



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

**DEPARTAMENTO DE ELECTRÓNICA Y
COMPUTACIÓN**

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
& AVIÓNICA**

**TRABAJO DE GRADUACIÓN PARA LA OBTENCIÓN DEL
TÍTULO DE:**

**TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**TEMA: “CONFIGURACIÓN Y PROGRAMACIÓN DE UNA
COMUNICACIÓN MODBUS RTU ENTRE PLCs S7 1200
EMPLEANDO LOS MÓDULOS CM 1241-RS485 PARA
PRÁCTICAS DE REDES INDUSTRIALES”.**

AUTOR: MOSQUERA YASIG, WALTER GEOVANNI

**DIRECTORA: Ing. María Mogro
LATACUNGA, 2015**

CERTIFICADO

Certifico que el presente Trabajo de Graduación fue realizado en su totalidad por el Sr. **MOSQUERA YASIG WALTER GEOVANNI**, como requerimiento parcial para la obtención del título de **TECNÓLOGO EN ELECTRÓNICA MENCION INSTRUMENTACIÓN Y AVIONICA**.

SRA. ING. MARÍA MOGRO
DIRECTORA DEL TRABAJO DE GRADUACIÓN

Latacunga, Agosto 2015.

AUTORÍA DE RESPONSABILIDAD

Yo, Mosquera Yasig Walter Geovanni

DECLARO QUE:

El trabajo de grado denominado “**CONFIGURACIÓN Y PROGRAMACIÓN DE UNA COMUNICACIÓN MODBUS RTU ENTRE PLCs S7 1200 EMPLEANDO LOS MÓDULOS CM 1241-RS485 PARA PRÁCTICAS DE REDES INDUSTRIALES**”, ha sido desarrollado en base a una investigación científica, respetando derechos intelectuales de terceros conforme las citas constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente, este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de trabajo de grado en mención.

Latacunga, Agosto 2015.

Mosquera Yasig Walter Geovanni.

C.I: 0503222168

DEDICATORIA

Dedico esta tesis a mi hija Fernanda, quien fue un gran apoyo emocional durante el tiempo en que escribía esta tesis.

A mis padres y mi familia quienes me apoyaron todo el tiempo.

A mi esposa Andrea quien me apoyo y alentó para continuar, cuando parecía que me iba a rendir.

A mis maestros quienes nunca desistieron al enseñarme, aun sin importar que muchas veces no ponía atención en clase, a ellos que continuaron depositando su esperanza en mí.

A los sinodales quienes estudiaron mi tema y la aprobaron.

A todos los que me apoyaron para escribir y concluir esta tesis.

Para ellos es esta dedicatoria de tesis, pues es a ellos a quienes se las debo por su apoyo incondicional

AGRADECIMIENTO

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Le doy gracias a mis padres José y Clara por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo por ser un excelente ejemplo de vida a seguir.

A mis hermanos por ser parte importante de mi vida y representar la unidad familiar.

A mi hija Fernanda y mi esposa Andrea por ser parte muy importante de mi vida, por haberme apoyado en las buenas y en las malas, sobre todo por su paciencia y amor incondicional.

Gracias Ingeniero Pablo Pilatasig por creer en mí, y haberme brindado la oportunidad de desarrollar mi tema por todo el apoyo y facilidades que me fue otorgado en el laboratorio. Por darme la oportunidad de crecer profesionalmente y aprender cosas nuevas.

ÍNDICE DE CONTENIDOS

CERTIFICADO.....	ii
AUTORÍA DE RESPONSABILIDAD	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS.....	x
RESUMEN	xii
ABSTRACT.....	xiii
CAPÍTULO I	
PLANTEAMIENTO DEL PROBLEMA.....	1
1.1 ANTECEDENTES.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	1
1.3 JUSTIFICACIÓN.....	2
1.4 OBJETIVOS.....	2
1.4.1 Objetivo general.....	2
1.4.2 Objetivos específicos.....	2
1.5 Alcance.....	3
CAPÍTULO II	
MARCO TEÓRICO.....	4
2.1 Comunicación Serial.....	4
2.1.1 RS 232	6
2.1.2 RS 422	7
2.1.3 RS 485	7

2.2	Modbus	8
2.2.1	Características	9
2.2.2	Estructura Lógica	10
2.2.3	Aplicación.....	11
2.2.4	Protocolos	12
2.2.5	Modbus RTU	13
2.2.6	Trama RTU	14
2.2.7	Mapa de direcciones Modbus	15
	Fuente: (TIA Portal, 2013)	16
2.3	Módulo de Comunicación CM 1241 RS422/RS485	16
2.4	PLC S7 1200	17
2.4.1	Partes del PLC S7 1200.....	18
2.4.2	Tarjeta de señales.....	19
2.4.3	Módulo de señales	19
2.4.4	Módulo de comunicación.....	20
2.4.5	Almacenamiento de datos, áreas de memoria y direccionamiento .	20
2.4.6	Acceso a los datos en las áreas de memoria de la CPU.....	21
2.5	TIA PORTAL.....	23
2.5.1	Apariencia homogénea	24
2.5.2	Inteligencia integrada	24
2.5.3	Máxima transparencia de datos	25
2.5.4	Soluciones reutilizables.....	25
2.6	Instrucciones Modbus en PLC S7 1200.....	25
2.6.1	MB_COMM_LOAD.....	25
2.6.2	MB_MASTER.....	27
2.6.3	MB_SLAVE	29

CAPÍTULO III

COMUNICACIÓN MODBUS RTU ENTRE DOS PLCs S7 1200.....	32
3.1 Preliminares.....	32
3.2 Conexiones físicas de los dispositivos.....	32
3.3 Configuración de la comunicación en TIA Portal V12.....	33
3.4 Programación del maestro Modbus.....	37
3.4.1 Creación del bloque de datos del maestro.....	37
3.4.2 Configuración del puerto del maestro.....	39
3.4.3 Programación del maestro.....	40
3.5 Programación del esclavo Modbus.....	47
3.5.1 Creación del bloque de datos del esclavo.....	47
3.5.2 Configuración del puerto del esclavo.....	48
3.5.3 Programación del esclavo.....	49
3.6 Pruebas de funcionamiento.....	51
3.7 Comunicación maestro/esclavo práctica 2.....	52
3.8 Comunicación maestro/esclavo práctica 2.....	53

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES.....	58
4.1 Conclusiones.....	58
4.2 Recomendaciones.....	58
GLOSARIO DE TÉRMINOS.....	59
REFERENCIA BIBLIOGRAFÍA.....	60
ANEXOS.....	61

ÍNDICE DE TABLAS

Tabla 1. Trama del mensaje RTU.....	15
Tabla 2. Estructura de los datos en Modbus	15
Tabla 3. Memoria del proceso de las entradas.....	22
Tabla 4. Memoria del proceso de salidas	22
Tabla 5. Área de marcas	23
Tabla 6. Parámetros del bloque MB_MASTER	29
Tabla 7. Parámetros del Bloque MB_SLAVE	30

ÍNDICE DE FIGURAS

Figura 1. Conector DB9 para RS 232	7
Figura 2. Conector DB9 para RS 422 y RS 485.....	8
Figura 3. Estructura de Red Modbus	10
Figura 4. Versiones Modbus	13
Figura 5. Módulo CM 1241 RS485.....	17
Figura 6 Partes del PLC S7 1200.....	18
Figura 7. Signal Board	19
Figura 8. Partes de un módulo de señal	19
Figura 9. Partes de un módulo de comunicación	20
Figura 10. Bloque MB_COMM_LOAD	27
Figura 11. Bloque MB_MASTER.....	29
Figura 12. Bloque MB_SLAVE	30
Figura 13. Conexión física de la comunicación Modbus	32
Figura 14. Agregar PLC al proyecto.....	33
Figura 15. Opción Agregar dispositivo	34
Figura 16. Asignación de nombres a los PLCs	34
Figura 17. Conexión Profinet entre PLCs.....	35
Figura 18. Asignación de IP al PLC esclavo	35
Figura 19. Asignación de IP al PLC maestro	36
Figura 20. Activación bytes de marca de sistema.....	36
Figura 21. Selección del CM 1241	37
Figura 22. CM 1241 conectado al PLC	37
Figura 23. Agregar bloque de datos maestro	38
Figura 24. Deshabilitar Acceso optimizado al bloque.....	38
Figura 25. Bloque de datos del maestro	38
Figura 26. Protocolo de comunicación Modbus maestro	39
Figura 27. Envío de datos desde el maestro.....	40
Figura 28. Incremento de dato enviado.....	41
Figura 29. Petición de envío al esclavo.....	41

Figura 30. Transición al paso 2 del esclavo	42
Figura 31. Lectura de datos del esclavo	42
Figura 32. Maestro solicita respuesta del esclavo.....	43
Figura 33. Cambio de paso del maestro	43
Figura 34. Creación bloque datos esclavo	47
Figura 35. Deshabilitar Acceso optimizado al bloque esclavo	47
Figura 36. Bloque datos esclavo	48
Figura 37. Protocolo de comunicación Modbus esclavo	48
Figura 38. Intercambio de datos maestro esclavo.....	49
Figura 39. Intercambio de datos del contador	50
Figura 40. Pruebas funcionales comunicación Modbus	52
Figura 41. Envío de una constante al esclavo	52
Figura 42. Bloque datos práctica 1.....	53
Figura 43. Resultados práctica 1.....	53
Figura 44. Segmento 3 del maestro práctica 2	54
Figura 45. Segmento 3 del esclavo práctica 2	54
Figura 46. Segmento 9 del maestro práctica 2	55
Figura 47. Conexión física maestro práctica 2	55
Figura 48. Segmento 4 del esclavo práctica 2	56
Figura 49. Segmento 5 del esclavo práctica 2	56
Figura 50. Prueba funcional práctica 2.....	57

RESUMEN

El presente Trabajo de Graduación consiste en configurar una comunicación Modbus RTU entre dos PLC S7 1200 CPU 1214C AC/DC/RLY, con la ayuda de los módulos de comunicación CM 1241 RS422/RS485. La comunicación que se implementó es half dúplex; es decir, que los esclavos responden solo cuando el maestro solicita. Para este intercambio de datos se creó bloques de datos globales tanto en el maestro como en el esclavo, cada bloque de datos consiste de un array con 2 elementos de tipo entero para el envío y de un array con 2 elementos de tipo entero para la recepción. En la primera práctica el programa del maestro realiza la comunicación en dos pasos, en el primer paso el maestro envía mediante el elemento 2 del array correspondiente al esclavo el valor de un contador, luego si no existe errores, el maestro recibe el dato del contador dividido para dos a través del elemento 2 del array de recepción. En la segunda práctica con la misma estructura de programación del maestro, se enciende la salida digital cero del esclavo, activando la entrada digital cero del maestro. Finalmente en el maestro se conecta un módulo de señales de salidas analógicas, para que cuando se active la entrada digital cero, el maestro envíe al esclavo un valor de 27648, el esclavo envía al maestro el valor de 13824, que corresponde al valor enviado por el maestro dividido para dos. El valor recibido por el maestro se envía al módulo de salidas analógicas para generar un voltaje de 5 voltios.

PALABRAS CLAVES

MODBUS RTU

CM 1241-RS485

TIA PORTAL

PLC S7 1200

ABSTRACT

This work is to set up a Graduation Modbus RTU communication between two S7 1200 PLC CPU 1214C AC / DC / RLY, with the help of the communication modules CM 1241 RS422 / RS485. The communication is half duplex implemented; in the slaves respond only when the master requests. For data exchange global data blocks are created in both the master and slave, each data block consists of an array with 2 elements of type integer for sending and an array with 2 elements of type integer for reception. In the first practice of the master program performs the communication in two steps, the first step the teacher sent by the array element 2 corresponding to the slave the value of a counter, then if there is no error, the master receives the data counter two divided element 2 through the receiving array. In the second practice with the same structure master programming, the digital output of the slave zero turn on, activating the digital input zero teacher. Finally in the master module of analog outputs connected, so that when the zero digital input is activated, a teacher I sent to the slave a value of 27648, the slave sends to the master the value of 13824, which corresponds to the value sent by the teacher split for two. The value received by the master is sent to the analog output module to generate a voltage of 5 volts.

KEYWORDS

MODBUS RTU

CM 1241 -RS485

TIA PORTAL

S7 1200 PLC

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA.

CONFIGURACIÓN Y PROGRAMACIÓN DE UNA COMUNICACIÓN MODBUS RTU ENTRE PLCs S7 1200 EMPLEANDO LOS MÓDULOS CM 1241-RS485 PARA PRÁCTICAS DE REDES INDUSTRIALES

ANTECEDENTES.

La Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas-ESPE es un centro académico de formación tecnológica superior situado en la provincia de Cotopaxi, cantón Latacunga; en la calle Xavier Espinoza y Av. Amazonas.

En el laboratorio de Máquinas Eléctricas y Control Industrial existen dispositivos de comunicación Profibus y Ethernet por tal razón se va a implementar el protocolo **Modbus RTU y los módulos CM 1241-RS485**.

De esta manera se propone que el laboratorio cuente con dicho modulo ayudara fundamentalmente para el desarrollo de habilidades, destrezas en los estudiantes mediante esto adquiriran el aprendizaje significativo.

PLANTEAMIENTO DEL PROBLEMA.

El laboratorio de Máquinas Eléctricas y Control Industrial de la Unidad de Gestión de Tecnologías de la Universidad de Fuerzas Armadas – ESPE, cuenta con dispositivos de comunicación Profibus y Ethernet, sin embargo se ha evidenciado que es necesario la implementación de módulos de comunicación **Modbus RTU y los módulos CM 1241-RS485** para completar la pirámide de comunicación industrial.

Por tal razón, los estudiantes podrán realizar prácticas que complementen su conocimiento teórico, lo cual puede favorecer a futuro en un mejor desempeño profesional.

JUSTIFICACIÓN

Actualmente el avance tecnológico va desarrollando nuevos y mejores equipos e instrumentos en el campo de la comunicación industrial, debido a esto es necesario que la Unidad de Gestión de Tecnologías de la Universidad de Fuerzas Armadas – ESPE realice una optimización de sus laboratorios, ya que de este modo los estudiantes podrán adquirir mayores conocimientos prácticos, para complementar el conocimiento teórico impartido en las aulas, aspectos de vital importancia para brindar a futuro una educación de calidad.

Por todo lo anteriormente descrito, se evidencia la optimización de los laboratorios, ya que de esta manera se podrá mejorar el aprendizaje teórico-práctico de los estudiantes. Tomando en cuenta la disponibilidad económica, técnica, operacional u organizacional de los recursos necesarios para llevar a cabo los objetivos y metas señaladas.

Mediante **los módulos CM 1241-RS485** se realizará una comunicación Modbus RTU por medio de un maestro y un esclavo entre PLCs S7 1200 para prácticas de redes industriales.

OBJETIVOS.

1.1.1 Objetivo general.

Configurar y programar una comunicación Modbus RTU entre PLCs S7 1200 empleando los módulos de comunicación CM 1241-RS485 para prácticas de redes industriales

1.1.2 Objetivos específicos.

- Indagar las características de los módulos de comunicación Modbus RTU.
- Configurar los PLCs S7 1200 para que se comuniquen mediante los módulos **CM1241-RS485**.
- Realizar la programación de los PLCs S7 1200 mediante el Tia Portal.

Alcance.

Este proyecto está dirigido a la Carrera de Electrónica Mención Instrumentación y Aviónica de la Universidad de Fuerzas Armadas-ESPE, para la asignatura de Automatización y Control de Procesos, brindando a los estudiantes los dispositivos para que puedan desarrollar prácticas del protocolo Modbus, lo que permitirá al estudiante obtener mayor experiencia en el campo práctico para posteriormente desempeñarse de mejor manera en el ámbito laboral, logrando contar con profesionales altamente capacitados y competitivos, capaces de contribuir con el desarrollo de nuestro país.

CAPÍTULO II

MARCO TEÓRICO.

Comunicación Serial

La comunicación serial es un protocolo muy común (no hay que confundirlo con el Bus Serial de Comunicación, o USB) para comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen dos puertos seriales RS-232. La comunicación serial es también un protocolo común utilizado por varios dispositivos para instrumentación; existen varios dispositivos compatibles con GPIB que incluyen un puerto RS-232. Además, la comunicación serial puede ser utilizada para adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo (National Instruments, 2012).

El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias. Por ejemplo, la especificación IEEE 488 para la comunicación en paralelo determina que el largo del cable para el equipo no puede ser mayor a 20 metros, con no más de 2 metros entre cualesquier dos dispositivos; por el otro lado, utilizando comunicación serial el largo del cable puede llegar a los 1200 metros.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra. Existen otras líneas disponibles para realizar handshaking, o intercambio de pulsos de sincronización, pero no son requeridas. Las características más importantes de la comunicación serial son

la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

Velocidad de transmisión (baud rate): Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds). Por ejemplo, 300 baudios representan 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión. Por ejemplo, si el protocolo hace una llamada a 4800 ciclos de reloj, entonces el reloj está corriendo a 4800 Hz, lo que significa que el puerto serial está muestreando las líneas de transmisión a 4800 Hz. Las velocidades de transmisión más comunes para las líneas telefónicas son de 14400, 28800, y 33600. Es posible tener velocidades más altas, pero se reduciría la distancia máxima posible entre los dispositivos. Las altas velocidades se utilizan cuando los dispositivos se encuentran uno junto al otro, como es el caso de dispositivos GPIB.

Bits de datos: Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Por ejemplo, el ASCII estándar tiene un rango de 0 a 127, es decir, utiliza 7 bits; para ASCII extendido es de 0 a 255, lo que utiliza 8 bits. Si el tipo de datos que se está transfiriendo es texto simple (ASCII estándar), entonces es suficiente con utilizar 7 bits por paquete para la comunicación. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usa para referirse a todos los casos.

Bits de parada: Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la

transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo la transmisión será más lenta.

Paridad: Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Para paridad par e impar, el puerto serial fijará el bit de paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado alto lógico. Por ejemplo, si la información a transmitir es 011 y la paridad es par, el bit de paridad sería 0 para mantener el número de bits en estado alto lógico como par. Si la paridad seleccionada fuera impar, entonces el bit de paridad sería 1, para tener 3 bits en estado alto lógico. La paridad marcada y espaciada en realidad no verifican el estado de los bits de datos; simplemente fija el bit de paridad en estado lógico alto para la marcada, y en estado lógico bajo para la espaciada. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados.

2.1.1 RS 232

RS-232 (Estándar ANSI/EIA-232) es el conector serial hallado en las PCs IBM y compatibles. Es utilizado para una gran variedad de propósitos, como conectar un ratón, impresora o modem, así como instrumentación industrial. Gracias a las mejoras que se han ido desarrollando en las líneas de transmisión y en los cables, existen aplicaciones en las que se aumenta el desempeño de RS-232 en lo que respecta a la distancia y velocidad del estándar. RS-232 está limitado a comunicaciones de punto a punto entre los dispositivos y el puerto serial de la computadora. El hardware de RS-232 se puede utilizar para comunicaciones seriales en distancias de hasta 50 pies.

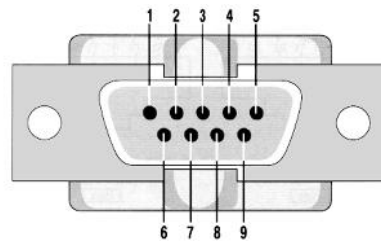


Figura 1. Conector DB9 para RS 232

Fuente: (National Instruments, 2012)

Funciones de los pines en RS-232:

Datos: TXD (pin 3), RXD (pin 2)

Handshake: RTS (pin 7), CTS (pin 8), DSR (pin 6), DCD (pin 1), DTR (pin 4)

Tierra: GND (pin 5)

Otros: RI (pin 9)

2.1.2 RS 422

RS-422 (Estándar EIA RS-422-A) es el conector serial utilizado en las computadoras Apple de Macintosh. RS-422 usa señales eléctricas diferenciales, en comparación con señales referenciadas a tierra como en RS-232. La transmisión diferencial, que utiliza dos líneas para transmitir y recibir, tiene la ventaja que es más inmune al ruido y puede lograr mayores distancias que RS-232. La inmunidad al ruido y la distancia son dos puntos clave para ambientes y aplicaciones industriales.

2.1.3 RS 485

RS-485 (Estándar EIA-485) es una mejora sobre RS-422 ya que incrementa el número de dispositivos que se pueden conectar (de 10 a 32) 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 serial sea la elección de muchas aplicaciones industriales que necesitan

dispositivos distribuidos en red conectados a una PC u otro controlador para la colección de datos, HMI, u otras operaciones. RS-485 es un conjunto que cubre RS-422, por lo que todos los dispositivos que se comunican usando RS-422 pueden ser controlados por RS-485. El hardware de RS-485 se puede utilizar en comunicaciones seriales de distancias de hasta 4000 pies de cable.

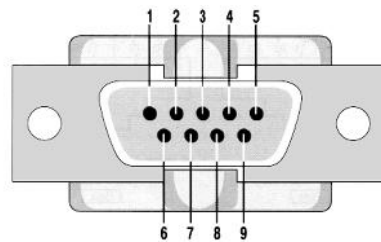


Figura 2. Conector DB9 para RS 422 y RS 485

Fuente: (National Instruments, 2012)

Funciones de los pines en RS-485 y RS-422:

Datos: TXD+ (pin 8), TXD- (pin 9), RXD+ (pin 4), RXD- (pin 5)

Handshake: RTS+ (pin 3), RTS- (pin 7), CTS+ (pin 2), CTS- (pin 6)

Tierra: GND (pin 1)

Modbus

Modbus es un protocolo de comunicaciones serie situado en el nivel 2 del Modelo OSI, desarrollado y publicado por Modicon para su gama de controladores lógicos programables (PLCs) en 1979. La designación Modbus Modicon corresponde a una marca registrada por Gould Inc. Como en tantos otros casos, la designación no corresponde propiamente al estándar de red, incluyendo todos los aspectos desde el nivel físico hasta el de aplicación, sino a un protocolo de enlace. Puede, por tanto, implementarse con diversos tipos de conexión física y cada fabricante suele suministrar un software de aplicación propio, que permite parametrizar sus productos. (Barragan , 2013)

Su objeto es bien sencillo: La transmisión de información entre distintos equipos electrónicos conectados a un mismo bus. Existiendo en dicho bus un solo dispositivo maestro (Master) y varios equipos esclavos (Slaves) conectados(cliente/servidor), No hace falta más que echar un vistazo al

mercado industrial actual para darse cuenta que, 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 Grids, redes de sensores, telecontrol y un largo etc de sistemas de información que ya empiezan a asomar la cabeza en nuestro día a día.

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 la Modbus TCP, que permite el encapsulamiento del 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.

2.1.4 Características

2.1.4.1 Estructura de Red

Medio Físico

El medio físico de conexión puede ser un bus semidúplex (half duplex) (RS-485 o fibra óptica) o dúplex (full duplex) (RS-422, BC 0-20mA o fibra óptica).

La comunicación es asíncrona y las velocidades de transmisión previstas van desde los 75 baudios a 19.200 baudios. La máxima distancia entre estaciones depende del nivel físico, pudiendo alcanzar hasta 1200 m sin repetidores.

Acceso al Medio

La estructura lógica es del tipo maestro-esclavo, con acceso al medio controlado por el maestro. El número máximo de estaciones previsto es de 63 esclavos más una estación maestra.

Los intercambios de mensajes pueden ser de dos tipos:

- **Intercambios punto a punto**, que comportan siempre dos mensajes: una demanda del maestro y una respuesta del esclavo.
- **Mensajes difundidos**. Estos consisten en una comunicación unidireccional del maestro a todos los esclavos. Este tipo de mensajes no tiene respuesta por parte de los esclavos y se suelen emplear para mandar datos comunes de configuración, reset, etc.

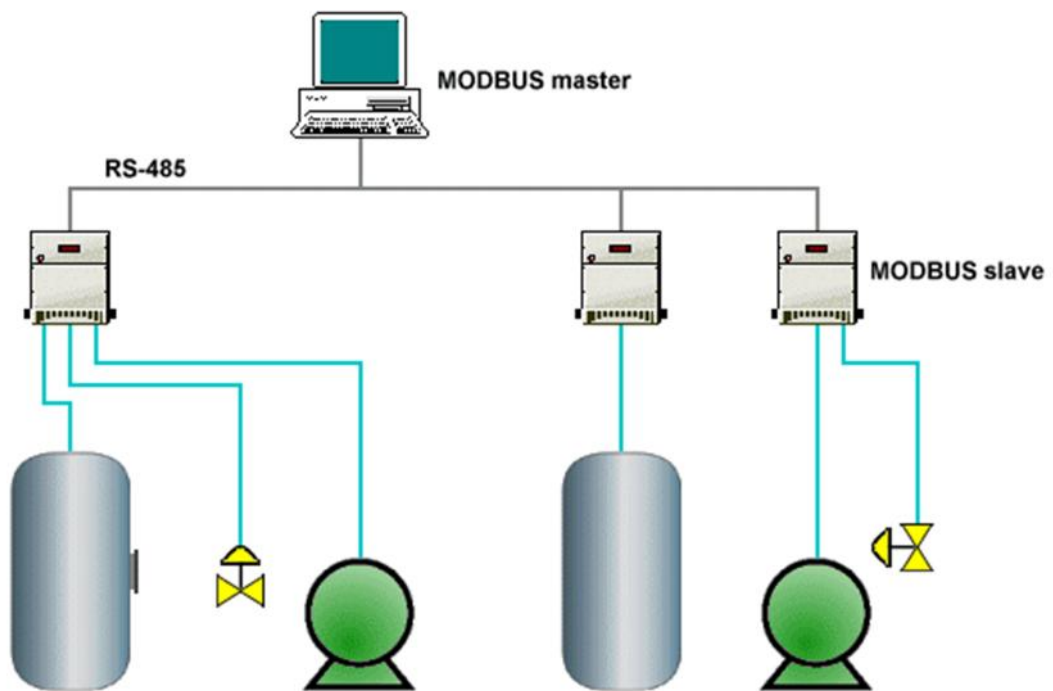


Figura 3. Estructura de Red Modbus

Fuente: (Barragan , 2013)

2.1.5 Estructura Lógica

El funcionamiento tiene una base muy sencilla: El Master (maestro) pregunta y los Slaves (esclavos) responden o actúan en función de lo que este diga.

El MODBUS siempre funciona con un maestro y uno o más esclavos, siendo el maestro quién controla en todo momento el inicio de la comunicación con los esclavos, solicitando información del resto de dispositivos conectados que ejercen como esclavos y son quienes suministran la información al

primero, que según la especificación pueden ser hasta 247 en una misma red. El esclavo por otro lado se limita a retornar los datos solicitados por el maestro, Cada esclavo debe tener una única dirección, así el maestro sabe con quién se debe comunicar.

- Cada esclavo tiene su propia dirección y se representa con un solo Byte, que puede ir desde 1 hasta 247. (Desde un punto de vista práctico, no pueden co-existir dos dispositivos esclavos con la misma dirección Modbus).
- El maestro siempre inicia la comunicación enviando un paquete de información bien estructurado a todos los esclavos, entre otras muchas cosas en la información se incluye el número del esclavo.
- El esclavo elegido responde, enviando lo que se le pide por medio también de un paquete de información bien estructurado.

2.1.6 Aplicación

Como se ha dicho a nivel general de buses de campo, el nivel de aplicación de no está cubierto por un software estándar, sino que cada fabricante suele suministrar programas para controlar su propia red. No obstante, el nivel de concreción en la definición de las funciones permite al usuario la confección de software propio para gestionar cualquier red, incluso con productos de distintos fabricantes.

Modbus permite el control de una red de dispositivos, por ejemplo un sistema de medida de temperatura y humedad, y comunicar los resultados a un ordenador. Modbus también se usa para la conexión de un ordenador de supervisión con una unidad remota (RTU) en sistemas de supervisión adquisición de datos (SCADA).

Existe gran cantidad de modems que aceptan el protocolo Modbus. Algunos están específicamente diseñados para funcionar con este protocolo. Existen implementaciones para conexión por cable, wireless, SMS o GPRS. La mayoría de problemas presentados hacen referencia a la latencia y a la sincronización.

2.1.7 Protocolos

Existen dos versiones principales: MODBUS RTU para la comunicación serial convencional, y MODBUS/TCP, que emplea Ethernet como medio físico de transmisión.

Funciones especialmente dignas de mención:

MODBUS RTU/ASCII

- Comunicación con 16 dispositivos (esclavos) por canal (hay disponibles 1 canal)
- Velocidad de transferencia de hasta 19,2 Kbps

MODBUS TCP/IP

- Comunicación con 64 dispositivos (esclavos) (empleando la función automática de comunicación)
- Velocidad de transferencia de 10/100 Mbaud

Los modos de transmisión definen como se envían los paquetes de datos entre maestros y esclavos, el protocolo MODBUS define dos principales modos de transmisión, con diferentes representaciones numéricas de los datos y detalles del protocolo ligeramente desiguales:

MODBUS RTU (Remote Terminal Unit). La comunicación entre dispositivos se realiza por medio de una representación binaria compacta de los datos.. Esta es la opción más usada del protocolo.

MODBUS ASCII (American Standard Code for Information Interchange). Es una representación legible del protocolo pero menos eficiente. La comunicación entre dispositivos se hace por medio de caracteres ASCII.

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).

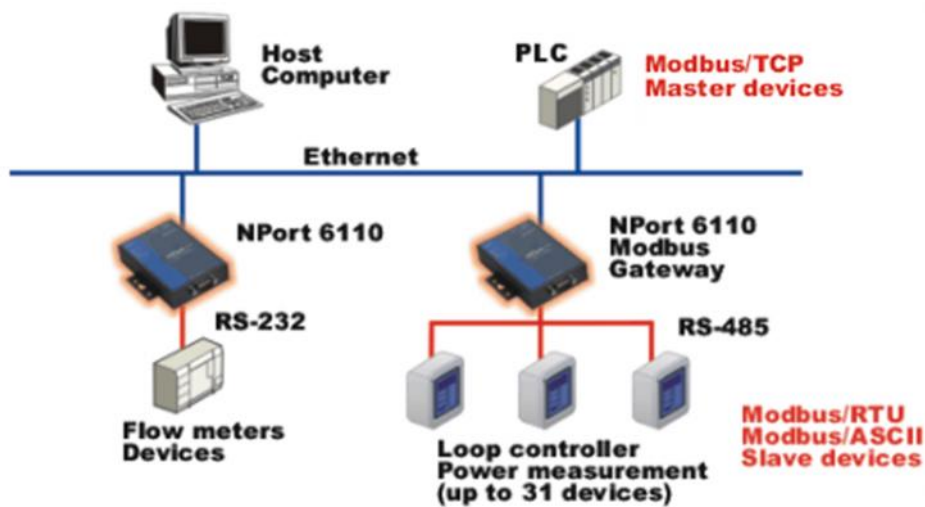


Figura 4. Versiones Modbus

Fuente: (Barragan , 2013)

2.1.8 Modbus RTU

Cuando los controladores son configurados para comunicarse en una red Modbus usando el modo RTU (Remote Terminal Unit), cada byte de 8 bits en un mensaje contiene dos dígitos hexadecimales de 4 bits. La principal ventaja de este modo es que su mayor densidad de carácter permite mejor rendimiento que el modo ASCII, para la misma velocidad. Cada mensaje debe ser transmitido en un flujo continuo. (Cevallos, 2007)

El formato para cada byte en modo RTU es:

Sistema de codificación: Binario 8-bits, hexadecimal 0-9, A-F.

Dos dígitos hexadecimales contenidos

En cada campo de 8 bits del mensaje.

Bits por byte:

1 bit de arranque.

8 bits de datos, el menos significativo se envía primero.

1 bit para paridad Par o Impar; ningún

bit para No paridad.

1 bit de paro si se usa paridad; 2 bits si no se usa paridad.

Campo de Comprobación de error: Comprobación Cíclica Redundante (CRC).

2.1.9 Trama RTU

En modo RTU, los mensajes comienzan con un intervalo silencioso de al menos 3.5 tiempos de carácter. Esto es más fácilmente implementado como un múltiplo de tiempos de carácter a la velocidad de transmisión configurada en la red (mostrado como T1-T2-T3-T4 en la tabla 1).

El primer campo transmitido es entonces la dirección del dispositivo destinatario. Los caracteres a transmitir permitidos para todos los campos son 0-A, A-F hexadecimal.

Los dispositivos conectados en red monitorizan el bus de red continuamente incluso durante los intervalos 'silencioso'. Cuando el primer campo (el campo de dirección) es recibido, cada dispositivo lo decodifica para enterarse si es el dispositivo direccionado. Siguiendo al último carácter transmitido, un intervalo de al menos 3.5 tiempos de carácter señala el final del mensaje. Un nuevo mensaje puede comenzar después de este intervalo.

La trama completa del mensaje debe ser transmitida como un flujo continuo. Si un intervalo silencioso de más de 1.5 tiempos de carácter tiene lugar antes de completar la trama, el dispositivo receptor desecha el mensaje incompleto y asume que el próximo byte será el campo de dirección de un nuevo mensaje.

De forma similar, si un nuevo mensaje comienza antes de que transcurran 3.5 tiempos de carácter después de un mensaje previo, el dispositivo receptor lo considerará una continuación del mensaje previo. Esto dará lugar a un error, ya que el valor en el campo final CRC no será válido para el mensaje combinado.

Tabla 1

Trama del mensaje RTU

INICIO	DIRECCIÓN	FUNCIÓN	DATOS	CRC	FINAL
T1-T2-T3-T4	8 BITS	8 BITS	N * 8 BITS	16 BITS	T1-T2-T3-T4

Fuente: (TIA Portal, 2013)

2.1.10 Mapa de direcciones Modbus

El protocolo Modbus, en su versión original, soporta 4 tipos de datos:

Salidas digitales (direcciones 00001-09999): son salidas físicas discretas. Requieren un bit que puede tomar los valores 0 o 1 y permiten acceso de escritura.

Entradas digitales (direcciones 10001-19999): son entradas físicas discretas. Requieren un bit que puede tomar los valores 0 o 1 y permiten acceso de escritura/lectura.

Entradas analógicas (direcciones tipo 30001-39999): son entradas físicas analógicas que funcionan con registros de 16 bits y que permiten acceso de escritura.

Salidas analógicas (direcciones 40001-49999): se trata de salidas físicas analógicas o registros internos del equipo. También conocidas con el nombre de holding registers. Registros de 16 bits que permiten acceso de lectura y de escritura.

Tabla 2

Estructura de los datos en Modbus

Tipo de Datos	Tipo de Acceso	Rango de Memoria
Salidas Digitales	Escritura	00001-09999
Entradas Digitales	Escritura / Lectura	10001-19999

Entradas Analógicas	Escritura	30001-39999
Registros de memoria	Escritura / Lectura	40001-49999

Fuente: (TIA Portal, 2013)

Módulo de Comunicación CM 1241 RS422/RS485

En los módulos de comunicaciones CM 1241 se dispone de los siguientes protocolos estándar: (Siemens, Hoja de datos 6ES7241-1CH32-0XB0, 2015)

ASCII:

Para la conexión a sistemas externos con protocolos de transferencia sencillos, por ejemplo protocolos con caracteres de inicio y de fin o protocolos con caracteres de control. El programa de usuario permite consultar y controlar las señales handshake de la interfaz.

MODBUS:

Para la comunicación según el protocolo MODBUS con formato RTU:

MODBUS maestro:

Conexión maestro-esclavo con SIMATIC S7 como maestro.

MODBUS esclavo:

Conexión maestro-esclavo con SIMATIC S7 como esclavo; no es posible el tráfico de telegramas de esclavo a esclavo.

Protocolo para accionamientos USS:

En particular, se admiten las instrucciones para la conexión de accionamientos conformes al protocolo USS. Los accionamientos intercambian datos en este caso a través de RS485. Con ello es posible controlar estos accionamientos, así como leer y escribir parámetros.



Figura 5. Módulo CM 1241 RS485

Fuente: (Siemens, Hoja de datos 6ES7241-1CH32-0XB0, 2015)

PLC S7 1200

El controlador lógico programable (PLC) S7-1200 ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización.

Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones. (Siemens, Controlador Programable S7 1200, 2009)

La CPU incorpora un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y salida en una carcasa compacta, conformando así un potente PLC.

Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de

usuario, que puede incluir lógica booleana, instrucciones de contaje y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

La CPU incorpora un puerto PROFINET para la comunicación en una red PROFINET. Los módulos de comunicación están disponibles para la comunicación en redes RS485 o RS232.

2.1.11 Partes del PLC S7 1200

Los diferentes modelos de CPUs ofrecen una gran variedad de funciones y prestaciones que permiten crear soluciones efectivas destinadas a numerosas aplicaciones.

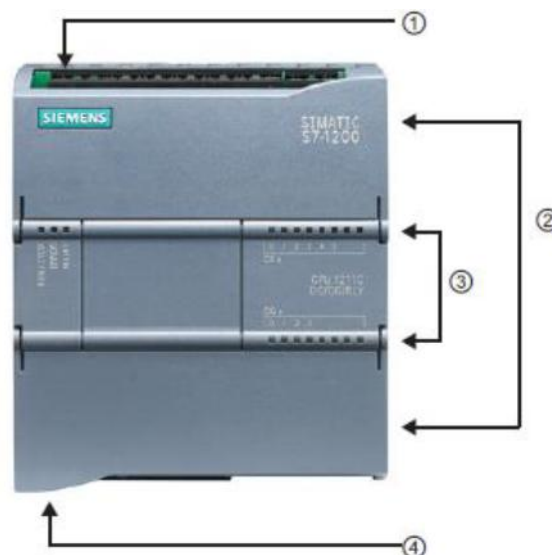


Figura 6 Partes del PLC S7 1200

Fuente: (Siemens, Controlador Programable S7 1200, 2009)

- ① Conector de corriente
- ② Conectores extraíbles para el cableado de usuario (detrás de las tapas)
- ② Ranura para Memory Card (debajo de la tapa superior)
- ③ LEDs de estado para las E/S integradas
- ④ Conector PROFINET (en el lado inferior de la CPU)

2.1.12 Tarjeta de señales

Una Signal Board (SB) permite agregar E/S a la CPU. Es posible agregar una SB con E/S digitales o analógicas. Una SB se conecta en el frente de la CPU.

- SB con 4 E/S digitales (2 entradas DC y 2 salidas DC)
- SB con 1 entrada analógica

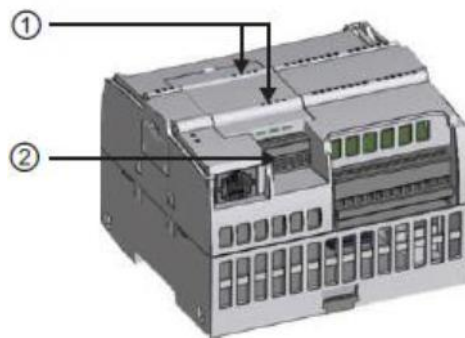


Figura 7. Signal Board

Fuente: (Siemens, Controlador Programable S7 1200, 2009)

- ① LEDs de estado en la SB
- ② Conector extraíble para el cableado de usuario

2.1.13 Módulo de señales

Los módulos de señales se pueden utilizar para agregar funciones a la CPU. Los módulos de señales se conectan a la derecha de la CPU.

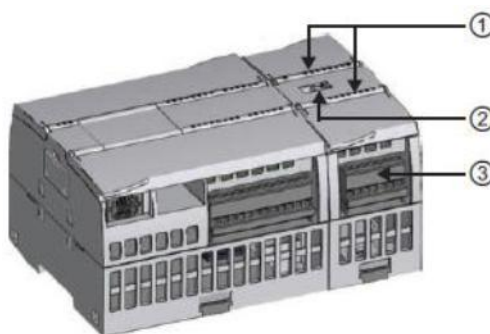


Figura 8. Partes de un módulo de señal

Fuente: (Siemens, Controlador Programable S7 1200, 2009)

- ① LEDs de estado para las E/S del módulo de señales

- ② Conector de bus
- ③ Conector extraíble para el cableado de usuario

2.1.14 Módulo de comunicación

La gama S7-1200 provee módulos de comunicación (CMs) que ofrecen funciones adicionales para el sistema. Hay dos módulos de comunicación, a saber: RS232 y RS485.

- La CPU soporta como máximo 3 módulos de comunicación
- Todo CM se conecta en lado izquierdo de la CPU (o en lado izquierdo de otro CM)

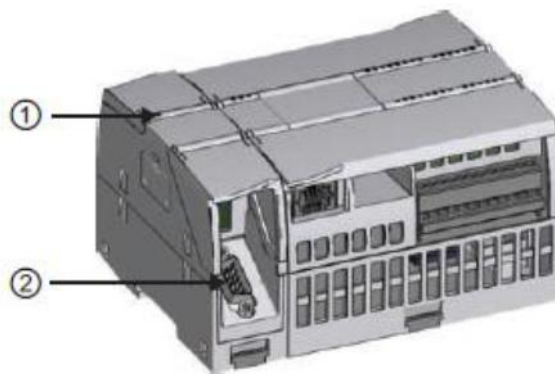


Figura 9. Partes de un módulo de comunicación

Fuente: (Siemens, Controlador Programable S7 1200, 2009)

- ① LEDs de estado del módulo de comunicación
- ② Conector de comunicación

2.1.15 Almacenamiento de datos, áreas de memoria y direccionamiento

La CPU ofrece varias opciones para almacenar datos durante la ejecución del programa de usuario:

Memoria global: La CPU ofrece distintas áreas de memoria, incluyendo entradas (I), salidas (Q) y marcas (M). Todos los bloques lógicos pueden acceder sin restricción alguna a esta memoria.

Bloque de datos (DB): Es posible incluir DBs en el programa de usuario para almacenar los datos de los bloques lógicos. Los datos almacenados se conservan cuando finaliza la ejecución del bloque lógico asociado. Un DB "global" almacena datos que pueden ser utilizados por todos los bloques lógicos, mientras que un DB instancia almacena datos para un bloque de función (FB) específico y está estructurado según los parámetros del FB.

Memoria temporal: Cada vez que se llama un bloque lógico, el sistema operativo de la CPU asigna la memoria temporal o local (L) que debe utilizarse durante la ejecución del bloque. Cuando finaliza la ejecución del bloque lógico, la CPU reasigna la memoria local para la ejecución de otros bloques lógicos.

A los datos de la mayoría de las áreas de memoria (I, Q, M, DB y L) se puede acceder como bytes, palabras o palabras dobles utilizando el formato "dirección de byte". Para acceder a un byte, una palabra o una palabra doble de datos en la memoria, la dirección debe especificarse de forma similar a la dirección de un bit. Esto incluye un identificador de área, el tamaño de los datos y la dirección de byte inicial del valor de byte, palabra o palabra doble. Los designadores de tamaño son B (byte), W (palabra) y D (palabra doble), p. ej. IB0, MW20 ó QD8. Las direcciones tales como I0.3 y Q1.7 acceden a la memoria imagen de proceso. Para acceder a la entrada o salida física es preciso añadir ":P" a la dirección (p. ej. I0.3:P, Q1.7:P o "Stop:P"). (Siemens, Controlador Programable S7 1200, 2009)

2.1.16 Acceso a los datos en las áreas de memoria de la CPU

I (memoria imagen de proceso de las entradas): La CPU consulta las entradas de periferia (físicas) inmediatamente antes de ejecutar el OB de ciclo en cada ciclo y escribe estos valores en la memoria imagen de proceso de las entradas. A la memoria imagen de proceso de las entradas se puede acceder en formato de bit, byte, palabra o palabra doble. Aunque se permiten accesos de lectura y escritura, generalmente sólo se leen las entradas de la memoria imagen de proceso.

Tabla 3

Memoria del proceso de las entradas

Bit	M[dirección de byte][dirección de bit]	I0.1
Byte, palabra, palabra doble	M[tamaño][dirección de byte inicial]	IB4, IW5, ID12

Fuente: (TIA Portal, 2013)

Q (memoria imagen de proceso de las salidas): La CPU copia los valores almacenados en la imagen de proceso de las salidas en las salidas físicas. A la memoria imagen de proceso de las salidas se puede acceder en formato de bit, byte, palabra o palabra doble. Se permiten accesos de lectura y escritura a la memoria imagen de proceso de las salidas.

Tabla 4

Memoria del proceso de salidas

Bit	Q[dirección de byte][dirección de bit]	Q1.1
Byte, palabra, palabra doble	M[tamaño][dirección de byte inicial]	QB5, QW10, QD40

Fuente: (TIA Portal, 2013)

M (área de marcas): El área de marcas (memoria M) puede utilizarse para relés de control y datos para almacenar el estado intermedio de una operación u otra información de control. Al área de marcas se puede acceder en formato de bit, byte, palabra o palabra doble. Se permiten accesos de lectura y escritura al área de marcas.

Tabla 5

Área de marcas

Bit	M[dirección de byte][dirección de bit]	M26.7
Byte, palabra, palabra doble	M[tamaño][dirección de byte inicial]	MB20,MW30,MD50

Fuente: (TIA Portal, 2013)

Temp (memoria temporal): La CPU asigna la memoria temporal según sea necesario. La CPU asigna la memoria temporal al bloque lógico cuando éste se inicia (en caso de un OB) o se llama (en caso de una FC o un FB). La asignación de la memoria temporal a un bloque lógico puede reutilizar las mismas posiciones de memoria temporal usadas anteriormente por un OB, FC o FB diferente. La CPU no inicializa la memoria temporal durante la asignación. Por lo que esta memoria puede contener un valor cualquiera. La memoria temporal es similar al área de marcas, con una excepción importante: el área de marcas tiene un alcance "global", en tanto que la memoria temporal tiene un alcance "local".

DB (bloque de datos): Los bloques de datos se utilizan para almacenar diferentes tipos de datos, incluyendo el estado intermedio de una operación u otros parámetros de control de FBs, así como estructuras de datos requeridas para numerosas instrucciones, p. ej. temporizadores y contadores. Es posible determinar que un bloque de datos sea de lectura/escritura o de sólo lectura. A los bloques de datos se puede acceder en formato de bit, byte, palabra o palabra doble. A los bloques de datos que se pueden leer y escribir se permiten accesos de lectura y escritura. A los bloques de datos de sólo lectura se permiten sólo los accesos de lectura.

TIA PORTAL

Tía portal realiza conversiones implícitas de tipos de datos en las instrucciones en las que tipos de datos más pequeños (como SInt o Byte) se

convierten automáticamente a tipos de datos más grandes (como DInt, DWord, Real o LReal). (Siemens, Controlador Programable S7 1200, 2009)

Por ejemplo, un valor entero (Int) se convertirá automáticamente a entero doble (DInt) o a Real por una instrucción que esté configurada para utilizar DInt o Real. No hace falta utilizar una instrucción adicional de conversión para convertir el valor.

Para la ingeniería de un sistema de automatización se ha popularizado varias herramientas de configuración. Gracias al framework de ingeniería que ofrece el Portal de Totally Integrated Automation (TIA Portal) prácticamente desaparecen las fronteras entre estos productos de software. En el futuro, este framework será la base de todos los sistemas de ingeniería para la configuración, programación y puesta en marcha de autómatas/controladores (PLC), sistemas de supervisión / pantallas y accionamientos incluidos en Totally Integrated Automation. Un auténtico hito Totally Integrated Automation Portal reúne todas las herramientas de software de automatización dentro de un único entorno de desarrollo. Con el primer software de la industria con un solo entorno de ingeniería, TIA Portal supone un hito en el desarrollo de software. Un proyecto de software único para todas las tareas de automatización.

2.1.17 Apariencia homogénea

El framework de ingeniería común en el que están integrados los productos de Software unifica todas las funciones comunes, incluso en lo relativo a su representación en la pantalla. La unificación del manejo de distintos editores facilita la tarea de aprendizaje y permite al usuario concentrarse en lo esencial de su trabajo.

2.1.18 Inteligencia integrada

Editores inteligentes muestran de modo contextualizado justo lo que el usuario necesita en el momento para la tarea que esté realizando: funciones, propiedades, librerías, etc. El método de la pantalla partida permite tener abiertos varios editores a la vez e intercambiar datos entre ellos. Este

intercambio de datos se ejecuta con facilidad mediante la función “Arrastrar y colocar”.

2.1.19 Máxima transparencia de datos

Sólo es necesario introducir una vez los datos cuando se utilizan en distintos editores y para sistemas de destino diferentes. Gracias a la gestión de datos centralizada y orientada al objeto que ofrece el TIA Portal, los datos de aplicación modificados se actualizan automáticamente para todos los equipos (PLC y HMI) implicados en el proyecto. La base de datos compartida garantiza una consistencia absoluta en todo el proyecto de automatización.

2.1.20 Soluciones reutilizables

En librerías claramente estructuradas se administran bloques de programa incluidos en el suministro o creados por el propio usuario, así como equipos y módulos ya configurados.

En el TIA Portal también se pueden reutilizar bloques o proyectos enteros, creados con versiones anteriores de los productos de software integrados en el TIA Portal. La reutilización reduce el trabajo de ingeniería y, al mismo tiempo, incrementa la calidad del sistema de automatización.

Instrucciones Modbus en PLC S7 1200

2.1.21 MB_COMM_LOAD

La instrucción MB_COMM_LOAD configura un puerto del módulo de comunicación punto a punto (PtP) CM 1241 RS485 o CM 1241 RS232 para la comunicación vía el protocolo Modbus RTU. (Siemens, Controlador Programable S7 1200, 2009)

El bloque MB_COMM_LOAD se usa para seleccionar el módulo de comunicación, ajustar los parámetros de la comunicación y parametrizar el enlace con los parámetros del maestro o el esclavo. El bloque MB_COMM_LOAD se debe llamar en el primer ciclo de programa (mediante la activación de la marca de sistema M1.0 en los ajustes de hardware o

mediante una llamada en el OB 100 de arranque). Después de insertar el módulo de comunicación en la configuración de hardware se puede seleccionar el nombre simbólico del módulo de comunicación en el parámetro PORT. Los parámetros de comunicación BAUD (velocidad de transmisión) y PARITY (paridad) deben ser idénticos para todos los nodos. La configuración del puerto de la interfaz RS485 en la vista de equipo del STEP 7 (TIA Portal) es irrelevante en este caso. El bloque de datos de instancia del bloque maestro o del bloque esclavo se transfiere en el parámetro MB_DB y, por consiguiente, define el módulo de comunicación (parámetro PORT) como maestro o esclavo MODBUS.

MB_COMM_LOAD se ejecuta para configurar un puerto para el protocolo Modbus RTU. Una vez configurado el puerto, la comunicación se realiza en el Modbus ejecutando las instrucciones MB_SLAVE o MB_MASTER.

MB_COMM_LOAD se debería llamar una vez para inicializar el puerto. Sólo es necesario volver a llamar MB_COMM_LOAD si debe cambiar uno de los parámetros de comunicación.

Es posible llamar MB_COMM_LOAD desde un OB de arranque y ejecutar esta instrucción una vez, o bien utilizar la marca de sistema del primer ciclo con el fin de iniciar la llamada para ejecutarla una vez.

Es preciso utilizar una instancia de MB_COMM_LOAD para configurar cada uno de los puertos de todo módulo de comunicación utilizado para la comunicación Modbus. Un bloque de datos instancia MB_COMM_LOAD unívoco se debe asignar a cada puerto utilizado. La CPU S7-1200 está limitada a 3 módulos de comunicación.

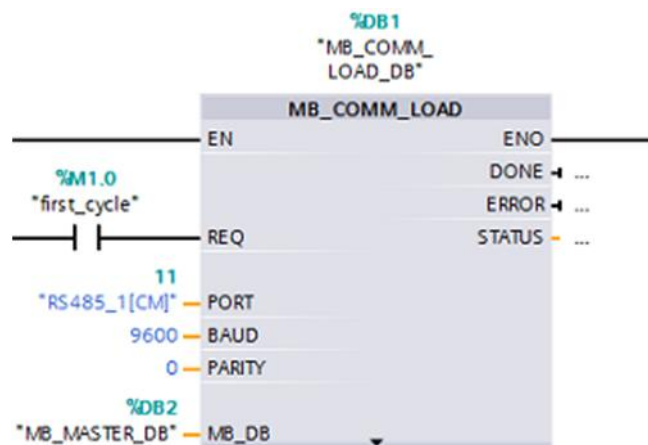


Figura 10. Bloque MB_COMM_LOAD

Fuente: (TIA Portal, 2013)

2.1.22 MB_MASTER

La instrucción MB_MASTER permite al programa comunicarse como maestro Modbus utilizando un puerto del módulo de comunicación punto a punto (PtP) CM 1241 RS485 o CM 1241 RS232. Es posible acceder a los datos de uno o más esclavos Modbus. (Siemens, Controlador Programable S7 1200, 2009)

Un bloque de datos instancia se asigna cuando la instrucción MB_MASTER se inserta en el programa. El nombre de este bloque de datos instancia MB_MASTER se utiliza cuando se especifica el parámetro MB_DB de la instrucción MB_COMM_LOAD.

Reglas de comunicación del maestro Modbus

- MB_COMM_LOAD debe ejecutarse para configurar un puerto antes de que la instrucción MB_MASTER pueda comunicarse con ese puerto.
- Si un puerto debe utilizarse para iniciar peticiones de maestro Modbus, MB_SLAVE no podrá utilizar este puerto. Una o más instancias de ejecución de MB_MASTER pueden utilizarse en este puerto.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa

debe consultar la instrucción MB_MASTER para transmitir y recibir condiciones completas.

- Si el programa opera un maestro Modbus y utiliza MB_MASTER para enviar una petición a un esclavo, MB_MASTER se deberá seguir ejecutando hasta que se devuelva la respuesta del esclavo.
- Todas las ejecuciones de MB_MASTER para un determinado puerto deben llamarse desde un mismo OB (o clase de prioridad de OB).

Parámetro REQ

Si el valor de REQ es FALSE (falso), no hay ninguna petición.

Si el valor de REQ es TRUE (verdadero), hay una petición de transmitir datos a uno o varios esclavos Modbus.

Esta entrada debe recibir suministro de un contacto disparado por un flanco ascendente en la primera llamada de la ejecución de MB_MASTER. El impulso disparado por flancos llama una vez la petición de transmisión. Todas las entradas se capturan y no se modifican durante una petición y respuesta disparadas por esta entrada.

MB_MASTER inicia internamente una máquina de estados para asegurarse de que ninguna otra MB_MASTER pueda lanzar una petición hasta que no haya finalizado esta petición.

Además, si la misma instancia de la llamada del FB MB_MASTER se ejecuta nuevamente con la entrada REQ = TRUE (verdadero) antes de que se finalice la petición, no se realizarán más transmisiones. No obstante, en cuanto finalice la petición, se lanzará otra petición si MB_MASTER se ejecuta con la entrada REQ puesta a TRUE.

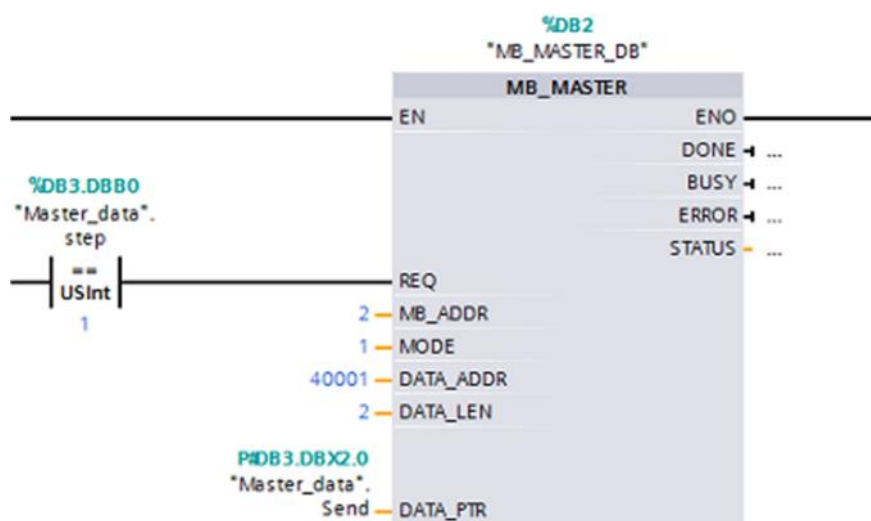


Figura 11. Bloque MB_MASTER

Fuente: (TIA Portal, 2013)

Tabla 6

Parámetros del bloque MB_MASTER

Parámetro	Descripción
REQ	Habilita la comunicación.
MB_ADDR	Dirección estación MODBUS-RTU.
MODE	Seleccionar la dirección de la transmisión ("0" = lectura, "1" = escritura o funciones de diagnóstico adicionales).
DATA_ADDR	Especifica la dirección de inicio MODBUS.
DATA_LEN	Especifica la longitud de datos MODBUS.
DATA_PTR	Define las zonas de datos de envío y recepción locales del maestro. El parámetro DATA_PTR debe hacer referencia a un bloque de datos global creado con la opción "Sólo acceso simbólico" deshabilitada.

Fuente: (TIA Portal, 2013)

2.1.23 MB_SLAVE

La instrucción MB_SLAVE permite al programa comunicarse como esclavo Modbus utilizando un puerto del módulo de comunicación punto a punto (PtP) CM 1241 RS485 o CM 1241 RS232. Un maestro Modbus RTU puede lanzar una petición y el programa responde ejecutando la instrucción MB_SLAVE. (Siemens, Controlador Programable S7 1200, 2009)

Es preciso asignar un bloque de datos instancia unívoco al insertar la instrucción MB_SLAVE en el programa. El nombre de este bloque de datos instancia MB_SLAVE se utiliza cuando se especifica el parámetro MB_DB de la instrucción MB_COMM_LOAD.

Los códigos de las funciones de comunicación Modbus (1, 2, 4, 5 y 15) pueden leer y escribir bits y palabras directamente en la memoria imagen de proceso de las entradas y salidas del PLC. La tabla siguiente muestra el mapeo de las direcciones Modbus en la memoria imagen de proceso de la CPU.

El bloque MB_SLAVE se usa para definir la dirección de estación MODBUS-RTU y especificar la zona de almacenamiento de datos local para la transferencia de datos con el registro de participación.

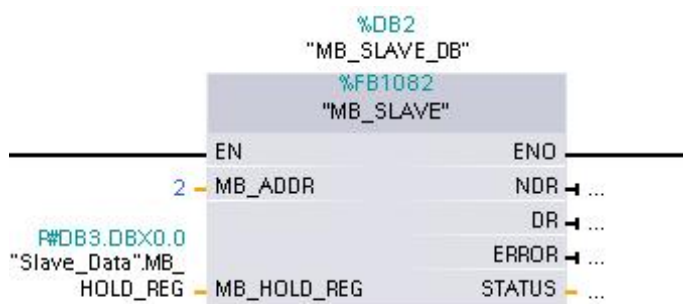


Figura 12. Bloque MB_SLAVE

Fuente: (TIA Portal, 2013)

Tabla 7

Parámetros del Bloque MB_SLAVE

Parámetro	Descripción
MB_ADDR	Transfiere la dirección estación MODBUS-RTU.
MB_HOLD_REG	Registro de participación del esclavo. El parámetro MB_HOLD_REG debe hacer referencia a un bloque de datos global creado con la opción "Sólo acceso simbólico" deshabilitada.

Fuente: (TIA Portal, 2013)

Cuando se tienen que transferir múltiples registros (palabras), con el parámetro "MB_HOLD_REG" se tiene que transferir el nombre simbólico de

un array o estructura que cubra el volumen de datos especificado del maestro (DATA_ADDR y DATA_PTR). La primera palabra de este array o estructura se corresponde con la dirección de inicio 40001 del registro de participación de MODBUS.

Los parámetros se pueden conmutar con las entradas y salidas de los bloques o mediante el acceso al correspondiente bloque de datos de instancia.

CAPÍTULO III

COMUNICACIÓN MODBUS RTU ENTRE DOS PLCs S7 1200

Preliminares

Para la implementación de la comunicación Modbus RTU entre 2 PLCs S7 1200, fue necesario los siguientes dispositivos:

- 2 PLCs S7 1200 CPU 1214C AC/DC/RLY
- 2 Módulos de comunicación CM 1241 RS422/RS485
- 1 Switch CSM 1277 Siemens o cualquier marca
- 3 cables de red Ethernet
- 1 cable Profibus
- 1 computador con TIA Portal V12

Conexiones físicas de los dispositivos

Los dispositivos empleados para la comunicación Modbus entre 2 PLCs, se conectaron como se muestra en la figura 13.

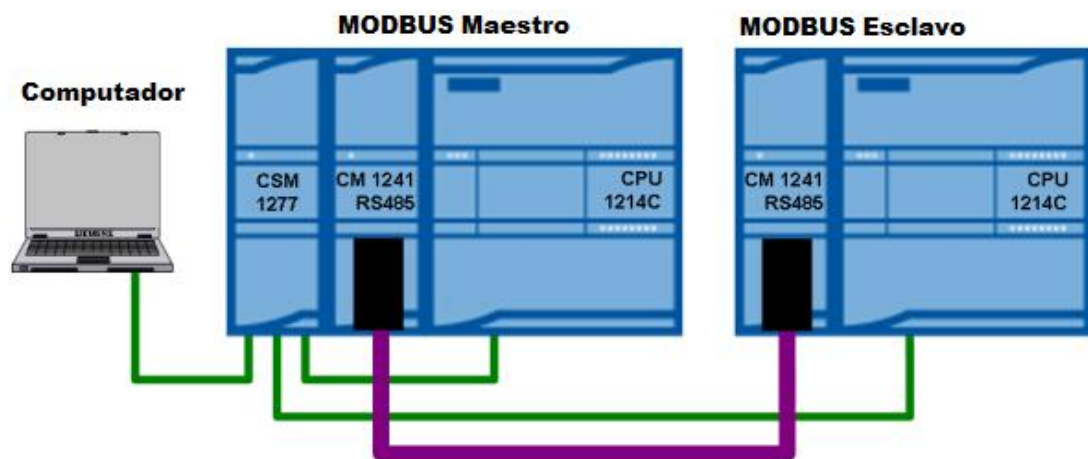


Figura 13. Conexión física de la comunicación Modbus

Fuente: (Cevallos, 2007)

Las líneas de color verde representan los cables Ethernet, mientras que las de color morado, representa el cable Profibus. Los módulos de comunicación CM 1241 RS485, se conectan en la parte izquierda de cada PLC. El Switch industrial marca siemens CSM 1277 puede ser reemplazado por otro de cualquier otra marca, en vista que las prácticas se realizaran en el laboratorio.

Configuración de la comunicación en TIA Portal V12

Para configurar la comunicación Modbus entre 2 PLCs empleando los módulos CM 1241 RS 485, se procede de la siguiente manera:

1. Abrir TIA Portal V12, crear un nuevo proyecto y agregar el PLC, para el presente trabajo el nombre es RedModbus.

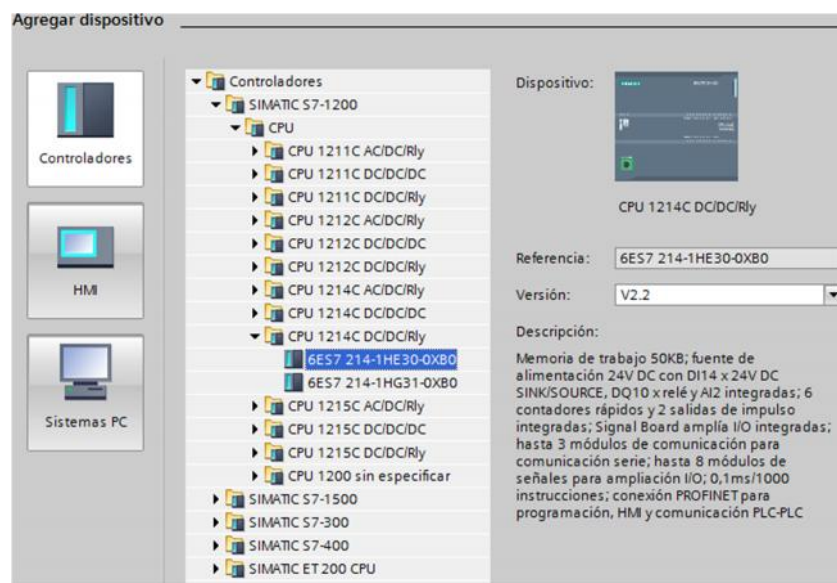


Figura 14. Agregar PLC al proyecto

Fuente: (TIA Portal, 2013)

2. Abierto el proyecto, agregar otro PLC de las mismas características del anterior, desde el árbol del proyecto

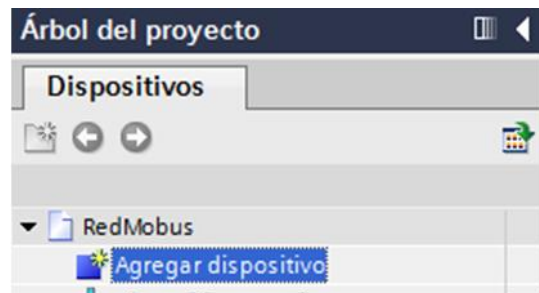


Figura 15. Opción Agregar dispositivo

Fuente: (TIA Portal, 2013)

3. Agregados los dos PLCs al proyecto, colocar el nombre de Maestro al uno y Esclavo al otro.

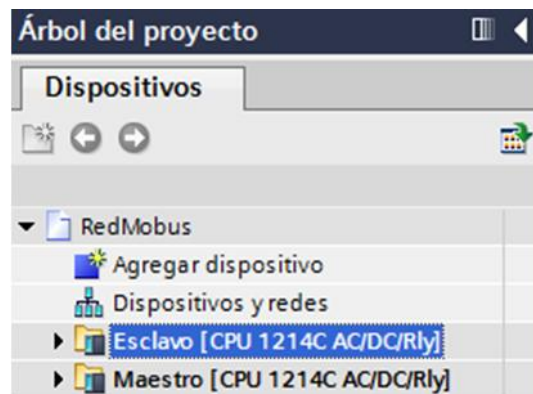


Figura 16. Asignación de nombres a los PLCs

Fuente: (TIA Portal, 2013)

4. Abrir Dispositivos y redes desde el Árbol del proyecto, conecte los dos PLCs mediante sus puertos Profinet.

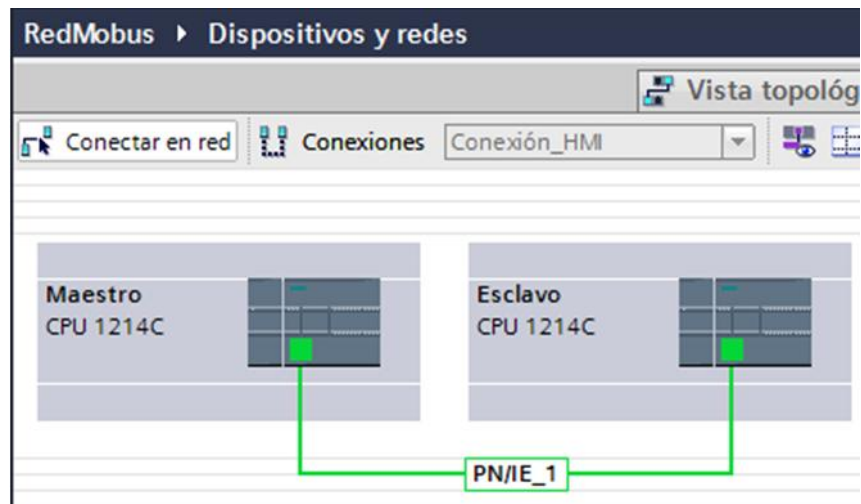


Figura 17. Conexión Profinet entre PLCs

Fuente: (TIA Portal, 2013)

5. Abrir la carpeta correspondiente al PLC llamado Esclavo, dar doble clic en configuración de dispositivos y luego doble clic en el puerto Profinet (color verde), agregar subred y asignar la dirección IP 192.168.0.2.

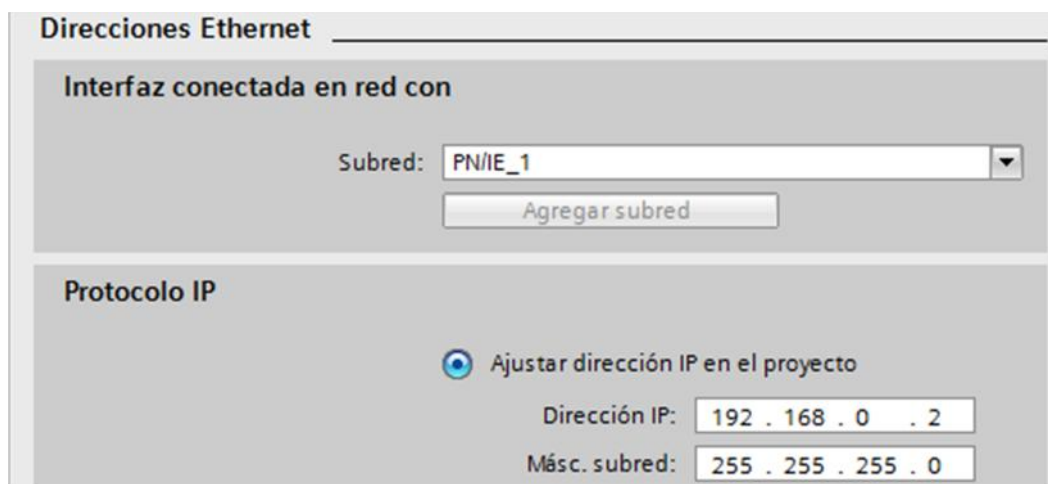


Figura 18. Asignación de IP al PLC esclavo

Fuente: (TIA Portal, 2013)

6. Repita los mismos pasos para el PLC Maestro



Figura 19. Asignación de IP al PLC maestro

Fuente: (TIA Portal, 2013)

7. Abra las propiedades de cada PLC y active los bytes de marca del sistema, necesarios para configurar la comunicación.

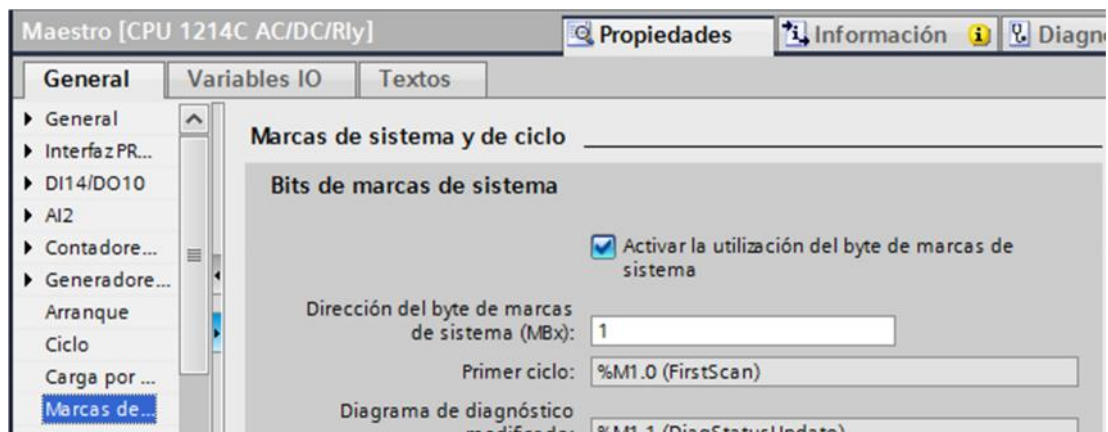


Figura 20. Activación bytes de marca de sistema

Fuente: (TIA Portal, 2013)

8. Agregar los módulos de comunicación a cada PLC desde el catálogo de hardware.



Figura 21. Selección del CM 1241

Fuente: (TIA Portal, 2013)

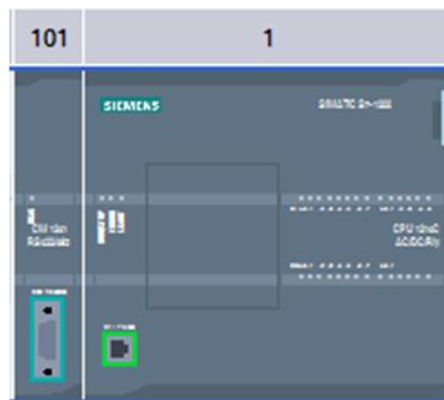


Figura 22. CM 1241 conectado al PLC

Fuente: (TIA Portal, 2013)

9. Cargar la configuración a los 2 PLCs

Programación del maestro Modbus

El maestro Modbus envía un valor entero 11111 por la dirección 40001 y el valor de un contador ascendente por la dirección 40002 al esclavo. El esclavo envía el valor entero 22222 y el valor del contador dividido para 2. Todo este el maestro lo realiza en dos pasos.

3.1.1 Creación del bloque de datos del maestro

La comunicación Modbus intercambia información mediante bloque de datos creados en cada PLC.

En el PLC maestro se creó un bloque de datos globales llamados Datos_Master desde la opción Agregar nuevo bloque del PLC maestro.

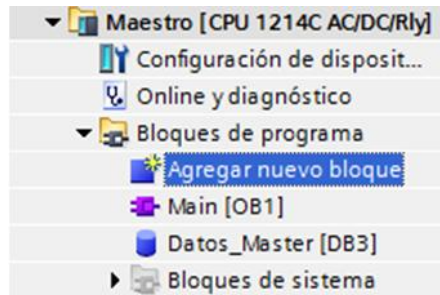


Figura 23. Agregar bloque de datos maestro

Fuente: (TIA Portal, 2013)

Una vez creado el bloque de datos, ingresar a propiedades haciendo un clic derecho sobre el bloque, en la opción Atributos, desmarcar Acceso optimizado al bloque, pulse aceptar.

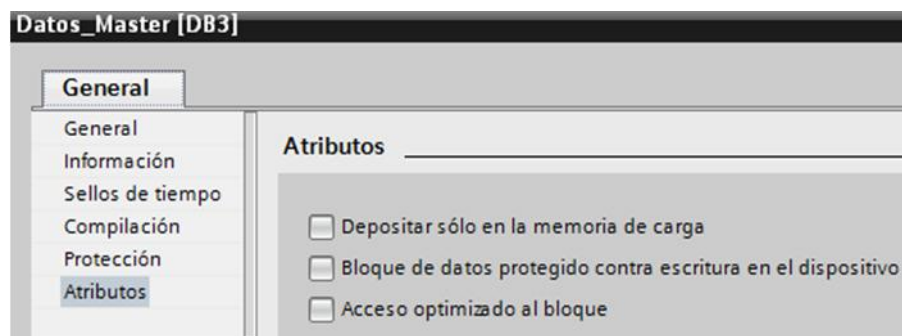


Figura 24. Deshabilitar Acceso optimizado al bloque

Fuente: (TIA Portal, 2013)

Crear el bloque de datos que se indica en la figura 25

Datos_Master			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	step	USInt	1
3	edge	Bool	false
4	Send	Array [0..1] of Int	
5	Send[0]	Int	11111
6	Send[1]	Int	0
7	Receive	Array [0..1] of Int	
8	Receive[0]	Int	0
9	Receive[1]	Int	0

Figura 25. Bloque de datos del maestro

Fuente: (TIA Portal, 2013)

El dato step, sirve para seleccionar los pasos que realiza el maestro para la comunicación.

El dato edge, detecta un flanco positivo para incrementar el valor a ser enviado al esclavo.

El dato Send es de tipo Array, contiene dos datos tipo Int, el primero llamado Send[0], que envía el valor 11111 al esclavo y Send[1], que envía el valor del contador al esclavo.

El dato Receive es de tipo Array, contiene dos datos de tipo Int, el primero llamado Receive[0], recibe el dato 22222 enviado por el esclavo y Receive[1], recibe el dato del contador enviado por el esclavo dividido por 2.

3.1.2 Configuración del puerto del maestro

Mediante MB_COMM_LOAD, se configura el protocolo de comunicación Modbus, especificando el puerto, la velocidad de transmisión, la paridad y el bloque de datos del maestro.

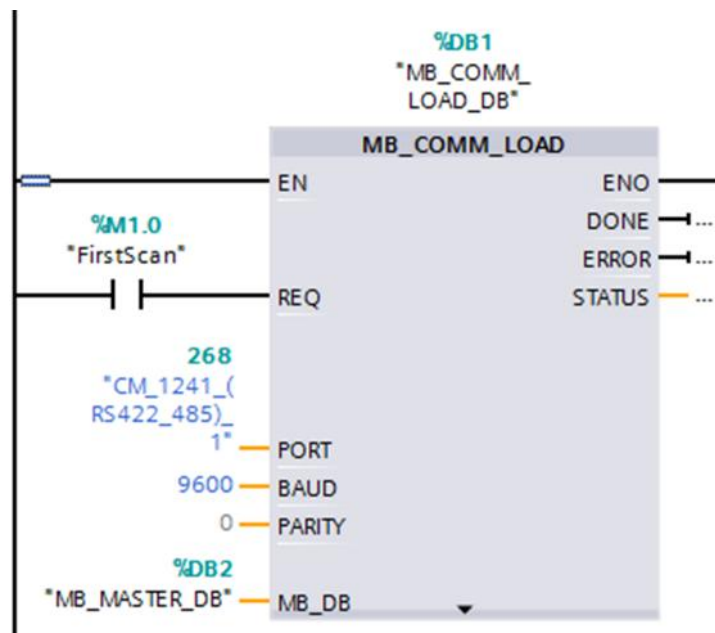


Figura 26. Protocolo de comunicación Modbus maestro

Fuente: (TIA Portal, 2013)

El puerto es el 268, la velocidad de transmisión es de 9600, sin paridad y el bloque de datos DB2 que corresponde a la instrucción `MB_MASTER_DB`.

La instrucción MB_COMM_LOAD_DB solo se ejecuta una vez, por lo que se utiliza el bit de marca de sistema FirstScan el mismo que solo se activa cuando el PLC realiza el primer escaneo.

3.1.3 Programación del maestro

Cuando el dato step tiene un valor de 1, el maestro envía el Array de datos Send hacia el esclavo. Este Array contiene el valor 11111 y el valor del contador. Este envío se realiza mediante la instrucción MB_MASTER.

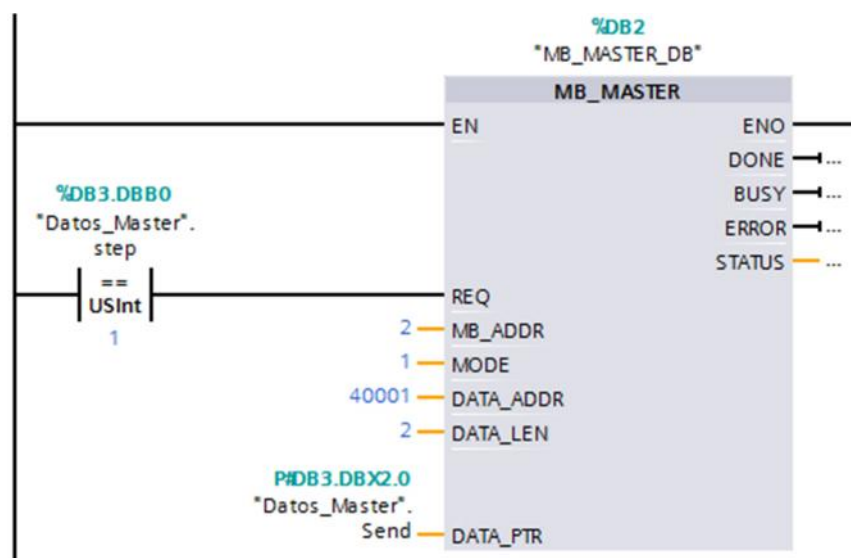


Figura 27. Envío de datos desde el maestro

Fuente: (TIA Portal, 2013)

La instrucción MB_MASTER_DB es llamada cuando la entrada REQ se activa; es decir, cuando el dato step es igual a 1. En la entrada MB_ADDR se especifica la dirección del esclavo, en MODE se coloca 1 porque se realiza una operación de escritura, en DATA_ADDR se coloca la dirección del primer holding registers que es la 40001, en DATA_LEN se coloca 2 porque se utiliza para enviar datos al esclavo la dirección 40001 y 40002, en DATA_PTR se especifica el dato tipo Array a ser enviado.

El contador se envía mediante el siguiente segmento.

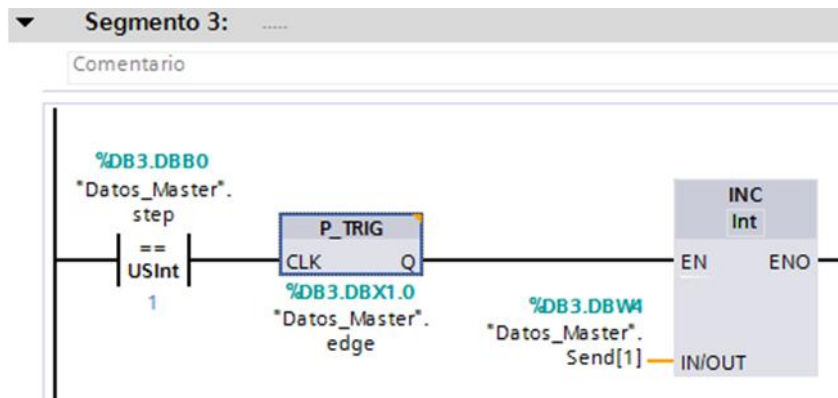


Figura 28. Incremento de dato enviado

Fuente: (TIA Portal, 2013)

Cuando el maestro envía, éste solicita al esclavo que le envíe datos, mediante el siguiente segmento.



Figura 29. Petición de envío al esclavo

Fuente: (TIA Portal, 2013)

La transición al paso 2 se realiza con la confirmación positiva (DONE) o negativa (ERROR) del MB_MASTER.

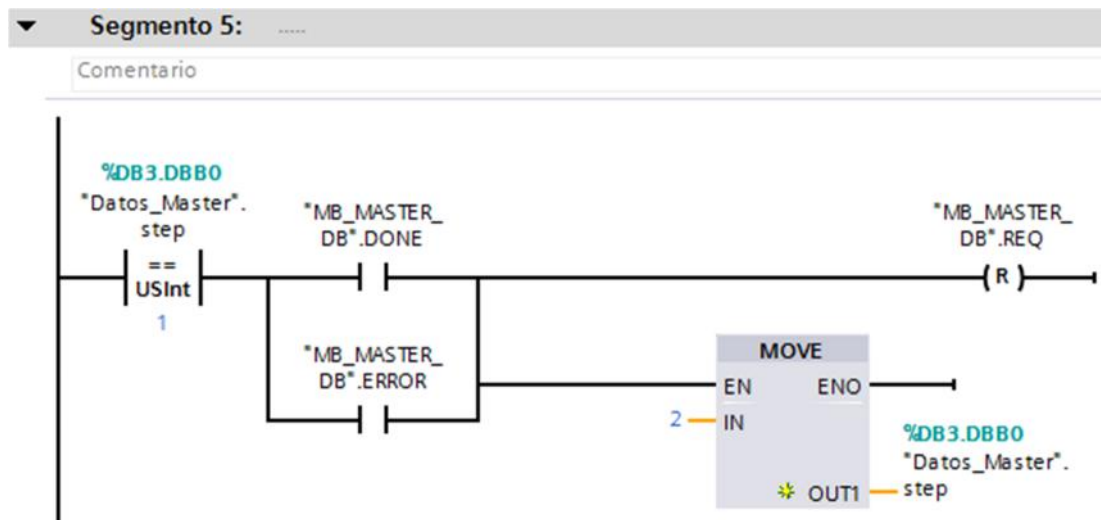


Figura 30. Transición al paso 2 del esclavo

Fuente: (TIA Portal, 2013)

En el paso 2 el maestro recibe los datos enviados por el esclavo, de la siguiente forma.

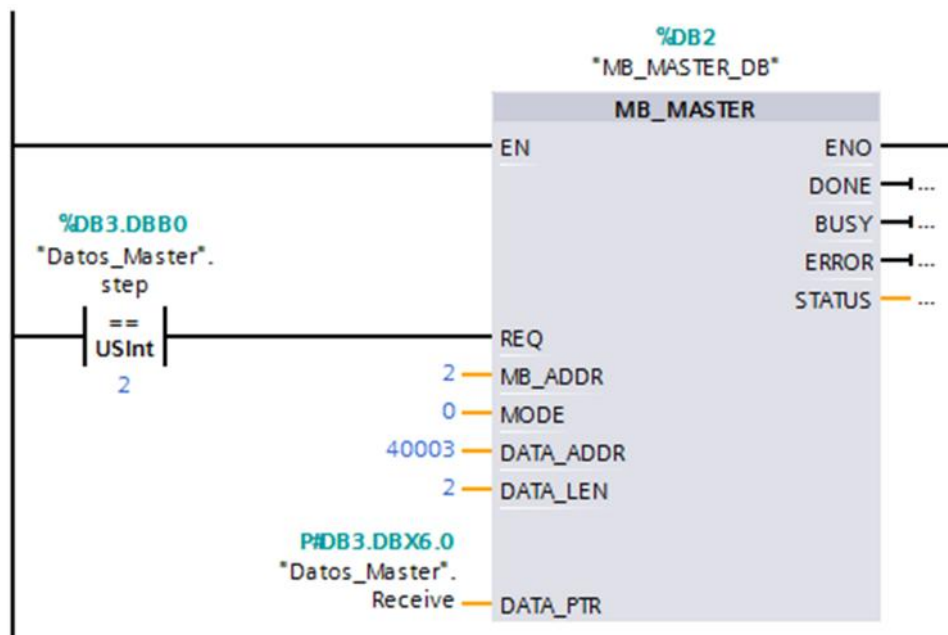


Figura 31. Lectura de datos del esclavo

Fuente: (TIA Portal, 2013)

La instrucción MB_MASTER_DB es llamada cuando la entrada REQ se activa; es decir, cuando el dato step es igual a 2. En la entrada MB_ADDR se especifica la dirección del esclavo, en MODE se coloca 0 porque se realiza una operación de lectura, en DATA_ADDR se coloca la dirección del holding

registers que es la 40003, en DATA_LEN se coloca 2 porque se utiliza para recibir datos del esclavo la dirección 40003 y 40004, en DATA_PTR se especifica el dato tipo Array a ser recibido.

En el paso 2 nuevamente el maestro solicita respuesta del esclavo a través del siguiente segmento.



Figura 32. Maestro solicita respuesta del esclavo

Fuente: (TIA Portal, 2013)

Si la confirmación es positiva el maestro regresa al paso 1 mediante el siguiente segmento.

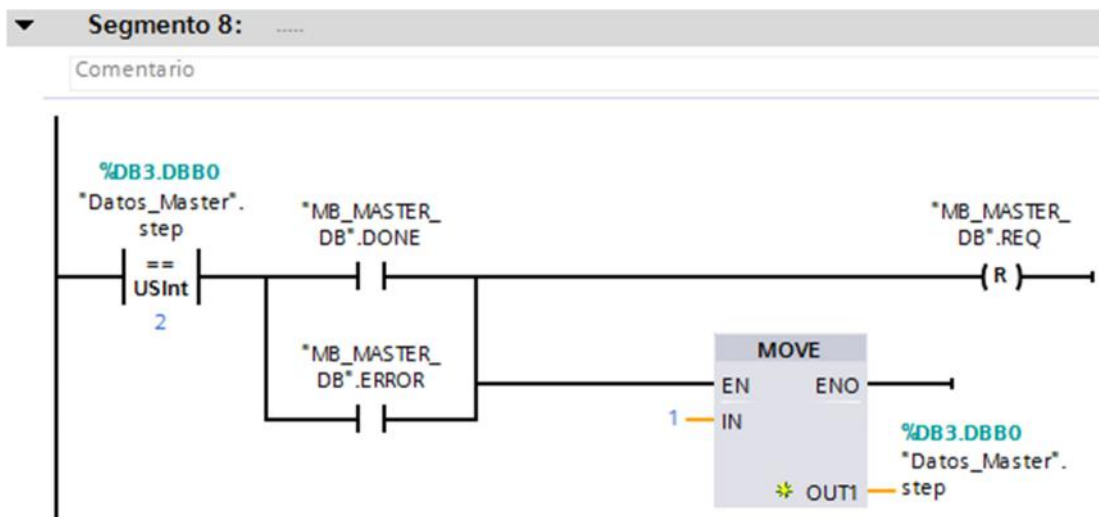


Figura 33. Cambio de paso del maestro

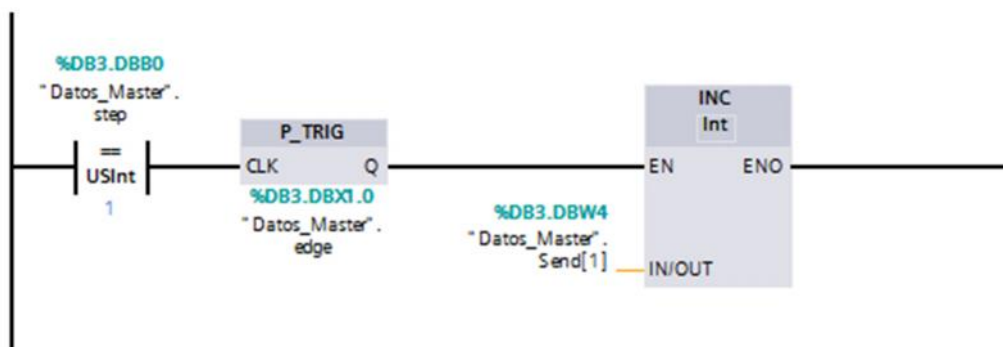
Fuente: (TIA Portal, 2013)

El programa complejo del maestro es el siguiente.



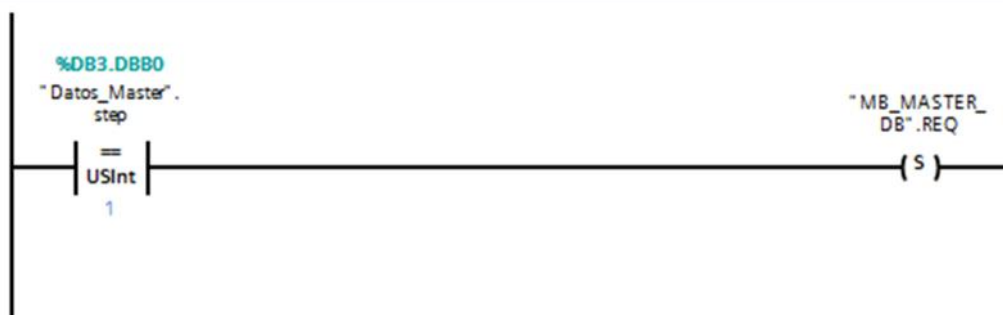
Segmento 3:

Comentario



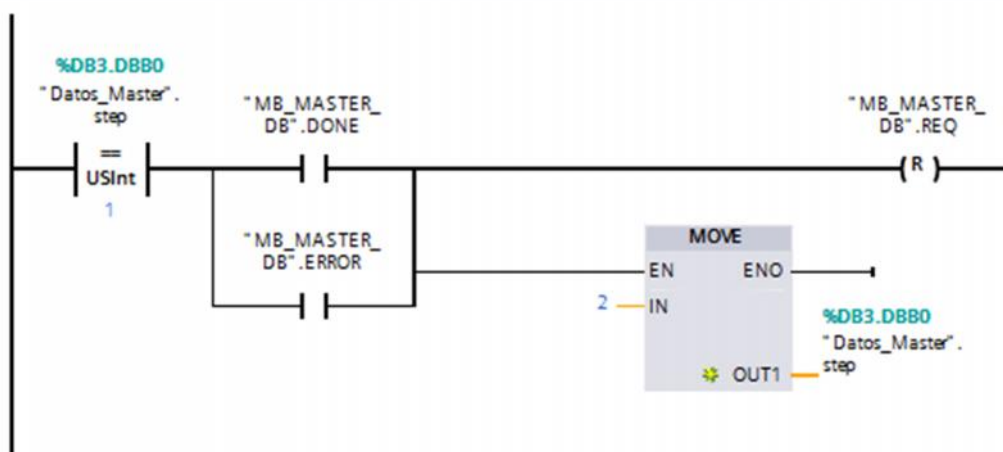
Segmento 4:

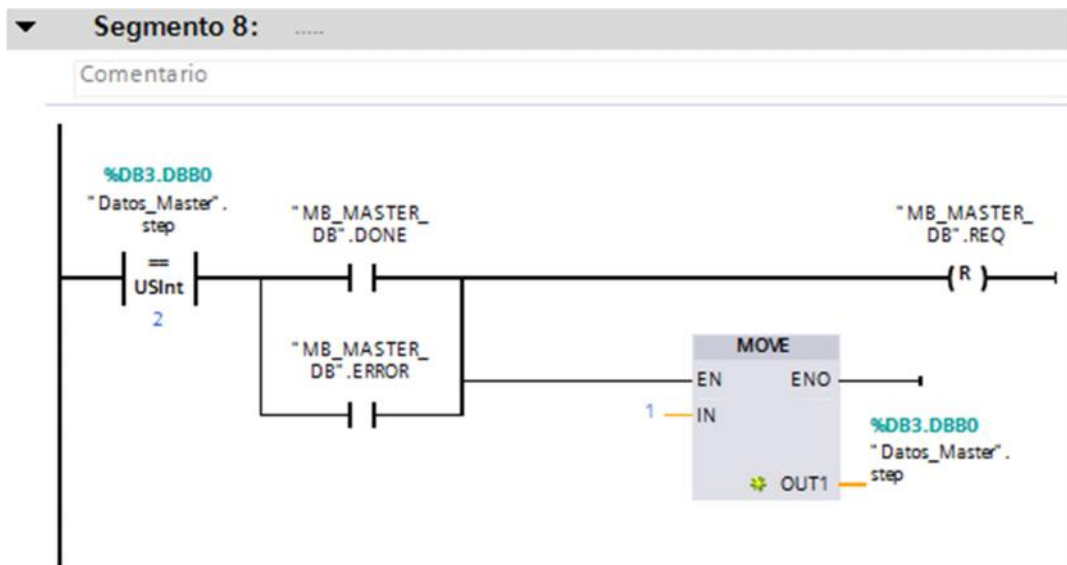
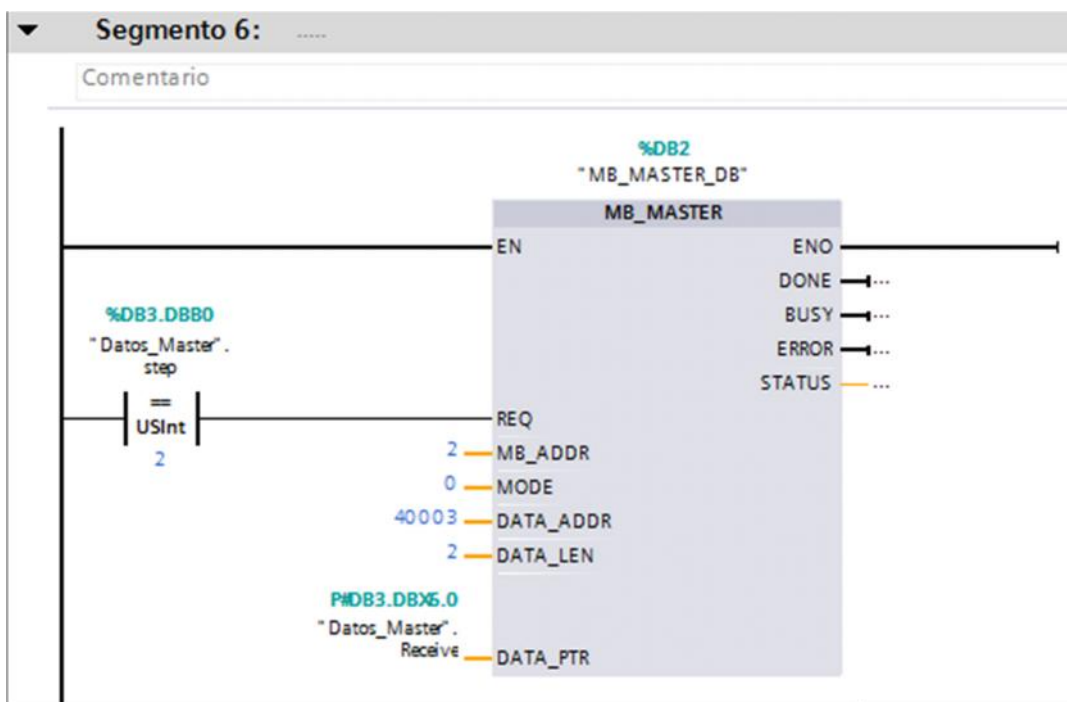
Comentario



Segmento 5:

Comentario





Programación del esclavo Modbus

El esclavo Modbus envía un valor entero 22222 por la dirección 40003 y el valor del contador dividido para 2 por la dirección 40004.

3.1.4 Creación del bloque de datos del esclavo

En el PLC esclavo se creó un bloque de datos globales llamados Datos_Slave desde la opción Agregar nuevo bloque del PLC maestro.

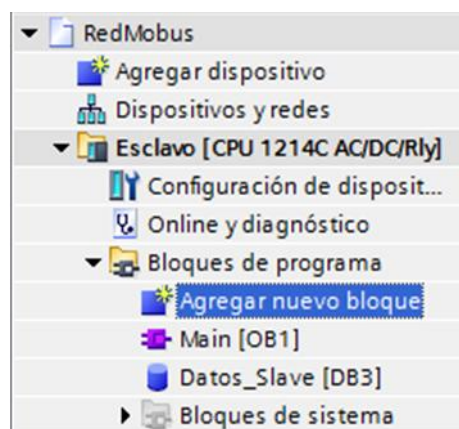


Figura 34. Creación bloque datos esclavo

Fuente: (TIA Portal, 2013)

Una vez creado el bloque de datos, ingresar a propiedades haciendo un clic derecho sobre el bloque, en la opción Atributos, desmarcar Acceso optimizado al bloque, pulse aceptar.

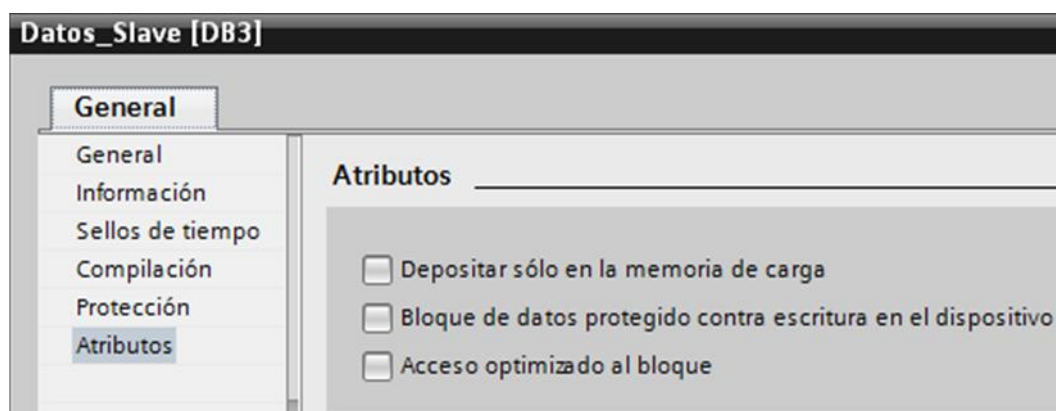


Figura 35. Deshabilitar Acceso optimizado al bloque esclavo

Fuente: (TIA Portal, 2013)

Crear el bloque de datos que se indica en la figura 36

Datos_Slave			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	MB_HOLD_REG	Struct	
3	Receive_0	Int	0
4	Receive_1	Int	0
5	Send_0	Int	22222
6	Send_1	Int	0

Figura 36. Bloque datos esclavo

Fuente: (TIA Portal, 2013)

El dato creado de tipo Struct llamado MB_HOLD_REG, el mismo que contiene Receive_0 que recibe el valor 11111 proveniente del maestro, Receive_1, recibe el valor del contador que envía el maestro, Send_0 envía el valor 22222 al maestro y Send_1 envía el valor del contador enviado por el maestro dividido para 2.

3.1.5 Configuración del puerto del esclavo

Mediante MB_COMM_LOAD, se configura el protocolo de comunicación Modbus, especificando el puerto, la velocidad de transmisión, la paridad y el bloque de datos del esclavo.

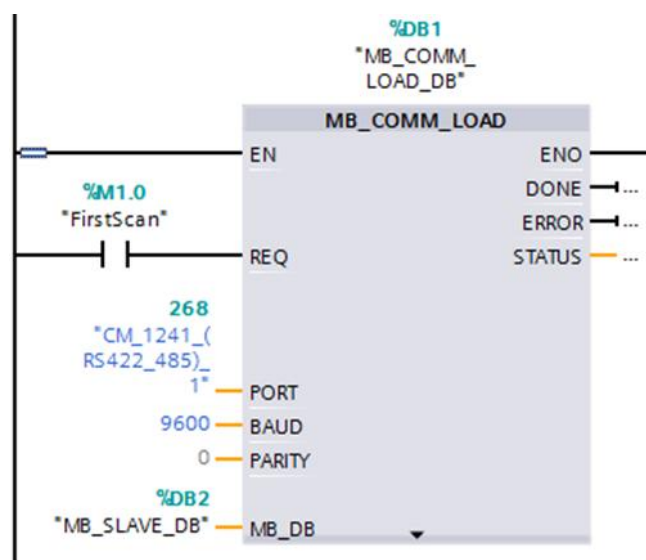


Figura 37. Protocolo de comunicación Modbus esclavo

Fuente: (TIA Portal, 2013)

El puerto es el 268, la velocidad de transmisión es de 9600, sin paridad y el bloque de datos DB2 que corresponde a la instrucción MB_SLAVE_DB.

La instrucción MB_COMM_LOAD_DB solo se ejecuta una vez, por lo que se utiliza el bit de marca de sistema FirstScan el mismo que se activa cuando el PLC realiza el primer escaneo.

3.1.6 Programación del esclavo

El esclavo intercambia los datos con el maestro mediante el siguiente segmento.

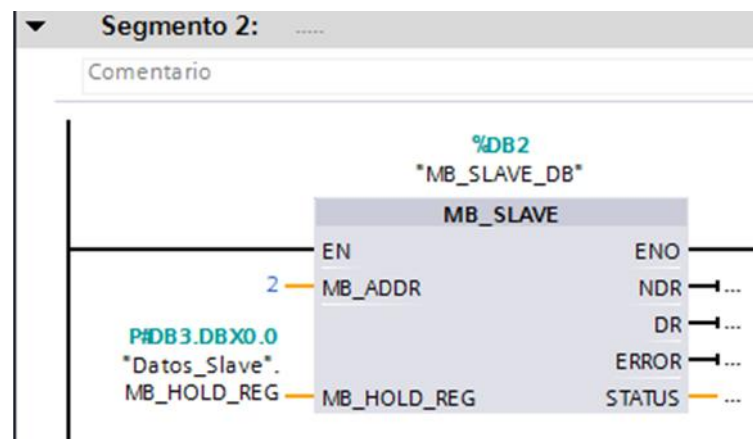


Figura 38. Intercambio de datos maestro esclavo

Fuente: (TIA Portal, 2013)

La instrucción MB_SLAVE_DB, realiza el intercambio de datos entre el esclavo y el maestro mediante el bloque de datos MB_HOLD_REG. En la entrada MB_ADDR se especifica la dirección Modbus del esclavo.

En el siguiente segmento se recibe el valor del contador enviado por el maestro en Receive_1, el mismo que es dividido para 2 y enviado el resultado al maestro mediante Send_1.

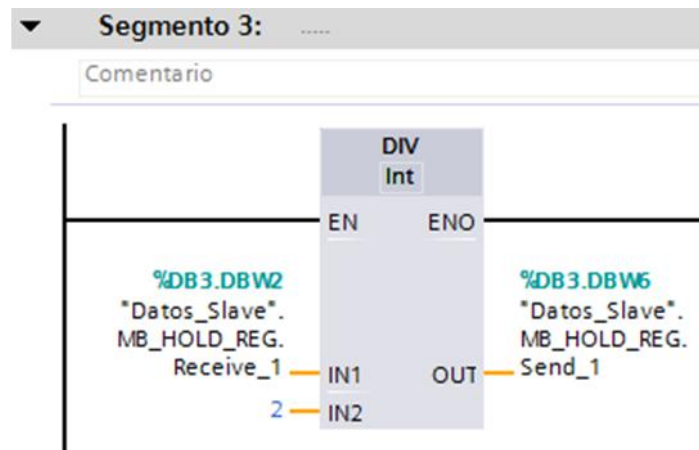
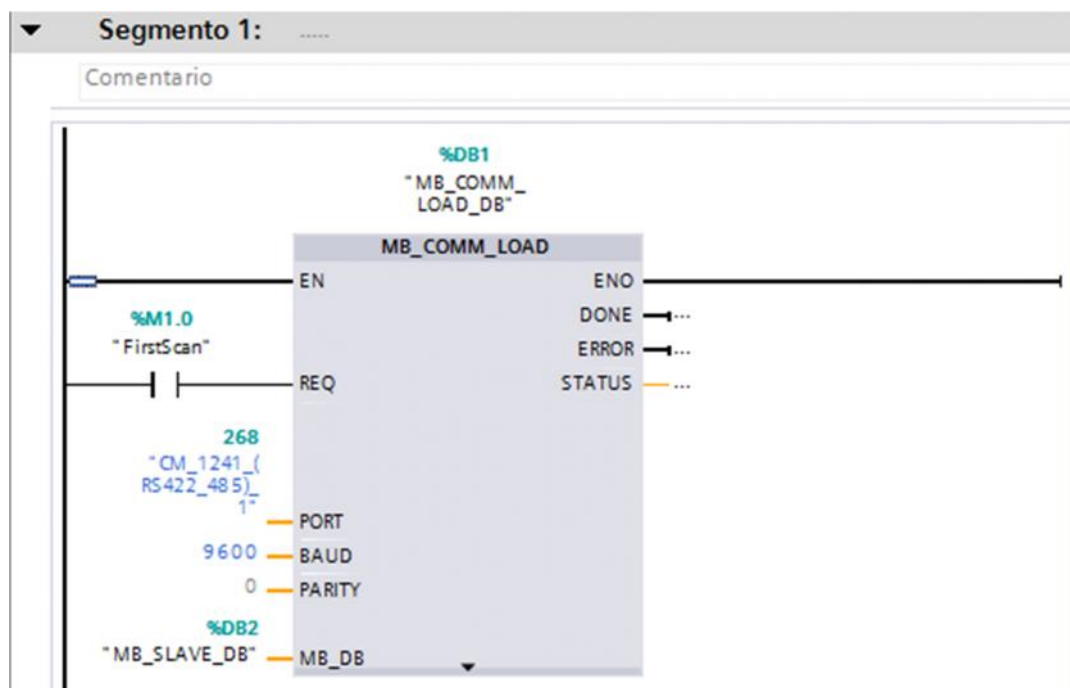
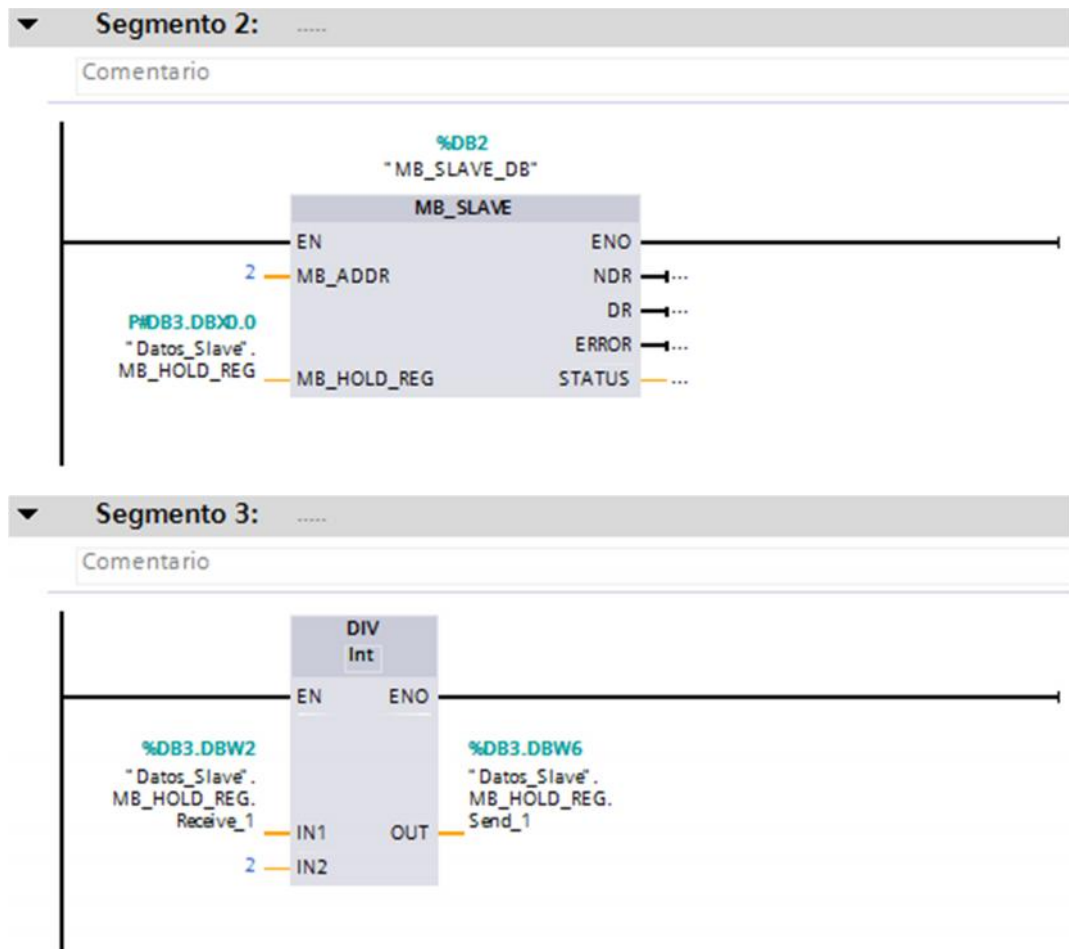


Figura 39. Intercambio de datos del contador

Fuente: (TIA Portal, 2013)

El programa final del esclavo es el siguiente.





Pruebas de funcionamiento

Una vez cargados los programas en los PLCs, para comprobar el funcionamiento del ejemplo descrito, se estableció la conexión en línea y se activó la opción Observar todo en el bloque de datos del maestro, cuyos resultados de muestran en la figura 40.

Nombre	Tipo de datos	Valor de arranq...	Valor de observación	Remanen...	Accesible d...
Static					
step	LSInt	1	1		
edge	Bool	false	TRUE		
Send	Array [D..1] ...				
Send[0]	Int	11111	11111		
Send[1]	Int	0	28081		
Receive	Array [D..1] of Int				
Receive[0]	Int	0	22222		
Receive[1]	Int	0	14039		

Figura 40. Pruebas funcionales comunicación Modbus

Fuente: (TIA Portal, 2013)

En Send[0] se observa el valor 11111 que envía el maestro, en Send[1] el valor del contador en este caso 28081, Receive[0] muestra el valor 22222 enviado por el maestro y en Receive[1] está el valor del contador dividido para 2.

Comunicación maestro/esclavo práctica 2

En base a la comunicación implementada, se realiza una modificación en el segmento 3 del programa del maestro, en lugar de enviar el valor del contador, se envía una constante 27648 al esclavo.

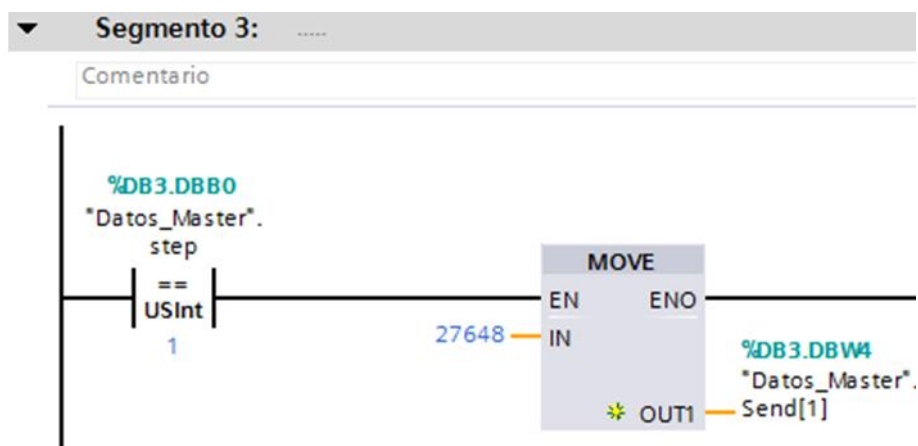


Figura 41. Envío de una constante al esclavo

Fuente: (TIA Portal, 2013)

Como ya no se utiliza el dato edge, el bloque de datos del maestro es el siguiente.

Datos_Master			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	step	USInt	1
3	Send	Array [0..1] of Int	
4	Send[0]	Int	11111
5	Send[1]	Int	0
6	Receive	Array [0..1] of Int	
7	Receive[0]	Int	0
8	Receive[1]	Int	0

Figura 42. Bloque datos práctica 1

Fuente: (TIA Portal, 2013)

En este ejemplo solo se modifica el programa del maestro en el segmento 3, el del esclavo permanece.

Al cargar el programa en el maestro y enviar a línea para visualizar los datos, se obtuvieron los siguientes resultados.

Datos_Master						
	Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación	Remanen...
1	Static					
2	step	USInt	0.0	1	1	
3	Send	Array [0..1] of Int	2.0			
4	Send[0]	Int	0.0	11111	11111	
5	Send[1]	Int	2.0	0	27648	
6	Receive	Array [0..1] of Int	6.0			
7	Receive[0]	Int	0.0	0	22222	
8	Receive[1]	Int	2.0	0	11111	

Figura 43. Resultados práctica 1

Fuente: (TIA Portal, 2013)

Comunicación maestro/esclavo práctica 2

A partir del practica 1, en el segmento 3 del maestro se modifica para que cuando se pulse la entrada I0.0, se envíe el valor 1 al esclavo caso contrario se envíe el valor 0.

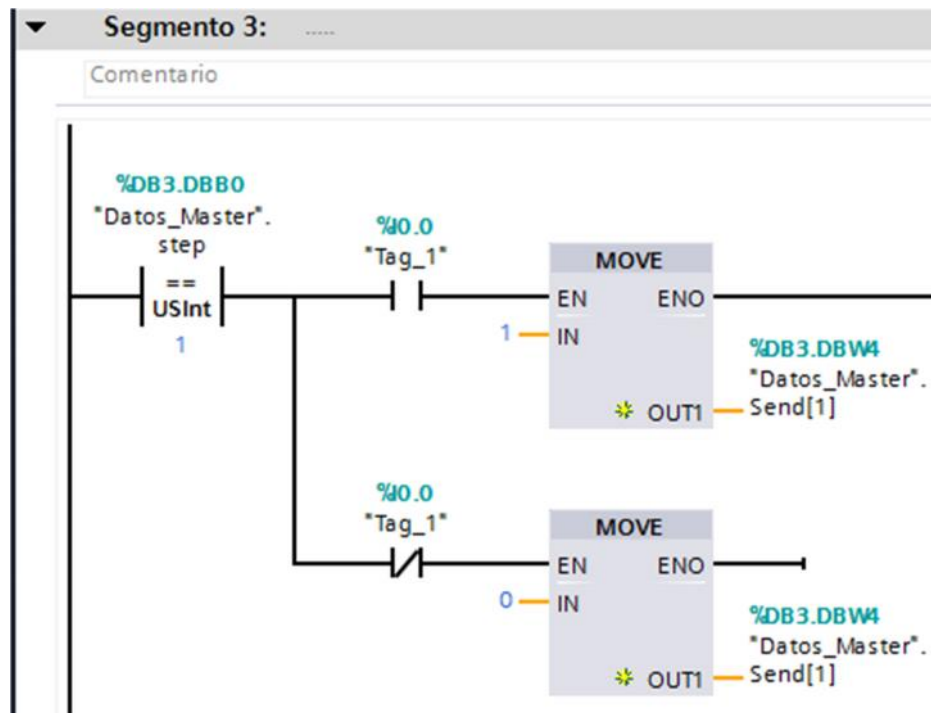


Figura 44. Segmento 3 del maestro práctica 2

Fuente: (TIA Portal, 2013)

En el segmento 3 del esclavo se mueve el dato recibido al byte de salidas digitales.

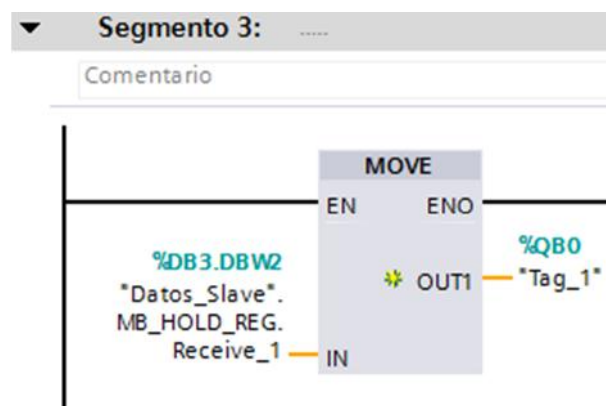


Figura 45. Segmento 3 del esclavo práctica 2

Fuente: (TIA Portal, 2013)

Cargar los programas en el maestro y el esclavo, como resultado cuando se active la entrada digital cero del maestro, se activara la salida digital cero del esclavo.

Una vez comprobado el funcionamiento, se agrega un segmento al final del programa del maestro, para mover el dato que envía el esclavo a la salida analógica cero del módulo de señales SM1232 AQ, que corresponde a la palabra QW96.

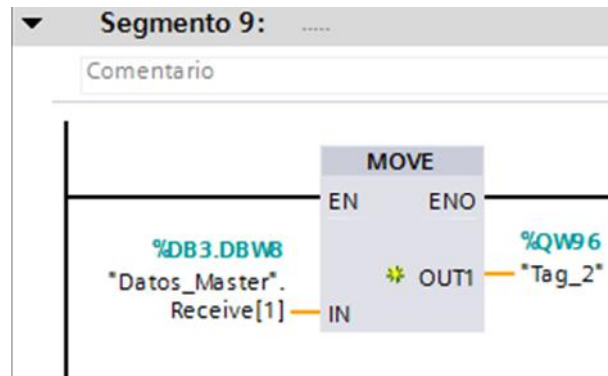


Figura 46. Segmento 9 del maestro práctica 2

Fuente: (TIA Portal, 2013)

Por lo tanto es necesario agregar en el maestro el módulo de señales SM 1232 AQ.

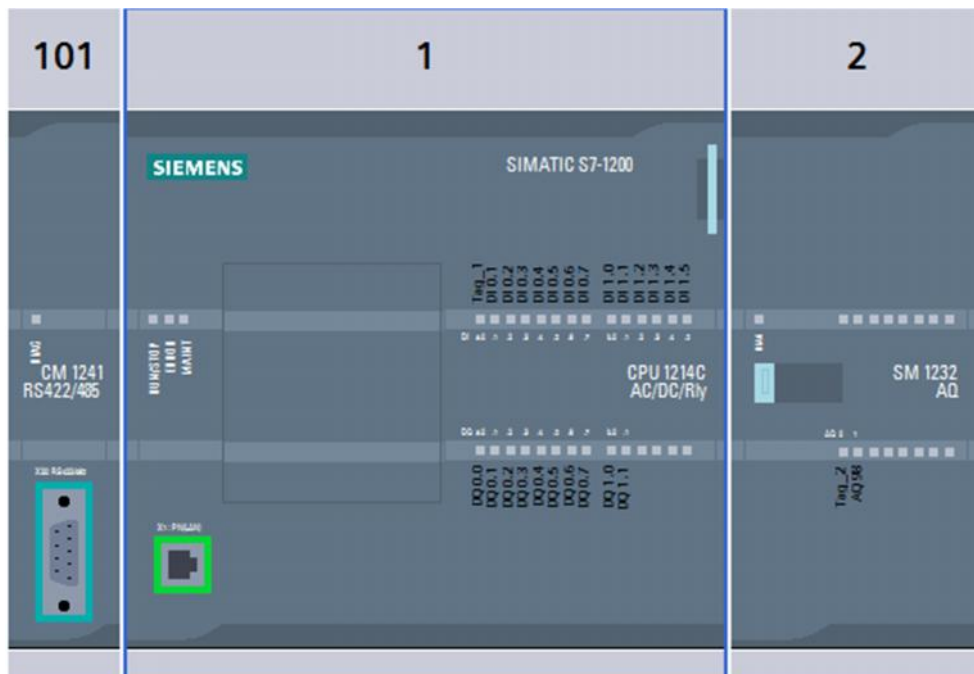


Figura 47. Conexión física maestro práctica 2

Fuente: (TIA Portal, 2013)

En el programa del esclavo se agrega 2 segmentos, el segmento 4 cuando se activa la salida digital cero, se envía al maestro el valor de 13824 caso contrario en el segmento 5 se envía 0.

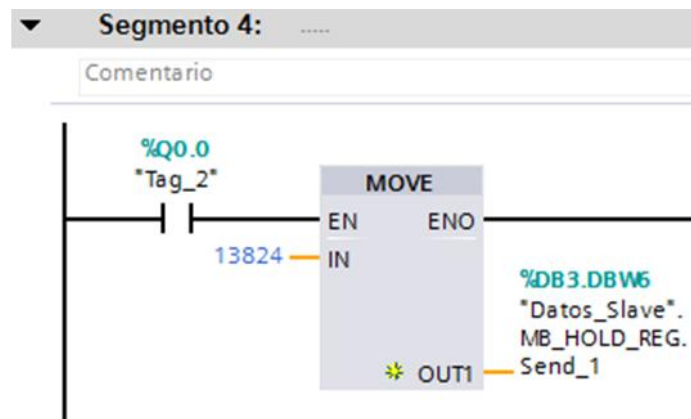


Figura 48. Segmento 4 del esclavo práctica 2

Fuente: (TIA Portal, 2013)

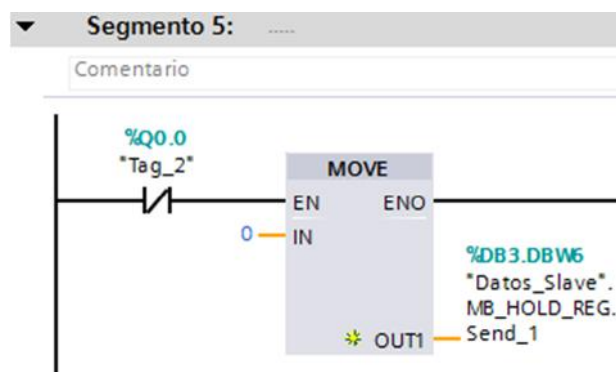


Figura 49. Segmento 5 del esclavo práctica 2

Fuente: (TIA Portal, 2013)

Cargados los programas en los PLCs se comprobó en el maestro la recepción de datos.

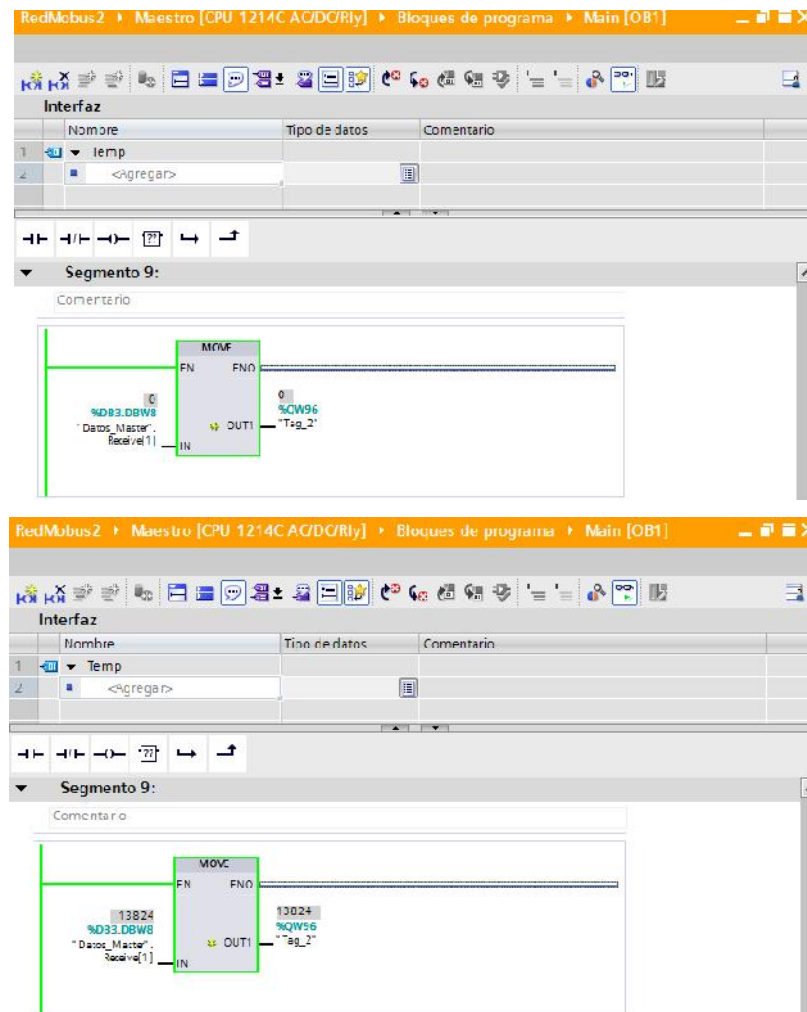


Figura 50. Prueba funcional práctica 2

Fuente: (TIA Portal, 2013)

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Se implementó la comunicación Modbus RTU entre dos PLCs S7 1200 CPU 1214C con la ayuda de los módulos de comunicación CM 1241 RS422/RS485.
- Se realiza la comunicación en dos pasos mediante el PLC maestro, en el primer paso envía datos al esclavo y en el segundo paso recibe los datos enviados por el maestro.
- En el PLC maestro y el PLC esclavo se deben activar los bytes de marca del sistema, necesarios para configurar la comunicación Modbus RTU.
- La comunicación implementada emplea el protocolo RS485 Half Duplex a dos hilos, por esta razón se utiliza el cable Profibus.
- Para el intercambio de datos se utilizó las direcciones 40001 hasta la 40004 que corresponden a los Holdign Registers.

Recomendaciones

- No conectar los módulos de comunicación a ninguna alimentación externa por que su alimentación es a través del bus del PLC.
- Para realizar la comunicación Modbus RTU con los módulos utilizados en el proyecto se debe utilizar TIA Portal versión 12.
- Realizar más trabajos de graduación referente al tema de comunicación Modbus para incrementar el material didáctico.

GLOSARIO DE TÉRMINOS.

LAN: Local Area Network (por sus siglas en inglés); Red de Área Local (en español)

CSMA/CD: Carrier Sense Multiple Access / Collision Detect (por sus siglas en inglés) - Acceso múltiple con detección de portadora y detección de colisiones (en español)

Mbps: Mega bytes por segundo

Gbps: Giga bytes por segundo

Conector BNC: Bayonet Neil-Concelman, o a veces British Naval Connector; se usa para la interconexión de equipos y/o dispositivos en redes locales 10BASE2 Ethernet con cable coaxial

UTP: Unshielded Twisted Pair; Par trenzado sin apantallar

MAC: Media Access Control [Control de acceso al medio]

protocolo IP: Internet Protocol [Protocolo de Internet]

REFERENCIA BIBLIOGRAFÍA.

Barragan , A. (10 de Junio de 2013). Modbus.

Cevallos, W. (Noviembre de 2007). Protocolo Modbus. Quito.

National Instruments. (2012). *Comunicación Serial*. Obtenido de <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>

Siemens. (11 de 2009). Controlador Programable S7 1200.

Siemens. (Enero de 2015). Hoja de datos 6ES7241-1CH32-0XB0.

TIA Portal, V. (2013). Software.

ANEXOS