit de inicio es un 0 y el bit de paro es un 1.

- Bit de paridad

Existe otro bit que se agrega a la trama, es el llamado bit de paridad, que sirve para detectar errores en las tramas enviadas. Su uso es opcional.

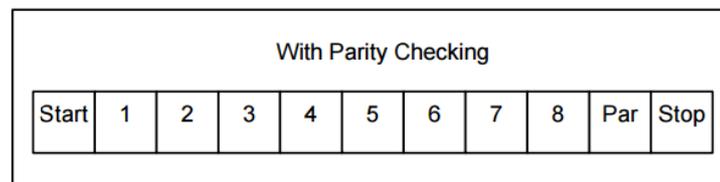


Figura 3-6 Bit de secuencia modo RTU con paridad.

- Velocidad de transmisión

Debemos tener en cuenta la velocidad a la que se envían los bits en serie, es de suma importancia debido a que ambos dispositivos deben estar a la misma velocidad para transmitir y recibir bits, si no lo están, simplemente la transmisión serial no sería correcta.

La velocidad de transferencia de información indica el número de bits que se transmiten en un segundo, se mide en baudios, algunas velocidades de transmisión más usadas son las siguientes: 1200, 2400, 4800, 9600, 19200. 9600, 19200.

3.3.2 Normas de comunicación en series

Este tipo de comunicaciones brinda el escenario ideal para construir redes de baja cobertura y bajo costo, ya sea con conexiones de corriente o de tensión.

En los circuitos de conexión en serie por corriente o bucle de corriente se emplea esta característica para representar la información del sistema a través de un cable que una los equipos terminales emisor y receptor. Por otro lado, en los circuitos de conexión por tensión o bucle de tensión es esta señal eléctrica la que se

emplea para representar la información a través del enlace establecido entre los dos conductores enlazados.

En el diseño moderno de comunicaciones en serie punto a punto el bucle de corriente no es utilizado debido a que ha sido sustituido por la transmisión de bucle de tensión con el uso cada vez mayor de los protocolos RS-232, RS-423, RS-422 y RS-485.

A diferencia de la conexión serie de corriente la conexión serie de tensión es muy utilizada en la actualidad debido a sus mejores prestaciones en alcance y velocidad. Las diferentes propuestas de normalización se clasifican en:

1. Especificaciones mecánicas, funcionales y de procedimiento dadas por la EIA RS-232C o CCITT V.24 y la EIA RS-449.
2. Especificaciones eléctricas dadas por la EIA RS-232C o CCITT V.28, EIA RS-423A (no equilibrada), EIA RS-422A (equilibrada) y la EIA RS-485 (equilibrada y con tercer estado).

Al transmitir en una banda de base digital se puede establecer una configuración mínima prescindiendo de las señales de control de tal forma que se puede utilizar únicamente dos líneas (una de masa y una de datos) para realizar una transmisión en un único sentido simultáneamente (half dúplex); éste es el caso que se ilustra en la Figura 3-9 como un diagrama de bloques de una comunicación asíncrona simultánea entre dos microcontroladores.

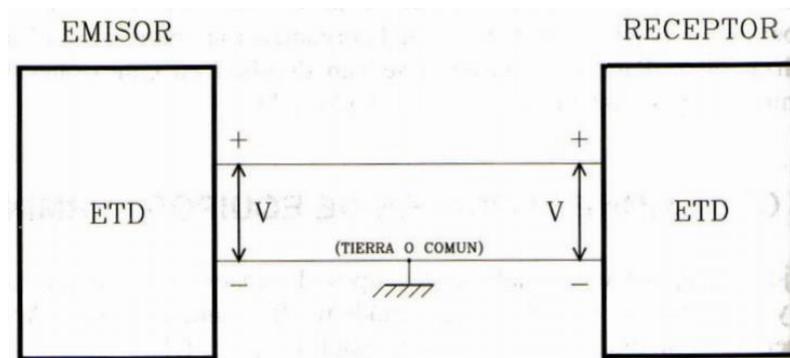


Figura 3-7 Diagrama de bloques con bucle de tensión.

A continuación, analizaremos el estándar que hemos elegido para nuestro Proyecto, el RS-485.

3.3.3 Estándar RS-485

RS-485 o también conocido como EIA-485, que lleva el nombre del comité que lo convirtió en estándar en 1983. Es un estándar de comunicaciones en bus de la capa física del Modelo OSI.

Está definido como un sistema de bus diferencial multipunto, es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbit/s hasta 10 metros y 100 kbit/s en 1200 metros) y a través de canales ruidosos, ya que el par trenzado reduce los ruidos que se inducen en la línea de transmisión. El medio físico de transmisión es un par trenzado que admite 32, 128 o 254 estaciones en 1 solo par, con una longitud máxima de 1200 metros operando entre 300 y 19 200 bit/s y la comunicación half-duplex (semiduplex) dependiendo del consumo de cada driver. La transmisión diferencial permite alcanzar mayor distancia con una notable inmunidad al ruido, siempre que el bus de comunicación conserve las características de bus balanceado dando la posibilidad de una configuración multipunto.

Desde 2003 está siendo administrado por la Telecommunications Industry Association (TIA) y titulado como TIA-485-A.222. [11]

RS-485 es una mejora sobre RS-422 ya que incrementa el número de dispositivos que se pueden conectar y define las características necesarias para asegurar los valores adecuados de voltaje cuando se tiene la carga máxima. Gracias a esta capacidad, es posible crear redes de dispositivos conectados a un solo puerto RS-485.

Esta capacidad, y la gran inmunidad al ruido, hacen que este tipo de transmisión serie sea la elección de muchas aplicaciones industriales que necesitan dispositivos distribuidos en red conectados a un PC u otro controlador para la colección de datos, HMI u otras operaciones.

3.3.3.1 Características mecánicas

El estándar define conexiones con cable de par de cobre trenzado y terminales RJ11 por lo que existe mayor resistencia a la interferencia electromagnética y mayor velocidad de transmisión que con la norma RS232 [12].

Permite la conexión de hasta 32 emisores con 32 receptores en transmisión doble simultánea full dúplex capaz de enlazar procesadores de comunicación principal (maestro) con procesadores subordinados (Esclavos) cuyo funcionamiento (acceso priorizado) está definido por los mismos arreglos topológicos de las redes de datos.

Los dispositivos de la norma RS-485 mantienen compatibilidad con el estándar RS-232 como se aprecia en la Figura 3-10 donde, se tiene un adaptador PCI Express de 16 puertos y alto rendimiento con enlaces para terminal serial RS-232.



Figura 3-8 Interfaz RS485 de 16 Puerto de Alto Rendimiento PCI Express.

3.3.3.2 Características eléctricas

En la comunicación de la norma RS-485 se tiene que el emisor opera el “1” lógico a un voltaje de -1.5 a -5 Volt. el “0” lógico a la entrada del receptor en el rango de +0.2 a +12 Volt y la máxima tensión aplicada a la línea de salida es de -7 a +12 Volt. El alcance de la transmisión está dado por la relación existente entre el volumen de los datos a transferir y el tiempo de la señal en la portadora determinado por la velocidad de transferencia de donde se obtiene que la longitud máxima del cable es de 1.200 m y la velocidad máxima de 10 Mbps que se obtiene en una distancia de 12 m.

Finalmente, aunque de manera oficial los drivers RS-485 pueden direccionar hasta 32 dispositivos en la red, con la nueva generación de productos de conversión de señales se logra una expansión de entre 128 y 255 nodos.

El bucle de tensión ha desplazado progresivamente al bucle de corriente en el diseño de redes industriales gracias a la flexibilidad y potencia obtenidos en el desarrollo de conceptos eléctricos de mayor complejidad sobre los cuales se estructura.

Los materiales de fabricación de la portadora, así como la distancia de transporte de datos impactan de manera inversamente proporcional a la capacidad y a su velocidad de transferencia. El futuro de las redes locales industriales demanda diariamente de estándares que sean capaces de manejar un mayor número de equipos maestro y proporcionalmente una mayor cantidad de terminales Esclavos para cumplir con las exigencias del mundo productivo moderno. [13]

3.3.4 Puerto serie del PLC M340.

El puerto serie RJ45 en nuestro PLC se localiza en la zona más baja del módulo BMX P34 2020 como se observa en la Figura 3-11

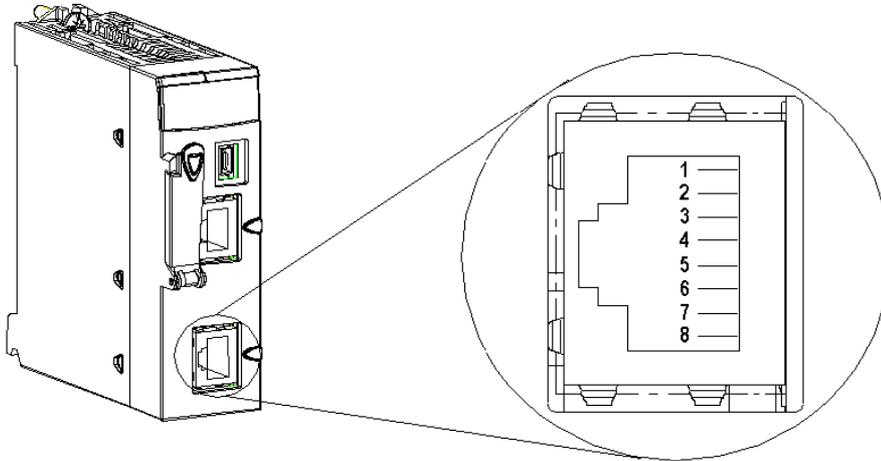


Figura 3-9 Puerto serie en el PLC M340.

El estado de la comunicación serie se indica mediante un indicador LED SER COM amarillo en la parte frontal de estos procesadores, si el LED está intermitente sabemos que la comunicación serie está en curso mientras que si está apagado no se estará comunicando.

El conector RJ45 tiene ocho pines. Los pines utilizados varían en función de la conexión física que se utilice, en nuestro caso será la RS485

Los pines utilizados para la conexión serie RS485 son los siguientes:

- Pin 4: señal D1
- Pin 5: señal D0

El pin 7 se utiliza exclusivamente para suministrar alimentación a interfaces hombre-máquina o a dispositivos de pequeñas dimensiones mediante el cable de la conexión serie:

- Pin 7: fuente de alimentación de la conexión serie: 5 VCC/190 mA

Las características de CC son:

- Consumo máximo de alimentación estabilizada: 190 mA,
- Tensión mínima en el conector de la CPU para 190 mA: 4,9 V,
- Tensión máxima en el conector de la CPU para 190 mA: 5,25 V,
- Tensión máxima en el conector de la CPU sin carga: 5,5 V.

Las características de CA son:

- Carga del condensador: (a 5 V)
 - Máximo 1 μ F en condensador cerámico;
 - 10 μ F en tántalo
- Inicio de carga de la bomba: (a 5 V)
 - 4 x 1 μ F en condensador cerámico;
 - 2 x 10 μ F en tántalo.

3.4 Configuración del PLC

Para configurar el funcionamiento de un puerto de comunicaciones de un PLC M340 de Schneider se requiere utilizar la aplicación Unity Pro XL, que también será necesaria para programar el PLC.

En primer lugar, se debe abrir o crear un proyecto con la configuración de bastidor que se utilizará. La Figura 3-12 muestra la configuración que se utilizará en el proyecto.



Figura 3-10 Configuración del bastidor.

Seguidamente, en el Explorador de Proyectos se debe hacer doble clic con el ratón sobre el puerto serie de la CPU, que está nombrado como SerialPort. Esto abrirá la ventana de propiedades y configuración del puerto serie. Dentro de esta ventana, se seleccionará el Canal 0 en la parte izquierda, como muestra la Figura 3-11, para acceder al formulario de configuración.

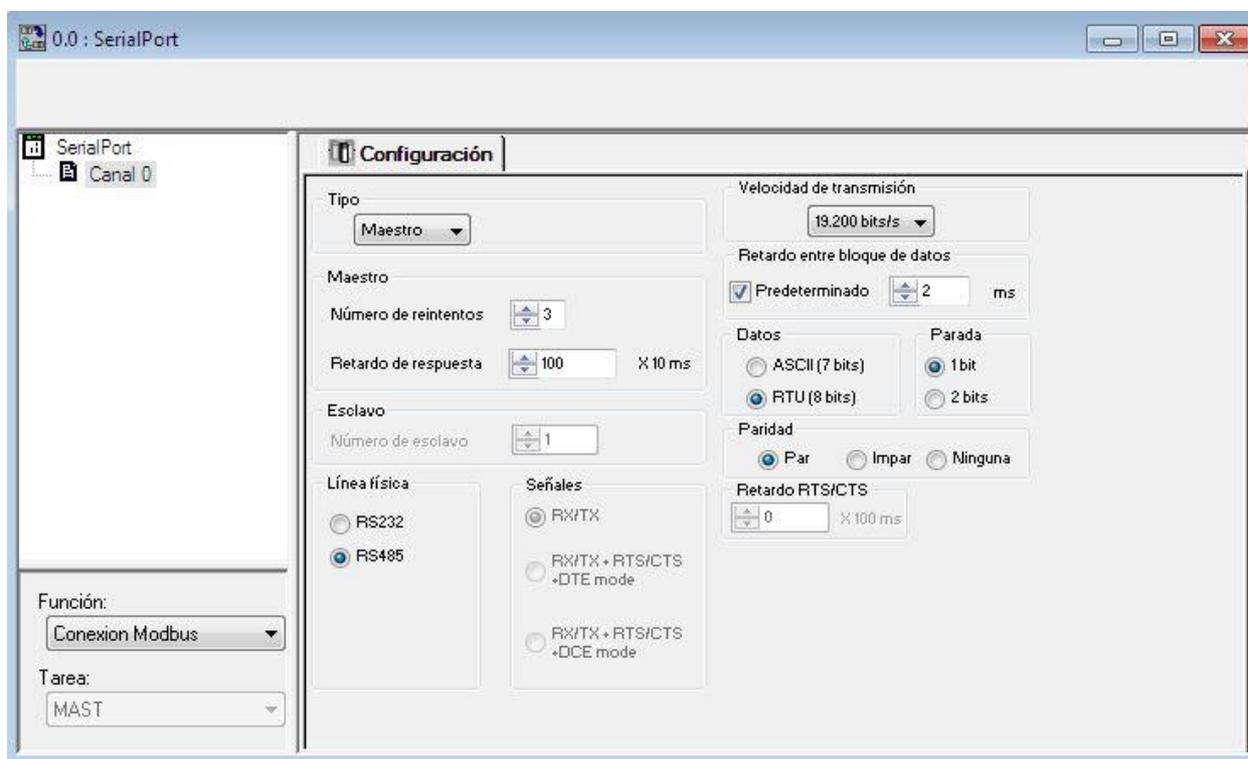


Figura 3-11 Configuración del puerto serie.

Bajo la lista de puertos y canales, después de seleccionar el canal, se puede seleccionar el protocolo a utilizar con el control de Función, que puede ser “Conexión modalidad de caracteres” o “Conexión MODBUS”. En función de la selección, la pestaña de “Configuración” mostrará un formulario con los parámetros configurables del protocolo escogido. A continuación, se describen los parámetros que se pueden configurar para MODBUS, que será el protocolo usado en el proyecto.

- Tipo (Esclavo, Maestro).

Selecciona el modo de MODBUS con que el PLC accederá al puerto serie. Hay que tener en cuenta que, en un momento dado, en una conexión o bus serie solo debe haber un dispositivo maestro.

- Maestro: Número de reintentos (0-15).

Número de intentos de conexión que lleva a cabo el maestro antes de declarar ausente al esclavo. El valor predeterminado es 3. Los valores posibles están comprendidos entre 0 y 15. El valor 0 indica que no hay reintentos por parte del maestro.

- Maestro: Retardo de respuesta (10ms-10s, en pasos de 10ms).

Es el tiempo transcurrido entre la petición inicial enviada por el maestro y un intento repetido de ésta en caso de que el esclavo no responda. Se corresponde con el tiempo máximo entre el envío del último carácter de la petición enviada por el maestro y la recepción del primer carácter de la petición reenviada por el esclavo. El valor predeterminado es 1 segundo (100*10 ms). Los valores posibles están comprendidos entre 10 ms y 10 s.

- Esclavo: Número de esclavo (1-247).

Indica la dirección del esclavo para MODBUS. En configuraciones multipunto (con un maestro y más de un esclavo) se usan las direcciones 1 a 247. El valor 248 se utiliza sólo para conexiones punto a punto (un maestro y un esclavo). Cada esclavo debe tener una dirección diferente al resto.

- Línea Física (RS-232, RS-485).

Especifica qué tipo de líneas físicas del puerto serie se van a utilizar. Aunque el puerto serie de los M340 puede funcionar físicamente como RS-232 o RS-485, no puede hacerlo simultáneamente, ya que solo dispone de un canal serie que debe ser configurado en uno de los dos modos.

- Señales (RX/TX, RX/TX + RTS/CTS).

En el caso de utilizar el nivel físico RS-232, este campo se habilita para escoger si se van a usar las líneas de control del interfaz serie además de las líneas de datos.

- Velocidad de transmisión (300, 600, 1.200, 2.400, 4.800, 9.600, 19.200 y 38.400 bps).

Es la velocidad de transmisión que se usará para el puerto serie, y que debe ser igual a la configurada en los otros dispositivos del bus.

- Retardo entre bloque de datos (2ms – 3.413ms).

Indica el tiempo mínimo que puede separar dos tramas en la recepción. Este valor debe ajustarse en función de la velocidad. Según el estándar de MODBUS, para velocidades de hasta 19.200bps, el tiempo entre tramas debe ser como mínimo 3,5 veces la duración de un carácter, y para velocidades superiores se recomienda un tiempo fijo de 1,75ms. La duración de un carácter depende de la velocidad (V_t) y el número de bits (N) según N/V_t . Por ejemplo, para 19.200bps, con un bit de parada y un bit de paridad (11 bits en total, sumando el de inicio y 8 de datos) se tiene: $3,5 \cdot 11 / 19.200 = 2\text{ms}$.

- Datos (ASCII, RTU).

Selecciona el formato para las tramas de MODBUS serie.

- Parada (1, 2).

Permite introducir el número de bits de parada utilizados en la comunicación, que debe ser igual a la configurada en los otros dispositivos del bus.

- Paridad (par, impar, ninguna).

Permite determinar si se agrega un bit de paridad, así como su tipo. El bit de paridad permite al equipo receptor de una trama de datos determinar si los bits de datos son correctos, o ha habido una alteración en uno de ellos. Esta configuración debe ser igual a la configurada en los otros dispositivos del bus.

- Retardo RTS/CTS (0-10s, pasos de 100ms).

Cuando es mayor que 0, indica que el PLC tendrá en cuenta las señales RST y CTS para controlar el envío de datos. En este caso, antes de enviar una trama, el PLC activa su salida RTS y espera que, en respuesta, el DCE active la entrada. Solo está disponible para RS232. [14]

El esclavo solo requiere ser configurado con las mismas características que el maestro para que se pueda establecer la comunicación y asignarle un número de esclavo. La figura 3-14 muestra como es la pantalla de configuración del PLC que será establecido como esclavo.

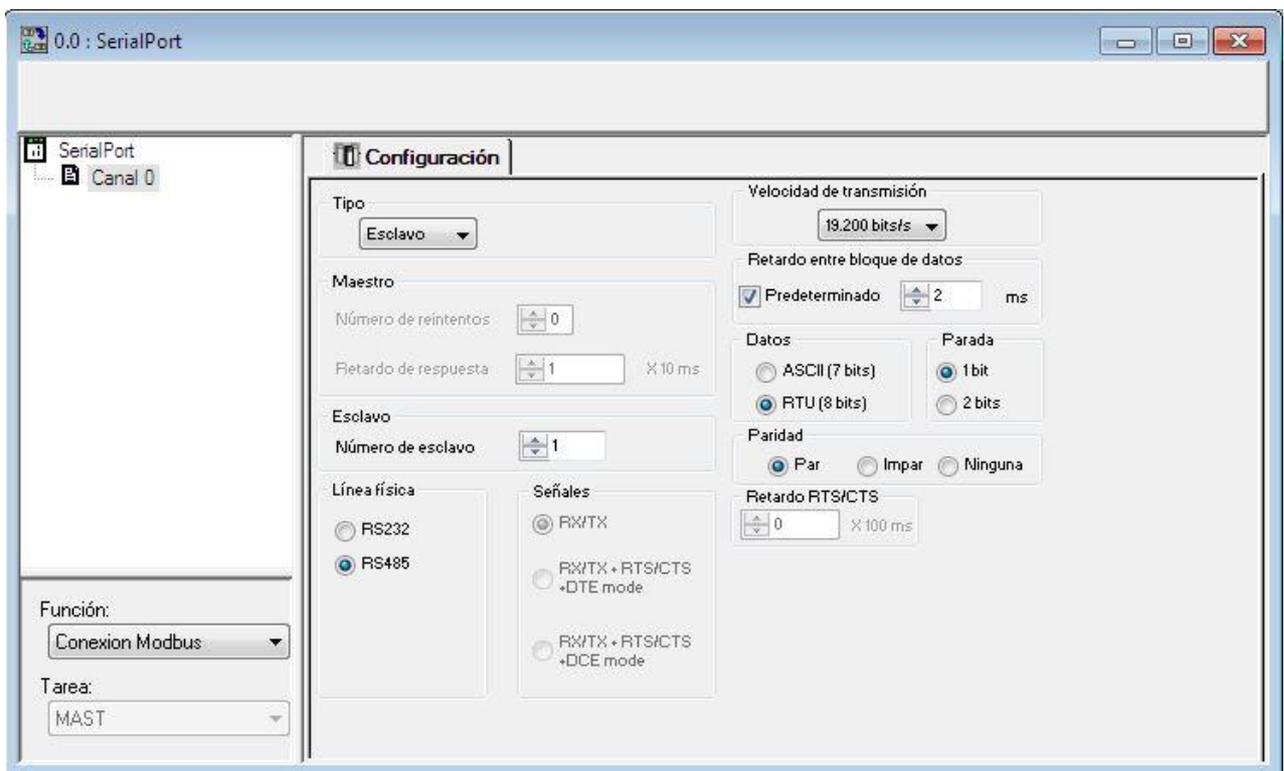


Figura 3-12 Configuración del cliente Modbus.

3.5 Tiempo de intercambio de una palabra

En la siguiente tabla se muestran los tiempos de intercambio de una palabra de comunicación Modbus en un procesador BMX P34 2020⁵:

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de Intercambio en ms El esclavo Modbus es un BMX P34 1000 cíclico
4.800	Cíclico	68
4.800	10	72
4.800	50	100
9.600	Cíclico	35
9.600	10	40
9.600	50	50
19.200	Cíclico	20
19.200	10	27
19.200	50	50
38.400	Cíclico	13
38.400	10	20
38.400	50	50

Tabla 3-5 Tiempo intercambio de una palabra en Modbus.

⁵ Todos los tiempos de intercambio enumerados provienen de medidas con un margen de precisión de +/-10 ms.

4 COMUNICACIONES PLC VÍA ETHERNET

Ethernet es un estándar de redes de área local para computadores con acceso al medio por detección de la onda portadora y con detección de colisiones (CSMA/CD). Su nombre viene del concepto físico de *ether*. Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

Ethernet se tomó como base para la redacción del estándar internacional IEEE 802.3, siendo usualmente tomados como sinónimos. Se diferencian en uno de los campos de la trama de datos. Sin embargo, las tramas Ethernet e IEEE 802.3 pueden coexistir en la misma red.

Los protocolos asociados de los que disponen el PLC Modicon M340 son:

Estos protocolos sirven para: transferencia de ficheros, gestión de red, asignación de dirección:

- Los protocolos de aplicaciones: HTTP, FTP, SNMP, DNS, TELNET.
- Los protocolos de adquisición de configuración: BootP, DHCP.
- Los protocolos de comunicación: ARP, RARP, ICMP.

Las comunicaciones Ethernet ya no se rigen por el modelo Maestro-Esclavo. En este tipo de redes todos los PLCs pueden leer y escribir entre sí.

La trama Ethernet encapsula la de Modbus, pero se rige por la norma IEEE que indica cómo deben comportarse las comunicaciones. El hecho de encapsular la trama Modbus es para poder utilizar los Código de Funciones Modbus, pero con las ventajas que ofrece las comunicaciones a través del protocolo Ethernet.

Los nodos se comunican entre ellos por medio de "frames" (marcos), su unidad básica de comunicación, sabiendo a quien debe llegar y de quién procede. En el Anexo II se detallan las principales normas para realizar la comunicación que utilizan la norma IEEE.

La idea principal para lograr este objetivo, a cada nodo se le asigna una dirección única, diferente de la del resto de los nodos (pareja Mac/IP)⁶. Así, un frame se estructura en tres campos de datos, uno para la dirección de destino, otro para la dirección fuente, y un tercero donde se envían los datos en sí mismo que queremos enviar (datos del mensaje o payload).

La manera en la que se gestionan los frames es la siguiente, en una red compuesta por 4 nodos, A, B, C y D, si A genera un frame con destino a D, este frame es "escuchado" por B, C y D, pero solo lo acepta D, ya que B y C lo descartan porque la dirección destino del frame no es la suya (lo "filtran"), y D al ver que tiene como destino a él lo coge.

Esto tiene una consecuencia muy importante: solo un nodo puede transmitir a la vez en la red, ya que si otro emite habría problemas, para ello existe un mecanismo que implemente una serie de reglas para el acceso al medio (cable), sin profundizar demasiado, decir que se CSMA/CD⁷:

- CS - Carrier Sense ("sentir" la portadora). ¿Hay alguien hablando?
- MA - Multiple Access (Acceso múltiple). Lo que tu oyes yo también lo oigo.
- CD - Collision Detection (detección de portadora). ¡Mira, estamos hablando a la vez!

Así sería su funcionamiento:

⁶ Mediante el uso del protocolo ARP documentado en el RFC826. Anexo III.

⁷ Definido en la norma IEEE 802.3

1. Si el medio está desocupado, transmitir.
2. Si está ocupado, esperar.
3. Si ocurre una colisión, esperar un tiempo aleatorio e ir al paso 1.

También existe una dirección especial, la dirección "broadcast", los frames con esta dirección de destino son escuchados por todos los miembros y procesados por todos ellos. Un uso típico de esta utilidad es hacer desde un nodo una petición a todos los nodos para saber qué servicios provee cada uno a la red y que sean accesibles desde el nodo que realizó la petición.

Es impórtate a la hora de diseñar nuestra solución de qué manera van a acceder los PLCs unos a otros, con qué frecuencia y así optimizar los tiempos de acceso al nodo.

En nuestro proyecto recurrimos a este protocolo para transmitir Modbus llamado Modbus TCP/IP

4.1 Modbus sobre Ethernet

Modbus/TCP se refiere a menudo como MODBUS sobre Ethernet. En su mayor parte, es simplemente paquetes Modbus encapsulados en el estándar TCP/IP. Esto permite a los dispositivos Modbus/TCP conectarse inmediatamente, fácilmente y comunicarse a través de redes Ethernet y fibra.

Modbus-TCP es simplemente el protocolo Modbus RTU con una interfaz TCP que se ejecuta en Ethernet.

La estructura de mensajería Modbus es el protocolo de aplicación que define las normas para la organización y la interpretación de los datos independiente de los datos medio de transmisión.

TCP/IP se refiere al Protocolo de Control de Transmisión y Protocolo de Internet, que proporciona el medio de transmisión de mensajes Modbus TCP/IP.

En pocas palabras, TCP/IP permite que bloques de datos binarios se intercambien entre ordenadores. También es un estándar mundial como base para la World Wide Web. La función principal de TCP es que todos los paquetes de datos se reciban correctamente, mientras que se asegura con la IP el destino del mensaje y su paso correcto. Teniendo en cuenta que el TCP/IP no es más que un protocolo de transporte, y no define los datos ni cómo deben interpretarse (esto es el trabajo del protocolo de aplicación, Modbus en este caso).

En resumen, Modbus TCP utiliza TCP/IP y Ethernet para transportar los datos de la estructura del mensaje Modbus entre dispositivos compatibles. Es decir, Modbus TCP/IP combina una red física (Ethernet), con una creación de redes estándar (TCP/IP), y un método estándar de representación de datos (Modbus como el protocolo de aplicación). En esencia, el mensaje Modbus TCP/IP es simplemente una comunicación Modbus encapsulado en una red Ethernet TCP/IP.

En la práctica, Modbus TCP incorpora una trama de datos estándar Modbus TCP en un marco, sin la suma de control Modbus, como se muestra la Figura 3-10.

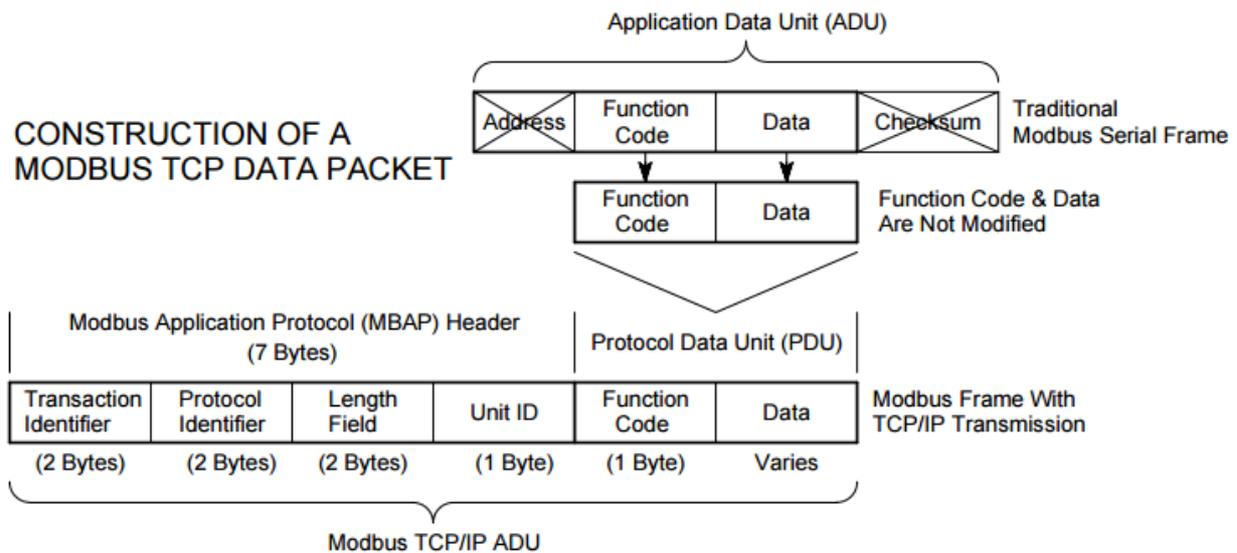


Figura 4-1 Paquete Modbus TCP.

Los comandos de Modbus y los datos del usuario se encapsulan en el mismo campo de la trama TCP IP sin ser modificados. Sin embargo, no se utiliza el CRC de Modbus (checksum) y el campo de dirección de trama Modbus es suplantado por el identificador de la unidad en Modbus TCP/IP, y se convierte en parte de la Protocolo de aplicación Modbus.

A partir de la figura, vemos que los campos de código de función y de datos son absorbidos en su forma original. Por lo tanto, una unidad de datos de aplicación Modbus TCP/IP (ADU) toma la forma de un identificador de cabecera 7 byte (identificador de transacción + protocolo + Campo de longitud de identificador de la unidad), y la unidad de datos de protocolo (código de función + datos). La cabecera MBAP⁸ es de 7 bytes de longitud e incluye los siguientes campos:

- **Identificador de Transacción (2 bytes):** Esta identificación se utiliza para el emparejamiento de transacciones cuando son múltiples mensajes enviado a lo largo de la misma conexión de TCP de un cliente sin esperar una respuesta previa.
- **Protocolo de identificación (2 bytes):** Este campo es siempre 0 para Modbus y los demás valores están reservados para futuras ampliaciones.
- **Longitud (2 bytes):** Contador de bytes de los campos restantes e incluye el identificador de la unidad, el código de función y el campo datos.
- **Identificador de unidad (1 byte):** Este campo se utiliza para identificar de forma remota un servidor ubicado en una red no TCP/IP (por puente de serie). En una aplicación típica del servidor Modbus TCP, el ID de la unidad se establece en 00 o FF, ignorado por el servidor, y simplemente se recibe eco de vuelta en la respuesta. [14]

La unidad de datos de aplicación Modbus TCP / IP completa se incrusta en el campo de datos de una trama TCP estándar y se envía a través de TCP por el puerto 502, que está reservado específicamente para aplicaciones Modbus. Los clientes Modbus TCP y servidores escuchan y reciben datos a través del puerto Modbus 502. Podemos ver que el funcionamiento de Modbus a través de Ethernet es casi transparente a la estructura de registro Modbus. Por lo tanto, para una persona familiarizada con el funcionamiento de Modbus tradicional no tendrá problemas para manejar Modbus TCP / IP.

⁸ Modbus Application Header

4.2 ¿Por qué combinar Modbus con Ethernet?

IEEE 802.3 Ethernet es un protocolo para estandarizar redes basadas en Ethernet que ha ganado aceptación en todo el mundo. También es un estándar abierto apoyado por muchos fabricantes y su infraestructura está disponible en la mayoría de instalaciones. [15]

Como muchos dispositivos ya son compatibles con Ethernet, ha aumentado su uso en aplicaciones industriales. Al igual que con Ethernet, Modbus está libremente disponible, accesible a cualquier persona, y ampliamente soportado por muchos fabricantes de equipos industriales. También es fácil de entender y un candidato natural para su uso en la construcción de otros estándares de comunicación industriales. Con tanto en común, la unión del protocolo de aplicación Modbus con el tradicional IEEE 802.3 de Ethernet constituye un poderoso estándar de comunicación industrial en Modbus TCP / IP. Y debido a que comparten las mismas características físicas y de capa de enlace sigue siendo totalmente compatible con la infraestructura Ethernet instalada de cables, conectores, interfaz de red tarjetas, los concentradores y conmutadores. [5]

Modbus/TCP también permite muchas más direcciones que RS485, el uso de múltiples Maestros y velocidades en el rango de gigabit. Mientras MODBUS RTU tiene una limitación de 247 nodos por red, las redes Modbus/TCP pueden tener tantos esclavos como la capa física pueda manejar ya que se identifica el PLC de destino mediante su IP. A menudo, este número está entorno a los 1.024. La rápida adaptación de Ethernet dentro del control de procesos y automatización de la industria ha permitido que Modbus/TCP para convertirse en el más utilizado, con un rápido crecimiento en los protocolos industriales a través de Ethernet.

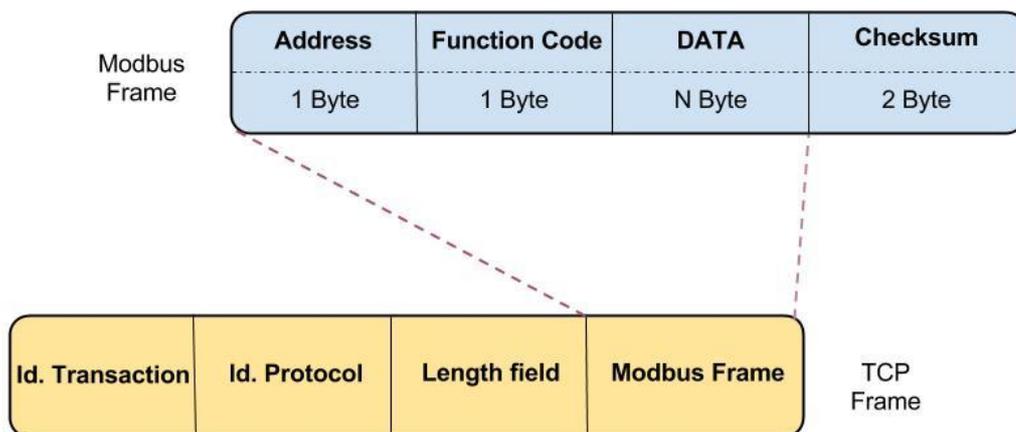


Figura 4-2 Trama Modbus TCP/IP.

Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502. El campo dirección esclavo de MODBUS es reemplazado por un byte identificador de unidad el cual puede ser usado para comunicar a través de dispositivos tales como puentes y gateways, los cuales usan una dirección IP única para soportar múltiples unidades terminales independientes.

Aunque los proveedores de PLCs de todos tipos han adoptado sus propios protocolos propietarios a través de Ethernet, casi todos ellos soportan Modbus/TCP. Para los fabricantes de autómatas programables que actualmente no lo hacen, hay muchas empresas como Prosoft Tecnologías y SST que ofrecen pasarelas para poder incorporar este protocolo en nuestra red.

A diferencia de Modbus RTU y Modbus ASCII, Modbus/TCP permitirá a varios patrones para sondear el mismo Esclavo simultáneamente. Esto es posible porque Ethernet permite el envío múltiple de mensajes. [3]

4.2.1 Ventajas

- Es simple para administrar y expandir. No se requiere usar herramientas de configuración complejas cuando se añade una nueva estación a una red Modbus/TCP. No es necesario equipo o software

propietario de algún vendedor. Cualquier sistema de cómputo con una pila de protocolos TCP/IP puede usar Modbus/TCP.

- Puede ser usado para comunicación con una gran base instalada de dispositivos MODBUS, usando productos de conversión los cuales no requieren configuración.
- Es de muy alto desempeño, limitado típicamente por las capacidades de comunicación del sistema operativo del computador. Se pueden obtener altas tasas de transmisión sobre una estación única y la red puede ser configurada para lograr tiempos de respuesta garantizados en el rango de milisegundos.
- Realizar reparaciones o mantenimiento remoto desde la oficina utilizando un PC, reduciendo así los costes y mejorando el servicio al cliente.
- El ingeniero de mantenimiento puede entrar al sistema de control de la planta desde su casa, evitando desplazamientos.
- Permite realizar la gestión de sistemas distribuidos geográficamente mediante el empleo de las tecnologías de Internet/Intranet actualmente disponibles. [17]

4.3 Cableado

El cable Ethernet es el tipo de cable utilizado habitualmente para interconectar todos los dispositivos que conforman una LAN, incluyendo impresoras, discos externos, routers, escáneres, switches y por supuesto las propias computadoras.

4.3.1 Topología de la red

Dado que los sistemas industriales usualmente consisten de dos o más dispositivos, este puede ser bastante grande, por lo que debe considerarse la topología de la red. Las más comunes son: bus y estrella

- Bus: Una topología en bus es multipunto. Un cable largo actúa como una red troncal que conecta todos los dispositivos en la red. Este cable provee un camino para las señales eléctricas que es común para todos los dispositivos conectados y transporta todas las transmisiones entre los dispositivos.

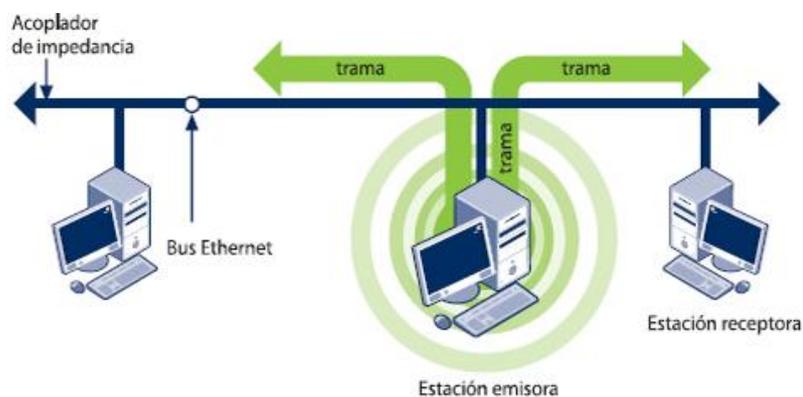


Figura 4-3 Topología en bus para la red Ethernet.

Entre las ventajas de la topología de bus se incluye la sencillez de instalación. El cable troncal puede tenderse por el camino más eficiente y, después, los nodos se pueden conectar al mismo mediante líneas

de conexión de longitud variable. De esta forma se puede conseguir que un bus use menos cable que una malla, una estrella o una topología en árbol. Un problema asociado con el diseño bus de cableado es que un fallo en cualquier parte del cable va a interrumpir la operación de todos los dispositivos conectados.

- Estrella: una topología en estrella, todos los dispositivos están conectados a través de un dispositivo intermedio. Este dispositivo intermedio puede ser un Hub o un Switch. La estrella es la topología que se utiliza habitualmente en redes corporativas y actualmente se adopta en casi todas las aplicaciones de automatización.

Tiene dos ventajas grandes a diferencia de la topología Bus:

Es más tolerante, esto quiere decir que si un dispositivo se desconecta o si su cable sufre un deterioro solo ese dispositivo es afectado y el resto de la red mantiene su comunicación normalmente (considerando que el switch o hub no debe fallar).

Es fácil de reconfigurar. Añadir o remover un equipo es tan simple como conectar o desconectar el cable.



Figura 4-4 Topología en estrella para la red Ethernet.

Cada intercambio puede llevarse a cabo a una velocidad nominal (mayor división de ancho de banda), sin colisiones y con un aumento considerable en el ancho de banda de la red (también a una velocidad nominal). El correcto funcionamiento de la comunicación requiere que el tiempo de ida y vuelta entre dos estaciones cuales quiera de la red no supere el tiempo que tarda en emitirse la trama mínima permitida. Este tiempo, que depende de la velocidad de la red, fija a su vez unas distancias máximas entre las estaciones. Estos cuatro parámetros (velocidad de la red, tamaño de trama mínimo, tiempo de ida y vuelta y distancia máxima) están relacionados entre sí, como se muestra en la tabla siguiente:

Velocidad (Mb/s)	Tamaño de trama mínimo (bits)	Tiempo de ida y vuelta (μ s)	Distancia máxima (m)
10	512	51,2	4000
100	512	5,12	412
1000	4096	4,096	330

Tabla 4-1 Velocidad y tiempo en una red Ethernet.

Las distancias indicadas en la tabla son el caso óptimo; en la práctica la distancia depende de múltiples factores, como el número de repetidores intermedios o el tipo de cable utilizado. La comprobación última de que una determinada topología de red es válida sería medir o calcular el tiempo de ida y vuelta entre cada posible par de estaciones de la red para comprobar que es inferior al que aparece en la tabla. [18]

En una red en estrella disponemos de dos tipos de dispositivos intermedios:

- Hub: es un concentrador multipuerto que opera en la capa física, contiene varios puertos. Cuando un paquete es recibido en un puerto, es copiado a todos los demás puertos, para que cualquier nodo conectado a la red pueda ver el paquete.
- Switch: A diferencia del hub, el switch (conmutador) puede detectar qué dispositivos están conectados, de manera que sólo transmite la señal a los que están activos. También es capaz de

reconocer las direcciones MAC de los dispositivos, por lo que al conectarlos puede saber cuáles son más rápidos y más lentos, de forma que posee un mejor funcionamiento en cuanto a la velocidad y el rendimiento. Mantiene una tabla en la que figuran las direcciones MAC de todos los dispositivos conectados (y en que puerto). Así, cuando un paquete es recibido, el Switch sabe exactamente a donde debe emitir el Frame, sin congestionar la red. Al contrario que el HUB, el Switch no comparte el ancho de banda entre los nodos, de esta manera cada nodo tiene el 100% del ancho de banda disponible.

En nuestra aplicación haremos uso de un switch para unir las comunicaciones Ethernet.

4.3.2 Tipos de cables.

Ethernet es cable encargado de llevar todos los datos que usualmente transitan por una red, siendo el más utilizado en las instalaciones estándar el denominado como “Cable de par trenzado” (Figura 4-3).

El cable de par trenzado es un medio de conexión usado en telecomunicaciones en el que dos conductores eléctricos, aislados son entrelazados para anular las interferencias de fuentes externas y diafonía de los cables adyacentes.

4.3.2.1 Cable UTP. Par trenzado sin apantallar.

Tipo de cable que no está blindado, utilizado principalmente para comunicaciones. Se divide en diferentes categorías, adecuadas cada una a una función específica:

- Cat. 1.: Esta categoría de cable UTP se compone de dos pares de hilos, y es utilizado únicamente en la implementación de instalaciones telefónicas.
- Cat. 2.: Se utiliza principalmente para el uso del teléfono, a diferencia del cable Cat 1, el Cat 2 es capaz de transferir datos a una tasa máxima de 4 Mbps.
- Cat. 3: Conocida como “Ethernet 10BaseT” es capaz de ofrecer una tasa de transferencia máxima de 10 Mbps. Hasta la aparición de los cables UTP Categoría 5 era el estándar para las conexiones de red, tanto en las empresas como en la pequeña oficina y el hogar.
- Cat. 4: Esta categoría de cable es conocida también como “Ethernet 10baseT/TokenRing”, y es capaz de ofrecer una velocidad de transferencia máxima de hasta 20 Mbps.
- Cat. 5: El tipo de cable UTP más utilizado en la actualidad. Conocido técnicamente como “Ethernet 100BaseT/10BaseT” puede llegar a alcanzar hasta una tasa de transferencia máxima de 100 Mbps. Esta categoría de cables se usa en la mayoría de redes corporativas y del hogar. Con el tiempo se acabó convirtiendo en un estándar.
- Cat. 6: Este tipo de cable UTP ha sido desarrollado con el propósito de obtener altas velocidades de transferencia a distancias más largas, y es por ello que puede alcanzar tasas de transferencia de hasta 1 Gbps a una distancia de 100 metros.
- Cat. 7: Desarrollado para que sea capaz de alcanzar 10 Gbps de velocidad de transferencia a distancias de hasta 100 metros, el cable UTP Categoría 7 además es extremadamente resistente a las interferencias eléctricas. Para ello está provisto de aislamiento suficiente como para que el ruido eléctrico no sea un problema. Además, es compatible con las anteriores categorías. [16]

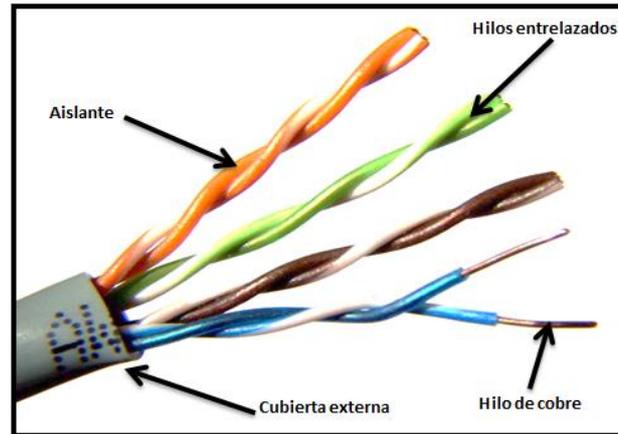


Figura 4-5 Cable par trenzado sin apantallar.

Las ventajas de este tipo de cableado son:

- Bajo costo en su contratación
- Alto número de estaciones de trabajo por segmento.
- Facilidad para el rendimiento y la solución de problemas.
- Puede estar previamente cableado en un lugar o en cualquier parte.

Mientras que las desventajas son:

- Altas tasa de error a altas velocidades.
- Ancho de banda limitado.
- Baja inmunidad al ruido.
- Distancia limitada a 100 metros por segmentos.

El conector utilizado en los cables STP son los RJ45, uno de los conectores principales utilizados con tarjetas de red ethernet, utiliza 8 pines, que transmiten información a través de cables de par trenzado. Por este motivo se le suele denominar puerto Ethernet.

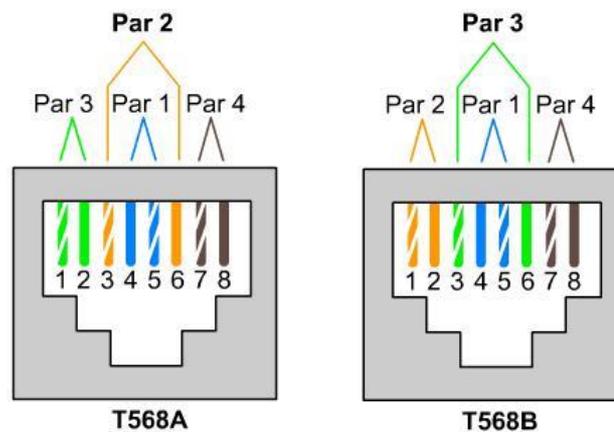


Figura 4-6 Código de colores conector RJ45.

Existen dos clases de configuraciones para los pines de los conectores del cable trenzado denominadas T568A y T568B.

4.3.2.2 Cable STP. Par trenzado apantallado.

El cable de par trenzado blindado (STP) combina las técnicas de blindaje, cancelación y trenzado de cables.

Cada par está envuelto en un papel metálico. Los dos pares de hilos están envueltos junto a una trenza o papel metálico.

El cable STP Brinda mayor protección ante toda clase de interferencias externas, pero es más caro y de instalación más difícil que UTP.

Si bien en la actualidad las redes inalámbricas están cada vez más extendidas, lo cierto es que las redes mediante cables ofrecen ventajas que las redes WLAN por el momento no pueden alcanzar, como largas distancias y estar libres de interferencias.

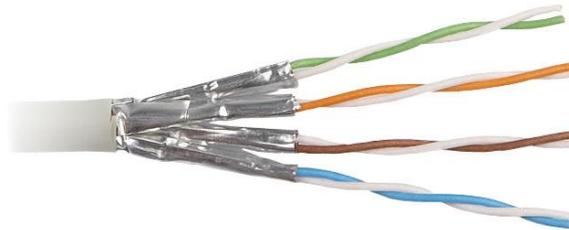


Figura 4-7 Cable STP. Par trenzado apantallado.

El conector del cable STP es el RJ-49. Igual que el RJ45, pero recubierto con una platina metálica para que haga contacto con la que recubre el cable STP

4.3.2.3 Cable FTP. Par trenzado con pantalla global.

En este tipo de cable como en el UTP, sus pares no están apantallados, pero sí dispone de una apantalla global para mejorar su nivel de protección ante interferencias externas. Su impedancia típica es de 120 Ohm y sus propiedades de transmisión son más parecidas a las del UTP. Además, puede utilizar los mismos conectores RJ45.

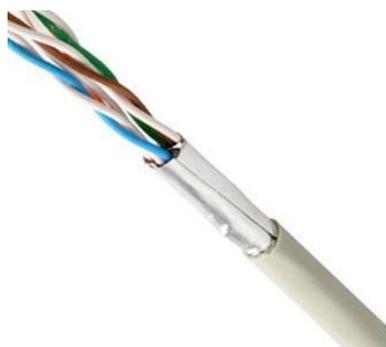


Figura 4-8 Cable FTP. Par trenzado con pantalla global.

En la actualidad, un cable Ethernet de red estándar es un cable UTP 8 Categoría 5 que en su interior aloja 4 pares de hilos, el secreto de su velocidad, recubiertos con una malla metálica cuyo propósito es el de proteger los datos de todo tipo de interferencias eléctricas.

Aunque los criterios de rendimiento de la Categoría 5e y Categoría 6 productos son los mismos para ambos entornos, los requisitos para proteger a los productos procedentes de las condiciones ambientales pueden ser muy diferentes.

4.3.3 Puerto Ethernet del PLC

El puerto serie RJ45 en nuestro PLC se localiza en la zona media del módulo BMX P34 2020 como se observa en la Figura 4-9.

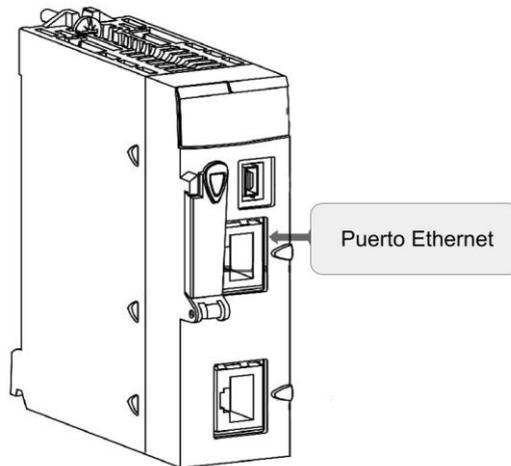


Figura 4-9 Puerto RJ45 Ethernet.

El estado de la comunicación Ethernet se indica mediante los leds del frontal del módulo. Los colores y los modelos de intermitencia de los LED indican el estado y las condiciones de funcionamiento de las comunicaciones Ethernet en el módulo:

ETH STS (verde): estado de comunicación Ethernet	Encendido	Comunicación correcta.
	2 intermitencias	Dirección MAC no válida.
	3 intermitencias	Enlace sin conexión.
	4 intermitencias	Dirección IP duplicada.
	5 intermitencias	En espera de una dirección IP del servidor.
	6 intermitencias	Modalidad segura (con una dirección IP predeterminada).
	7 intermitencias	Conflicto de configuración entre los conmutadores rotatorios y la configuración interna.

Tabla 4-2 Código errores puerto Ethernet.

La interfaz 10/100 BASE-T del módulo es un conector RJ45 estándar. En entornos industriales, los cables utilizados deben tener las siguientes características:

- par doble trenzado y blindado
- impedancia $100 \pm 15 \Omega$ (de 1 a 16 MHz)
- atenuación máxima 11,5 dB/100 metros
- longitud máxima 100 metros

Las diferentes velocidades de la línea disponibles son:

- 100 Mb en semiduplex

- 100 Mb en dúplex completo
- 10 Mb en semidúplex
- 10 Mb en dúplex completo

El usuario no puede configurar la velocidad de la línea. Las características de adaptación de velocidad son:

- La detección automática y la negociación automática permiten al módulo Ethernet configurarse rápidamente al modo de dúplex y a la velocidad del conmutador Ethernet local.
- La velocidad negociada entre dos dispositivos Ethernet está limitada a la velocidad del dispositivo más lento.

4.4 Configuración del PL

Para configurar el funcionamiento de un puerto Ethernet de comunicaciones de un PLC M340 de Schneider se requiere utilizar la aplicación Unity Pro XL, que también será necesaria para programar el PLC.

Tras haber creado el proyecto, en primer lugar, debemos crear una nueva red en el menú 'Explorador de proyectos' como se puede observar en la Figura 4-10.

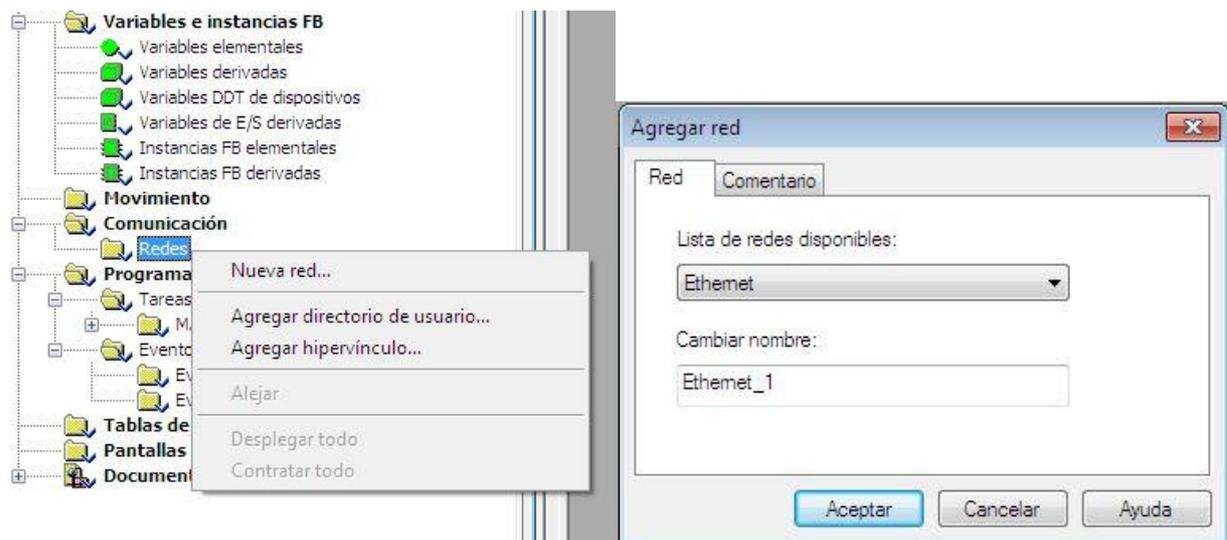


Figura 4-10 Crear una nueva red Ethernet

Le asignaremos en este caso el nombre Ethernet_1 para poder identificarla en futuras configuraciones.

A continuación, podremos ver la pantalla donde le asignaremos la IP que identificará el PLC en nuestra red, cada dirección debe ser válida y única en la red. Debemos elegir en primer lugar la Familia de Modelo, una vez seleccionada aparecerán las opciones correspondientes como muestra la Figura 4-11.

Accedemos al módulo que contiene el puerto Ethernet y podremos corroborar la configuración de la IP asignada como indica la Figura 4-12, y validamos los cambios.

No es necesario configurar parámetros relacionados con la velocidad y la comunicación como ocurría en Modbus, en este caso se utiliza todas las características del protocolo Ethernet aplicando el estándar 802.3 de la norma IEEE⁹ que define las reglas para configurar la red.

⁹ Definición en el Anexo II

The screenshot shows the 'Ethernet_1' configuration window. At the top, there is a dropdown menu for 'Familia de modelo' set to 'CPU 2020, CPU 203, W0100'. Below this are fields for 'Dirección del módulo' (Bastidor, Módulo, Canal) and 'Utilidades del módulo'. The 'Dirección IP del módulo' section includes fields for 'Dirección IP' (192.168.1.1), 'Máscara de subred' (255.255.255.0), and 'Dirección de pasarela' (0.0.0.0). There are tabs for 'Configuración IP', 'Mensajes', 'SNMP', and 'Ancho de banda'. The 'Configuración de dirección IP' section has two radio buttons: 'Configurada' (selected) and 'Desde un servidor'. The 'Configuración Ethernet' section has two radio buttons: 'Ethernet II' (selected) and '802.3'.

Figura 4-11 Pantalla configuración IP.

The screenshot shows the 'Ethernet' configuration window with the 'Configuración' tab selected. On the left, there is a tree view showing 'Ethernet' and 'Canal 3'. The main area is divided into several sections: 'Información de dirección' with fields for 'Desde un servidor' (unchecked), 'Dirección MAC' (00:80:F4:02:B3:53), 'Dirección IP' (192.168.1.5), 'Máscara de subred' (255.255.255.0), and 'Dirección de pasarela' (0.0.0.0). Below this is the 'Mensajes' section with counters for 'Conexiones abiertas', 'Accesos no autorizados', 'Mensajes rechazados', 'Mensajes recibidos', and 'Mensajes enviados', all showing 0. There are buttons for 'Restablecer contadores', 'Ping', and 'Ancho de banda'. The 'Tráfico de mensajes (mens./seg.)' section features a line graph with a peak at 700 and buttons for 'Mín.', 'Media', 'Máx.', and 'Restablecer'. At the bottom, there are dropdown menus for 'Función' (ETH TCP/IP), 'Tarea' (MAST), and 'Conexión de red' (Ethernet_1).

Figura 4-12 Configuración del puerto Ethernet.

5 PROGRAMACIÓN DEL PLC

Para poder llevar a cabo las comunicaciones entre los conmutadores después de realizar la configuración en cada equipo será necesario conocer las diferentes funciones de las que disponemos para poder recoger los datos necesarios para el funcionamiento deseado del PLC y la distribución de la memoria a la que tienen accesos los PLCs entre ellos.

5.1 Distribución de la memoria

Aunque nuestros PLCs dispongan de varios módulos la comunicación entre autómatas solo se puede realizar con los registros de la memoria principal en el preprocesador (P34 2020). Estos registros se pueden ver en la pestaña 'Objetos de E/S' de las propiedades de dicho módulo.

Seleccionamos los registros de la memoria como se puede ver en la Figura 5-1 y seleccionamos actualizar cuadrícula en el botón que aparece al final de todas las opciones.

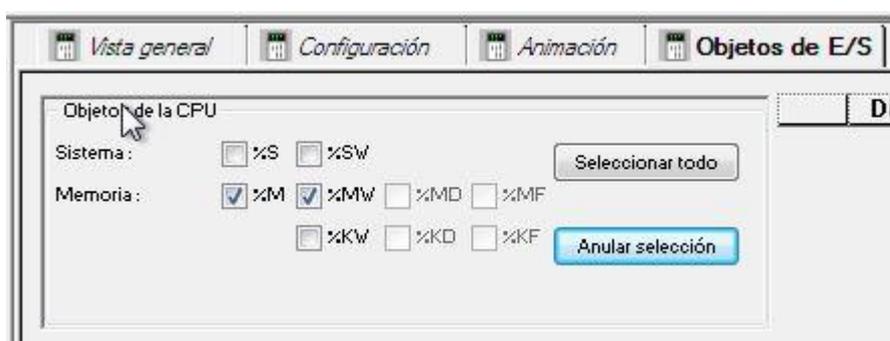


Figura 5-1 Opciones visualizar memoria P34 2020.

Al actualizar la cuadrícula nos aparece a la derecha la memoria de la que disponemos con el tipo de dato y la dirección que le corresponde como indica la Figura 5-2.

Comprobamos que los tipos de datos a los que se puede acceder son de tipo BOOL e INT.

	Dirección	Nombre	Tipo	Comentario
499	%M498		BOOL	
500	%M499		BOOL	
501	%M500		BOOL	
502	%M501		BOOL	
503	%M502		BOOL	
504	%M503		BOOL	
505	%M504		BOOL	
506	%M505		BOOL	
507	%M506		BOOL	
508	%M507		BOOL	
509	%M508		BOOL	
510	%M509		BOOL	
511	%M510		BOOL	
512	%M511		BOOL	
513	%MW0		INT	
514	%MW1		INT	
515	%MW2		INT	
516	%MW3		INT	
517	%MW4		INT	
518	%MW5		INT	
519	%MW6		INT	
520	%MW7		INT	
521	%MW8		INT	
522	%MW9		INT	
523	%MW10		INT	
524	%MW11		INT	

Figura 5-2 Memoria principal.

A la hora de diseñar nuestra solución independientemente del protocolo a utilizar debemos tener claro:

- Las variables que necesitan ser leídas por otro PLC deben estar almacenadas en la memoria principal.
- Si vas a leer de otro autómatas, saber dónde se almacenan las variables que necesitas en el otro PLC y conocer la dirección en la que la vas a guardar los registros leídos en tu memoria principal con el buffer de recepción.
- Si vas a escribir en otro autómatas, conocer la dirección en la que debe ir el dato que te interesa escribir.

En la configuración de Variables e instancias FB podremos declarar las variables y asociarlas a la zona de memoria que nos convenga. Figura 5-3

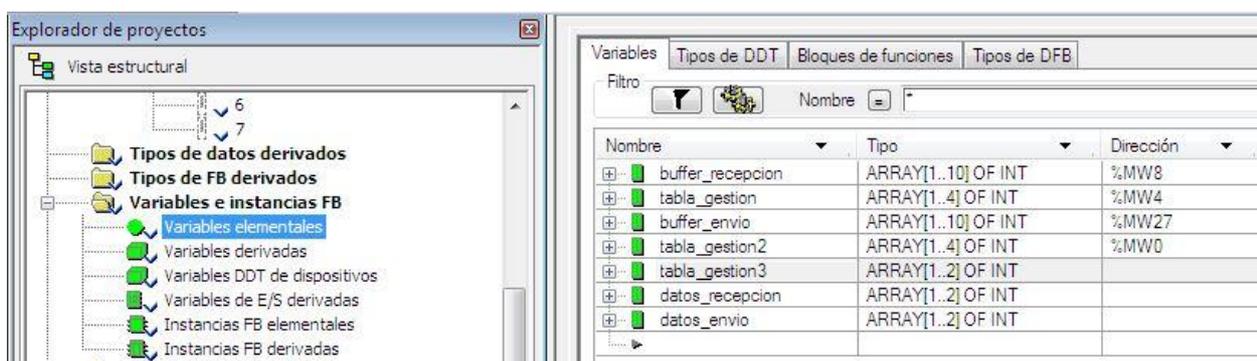


Figura 5-3 Declarar variables.

5.2 Funciones

Para la comunicación entre el maestro y los esclavos normalmente en la programación desde cualquier lenguaje se debería montar la trama como se detalló anteriormente haciendo uso de los diferentes códigos de funciones de los que disponemos.

En este caso, utilizando Unity Pro tenemos las funciones de comunicación `READ_VAR` y `WRITE_VAR` en las que indicándoles los parámetros necesarios generan la trama a enviar sin necesidad de indicarle el Código de Función de Modbus. La tabla 5-1 muestra lo que sería la equivalencia con el Código Modbus del que habría que hacer si no dispusiéramos de Unity Pro.

Básicamente las funciones de Unity Pro nos facilita el tedioso trabajo de montar la trama como se haría en un lenguaje de programación como C o Java.

Por otro lado, es posible enviar cualquier solicitud de Modbus a un dispositivo esclavo mediante la función de comunicación `DATA_EXCH`. Con esta función podremos usar cualquiera Código de Función de Modbus, la desventaja es que es más complicado de usar. Por lo general, la mayoría de los problemas se resuelven sin la necesidad de acudir a esta función.

Petición Modbus	Código de función	Función de comunicación
Lectura de bits	16#01 o 16#02	<code>READ_VAR</code>
Lectura de palabras	16#03 o 16#04	<code>READ_VAR</code>
Escritura de bits	16#0F	<code>WRITE_VAR</code>
Escritura de palabras	16#10	<code>WRITE_VAR</code>

Tabla 5-1 Código de Funciones Modbus equivalentes en Unity Pro.

Básicamente las funciones de Unity Pro nos facilita el tedioso trabajo de montar la trabaja como se haría en un lenguaje de programación como C o Java.

Debemos tener en cuenta que las utilidades de escritura se pueden enviar en modalidad Broadcast, en este caso, no recibe respuesta al emisor.

- `READ_VAR`: Permite al maestro enviar una petición de lectura de valores a un esclavo, para que este devuelva los valores de una zona de memoria o de entradas. En concreto, se pueden leer objetos de tipo `%M`, `%I`, `%MW` y `%IW` de los esclavos, esto es, valores de bit (booleanos) o enteros en formato palabra (Word), de memoria o de entradas.
- `WRITE_VAR`: Permite al maestro enviar valores a un esclavo para que los escriba en una zona de su memoria. En concreto, se pueden escribir valores en objetos de memoria de tipo `%M` y `%MW` de los esclavos, esto es, valores de bit o enteros en formato palabra de la memoria interna.
- `DATA_EXCH`: Permite especificar el código de función de MODBUS que el esclavo debe ejecutar, enviar una tabla de valores enteros a un esclavo, y esperar la recepción de otra tabla de valores enteros desde el esclavo. De este modo, se pueden programar procedimientos específicos de acceso a los parámetros de un dispositivo esclavo determinado, siguiendo el protocolo MODBUS.

Más generalmente, es posible enviar cualquier solicitud de Modbus a un dispositivo esclavo mediante la función de comunicación `DATA_EXCH`.

Las tres funciones están disponibles en los lenguajes FBD (diagrama de bloques funcionales), Ladder (diagrama de contactos) y ST (texto estructurado).

En contraste, un equipo o PLC que se configura como esclavo de MODBUS, atenderá automáticamente las peticiones del maestro del bus, y no requiere una programación especial de las comunicaciones. Su programa solamente deberá encargarse de mantener correctamente actualizadas las direcciones de memoria cuyos valores puede solicitar el maestro, así como de atender los valores de las zonas de memoria donde el maestro escribe.

En el caso de las comunicaciones con el protocolo Ethernet las funciones serán las mismas con la diferencia de que en una red no tiene el modelo Maestro-Esclavo y por lo tanto todos pueden leer y escribir en otro PLC mientras que también pueden ser leídos y escritos por otro equipo de la red.

Es necesario activar antes en Unity Pro las matrices dinámicas en Ajustes de Proyecto dentro del menú Herramientas.

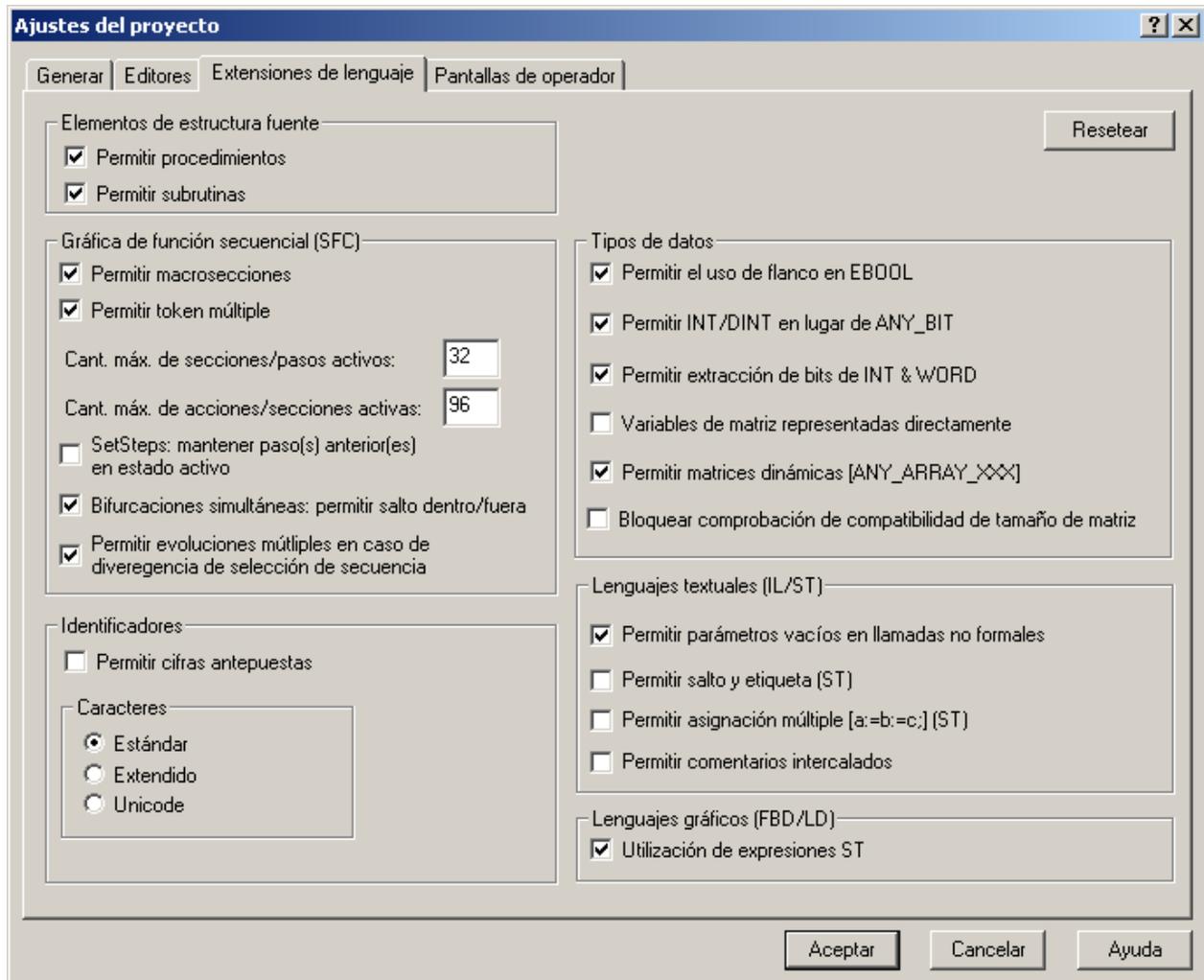


Figura 5-4 Pantalla para permitir matrices dinámicas.

A continuación, se detalla el uso de cada función en Modbus y su símil en Ethernet. Se puede observar que son equivalentes para los dos tipos de protocolos, la principal diferencia es la dirección a utilizar.

5.2.1 Gestión de direcciones

El primer paso para poder trabajar con las funciones para transmisión de datos en el PLC maestro es generar las direcciones de los esclavos a los que se desea acceder con un formato que entienden las funciones de transmisión. Para ello se dispone de la función ADDM. La Figura 5-2 muestra ejemplos del uso del bloque FBD y de la sintaxis de ST para esta función.

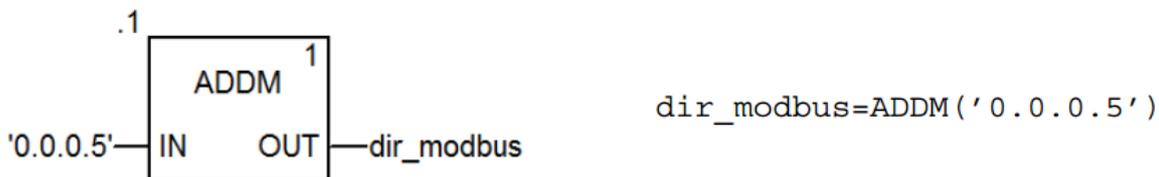


Figura 5-5 Sintaxis de la función ADDM (Modbus).

Las entradas y salidas de esta función son las indicadas en la Tabla 5-2:

Parámetro	Tipo	Descripción
IN	STRING	Dirección de dispositivo en una conexión bus o serie. La sintaxis de la dirección es del tipo 'r.m.c.node'. La dirección se compone de los siguientes parámetros: <ul style="list-style-type: none"> ● r: Número de bastidor del procesador, siempre = 0. ● m: Número de ranura del procesador del bastidor, siempre = 0 ● c: Número de canal, siempre = 0, ya que la conexión serie de un procesador es siempre el canal 0 ● node: Número de esclavo al que se envía la solicitud.
OUT	ARRAY [0..7] OF INT	Matriz que representa la dirección de un dispositivo. Este parámetro se puede utilizar como parámetro de entrada para varias funciones de comunicación.

Tabla 5-2 Parámetros función ADDM (Modbus).

En el caso de la comunicación Ethernet solo cambia la entrada que se debe indicar con el nombre de la red seguido de la IP con la que queremos conectar como podemos ver en la Figura 5-3.

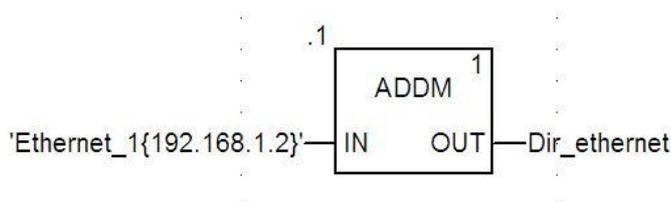


Figura 5-6 Función ADDM (Ethernet).

5.2.2 Lectura de valores

La función READ_VAR permite solicitar valores a un esclavo de MODBUS identificado por su dirección, previamente formateada con la función ADDM. La Figura 5-4 muestra ejemplos del uso del bloque FBD y de la sintaxis de ST para esta función. La *tabla_gestion* se detalla en el apartado 5-3.

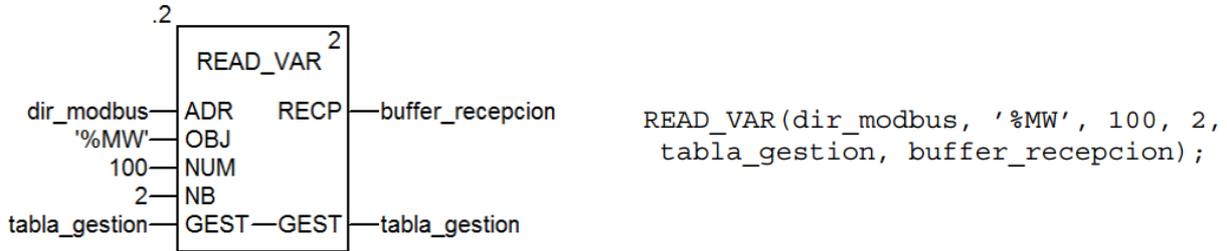


Figura 5-7 Sintaxis de la función READ_VAR.

Al usar esta función, los registros que se quieren leer del esclavo se indican directamente mediante su tipo y ubicación en la memoria del PLC esclavo, y no es necesario tener en cuenta la tabla de registros de dispositivo.

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función ADDM.
OBJ	STRING	Tipo de objeto que se va a leer. Los tipos disponibles son: <ul style="list-style-type: none"> ● %M: bit interno ● %MW: palabra interna ● %I: bit de entrada externa ● %IW: palabra de entrada externa
NUM	DINT	Dirección del primer objeto que se va a leer.
NB	INT	Número de objetos consecutivos que se van a leer.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: <ul style="list-style-type: none"> ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: <ul style="list-style-type: none"> ● Byte de mayor valor: Número de intercambio ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: <ul style="list-style-type: none"> ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	ARRAY [n..m] OF INT	Tabla de palabras que contiene el valor de los objetos leídos.

Tabla 5-3 Parámetros de la función READ_VAR.

La función `READ_VAR` aplicada en una comunicación Ethernet es exactamente igual como podemos comprobar en la Figura 5-5.

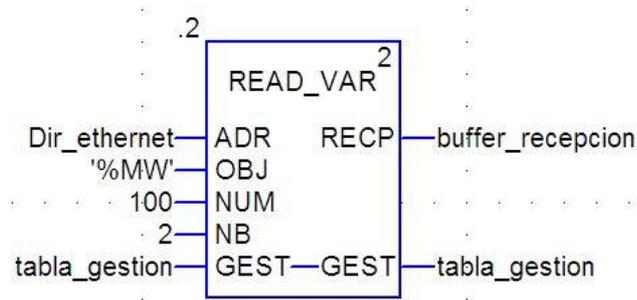
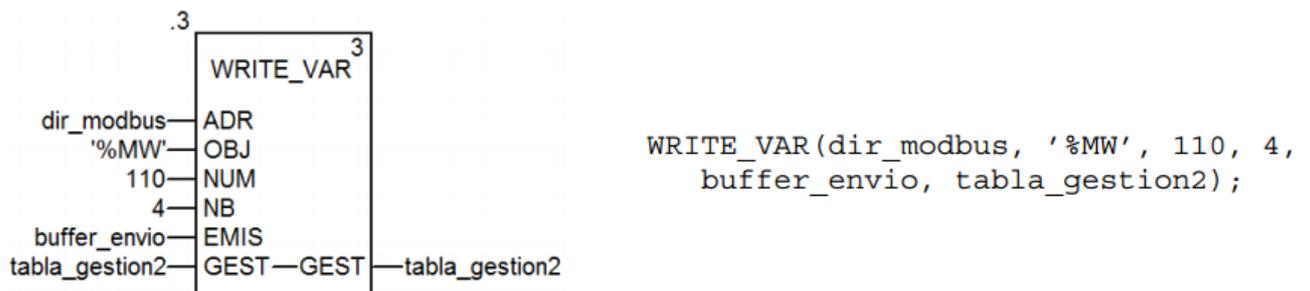


Figura 5-8 Función `READ_VAR` (Ethernet).

5.2.3 Escritura de valores

La función `WRITE_VAR` permite enviar valores a un esclavo de MODBUS identificado por su dirección, previamente formateada con la función `ADDM`. La Figura 5-6 muestra ejemplos del uso del bloque FBD y de la sintaxis de ST para esta función.



```
WRITE_VAR(dir_modbus, '%MW', 110, 4,
          buffer_envio, tabla_gestion2);
```

Figura 5-9 Sintaxis de la función `WRITE_VAR`.

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función ADDM.
OBJ	STRING	Tipo de objeto que se va a escribir. Los tipos disponibles son: <ul style="list-style-type: none"> • %M: bit interno • %MW: palabra interna Nota: WRITE_VAR no puede utilizarse para las variables %I y %IW.
NUM	DINT	Dirección del primer objeto que se va a escribir.
NB	INT	Número de objetos consecutivos que se van a escribir.
EMIS	ARRAY [n..m] OF INT	Tabla de palabras que contienen el valor de los objetos que van a escribirse.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: <ul style="list-style-type: none"> • Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: <ul style="list-style-type: none"> • Byte de mayor valor: Número de intercambio • Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). • Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: <ul style="list-style-type: none"> • Byte de mayor valor: Informe de operación • Byte de menor valor: Informe de comunicación • Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. • Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.

Tabla 5-4 Parámetros de la función WRITE_VAR.

En esta función los registros que se quieren leer del esclavo también se indican directamente mediante su tipo y ubicación en la memoria del PLC esclavo, y no es necesario tener en cuenta la tabla de registros de dispositivo.

Al igual que en la anterior función en una comunicación Ethernet el uso de WRITE_VAR es idéntico que en MODBUS. La Figura 5-7 muestra un ejemplo.

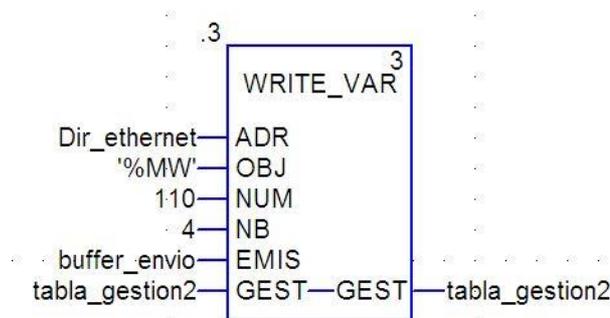


Figura 5-10 Función WRITE_VAR (Ethernet).

5.2.4 Función DATA_EXCH

La función DATA_EXCH permite enviar y recibir valores enteros a un esclavo de MODBUS identificado por su dirección, previamente formateada con la función ADDM. Pero a diferencia de las funciones READ_VAR y WRITE_VAR, con DATA_EXCH se puede especificar el código de la función de MODBUS que el esclavo debe ejecutar. De este modo, se puede acceder a gran variedad de dispositivos que soportan MODBUS. Sin embargo, esta versatilidad hace que el uso de esta función sea más complicado. Por eso, se aconseja el uso de las funciones READ_VAR y WRITE_VAR. La Figura 5-8 muestra ejemplos del uso de esta función con los lenguajes FBD y ST.

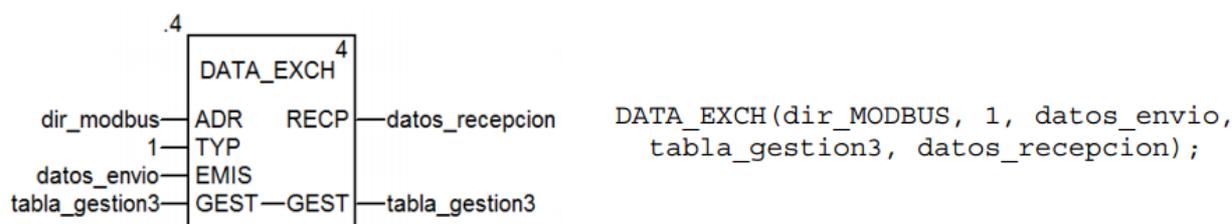


Figura 5-11 Sintaxis de la función DATA_EXCH

Parámetro	Sentido	Tipo	Significado
ADR	Entrada	ARRAY [0..7] OF INT	Dirección proporcionada por la función ADDM
TYP	Entrada	STRING	Para PLC Modicon M340, el único valor posible es 1 , que significa una transmisión del vector EMIS, y, a continuación, esperar a la recepción del vector RECP.
EMIS	Entrada	ARRAY [n..m] OF INT	Vector de enteros que sirve para indicar la función MODBUS que el esclavo debe ejecutar, los registros del esclavo a los que hay que acceder, y los valores que hay que enviar. Debe tener, como mínimo un elemento, aunque no haya datos que enviar. Antes de llamar a la función, hay que especificar el tamaño del array, en bytes , en la cuarta palabra de la tabla de gestión de intercambios
GEST	E/S	ARRAY [0..3] OF INT	Vector que contiene la "tabla de gestión de intercambios" según se describe en el Anexo del apartado 4.1. Cada función de comunicación usada debe tener su propia variable de tabla de gestión.
RECP	Salida	ARRAY [n..m] OF INT	Vector de enteros en donde se depositarán los valores leídos del esclavo. Debe indicarse y tener un tamaño mínimo de 1, aunque la función se use solo para enviar. Después de la recepción, se actualiza la cuarta palabra de la tabla de gestión con el número de bytes recibidos.

Tabla 5-5 Parámetros función DATA_EXCH.

El número máximo de valores que se pueden enviar o recibir es de 256 bytes. La cuarta palabra de la tabla de gestión de intercambios (parámetro GEST) se usa para indicar el número de valores enviados, y para conocer el número de valores recibidos.

El código de función MODBUS que el esclavo debe ejecutar, y la ubicación de los datos que hay que enviarle, se indican a través del vector de palabras en el parámetro de entrada EMIS, que debe organizarse de forma similar a una trama de MODBUS, según el código de función que se ejecute.

Por último, Figura 5-9 muestra un ejemplo de esta función aplicado en una comunicación Ethernet.

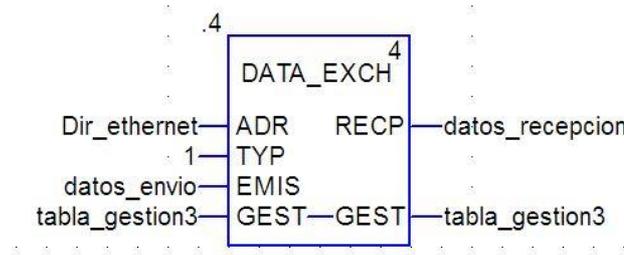


Figura 5-12 Función WRITE_VAR (Ethernet).

En la ayuda de Unity Pro disponemos detalladamente del uso de esta función, a partir de ahora nos centraremos en las utilizadas en nuestro proyecto, READ_VAR, WRITE_VAR y ADDM. [17]

5.3 Tabla de gestión

La tabla de gestión de intercambios en una variable vector compuesta de cuatro palabras enteras (ARRAY [0..3] OF INT), que permite controlar la ejecución y conocer el estado de la transmisión iniciada con funciones de comunicaciones. Cada función de comunicación en el programa debe tener su propia variable de tabla de gestión, ya que varias funciones de comunicación se pueden ejecutar a la vez (hasta 16 en una CPU BMX P34 2020), y cada función debe mantener su propio estado. La tabla de gestión se compone de las siguientes palabras:

- Palabra 1 (índice 0): Palabra administrada por el sistema que se compone de dos bytes:
 - Byte de menor peso: Contiene el Bit de actividad (bit 0), que está a 1 mientras se ejecuta la función, y el Bit de cancelación (bit 1), que se puede poner a 1 para forzar la cancelación anticipada de la transmisión.
 - Byte de mayor peso: Representa el Número de intercambio, que sirve para identificar la transmisión.
- Palabra 2 (índice 1): Palabra administrada por el sistema que se compone de dos bytes:
 - Byte de menor peso: Contiene el *Informe de comunicación*, que especifica el resultado de la operación de transmisión y tiene significado cuando el bit de actividad pasa de 0 a 1. El valor 0 indica una transmisión correcta. El resto de valores indican posibles fallos en la transmisión, conforme se describe en la Tabla X
 - Byte de mayor peso: Contiene el *Informe de operación*, cuyo valor detalla los motivos de posibles fallos en la transmisión, según se describe en la Tabla X
- Palabra 3 (índice 2): Es una palabra gestionada por el usuario, que define el tiempo máximo (Timeout) que el maestro esperará la respuesta desde el esclavo, con una base de tiempo de 100 ms. El valor 0 indica una espera infinita. Debe indicar un valor mayor el resultado de multiplicar los valores configurados como Retardo de respuesta y Numero de reintentos (ver apartado 2.2).
- Palabra 4 (índice 3): Palabra gestionada por el usuario, que sirve para indicar el número de bytes enviados o recibidos con algunas funciones, como DATA_EXCH.

El informe de comunicación es un valor devuelto por las funciones de comunicaciones de un PLC M340, y es significativo cuando el valor del bit de actividad de la función cambia de 1 a 0. Los valores entre 1 y 254 (16#FE) se refieren a errores detectados por el procesador que ejecutó la función. La Tabla 5-6 indica los diferentes valores que pueden obtenerse:

Valor	Significado del informe de comunicación
16#00	Intercambio correcto
16#01	Detención del intercambio al producirse un exceso en el tiempo de espera (timeout)
16#02	Detención del intercambio a petición del usuario (CANCEL)
16#03	Formato de dirección incorrecto
16#04	Dirección de destino incorrecta
16#05	Formato incorrecto de parámetro de gestión
16#06	Parámetros específicos incorrectos
16#07	Problema en el envío al destino
16#08	Reservado
16#09	Tamaño del búfer de recepción insuficiente
16#0A	Tamaño del búfer de envío insuficiente
16#0B	Sin recursos de sistema del procesador
16#0C	Número de intercambio incorrecto
16#0D	Ningún telegrama recibido
16#0E	Longitud incorrecta
16#0F	Servicio de telegramas sin configurar
16#10	Módulo de red ausente
16#11	Petición ausente
16#12	Servidor de la aplicación ya activo
16#13	Número de transacción UNI-TE V2 incorrecto
16#FF	Mensaje rechazado, o intercambio correcto tras utilizar WRITE_VAR en una petición de broadcast MODBUS

Tabla 5-6 Tabla de los valores del informe de operación.

El valor de informe de operación es específico de cada función y especifica el resultado de la operación en la aplicación remota. Sólo es significativo si el informe de comunicación tiene los valores 0 (16#00) o 255 (16#FF).

En las Tablas 5-7 y 5-8 indican el significado de cada valor.

Valor	Significado del informe de operación cuando el informe de comunicación es 0
16#00	Resultado positivo
16#01	Petición no procesada
16#02	Respuesta incorrecta
16#FB	Si existe una respuesta a solicitudes menores
16#FD	Error operativo
16#FE	Si la respuesta es positiva para ciertas solicitudes

Tabla 5-7 Tabla de informe de operación I.

Valor	Significado del informe de operación cuando el informe de comunicación es 255
16#01	No hay recursos respecto al procesador
16#02	No hay recursos de línea
16#03	Sin dispositivo o bien dispositivo sin recursos (para dispositivos extraíbles)
16#04	Error de línea
16#05	Error de longitud
16#06	Canal de comunicación defectuoso
16#07	Error de direccionamiento
16#08	Error de aplicación
16#0B	Sin recursos de sistema
16#0C	Función de comunicación inactiva
16#0D	Destino ausente
16#0F	Problema de acceso entre estaciones o canal sin configurar
16#11	Formato de dirección no gestionado
16#12	Sin recursos de destino
16#14	Conexión no operativa (ejemplo: Ethernet TCP/IP)
16#15	Sin recurso en el canal local
16#16	Acceso no autorizado (ejemplo: Ethernet TCP/IP)
16#17	Configuración de red incoherente (ejemplo: Ethernet TCP/IP)
16#18	Conexión no disponible temporalmente
16#21	Servidor de la aplicación detenido
16#30	Error de transmisión

Tabla 5-8 Tabla de informe de operación II.

5.4 Acceso a las características de un canal de E/S

Si se desea acceder a toda la información de estado de un canal E/S del PLC, es necesario definir una variable IODDT (objeto de tipos derivado de datos de entrada/salida) del tipo adecuado al canal deseado. A través de los campos de esa variable se puede acceder al estado del canal, para detectar posibles errores, o para cambiar alguna configuración durante la ejecución del programa.

Cada IODDT tiene un conjunto de objetos de lenguaje que se usan para controlar y comprobar el funcionamiento del IODDT. Existen dos tipos de objetos de lenguaje:

- implícito: los objetos de intercambio implícito se intercambian automáticamente en cada evolución de ciclo de la tarea asociada con el módulo. Estos intercambios implícitos son aquellos relativos al estado de los módulos, las señales de comunicación, los esclavos, etc.
- explícito: los objetos de intercambio explícito se intercambian a petición de la aplicación mediante instrucciones de intercambio explícito. Estos intercambios establecen parámetros y comprueban el módulo.

Por otra parte, cuando se desea un mayor control por programa de un canal E/S de un PLC, es necesario recurrir a un acceso directo e inmediato a las variables del módulo. Esto es lo que se denomina un intercambio de datos explícito a petición del programa de usuario que ejecuta el PLC, en contraste con el intercambio implícito determinado por el ciclo de lectura de entradas, ejecución de programa y escritura de salidas. Para realizar un intercambio explícito en un M340, se dispone de dos funciones: READ_STS para lectura de palabras de estado y WRITE_CMD para escritura de palabras de comando. Las instrucciones READ_STS y WRITE_CMD se ejecutan al mismo tiempo que la tarea que las llama y siempre correctamente, y su resultado queda disponible automáticamente después de su ejecución.

5.4.1 Acceso a las características del protocolo Modbus

Para el protocolo MODBUS sobre el puerto serie, el PLC M340 admite un tipo de datos IODDT denominado

T_COM_MB_BMX. La forma más sencilla de definir una variable de este tipo para una CPU BMX P34 2020 es crear directamente en la tabla de datos derivados esa variable indicando su nombre, el tipo T_COM_MB_BMX, y la dirección %CH0.0.0, que se corresponde con el canal 0 del puerto serie. La Figura 5-9 muestra la lista de campos que contiene la variable *port*, creada según se ha descrito.

Nombre	Tipo	Dirección	Valor	Comentario
port	T_COM_...	%CH0.0.0		
CH_ERR...	BOOL	%I0.0.0.ERR		Error de canal
INPUT_SI...	INT	%IW0.0.0.0		Señales de entrada
DCD	BOOL	%IW0.0.0.0.0		Detección de portadora de datos
CTS	BOOL	%IW0.0.0.0.2		Listo para transmitir
DSR	BOOL	%IW0.0.0.0.3		Conjunto de datos preparado
LISTEN_...	BOOL	%IW0.0.0.0.8		Modalidad de sólo escucha (sólo esclavo de Modbus)
EXCH_STS	INT	%MW0.0.0.0		Estado de intercambio
STS_IN_...	BOOL	%MW0.0.0.0.0		Lectura de parámetros de estado en curso
CMD_IN_...	BOOL	%MW0.0.0.0.1		Escritura de parámetros de comando en curso
ADJ_IN_P...	BOOL	%MW0.0.0.0.2		Intercambio de parámetros de ajuste en curso
EXCH_RPT	INT	%MW0.0.0.1		Informe de canal
STS_ERR	BOOL	%MW0.0.0.1.0		Error al leer el estado del canal
CMD_ERR	BOOL	%MW0.0.0.1.1		Error al enviar un comando por el canal
ADJ_ERR	BOOL	%MW0.0.0.1.2		Error al ajustar el canal
CH_FLT	INT	%MW0.0.0.2		Fallos de canal
NO_DEVI...	BOOL	%MW0.0.0.2.0		No hay ningún dispositivo funcionando en el canal
ONE_DE...	BOOL	%MW0.0.0.2.1		Un dispositivo del canal tiene fallos
BLK	BOOL	%MW0.0.0.2.2		Fallo externo: Bloque de terminales
TO_ERR	BOOL	%MW0.0.0.2.3		Error de timeout (verificar cableado)
INTERNA...	BOOL	%MW0.0.0.2.4		Fallo interno: Canal inoperativo
CONF_FLT	BOOL	%MW0.0.0.2.5		Fallo de configuración de hardware o software
COM_FLT	BOOL	%MW0.0.0.2.6		Fallo de comunicación de bus
APPLI_FLT	BOOL	%MW0.0.0.2.7		Fallo de aplicación
PROTOCOL	INT	%MW0.0.0.3		6 para maestro Modbus, 7 para esclavo Modbus, 3 para .
CONTROL	INT	%MW0.0.0.24		Protocolo/señales de control
DTR_ON	BOOL	%MW0.0.0.24.8		Terminal de datos preparada Con
DTR_OFF	BOOL	%MW0.0.0.24.9		Terminal de datos preparada Des
TO_MOD...	BOOL	%MW0.0.0.24.12		Conmutar a maestro Modbus
TO_MOD...	BOOL	%MW0.0.0.24.13		Conmutar a esclavo Modbus
TO_CHA...	BOOL	%MW0.0.0.24.14		Conmutar a modalidad de caracteres

Figura 5-13 Figura de los campos de la variable *port* en Modbus.

La columna “comentario” indica la utilidad de cada campo. En particular los siguientes campos son interesantes para nuestro proyecto:

- CH_ERROR. Este campo booleano de lectura se pone a TRUE cuando el protocolo MODBUS detecta algún error. Como es un valor %I que se actualiza de forma implícita, se puede consultar directamente desde el programa, con una sintaxis como PuertoSerie.CH_ERROR. Para conocer más detalles sobre el error hay que consultar otros campos de la variable.
- TO_MODBUS_MAESTRO. Al poner a TRUE este campo, se fuerza el cambio a modalidad de maestro de MODBUS. Pero para acceder a campo de tipo %MW hay que ejecutar una operación explícita. Así, para asignar un nuevo valor, es necesario actualizar primero el campo con PuertoSerie.TO_MODBUS_MAESTRO:=true y después hacer una operación de escritura explícita: WRITE_CMD (PuertoSerie). De forma similar, para leer su valor, primero hay que ejecutar READ_STS (PuertoSerie).
- TO_MODBUS_ESCLAVO. Este campo funciona como el anterior, pero sirve para forzar el cambio a

modalidad de esclavo de MODBUS.

Existe una gran variedad de tipos IODDT y de posiciones de memoria que permiten acceder a propiedades concretas de los canales de E/S, para comprobar su estado y cambiar su configuración en tiempo de ejecución. Más información sobre estos parámetros, se pueden consultar los manuales del PLC o la ayuda de Unity Pro XL.

5.4.2 Acceso a las características del protocolo Ethernet

De forma similar que como se indica en el apartado anterior la mejor forma para definir esta variable es crear directamente una tabla de datos de derivados de tipo T_COM_ETH_BMX.

Esta variable cuenta con numerosos campos por lo que se aconseja mirar el manual de Unity Pro [18].

El puerto Ethernet ofrece un gran abanico de posibilidades, enviar correos, acceder vía web, asignar direcciones mediante DHCP y SNMP. Si se utilizan varios servicios a la vez es necesario controlar las comunicaciones y que no se produzcan varias a la vez del mismo tipo y se solapen datos, por ello es útil dominar la variable T_COM_ETH_BMX

5.5 Información de depuración de comunicación

5.5.1 Depurar comunicación Modbus

Con Unity Pro se pueden monitorizar la información básica de una comunicación serie cuando la aplicación está conectada a una CPU M340 en ejecución.

Esta información está disponible en la pantalla de depuración para MODBUS serie, a la cual se accede a través de la configuración del puerto serie (ver sección 3.4). En esa pantalla, se dispone de dos áreas que muestran el modo de trabajo de la CPU, maestro o esclavo, y un área de contadores como la mostrada en la Figura 5-10.

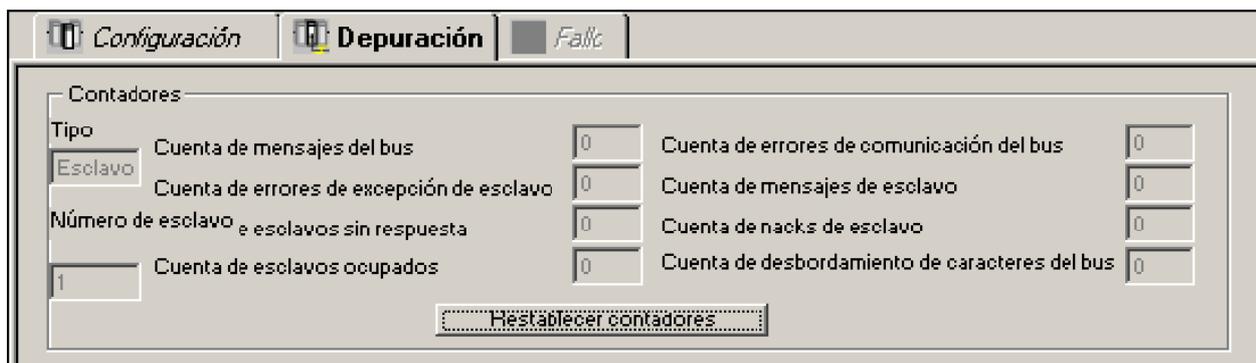


Figura 5-14 Contadores para depuración Modbus.

Los contadores de depuración de la comunicación MODBUS serie son los siguientes:

- Contador de mensajes del bus: este contador indica el número de mensajes que el procesador ha detectado en la conexión serie. No incluye mensajes con un resultado negativo de CRC.
- Contador de errores de comunicación del bus: este contador indica el número de resultados negativos de CRC que ha contado el procesador. Si se detecta un error de caracteres (error de paridad, desborde), o el mensaje es inferior a tres bytes de longitud, el sistema que recibe los datos no puede calcular el CRC. En estos casos, el contador se incrementa en consecuencia.
- Contador de errores de excepción de esclavo: este contador indica el número de errores de excepción

de MODBUS detectados por el procesador.

- Contador de mensajes de esclavo: este contador indica el número de mensajes recibidos y procesados por la conexión MODBUS.
- Contador de esclavos sin respuesta: este contador indica el número de mensajes enviados de los que no ha recibido respuesta (ni una respuesta normal ni una respuesta de excepción). Incluye además el número de mensajes recibidos en modalidad Broadcast.
- Contador de confirmaciones de esclavo negativas: este contador indica el número de mensajes enviados al sistema remoto para los que ha devuelto una confirmación negativa.
- Contador de esclavos ocupados: este contador indica el número de mensajes enviados al sistema remoto para los que ha devuelto un mensaje de excepción de esclavo ocupado.
- Contador de desbordamiento de caracteres del bus: este contador indica el número de mensajes enviados al procesador que no pueden adquirirse debido al desbordamiento de caracteres del bus. El desbordamiento ha sido provocado por uno de estos motivos: Datos de tipo carácter que se transmiten por el puerto serie más rápidamente de lo que pueden almacenarse, o pérdida de datos debida a una anomalía del hardware.

Para todos los contadores, la cuenta comienza en el reinicio, la operación de borrado de contadores o el arranque del procesador más reciente. [19]

5.5.2 Depurar comunicación Ethernet

Se puede depurar el módulo Ethernet examinando los indicadores LED físicos de la parte delantera del módulo o de la pantalla de depuración del software Unity Pro.

En la zona de configuración de nuestro módulo Ethernet y estando en modo conexión podremos ver, como se muestra en la Figura 5-11, información para poder depurar una comunicación Ethernet.



Figura 5-15 Pantalla depuración comunicación Ethernet.

- SMTP: muestra el estado del servidor SMTP (sólo módulos de CPU) en los casos que se esté haciendo uso del mismo.
- Exploración E/S: muestra el estado de cada módulo remoto de entrada y salida.

- Datos globales: muestra el estado de las variables de datos globales

A la izquierda disponemos de un contador similar al de Modbus y una gráfica con el tráfico de los mensajes transmitidos como vemos en la Figura 5-12.



Figura 5-16 Depuración Ethernet, tráfico de mensajes.

Esta ventana informa del número de:

- Conexiones TCP/IP abiertas (el número combinado de servidor, cliente y conexiones de acceso de dispositivo transparente que están abiertas).
- Conexiones TCP/IP no autorizadas.
- Mensajes TCP/IP rechazados.
- Mensajes TCP/IP recibidos.
- Mensajes TCP/IP enviados.

Esta ventana incluye tres botones:

- Restablecer contadores: para restablecer los contadores en 0.
- Ping: para probar el enrutamiento entre el módulo y otro dispositivo mediante una solicitud PING
- Ancho de banda: para ver el número de mensajes recibidos por segundo en la ventana ancho de banda.

5.6 Ejemplo comunicación Modbus

A continuación, se expondrán dos ejemplos básicos de las comunicaciones Modbus y Ethernet. Las funciones a utilizar y la declaración de variables se realizan de la misma manera, el cambio a destacar es que en Modbus deberemos darle la dirección del esclavo mientras que en Ethernet se proporciona la dirección IP de otro PLC.

Se debe tener claro:

- De qué manera tenemos conectados los PLC, con qué configuración y con qué cableado para evitar

problemas de conectividad.

- En una comunicación Modbus solo puede leer y escribir el maestro en los esclavos y no al revés. En Ethernet todos los PLC pueden leerse y escribirse entre ellos.

➤ PLC Maestro

Comenzando por el lado del maestro, lo primero será configurar el PLC como se indicó en la sección 3.4.

Una vez indicado que será el maestro y los parámetros de configuración, el siguiente paso será declarar las variables necesarias para utilizar las funciones indicadas en la sección 5.1. (Figura 5-11).

- Buffer de recepción: almacena lo leído por la función READ_VAR, en este caso vamos a leer en la memoria del esclavo los registros que van desde la posición 100 hasta la 109. En el maestro se guardarán desde la dirección 8 hasta la 17.
- Tabla de gestión: necesaria una por cada función.
- Buffer de envío: almacena los valores a escribir en el PLC esclavo, puede escribirse directamente, el almacenado en la memoria o de una entrada mediante la dirección. En este caso se escribirá en el esclavo en los registros del 50 al 59.

Se recomienda asignar una dirección de memoria a las variables, en este caso se han puesto consecutivas, pero se puede elegir una distribución estratégica según el problema a resolver.

En la Figura 5-12 se representa un ejemplo básico en el que el maestro lee continuamente del esclavo 5 y escribe los valores indicados.

Variables				
Tipos de DDT		Bloques de funciones		Tipos de DFB
Filtro				
Nombre <input type="text" value="*"/>				
Nombre	Tipo	Dirección	Valor	
dir_modbus	ADDM_TYPE			
buffer_recepcion	ARRAY[1..10] OF INT	%MW8		
buffer_recepcion[1]	INT	%MW8		
buffer_recepcion[2]	INT	%MW9		
buffer_recepcion[3]	INT	%MW10		
buffer_recepcion[4]	INT	%MW11		
buffer_recepcion[5]	INT	%MW12		
buffer_recepcion[6]	INT	%MW13		
buffer_recepcion[7]	INT	%MW14		
buffer_recepcion[8]	INT	%MW15		
buffer_recepcion[9]	INT	%MW16		
buffer_recepcion[10]	INT	%MW17		
tabla_gestion	ARRAY[1..4] OF INT	%MW4		
tabla_gestion[1]	INT	%MW4		
tabla_gestion[2]	INT	%MW5		
tabla_gestion[3]	INT	%MW6		
tabla_gestion[4]	INT	%MW7		
buffer_envio	ARRAY[1..10] OF INT	%MW27		
buffer_envio[1]	INT	%MW27	%Q0.3.16	
buffer_envio[2]	INT	%MW28	1	
buffer_envio[3]	INT	%MW29	0	
buffer_envio[4]	INT	%MW30		
buffer_envio[5]	INT	%MW31		
buffer_envio[6]	INT	%MW32		
buffer_envio[7]	INT	%MW33		
buffer_envio[8]	INT	%MW34		
buffer_envio[9]	INT	%MW35		
buffer_envio[10]	INT	%MW36		
tabla_gestion2	ARRAY[1..4] OF INT	%MW0		
tabla_gestion2[1]	INT	%MW0		
tabla_gestion2[2]	INT	%MW1		
tabla_gestion2[3]	INT	%MW2		
tabla_gestion2[4]	INT	%MW3		

Figura 5-17 Variables básicas en el maestro para comunicación Modbus.

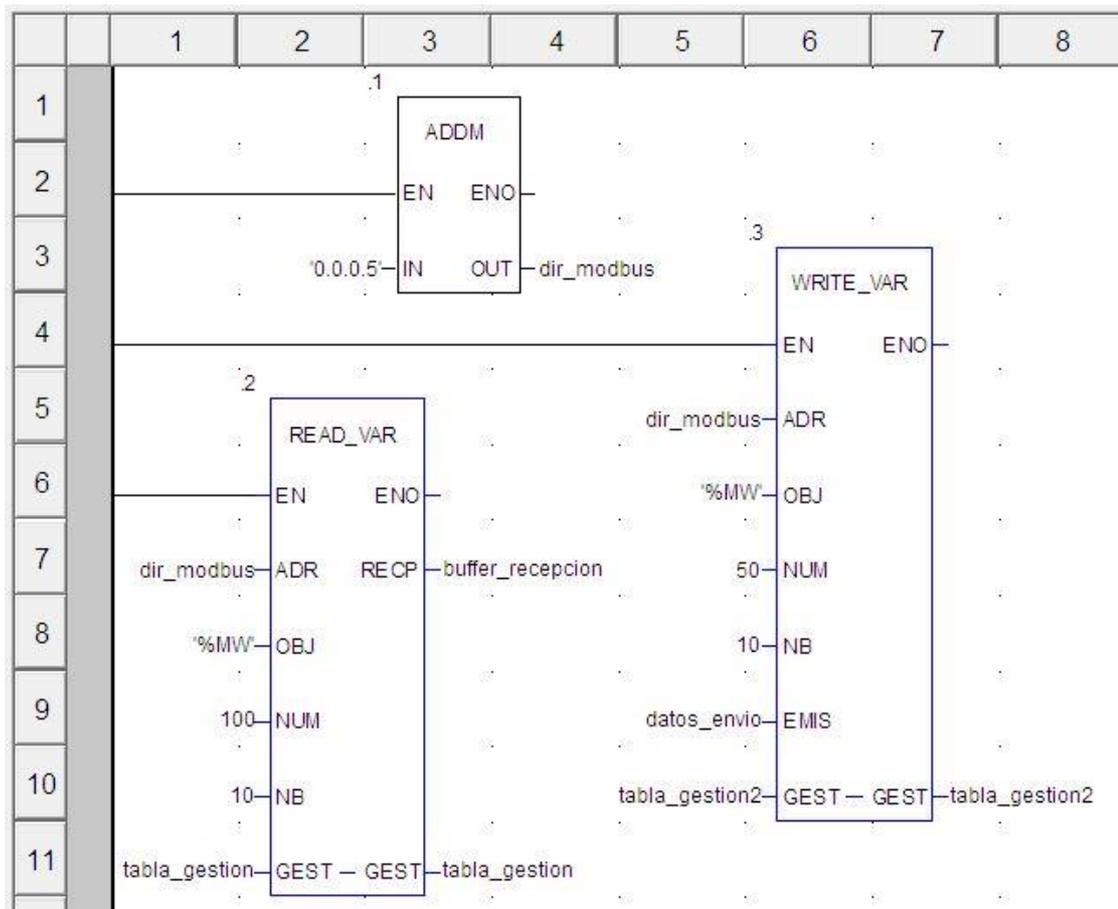


Figura 5-18 Programación Maestro en una comunicación Modbus.

➤ PLC Esclavo

El esclavo solo necesita la configuración indicada en el apartado 3-4, se programará normalmente almacenando las variables que nos interesa que lea el Maestro en la zona de memoria a la que apunta READ_VAR. En este ejemplo en el esclavo se deben almacenar en los registros 100 al 109 de la memoria principal.

5.7 Ejemplo comunicación Ethernet

Como se ha podido observar a lo largo de la memoria existe una gran similitud en la programación de las comunicaciones de los protocolos. En este caso ocurre de igual manera, tras configurar nuestro PLC como indica la sección 4-4 declaramos las variables necesarias para recoger los datos y escribir en otro PLC.

Realizaremos el mismo ejemplo que en Modbus para poder entender de forma clara la base de las comunicaciones entre autómatas utilizando las mismas variables, la única diferencia es la dirección, en este caso se utiliza la IP precedida del nombre de la red creada en la configuración del PLC (Figura 5-13).

Este tipo de protocolo los dos PLCs pueden estar leyendo y escribiendo el uno del otro simultáneamente.

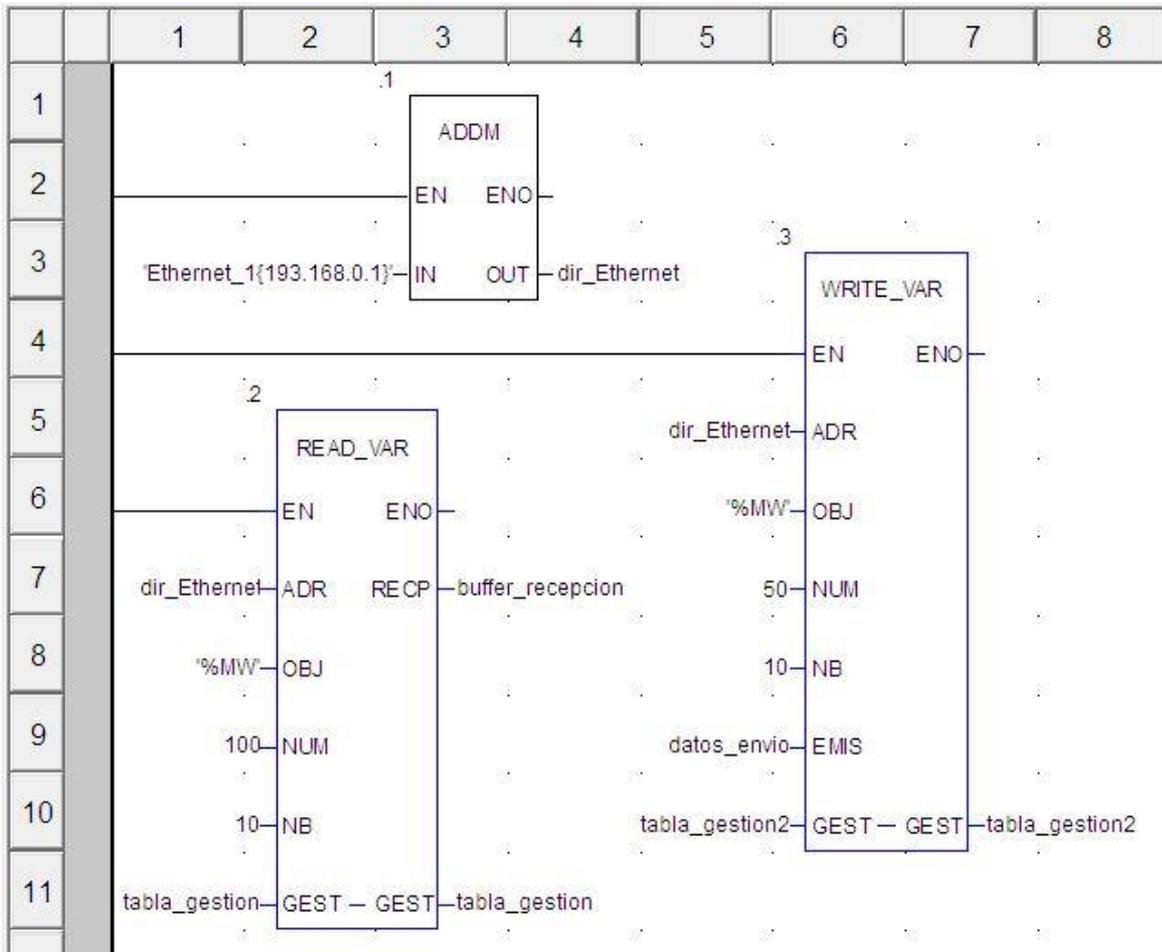


Figura 5-19 Programación PLC en una comunicación Ethernet.

6 IMPLEMENTACIÓN DE UNA RED INDUSTRIAL PARA LABORATORIO

6.1 Objetivo de la red industrial

Pero la mejora en los procesos de automatización pasa hoy en día por el desarrollo de las redes de comunicación. La intercomunicación de sistemas y procesos industriales no es un concepto nuevo, pues es ampliamente conocido el uso de sistemas como IEEE-488¹⁰ y RS485/422 que durante más de 20 años han sido capaces de ofrecer los requerimientos necesarios en las instalaciones de baja y media complejidad en cuanto a las capacidades de intercomunicación se refiere. Este tipo de enlaces entre sistemas se ha empleado esencialmente para equipos de instrumentación y sistemas de automatización, donde es necesaria una baja tasa de transferencia de datos entre equipos, pero que en gran número de casos hoy en día ya no puede responder a las necesidades de intercomunicación entre dispositivos que se demandan. Por ello, redes de comunicación como Ethernet han conseguido ser indispensables en un entorno de trabajo donde cada día es más necesaria la integración global.

Por tanto, los sistemas de automatización industrial y su funcionamiento distribuido forman parte de un concepto de mayor entidad, encaminado a la optimización global de las industrias mediante la mejora de los procesos de fabricación, la reducción de costes, el incremento en la calidad, la mejora de la eficiencia, la mayor flexibilidad en los procesos de producción y, en general, todos aquellos factores que permitan adaptarse de manera eficiente a las necesidades del mercado al que está orientado el producto.

6.1.1 Sistemas industriales de control

Se distinguen tres tipos de sistemas de control industrial: control centralizado, control híbrido y control distribuido. La importancia de las tareas a realizar, o la posibilidad de subdividir la tarea de control del proceso o conjunto de máquinas en esas funciones autónomas, determinará en muchos casos la elección de un tipo u otro de control.

6.1.1.1 Control centralizado

Esta aproximación es la que se sigue en el caso de sistemas poco complejos, donde un proceso puede ser gestionado directamente mediante un único elemento de control encargado de realizar todas las tareas del proceso de producción y que puede incluir un sistema de monitorización y supervisión. Conforme las necesidades de producción han requerido mayor complejidad, una tendencia ha sido la de emplear elementos de control más complejos y potentes, manteniendo en un único elemento todo el control del proceso, con la complejidad que ello supone, ya que se hace necesario hacer llegar todas las señales de sensores y cablear todos los actuadores allá donde se encuentren.

Como ventajas de esta metodología se tiene que no es necesario planificar un sistema de intercomunicación entre procesos, ya que todas las señales están gestionadas por el mismo sistema. Por otro lado, para sistemas poco complejos, posee un menor coste económico. En cambio, posee numerosas desventajas, ya que, si el sistema falla, toda la instalación queda paralizada, siendo necesario un sistema redundante para evitar estas situaciones.

¹⁰ Creado en 1970. El IEEE-488 permite que hasta 15 dispositivos inteligentes compartan un simple bus paralelo de 8 bits, mediante conexión en cadena, con el dispositivo más lento determinando la velocidad de transferencia. La máxima velocidad de transmisión está sobre 1 MB/s en el estándar original y en 8 MB/s con IEEE-488.1-2003

También se hace necesario el empleo de unidades de control (generalmente autómatas programables) de mayor capacidad de proceso, dada la complejidad de los problemas que debe abordar y las restricciones de tiempo límite que son habituales en los procesos industriales. Por otro lado, el cableado puede aumentar notablemente debido a las mayores distancias que pueden existir entre los sensores, actuadores y la unidad de control, aunque este problema se pueda simplificar en cierta medida debido al uso de buses de campo.

6.1.1.2 Control distribuido

La opción de control distribuido requiere que puedan considerarse procesos, grupos de procesos o áreas funcionales susceptibles de ser definidas por un algoritmo de control que pueda realizarse de forma autónoma. A cada unidad se destinará un autómata (o elemento de control) dimensionado de acuerdo con los requerimientos del proceso considerado. Debido a la interdependencia que existe entre las operaciones que tienen lugar en cada proceso, hay que tener en cuenta que es necesario interconectar los autómatas entre sí mediante entradas y salidas digitales, o a través de una red de comunicaciones para intercambio de datos y estados. Por tanto, el autómata o elemento de control evaluado debe permitir las comunicaciones.

Con esta metodología de control es posible que cada unidad funcional consista en un proceso relativamente sencillo comparado con el proceso global, reduciendo la posibilidad de errores en la programación y permitiendo el empleo de unidades de control (autómatas programables principalmente) más sencillas y, por tanto, más económicas. Al mismo tiempo, la existencia de fallos en otras unidades de control no implica necesariamente la paralización de todos los procesos que se llevan a cabo en la planta.

Como desventaja, es necesario realizar un estudio de implantación previo, ya que se deben identificar los procesos autónomos, asignar elementos a cada proceso y diseñar el modelo de intercomunicación para responder a las necesidades del proceso planteado.

6.1.1.3 Control híbrido

El control híbrido no está muy bien definido, ya que este tipo de gestión de planta puede considerarse a cualquier estrategia de distribución de elementos de control a medio camino entre el control distribuido y el control centralizado. En numerosas ocasiones no resulta sencillo separar los procesos de manera completamente autónoma, por lo que se debe recurrir a la gestión de varios procesos desde una misma unidad de control, debido a que la complejidad de la separación es mayor que la complejidad que supone su gestión conjunta.

Por otro lado, una estrategia de este tipo también conduce a una gestión estructurada, de modo que existen elementos de control de nivel superior que supervisan e intercomunican los procesos autónomos más sencillos, siendo los encargados de gestionar la información común. Para este tipo de gestión también es necesario el uso de redes de comunicación.

6.1.2 La pirámide CIM

El ideal de factoría completamente automatizada (Computer Integrated Manufacturing) se representa como una pirámide en la que en los niveles bajos se encuentran los sensores y actuadores; en los niveles intermedios se interconectan estos elementos para funcionar cooperativamente realizando funciones más o menos sincronizadas y finalmente, en el nivel superior aparece la red informática técnico-administrativa donde se recogen informaciones de estado, registros históricos, datos de partida, consignas, etc. (Figura 6-1.)



Figura 6-1 Pirámide CIM de comunicación.

6.1.2.1 Nivel E/S

También llamado nivel de instrumentación. Está formado por los elementos de medida (sensores) y mando (actuadores) distribuidos en una línea de producción. Son los elementos más directamente relacionados con el proceso productivo. Así, los actuadores son los encargados de ejecutar las órdenes de los elementos de control para modificar el proceso productivo y los sensores miden variables en el proceso de producción.

6.1.2.2 Nivel de campo y proceso

En este nivel se sitúan los elementos capaces de gestionar los actuadores y sensores del nivel anterior, tales como autómatas programables o equipos de aplicación específica basados en microprocesador como robots, máquinas herramienta o controladores de motor.

Estos dispositivos permiten que los actuadores y sensores funcionen de forma conjunta para ser capaces de controlar el proceso industrial deseado. Los dispositivos de este nivel de control, junto con los del nivel inferior de entrada/salida, poseen entidad suficiente como para realizar procesos productivos por sí mismos. De hecho, gran cantidad de procesos industriales están basados exclusivamente en estos dos niveles, de tal modo que un proceso productivo completo se desglosa en subprocesos de este tipo sin que exista un intercambio de información entre ellos (excepto algunas señales de control para sincronizar el fin de un proceso con el inicio del siguiente). Son pues dispositivos programables, de tal modo que es posible ajustar y personalizar su funcionamiento según las necesidades de cada caso.

No obstante, a pesar de que puedan presentarse como procesos aislados, esto no implica que no se empleen buses de comunicación, ya que para procesos que requieran de un gran número de sensores y actuadores, es recomendable la utilización de buses de campo para leer el estado de los sensores, proporcionar señales de control a los actuadores y conectar diferentes autómatas programables para compartir información acerca de la marcha del proceso completo. También es importante que estos dispositivos posean unas buenas características de interconexión, para ser enlazados con el nivel superior (Nivel de control), generalmente a través de buses de campo.

6.1.2.3 Nivel de control

Todos los dispositivos de control existentes en planta es posible monitorizarlos si existe un sistema de comunicación adecuado, capaz de comunicar estos elementos con otros tipos de dispositivos dedicados a la gestión y supervisión, que habitualmente están constituidos por computadores o sistemas de visualización tales como pantallas industriales.

En este nivel es posible visualizar cómo se están llevando a cabo los procesos de planta y, a través de entornos

SCADA (Supervisión, Control y Adquisición de Datos), poseer una “imagen virtual de la planta” de modo que ésta se puede recorrer de manera detallada, o bien mediante pantallas de resumen ser capaces de disponer de un “panel virtual” donde se muestren las posibles alarmas, fallos o alteraciones en cualquiera de los procesos que se llevan a cabo.

Mediante este tipo de acciones es posible disponer de acceso inmediato a cada uno de los sectores de la planta. Para ello, resulta imprescindible la conexión con el nivel de control mediante buses de campo o en este caso pueden emplearse redes LAN industriales de altas prestaciones, pues a veces es necesaria la transmisión de importantes cantidades de datos y la conexión con un gran número de elementos de control.

Por ejemplo, en un proceso industrial que consta de varias fases para realizar un determinado producto se utilizan varios autómatas para cada proceso, por lo que un sistema de supervisión debe ser capaz de acceder al estado de cada uno de ellos, visualizar el proceso que lleva a cabo y, de manera global, tener información de cómo está trabajando cada uno individualmente, así como poder acceder a informes generados por el autómata. También es posible modificar los procesos productivos desde los computadores de supervisión. Este nivel sustituye a los grandes paneles y salas de control que durante los años 70 y 80 eran habituales en las grandes empresas.

6.1.2.4 Nivel de gestión

El nivel de gestión estará principalmente constituido por computadores, ya que se encuentra más alejado de los procesos productivos. De hecho, en este nivel no es relevante el estado y la supervisión de los procesos de planta, en cambio, sí adquiere importancia toda la información relativa a la producción y su gestión asociada, es decir, a través del nivel de control es posible obtener información global de todos los niveles inferiores de una o varias plantas.

Con esta información, los gestores de la empresa pueden extraer estadísticas acerca de los costes de fabricación, rendimiento de la planta, estrategias de ventas para liberar posibles excesos de producto almacenado y, en general, disponer de datos que permitan a los niveles directivos la toma de decisiones conducentes a una mejor optimización en el funcionamiento de la planta, todo ello de una manera rápida y flexible. Las comunicaciones con este nivel de la pirámide industrial ya no necesitan ser de tipo estrictamente industrial, es decir, muy robustas, de corto tiempo de respuesta, sino que ahora lo verdaderamente importante son los datos que se transmiten, informes que pueden tener un tamaño medio-grande.

Por este motivo habitualmente se emplean redes de comunicación menos costosas, como redes Ethernet, que se adaptan mejor al tipo de datos que se desean transmitir y, además, permiten la comunicación eficaz entre los diferentes computadores del mismo nivel de gestión. [20]

6.2 Planteamiento de la red industrial en el laboratorio

Para la puesta en práctica de lo desarrollado a lo largo del proyecto se ha implantado en el laboratorio un control distribuido de dos plantas industriales con la ayuda del simulador del que disponemos.

Controlaremos una planta de clasificación de cajas y un procesamiento por lotes de pintura que proporciona una mezcla de color elegida por el usuario utilizando los tres contenedores que la componen (azul, rojo y verde).

La cinta transportadora proporcionará cajas altas y bajas clasificadas previamente, las altas proporcionarán pintura azul y las bajas pintura roja. Para poder realizar la mezcla la planta necesita dos cajas de pintura roja y otros dos azules, la pintura verde se supondrá que se abastece en la propia planta mezcladora.

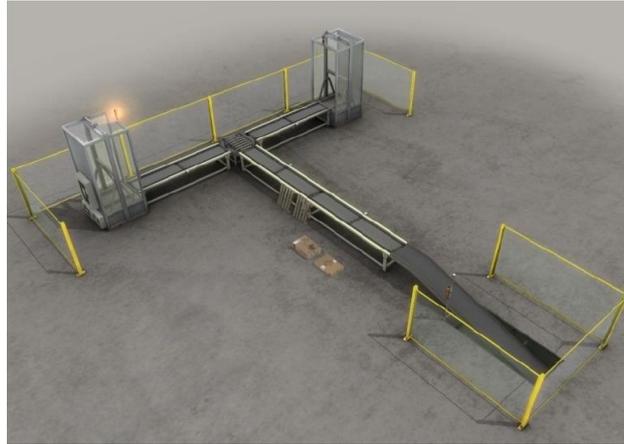


Figura 6-2 Planta clasificadora de cajas.



Figura 6-3 Planta mezcladora de pintura.

6.2.1 Escenario

El laboratorio dispone de 12 PLCs dispuestos en 4 filas con 3 autómatas cada una, las filas estarán unidas por el PLC del extremo que a partir de ahora denominaremos cabeceras.

Los equipos de cada fila se comunican entre ellos mediante el protocolo Modbus utilizando un cableado RS-485, los PLCs situados a la derecha serán denominados a partir de ahora la cabecera de la fila.

Las cabeceras de cada grupo se comunican entre ellas mediante Ethernet, cada una estará conectada al switch utilizando cableado estructurado. (Figura 6-4)

La aplicación práctica se realizará utilizando dos de las filas, una gestionará la planta clasificadora de cajas y la otra la planta mezcladora de pintura, pudiendo coordinarse gracias a las cabeceras.

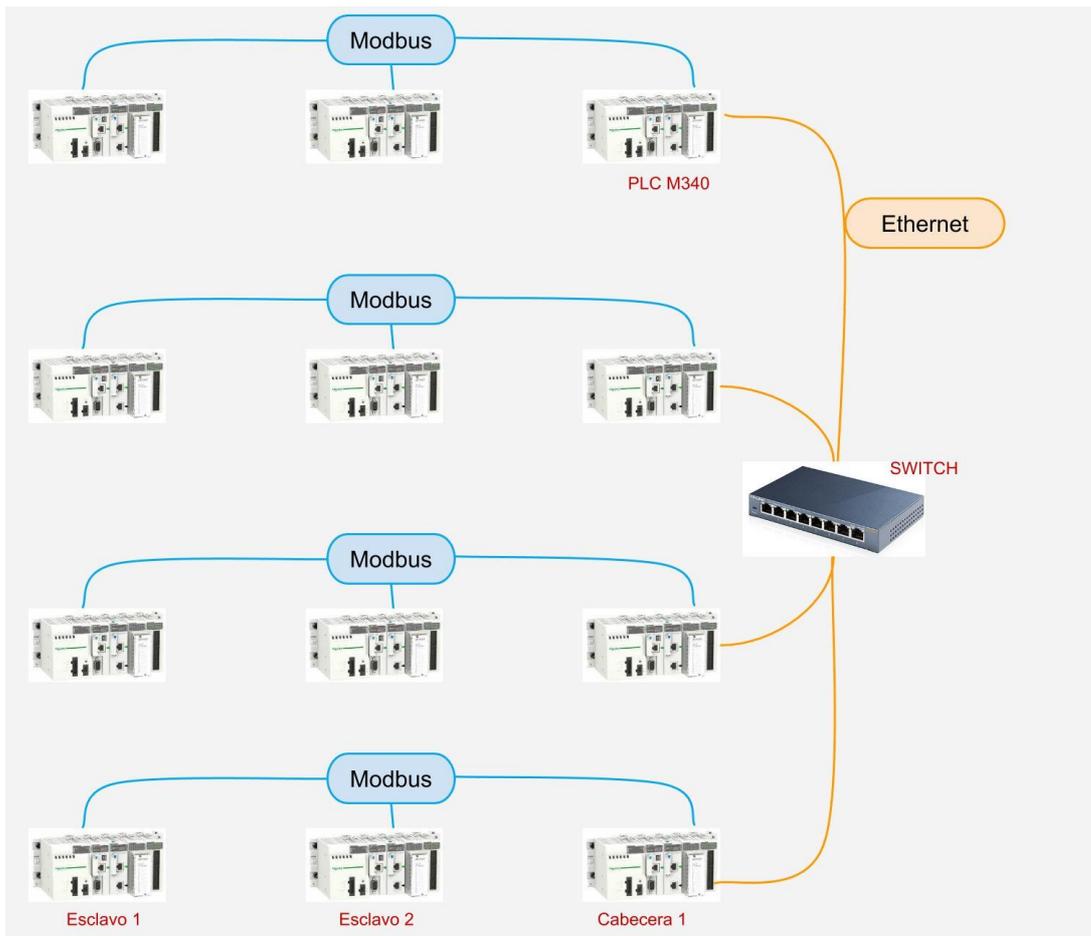


Figura 6-4 Escenario de la aplicación práctica.

6.3 Diseño del problema

A continuación, se asigna la función que tendrá cada autómatas y como cooperarán entre ellos.

6.3.1 Planta clasificadora

Este sistema de clasificación está compuesto por una bahía de entrada, mesas transportadoras y dos bahías de salida.

La banda alimentadora (A) suministra aleatoriamente cajas altas y bajas cargadas en pallets. Los pallets son transportados por las mesas transportadoras (B) a la plataforma giratoria (C), y luego son cargados a través de los rodillos (D). La plataforma giratoria (C) hace rotar los pallets en un ángulo de 90 grados, dependiendo de la altura de las cajas, que es detectada al momento de su entrada a las mesas transportadoras (B). Los pallets son entonces enviados a través de los rodillos (D) a las mesas transportadoras (E o G). Finalmente, los pallets son enviados a los elevadores automáticos (F o H).

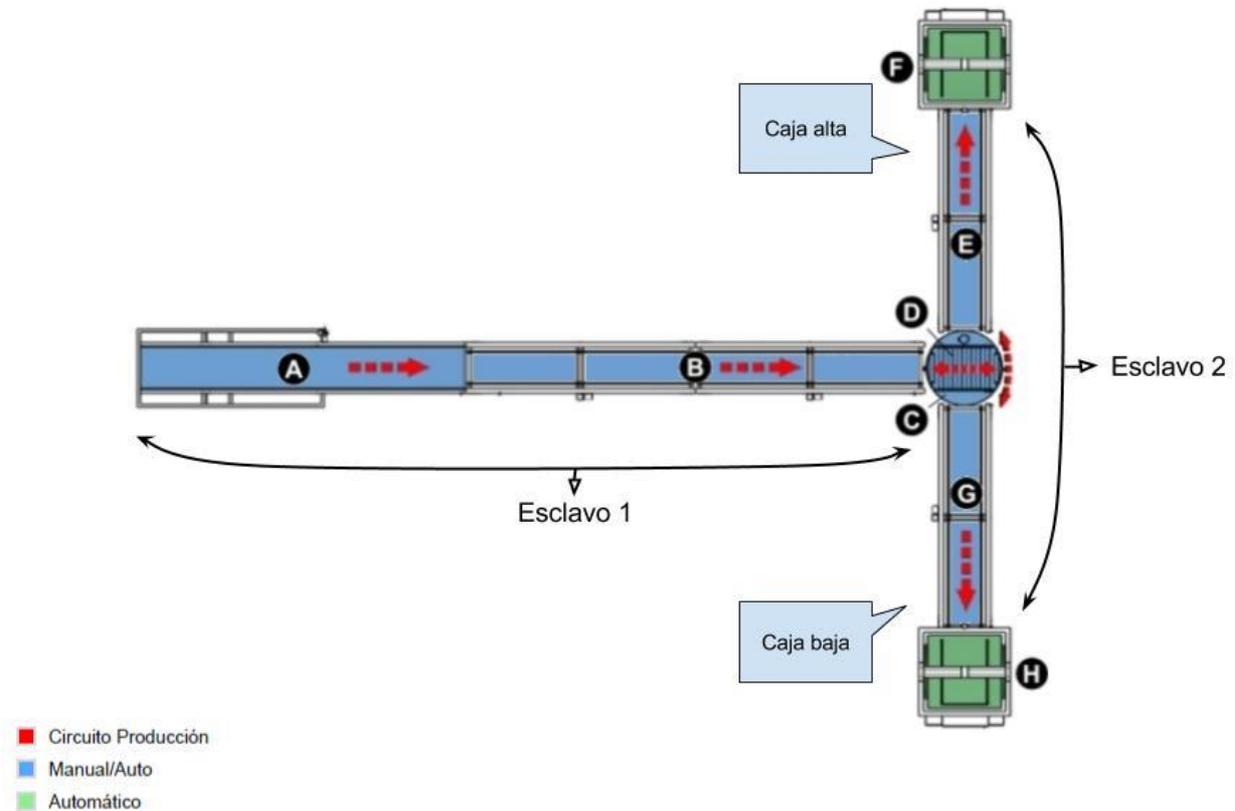


Figura 6-5 Esquemático de la planta transportadora.

En este escenario, el esclavo uno se encargará de los sensores y actuadores de la primera cinta que proporciona las cajas y la segunda que las mueve hasta el rodillo que la coloca en la banda transportadora correspondiente según la altura. El esclavo dos gestionará el rodillo y las cintas transportadoras finales. La figura 6-5 muestra el esquemático.

La cabecera actúa como maestro del esclavo 1 y 2, se encargará de contabilizar las cajas recibidas para cuando se obtengan las necesarias para poder realizar la mezcla avisar a través de la conexión Ethernet a la cabecera de la fila contigua que identifica con su IP.

6.3.2 Planta mezcladora de pintura

El sistema de procesamiento por lotes consta de tres depósitos de pintura, tres tanques medidores y un tanque de mezclado.

Los depósitos de pintura (A, B, C) contienen pintura roja, verde y azul, respectivamente. La pintura de los tanques es descargada a través de las válvulas (D, E, F) a los tanques medidores (G, H, I). Cada uno de estos tanques tiene dos puntos de medición. La pintura contenida en los tanques de medición es descargada a través de las válvulas (J, K, L) al tanque de mezclado (M). Si el volumen de la pintura descargada excede la capacidad del tanque, el excedente es descargado por medio del tubo de purga (O). El proceso de mezclado debe tener una duración mínima de cinco segundos. La pintura final es descargada a través de la válvula (N) al interior del tubo de salida (P).

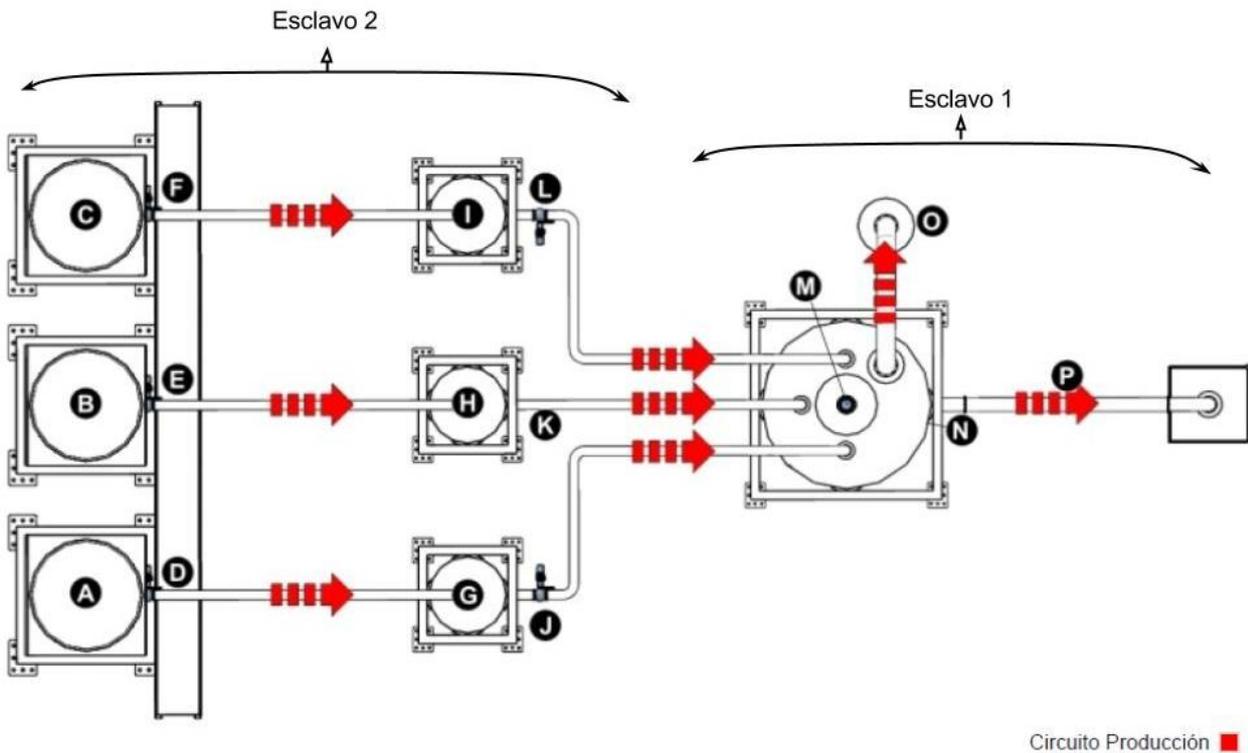


Figura 6-6 Esquemático procesamiento por lotes.

En este caso el esclavo dos se encargará de descargar la pintura a los tanques medidores y controlará las válvulas para llenar el tanque mezclador. El esclavo uno ejecuta el proceso de mezclado y la descarga final.

La cabecera dará la orden al esclavo dos de comenzar el proceso cuando la cabecera de la planta de cajas le informe de que ya se dispone pintura necesaria.

6.3.3 Visión general del escenario

En cada proceso asignado a cada autómatas definimos qué cambio hace que el sistema pase al siguiente paso en el siguiente PLC.

En la Figura 6-7 vemos un diagrama de secuencia en el que se refleja que acción hace que un proceso finalice y comience el siguiente. El simulador se visualiza siempre en los esclavos, uno por ello para que lo que el esclavo dos realice se pueda ver en la simulación, la cabecera debe realizar de intermediario entre los dos esclavos. De esta manera lo que modifique el segundo se podrá visualizar en la pantalla del primero, y poder seguir la simulación.

→ Planta clasificadora.

- Esclavo 1: recibe las cajas, almacena en una variable si es alta o baja y la lleva hasta la plataforma de descarga.
- Esclavo 2: detecta que la caja está en la plataforma de descarga, lee la variable que el maestro le actualiza con la información de la altura que lee del esclavo 1 y la gira para que salga por la cinta correcta para almacenar.
- Cabecera 1: detecta que se ha almacenado una caja, lee la variable que indica la altura y actualiza los contadores que almacenan el número de cajas recibidas.

La cabecera también debe controlar que el proceso que realiza el segundo esclavo se visualice en el primero.

➔ Planta mezcladora de pintura.

- Cabecera 2: lee continuamente los contadores de la cabecera 1, cuando se dispone de las cajas necesarias ordena al esclavo 2 comenzar el proceso.
- Esclavo 2: la cabecera le indica el comienzo del proceso y descarga la pintura en los tanques medidores.
- Esclavo 1: cuando los sensores de los tanques medidores indican que están llenos comienza la mezcla.

Al igual que en la planta clasificadora la cabecera actualizará los actuadores en el esclavo uno para que se pueda visualizar en el simulador.

Al finalizar el mezclado el esclavo 1 actualiza la variable fin de proceso que la cabecera 2 lee y esta se lo comunica a la cabecera 1 para que inicie el proceso en su esclavo 1.

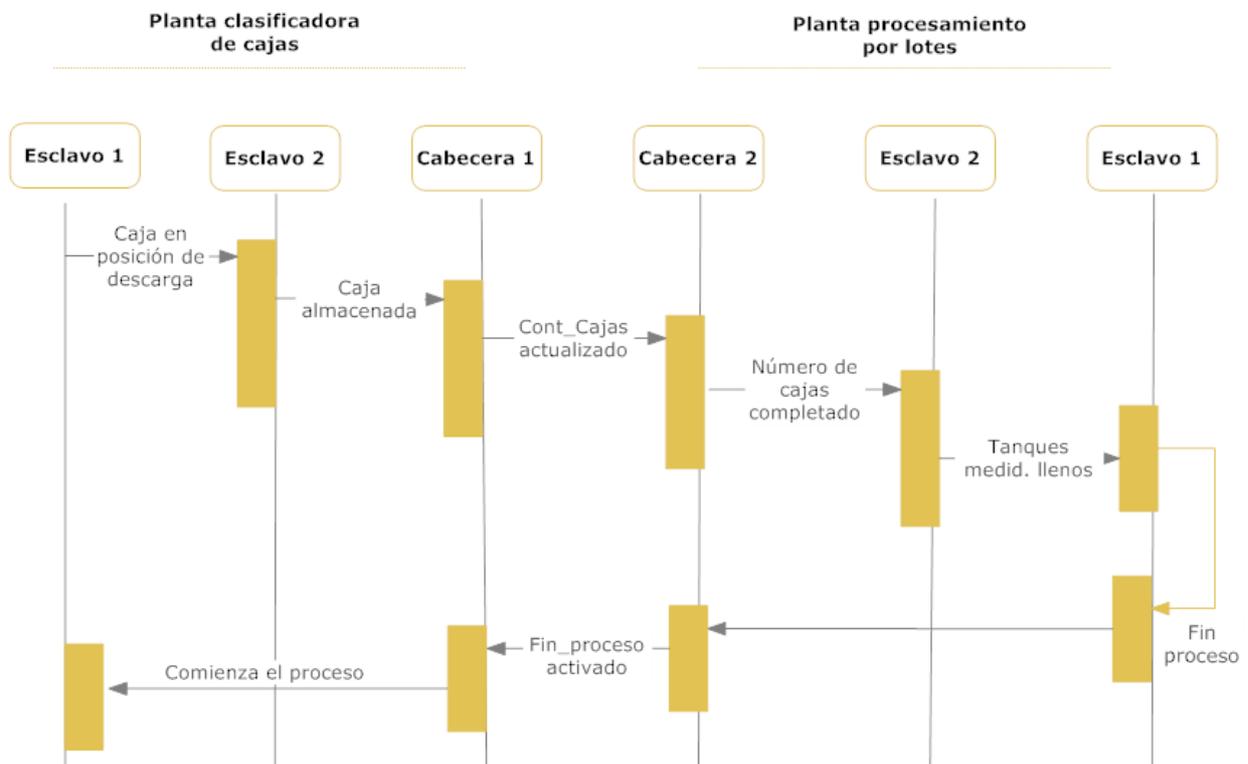


Figura 6-7 Diagrama de secuencia.

6.4 Desarrollo de la solución.

6.4.1 Declaración de variables

A continuación, declararemos las variables que necesitamos para la programación y la comunicación de los autómatas. Previamente deberán ser configurados cada PLC con su IP o el número de esclavo, según el rol asignado.

6.4.1.1 Planta clasificadora de cajas

La planta dispone de sensores y actuadores que se leen gracias al módulo de adquisición de datos y que son guardados en la memoria principal del autómata para que puede ser leída o modificada por otros PLCs cuando sea necesario.

- Esclavos:

Los esclavos uno y dos son los que se comunican a tiempo real con el simulador, les asignaremos en la memoria principal el valor de la dirección de memoria del ITS de los sensores y actuadores.

El esclavo uno también contará con una variable donde almacenará si la caja que transporta es alta o baja, el esclavo 2 tendrá otra variable con la altura que le actualiza el maestro.

Dir. adq. de Datos	Dir. Mem.	Sensor	Descripción
%I0.1.0	%M0	0	Detector de salida de la banda alimentadora.
%I0.1.1	%M1	1	Detector de altura de caja alta.
%I0.1.2	%M2	2	Detector de salida de las mesas transport. de entrada.
%I0.1.3	%M3	3	Detector de posición de carga de plataforma giratoria.
%I0.1.4	%M4	4	Detector de salida de las mesas transport. de entrada.
%I0.1.5	%M5	5	Detector de posición de descarga de plataforma giratoria.
%I0.1.6	%M6	6	Detector de pallet de plataforma giratoria.
%I0.1.7	%M7	7	Detector de entrada de la mesa transportadora de salida.
%I0.1.8	%M8	8	Detector de entrada de la mesa transportadora de salida
%I0.1.9	%M9	9	Detector de salida de la mesa transportadora de salida.
%I0.1.10	%M10	10	Detector de salida de la mesa transportadora de salida.

Tabla 6-1 Sensores planta clasificadora.

Dir. adq. de Datos	Dir. Mem.	Actuador	Descripción
%Q0.1.16	%M11	0	Banda alimentadora.
%Q0.1.17	%M12	1	Mesa transportadora de entrada.
%Q0.1.18	%M13	2	Rodillos de plataforma giratoria (carga).
%Q0.1.19	%M14	3	Rodillos de plataforma giratoria.
%Q0.1.20	%M15	4	Plataforma giratoria.
%Q0.1.21	%M16	5	Mesa transportadora de salida.
%Q0.1.22	%M17	6	Mesa transportadora de salida.

Tabla 6-2 Actuadores planta clasificadora.

- Cabecera: necesitara consultar la variable del esclavo 1 que almacena la altura de la caja que le ha llegado y la actualizará en el esclavo 2, también leerá del esclavo 2 la dirección de memoria %M9 y %M10 para saber cuándo se almacena una caja y actualizar las dos variables que contabilizan las cajas almacenadas. Por otro lado, consultará de la cabecera 2 la variable que indica fin del proceso para poder saber cuándo acaba la mezcla y dar paso a otra caja en el esclavo 1.

6.4.1.2 Planta procesamiento por lotes

En este caso guardaremos de forma similar el valor de los sensores y los actuadores en los esclavos uno y dos que son los que realizarán el proceso cuando le llegue la orden de la cabecera 2.

- Esclavos: los esclavos uno y dos almacenaran en su memoria el valor de la dirección de los sensores y los actuadores del sistema de adquisición de datos. En este caso no hará falta tener variables extras.

Dir. adq. de Datos	Dir. Mem.	Sensor	Descripción
%I0.1.0	%M0	0	Detector de nivel bajo del tanque medidor de pintura roja.
%I0.1.1	%M1	1	Detec.de nivel medio del tanque medidor de pintura roja.
%I0.1.2	%M2	2	Detector de nivel alto del tanque medidor de pintura roja.
%I0.1.3	%M3	3	Detec. de nivel bajo del tanque medidor de pintura verde.
%I0.1.4	%M4	4	Det. de nivel medio del tanque medidor de pintura verde
%I0.1.5	%M5	5	Detec. de nivel alto del tanque medidor de pintura verde.
%I0.1.6	%M6	6	Detector de nivel bajo del tanque medidor de pintura azul
%I0.1.7	%M7	7	Detec. de nivel medio del tanque medidor de pintura azul.

%I0.1.8	%M8	8	Detector de nivel alto del tanque medidor de pintura azul.
%I0.1.9	%M9	9	Detector de nivel bajo del tanque de mezclado.
%I0.1.10	%M10	10	Detector de nivel alto del tanque de mezclado.

Tabla 6-3 Sensores planta procesamiento por lotes.

Dir. adq. de Datos	Dir. Mem.	Actuador	Descripción
%Q0.1.16	%M11	0	Válvula de descarga del depósito de pintura roja.
%Q0.1.17	%M12	1	Válvula de descarga del tanque medidor de pintura roja
%Q0.1.18	%M13	2	Valv. de descarga del depósito de pintura verde.
%Q0.1.19	%M14	3	Válvula de descarga del tanque medidor de pintura verde.
%Q0.1.20	%M15	4	Valv. de descarga del depósito de pintura azul.
%Q0.1.21	%M16	5	Válvula de descarga del tanque medidor de pintura azul.
%Q0.1.22	%M17	6	Mezclador.
%Q0.1.23	%M18	7	Válvula de descarga del tanque de mezclado.

Tabla 6-4 Actuadores planta procesamiento por lotes.

- Cabecera 2: leerá continuamente de la cabecera uno los contadores. Cuando tenga las cajas necesarias para llevar a cabo la mezcla activará el inicio de proceso del esclavo 1. También consulta del esclavo 1 la variable que indica el fin del procesa y que es consultada por la cabecera 1.

Se debe tener en cuenta:

- Para asignar un sensor/actuador a una variable esta debe ser EBOOL.
- Para asignar la variable de un sensor/actuador debe ser a través de una bobina.

Por ejemplo, la válvula de descarga le llamamos VAL_DESCARGA asignada a la dirección %Q0.1.23 del ITS. Ahora necesitamos asignarla a la memoria principal de la siguiente manera:

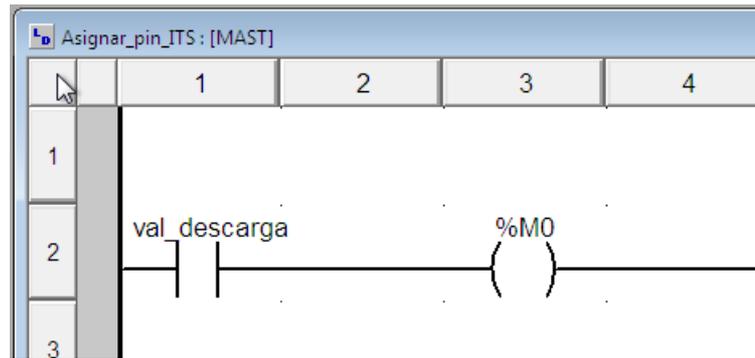


Figura 6-8 Asignar valor ITS en la memoria principal.

- Al leer los datos en la cabecera se almacenan en la tabla_recepcion en un array de enteros. Necesitamos convertir el dato que nos interesa a BOOL para poder asignarle el dato leído a otra variable.

En la Figura 6-9 se observa que convertimos la variable que hemos leído del esclavo que indica que debe comenzar el proceso, al leerlo se almacena en valor que nos interesa en la segunda posición de la tabla_recepc. Para poder utilizarla la convertimos a BOOL con la función INT_TO_BOOL.

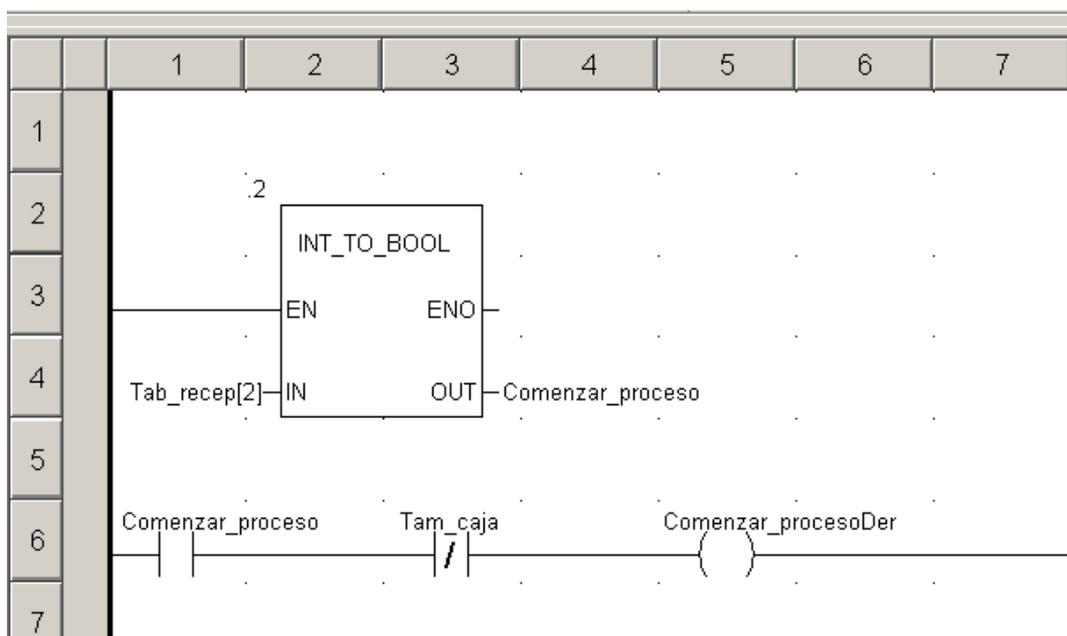


Figura 6-9 Función INT_TO_BOOL.

6.5 Especificaciones de cableado del laboratorio

6.5.1 RS-485. Polarización y terminación de la línea Modbus

El puerto serie puede funcionar con las señales físicas RS-232 o RS-485, y tiene una conexión RJ-45 que incluye las líneas básicas para los dos modos. El conector RJ-45 del puerto serie es el situado en la parte inferior y con una marca negra. Aunque el conector RJ-45 es el mismo que el usado en redes de datos Ethernet, el cableado y las señales eléctricas son diferentes. En nuestro proyecto usaremos el RS485.

Como explicamos en el punto 5.1 la comunicación Modbus serie se utilizará para comunicar cada trio de PLC. Para ello los compañeros del laboratorio tuvieron que realizar el cableado para el bus RS485 de cada fila.

Figura 6-8 muestra la disposición de las conexiones del puerto serie, así como los dos modos de cableado de colores estándar para un conector RJ-45 de Ethernet. Con el puerto serie en modo RS-232, el PLC actúa como equipo DTE (equipo inteligente) y se puede comunicar con un DCE (módem). Así, TD y RTS son señales de salida del PLC, y RD y CTS son señales de entrada. Si se usa el modo RS-485, se dispone de las líneas D1 y D0 (también denominadas A y B, o + y -) para el bus, y de la línea de referencias de señales de 0V (GND, o signal ground). Se observa como las líneas D1, D0 y 0V coinciden, respectivamente, con los cables de color estándar azul, azul/blanco y marrón.

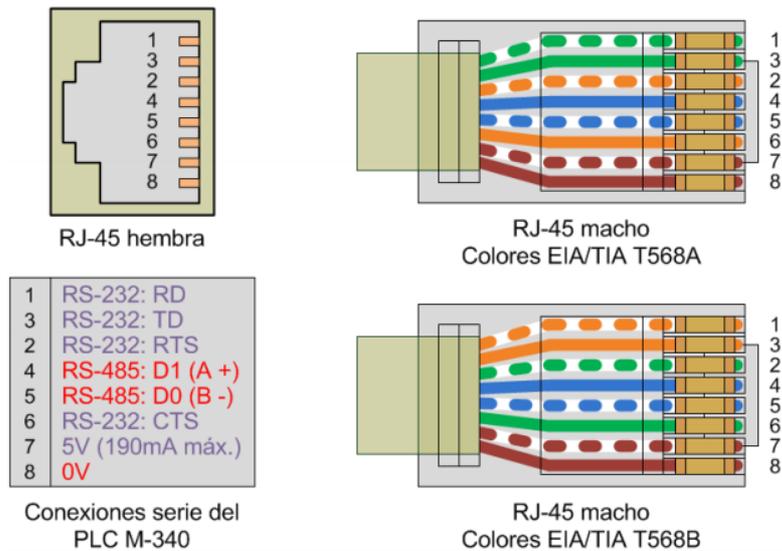


Figura 6-10 Conector RJ-45.

El estándar RS-485 define un bus para la transmisión serie multipunto, donde, en un instante, puede haber un equipo transmitiendo y varios recibiendo. La comunicación es semiduplex, de forma un equipo puede enviar y recibir, pero no a la vez.

El cableado básico consiste en un par de hilos de cobre trenzados sobre el que se transmite una señal diferencial para enviar los bits de datos, que es bastante inmune a las interferencias y admite largas distancias. Además del par trenzado para datos, pueden usarse líneas de 0V y 5V para alimentar dispositivos del bus. Los bits se transmiten mediante una trama asíncrona.

La Figura 6-9 describe como se debe realizar la conexión con RS-485 de tres PLC M340. Se usan las líneas de datos D0 y D1 y la de 0V. También se puede utilizar la de chasis (earth), que se conecta a la malla de blindaje del cable. En los extremos del cable trenzado del bus RS-485 se requiere unas resistencias terminales R_t de 120Ω , que se pueden conectar en serie con una capacidad de $1nF$. Los PLC M-340 no incluyen internamente las resistencias terminales, por lo que esos componentes hay que colocarlos en el cable. La capacidad hace que la resistencia solo tenga efecto con señales de frecuencias altas, esto es, cuando se transmiten datos. Las resistencias R_p , que se encargan de aplicar la tensión de polarización del bus, las pone el PLC configurado como maestro, por lo que no es necesario cablearlas externamente. Finalmente, se puede mejorar la conexión con unas resistencias R_g entre la línea de 0V y los equipos para limitar las corrientes de retorno al común.

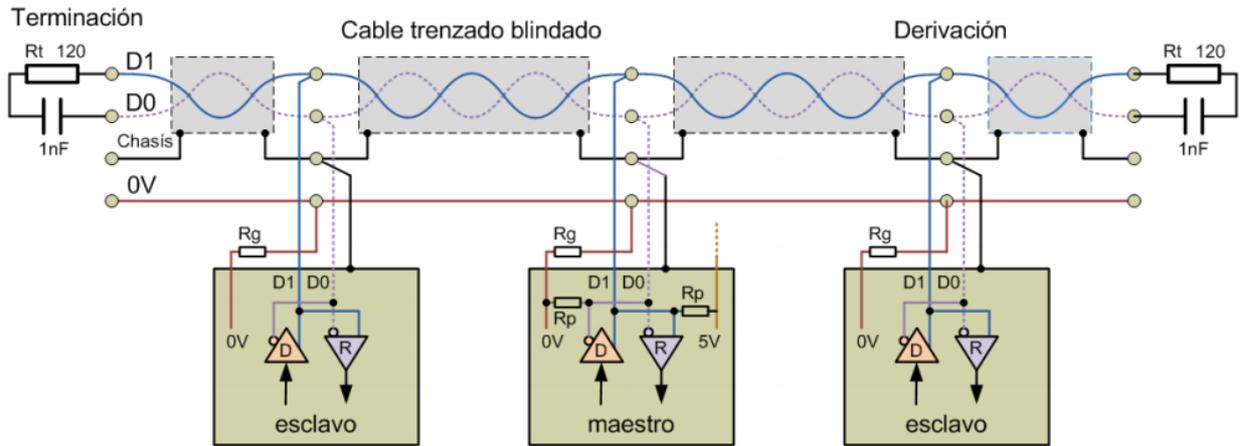


Figura 6-11 Cableado de los interfaces serie con RS-485.

En la realización de esta práctica, para simplificar las conexiones, y dado que las distancias entre equipos serán pequeñas y no habrá un ambiente con ruido electromagnético, se prescindirá de las resistencias R_g , y de la malla del cable. Tampoco se usará la línea de alimentación de 5V, que puede utilizarse para suministrar alimentación a los dispositivos del bus RS-485 sin fuente de alimentación propia. De este modo, en la práctica se montará un cableado como el que se puede ver en el esquema de la Figura 6-10. [19]

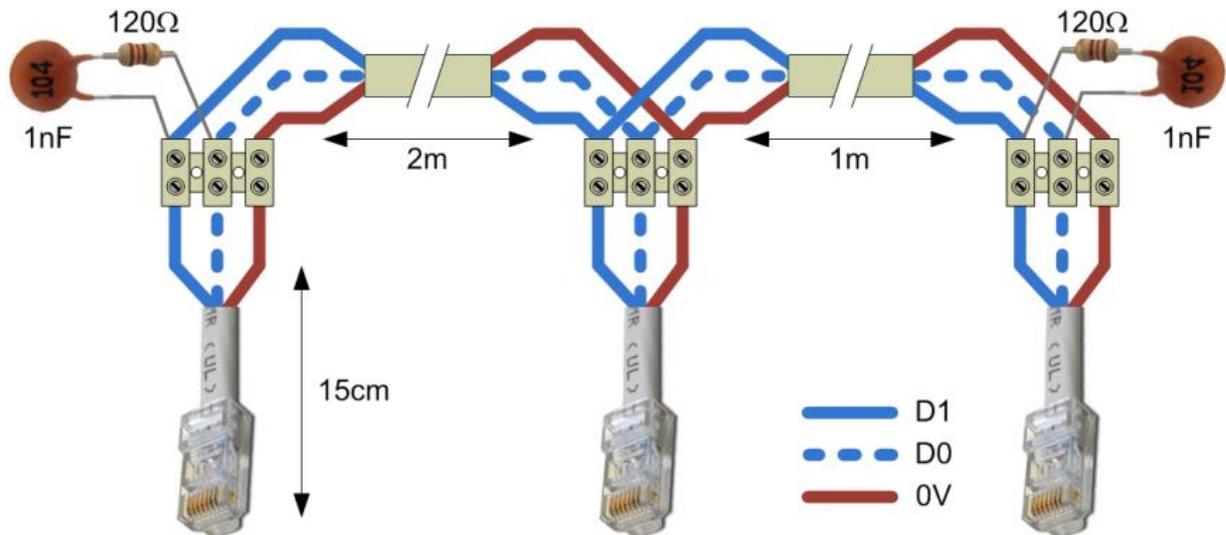


Figura 6-12 Montaje cable bus Modbus RS-485.

6.5.2 Montaje cableado Ethernet.

Para la conexión de las cabeceras se ha utilizado un cable Ethernet, par trenzado sin apantalla 10BASE-T uniendo las cabeceras a un switch.

Los conectores son RJ45 como en Modbus.

Se ha utilizado un switch del modelo X que dispone de X puertos. Para esta parte del montaje no se necesita tener en cuenta nada especial, ni ha sido necesario diseñar el cableado.



7 CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

7.1 Conclusiones

El uso de redes de comunicación en arquitecturas de automatización industrial incrementa su flexibilidad de tal manera que pueden cumplir los requerimientos para adaptar máquinas o plantas. Ello implica hacer decisiones que requieren de un conocimiento específico de las soluciones adecuadas entre una amplia gama de redes de comunicación. Deben utilizarse criterios simples: los productos deben ser abiertos, estandarizados y adecuados.

- Una red abierta, lo contrario a una red propietario, permite la libre elección del fabricante de dispositivos de automatización.
- Una red estandarizada internacionalmente garantiza su durabilidad y posibilidad de actualización y mejora.
- Una elección equilibrada entre los requerimientos de la máquina o de la planta y las prestaciones de la red es la manera de optimizar la inversión.

Este último punto es el que requiere, evidentemente, un conocimiento exacto de la oferta de redes de comunicación, que tradicionalmente se tenían como complicadas de seleccionar, implementar y mantener.

En nuestro proyecto hemos utilizado autómatas Modicon que ofrece redes abiertas basadas en normas internacionales y adaptadas a los requerimientos de todos los niveles de las arquitecturas de automatización, donde hay definidas clase de implementación para una elección simple y óptima.

7.2 Líneas futuras de trabajo

Como líneas futuras de trabajo se plantean:

- Establecer toda la red con comunicaciones Ethernet y aprovechar las ventajas que el autómata nos ofrece en esta modalidad, principalmente controlar todos los autómatas desde un único PC. Esto es útil a la hora de controlarlos todos desde un solo puesto, cargar los programas en cada PLC sin tener que desplazarse, puede haber comunicación en entre todos al contrario que en el modo Esclavo-Maestro, poder tener un mayor número de autómatas en nuestra red.
- Poder instalar un sistema SCADA desde el que gestionar toda la red de forma gráfica.
- Establecer un sistema de alarma ante errores desde una centralita.

8 BIBLIOGRAFÍA

- [1] [«http://www.equiposdidacticos.com/pdf/catalogos/Manual_Unity.pdf»](http://www.equiposdidacticos.com/pdf/catalogos/Manual_Unity.pdf) [En línea].
- [2] Guía del usuario. ITS PLC. Professional edition..
- [3] [«http://www.miinet.com/Portals/0/articles/Using_MODBUS_for_Process_Control_and_Automation.pdf»](http://www.miinet.com/Portals/0/articles/Using_MODBUS_for_Process_Control_and_Automation.pdf) [En línea].
- [4] [«http://www.modbus.org/specs.php»](http://www.modbus.org/specs.php) [En línea].
- [5] [«https://www.amplicon.com/docs/white-papers/modbus-in-process-control.pdf»](https://www.amplicon.com/docs/white-papers/modbus-in-process-control.pdf) [En línea].
- [6] [«http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf»](http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf) [En línea].
- [7] [«http://www.bairesrobotics.com.ar/data/protocolo%20modbus.pdf»](http://www.bairesrobotics.com.ar/data/protocolo%20modbus.pdf) [En línea].
- [8] J. Bartolomé, «El protocolo MODBUS».
- [9] [«http://www.tecnical.cat/PDF/Omron/PLC/CJ/GR_MODBUS_TCP.pdf»](http://www.tecnical.cat/PDF/Omron/PLC/CJ/GR_MODBUS_TCP.pdf) [En línea].
- [10] [«http://www.m-system.co.jp/mssenglish/service/emmodbus.pdf»](http://www.m-system.co.jp/mssenglish/service/emmodbus.pdf) [En línea].
- [11] [«https://en.wikipedia.org/wiki/RS-485»](https://en.wikipedia.org/wiki/RS-485) [En línea].
- [12] P. Mariño, Las Comunicaciones En La Empresa: Normas, Redes y Servicios..
- [13] N. G. Forero, «Normas de Comunicación en Serie: RS-232, RS-422 y RS-485».
- [14] S. Electric, «Conexión serie. Manual de usuario.».
- [15] [«http://www.simplymodbus.ca/TCP.htm»](http://www.simplymodbus.ca/TCP.htm) [En línea].
- [16] [«https://es.wikipedia.org/wiki/IEEE_802.3»](https://es.wikipedia.org/wiki/IEEE_802.3) [En línea].
- [17] M. A. T. Farfán, «Red Modbus/TCP».
- [18] A. J. B. Piña, [«http://uhu.es/antonio.barragan/content/5topologias»](http://uhu.es/antonio.barragan/content/5topologias) [En línea].
- [19] [«http://www.zonasystem.com/2012/03/diferencias-categorias-y-tipos-de.html»](http://www.zonasystem.com/2012/03/diferencias-categorias-y-tipos-de.html) [En línea].
- [20] S. Electric, «Modicon M340 para Ethernet. Procesadores y módulos de comunicaciones. Manual de usuario.».

- [21] F. A. C. Herías, «Comunicación con RS-485 y MODBUS,» Universidad de Alicante.
- [22] J. M. H. Torres, Introducción a las redes de comunicación industrial..
- [23] «<http://uhu.es/antonio.barragan/content/modbus-tcp>,» [En línea].
- [24] J. Á. G. C. Javier Gámiz Caro, «Ethernet como soporte de sistemas de control en red.».

9 ANEXO I: FUNCIONES MODBUS

Modbus Function Codes	Descriptions
16#01	Read Coil Status
16#02	Read Input Status
16#03	Read Holding Registers
16#04	Read Input Registers
16#05	Force Single Coil
16#06	Preset Single Register
16#07	Read Exception Status
16#08	Diagnostics
16#09	-
16#10	-
16#11	Fetch Communication Event Counter
16#12	Fetch Communication Event Log
16#13	Program Controller*
16#14	Poll Controller*
16#15	Force Multiple Coils
16#16	Preset Multiple Registers
16#17	Report Esclavo ID
16#18	Program 884/M84
16#19	Reset Comm. Link*
16#20	Read General Reference
16#21	Write General Reference
16#22	Mask Write 4X Register*
16#23	Read/Write 4X Registers*
16#24	Read FIFO Queue*

9.1 Direcciones

Generalmente en MODBUS cada tipo de dato se mapea en un rango de memoria concreto:

@1-10000 (DOs - digital outputs): 1 bit por dirección para indicar el estado de una salida, mando o relé (0 desactivado, 1 activado). Las direcciones de este rango se suelen acceder mediante las funciones 1 (lectura), 5 (escritura), 15 (escritura múltiple).

@10001-20000 (DIs - digital inputs): 1 bit por dirección para leer el estado de una entrada digital (0 desactivada, 1 activada) también denominadas DIs (Digital Inputs). Las direcciones de este rango se suelen acceder con la función 2 (lectura) y llevan implícita la dirección 10001 como dirección base (para acceder a una dirección bastará con especificar la distancia entre esta y la dirección base).

@20001-30000: el protocolo MODBUS estándar no hace uso de este rango de direcciones.

@30001-40000 (AIs - analog inputs): 16 bits por dirección con el estado de las medidas o entradas analógicas también denominadas AIs (Analog Inputs). Dependiendo del dispositivo este puede hacer uso de más de un registro para almacenar la información de la medida, así con 2 registros consecutivos podríamos almacenar medidas de 32 bits. Las direcciones de este rango se acceden mediante la función 4 (lectura) y llevan implícita la dirección 30001 como dirección base (para acceder a una dirección bastará con especificar la distancia entre esta y la dirección base).

@40001-50000 (AOs - analog outputs): 16 bits con los registros de salidas analógicas o de propósito general (Output Registers – Holding Registers). Se acceden con las funciones 3 (lectura), 6 (escritura) o 16 (escritura múltiple) y llevan implícita la dirección 40001 como dirección base (para acceder a una dirección bastará con especificar la distancia entre esta y la dirección base).

Algunos fabricantes expresan la dirección de forma compuesta, separando la dirección de palabra y la dirección de bit: p.ej word 0x30 bit 1

Como se cita en el apartado de “Código de Operación o Función”, cuando se produce un error en la ejecución de un comando en el esclavo, este responde poniendo a 1 el bit de más peso del código de función (0x80). Con este bit el maestro sabe que se ha producido un error, pero para obtener más detalle sobre el tipo de error, ha de comprobar el campo de datos:

Código	Nombre	Significado
01	ILLEGAL FUNCTION	El código de función recibido no se corresponde a ningún comando disponible en el esclavo
02	ILEGAL DATA ADDRESS	La dirección indicada en la trama no se corresponde a ninguna dirección válida del esclavo
03	ILLEGAL DATA VALUE	El valor enviado al esclavo no es válido
04	SLAVE DEVICE FAILURE	El esclavo ha recibido la trama y la ha comenzado a procesar, pero se ha producido algún error y no ha podido termina la tarea.
05	ACKNOWLEDGE	El esclavo ha recibido la trama y la está procesando pero esto le llevará un periodo un poco largo. Mediante esta respuesta se evita que el máster considere un error de Timeout. El máster podrá enviar más tarde una trama una trama de tipo Poll Program Complete para verificar si ha completado el comando
06	SLAVE DEVICE BUSY	El esclavo está ocupado realizando otra tarea y no puede atender a esa petición en ese instante por lo que el máster tendrá que reintentarlo más adelante.

9.2 Función 1 o 2:

(1 Read Coil Status - 2 Read Input Status)

Permite realizar la lectura del estado de las DIs (@1XXXX el comando 2-Read input status) o DOs (@0XXXX el comando 1-Read Coil Status). Para ello el maestro solicita el número de bits que desea leer a partir de una determinada dirección. Cada dirección se corresponde con un registro de 1 bit con el estado de la entrada digital. El esclavo responde indicando el número de bits que retorna y sus valores. En la trama de respuesta se aprovechan todos los bits del byte, y puede haber hasta 256 bytes.

Petición del máster (modo RTU):

NºEsclavo
Código Operación: 0x01 o 0x02
Dirección del registro (de 1 bit) a leer H
Dirección del registro (de 1 bit) a leer L
Nº de bits que se desea leer H
Nº de bits que se desea leer L
CRC(16): H
CRC(16):

Respuesta del esclavo (modo RTU):

NºEsclavo
Código Operación: 0x01 o 0x02
Nº bytes leídos: 1 byte
Octetos: máx. 256 bytes
CRC (16): H
CRC (16): L

9.3 Función 3 o 4

(3 Read Holding Registers – 4 Read Input Registers)

Permite realizar la lectura del valor de las AIs (@4XXXX el comando 3 Read Holding Registers) o AOs (@3XXXX el comando 4 Read Input Registers). El máster indica la dirección base y número de palabras a leer a partir de esta, mientras que el esclavo indica en la respuesta el número bytes retornados, seguido de estos valores. Aunque en realidad se está escribiendo en el rango de registros o valores numéricos, los registros son direccionados a partir de la dirección 0 (así el registro @40001 se direcciona 0)

Petición del máster (modo RTU):

NºEsclavo
Código Operación: 0x03 o 0x04
Dirección del registro:
Nº de datos que se desea leer: máx. 128 datos
CRC(16): H L

Respuesta del esclavo (modo RTU):

NºEsclavo
Código Operación: 0x03 o 0x04
Nº de bytes leídos: 1 byte
Datos: ¿máx.128 datos ?
CRC (16): H L

Ej 1:

P:[01][03][01][00][00][06][C4][34]

R:[01][03][0C][20][81][00][00][00][00][00][00][00][1C][01][76][F1]

Ej 2:

P:[0A][04][00][00][00][0A][71][76]

0A numero de periférico

04 función de lectura

00 00 registro donde se va a comenzar la lectura

00 0A número de registros a leer: 10

71 76 CRC

R:[0A][04][14][00][00][08][4D][00][00][23][28][00][00][0F][A0][00][00][00][90][00][00][00][60][CB][2E]

0A Número del periférico que responde, 10 en decimal

04 Función de lectura - la que se ha utilizado en la pregunta

14 Número de bytes recibidos (20).

00 00 08 4D V1x 10 (registro 00 Hex) con valor en decimal 212,5 V

00 00 23 28 mA 1, en decimal 9000 mA

00 00 0F A0 W 1, en decimal 4000 W

00 00 00 90 varL 1, en decimal 144 varL

00 00 00 60 PF1 x 100, en decimal 96

CB 2E CRC

9.4 Función 5

(Force Single Coil):

Permite modificar el estado de una DO del esclavo (mando o relé) . Es decir mediante este comando podemos modificar algún bit de alguna de las variables internas del esclavo u ordenar la ejecución o activación de un mando. Actúa sobre la zona de memoria de los DOs @0XXXX . El Maestro especifica la dirección del bit o mando que quiere modificar seguido de 0x00 para ponerlo a 0 o 0xFF para ponerlo a 1. El esclavo responde con una trama similar indicando la dirección que ha modificado y el valor que ha establecido en el bit o mando.

Petición del máster (modo RTU):

NºEsclavo

Código Operación: 0x05

Dirección del bit:

Estado para el bit: 0xXX (0x00 : 0 , 0xFF : 1)

Byte a 0x00

CRC(16): H L

Respuesta del esclavo (modo RTU):

NºEsclavo

Código Operación: 0x05

Dirección del bit:

Estado para el bit: 0xXX (0x00 : 0 , 0xFF : 1)

Byte a 0x00

CRC(16): H L

Ej 1 :

P:[01][05][00][01][FF][00] [DD][FA]
R:[01][05][00][00][FF][00][8C][3A]

Ej 2:

P:[01][05][00][00][00][00][CD][CA]
R:[01][05][00][01][00][00] [9C][0A]

9.5 Función 6

(Preset Single Register) :

Permite la escritura en las AOs del esclavo (ya sea una señal o valor interno del equipo), y por tanto actúa sobre la zona de memoria de las AOs (@4XXXX). Debemos indicar la dirección del valor que queremos modificar y la magnitud que queremos asignarle. Luego el esclavo debería responder con la dirección del dato que ha modificado y el valor que le ha asignado, que debería coincidir con el enviado. Aunque en realidad se está escribiendo en el rango de AOs , los registros son direccionados a partir de la dirección 0 (así el registro @40001 se direcciona 0)

Petición del máster (modo RTU):

NºEsclavo
Código Operación: 0x06
Dirección del dato:
Valor del dato:
CRC(16): H L

Respuesta del esclavo (modo RTU):

NºEsclavo
Código Operación: 0x06
Dirección del dato:
Valor del dato:
CRC(16): H L

Ej:

P:[01][06][01][F1][00][02][58][04]
R:[01][06][01][F1][00][02][58][04]

9.6 Función 7

(Read Exception Status)

Permite la lectura rápida de un byte fijo de un esclavo, que generalmente es el de excepción y que informa del estado del equipo. No tiene dirección del byte debido a que siempre se lee el mismo byte (determinado por el propio dispositivo esclavo) :

Petición del master (modo RTU):

NºEsclavo
Código Operación: 0x07
CRC(16): H L

Respuesta del esclavo (modo RTU):

NºEsclavo
Código Operación: 0x07
Valor del octeto:
CRC(16): H L

9.7 Función 15

(Force Multiple Coils)

Permite la modificación simultanea de varios bits de DOs en el esclavo, pasándolos a OFF ('0') o a ON ('1') según convenga. Actúa sobre la zona de memoria de las DOs (@0XXXX). Así en el comando se pasan la dirección inicial (dirección del primer bit o mando a modificar) y la cantidad y estado de cada uno de los sucesivos mandos (bits) a modificar.

Petición del máster (modo RTU):

NºEsclavo

Código Operación: 0x0F

Dirección inicial de los mandos (bits) H

Dirección inicial de los mandos (bits) L

Cantidad de mandos (bits) H

Cantidad de mandos (bits) L

Cantidad de bytes enviados con el estado de los mandos (bits)

Estado de los 8 primeros mandos (bits) a modificar

Estado de los 8 siguientes mandos (bits) a modificar

...

Estado de los 8 últimos mandos (bits) a modificar

CRC(16): H L

Aunque el estado de las DOs se especifica bit a bit, las tramas se componen de bytes, y esto obliga a enviar los estados en grupos de 8. El esclavo no debería hacer caso a los bits sobrantes, es decir, no debería considerar los que queden por encima del último bit indicado en el campo "cantidad de mandos a modificar". Así, si quisiéramos modificar 12 mandos o relés a partir de la dirección 7, indicaríamos como dirección origen la dirección 7, como cantidad de mandos a modificar 12, y en el campo de estado de mandos: 0x3C , 0x0B (el esclavo no considerará los que queden por encima del 12o bit)

Relé: 14 13 12 11 10 09 08 07 - 22 21 20 19 18 17 16 15

Bits: 0 1 1 1 1 0 0 0 X X X X 1 0 1 1

Respuesta del esclavo (modo RTU):

NºEsclavo

Código Operación: 0x0F

Dirección inicial de los mandos o bits consecutivos H

Dirección del mando o bit L

Cantidad de mandos o bits H

Cantidad de mandos o bits L

CRC(16): H L

9.8 Función 16

(Preset multiple registers)

Permite realizar la escritura en un grupo de AOs, y por tanto actúa sobre la zona de AOs (@4XXXX). Se debe especificar la dirección a partir de la que queremos comenzar a actualizar valores, el número de valores que queremos actualizar, y la lista de valores que queremos asignar a estos registros. Aunque se está escribiendo en el rango de registros o valores numéricos , los registros son direccionados a partir de la dirección 0 (es decir el registro @40001 se direcciona 0)

Petición del máster (modo RTU):

NºEsclavo

Código Operación: 0x10

Dirección base de los datos: 2 bytes
 Número de datos: 2 bytes
 Valor del dato 0: 2 bytes
 Valor del dato 1: 2 bytes
 ...
 Valor del dato n: 2 bytes
 CRC (16): H L

Respuesta del esclavo (modo RTU):

NºEsclavo
 Código Operación: 10H
 Dirección base de los datos: 2 bytes
 Número de datos: 2 bytes
 CRC (16): H L

9.9 Función 20

(Read General Reference):

Retorna el contenido de los registros de la Extended Memory (@6XXXX). En la trama se envía la dirección del esclavo, el código de función, el número de bytes que se va a leer y los bytes de comprobación de error. En los datos se establece el grupo o grupos o referencias a leer. Cada grupo se define en un campo "sub-request" que contiene:

- Reference type: 1 byte (debe ser 6)
- Extended Memory file number: 2 bytes (0x01 a 0x0A)
- La dirección del "fichero" a partir de la que queremos comenzar a leer: 2 bytes
- La cantidad de registros a leer: 2 bytes

Los datos a leer seguidos de los demás campos no debe superar la longitud máxima permitida en los mensajes MODBUS, es decir los 256 bytes.

Petición del máster (modo RTU):

Slave Address 0x11
 Function 0x14
 Byte Count 0x0E
 Sub-Req 1, Reference Type 0x06
 Sub-Req 1, File Number Hi 0x00
 Sub-Req 1, File Number Lo 0x04
 Sub-Req 1, Starting Addr Hi 0x00
 Sub-Req 1, Starting Addr Lo 0x01
 Sub-Req 1, Register Count Hi 0x00
 Sub-Req 1, Register Count Lo 0x02
 Sub-Req 2, Reference Type 0x06
 Sub-Req 2, File Number Hi 0x00
 Sub-Req 2, File Number Lo 0x03
 Sub-Req 2, Starting Addr Hi 0x00
 Sub-Req 2, Starting Addr Lo 0x09
 Sub-Req 2, Register Count Hi 0x00
 Sub-Req 2, Register Count Lo 0x02
 Error Check (LRC or CRC)--

Respuesta del esclavo (modo RTU):

Slave Address 0x11
 Function 0x14
 Byte Count 0x0C

Sub-Res 1, Byte Count 0x05
 Sub-Res 1, Reference Type 0x06
 Sub-Res 1, Register Data Hi 0x0D
 Sub-Res 1, Register Data Lo 0xFE
 Sub-Res 1, Register Data Hi 0x00
 Sub-Res 1, Register Data Lo 0x20
 Sub-Res 2, Byte Count 0x05
 Sub-Res 2, Reference Type 0x06
 Sub-Res 2, Register Data Hi 0x33
 Sub-Res 2, Register Data Lo 0xCD
 Sub-Res 2, Register Data Hi 0x00
 Sub-Res 2, Register Data Lo 0x40
 Error Check (LRC or CRC) —

9.10 Función 21

(Write General Reference):

Modifica el contenido de los registros de la Extended Memory (6XXXX). Esta función permite trabajar en grupos de registros que no tienen por qué ser contiguos (internamente sí que han de ser contiguos).

Cada grupo se define en un campo "sub-request" que contiene:

- Reference type: 1 byte (debe ser 6)
- Extended Memory file number: 2 bytes (0x01 a 0x0A)
- La dirección del "fichero" a partir de la que queremos comenzar a leer: 2 bytes
- La cantidad de registros a leer: 2 bytes

Los datos a leer seguidos de los demás campos no debe superar la longitud máxima permitida en los mensajes MODBUS, es decir los 256 bytes.

Petición del máster (modo RTU):

Slave Address 0x11
 Function 0x5
 Byte Count 0x0D
 Sub-Req 1, Reference Type 0x06
 Sub-Req 1, File Number Hi 0x00
 Sub-Req 1, File Number Lo 0x04
 Sub-Req 1, Starting Addr Hi 0x00
 Sub-Req 1, Starting Addr Lo 0x07
 Sub-Req 1, Register Count Hi 0x00
 Sub-Req 1, Register Count Lo 0x03
 Sub-Req 1, Register Data Hi 0x06
 Sub-Req 1, Register Data Lo 0xAF
 Sub-Req 1, Register Data Hi 0x04
 Sub-Req 1, Register Data Lo 0xBE
 Sub-Req 1, Register Data Hi 0x10
 Sub-Req 1, Register Data Lo 0x0D
 Error Check (LRC or CRC) —

Respuesta del esclavo (modo RTU):

Slave Address 0x11
 Function 0x15
 Byte Count 0x0D
 Sub-Req 1, Reference Type 0x06
 Sub-Req 1, File Number Hi 0x00
 Sub-Req 1, File Number Lo 0x04

Sub-Req 1, Starting Addr Hi 0x00
Sub-Req 1, Starting Addr Lo 0x07
Sub-Req 1, Register Count Hi 0x00
Sub-Req 1, Register Count Lo 0x03
Sub-Req 1, Register Data Hi 0x06
Sub-Req 1, Register Data Lo 0xAF
Sub-Req 1, Register Data Hi 0x04
Sub-Req 1, Register Data Lo 0xBE
Sub-Req 1, Register Data Hi 0x10
Sub-Req 1, Register Data Lo 0x0D
Error Check (LRC or CRC) —

10 ANEXO II: NORMA IEEE 802

IEEE 802 es un proyecto del *Institute of Electrical and Electronics Engineers* (más conocido por sus siglas, IEEE). Se identifica también con las siglas LMSC (LAN/MAN Standards Committee). Su misión se centra en desarrollar estándares de redes de área local (LAN) y redes de área metropolitana (MAN), principalmente en las dos capas inferiores del modelo OSI.

IEEE que actúa sobre Redes de computadoras. Concretamente y según su propia definición sobre redes de área local (RAL, en inglés LAN) y redes de área metropolitana (MAN en inglés). También se usa el nombre IEEE 802 para referirse a los estándares que proponen, algunos de los cuales son muy conocidos: Ethernet (IEEE 802.3), o Wi-Fi (IEEE 802.11). Está, incluso, intentando estandarizar Bluetooth en el 802.15 (IEEE 802.15).

Se centra en definir los niveles más bajos (según el modelo de referencia OSI o sobre cualquier otro modelo). Concretamente subdivide el segundo nivel, el de enlace, en dos subniveles: el de Enlace Lógico (LLC), recogido en 802.2, y el de Control de Acceso al Medio (MAC), subcapa de la capa de Enlace Lógico. El resto de los estándares actúan tanto en el Nivel Físico, como en el subnivel de Control de Acceso al Medio.



Figura 10-1 Capas del Modelo OSI.

Los diferentes estándares se numeran en la Tabla 10-1.

Nombre	Descripción	Nota
IEEE 802.1	Normalización de interfaz	
802.1d	<i>Spanning Tree Protocol</i>	
802.1p	Asignación de Prioridades de tráfico	
802.1q	<i>Virtual Local Area Networks (VLAN)</i>	
802.1x	Autenticación en redes LAN	
802.1aq	<i>Shortest Path Bridging (SPB)</i>	
IEEE 802.2	Control de enlace lógico LLC	Activo
IEEE 802.3	CSMA / CD (ETHERNET)	
IEEE 802.3a	Ethernet delgada 10Base2	
IEEE 802.3c	Especificaciones de Repetidor en Ethernet a 10 Mbps	
IEEE 802.3i	Ethernet de par trenzado 10BaseT	
IEEE 802.3j	Ethernet de fibra óptica 10BaseF	
IEEE 802.3u	Fast Ethernet 100BaseT	
IEEE 802.3z	Gigabit Ethernet parámetros para 1000 Mbps	
IEEE 802.3ab	Gigabit Ethernet sobre 4 pares de cable UTP Cat5e o superior	
IEEE 802.3ae	10 Gigabit Ethernet	
IEEE 802.4	Token bus LAN	Disuelto
IEEE 802.5	Token ring LAN (topología en anillo)	Inactivo
IEEE 802.6	Redes de Área Metropolitana (MAN) (ciudad) (fibra óptica)	Disuelto
IEEE 802.7	Grupo Asesor en Banda ancha	Disuelto
IEEE 802.8	Grupo Asesor en Fibras Ópticas	Disuelto
IEEE 802.9	Servicios Integrados de red de Área Local (redes con voz y datos integrados)	Disuelto
IEEE 802.10	Seguridad de red	Disuelto
IEEE 802.11	Redes inalámbricas WLAN. (Wi-Fi)	
IEEE 802.12	Acceso de Prioridad por demanda 100 Base VG-Any Lan	Disuelto
IEEE 802.13	Se ha evitado su uso por superstición ²	Sin uso
IEEE 802.14	Módems de cable	Disuelto
IEEE 802.15	WPAN (Bluetooth)	
IEEE 802.16	Redes de acceso metropolitanas sin hilos de banda ancha (WIMAX)	
IEEE 802.17	Anillo de paquete elástico script	
IEEE 802.18	Grupo de Asesoría Técnica sobre Normativas de Radio	En desarrollo a día de hoy
IEEE 802.19	Grupo de Asesoría Técnica sobre Coexistencia	
IEEE 802.20	<i>Mobile Broadband Wireless Access</i>	
IEEE 802.21	Media Independent Handoff	
IEEE 802.22	<i>Wireless Regional Area Network</i>	

Tabla 10-1 Relación de normas IEEE.

10.1 802.1D

802.1D es el estándar de IEEE para bridges MAC (puentes MAC), que incluye bridging (técnica de reenvío de paquetes que usan los switches), el protocolo Spanning Tree y el funcionamiento de redes 802.11, entre otros.

También impide que los bucles que se forman cuando los puentes o los interruptores están interconectados a través de varias rutas. El algoritmo BPDU logra mediante el intercambio de mensajes con otros switches para detectar bucles y, a continuación, elimina el bucle por el cierre de puente seleccionado interfaces. Este algoritmo garantiza que hay una y sólo una ruta activa entre dos dispositivos de red.

Las VLANs (redes virtuales) no son parte de 802.1D, sino de IEEE 802.1Q.

- Spanning Tree: en comunicaciones, STP (del inglés Spanning Tree Protocol) es un protocolo de red de nivel 2 del modelo OSI (capa de enlace de datos). Su función es la de gestionar la presencia de bucles en topologías de red debido a la existencia de enlaces redundantes (necesarios en muchos casos para garantizar la disponibilidad de las conexiones). El protocolo permite a los dispositivos de interconexión activar o desactivar automáticamente los enlaces de conexión, de forma que se garantice la eliminación de bucles. STP es transparente a las estaciones de usuario.

10.2 802.2

IEEE 802.2 es el IEEE 802 estándar que define el control de enlace lógico (LLC), que es la parte superior de la capa enlace en las redes de área local. La subcapa LLC presenta una interfaz uniforme al usuario del servicio enlace de datos, normalmente la capa de red. Bajo la subcapa LLC está la subcapa Media Access Control (MAC), que depende de la configuración de red usada (Ethernet, token ring, FDDI, 802.11, etc.).

10.3 802.3

IEEE 802.3 fue el primer intento para estandarizar redes basadas en ethernet, incluyendo las especificaciones del medio físico subyacente. Aunque hubo un campo de la cabecera que se definió de forma diferente, posteriormente ha habido ampliaciones sucesivas al estándar que cubrieron las ampliaciones de velocidad (Fast Ethernet, Gigabit Ethernet y el de 10 Gigabit Ethernet), redes virtuales, hubs, conmutadores y distintos tipos de medios, tanto de fibra óptica como de cables de cobre (tanto par trenzado como coaxial).

En la práctica, el CSMA/CD requiere que cada host que desea utilizar el cable primero lo escuche para determinar si está limpio. Cuando está limpio, el host puede transmitir. Debido a que existe la posibilidad de que otra estación haya realizado una transmisión simultáneamente, cada estación que transmite escucha el cable a medida que envía la primera parte de su mensaje. Si no escucha ninguna otra señal, continúa hasta que el mensaje finaliza, y luego comienza el proceso nuevamente para el mensaje siguiente. SI la estación escucha otra señal mientras todavía está transmitiendo, detiene la transmisión. Después, la estación envía una señal de atascamiento. Todas las estaciones que escuchan la señal de atascamiento borran el paquete recibido parcialmente y, esperan un período aleatorio antes de volver a comenzar la transmisión.

Este método de comunicación se denomina no determinista, es decir, no se puede predecir cuál estación transmitirá y cuándo transmitirá. No obstante, cada estación en algún punto en el tiempo tendrá la oportunidad de transmitir. La ventaja de este sistema es que se ejecuta a sí mismo sin requerir ninguna administración.

IEEE 802.3 define los siguientes estándares de cableado para las LAN que operan a una velocidad de señalización de banda base de 10 o 100 Mbps, denominada 10Base o 100Base:

- 10Base2 – Utiliza cableado thinnet con una longitud de segmento máxima de 185 m, y se utiliza con topología de bus física y topología de bus lógica.
- 10Base5 – Utiliza cableado thinnet con una longitud de segmento máxima de 500 m, y se utiliza con topología de bus física y topología de bus lógica.

- 10Base-T – Utiliza cableado UTP Categoría 3, 5, 5e o 6 con una longitud de segmento máxima de 100 m y se utiliza con topología en estrella física o extendida y topología de bus lógica.
- 10Base-FL – Utiliza cableado de fibra óptica multimodo que opera a 850 nm. La distancia máxima desde una NIC a un hub es de 2000 metros.
- 100Base-TX – Utiliza cableado UTP Categoría 5, 5e o 6 con una longitud de segmento máxima de 100 m, y se utiliza con topología en estrella física o extendida y topología de bus lógica.
- 100Base-FX – Utiliza cableado de fibra óptica multimodo que opera a 1300 nm con una longitud de segmento máxima no especificada, que depende del uso de un hub nox Clase I o Clase II.
- 1000Base-T – Utiliza cuatro pares trenzados de cobre, Categoría 5 o superior. (IEEE 802.3ab)
- 1000Base-TX – Utiliza cuatro pares trenzados de cobre, Categoría 6.
- 1000Base-CX – Ensamble de cable blindado de cuatro conductores para fines especiales (IEEE 802.3z)
- 1000Base-SX – Dos fibras ópticas que operan a 850 nm. (IEEE 802.3z)
- 1000Base-LX – Dos fibras ópticas que operan a 1300 nm. (IEEE 802.3z)
- Ethernet de diez gigabits (10GbE) (IEEE 802.3ae)

11 ANEXO III: RFC 826

En red de computadoras, el protocolo de resolución de direcciones (ARP, del inglés *Address Resolution Protocol*) es un protocolo de comunicaciones de la capa de enlace, responsable de encontrar la dirección de hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (*ARP request*) a la dirección de difusión de la red (*broadcast*, MAC = FF FF FF FF FF FF) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (*ARP reply*) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet ser independiente de la dirección Ethernet, pero esto solo funciona si todas las máquinas lo soportan.

ARP está documentado en el RFC 826¹¹. El protocolo RARP realiza la operación inversa y se encuentra descrito en el RFC 903.

En Ethernet, la capa de enlace trabaja con direcciones físicas. El protocolo ARP se encarga de traducir las direcciones IP a direcciones MAC (direcciones físicas). Para realizar esta conversión, el nivel de enlace utiliza las tablas ARP, cada interfaz tiene tanto una dirección IP como una dirección física MAC.

ARP se utiliza en cuatro casos referentes a la comunicación entre dos *hosts*:

- Cuando dos hosts están en la misma red y uno quiere enviar un paquete a otro.
- Cuando dos hosts están sobre redes diferentes y deben usar un gateway o router para alcanzar otro host.
- Cuando un router necesita enviar un paquete a un host a través de otro router.
- Cuando un router necesita enviar un paquete a un host de la misma red.

La Figura 11-1 muestra un ejemplo de un mensaje ARP Request y Figura 11-2 muestra la respuesta.

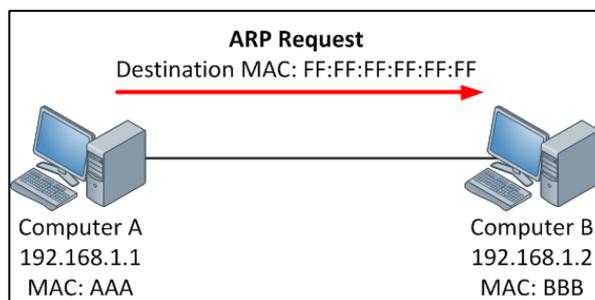


Figura 11-1 ARP Request.

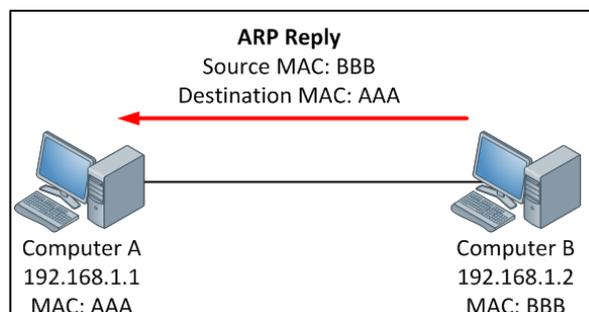


Figura 11-2 ARP Reply.

¹¹ <http://www.networksorcery.com/enp/rfc/rfc826.txt>

