



UNIVERSIDAD POLITÉCNICA SALESIANA

CARRERA DE INGENIERÍA ELECTRÓNICA

**PROYECTO DE TITULACIÓN PREVIO A LA
OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICO**

TEMA:

**“IMPLEMENTACIÓN DE UNA RED MODBUS TCP
WIRELESS CON INTEGRACIÓN DE TECNOLOGÍA
SIEMENS Y SCHNEIDER PARA PRÁCTICAS
ESTUDIANTILES DEL LABORATORIO DE
AUTOMATIZACIÓN INDUSTRIAL”**

AUTORES

**CARLOS ANDRÉS CASTRO SILVA
JOSIE ESTEBAN ALVARADO FEIJOO**

DIRECTOR

**ING. CÉSAR CÁCERES GALÁN
GUAYAQUIL, ABRIL DEL 2017**

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Carlos Andrés Castro Silva, portador de la cedula de identidad N° 0930670252 y Josie Esteban Alvarado Feijoo, portador de la cedula de identidad N° 093033080, estudiantes de la Universidad Politécnica Salesiana con sede Guayaquil, declaramos que la responsabilidad del contenido de este proyecto de titulación corresponde exclusivamente y es propiedad intelectual de la Universidad Politécnica Salesiana.

Guayaquil, Abril 2017.

Carlos Castro Silva

CI: 0930670252

Josie Alvarado Feijoo

CI: 0930330980

**CERTIFICADOS DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Nosotros, CARLOS ANDRÉS CASTRO SILVA y JOSIE ESTEBAN ALVARADO FEIJOO autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de Los Autores.

Guayaquil, Abril del 2017

Carlos Castro Silva
CI: 0930670252

Josie Alvarado Feijoo
CI: 0930330980

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES

Nosotros, Carlos Andrés Castro Silva con documento de identificación N° 0930670252 y Josie Esteban Alvarado Feijoo con documento de identificación N° 093033080, manifestamos nuestra voluntad y ceder a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación titulado “IMPLEMENTACIÓN DE UNA RED MODBUS TCP WIRELESS CON INTEGRACIÓN DE TECNOLOGÍA SIEMENS Y SCHNEIDER PARA PRÁCTICAS ESTUDIANTILES DEL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL “, mismo que ha sido desarrollado para optar por el título de “INGENIERO ELECTRÓNICO”, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hagamos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Carlos Castro Silva
CI: 0930670252

Josie Alvarado Feijoo
CI: 0930330980

Guayaquil, Abril del 2017

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

SUSCRITO POR EL TUTOR

Yo Ing. Cesar Cáceres Galán declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación:

“IMPLEMENTACIÓN DE UNA RED MODBUS SP WIRELESS CON INTEGRACIÓN DE TECNOLOGÍA SIEMENS & SCHNEIDER PARA PRÁCTICAS ESTUDIANTILES DEL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL”.

Con resolución de aprobación de Consejo de Carrera **Nº RESOLUCION** realizado por los estudiantes:

CASTRO SILVA CARLOS ANDRÉS cédula de ciudadanía: 0930670252

ALVARADO FEJOO JOSIE ESTEBAN cédula de ciudadanía: 0930330980

Obteniendo un producto que cumple con los objetivos del diseño de aprobación, informe final y demás requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Guayaquil, 15 de febrero del 2017

Tutor del Trabajo de titulación
Docente: Ing. César Cáceres Galán
C.I. 0911477776

DEDICATORIA

Este trabajo va dedicado a mi familia, mis padres quienes con esfuerzo y perseverancia me han acompañado y apoyado durante todo este tiempo a formarme como un profesional en el campo y en el aula, brindándome los recursos necesarios para culminar mi proyecto de estudio y aportar a la sociedad y el país mi desempeño y conocimiento adquirido durante mi tiempo de formación académica.

Agradecer a mi padre, quien con mucha sabiduría, paciencia y sacrificio me ha brindado y enseñado lo necesario para desenvolverme en el área productiva, por estar siempre a mi lado dándome el soporte moral para superarme diariamente y demostrándome siempre con hechos lo valioso que es pertenecer a esta familia.

Dedicado a mi hermosa hija Maia Arlette Castro Medina, quien se convirtió en mi mayor motivo para culminar esta etapa de mi vida y alcanzar este logro para ser un ejemplo para ella.

Carlos Castro Silva

DEDICATORIA

Este proyecto va dedicado a Dios y a todos mis familiares quienes han sido los que me han acompañado y me han apoyado en cada momento de mi vida, a mi madre quien ha sido el pilar fundamental en este largo camino de preparación para la obtención de un título profesional, siempre junto a mí para enseñarme que con dedicación y esfuerzo se puede lograr todas las cosas que nos proponemos.

A mi padre quien gracias a su conocimiento y su experiencia me guio en cada ciclo de mi etapa como estudiante de él destaco el esfuerzo que siempre realizó para que nunca nos falte nada en el hogar.

Josie Alvarado Feijoo

RESUMEN DEL PROYECTO

AÑO	ALUMNOS	DIRECTOR DE PROYECTO	TEMA DE PROYECTO DE TITULACIÓN
2017	CASTRO SILVA CARLOS ÁNDRES JOSIE ESTEBAN ALVARADO FEIJOO	ING. CÉSAR CÁCERES GALÁN	IMPLEMENTACIÓN DE UNA RED MODBUS – TCP WIRELESS CON INTEGRACIÓN DE TECNOLOGÍA SIEMENS Y SCHNEIDER PARA PRÁCTICAS ESTUDIANTILES DEL LABORATORIO DE AUTOMATIZACIÓN INDUSTRIAL

La comunicación TCP permiten una mejor gestión y control de envío y recepción de datos, esto ha hecho que este protocolo sea ampliamente utilizado a nivel industrial ya que con el pasar del tiempo los dispositivos de control y campo conectados en los procesos industriales se van devaluando tecnológicamente como son en el caso de dispositivos en comunicación Modbus. La red que se implementa es MODBUS – TCP en la cual se integran diferentes tecnologías de PLC's de distintos fabricantes, esto con el fin de demostrar que mantener comunicación y control entre equipos de diferentes protocolos. La red está diseñada para enviar y recibir datos de control y monitoreo entre las estaciones maestro – esclavo, esta se implementó con estaciones de trabajo ya existentes en el laboratorio de automatización industrial y fabricación flexible de la Universidad Politécnica Salesiana sede Guayaquil, de las cuales se añadió el modulo correspondiente para la conversión de protocolos del presente proyecto, la estación que trabaja como maestro será una maleta didáctica SIEMENS S7 – 1200, el cual se encarga de enviar y recibir señales de escritura/lectura a las estaciones esclavos, los cuales son cuatro maletas didácticas TWIDO de SCHNEIDER conectados a un GATEWAY MODBUS – TCP.

ABSTRAC

YEAR	STUDENTS	ADVISOR	TITLE
2017	CASTRO SILVA CARLOS ANDRES JOSIE ESTEBAN ALVARADO FEIJOO	ING. CÉSAR CÁCERES GALÁN	IMPLEMENTATION OF A MODBUS - TCP WIRELESS NETWORK WITH SIEMENS AND SCHNEIDER TECHNOLOGY INTEGRATION FOR STUDENT PRACTICES OF THE INDUSTRIAL AUTOMATION LABORATORY

TCP communication allows a better management and control of sending and receiving data, this has made this protocol widely used at industrial level since over time the control devices and field connected in industrial processes are devalued technologically as they are in the case of devices in Modbus communication. The network that is implemented is MODBUS - TCP in which are integrated different technologies of PLC 's of different manufacturers, this in order to demonstrate that to maintain communication and control between teams of different protocols. The network is designed to send and receive control and monitoring data between the master - slave stations, which was implemented with existing workstations in the industrial automation and flexible manufacturing laboratory of the Salesian Polytechnic University of Guayaquil. Added the corresponding module for the protocol conversion of the present project, the station that works as master will be a SIEMENS S7 - 1200 didactic suitcase, which is responsible for sending and receiving write / read signals to the slave stations, which are four SCHNEIDER TWIDO teaching kits connected to a MODBUS - TCP GATEWAY.

ÍNDICE GENERAL

1.	El Problema.....	2
1.1	Planteamiento del problema.....	2
1.2	Delimitación.....	3
1.2.1.	Temporal	3
1.2.2.	Espacial	3
1.2.3.	Académica	3
1.3	Objetivos	4
1.3.1.	Objetivo General	4
1.3.2.	Objetivos Específicos	4
1.4	Justificación.....	5
1.5	Descripción De La Propuesta.....	5
1.6	Beneficiarios De La Propuesta e Intervención.....	6
2.	Marco Teórico.....	7
2.1.	Fundamentos De Comunicaciones Aplicadas En Este Proyecto	7
2.1.1.	Modelo OSI De Comunicación	7
2.1.2.	Fundamentos De Comunicaciones De Campo.....	9
2.1.3.	Fundamentos De Comunicaciones Sobre IP	10
2.1.4.	Ventajas de comunicaciones de campo	11
2.1.5.	Ventajas de comunicación sobre IP.....	12
2.1.6.	Topología de buses de comunicación industrial.....	13
2.2.	Protocolo de comunicación Modbus.....	15

2.2.1.	Descripción general del protocolo.....	15
2.2.2.	Mapa de registro	17
2.2.3.	Estructura de tablas	19
2.2.4.	Códigos de función.....	21
2.3.	Protocolo de comunicación Ethernet	26
2.3.1.	Descripción de funcionamiento del protocolo.....	26
2.3.2.	Descripción de trama de datos.....	29
2.4.	Controladores lógicos programables utilizados en el proyecto.....	30
2.4.1.	Descripción general de funcionamiento	31
2.4.2.	SIEMENS S7 – 1200.....	32
2.4.2.1.	Características del PLC.....	32
2.4.2.2.	Software de programación	34
2.4.2.3.	Librería Modbus TCP	35
2.4.3.	TWIDO TWDLCAA24DRF	38
2.4.3.1.	Características del PLC.....	38
2.4.3.2.	Software de programación	40
2.4.3.3.	Macros de comunicación	42
2.5.	Red de comunicación Modbus – TCP.....	44
2.5.1.	Encapsulamiento de trama Modbus sobre TCP.....	45
2.5.2.	Ventajas de comunicación Modbus – TCP	47
3.	Marco metodológico	48
3.1.	Descripción de hardware de maletas didácticas con PLC TWIDO	48

3.2.	Descripción del hardware de maleta didáctica con S7 – 1200.....	49
3.3.	Descripción de repartidor Modbus LU9GC3.....	50
3.4.	Descripción de Gateway y equipos de comunicación inalámbrica.....	51
3.5.	Diseño y ensamble de modulo convertidor Modbus TCP	52
3.5.1.	Diagrama de conexión de componentes del módulo.....	52
3.5.2.	Estructura de carcasa y montaje de equipos.....	52
3.5.3.	Cableado de Gateway, Router y fuente de alimentación.....	54
4.	Guía De Prácticas De Laboratorio	56
4.1.	Práctica 1	56
4.2.	Práctica 2.....	61
4.3.	Práctica 3.....	67
4.4.	Práctica 4.....	71
4.5.	Práctica 5.....	79
4.6.	Práctica 6.....	85
4.7.	Práctica 7.....	89
4.8.	Práctica 8.....	95
4.9.	Práctica 9.....	101
4.10.	Práctica 10.....	112
5.	Análisis Y Resultados	122
5.1	Resultados Obtenidos.....	122
5.2	Análisis de Resultados	125
	CONCLUSIONES	127

RECOMENDACIONES	128
CRONOGRAMA	129
PRESUPUESTO	130
BIBLIOGRAFÍA	131
ANEXOS	133
ANEXO 1.....	133
ANEXO 2.....	136
ANEXO 3.....	139

ÍNDICE DE FIGURAS

Figura 1. Capas del modelo OSI	8
Figura 2. Clasificación de diversos buses de campo.....	10
Figura 3. Pirámide de Comunicación Industrial	10
Figura 4. Cableado de red: a) Cableado convencional; b) Cableado con red industrial	12
Figura 5. Conexión Bus	13
Figura 6. Conexión anillo.....	13
Figura 7. Conexión estrella	14
Figura 8. Conexión árbol.....	14
Figura 9. Conexión RS – 485 de dispositivos	17
Figura 10. Mapa de Registro Modbus.....	17
Figura 11. Modelo TCP/IP y algunos de sus protocolos.....	27
Figura 12. Cable RJ45 para conexión TCP/IP	29
Figura 13. Trama Ethernet	29
Figura 14. Arquitectura del PLC	31
Figura 15. Vista de partes de CPU S7 – 1200.....	32
Figura 16. Vista del Portal y Vista del Proyecto	34
Figura 17. Partes de un Controlador Compacto Twido	39
Figura 18. Ejemplo de conexión remota	40
Figura 19. Plataforma TwidoSuite	41
Figura 20. Esquema encapsulado de Modbus TCP.....	45
Figura 21. Esquema de maletas didácticas SCHNEIDER	48
Figura 22. Diagrama general de conexiones de CPU 1214 AC/DC/RLY	49
Figura 23. Características HMI Siemens KTP600.....	50

Figura 24. Esquema interno de conexión de repartidor	51
Figura 25. Diagrama de conexión de equipos.....	52
Figura 26. Caja metálica para modulo convertidor Modbus TCP	53
Figura 27. Conexión de equipos de módulo Modbus TCP	54
Figura 28. Montaje de equipos del módulo convertidor Modbus TCP.....	122
Figura 29. Conexión de los equipos de comunicación y ajuste de cables.....	123
Figura 30. Conexión de los dispositivos esclavos en el HUB Modbus	123
Figura 31. Conexión de 3 maletas Modbus – Práctica 9.....	124
Figura 32. Conexión de una maleta didáctica Modbus – Práctica 1	124
Figura 33. Paso 1 para seleccionar Macros.....	133
Figura 34. Paso 2 para seleccionar Macros.....	134
Figura 35. Paso 3 Selección de Puertos y Número de esclavo.....	134
Figura 36. Paso 4 Selección de función Macro.....	135
Figura 37. Página de configuración Gateway	137
Figura 38. Página de configuración LAN Router (Autores, 2016).....	137
Figura 39. Página de configuración WLAN Router (Autores, 2016).....	138
Figura 40. Direccionamiento RTU del bloque MB_CLIENT de SIEMENS.....	140

ÍNDICE DE TABLAS

Tabla 1. Estructura de tabla de datos	19
Tabla 2. Principales códigos de funciones	21
Tabla 3. Tabla de peticiones 01 y 02.....	21
Tabla 4. Tabla de peticiones 03 y 04.....	22
Tabla 5. Tabla de peticiones 05.....	23
Tabla 6. Tabla de peticiones 06.....	24
Tabla 7. Tabla de peticiones 15.....	24
Tabla 8. Tabla de peticiones 16.....	25
Tabla 9. Especificaciones de CPU's S7 – 1200	33
Tabla 10. Bloque Instrucción MB_CLIENT	35
Tabla 11. Tipo de datos de parametrización	36
Tabla 12. Bloque de Instrucción MB_SERVER.....	37
Tabla 13. Tipo de datos de parametrización	37
Tabla 14. Características de controlador TWIDO.....	38
Tabla 15. Macro de lectura de un bit.....	42
Tabla 16. Macro de lectura de una palabra	43
Tabla 17. Macro de escritura de un bit.....	43
Tabla 18. Macro de escritura de una palabra	43
Tabla 19. Macro de lectura de N palabras.....	44
Tabla 20. Macro de escritura de N palabras.....	44
Tabla 21. Estructura del prefijo Modbus/TCP	46
Tabla 22. Estructura de mensajes en Modbus/TCP	46
Tabla 23. Configuraciones iniciales de ETG100	136

INTRODUCCIÓN

La red industrial Ethernet ha generado el deseo de expandir este protocolo de comunicación a su aplicación en procesos de planta. Ethernet se ha implementado en el mercado de los sistemas de control de procesos y automatización para una mejor interconexión a nivel de campo de sensores y actuadores, de esta forma se realiza una mejor gestión y control de flujo de datos en tiempo real.

En las aplicaciones industriales, Ethernet es usado en conjunto con la pila de protocolos TCP/IP universalmente aceptada, suministrando un mecanismo de transporte de datos confiables entre máquinas y permitiendo interoperabilidad entre diversas plataformas. El presente proyecto contempla la implementación de una red Modbus TCP/IP sobre Ethernet para control y supervisión del nivel de campo en donde nos enfocamos a demostrar lo que comúnmente se ve en los procesos industriales, lo cual es la integración de tecnología de distintos fabricantes.

La fácil implementación de esta red permite que cualquier dispositivo de campo, se comunique vía Ethernet sin que sea necesario un potente microprocesador o de mucha memoria. El Ethernet Modbus TCP tiene un protocolo muy sencillo y una velocidad de salida a 100 Mbps permitiendo aplicaciones de tiempo real. Dado que el protocolo es idéntico que en un Modbus serie, las tramas de mensajes se pueden direccionar de una red a otra sin tener que cambiar protocolos. (Schneider Electric, 2009).

1. El Problema

1.1 Planteamiento del problema

El desarrollo de diferentes proyectos de titulación que se han generado en la Universidad Politécnica Salesiana sede Guayaquil que incorporan comunicaciones industriales deben ser utilizados como prácticas de laboratorios, con el objetivo de que los estudiantes puedan analizar e implementar prácticas en dichos módulos aplicando protocolos de comunicación conocidos entre tecnología del mismo fabricante. Con los módulos disponibles en los laboratorios de la Universidad, los estudiantes no tienen la oportunidad de establecer una conexión entre equipos de distintos fabricantes ni transmitir información de un tipo de comunicación a otra, en caso de los módulos implementados con buses de campo (As-i y Modbus), y desarrollar lo que en la industrial se aplica comúnmente que es la integración de distintas tecnologías.

En el caso de las comunicaciones de campo, los protocolos que usan los equipos de comunicación industrial son los mismos pero la programación y diseño de la trama difiere según el fabricante del equipo y el software de programación.

Para el caso de comunicación entre tecnologías de PLC's SIEMENS y TWIDO que están disponibles en maletas didácticas de los laboratorios de la Universidad, se conoce que el protocolo de comunicación por defecto de TWIDO es Modbus (RTU y ASCII) y el protocolo por defecto de SIEMENS es PROFINET (Ethernet) en donde por medio de librerías se puede establecer con el mismo puerto RJ45 una comunicación Modbus TCP, es así que se puede enlazar estas distintas tecnologías a través de un GATEWAY.

1.2 Delimitación

1.2.1. Temporal

El presente proyecto se implementó en un tiempo estimado de 1 año y 6 meses en el periodo de 2015 – 2017.

1.2.2. Espacial

El trabajo de grado para la obtención del título de “Ingeniero Electrónico”, fue implementado para el laboratorio de “Sistemas de Control y SCADA” en el bloque B de la Universidad Politécnica Salesiana ubicada en Chambers #277 entre Laura Vicuña y Robles frente a Villa la Joya.

1.2.3. Académica

El proyecto será aplicado únicamente a una estación maestro correspondiente a un PLC S7 – 1200 de la marca SIEMENS, mientras que los esclavos serán 4 estaciones que utilizan PLC TWIDO de SCHNEIDER con módulos perteneciente a la Universidad Politécnica Salesiana. Las diversas prácticas que se desarrollaran en la red serán 10 y harán uso del sistema HMI del PLC SIEMENS de la estación maestro S7 – 1200, mientras que las estaciones TWIDO que funcionaran como esclavos, harán uso de variadores de frecuencias, motores y simulación de señales de sensores y salidas de actuadores por medio de su panel frontal. Los datos de comunicación entre maestro y los esclavos se enviaran inalámbricamente. El proyecto descrito en este documento pretende la enseñanza y aplicación de la librería de bloques de comunicación Modbus TCP Cliente/Servidor de SIEMENS y una breve descripción de bloques de funciones “Macros de Comunicación” para los PLC’s TWIDO, con el fin que maestros y/o

estudiantes puedan desarrollar o dar continuidad a diversas aplicaciones basadas en comunicación Modbus TCP.

1.3 Objetivos

1.3.1. Objetivo General

Diseñar la primera plataforma de comunicación Modbus - TCP inalámbrica entre PLC's de distintos fabricantes como son SIEMENS y SCHNEIDER para prácticas de comunicación industrial de control y SCADA de la Universidad Politécnica Salesiana sede Guayaquil y brindar la oportunidad a los estudiantes de desarrollarse en este tipo de comunicación con ayuda de tecnología de conversión de protocolos.

1.3.2. Objetivos Específicos

- Diseñar una red industrial con PLC de distintas marcas como son SIEMENS y SCHNEIDER.
- Implementar el protocolo de comunicación MODBUS – TCP entre el PLC maestro y 4 estaciones de trabajo esclavos.
- Transmitir datos a las estaciones correspondientes mediante comunicación WIRELESS.
- Desarrollar 10 prácticas estudiantiles, de procesos industriales para los estudiantes del laboratorio de automatización industrial que ocuparan estos módulos PLC's.
- Incentivar el desarrollo de proyectos en la misma plataforma de comunicación con un PLC Twido como maestro.

1.4 Justificación

Actualmente en la Universidad Politécnica Salesiana sede Guayaquil, se han estado implementando módulos de prácticas con diferentes protocolos de comunicación industrial, debido a esto el presente proyecto se desarrolla con el fin de establecer una comunicación de campo por medio un trama TCP, por lo que se pretende complementar los módulos disponibles con red MODBUS de las estaciones esclavos SCHNEIDER para que se conecten a un GATEWAY MODBUS/TCP y los datos sean transmitido a la estación maestro SIEMENS.

El protocolo TCP/IP se ha convertido en un estándar de comunicación en la actualidad, por lo que a nivel industrial se ha comenzado a implementar en las diversas comunicaciones de campo ya que esto permite monitorear sistemas y controlar en tiempo real desde cualquier computador con una conexión a internet, así como la integración de tecnología de distintos fabricantes que se emplea habitualmente a nivel industrial, estos deben mantener una comunicación en común para que puedan intercambiar datos, por lo que necesariamente se hace uso de pasarelas de comunicación.

1.5 Descripción De La Propuesta

En la Universidad Politécnica Salesiana sede Guayaquil, se implementa una red Modbus TCP en la cual se integran diferentes tecnologías de PLC's de distintos fabricantes. La red puede enviar y recibir datos de control y monitoreo entre las estaciones maestro – esclavo. La red se implementó con estaciones de trabajo ya existentes en el laboratorio de automatización industrial y fabricación flexible de la

Universidad Politécnica Salesiana, por lo que a estas maletas se les añadirá los respectivos equipos para poder establecer conexión inalámbrica.

La estación que trabaja como maestro usa un PLC SIEMENS S7 – 1200, el cual es el encargado de realizar peticiones y enviar señales de escritura digitales y analógicas a los equipos que trabajaran como esclavos que son los 4 maletas didácticas que incorporan PLC TWIDO y VDF de SCHNEIDER, los mismo que están conectados al GATEWAY, para la respectiva conversión de comunicación.

1.6 Beneficiarios De La Propuesta e Intervención

- Los estudiantes de la Universidad Politécnica Salesiana sede Guayaquil, serán los mayores beneficiarios, ya que tendrán la oportunidad de desarrollar y aprender acerca de este protocolo por medio de las prácticas descritas en el capítulo 4.

- Los profesores, que podrán practicar y mejorar este proyecto por medio de la configuración del Gateway para hacerlo funcionar como esclavo TCP según la aplicación que se deseen simular.

2. Marco Teórico

2.1. Fundamentos De Comunicaciones Aplicadas En Este Proyecto

Según la Universidad de Oviedo (2006) menciona que la comunicación en las plantas industriales se ha hecho imprescindible en los procesos industriales modernos. Muchos sistemas están conformados por equipos de diferentes fabricantes y funcionan en diferentes niveles de automatización, pese a que puedan estar distanciados entre sí, a menudo se desea que trabajen de forma coordinada para un resultado satisfactorio del proceso.

El objetivo principal es la comunicación totalmente integrada en el sistema, esto reporta la máxima flexibilidad y permite integrar sin problemas productos de otros fabricantes a través de los interfaces software estandarizados. Esta integración total se conoce como CIM (Computer Integrated Manufacturing).

En la industria coexisten una serie de equipos y dispositivos dedicados al control de una maquina o una parte cerrada de un proceso. El desarrollo de las redes industriales ha establecido una forma de unir gran parte de dispositivos, aumentando el rendimiento y proporcionando nuevas posibilidades.

2.1.1. Modelo OSI De Comunicación

La Universidad de Oviedo (2006) menciona que el modelo OSI (Open Systems Interconnection) es una propuesta de estandarización para las redes de ordenadores que permiten interconectar sistemas abiertos y ofrece a los usuarios la posibilidad de garantizar y controlar la interoperatividad de los productos entre sí.

Stallings (2000) menciona que el modelo OSI es un protocolo de 7 capas, en donde los protocolos se desarrollan de forma tal que realicen las funciones de cada una de las capas, en la figura 1 se definen brevemente las funciones que se realizan en cada capa.

<p style="text-align: center;">Aplicación</p> <p style="text-align: center;">Proporciona el acceso al entorno OSI y servicios de información distribuida para los usuarios.</p>
<p style="text-align: center;">Presentación</p> <p style="text-align: center;">Proporciona a los procesos de aplicación, independencia respecto a las diferencias en la representación de los datos (sintaxis).</p>
<p style="text-align: center;">Sesión</p> <p style="text-align: center;">Proporciona el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las acciones cooperadoras.</p>
<p style="text-align: center;">Transporte</p> <p style="text-align: center;">Proporciona seguridad, transferencias y transparencia de datos entre los puntos finales; proporciona además procedimientos de recuperación de errores y control de flujo origen – destino.</p>
<p style="text-align: center;">Red</p> <p style="text-align: center;">Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y transmisión utilizada para conectar los sistemas, es responsable del establecimiento, mantenimiento y cierre de las conexiones.</p>
<p style="text-align: center;">Enlace de Datos</p> <p style="text-align: center;">Proporciona un servicio de transferencia de datos seguro a través del enlace físico; envía bloques de datos (tramas) llevando a cabo la sincronización, el control de errores y flujo necesario.</p>
<p style="text-align: center;">Físico</p> <p style="text-align: center;">Se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico, está relacionado con las características mecánicas, eléctricas, funciones y procedimientos para acceder al medio físico.</p>

Figura 1. Capas del modelo OSI (William Stallings, 2000)

Las capas del modelo OSI describen el proceso de transmisión de los datos dentro de una red. Las dos únicas capas del modelo con las que interactúa el usuario son la capa física y de aplicación.

2.1.2. Fundamentos De Comunicaciones De Campo

Rosado (2009) menciona que los buses de comunicación sustituyen el cableado entre sensores-actuadores y los correspondientes elementos de control, estos tipos de buses deben ser de bajo coste y que permitan envío/recepción en tiempo real, permitir la transmisión serie sobre un bus digital de datos con capacidad de interconectar controladores con todo tipo de dispositivos de entrada-salida y permitir controladores esclavos inteligentes.

Además, deben gestionar mensajes cortos eficientemente, tener la capacidad de manejar tráfico de eventos discretos, poseer mecanismos de control de errores (detección y corrección), transmitir mensajes prioritarios, poder recuperarse rápidamente de eventos anormales en la red y responder rápidamente a los mensajes recibidos. Por regla general, tienen un tamaño pequeño entre 5 a 50 nodos, utilizan tráfico de mensajes cortos para control y sincronización entre los dispositivos. Según la cantidad de datos a transmitir, se dividen en buses de alto nivel, buses de dispositivos (unos pocos bytes a transmitir) y buses actuador/sensor (se transmiten datos a nivel de bit), en ningún caso llegan a transmitir grandes bloques de información, entre los buses de comunicaciones de campo más destacados se menciona los siguientes:

- Red sensor/actuador AS-i
- Buses orientados a dispositivos: CAN BUS y LONworks
- Buses de campo: HART, PROFIBUS, DeviceNet, MODBUS

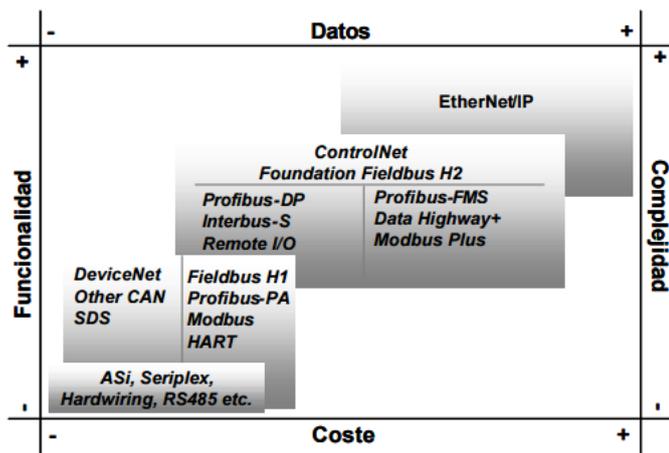


Figura 2. Clasificación de diversos buses de campo (Alfredo Rosado, 2009)

2.1.3. Fundamentos De Comunicaciones Sobre IP

Desde el punto de vista físico, Industrial Ethernet constituye una red eléctrica sobre la base de una línea coaxial apantallada, un cableado Twisted Pair o una red óptica sobre la base de un conductor de fibras ópticas. Industrial Ethernet está definida por el estándar internacional IEEE 802.3, es la red para el nivel de control y para el nivel de célula.



Figura 3. Pirámide de Comunicación Industrial (D. Olmos y Fco. Barros, 2008)

Desde entonces, tal como ocurrió con la señalización analógica, se realizaron continuos desarrollos en control de procesos para unificar tanto las comunicaciones

entre dispositivos como los perfiles a los que estos debían responder para garantizar el comportamiento estandarizado.

La implantación de Ethernet como soporte para los protocolos de nivel superior era clara a nivel empresarial (estructura de producción) y rápidamente bajó al nivel de Sala de Información (niveles MES y SCADA). El salto al nivel de Control (comunicación autómatas y sistemas HMI locales) se convirtió en una realidad a medida que la electrónica de red se implementó en las unidades de control de proceso, la clasificación de pirámide de automatización se muestra en la figura 3.

Hurtado (2015) caracteriza a Ethernet industrial como una potente red de área y célula con los estándares IEEE 802.3 con la que se pueden crear redes de comunicación eficaces con gran extensión. Es un sistema que ofrece todo el potencial Ethernet, pero utiliza medidas de seguridad, incluidas las de control de acceso y autenticación, seguridad en la conectividad y administración, a fin de asegurar y garantizar la confidencialidad e integridad de la red y ofrecer datos libres de interferencias.

2.1.4. Ventajas de comunicaciones de campo

Rosado (2009) menciona que las ventajas principales que se obtienen en su utilización son:

- Mejor calidad y cantidad en el flujo de datos.
- Ahorro de coste de cableado e instalación, según se muestra en la figura 4.
- Facilidad en ampliación o reducción del número de equipos conectados en el sistema.
- Reducción de errores en la instalación, número de terminales y cajas de conexión.

- Implementación sencilla y fácil monitorización.
- Estaciones compactas y flexibilidad de conexiones.
- Transmisión de señales en tiempo real.

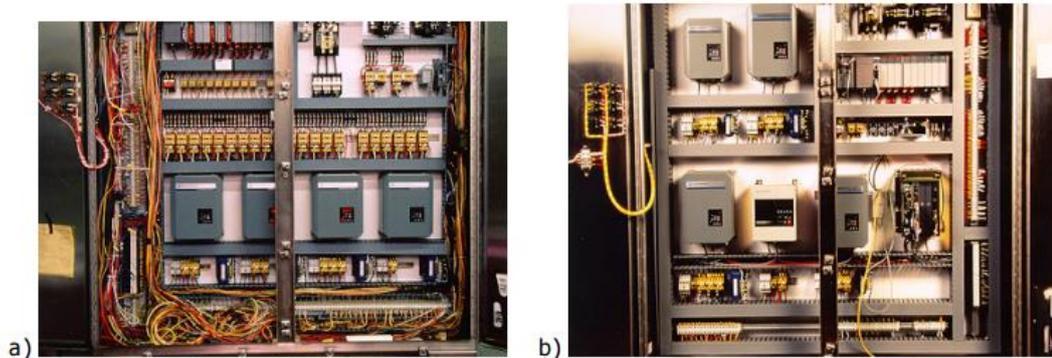


Figura 4. Cableado de red: a) Cableado convencional; b) Cableado con red industrial (Alfredo Rosado, 2009)

2.1.5. Ventajas de comunicación sobre IP

Hurtado (2015) describe las ventajas de las comunicaciones sobre IP:

- Funcionamiento en “tiempo real” para datos de I/O, es decir, programar/organizar el intercambio cíclico con cada esclavo, con alta prioridad y tiempos fijos.
- Se pueden utilizar componente estándares Ethernet como son los cables y switches.
- Los dispositivos se direccionan por medio de un nombre (dirección IP) y no mediante número de nodo.
- Comunicación fácil, rápida, flexible y abierta.
- Protocolo abierto, estándar industrial.
- Alta velocidad, tiempo de ciclo por dispositivo.
- Grandes velocidades de transmisión (10-100-1000 Mps).

2.1.6. Topología de buses de comunicación industrial

Olmos y Barros (2008) mencionan las topologías más utilizadas a nivel industrial son:

Conexión en Bus: Un único cable interconecta todos los equipos, esta topología está representada en la figura 5.

- Ventajas: Barato, flexible, fácil cablear, fallo de un host no provoca fallos de red.
- Desventajas: Rotura del cable afecta a todos los host, límites de longitud de cable y número de usuarios, difícil localizar fallas, añadir usuarios baja rendimiento de red.

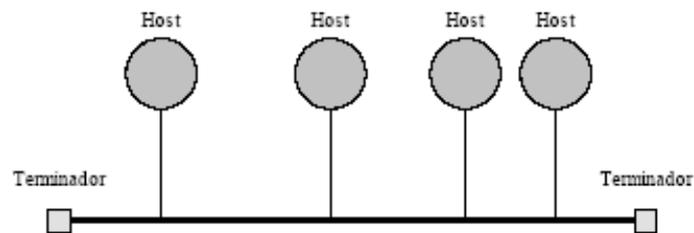


Figura 5. Conexión Bus (Barros y Olmos, 2008)

Anillo: Cada equipo se conecta con otros dos, esta topología se muestra en la figura 6.

- Ventajas: Igual acceso para todos los equipos, añadir usuarios no afecta excesivamente.
- Desventajas: Un fallo del cable afecta a muchos usuarios, conexionado y cableado costoso, difícil añadir equipos.

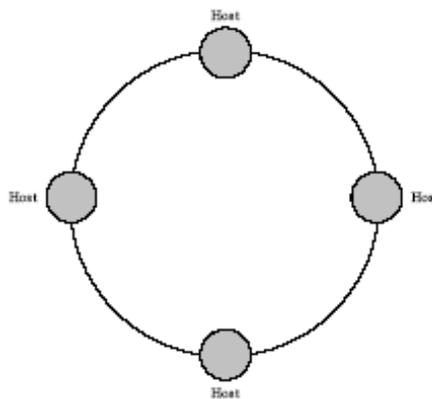


Figura 6. Conexión anillo (Barros y Olmos, 2008)

Estrella: Todos los nodos están conectados a un nodo central, esta topología está representada en la figura 7.

- Ventajas: Fácil añadir nuevas estaciones, el manejo y monitorización de la red está centralizado, la rotura de un cable solo afecta a un usuario.
- Desventajas: Mucho cableado, si falla el computador central se inutiliza la red.

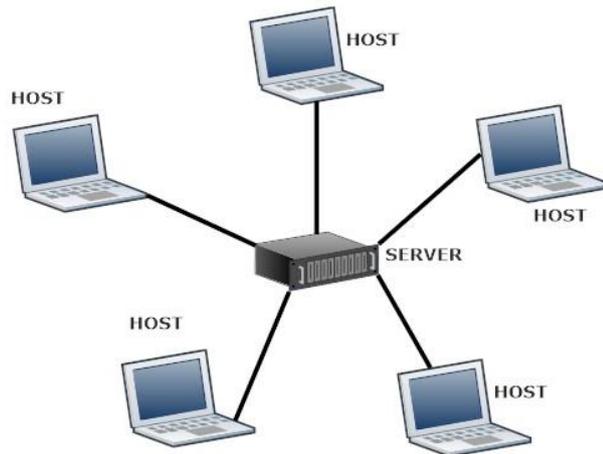


Figura 7. Conexión estrella (Autores, 2017)

Árbol: Sucesiones de estrellas, disminuyendo la longitud del cableado, esta topología está representada en la figura 8.

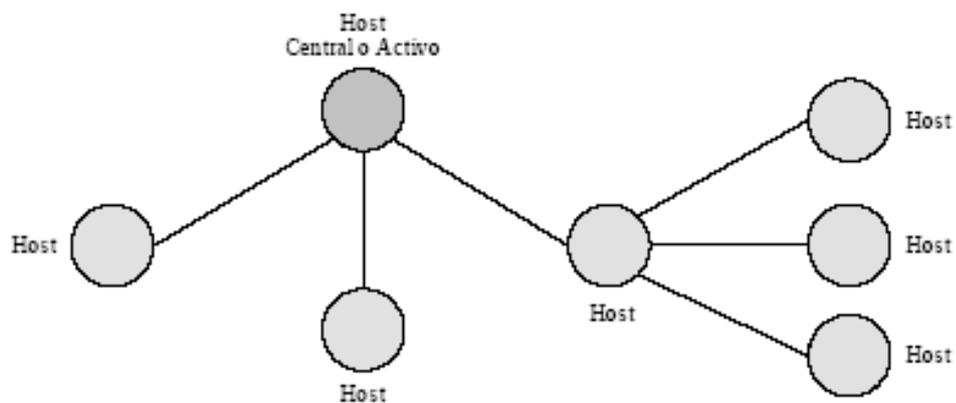


Figura 8. Conexión árbol (Barros y Olmos, 2008)

2.2. Protocolo de comunicación Modbus

Rosado (2009) menciona que este protocolo apareció en 1979 para transmitir y recibir datos de control entre controladores y los sensores a través del puerto RS-232 (comunicación punto a punto), con un alcance máximo de 350 metros. No ha sido estandarizado por ninguna entidad, pero sus especificaciones están disponibles y funciona mediante el sistema maestro/esclavo, posee dos modos esenciales de funcionamiento, modo ASCII enviando dos caracteres (2 bytes) para cada mensaje, pudiendo haber hasta 1 segundo de tiempo de diferencia entre ellos, y modo RTU (Remote Terminal UNit), donde se envían 4 caracteres hexadecimales (4 bits cada uno) para cada mensaje. Esta última opción es más empleada en transmisores inalámbricos ambos tipos emplean como medio físico el par trenzado apantallado y la tensión de alimentación es independiente para cada dispositivo.

2.2.1. Descripción general del protocolo

Ruiz (2002) describe a Modbus como un protocolo de comunicación sin estado, es decir, cada solicitud del maestro es tratada independientemente por el esclavo y es considerada una nueva solicitud no relacionada a las anteriores, haciendo a las transacciones de datos altamente resistentes a rupturas causadas por ruido y requiriendo mínima información de recuperación para ser mantenida la transacción en cualquiera de los dos terminales.

Las operaciones de programación del otro lado, esperan una comunicación orientada a la conexión, es decir, las máquinas de origen y destino establecen un canal de comunicaciones antes de transferir datos. Este tipo de operaciones son implementadas

de diferentes maneras por las diversas variantes de MODBUS (Modbus RTU, Modbus ASCII, Modbus PLUS).

Hurtado (2015) menciona que la transmisión de información en este protocolo no está comprometida a ningún tipo de datos, lo que implica cierta flexibilidad al momento del intercambio de información. La representación del valor vendrá definida por la especificación que el fabricante del dispositivo, lo que permite un amplio rango de valores. El bus se compone de una estación activa (principal) y de varias estaciones pasivas (subordinadas). La estación principal es la única que puede iniciar el intercambio de información, no pudiendo las estaciones subordinadas comunicarse directamente. Existen dos formas de intercambios:

Pregunta/respuesta: La estación principal transmite preguntas a una determinada subordinada, que a su vez transmite una respuesta a la principal.

Difusión: La estación principal transmite un mensaje a todas las estaciones subordinadas del bus, que ejecutan la orden sin transmitir ninguna respuesta.

En una red Modbus solo existe 1 estación principal y hasta 247 estaciones subordinadas (direcciones en el rango 1 a 247), sólo la principal puede iniciar una petición. Para comunicarse con las estaciones subordinadas, la principal envía unas tramas que llevan la dirección del receptor, la función a realizar, los datos necesarios para realizar dicha función y un código de comprobación de errores. Cuando la trama llega a la estación subordinada direccionada, ésta lee el mensaje, y si no ha ocurrido ningún error realiza la tarea indicada.

Entonces la subordinada envía una trama respuesta formada por: la dirección de la subordinada, la acción realizada, los datos adquiridos como resultado de la acción y un código de comprobación de errores. Si el mensaje enviado por la principal es de tipo difusión (broadcast), es decir, para todas las estaciones subordinadas (se indica con dirección 0), no se transmite ninguna respuesta. Si la estación receptora recibe un mensaje con algún error, contesta a la principal con un código de error (Función ilegal, Datos de direccionamiento ilegales, Datos de valores ilegales, Fallo en el dispositivo, Mensaje rechazado). En la figura 9 se muestra la conexión de dispositivos serie y la identificación de cada uno en la red.

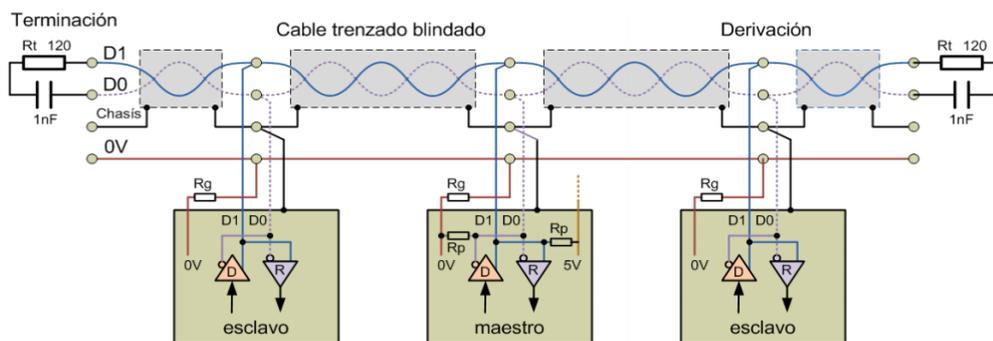


Figura 9. Conexión RS – 485 de dispositivos (Alfredo Rosado, 2009)

2.2.2. Mapa de registro

En Modbus cada tipo de dato se mapea en un rango de memoria concreto, dicha dirección está dentro de un rango definido según el tipo de datos y función que se quiera utilizar, el mapa de dirección se muestra en la figura 10:

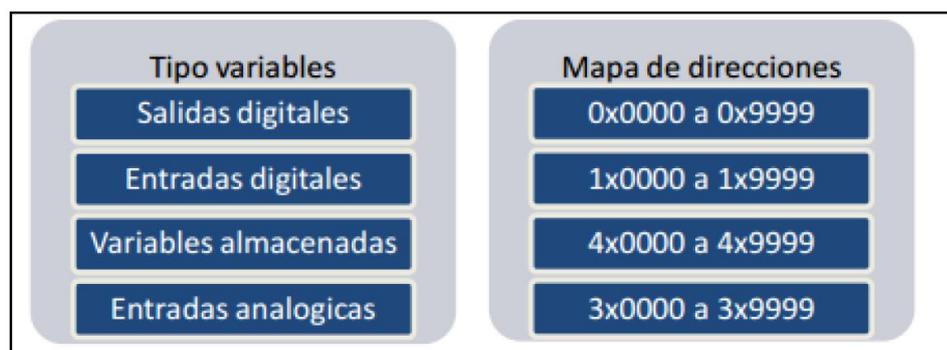


Figura 10. Mapa de Registro Modbus (Mista, 2012)

1 -1000 (Salidas Digitales): 1 bit por dirección para indicar el estado de una salida o relé (0_OFF, 1_ON). Las direcciones de este rango se suelen acceder mediante las funciones 1 para lectura, 5 para escritura, 15 de escritura múltiple, según (SIEMENS_AG, 2010).

10001 – 20000 (Entradas Digitales): 1 bit por dirección para leer el estado de una entrada digital (0 desactivada, 1 activada). Las direcciones de este rango se suelen acceder con la función 2 para lectura y llevan implícita la dirección 10001 como dirección base (para acceder a una dirección bastara con especificar la distancia entre esta y la dirección base) (SIEMENS_AG, 2010).

30001 – 40000 (Entradas Analógicas): 16 bits por dirección con el estado de las medidas o entradas analógicas. Dependiendo del dispositivo, este puede hacer uso de más de un registro para almacenar medidas de 32 bits. Las direcciones de este rango se acceden mediante la función 4 para lectura y llevan implícita la dirección 30001 como dirección base (para acceder a una dirección bastara con especificar la distancia entre esta y la dirección base) (Mista, 2012).

40001 – 50000 (Salidas Analógicas): 16 bits con los registros de salidas analógicas o de propósito general (Output Registers – Holding Registers). Se acceden con las funciones 3 para lectura, 6 para escritura o 16 de escritura múltiple y llevan implícita la dirección 40001 como dirección base (Bartolomé, 2011).

2.2.3. Estructura de tablas

Gálvez (2011) menciona que la configuración de maestro Modbus permite al controlador enviar una solicitud Modbus a un esclavo y esperar una respuesta. El tamaño máximo de las tramas enviadas o recibidas es de 250 bytes, la trama de palabras asociadas a la instrucción de envío se muestra en la tabla 1.

Tabla 1. Estructura de tabla de datos

	Byte más significativo	Byte menos significativo
Tabla de Control	Comando	Longitud (envío/recepción)
	Offset de recepción	Offset de envío
Tabla de Envío	Byte 1 enviado	Byte 1 enviado
	...	Byte N enviado
	Byte N+1 enviado	
Tabla de Recepción	Byte 1 recibido	Byte 2 recibido
	...	Byte p recibido
	Byte p+1 recibido	

Formación de trama de datos para comunicación Modbus (Paul Gálvez, 2011)

- **Estructura de Tabla Control**

El byte **Longitud** contiene la longitud de la tabla de envío (250 bytes máx.), que se sobrescribe con el número de caracteres recibidos al final de la recepción, en caso de que se solicite. Este parámetro es la longitud en bytes de la tabla de envío. Si el parámetro de offset del envío es igual a 0, será igual a la longitud de la trama de envío. Si el parámetro de offset del envío no es igual a 0, no se enviará un byte de la tabla de envío (indicado por el valor de offset).

El byte **Comando**, en caso de que se produzca una solicitud RTU Modbus (excepto para una difusión), debe ser siempre igual a 1 (para envío y la recepción).

El byte de **offset de envío** contiene el rango (1 para el primer byte, 2 para el segundo byte, etc.) dentro de la tabla de envío del byte que se ignorará cuando se envíen los bytes. Se utiliza para gestionar los envíos asociados a los valores de byte/palabra del protocolo Modbus.

El byte de **offset de recepción** contiene el rango (1 para el primer byte, 2 para el segundo byte, etc.) dentro de la tabla de recepción que se agregará cuando se envíe el paquete.

- **Estructura de Tabla de envío/recepción**

Las entradas de **byte X enviado** contienen los datos del protocolo Modbus (codificación RTU) que se va a emitir. El primer byte contiene la dirección del dispositivo (específica o general), el segundo byte contiene el código de función y el resto contiene información asociada al código de función.

Los **bytes X recibidos** contienen los datos del protocolo Modbus (codificación RTU) que se van a recibir, el primer byte contiene la dirección del dispositivo, el segundo byte contiene el código de función (o código de respuesta) y el resto contiene información asociada al código de función.

2.2.4. Códigos de función

Los siguientes códigos descritos en la tabla 2, determinan la función Modbus utilizados y soportados por todos los dispositivos que cumplen con las especificaciones del estándar.

Tabla 2. Principales códigos de funciones

FUNCIÓN	CÓDIGO	DESCRIPCIÓN
1	01H	Lectura de múltiples bits (bobinas) de salida o internos
2	02H	Lectura de múltiples (bobinas) bits de entradas
3	03H	Lectura múltiples registros (palabras) de salida o internos
4	04H	Lectura múltiples registros (palabras) de entrada
5	05H	Escritura de una bobina (Bit)
6	06H	Escritura de un registro (palabra)
15	0FH	Escritura de múltiples bits (bobinas)
16	10H	Escritura de múltiples registros (palabras)

Se describen los principales códigos de funciones. (Barragán, 2013)

- **Lectura de N bits.**

Permite realizar la lectura del estado de entradas o salidas digitales. 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 aprovecha todos los bits del byte (SIEMENS_AG, 2010), la tabla 3 se muestra la estructura de estas funciones.

Tabla 3. Tabla de peticiones 01 y 02

	Tabla índice	Byte más significativo	Byte menos significativo
Tabla de Control	0	01(emisión/recepción)	06(longitud de emisión)
	1	03(offset de recepción)	00(offset de emisión)
Tabla de Envío	2	Slave a (1 a 247)	01 o 02 (código de petición)
	3	Dirección del primer bit que se va a leer	
	4	N1= Numero de bits que se van a leer	

Tabla de Recepción	5	Slave a (1 a 247)	01 o 02 (código de respuesta)
	6	00 (byte añadido por la opción RX del offset)	N2= Numero de bytes de datos que se van a leer= $[1+(N1-1)/8]$
	7	Valor del 1er byte (00 o 01)	Valor del 2do byte (si N1>1)
	8	Valor del 3er byte (si N1>1)	
	...		
(N2/2)+6 (si N2 es par) (N2/2+1)+6 (si N2 es impar)		Valor del byte (N2)^2 (si N1>1)	

Formación de trama para peticiones de lectura de E/S digitales (Paul Gálvez, 2011)

- **Lectura de N palabras.**

Permite realizar la lectura del valor de las entradas y salidas analógicas. El maestro indica la dirección base y numero de palabras a leer a partir de esta, mientras que el esclavo indica en la respuesta el número de bytes retornados (Kyron_Ingenieria, 2014), la tabla 4 se muestra a detalle los parámetros y datos de envío/recepción que utiliza la trama para esta función Modbus.

Tabla 4. Tabla de peticiones 03 y 04

	Tabla índice	Byte más significativo	Byte menos significativo
Tabla de Control	0	01(emisión/recepción)	06(longitud de emisión)
	1	03(offset de recepción)	00(offset de emisión)
Tabla de Envío	2	Slave a (1 a 247)	03 o 04 (código de petición)
	3	Dirección de la primera palabra que se va a leer	
	4	N= Numero de palabras de lectura	
Tabla de Recepción	5	Slave a (1 a 247)	03 o 04 (código de respuesta)
	6	00 (byte añadido por la opción RX del offset)	2*N= (número de bytes leídos)
	7	Primera palabra leída	
	8	Segunda palabra leída (si N>1)	
	...		
	N+6	Palabra N leída (si N>2)	

Formación de trama para peticiones de lectura de N palabras (Paul Gálvez, 2011)

- **Escritura de 1 Bit.**

Permite modificar el estado de un coil del esclavo. Es decir mediante este comando podemos modificar el estado de un bit de alguna de las variables internas del esclavo u ordenar la ejecución o activación de un mando. Actúa sobre zona de memoria de las salidas digitales. 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 (Kyron_Ingenieria, 2014), véase la tabla 5, en donde se detalla los datos de envío/recepción de esta función de escritura Modbus.

Tabla 5. Tabla de peticiones 05

	Tabla índice	Byte más significativo	Byte menos significativo
Tabla de Control	0	01(emisión/recepción)	06(longitud de emisión)
	1	03(offset de recepción)	00(offset de emisión)
Tabla de Envío	2	Slave a (1 a 247)	05 (código de petición)
	3	Dirección del bit que se va a escribir	
	4	Valor del bit que se va a escribir	
Tabla de Recepción	5	Slave a (1 a 247)	05 (código de respuesta)
	6	Dirección del bit escrito	
	7	Valor escrito	

Formación de trama para peticiones de escritura de N bit (Paul Gálvez, 2011)

- **Escritura de 1 palabra**

Permite la escritura en las salidas analógicas del esclavo (ya sea una señal o valor interno del equipo), y por tanto actúa sobre la zona de memoria de las salidas analógicas. Se debe indicar la dirección del valor que se quiere modificar y la magnitud que quiera asignar. (SIEMENS_AG, 2010), en la tabla 6 se identifican los datos de envío/recepción de esta función.

Tabla 6. Tabla de peticiones 06

	Tabla índice	Byte más significativo	Byte menos significativo
Tabla de Control	0	01(emisión/recepción)	06(longitud de emisión)
	1	00(offset de recepción)	00(offset de emisión)
Tabla de Envío	2	Slave a (1 a 247)	06 (código de petición)
	3	Dirección de la palabra que se va a escribir	
	4	Valor de la palabra que se va a escribir	
Tabla de Recepción	5	Slave a (1 a 247)	06 (código de respuesta)
	6	Dirección de la palabra escrita	
	7	Valor escrito	

Formación de trama para peticiones de escritura de 1 palabra (Paul Gálvez, 2011)

- **Escritura de N bits**

Permite la modificación simultánea de varios bits de salida en el esclavo, pasándolos a OFF (0) o a ON (1) según convenga, actúa sobre la zona de memoria de las salidas digitales. El maestro especifica la dirección del primer bit o bobina que se quiere modificar seguido del número total de bit a escribir, el esclavo responde con una trama similar indicando las direcciones que han modificado en las direcciones correspondientes. (Kyron_Ingeniria, 2014), en la tabla 7 se muestra la estructura de este tipo de función.

Tabla 7. Tabla de peticiones 15

	Tabla índice	Byte más significativo	Byte menos significativo
Tabla de Control	0	01(emisión/recepción)	8 + número de bytes (emisión)
	1	00(offset de recepción)	07(offset de emisión)
Tabla de Envío	2	Slave a (1 a 247)	15 (código de petición)
	3	Número del primer bit que se va a escribir	
	4	N1= Numero de bits que se van a escribir	
	5	00 (byte no enviado efecto RX del offset)	N2= Numero de bytes de datos que se van a escribir= $[1+(N1-1)/8]$
	6	Valor del 1er byte	Valor del 2do byte
	7	Valor del 3er byte	Valor del 4to byte

	...		
	(N2/2)+6 (si N2 es par) (N2/2+1)+6 (si N2 es impar)	Valor del byte N2	
Tabla de Recepción		Slave a (1 a 247)	15 (código de respuesta)
		Dirección del primer bit escrito	
		Dirección de los bits escritos (=N1)	

Formación de trama para peticiones de escritura de N bits (Paul Gálvez, 2011)

- **Escritura de N palabras**

Permite realizar la escritura en un grupo de registros, se debe especificar la dirección a partir de a que queremos comenzar a actualizar valores, el número de valores que queremos actualizar y la lista de valores que queremos asignar a estos registros, estos registros son direccionados a partir de la dirección 0 (es decir el registro 40001 se direcciona 0) (SIEMENS_AG, 2010), la tabla 8 se muestra la descripción de la trama.

Tabla 8. Tabla de peticiones 16

	Tabla índice	Byte más significativo	Byte menos significativo
Tabla de Control	0	01(emisión/recepción)	8 + (2*N) (longitud de emisión)
	1	00(offset de recepción)	07(offset de emisión)
Tabla de Envío	2	Slave a (1 a 247)	16 (código de petición)
	3	Dirección de la primera palabra que se va a escribir	
	4	N= Numero de palabras que se va a escribir	
	5	00(byte no enviado por offset)	2*N= N de bytes que se escriben
	6	Primer valor de la palabra que se va a escribir	
	7	Segundo valor que se va a escribir	
	...		
	N+5	Valores N que se van a escribir	
Tabla de Recepción	N+6	Slave a (1 a 247)	16 (código de respuesta)
	N+7	Dirección de la primera palabra escrita	
	N+8	Dirección de las palabras escritas (=N)	

Formación de trama para peticiones de escritura de N palabras (Paul Gálvez, 2011)

2.3. Protocolo de comunicación Ethernet

Hurtado (2015) menciona que es una potente red de área y célula de acuerdo con los estándares IEEE 802.3 (Ethernet) con la que se pueden crear redes de comunicación eficaces de gran extensión. Es un sistema que ofrece todo el potencial que ofrece Ethernet, pero utiliza medidas de seguridad, incluidas las de control de acceso y autenticación, seguridad en la conectividad y administración, a fin de asegurar y garantizar la confidencialidad e integridad de la red y ofrecer datos libres de interferencias.

2.3.1. Descripción de funcionamiento del protocolo

Hurtado (2015) describe como un estándar de transmisión de datos para redes de área local que utiliza los protocolos TCP/IP, bajo el método de control de acceso al medio conocido como CSMA/CD "*Carrier Sense Multiple Access, with Collision Detection*". CSMA/CD determina cómo y cuándo un paquete de dato es ubicado en el la red. Antes que un dispositivo Ethernet esté habilitado a transmitir datos, primero tiene que asegurarse de que el medio está "libre" y no hay otros dispositivos transmitiendo. Así cuando la red está libre, los dispositivos inician la transmisión. Durante el proceso de transmisión, el dispositivo continúa escuchando en la red para ver si algún otro dispositivo está transmitiendo al mismo tiempo. Si no hay ningún otro, entonces el paquete de datos se considera enviado al receptor sin interrupciones. Sin embargo, si durante la transmisión detecta que otro dispositivo también está transmitiendo se puede dar una colisión de datos, así pues, ambos detendrán sus transmisiones y realizaran un proceso conocido como back-off en el que esperaran un tiempo aleatorio antes de intentar volver a transmitir nuevamente.

Rosado (2009) indica que se implementaba originalmente sobre cable coaxial, codificándose la señal en banda base mediante el código Manchester. Sin embargo se han desarrollado especificaciones para que la red Ethernet se pueda implementar sobre otros soportes físicos: par trenzado, fibra óptica, etc. y soportando mayores velocidades de transmisión. Es más, el original control de acceso al medio CSMA/CD ha sido prácticamente desplazado por las técnicas de conmutación (Ethernet conmutada), que agilizan el tráfico de la red, aumentan el ancho de banda de transmisión disponible, aumentan el número de nodos que se pueden conectar a una misma red local y minimizan tanto la posibilidad de pérdida de mensaje como retardo de propagación de estos hacia su destino. Este hecho ha provocado que Ethernet se haya incorporado definitivamente al entorno industrial como un medio de transmisión fiable, rápido y prácticamente determinista. La Universidad de Oviedo (2006) menciona que este modelo se puede considerar de 4 capas y es el estándar por defecto debido a su enorme difusión y eclipse al modelo OSI cuya implantación ha sido mucho menor en comparación al modelo TCP/IP.

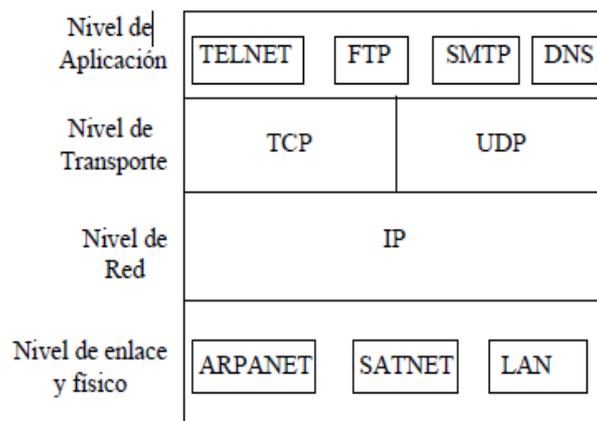


Figura 11. Modelo TCP/IP y algunos de sus protocolos (Universidad de Oviedo, 2006)

- **Capa de enlace y física**

Rosado (2009) menciona que es equivalente a las capas físicas y de enlace del modelo OSI. El medio de transmisión puede ser cualquiera que soporte el transporte de las unidades de datos que genera el protocolo IP de la capa de red.

- **Capa de Internet (Capa de Red)**

Rosado (2009) menciona su función es permitir que el equipo inserte paquetes en cualquier red, y que estos viajen independientemente hacia su destino (que quizá sea una subred distinta). Incluso pueden llegar a distintos hosts del que fueron enviados.

La capa de internet define un tipo oficial de paquete (datagrama IP) y un protocolo llamado IP (internet protocol). La principal obligación de la capa es distribuir los paquetes hacia su destino, por ello su función es el encaminamiento de los mensajes y evitar atascos, aunque su mecanismo de control de congestión son bastantes limitados. Equivale a la capa de red del modelo OSI.

- **Capa de Transporte**

Rosado (2009) indica que está diseñada para permitir el diálogo entre equipos extremo a extremo, igual que la capa de transporte de modelo OSI, pudiendo estar estos equipos situados en la misma habitación o miles de kilómetros de distancia.

Utiliza 2 protocolos: TCP (Transmission Control Protocol) y UDP (User Datagram Protocol). El primero es un protocolo orientado a conexión, libre de errores, que permite enviar bloques de bytes de una máquina a otra por un canal libre de errores, TCP también administra el control de flujo. El protocolo UDP es un protocolo sin conexión basado en datagramas simples que no se aseguran la llegada de los datos a su destino.

- **Capa de Aplicación**

Rosado (2009) menciona que el modelo TCP/IP no tiene las capas de presentación ni de sesión. La experiencia ha demostrado que esta aproximación es la correcta. Esta capa contiene todos los protocolos a alto nivel como por ejemplo: TELNET (terminal remoto), FTP (transferencia de ficheros), SMTP (correo electrónico), DNS (servidor de nombres), NNTP (news), HTTP (web) y todos aquellos nuevos protocolos que se van incorporando.

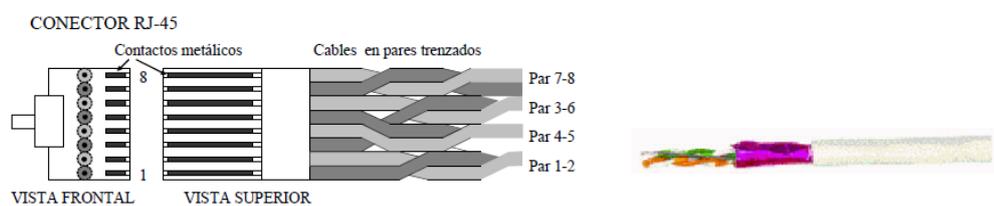


Figura 12. Cable RJ45 para conexión TCP/IP (Universidad de Oviedo, 2006)

2.3.2. Descripción de trama de datos

El paquete de un mensaje Ethernet (trama Ethernet) consta de los siguientes campos:

Preambulo	Dir. destino	Dir. origen	Tipo	Datos	CRC
8	6	6	2	46-1500	4

Figura 13. Trama Ethernet (Universidad de Oviedo, 2006)

- **Preámbulo:** Patrón de unos y ceros que indica a las estaciones receptoras que una trama es Ethernet o IEEE 802.3. La trama Ethernet incluye un byte adicional que es el equivalente al campo Inicio de Trama (SOF) de la trama IEEE 802.3.
- **Inicio de trama (SOF):** Byte delimitador de IEEE 802.3 que finaliza con dos bits 1 consecutivos, y que sirve para sincronizar las porciones de recepción de trama de todas las estaciones de la red. Este campo se especifica explícitamente en Ethernet.

- **Direcciones destino y origen:** Incluye las direcciones físicas (MAC) únicas de la máquina que envía la trama y de la máquina destino. La dirección origen siempre es una dirección única, mientras que la de destino puede ser de broadcast única, de broadcast múltiple o de broadcast.
- **Tipo (Ethernet):** Especifica el protocolo de capa superior que recibe los datos una vez que se ha completado el procesamiento Ethernet.
- **Longitud (IEEE 802.3):** Indica la cantidad de bytes de datos que sigue este campo.
- **Datos:** Incluye los datos enviados en la trama. En la especificación IEEE 802.3, si los datos no son suficientes para completar una trama mínima de 64 bytes, se insertan bytes de relleno hasta completar ese tamaño.
- **Secuencia de verificación de trama (FCS):** Contiene un valor de verificación CRC (Control de Redundancia Cíclica) de 4 bytes, creado por el dispositivo emisor y recalculado por el dispositivo receptor para verificar la existencia de tramas dañadas. Cuando un paquete es recibido por el destinatario adecuado, les retira la cabecera de Ethernet y el checksum de verificación de la trama, comprueba que los datos corresponden a un mensaje IP y entonces lo pasa a dicho protocolo para que lo procese.

2.4. Controladores lógicos programables utilizados en el proyecto

En el presente proyecto hace uso de PLC's específicos ya existentes en los laboratorios de la Universidad Politécnica Salesiana sede Guayaquil, en esta sección se detallara un breve resumen de las características de los mismos.

2.4.1. Descripción general de funcionamiento

De acuerdo a SIEMENS (2015) un controlador lógico programable o PLC, es un computador especialmente diseñado para la automatización industrial, para el control de una maquina o proceso industrial. El PLC nace como solución al control de circuitos complejos de automatización, a él se conectan los sensores (finales de carrera, pulsadores, etc) por una parte y los actuadores (contactores, lámparas, etc).

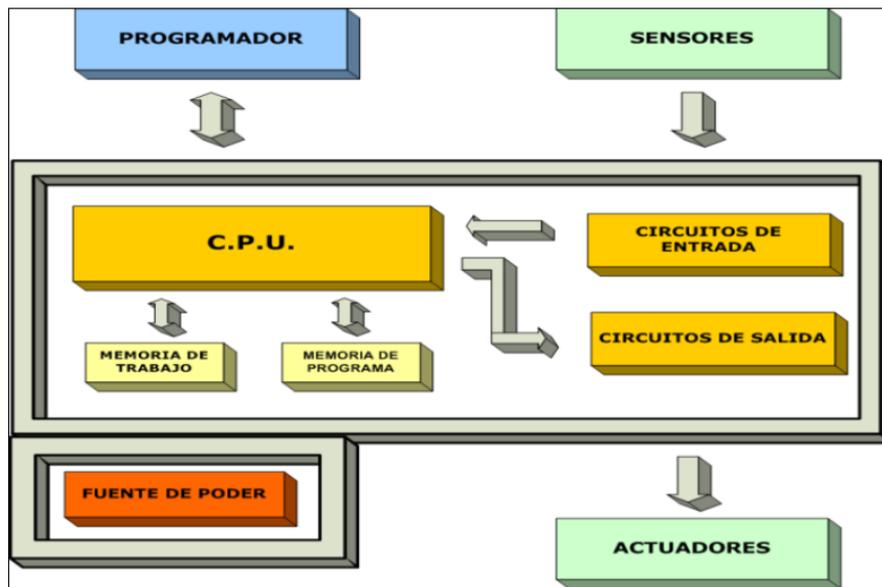


Figura 14. Arquitectura del PLC (Alfredo Rosado, 2009)

Un PLC permite controlar o proteger un proceso industrial, posibilitando además las opciones de monitoreo y diagnóstico de condiciones (alarmas), presentándolas en un HMI (Human-Machine Interface) o pantalla de operación (SIEMENS, 2015) o presentándolas a una red de control superior. Es un ejemplo de control en tiempo real, pues reacciona automáticamente ante las condiciones variables que se está vigilando.

2.4.2. SIEMENS S7 – 1200

2.4.2.1. Características del PLC

SIEMENS (2009) describe el controlador lógico programable (PLC) S7-1200 como un controlador que ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización.

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 conteo y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes, incorpora un puerto PROFINET para la comunicación en una red PROFINET. Las características técnicas se describen en la tabla 9.

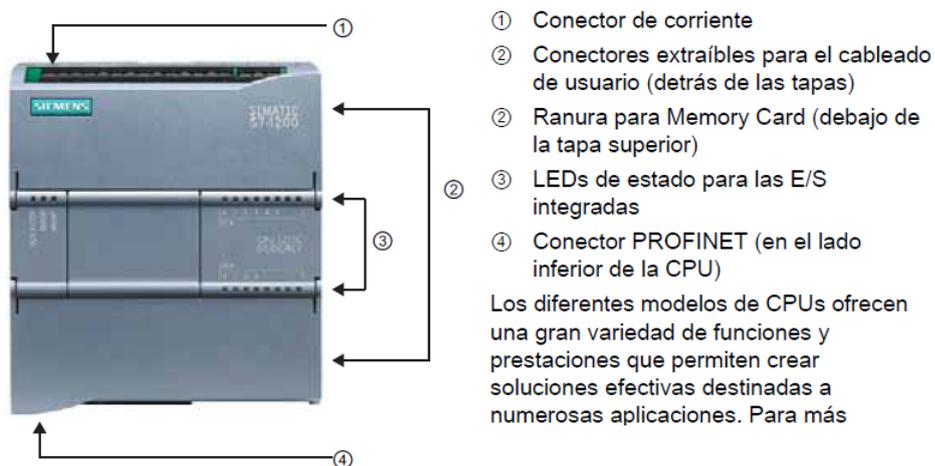


Figura 15. Vista de partes de CPU S7 – 1200 (SIEMENS, 2009)

SIEMENS (2009) menciona que la CPU soporta los siguientes tipos de bloques lógicos que permiten estructurar eficientemente el programa de usuario:

- Los bloques de organización (OBs) que definen la estructura del programa, algunos OBs tienen reacciones y eventos de arranque predefinidos. No obstante, también es posible crear OBs con eventos de arranque personalizados.
- Las funciones (FCs) y los bloques de función (FBs) contienen el código de programa correspondiente a tareas específicas o combinaciones de parámetros, cada FC o FB provee parámetros de entrada y salida para compartir datos con el bloque invocante.
- Los bloques de datos (DBs) que almacenan datos que pueden ser utilizados por los bloques del programa.

Tabla 9. Especificaciones de CPU's S7 – 1200

Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75		110 x 100 x 75
Memoria de usuario			
• Memoria de trabajo	• 25 KB		• 50 KB
• Memoria de carga	• 1 MB		• 2 MB
• Memoria remanente	• 2 KB		• 2 KB
E/S integradas locales			
• Digitales	• 6 entradas/4 salidas	• 8 entradas/6 salidas	• 14 entradas/10 salidas
• Analógicas	• 2 entradas	• 2 entradas	• 2 entradas
Tamaño de la memoria imagen de proceso	1024 bytes para entradas (I) y 1024 bytes para salidas (Q)		
Área de marcas (M)	4096 bytes		8192 bytes
Ampliación con módulos de señales	Ninguna	2	8
Signal Board	1		
Módulos de comunicación	3 (ampliación en el lado izquierdo)		
Contadores rápidos	3	4	6
• Fase simple	• 3 a 100 kHz	• 3 a 100 kHz 1 a 30 kHz	• 3 a 100 kHz 3 a 30 kHz
• Fase en cuadratura	• 3 a 80 kHz	• 3 a 80 kHz 1 a 20 kHz	• 3 a 80 kHz 3 a 20 kHz
Salidas de impulsos	2		
Memory Card	SIMATIC Memory Card (opcional)		
Tiempo de respaldo del reloj de tiempo real	Típico: 10 días / Mínimo: 6 días a 40 °C		
PROFINET	1 puerto de comunicación Ethernet		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Velocidad de ejecución booleana	0,1 µs/instrucción		

Características de CPU's más comunes de la serie S7 – 1200 (SIEMENS, 2009)

2.4.2.2. Software de programación

SIEMENS (2009) indica que el software STEP 7 Basic ofrece un entorno amigable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como PLCs y dispositivos HMI.

Ofrece dos lenguajes de programación (KOP y FUP) que permiten desarrollar el programa de control de la aplicación de forma fácil y eficiente. Para aumentar la productividad el TIA Portal ofrece dos vistas diferentes de las herramientas disponibles, la vista del portal y vista del proyecto las cuales se muestran en la figura 16.

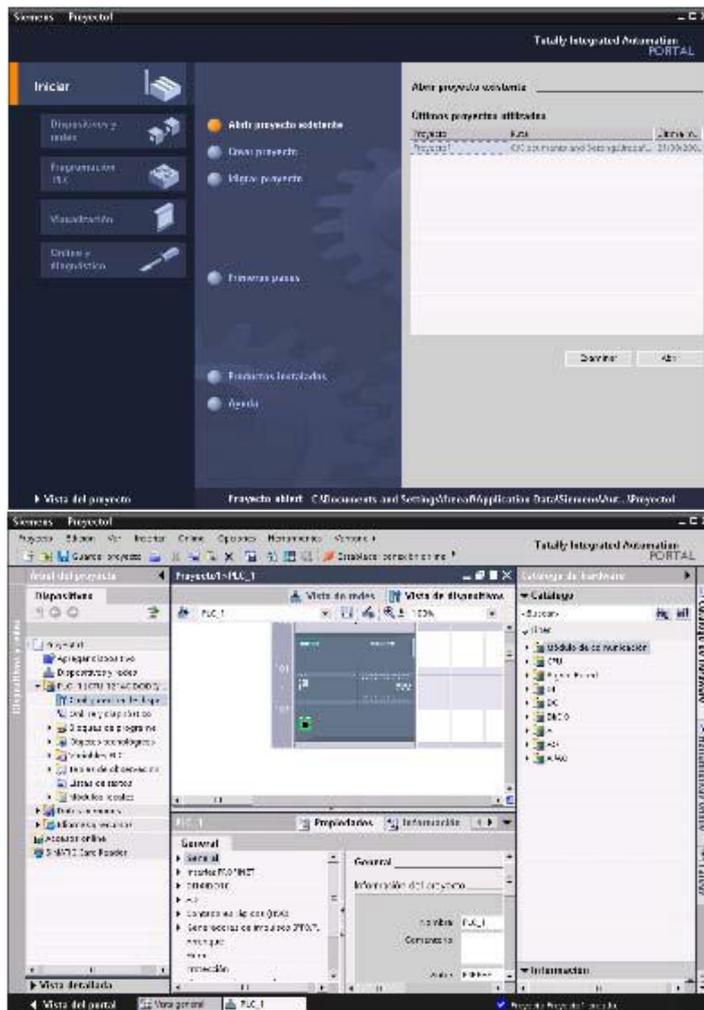


Figura 16. Vista del Portal y Vista del Proyecto (SIEMENS, 2012)

2.4.2.3. Librería Modbus TCP

SIEMENS (2014) menciona que un CPU operando como MODBUS RTU maestro (o Modbus TCP cliente) puede leer y escribir datos de estado en un MODBUS RTU esclavo (o Modbus TCP servidor). Un CPU operando como MODBUS RTU esclavo (o Modbus TCP servidor) puede ser supervisado por un dispositivo maestro, permitiendo el acceso a los datos de estado I/O en la memoria del CPU.

- Instrucciones Modbus TCP

MB_CLIENT: Realiza la conexión TCP cliente – servidor, enviando mensajes de comando, recibiendo respuesta del esclavo y controlando la desconexión de servidor, comunica como cliente Modbus TCP a través de un enlace PROFINET de la CPU S7 – 1200, para utilizar esta instrucción no se necesita hardware adicional, en la tabla 10 se describe en detalle este bloque.

Tabla 10. Bloque Instrucción MB_CLIENT

LAD / FBD	SCL	Description
	<pre>"MB_CLIENT_DB" (REQ:= bool_in_, DISCONNECT:= bool_in_, CONNECT_ID:= uint_in_, IP_OCTET_1:= byte_in_, IP_OCTET_2:= byte_in_, IP_OCTET_3:= byte_in_, IP_OCTET_4:= byte_in_, IP_PORT:= uint_in_, MB_MODE:= usint_in_, MB_DATA_ADDR:= udint_in_, MB_DATA_LEN:= uint_in_, DONE=> bool_out_, BUSY=> bool_out_, ERROR=> bool_out_, STATUS=> word_out_, MB_DATA_PTR:= variant_inout_);</pre>	<p>MB_CLIENT communicates as a Modbus TCP client through the PROFINET connector on the S7-1200 CPU. No additional communication hardware module is required.</p> <p>MB_CLIENT can make a client-server connection, send a Modbus function request, receive a response, and control the disconnection from a Modbus TCP server.</p>

Descripción de terminales de conexión de bloque de comunicación Modbus cliente

(SIEMENS, 2014)

Tabla 11. Tipo de datos de parametrización

Parameter and type		Data type	Description
REQ	In	Bool	FALSE = No Modbus communication request TRUE = Request to communicate with a Modbus TCP server
DISCONNECT	IN	Bool	The DISCONNECT parameter allows your program to control connection and disconnection with a Modbus server device. If DISCONNECT = 0 and a connection does not exist, then MB_CLIENT attempts to make a connection to the assigned IP address and port number. If DISCONNECT = 1 and a connection exists, then a disconnect operation is attempted. Whenever this input is enabled, no other operation will be attempted.
CONNECT_ID	IN	UInt	The CONNECT_ID parameter must uniquely identify each connection within the PLC. Each unique instance of the MB_CLIENT or MB_SERVER instruction must contain a unique CONNECT_ID parameter.
IP_OCTET_1	IN	USInt	Modbus TCP server IP address: Octet 1 8-bit part of the 32-bit IPv4 IP address of the Modbus TCPserver to which the client will connect and communicate using the Modbus TCP protocol.
IP_OCTET_2	IN	USInt	Modbus TCP server IP address: Octet 2
IP_OCTET_3	IN	USInt	Modbus TCP server IP address: Octet 3
IP_OCTET_4	IN	USInt	Modbus TCP server IP address: Octet 4
IP_PORT	IN	UInt	Default value = 502: The IP port number of the server to which the client will attempt to connect and ultimately communicate using the TCP/IP protocol.
Parameter and type		Data type	Description
MB_MODE	IN	USInt	Mode Selection: Assigns the type of request (read, write, or diagnostic). See the Modbus functions table below for details.
MB_DATA_ADDR	IN	UDInt	Modbus starting Address: Assigns the starting address of the data to be accessed by MB_CLIENT. See the Modbus functions table below for valid addresses.
MB_DATA_LEN	IN	UInt	Modbus data Length: Assigns the number of bits or words to be accessed in this request. See the Modbus functions table below for valid lengths
MB_DATA_PTR	IN_OUT	Variant	Pointer to the Modbus data register: The register buffers data going to or coming from a Modbus server. The pointer must assign a standard global DB or a M memory address.
DONE	OUT	Bool	The DONE bit is TRUE for one scan, after the last request was completed with no error.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 - No MB_CLIENT operation in progress 1 - MB_CLIENT operation in progress
ERROR	OUT	Bool	The ERROR bit is TRUE for one scan, after the MB_CLIENT execution was terminated with an error. The error code value at the STATUS parameter is valid only during the single cycle where ERROR = TRUE.
STATUS	OUT	Word	Execution condition code

Descripción de tipo de parámetros (SIEMENS, 2014)

MB_SERVER: Conecta bajo pedido a un cliente Modbus TCP, recibe mensajes Modbus y envía respuesta. Permite la implementación de códigos de funciona Modbus (1, 2, 4, 5 y 15) para leer o escribir bits y palabras directamente en las imágenes de proceso de entrada y salida. Para códigos de funciona de transferencia de datos (3, 6 y 16), el parámetro MB_HOLD_REG debe ser definido como un tipo de dato más largo que un byte. Comunica como servidor Modbus TCP a través de un enlace PROFINET de la CPU S7 – 1200, para poder utilizar esta instrucción no es necesario ningún

modulo hardware adicional, en la tabla 12 se muestra la descripción del bloque mencionado y en la tabla 13 se detalle cada función del bloque y el tipo de dato admitido por este bloque.

Tabla 12. Bloque de Instrucción MB_SERVER

LAD / FBD	SCL	Description
	<pre>"MB_SERVER_DB" (DISCONNECT:=_bool_in_, CONNECT_ID:=_uint_in_, IP_PORT:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER communicates as a Modbus TCP server through the PROFINET connector on the S7-1200 CPU. No additional communication hardware module is required.</p> <p>MB_SERVER can accept a request to connect with Modbus TCP client, receive a Modbus function request, and send a response message.</p>

Terminales de conexión de bloque de comunicación Modbus cliente (SIEMENS, 2014)

Tabla 13. Tipo de datos de parametrización

Parameter and type	Data type	Description
DISCONNECT IN	Bool	MB_SERVER attempts to make a "passive" connection with a partner device. This means that the server is passively listening for a TCP connection request from any requesting IP address. If DISCONNECT = 0 and a connection does not exist, then a passive connection can be initiated. If DISCONNECT = 1 and a connection exists, then a disconnect operation is initiated. This allows your program to control when a connection is accepted. Whenever this input is enabled, no other operation will be attempted.
CONNECT_ID IN	UInt	CONNECT_ID uniquely identifies each connection within the PLC. Each unique instance of the MB_CLIENT or MB_SERVER instruction must contain a unique CONNECT_ID parameter.
Parameter and type	Data type	Description
IP_PORT IN	UInt	Default value = 502: The IP port number that identifies the IP port that will be monitored for a connection request from a Modbus client. These TCP port numbers are not allowed for a MB_SERVER passive connection: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963, and 34964.
MB_HOLD_REG IN_OUT	Variant	Pointer to the MB_SERVER Modbus holding register: The holding register must either be a standard global DB or a M memory address. This memory area is used to hold the values a Modbus client is allowed to access using Modbus register functions 3 (read), 6 (write), and 16 (write).
NDR OUT	Bool	New Data Ready: 0 = No new data, 1 = Indicates that new data has been written by a Modbus client
DR OUT	Bool	Data Read: 0 = No data read, 1 = Indicates that data has been read by a Modbus client.
ERROR OUT	Bool	The ERROR bit is TRUE for one scan, after MB_SERVER execution was terminated with an error. The error code value at the STATUS parameter is valid only during the single cycle where ERROR = TRUE.
STATUS OUT	Word	Execution condition code

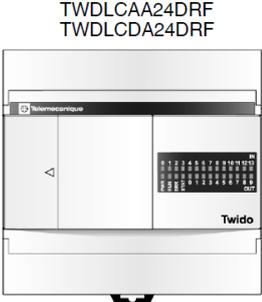
Descripción de tipo de parámetros (SIEMENS, 2014)

2.4.3. TWIDO TWDLCAA24DRF

2.4.3.1. Características del PLC

Esta gama de controlador programable es de tipo compacto, ofrece una solución “todo en uno” con unas dimensiones reducidas. Esta base compacta utiliza una alimentación de corriente alterna comprendida entre 100 y 240VAC. Este autómata cuenta con una capacidad de 14 entradas digitales y 10 salidas digitales las cuales sirven para conectar actuadores del tipo “todo o nada”, adicional se puede instalar un puerto serie para comunicación con protocolo Modbus RTU, el cual es el tipo de comunicación predefinido de este módulo compacto, en la tabla 14 se detalla las características de este controlador (Schneider Electric, 2011).

Tabla 14. Características de controlador TWIDO

Tipo de controlador	Ilustración
Controlador compacto de 24 E/S: <ul style="list-style-type: none">● 14 entradas digitales y 10 salidas de relé● 2 potenciómetros analógicos● 1 puerto serie integrado● 1 slot para un puerto serie adicional● Admite hasta 4 módulos de E/S de ampliación● Admite hasta dos módulos de interfase del bus AS-Interface V2● Admite un módulo master de interfase del bus de campo CANopen:● Admite un cartucho opcional (de reloj de tiempo real o de memoria, sólo 32 KB)● Admite un módulo de monitor de operación opcional● Admite un módulo de interfase Ethernet ConneXium TwidoPort	

Detalle conexiones y puertos de Autómata TWIDO (SCHNEIDER ELECTRIC, 2011)

Los controladores Twido compactos están formados por los componentes indicados en la figura 19, teniendo en cuenta que hay pequeñas diferencias entre modelos de controlador.

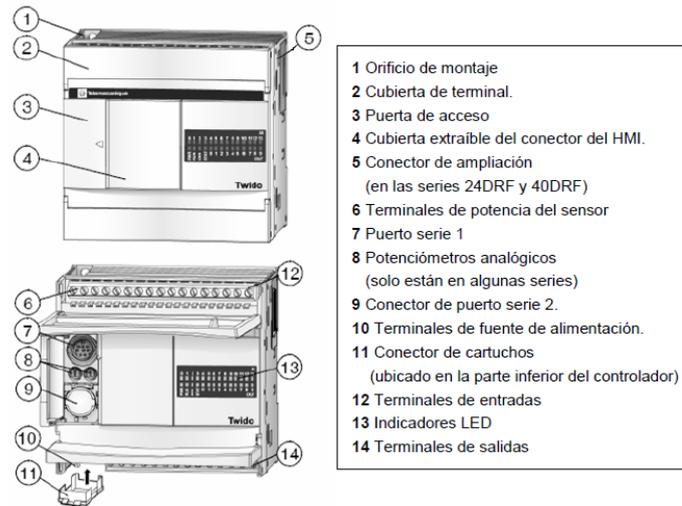


Figura 17. Partes de un Controlador Compacto Twido (SCHNEIDER ELECTRIC, 2011)

Los controladores Twido disponen de un puerto serie, o de un segundo puerto opcional, que se utiliza para servicios en tiempo real o de administración de sistemas. Los servicios en tiempo real proporcionan funciones de distribución de datos para intercambiar datos con dispositivos de E/S, así como funciones de administración para comunicarse con dispositivos externos. Los servicios de administración de sistemas controlan y configuran el controlador por medio de TwidoSoft. Cada puerto serie se utiliza para cualquiera de estos servicios, pero sólo el puerto serie 1 es válido para comunicarse con TwidoSoft, para poder utilizar estos servicios, existen tres protocolos implícitos disponibles en cada controlador:

- **Remote Link (Conexión remota):** permite realizar una comunicación entre autómatas Twido vía RS-485, utilizado para ver **E/S a distancia** (sin programa en las CPUs deportadas) o para red de Twidos con programa, con una longitud máxima de **200 m** y hasta **8 equipos** en una red (**maestro + 7 esclavos**), se muestra una referencia de conexión en la figura 20.

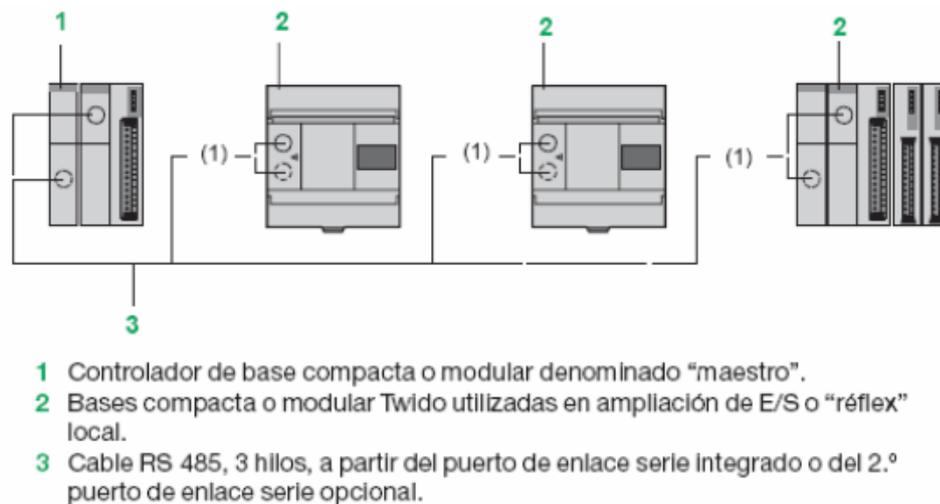


Figura 18. Ejemplo de conexión remota (SCHNEIDER ELECTRIC, 2008)

- **ASCII:** permite comunicar el autómata, vía **RS-485 y RS-232**, con un gran número de dispositivos: **impresoras** (para la impresión periódica de reportes de producción), lectores de códigos de barras y módems.
- **Modbus:** comunicación Modbus Maestro/Esclavo por ambos puertos (RS485 o 232), permite conectar Twido a un gran número de equipos industriales, basado en mensajería aperiódica entre equipos.

2.4.3.2. Software de programación

(Schneider Electric, 2011) TwidoSuite es el primer software que está organizado según el ciclo de desarrollo del proyecto. La navegación por el software es tan sencilla que se convierte en innata. TwidoSuite es un entorno de desarrollo gráfico, lleno de funciones para crear, configurar y mantener aplicaciones de automatización para los autómatas programables Twido de Telemecanique. TwidoSuite permite crear programas con distintos tipos de lenguaje, después de transferir la aplicación para que se ejecute en un autómata.



Figura 19. Plataforma TwidoSuite (SCHNEIDER ELECTRIC, 2008)

(Schneider Electric, 2008) TwidoSuite es un programa basado en Windows de 32 bits para un ordenador personal (PC) que se ejecuta en los sistemas operativos Microsoft Windows 2000 y XP Professional. Las principales características del software TwidoSuite son:

- Interface de usuario intuitiva y orientada a proyectos.
- Diseño de software sin menús. Las tareas y funciones del paso seleccionado de un proyecto siempre se encuentran visibles.
- Soporte de programación y configuración.
- Comunicación con el autómeta.
- Ayuda de primera mano acerca del nivel de tareas que ofrece enlaces relevantes a la ayuda en línea.
- TwidoSuite es un software fácil de usar que necesita poco o nada de aprendizaje. Este software tiene por objeto reducir de forma significativa el tiempo de desarrollo de los proyectos simplificando todas las intervenciones.

Los lenguajes de programación en el software TwidSuite son el KOP y por sentencias de programación, de lo cual se conoce previamente la simbología en ladder y

sentencias usadas para la programación que son idénticas en los distintos software de autómatas programables.

2.4.3.3. Macros de comunicación

Schneider Electric (2011) indica que se trata de una macro genérica destinada a implementar peticiones de comunicación sobre un puerto Modbus o Modbus TCP/IP. Esta permite configurar hasta 32 instancias. Cada instancia que se configure se ha de vincular a un puerto configurado en Modbus y se le ha de asignar el número de nodo esclavo al que se dirigirán las peticiones realizadas a través de las funciones de la macro. Estas macros se utilizan solo cuando el PLC se configura en modo maestro, ya que estas funciones setean los valores predeterminados de la trama según la función Modbus que se use, tales como los datos de la tabla de control, función Modbus, la identificación del esclavo y valores offset y solo se completa con información de los datos de envío, dirección de lectura/escritura y envían las solicitudes de peticiones.

Para la habilitación y configuración de la macro se tiene que seguir un procedimiento, el cual hemos descrito en el anexo 1 de este documento. Casanova (2010) indica que las funciones macro que se pueden generar son:

- C_RD1B: para realizar lecturas de 1 bit, los parámetros del bloque se muestra en la tabla 15.

Tabla 15. Macro de lectura de un bit

Símbolo	Descripción
C_RD1B_ADDR	Este símbolo debe inicializarse en el programa de lista/Ladder con la primera dirección de palabra que va a leerse.
C_RD1B_VAL	Al ejecutar la macro, el símbolo se actualiza para contener el valor del primer bit leído.

Sentencias usada para lectura de un bit con macros de comunicación (Marc Casanova, 2010)

- C_RD1W: Realiza lecturas de 1 palabra, se muestra en la tabla 16.

Tabla 16. Macro de lectura de una palabra

Símbolo	Descripción
C_RD1W_ADDR	Este símbolo debe inicializarse en el programa de lista/Ladder con la dirección de palabra que va a leerse.
C_RD1W_VAL	Al ejecutar la macro, el símbolo se actualiza para contener el valor de la primera palabra leída.

Sentencias usada para lectura de una palabra con macros de comunicación (Marc Casanova, 2010)

- C_WR1B: Realiza escrituras de 1 bit, los parámetros del bloque se muestra en la tabla 17.

Tabla 17. Macro de escritura de un bit

Símbolo	Descripción
C_WR1B_ADDR	Este símbolo debe inicializarse en el programa de lista/Ladder con la dirección de bit que va a escribirse.
C_WR1B_VAL	Este símbolo debe inicializarse en el programa de lista/Ladder con el valor de bit que va a escribirse.
C_WR1B_ADDRW	Al ejecutar la macro, el símbolo se actualiza para contener la dirección en la que se ha escrito el bit.
C_WR1B_VALW	Al ejecutar la macro, el símbolo se actualiza para contener el valor del bit escrito.

Sentencias usada para escritura de un bit con macros de comunicación (Marc Casanova, 2010)

- C_WR1W: Realiza escritura de 1 palabra, parámetros del bloque se muestra en la tabla 18.

Tabla 18. Macro de escritura de una palabra

Símbolo	Descripción
C_WR1W_ADDR	Este símbolo debe inicializarse en el programa de lista/Ladder con la dirección de palabra que va a escribirse.
C_WR1W_VAL	Este símbolo debe inicializarse en el programa de lista/Ladder con el valor de palabra que va a escribirse.
C_WR1W_ADDRW	Al ejecutar la macro, el símbolo se actualiza para contener la dirección en la que se ha escrito la palabra.
C_WR1W_VALW	Al ejecutar la macro, el símbolo se actualiza para contener el valor del bit escrito.

Sentencias usadas para escritura de una palabra con macros de comunicación (Marc Casanova, 2010)

- C_RDNW: Realiza lectura de múltiples palabras, parámetros del bloque se muestra en la tabla 19.

Tabla 19. Macro de lectura de N palabras

Símbolo	Descripción
C_RDNW_ADDR1	Este símbolo debe inicializarse en el programa de lista/Ladder con la primera dirección de palabra que va a leerse.
C_RDNW_VAL1	Al ejecutar la macro, el símbolo se actualiza para contener el valor de la primera palabra leída.

Sentencias usada para lectura de N palabras con macros de comunicación (Marc Casanova, 2010)

- C_WRNW: Realiza escritura de múltiples palabras, parámetros del bloque se muestra en la tabla 20.

Tabla 20. Macro de escritura de N palabras

Símbolo	Descripción
C_WRNW_ADDR1	Este símbolo debe inicializarse en el programa de lista/Ladder con la primera dirección de palabra que va a escribirse.
C_WRNW_VAL1	Al ejecutar la macro, el símbolo se actualiza para contener el valor de la primera palabra escrita.

Sentencias usada para escritura de N palabras mediante macros de comunicación (Marc Casanova, 2010)

Los macros de comunicación ofrecen una manera sencilla de establecer la enlace entre dispositivos modbus ya que no se requiere de generar los bloques de datos del trama y tan solo se hace uso de bloques preestablecidos para cada función.

2.5. Red de comunicación Modbus – TCP

Modbus/TCP es un protocolo de comunicación diseñado para permitir que un equipo industrial tal como PLCs, motores, sensores, y otros tipos de dispositivos físicos de entrada/salida pueda comunicarse sobre una red a nivel de célula. Modbus/TCP fue introducido por Schneider Automation como una variante de la familia MODBUS ampliamente usada, destinada para la supervisión y el control de equipo de automatización. El protocolo cubre el uso de mensajes MODBUS en un entorno Intranet o Internet usando los protocolos TCP/IP.

2.5.1. Encapsulamiento de trama Modbus sobre TCP

Modbus/TCP básicamente embebe un marco MODBUS dentro de un marco TCP en una manera simple como es mostrado en la figura 22.



Figura 20. Esquema encapsulado de Modbus TCP (Andrés Ruiz, 2002)

Ruiz (2002) describe la forma general de encapsulación de una solicitud o respuesta MODBUS cuando es llevada sobre una red Modbus/TCP. Es importante anotar que la estructura del cuerpo de la solicitud y respuesta, desde el código de función hasta el fin de la porción de datos, tiene exactamente la misma disposición y significado como en las otras variantes MODBUS, las únicas diferencias en esos otros casos son la especificación de los delimitadores inicial y final del mensaje, el patrón de chequeo de error y la interpretación de la dirección.

Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502. Las solicitudes normalmente son enviadas en forma half-duplex sobre una conexión dada, es decir, no hay beneficio en enviar solicitudes adicionales sobre una única conexión mientras una respuesta está pendiente. 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.

Los mensajes de solicitud y respuesta en Modbus/TCP poseen un prefijo o encabezado compuesto por seis bytes como se aprecia en la tabla 21.

Tabla 21. Estructura del prefijo Modbus/TCP

ref	ref	00	00	00	len
-----	-----	----	----	----	-----

Trama de Modbus TCP (Andrés Ruiz, 2002)

El “ref ref” anterior son los dos bytes del campo “referencia de transacción”, un número que no tiene valor en el servidor pero son copiados literalmente desde la solicitud a la respuesta a conveniencia del cliente. Este campo se utiliza para que un cliente Modbus/TCP pueda establecer simultáneamente múltiples conexiones con diferentes servidores y pueda identificar cada una de las transacciones, el tercer y cuarto campo del prefijo representa el “identificador de protocolo”, un número el cual debe ser establecido a cero.

El “len” especifica el número de bytes que siguen. La longitud es una cantidad de dos bytes, pero el byte alto se establece a cero ya que los mensajes son más pequeños que 256, de esta forma un mensaje Modbus/TCP completo posee una estructura como se muestra en la tabla 22.

Tabla 22. Estructura de mensajes en Modbus/TCP

Posición del Byte	Significado
Byte 0	Identificador de transacción. Copiado por el servidor – normalmente 0.
Byte 1	Identificador de transacción. Copiado por el servidor – normalmente 0.
Byte 2	Identificador de protocolo = 0.
Byte 3	Identificador de protocolo = 0.
Byte 4	Campo de longitud (byte alto) = 0. Ya que los mensajes son menores a 256.
Byte 5	Campo de longitud (byte bajo). Número de bytes siguientes.
Byte 6	Identificador de unidad (previamente “dirección esclavo”).
Byte 7	Código de función MODBUS.
Byte 8 y más	Los datos necesarios.

Identificación de bytes en trama Modbus TCP (Andrés Ruiz, 2002)

2.5.2. Ventajas de comunicación Modbus – TCP

Andrés Ruiz, 2002 menciona como ventajas:

- Es escalable en complejidad. Un dispositivo el cual tiene solo un propósito simple necesita solo implementar uno o dos tipos de mensaje.
- 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 computador o microprocesador con una pila de protocolos TCP/IP puede usar Modbus/TCP.
- Puede ser usado para comunicar 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 la capacidad del sistema operativo del computador para comunicarse. Altas ratas de transmisión son fáciles de lograr sobre una estación única, y cualquier red puede ser construida para lograr tiempos de respuesta garantizados en el rango de milisegundos.

3. Marco metodológico

3.1. Descripción de hardware de maletas didácticas con PLC TWIDO

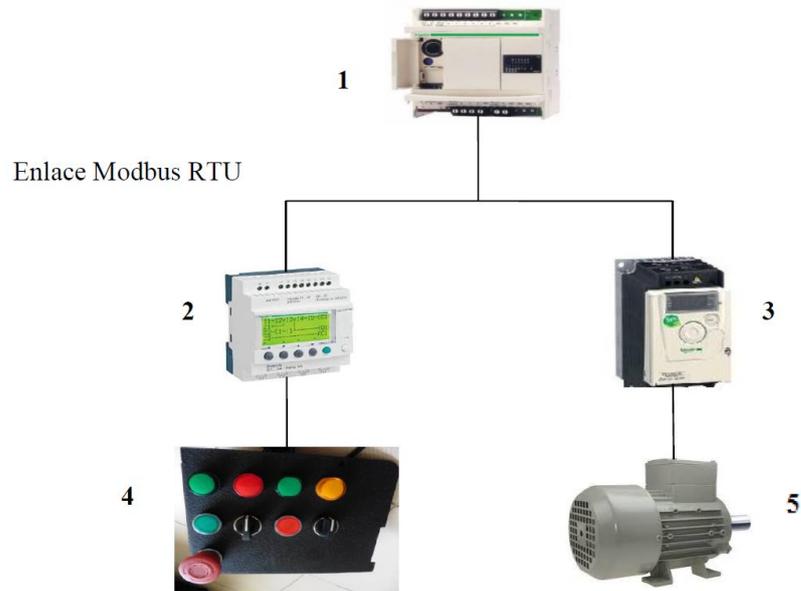


Figura 21. Esquema de maletas didácticas SCHNEIDER, (J. Gallegos y E. Delgado, 2015)

Los elementos conectados en las maletas didácticas se ilustran en la figura 23 y se describen brevemente de acuerdo a su numeración:

- 1. Autómata TWIDO:** Esta gama de controlador programable es de tipo compacto, ofrece una solución “todo en uno” con unas dimensiones reducidas. Cuenta con la capacidad de 14 entradas digitales y 10 salidas digitales las cuales sirven para realizar prácticas con actuadores del tipo “todo o nada”.
- 2. Zelio Logic:** Incluyendo 4 entradas digitales, 2 entradas análogas de 0-10V y 4 salidas, además posee reloj. Este equipo no es utilizado para el presente proyecto.
- 3. VDF ATV132:** Variador de frecuencia de la marca Schneider Electric que cumple como características de capacidad de 1HP (0.75KW) – Tensión de alimentación 220/240VAC – Frecuencia de alimentación 50/60HZ –

Monofásico – Corriente de línea 7.5/8.9 A – Disipación de potencia de 60W – Puerto de comunicación Modbus RTU.

4. **Tablero de control:** pulsadores, focos y switch para datos de entrada del PLC.
5. **Motor trifásico:** Se dispone de un motor de 0.5HP con alimentación trifásica y tensión a 220VAC, controlado por medio del variador de frecuencia.

3.2. Descripción del hardware de maleta didáctica con S7 – 1200

- **Autómata Siemens:** Controlador lógico programable del tipo compacto, el cual ofrece conectividad con una variedad de equipos para diversas tareas de automatización. Está CPU cuenta con 14 entradas y 10 salidas digitales, 2 entradas analógicas ideal para prácticas de simulación de procesos industriales, el diagrama de este CPU se muestra en la figura 24.

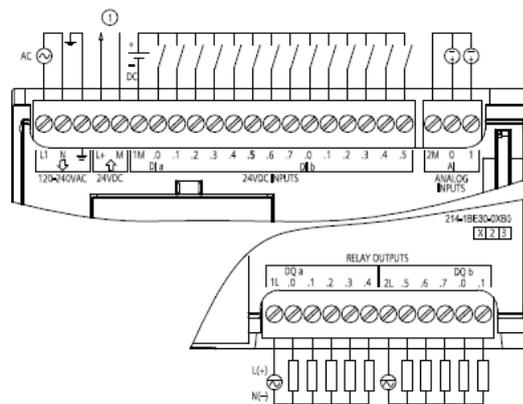


Figura 22. Diagrama general de conexiones de CPU 1214 AC/DC/RLY (Siemens, 2009)

- **KTP600 HMI:** Interfaz humano – maquina, para el desarrollo de prácticas con visualización de procesos en ejecución, disponible con un puerto RJ45 con comunicación Profibus, las partes de este con las que consta este interfaz se muestran en la figura 25.

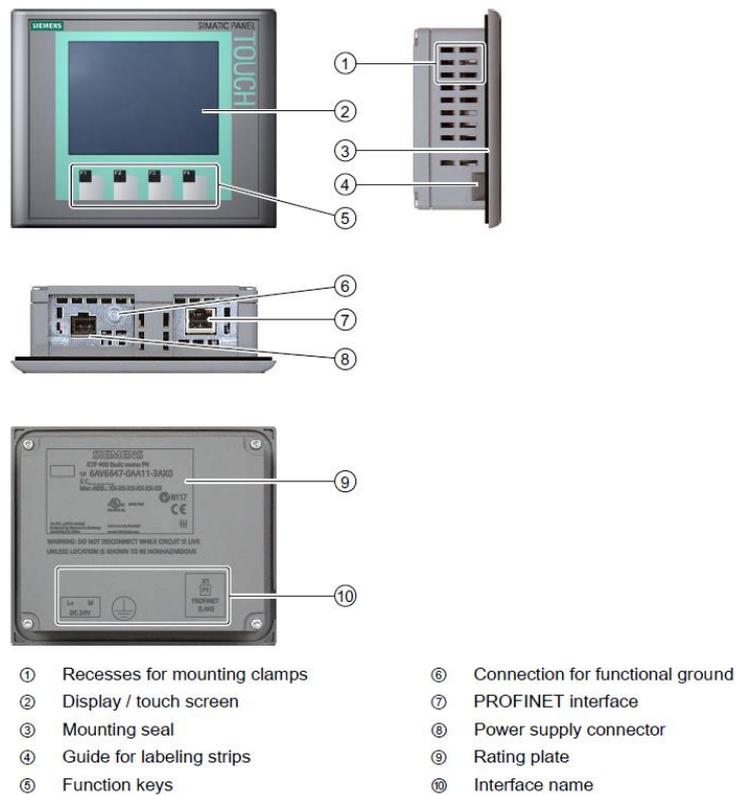


Figura 23. Características HMI Siemens KTP600 (Siemens, 2008)

3.3. Descripción de repartidor Modbus LU9GC3

Dispositivo Hub Modbus para conexión de equipos serie, la conexión interna del equipo se muestra en la figura 26 y las características del equipo se enlistan:

- Comunicación en RS232 y RS485.
- 8 puertos de conexión tipo RJ45.
- 1 Terminal de conexión tipo bornera.
- 1 puerto de entrada y salida de datos.
- Voltaje de alimentación de 24VDC.
- Soporta un máximo de corriente de 32A.
- Fácil instalación en montaje de rack.

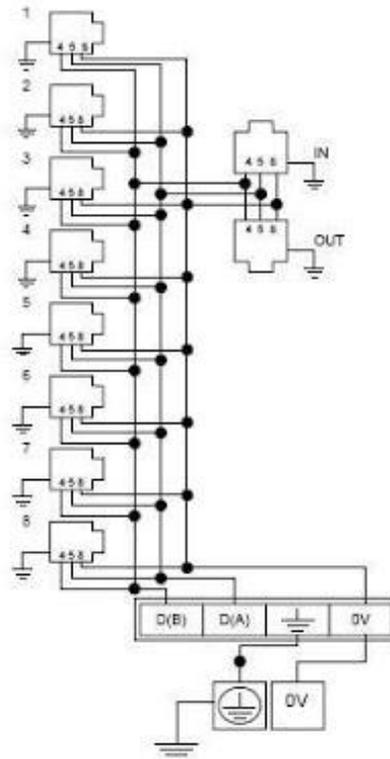


Figura 24. Esquema interno de conexión de repartidor (Schneider, 2012)

3.4. Descripción de Gateway y equipos de comunicación inalámbrica

La pasarela ETG100 es un dispositivo de comunicación que ofrece conectividad entre Ethernet (Modbus TCP/IP) y dispositivos en línea serie, con lo cual permite que los clientes Modbus TCP/IP accedan a la información desde dispositivos esclavos en serie, también permite que los dispositivos maestros serie accedan a información desde dispositivos esclavos distribuidos en una red Ethernet (Schneider, 2012). En la configuración dispuesta en este proyecto, la pasarela funciona en modo maestro y cumple la función de un maestro local para los dispositivos series conectados, lo cual permite direccionar el trama proveniente del cliente Modbus TCP a los esclavos RTU, mientras que para el envío de datos entre la estación cliente y los esclavos se dispuso un Router D-Link propiamente configurado en el rango IP que se va a trabajar. Las

respectivas configuraciones de estos equipos se encuentran en el anexo 2 de este documento, refiérase a ellas en caso de querer mejorar el proyecto.

3.5. Diseño y ensamble de modulo convertidor Modbus TCP

El diseño del módulo consta de una caja eléctrica de control, en donde se fijan los equipos de conexión y el convertidor en rieles de montaje. La conexión entre los equipos se realiza con cable de instrumentación AWG #18 para los datos y para la alimentación se conecta la fuente a un puerto hembra de 110VAC mediante cable AWG #14, todas las conexiones están protegidas con protectores espirales y aislantes térmicos en las uniones.

3.5.1. Diagrama de conexión de componentes del módulo

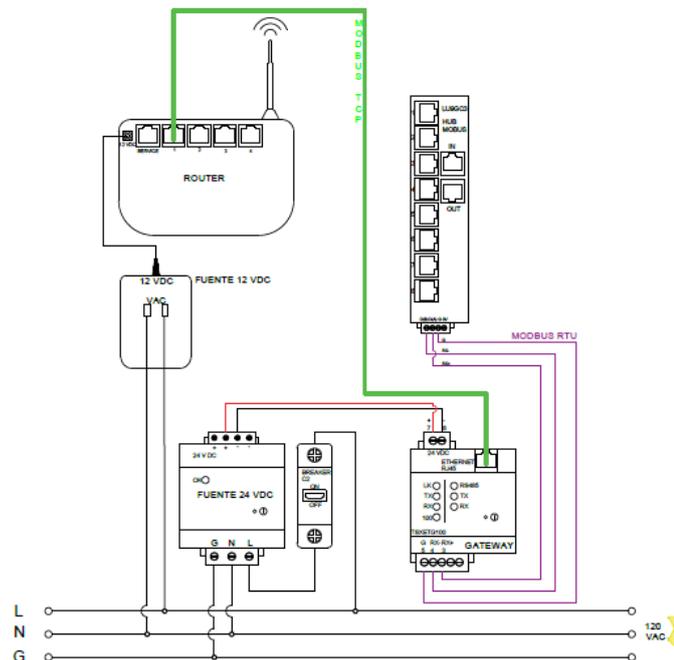


Figura 25. Diagrama de conexión de equipos (Autores, 2016)

3.5.2. Estructura de carcasa y montaje de equipos

La carcasa del módulo convertidor es una caja de control usualmente utilizada para la instalación de breake eléctrico. La elección de este tipo de caja metálica se debe al

poco espacio utilizado por los equipos y sus conexiones, protección por golpes o caídas y aislamiento eléctrico, en donde se montan los equipos en rieles fijados en la lámina de fondo de la caja, las medidas de la caja metálica son 30X40X15cm, un espacio suficiente y necesario para la instalación de los equipos.



Figura 26. Caja metálica para modulo convertidor Modbus TCP (Autores, 2016)

El montaje de los equipos se hace mediante rieles y todos los cables de conexiones de los equipos, son protegidos con protector espiral. El modulo presenta un puerto de conexión para la energización de la fuente de poder y tiene una perforación en la parte superior por donde sale la antena del Router, para que se pueda captar la señal al Router del PLC cliente.

La caja metálica tiene un doble fondo del cual la tapa interna es removible y se colocó para la protección de las interconexiones de los equipos y evitar que sean manipulados de una manera incorrecta. El cubrimiento de los equipos y sus conexiones se hace mediante una plancha de acrílico transparente, la cual protege la manipulación de los

cables de conexión de los equipos y evitan el movimiento innecesarios de estos para evitar algún problema de corto circuito.

3.5.3. Cableado de Gateway, Router y fuente de alimentación

En la figura 28 se muestra la conexión realizada de los equipos, de acuerdo al diagrama esquemático mostrado en la figura 27.



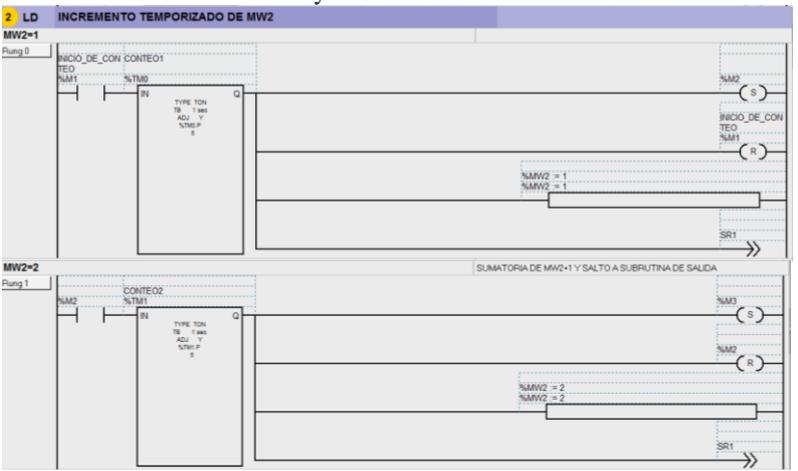
Figura 27. Conexión de equipos de módulo Modbus TCP (Autores, 2016)

En la figura 27 se muestra la conexión de los equipos y sus cables debidamente protegidos, asegurados y fijados, para evitar su desconexión o problemas de conexión. Los terminales del conector de alimentación de entrada se encuentran protegidos con un aislante térmico, esto para evitar corto circuitos o daños mayores.

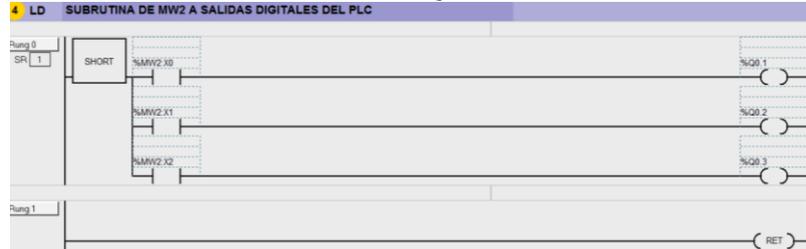
El módulo está protegido con un breaker de 3A/230VAC, la dimensión de este breaker es de suficiente capacidad debido al bajo consumo de estos equipos. La ubicación del Router es de forma tal que la antena WiFi, sobresale de la caja del módulo para evitar pérdidas de señales con el Router del cliente. Para la alimentación de este equipo, se conecta con el conector principal y se protege el cable con protección espiral y los terminales con caucho sobre temperatura.

4. Guía De Prácticas De Laboratorio

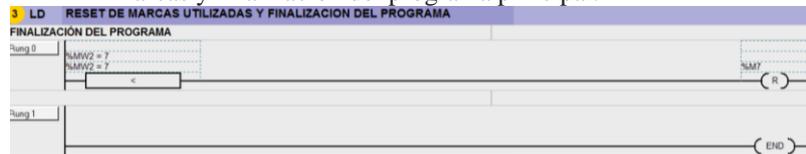
4.1. Práctica 1

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: Ing. Electrónica		ASIGNATURA: Informática Industrial/ Redes III	
NRO. PRÁCTICA:		1	TÍTULO PRÁCTICA: Activación de secuencia de luces mediante orden del Cliente Modbus TCP en un servidor Twido.
OBJETIVOS: <ul style="list-style-type: none"> • Entrenar al estudiante con los bloques Modbus TCP Siemens. • Comprender el mapeo de registro de ambos PLC's. • Enseñar un aplicativo sencillo de Modbus TCP. • Configurar los equipos necesarios para la práctica requerida. • Introducir al estudiante al protocolo Modbus TCP. 			
INSTRUCCIONES:		1. Introducción de funcionamiento del programa. El programa desarrolla una comunicación simple entre el maestro y un esclavo, el PLC maestro envía datos de activación para que el esclavo de inicio a una secuencia temporizada de encendido y apagado de indicadores luminosos que representan el valor binario del 0 al 7. El PLC maestro realiza un conteo hasta el valor 15 y este valor se transfiere al esclavo y se compara en un bloque para dar inicio al programa de temporización en donde cada cierto tiempo se muestra en los indicadores luminosos los valores binarios del 0 al 7.	
		2. Configuración y programación del dispositivo esclavo. <ul style="list-style-type: none"> • Configuración de adaptador serie TWDNAC485T • Dirección RTU del esclavo Twido (1) y parámetros de comunicación. • En el primer LD se ubica el bit de memoria de la palabra que será escrita (MW0:X0) para el arranque, seguidamente se pone un bloque de comparación que se accionara cuando la palabra MW1 sea igual a 15, esto sucederá por medio de un contador del maestro, lo que hará que se active una marca(M1).  <ul style="list-style-type: none"> • En el LD 2 se programa la rutina de incremento progresivo de la palabra MW2, al pasar un tiempo de 5 segundos, el valor de MW2 aumenta de en uno y se muestra en indicadores luminosos. 	

- El proceso de incremento es fijo, es decir el valor que adquiere MW2 es un valor establecido después de cada temporizador.
- Para la salida en los indicadores luminosos, se programa una subrutina de la cual se setean los 3 bits utilizados de la palabra MW2 y se conectan cada uno con una salida correspondiente.
- Después de finalizar el conteo cada temporizador, se activa un salto a la subrutina de salida Q.



- Previamente de la subrutina, esta función necesita que el programa principal finalice, por lo que en el LD 3 se programa el reset de las marcas y finalización del programa principal.



3. Configura y programación PLC cliente.

- En la configuración del PLC se activa las señales de marca del sistema (%MB1) y marcas de ciclo (%MB0).



- Se crean bloques de programación para el direccionamiento de los datos escritos, leídos y monitoreados en el servidor, también un bloque de datos de estado de los MB_Client utilizados.
- En el bloque de transferencia (FC) se programan las sentencias de control para el envío de datos de activación de la secuencia de incremento en el servidor.

- En el bloque de datos denominado HOLDING REGISTER se crean las variables que serán usadas como punteros de datos de los bloques MB_Client.

HOLDING REGISTER				
	Nombre	Tipo de datos	Offset	Valor de arranq...
1	Static			
2	ARRANQUE ESCRITURA_MWO	Word	0.0	16#0
3	MW1 SERVIDOR ESCRITURA	Word	2.0	16#0
4	MW2 SERVIDOR ESCRITURA	Word	4.0	16#0
5	MWO SERVIDOR LECTURA	Word	6.0	16#0
6	MW1 SERVIDOR LECTURA	Word	8.0	16#0
7	MW2 SERVIDOR LECTURA	Word	10.0	16#0

- También se añade un bloque de datos de estados en donde se nombran variables que monitorean el estado de la comunicación de los bloques MB_Client.

DATOS DE ESTADO				
	Nombre	Tipo de datos	Offset	Valor de arranq...
1	Static			
2	STATUS ESCRITURA	Word	0.0	16#0
3	STATUS LECTURA	Word	2.0	16#0
4	STATUS DE CONTEO	Word	4.0	16#0
5	INDICADORES_ESCRIT...	Array [0..2] of Bool	6.0	
6	INDICADORES_ESC...	Bool	0.0	false
7	INDICADORES_ESC...	Bool	0.1	false
8	INDICADORES_ESC...	Bool	0.2	false
9	INDICADORES_LECTURA	Array [0..2] ...	8.0	
10	INDICADORES_LEC...	Bool	0.0	false
11	INDICADORES_LEC...	Bool	0.1	false
12	INDICADORES_LEC...	Bool	0.2	false
13	STATUS FLANCO	Word	10.0	16#0

- Por último se muestra el programa principal, en donde se activa el bloque de transferencia (FC) y en los segmentos posteriores se programan las instrucciones MB_Cliente de lectura y escritura con los respectivos datos del servidor, puerto, dirección de memoria, cantidad de datos y puntero de datos.

ACTIVIDADES POR DESARROLLAR

- Dar seguimiento en el bloque HOLDING REGISTER de los valores escritos y leídos por el cliente.
- Enviar dato booleano al servidor, ya que se está enviando una valor tipo Word.
- Cambiar el direccionamiento del bloque Mb_Client para verificar que el valor es escrito y leído en la dirección que se programe.

RESULTADO(S) OBTENIDO(S):

Valores escritos en la variable del PLC servidor, monitoreado desde el OB del cliente, en el mismo bloque se verá el valor escrito y leído., en el primer grafico se ve los valores iniciales del servidor.

HOLDING REGISTER					
	Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
	Static				
	ARRANQUE ESCRITUR...	Word	0.0	16#0	16#0000
	MW1 SERVIDOR ESCRI...	Word	2.0	16#0	16#0000
	MW2 SERVIDOR ESCRI...	Word	4.0	16#0	16#0000
	MWO SERVIDOR LECTU...	Word	6.0	16#0	16#0000
	MW1 SERVIDOR LECTU...	Word	8.0	16#0	16#0000
	MW2 SERVIDOR LECTU...	Word	10.0	16#0	16#0000

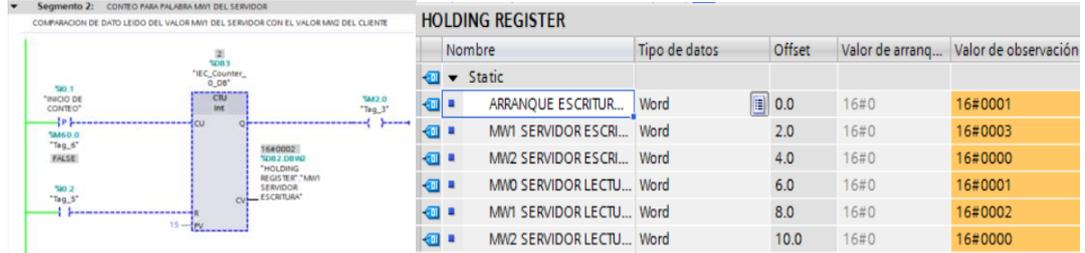
Valor de arranque enviado desde el cliente y leído en la palabra 6.0 del OB Holding Register

HOLDING REGISTER					
	Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
	Static				
	ARRANQUE ESCRITUR...	Word	0.0	16#0	16#0001
	MW1 SERVIDOR ESCRI...	Word	2.0	16#0	16#0000
	MW2 SERVIDOR ESCRI...	Word	4.0	16#0	16#0000
	MWO SERVIDOR LECTU...	Word	6.0	16#0	16#0001
	MW1 SERVIDOR LECTU...	Word	8.0	16#0	16#0000
	MW2 SERVIDOR LECTU...	Word	10.0	16#0	16#0000

Activación de marca en el PLC servidor.



Pulso de conteo desde el cliente y dato transmitido al servidor.

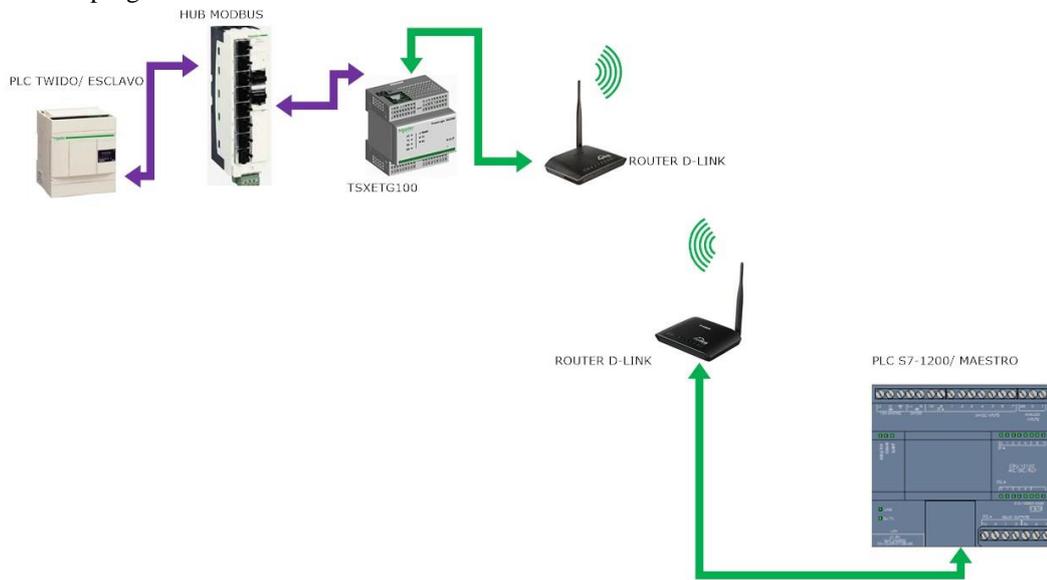


CONCLUSIONES:

El mapeo de los registro de datos de ambos PLC's se realiza en base a los manuales de los equipos para el direccionamiento de las palabras escrititas y/o leídas.
La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

RECOMENDACIONES:

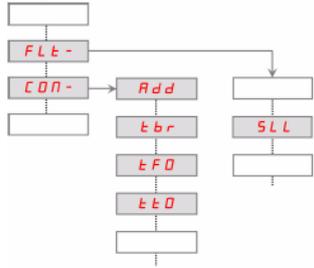
- Verificar el estado de los cables ya que si están en mal estado afectaría la comunicación.
- Verificar que los contactos de los conectores Modbus RTU no presenten sulfatación.
- En el caso de hacer uso de los PLC con conector de espadines, se debe tener en cuenta la correcta posición de conexión de los cables, ya que si se invierten los pines la comunicación será lenta.
- Disponer del siguiente diagrama de red que representa la conexión de los equipos utilizados, no se detalla la dirección de los equipos ya que estas pueden ser modificadas por el programador.



Docente / Técnico Docente: _____

Firma: _____

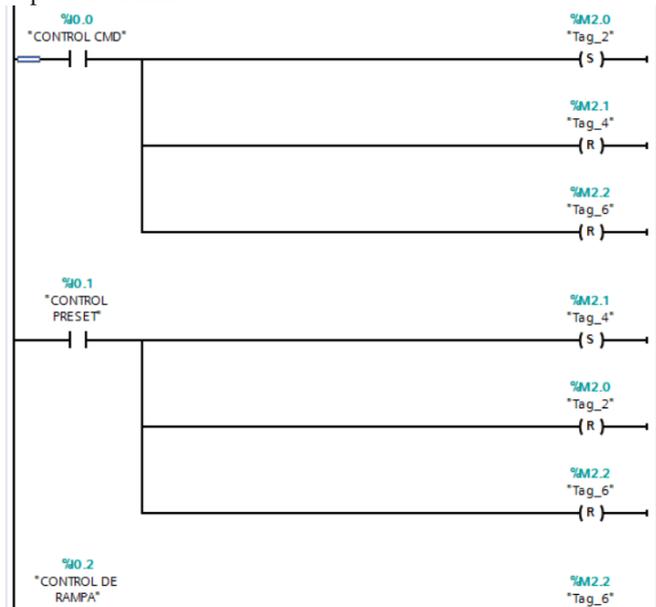
4.2. Práctica 2

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES																																				
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III																																					
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: control de variador atv312 mediante direccionamiento Modbus																																				
OBJETIVOS: <ul style="list-style-type: none"> • Entrenar en el mapeo de palabras hacia ATV312. • Programación de parámetros de velocidad de acuerdo al manual de programación Modbus del equipo ATV312. • Lectura de datos de velocidad, potencia y alimentación de ATV312. • Manejo de palabra de control para arranque del equipo. • Escritura directa entre Cliente y servidor (ATV312). 																																						
INSTRUCCIONES:	<p>1. Introducción de funcionamiento del programa.</p> <p>Esta práctica comunica directamente el maestro con un esclavo variador de frecuencia, en esta aplicación se hace funcionar el variador en 3 formas distintas, pero todas utilizando las mismas palabras de control del VDF (8501 y 8502).</p> <p>La primera forma de seteo del variador es un control de inversión de giro con velocidad fija esto se realiza mediante la activación de un pulsador en el control de mando del maestro.</p> <p>La segunda forma es mediante control variable de velocidad, se puede variar la velocidad mediante una de las entradas variables del PLC maestro.</p> <p>La tercera forma es mediante velocidades seteadas, pulsando el botón correspondiente, el variador arrancara con una velocidad definida.</p> <p>Adicional se incorpora la lectura del estado del variador en donde se visualiza el voltaje al que esta alimentado, la frecuencia de funcionamiento, el modelo del equipo, etc., para interpretar algunos de estos valores se debe disponer del manual del ATV312.</p>																																					
	<p>2. Configuración de direccionamiento RTU del esclavo (1).</p>  <p>Communication configuration - Parameters description</p> <table border="1"> <thead> <tr> <th>Parameter description</th> <th>Range or listed values</th> <th>default</th> <th>Long name</th> <th>Short name</th> <th>@</th> </tr> </thead> <tbody> <tr> <td>Modbus Address A d d</td> <td>1 to 247</td> <td>1</td> <td>[Modbus Address]</td> <td>A d d</td> <td>-</td> </tr> <tr> <td>Modbus baud rate t b r Important: only 19.2 kbps baud rate allows to communicate with the remote display)</td> <td>4.8 kbps 9.6 kbps 19.2 kbps</td> <td>19.2 kbps</td> <td>[Modbus baud rate]</td> <td>4. 8 9. 6 19. 2</td> <td>-</td> </tr> <tr> <td>Modbus format t F D</td> <td>8O1: 8 bits, odd parity, 1 stop bit. 8E1: 8 bits, even parity, 1 stop bit. 8N1: 8 bits, no parity, 1 stop bit 8N2: 8 bits, no parity, 2 stop bits.</td> <td>8E1</td> <td>[8 odd 1 stop] [8 even 1 stop] [8 no 1 stop] [8 no 2 stop]</td> <td>8 O 1 8 E 1 8 n 1 8 n 2</td> <td>-</td> </tr> <tr> <td>Modbus time out t t D</td> <td>Adjustable from 0.1 to 30s</td> <td>10 s</td> <td>[Modbus time out]</td> <td>t t D</td> <td>-</td> </tr> <tr> <td>Modbus fault mgt S L L (This parameter is not a communication management parameter)</td> <td>No action. Freewheel stop Ramp stop Fast stop*</td> <td>-</td> <td>[No] [Freewheel] [Ramp Stop] [Fast Stop]</td> <td>n 0 4 E 5 r n P F S t</td> <td>-</td> </tr> </tbody> </table>			Parameter description	Range or listed values	default	Long name	Short name	@	Modbus Address A d d	1 to 247	1	[Modbus Address]	A d d	-	Modbus baud rate t b r Important: only 19.2 kbps baud rate allows to communicate with the remote display)	4.8 kbps 9.6 kbps 19.2 kbps	19.2 kbps	[Modbus baud rate]	4. 8 9. 6 19. 2	-	Modbus format t F D	8O1: 8 bits, odd parity, 1 stop bit. 8E1: 8 bits, even parity, 1 stop bit. 8N1: 8 bits, no parity, 1 stop bit 8N2: 8 bits, no parity, 2 stop bits.	8E1	[8 odd 1 stop] [8 even 1 stop] [8 no 1 stop] [8 no 2 stop]	8 O 1 8 E 1 8 n 1 8 n 2	-	Modbus time out t t D	Adjustable from 0.1 to 30s	10 s	[Modbus time out]	t t D	-	Modbus fault mgt S L L (This parameter is not a communication management parameter)	No action. Freewheel stop Ramp stop Fast stop*	-	[No] [Freewheel] [Ramp Stop] [Fast Stop]	n 0 4 E 5 r n P F S t
Parameter description	Range or listed values	default	Long name	Short name	@																																	
Modbus Address A d d	1 to 247	1	[Modbus Address]	A d d	-																																	
Modbus baud rate t b r Important: only 19.2 kbps baud rate allows to communicate with the remote display)	4.8 kbps 9.6 kbps 19.2 kbps	19.2 kbps	[Modbus baud rate]	4. 8 9. 6 19. 2	-																																	
Modbus format t F D	8O1: 8 bits, odd parity, 1 stop bit. 8E1: 8 bits, even parity, 1 stop bit. 8N1: 8 bits, no parity, 1 stop bit 8N2: 8 bits, no parity, 2 stop bits.	8E1	[8 odd 1 stop] [8 even 1 stop] [8 no 1 stop] [8 no 2 stop]	8 O 1 8 E 1 8 n 1 8 n 2	-																																	
Modbus time out t t D	Adjustable from 0.1 to 30s	10 s	[Modbus time out]	t t D	-																																	
Modbus fault mgt S L L (This parameter is not a communication management parameter)	No action. Freewheel stop Ramp stop Fast stop*	-	[No] [Freewheel] [Ramp Stop] [Fast Stop]	n 0 4 E 5 r n P F S t	-																																	

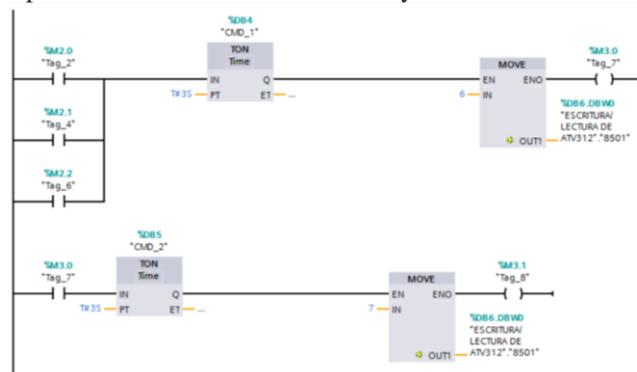
- Las palabras escritas son la 8501(CMD) y 8502(FRECUENCY REFERENCE).
- Las palabras leídas son la 3201(STATUS WORD) y 3202(OUTPUT FRECUENCY).
- La información detallada de la secuencia de arranque de motor y otras funciones se describen en el manual de variables MODBUS del equipo ATV312.

3. Configura y programación PLC cliente.

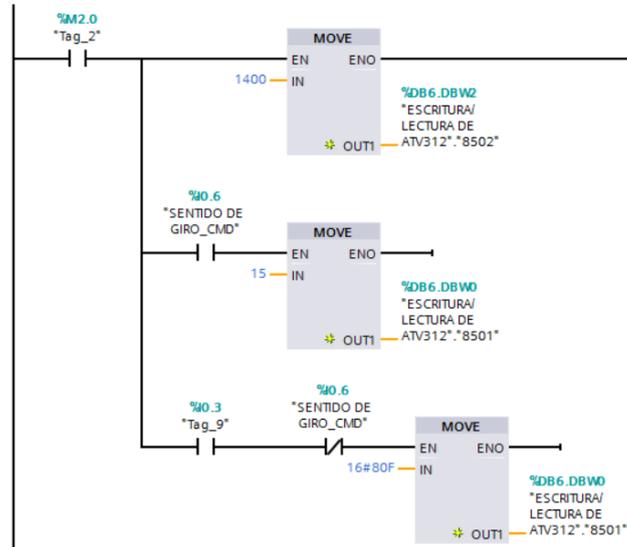
- En la configuración del PLC se activa las señales de marca del sistema (%MB1) y marcas de ciclo (%MB0).
- Se crea bloque de datos para el direccionamiento de los valores escritos, leídos y monitoreados en el servidor, también un bloque de datos de estado de los MB_Client utilizados.
- En el bloque de transferencia (FC) se programan las sentencias de control para la selección de funcionamiento del motor, el cual cuenta con tres estados, velocidad fija e inversión de giro, variación de velocidad con potenciómetro y selección de velocidades pre seteadas, cada una activando marcas SET para la activación de las funciones correspondientes.
- Estas funciones son activadas con el panel de control del cliente siendo %I0.0 para la activación de la primera funcion hasta %I0.2 para la última.



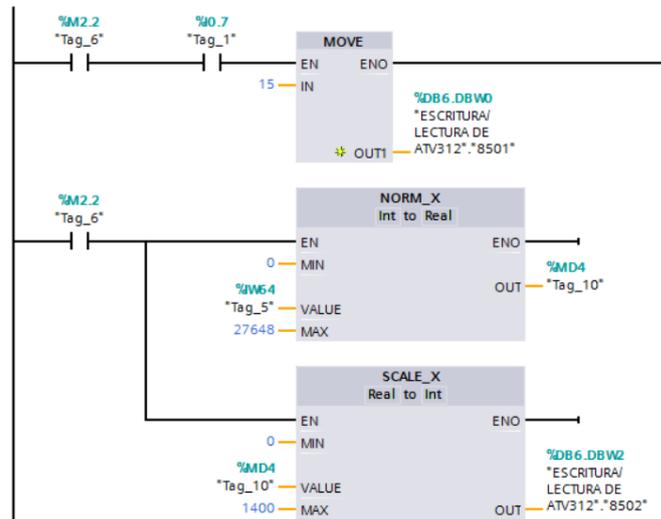
- Se programa los bloques de envío de las palabras de control para poner el variador en modo Stand By.



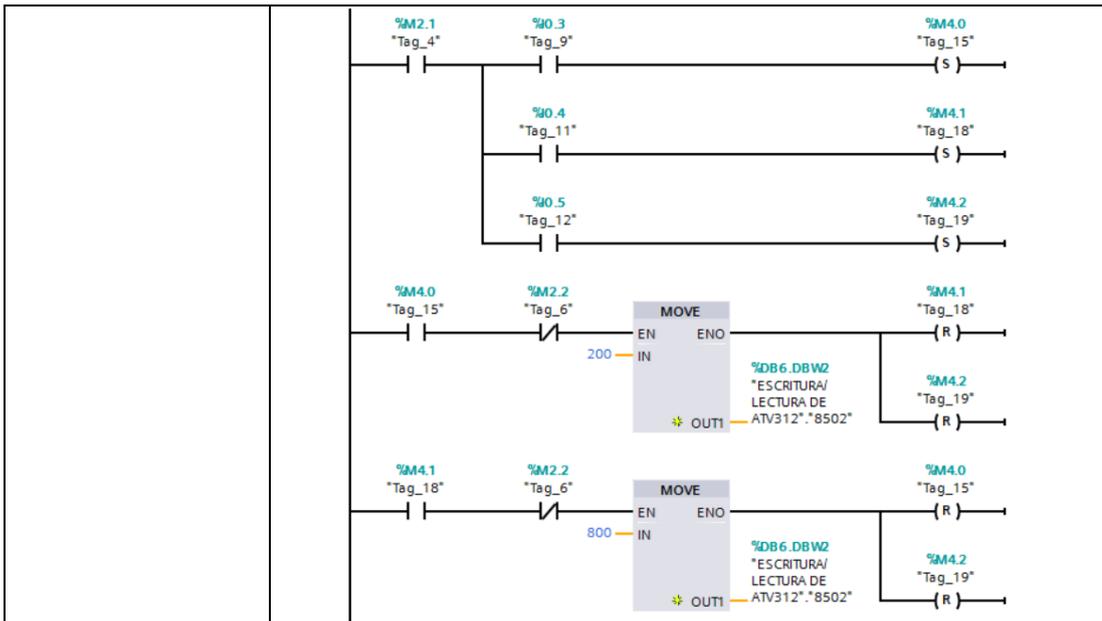
- En la programación de velocidad fija, se envían los parámetros correspondientes como son una velocidad fija y el valor de arranque del motor de acuerdo a la secuencia indicada en el manual.
- Mediante la utilización de un switch (%I0.6) se establece el sentido de giro del motor, la palabras enviada es de acuerdo a las funciones de los bit de la palabra de control CMD.



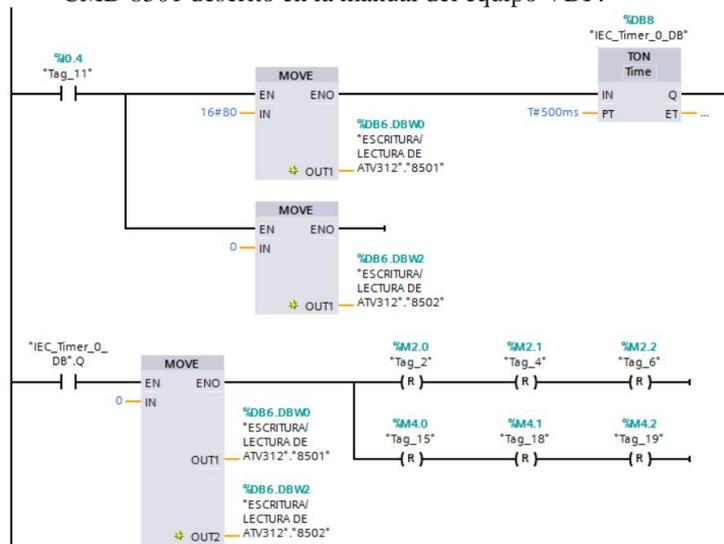
- En la programación de velocidad variable con potenciómetro, se hace la utilización de los bloques NORM y SCALE para el acondicionamiento de la señal %IW64 (0 – 10VDC), entre el rango mínimo y máximo de velocidad permitida por el ATV312.



- En la programación de velocidades pre setadas, se selecciona una velocidad establecida de acuerdo a un pulsador correspondiente (%I0.3-%I0.5), y este valor de velocidad seteada es transferido a la palabra 8502 del variador.



- Adicional se programa un bloque de transferencia para reset de falla, esta función se activa mediante un pulsador de reset y elimina toda falla almacenada en la memoria del ATV312. La palabra que se transfiere es de acuerdo a la activación del bit 7 de la palabra CMD 8501 descrito en la manual del equipo VDF.



ACTIVIDADES POR DESARROLLAR

1. Programar el variador mediante Modbus para funcionamiento por rampas de aceleración y desaceleración.
2. Incluir en la programación otro variador.

RESULTADO(S) OBTENIDO(S):

Los datos obtenidos son los parámetros de funcionamiento del motor: el voltaje de alimentación, velocidad actual, entre otros. La dirección de estos parámetros se detalla en el manual de variables del ATV312.

La siguiente figura muestra las direcciones escritas y leídas en el modo standby del ATV312.

ESCRITURA/LECTURA DE ATV312									
Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación	Remanen...	Accesible d...	Visible en ...	Valor de a...	
Static									
8501	Word	0.0	16#0	16#0007		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8502	Word	2.0	16#0	16#0578		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3201	Word	4.0	16#0	16#0223		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3202	Word	6.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3203	Word	8.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3204	Word	10.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3205	Word	12.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3206	Word	14.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3207	Word	16.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

La siguiente figura muestra los valores escritos y leídos al ATV312 en donde ya está consignada una velocidad.

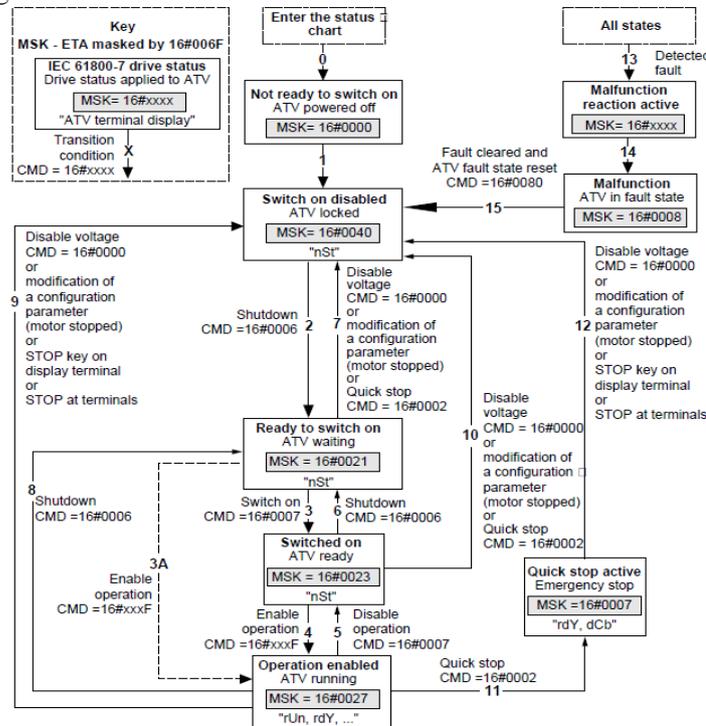
ESCRITURA/LECTURA DE ATV312									
Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación	Remanen...	Accesible d...	Visible en ...	Valor de a...	
Static									
8501	Word	0.0	16#0	16#000F		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8502	Word	2.0	16#0	16#0578		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3201	Word	4.0	16#0	16#0227		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3202	Word	6.0	16#0	16#0252		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3203	Word	8.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3204	Word	10.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3205	Word	12.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3206	Word	14.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3207	Word	16.0	16#0	16#0000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

CONCLUSIONES:

La activación de funcionamiento del motor se realizó con ayuda del manual de programación del ATV312 en donde se indica detalladamente la secuencia de palabras que deben ser enviadas.

La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

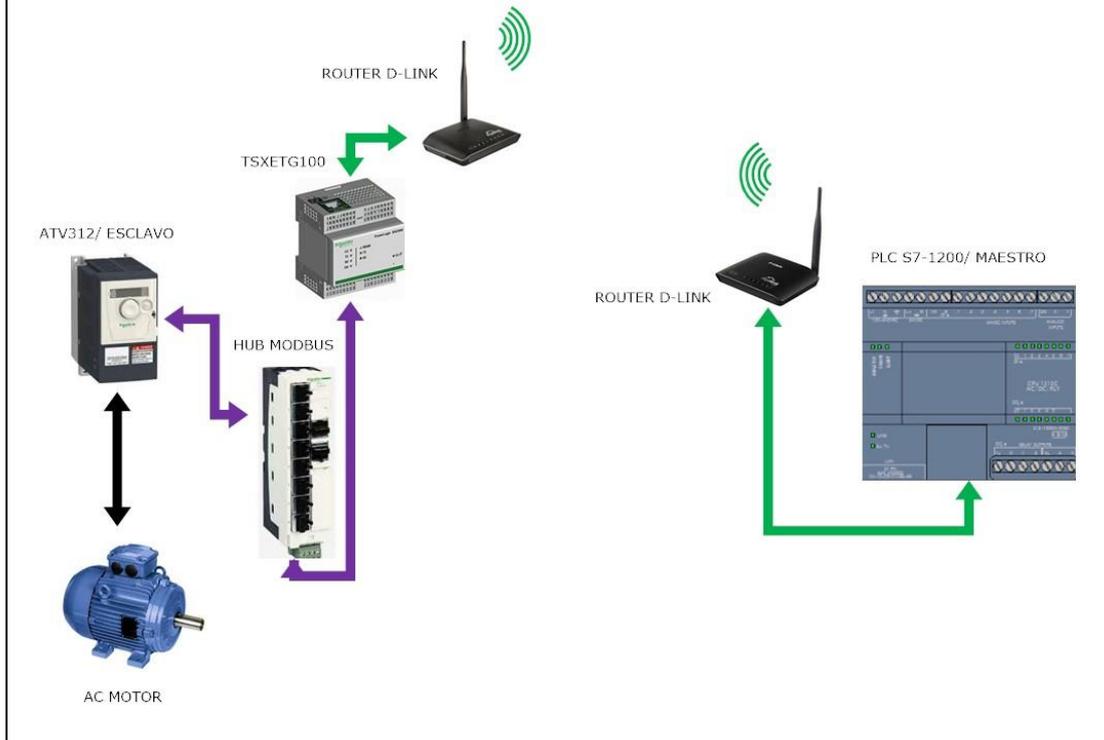
Para la comunicación de los variadores ATV312, solo basta la configuración antes descrita y si no se ha modificado otro parámetro, la escritura de velocidad y una secuencia de valores para el arranque del variador según se describe en la carta de estado.



RECOMENDACIONES:

- Se recomienda disponer del manual de variables del ATV312 para el correcto direccionamiento de las palabras de control y función del motor.
- Enviar la secuencia de activación del motor correctamente ya que si no envía la secuencia el motor no arrancará.

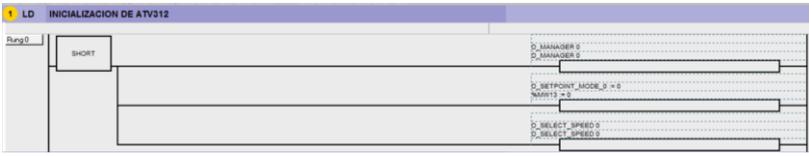
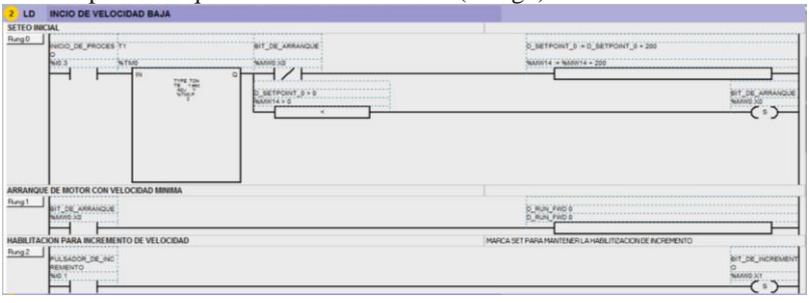
- Verificar el estado de los cables que no presenten torceduras ni falsos contactos.
- Asegurar y fijar el cable de conexión del variador al HUB Modbus.
- Dispones del siguiente diagrama de red que representa la conexión de los equipos utilizados, no se detalla la dirección de los equipos ya que estas pueden ser modificadas por el programador.

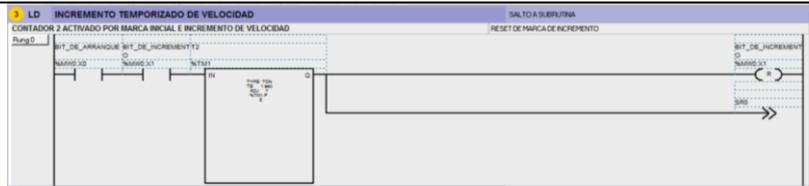


Docente / Técnico Docente: _____

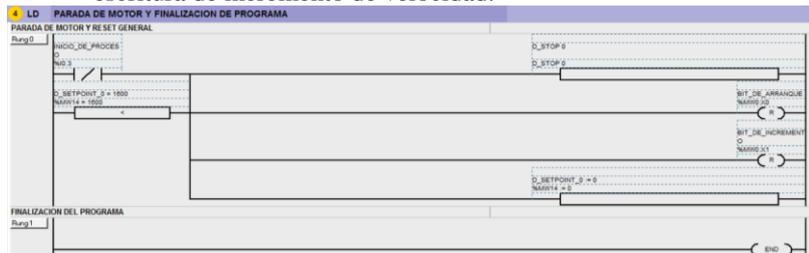
Firma: _____

4.3. Práctica 3

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III		
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Monitoreo y control de variador de frecuencia ATV312 mediante Macro Drive de PLC Twido	
OBJETIVOS: <ul style="list-style-type: none"> • Entrenar al estudiante con los bloques Modbus TCP Siemens. • Comprender el mapeo de registro de ambas marcas de PLC's. • Monitoreo de datos enviados entre los esclavos. • Manejo de envío/recepción de palabras. • Manejo y gestión de macro drive de PLC Twido. 			
INSTRUCCIONES:	1. Introducción de funcionamiento del programa. El PLC esclavo gestiona el control de activación del motor y los valores escritos por el esclavos son monitoreados desde el maestro.		
	2. Configuración de direccionamiento RTU del esclavo variador (2), que es controlado por medio de Macro Dirve de Twido.		
	3. Configuración y programación de esclavo Twido (1). <ul style="list-style-type: none"> • Configuración de parámetros de macro drive como, inicio de palabras a utilizar desde Twido (10) y máxima valor de palabras escritas (30). • Programación de circuito ladder. • LD1: Inicialización de parámetros de control para VDF ATV312. Sentencias obligadas que deben activarse para la comunicación con macro drive. 		
			
<ul style="list-style-type: none"> • LD2: Seteo de velocidad inicial para arranque de motor. Conteo de 3 segundos para arrancar el motor con la velocidad inicial de y activación de marca de arranque para enviar dato de arranque (Rung1). Habilitación de incremento de velocidad por medio de pulsador que activa una marca set (Rung2). 			
			
<ul style="list-style-type: none"> • LD3: Incremento temporizado de velocidad por medio de pulsador. Marca de incremento activa temporizador para aumentar un valor fijo la velocidad actual del motor, el temporizador al activarse resetea la marca de incremento. 			



- LD4: detención de motor mediante pulsador y límite máximo de velocidad establecida en el programa. Se finaliza el programa con la sentencia **END** para poder establecer una subrutina para la escritura de incremento de velocidad.

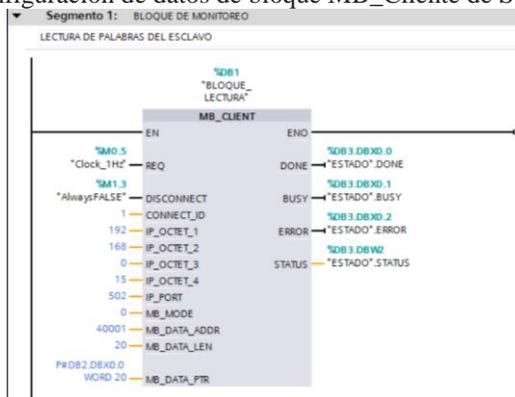


- LD5 (SR). Subrutina de incremento de velocidad, se suma un valor fijo a la velocidad actual del motor y luego de hacer su función, retorna a la última línea de donde se la llamo.



3. Configura y programación PLC cliente.

- En la configuración del PLC se activa las señales de marca del sistema (%MB1) y marcas de ciclo (%MB0).
- Configuración de datos de bloque MB_Cliente de S7-120.



- Creación de bloque de datos con direccionamiento de datos leídos del servidor.

HOLDING REGISTER	Nombre	Tipo de datos	Offset	Valor de arranq...
1	Static			
2	HOLDING REGISTER	Array [0..19]	0.0	
3	HOLDING REGISTE...	Word	0.0	16#0
4	HOLDING REGISTE...	Word	2.0	16#0
5	HOLDING REGISTE...	Word	4.0	16#0
6	HOLDING REGISTE...	Word	6.0	16#0
7	HOLDING REGISTE...	Word	8.0	16#0
8	HOLDING REGISTE...	Word	10.0	16#0
9	HOLDING REGISTE...	Word	12.0	16#0
10	HOLDING REGISTE...	Word	14.0	16#0
11	HOLDING REGISTE...	Word	16.0	16#0
12	HOLDING REGISTE...	Word	18.0	16#0
13	HOLDING REGISTE...	Word	20.0	16#0
14	HOLDING REGISTE...	Word	22.0	16#0
15	HOLDING REGISTE...	Word	24.0	16#0
16	HOLDING REGISTE...	Word	26.0	16#0
17	HOLDING REGISTE...	Word	28.0	16#0
18	HOLDING REGISTE...	Word	30.0	16#0
19	HOLDING REGISTE...	Word	32.0	16#0
20	HOLDING REGISTE...	Word	34.0	16#0
21	HOLDING REGISTE...	Word	36.0	16#0
22	HOLDING REGISTE...	Word	38.0	16#0

ACTIVIDADES POR DESARROLLAR

1. Verificar los valores enviados en los bloques Macro Drive del esclavo.
2. Monitorear la velocidad seteada actual que tenga el motor.
3. Modificar el programa del esclavo y que el PLC cliente pueda registrar las palabras utilizadas.
4. Incluir hasta 3 motores para control con Macro Drive y registrar los valores de velocidades seteadas.

RESULTADO(S) OBTENIDO(S):

Se obtienen los valores de velocidad escrita y los valores leídos en las palabras utilizadas en el PLC esclavo.

HOLDING REGISTER					
Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación	
Static					
HOLDING REGISTER	Array [0..19]...	0.0			
HOLDING REGISTE...	Word	0.0	16#0	16#0001	
HOLDING REGISTE...	Word	2.0	16#0	16#0000	
HOLDING REGISTE...	Word	4.0	16#0	16#0000	
HOLDING REGISTE...	Word	6.0	16#0	16#0000	
HOLDING REGISTE...	Word	8.0	16#0	16#0000	
HOLDING REGISTE...	Word	10.0	16#0	16#0000	
HOLDING REGISTE...	Word	12.0	16#0	16#0000	
HOLDING REGISTE...	Word	14.0	16#0	16#0000	
HOLDING REGISTE...	Word	16.0	16#0	16#0000	
HOLDING REGISTE...	Word	18.0	16#0	16#0000	
HOLDING REGISTE...	Word	20.0	16#0	16#0002	
HOLDING REGISTE...	Word	22.0	16#0	16#0027	
HOLDING REGISTE...	Word	24.0	16#0	16#7510	
HOLDING REGISTE...	Word	26.0	16#0	16#0000	
HOLDING REGISTE...	Word	28.0	16#0	16#00C8	
HOLDING REGISTE...	Word	30.0	16#0	16#000F	
HOLDING REGISTE...	Word	32.0	16#0	16#00C8	
HOLDING REGISTE...	Word	34.0	16#0	16#000F	
HOLDING REGISTE...	Word	36.0	16#0	16#00C8	
HOLDING REGISTE...	Word	38.0	16#0	16#0106	

Los valores mostrados representan las palabras escritas por medio de las Macro Drive, desde el PLC Twido hacia al variador ATV312.

HOLDING REGISTE...	Word	28.0	16#0	16#0190
HOLDING REGISTE...	Word	30.0	16#0	16#000F
HOLDING REGISTE...	Word	32.0	16#0	16#0190
HOLDING REGISTE...	Word	34.0	16#0	16#000F
HOLDING REGISTE...	Word	36.0	16#0	16#0190
HOLDING REGISTE...	Word	38.0	16#0	16#0106

CONCLUSIONES:

En esta práctica se muestra un sencillo ejemplo de monitoreo de un proceso, en este caso de la velocidad de un motor.

Se demuestra la configuración de los bloques correspondientes para la comunicación y la aplicación de subrutinas en el caso del PLC Twido.

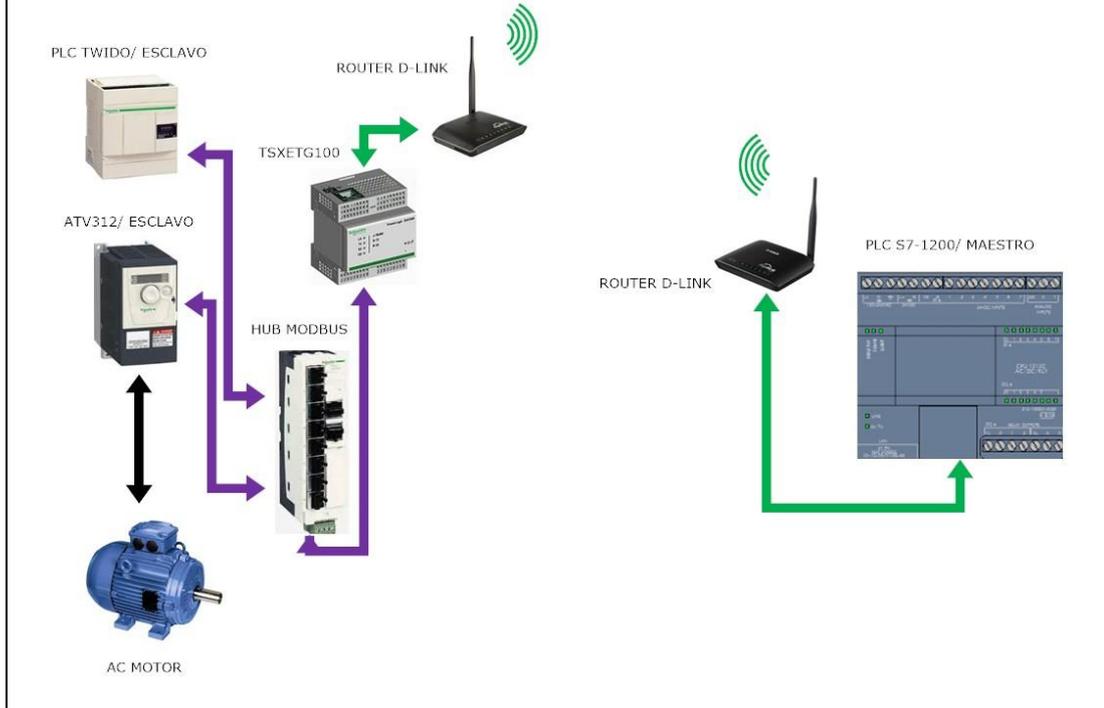
La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

En esta práctica se hace uso de control de motor por medio de Macro Drive tan solo para demostrar un ejemplo sencillo de las funcionalidades del PLC Twido, sin embargo esto no es recomendable que se ejecute para un proceso ya que no puede haber más de un maestro en la red y con esta configuración se está poniendo al Twido como un maestro local en Modbus RTU.

RECOMENDACIONES:

- Se recomienda hacer una inspección previa de los cables Modbus con los que se conectarán los equipos.
- La frecuencia de requisición de solicitud del bloque Mb_Client (REQ), debe ser al menos el doble de velocidad de respuesta del esclavo.
- Se recomienda un uso máximo de hasta 3 variadores para control por medio de Macro Drive.

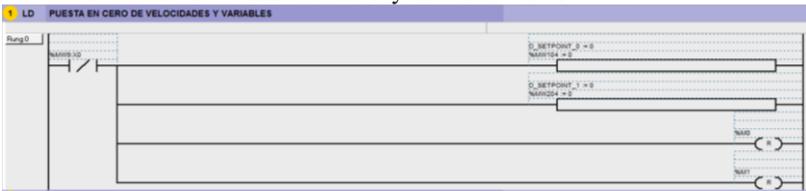
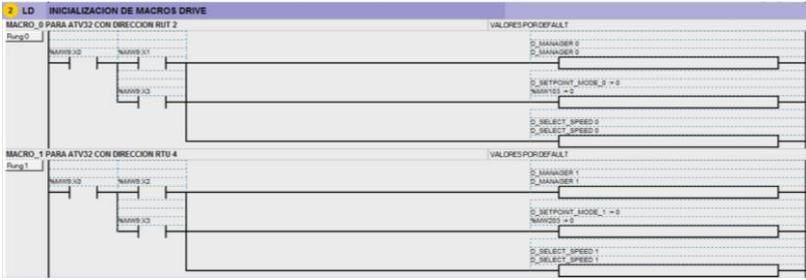
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.
- Se recomienda la conexión de los equipos de acuerdo al siguiente diagrama de red, no se especifica identificación de esclavos ya que este puede ser cambiado por el programador.



Docente / Técnico Docente: _____

Firma: _____

4.4. Práctica 4

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III		
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Secuencia de encendido de motores	
OBJETIVOS: <ul style="list-style-type: none"> • Arranque de dos motores. • Seteo de parámetros del motor. • Utilización de Macro Drive en PLC Twido. • Direccionamiento Modbus a variador de frecuencia. 			
INSTRUCCIONES:	1. Conexión de los equipos correspondientes al HUB Modbus del módulo convertidor.		
	2. Configuración de direccionamiento RTU de los esclavos variadores (2 y 3), que son controlados por medio de Macro Drive de Twido.		
	3. Configuración y programación de esclavo Twido (4). <ul style="list-style-type: none"> • Configuración de hardware Twido y parámetros de comunicación Modbus. • Configuración de parámetros de macro drive como, inicio de palabras a utilizar desde Twido (10) y máxima valor de palabras escritas (30). • En la ventana de programación se agrega la primera línea de programación. • En esta aplicación se selecciona una palabra distinta de la inicial, esto para poder direccionar desde el cliente cualquier palabra que se desee utilizar. • En el primer LD se ubica el bit de memoria de la palabra que será escrita (MW9:X0) para el inicio del sistema, se setea en cero todas las variables utilizadas incluyendo las velocidades. 		
			
<ul style="list-style-type: none"> • En el siguiente ladder, se realiza la programación de selección de motor que se desee arrancar, la activación de las marcas utilizadas se describen en el cuadro de estado. 			
			
<ul style="list-style-type: none"> • Ladder 3 describe el sentido de giro del motor mediante una sola marca, en donde en estado 0 se activa la marca M1 la cual setea el 			

sentido reverse del motor, mientras que en estado 1, la marca M0 setea en sentido adelante.

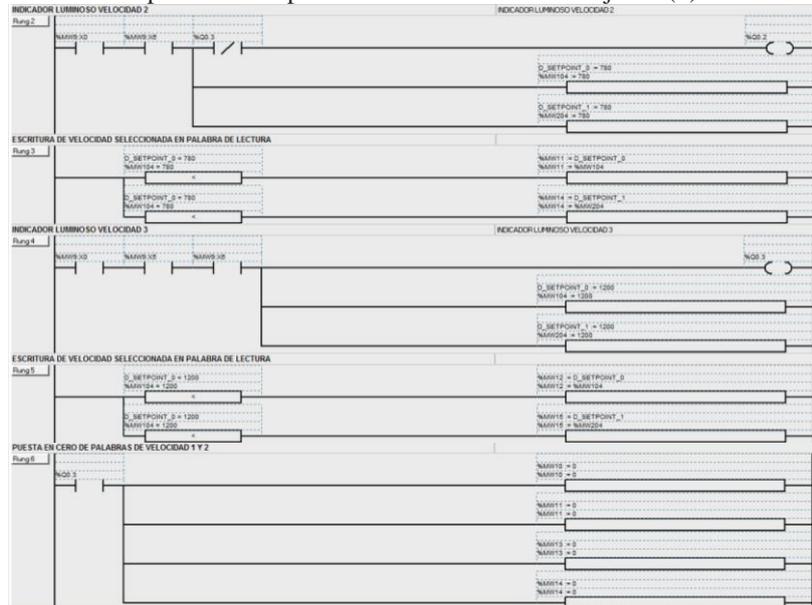


- En el siguiente bloque se programa el encendido de indicadores luminosos, según la velocidad que se haya setea. Las velocidades son fijas y se setean en las palabras correspondientes una vez que hayan sido seleccionadas.



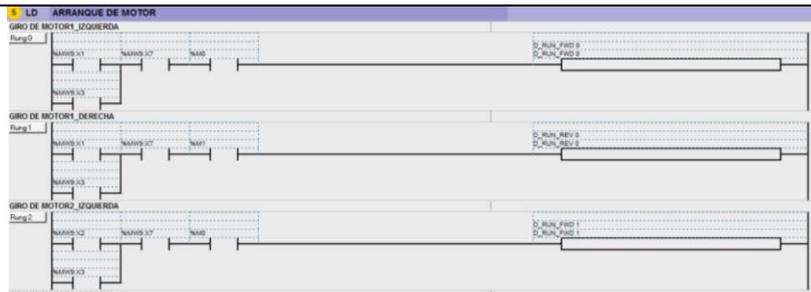
- Adicionalmente se programa un bloque de datos, con el fin que desde el cliente se lea la velocidad seleccionada en otra palabra que no es la de escritura en el variador, es decir, se mueve el valor escrito por el macro drive en el variador hacia una palabra del PLC, esto con el objetivo de demostrar que el cliente puede leer la dirección que se seleccione en el bloque.

- Este proceso se repite en todas la velocidades fijadas (3).



- En el Rung 6 se establece la puesta en cero de las palabras de velocidad por medio de la salida Q0.3, esto debido a que la selección de la velocidad 3 es pulsando los dos selectores de velocidad (1 y 2), el cuadro de estado se define en la programación del cliente.

- En el lader 5 se programa el arranque del motor seleccionado, esto se logra mediante un pulso desde el cliente y la selección de bits de estado de control de la palabra MW9.



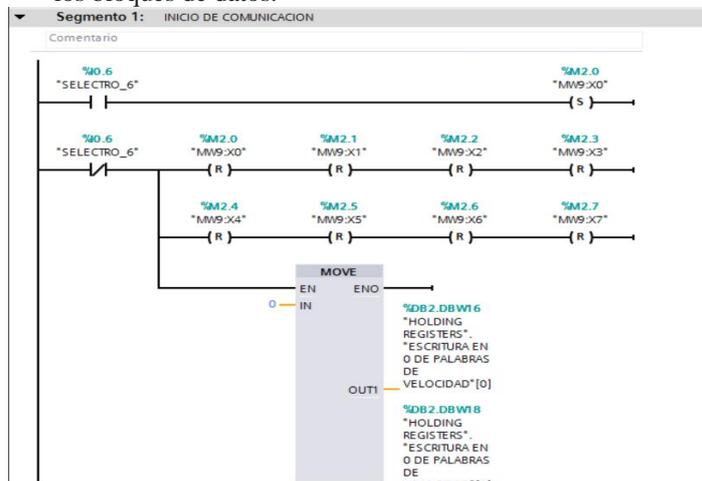
- El lader 6 establece el paro de los motores, antes, durante y después de la simulación.

3. Configura y programación PLC cliente.

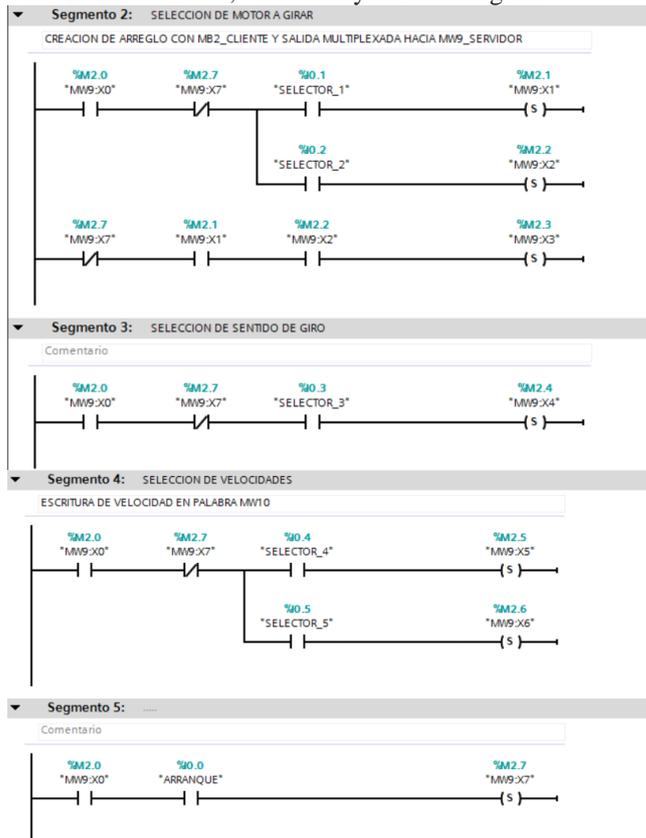
- En la configuración del PLC se activa las señales de marca del sistema (%MB1) y marcas de ciclo (%MB0).
- En la programación del maestro se establece la escritura de los bits desde los selectores del tablero de control.
- En el programa se utiliza una marca tipo byte (MB2), del cual se hace la utilización tipo set de las bobinas que almacenan el valor 0 o 1 al momento de seleccionar el respectivo selector.
- La tabla de variables correspondientes entre los selectores y los bits de MB2 se describen en la opción variables del software Tia Portal.

15	ARRANQUE	Tabla de variables e.. Bool	%I0.0
16	SELECTOR_1	Tabla de variables e.. Bool	%I0.1
17	SELECTOR_2	Tabla de variables e.. Bool	%I0.2
18	SELECTOR_3	Tabla de variables e.. Bool	%I0.3
19	SELECTOR_4	Tabla de variables e.. Bool	%I0.4
20	SELECTOR_5	Tabla de variables e.. Bool	%I0.5
21	MW9:X0	Tabla de variables e.. Bool	%M2.0
22	MW9:X1	Tabla de variables e.. Bool	%M2.1
23	MW9:X2	Tabla de variables e.. Bool	%M2.2
24	MW9:X3	Tabla de variables e.. Bool	%M2.3
25	MW9:X4	Tabla de variables e.. Bool	%M2.4
26	MW9:X5	Tabla de variables e.. Bool	%M2.5
27	MW9:X6	Tabla de variables e.. Bool	%M2.6
28	MW9:X7	Tabla de variables e.. Bool	%M2.7
29	SELECTRO_6	Tabla de variables e.. Bool	%I0.6
30	BYTE_WORD9	Tabla de variables e.. Byte	%MB2

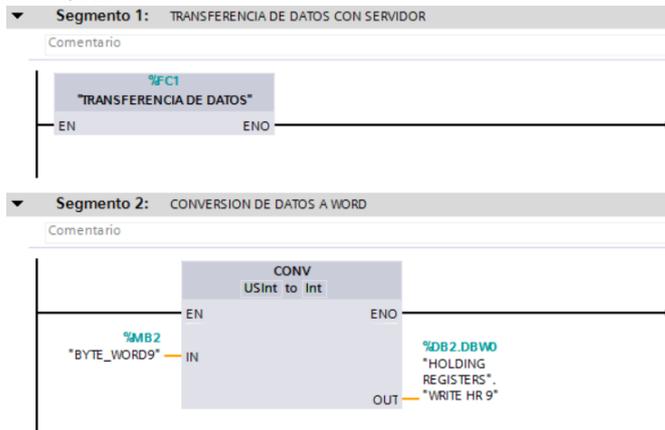
- En el bloque de transferencia se realiza la programación respectiva para la escritura de los bit de MB2, también se incluye la puesta en cero de los valores de velocidades en el servidor y las variables en los bloques de datos.



- En los siguientes segmentos se programa el seteo de los bits para selección de motor, velocidad y sentido de giro.



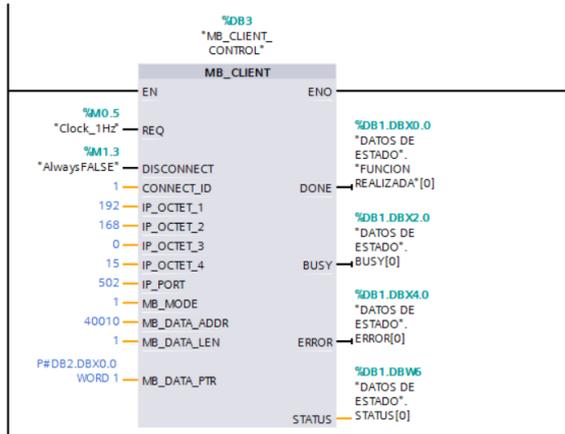
- En el bloque principal se hace el llamado del bloque de función (transferencia de datos) y en el segmento 2 se agrega un bloque convertidor del byte MB2 a su correspondiente valor en tipo WORD.



- El bloque de conversión se lo agrego debido a que en el maestro se manda a escribir en una palabra y no en bits (coil), por lo tanto el byte MB2 (8 bits) se convierte en un valor entero (16 bits = 1 Word) que se guarda en la variable correspondiente del bloque de datos Holding Register.
- En el segmento 3 se programa el bloque de transferencia de los datos de control, es decir el valor convertido de MB2.

Segmento 3: LECTURA/ESCRITURA DE PALABRA DE CONTROL

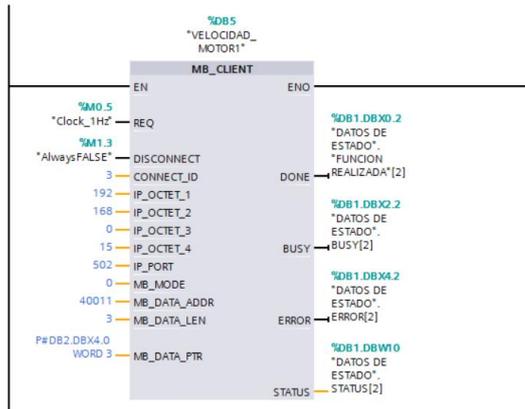
Comentario



- También se programa el bloque de lectura de la palabra MW9 del servidor, se tiene en cuenta que el valor leído es una palabra.
- En el siguiente segmento se programa los bloques de lectura de las velocidades, estos bloques se programan en diferentes instrucciones para poder clasificar las velocidades en las variables donde se almacenan.

Segmento 4: LECTURA DE VELOCIDADES DE MOTOR 1 Y 2

Comentario



- En el segmento 5 se programa un bloque de puesta a cero, esto para encerrar valores que puedan estar almacenados en las variables que se utilizan en el servidor.
- Como último paso se crean bloque de datos con variables correspondientes que deberán ser usadas.
- En el bloque de datos de Holding Register se nombran las variables de lectura y escritura del servidor.
- En el bloque de datos de estado se nombran variables de propósito general, como lo son indicaciones de las instrucciones MB_Cliente.

ACTIVIDADES POR DESARROLLAR

1. Verificar los valores enviados en los bloques Macro Drive del esclavo.
2. Monitorear la velocidad seteada actual de los motores.
3. Optimizar el diagrama de estado del programa, para que la programación del cliente se minimice.

RESULTADO(S) OBTENIDO(S):

En base al siguiente diagrama de estado se monitorea los valores de control y estado del sistema que se han escrito y leídos en el PLC esclavo Twido.

ARRANQUE	V2	V1	GIRO	M2	M1	INICIO	ESTADO
	V1 AND V2 = V3		0=Der 1= Izq	M1 AND M2 = MW9:X3			
MW9:X7	MW9:X6	MW9:X5	MW9:X4	MW9:X2	MW9:X1	MW9:X0	
I0.0	I0.5	I0.4	I0.3	I0.2	I0.1	I0.6	
0	0	0	0	0	0	0	OFF
0	0	0	0	0	0	1	ON
0	0	0	0	0	1	1	M1_D
0	0	0	0	1	0	1	M2_D
0	0	0	0	1	1	1	M1Y2_D
0	0	0	1	0	1	1	M1_IZ
0	0	0	1	1	0	1	M2_IZ
0	0	0	1	1	1	1	M1Y2_IZ
0	0	1	0	0	1	1	M1_V1_D
0	0	1	0	1	0	1	M2_V1_D
0	0	1	0	1	1	1	M1Y2_V1_D
0	0	1	1	0	1	1	M1_V1_IZ
0	0	1	1	1	0	1	M2_V1_IZ
0	0	1	1	1	1	1	M1Y2_V1_IZ
0	1	0	0	0	1	1	M1_V2_D
0	1	0	0	1	0	1	M2_V2_D
0	1	0	0	1	1	1	M1Y2_V2_D
0	1	0	1	0	1	1	M1_V2_IZ
0	1	0	1	1	0	1	M1_V2_IZ
0	1	0	1	1	1	1	M1Y2_V2_IZ
0	1	1	0	0	1	1	M1_V3_D
0	1	1	0	1	0	1	M2_V3_D
0	1	1	0	1	1	1	M1Y2_V3_D
0	1	1	1	0	1	1	M1_V3_IZ
0	1	1	1	1	0	1	M1_V3_IZ
0	1	1	1	1	1	1	M1Y2_V3_IZ
1	0	1	0	0	1	1	ARRANQUE EN VELOCIDA D 1
1	0	1	0	1	0	1	
1	0	1	1	0	1	1	
1	0	1	1	1	0	1	
1	0	1	1	1	1	1	
1	1	0	0	0	1	1	ARRANQUE EN VELOCIDA D 2
1	1	0	0	1	0	1	
1	1	0	1	0	1	1	
1	1	0	1	1	0	1	
1	1	0	1	1	1	1	
1	1	1	0	0	1	1	

1	1	1	0	1	0	1	ARRANQUE EN VELOCIDA D 3
1	1	1	0	1	1	1	
1	1	1	1	0	1	1	
1	1	1	1	1	0	1	
1	1	1	1	1	1	1	

- Se muestra el valor escrito en palabras del encendido de los dos motores en un sentido y con la velocidad 2 seteada.

	Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
1	Static				
2	WRITE HR 9	Word	0.0	16#0	16#00CF
3	READ HR 9	Word	2.0	16#0	16#00CF
4	LECTURA VELOCIDADES - MOTOR 1	Array [0..2] of Word	4.0		
5	LECTURA VELOCIDADES - MOTO...	Word	0.0	16#0	16#0104
6	LECTURA VELOCIDADES - MOTO...	Word	2.0	16#0	16#030C
7	LECTURA VELOCIDADES - MOTO...	Word	4.0	16#0	16#0480
8	LECTURA VELOCIDADES - MOTOR 2	Array [0..2] ...	10.0		
9	LECTURA VELOCIDADES - MOTO...	Word	0.0	16#0	16#0104
10	LECTURA VELOCIDADES - MOTO...	Word	2.0	16#0	16#030C
11	LECTURA VELOCIDADES - MOTO...	Word	4.0	16#0	16#0480

- Se muestra la selección de la velocidad 2 para el motor 2.

	Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
1	Static				
2	WRITE HR 9	Word	0.0	16#0	16#00C5
3	READ HR 9	Word	2.0	16#0	16#00C5
4	LECTURA VELOCIDADES - MOTOR 1	Array [0..2] of Word	4.0		
5	LECTURA VELOCIDADES - MOTOR 1[0]	Word	0.0	16#0	16#0000
6	LECTURA VELOCIDADES - MOTOR 1[1]	Word	2.0	16#0	16#030C
7	LECTURA VELOCIDADES - MOTOR 1[2]	Word	4.0	16#0	16#0000
8	LECTURA VELOCIDADES - MOTOR 2	Array [0..2] of Word	10.0		
9	LECTURA VELOCIDADES - MOTOR 2[0]	Word	0.0	16#0	16#0000
10	LECTURA VELOCIDADES - MOTOR 2[1]	Word	2.0	16#0	16#030C
11	LECTURA VELOCIDADES - MOTOR 2[2]	Word	4.0	16#0	16#0000
12	ESCRITURA EN 0 DE PALABRAS DE VELOCIDAD	Array [0..5] ...	16.0		
13	ESCRITURA EN 0 DE PALABRAS DE VELOCIDA...	Word	0.0	16#0	16#0000
14	ESCRITURA EN 0 DE PALABRAS DE VELOCIDA...	Word	2.0	16#0	16#0000
15	ESCRITURA EN 0 DE PALABRAS DE VELOCIDA...	Word	4.0	16#0	16#0000
16	ESCRITURA EN 0 DE PALABRAS DE VELOCIDA...	Word	6.0	16#0	16#0000
17	ESCRITURA EN 0 DE PALABRAS DE VELOCIDA...	Word	8.0	16#0	16#0000
18	ESCRITURA EN 0 DE PALABRAS DE VELOCIDA...	Word	10.0	16#0	16#0000

CONCLUSIONES:

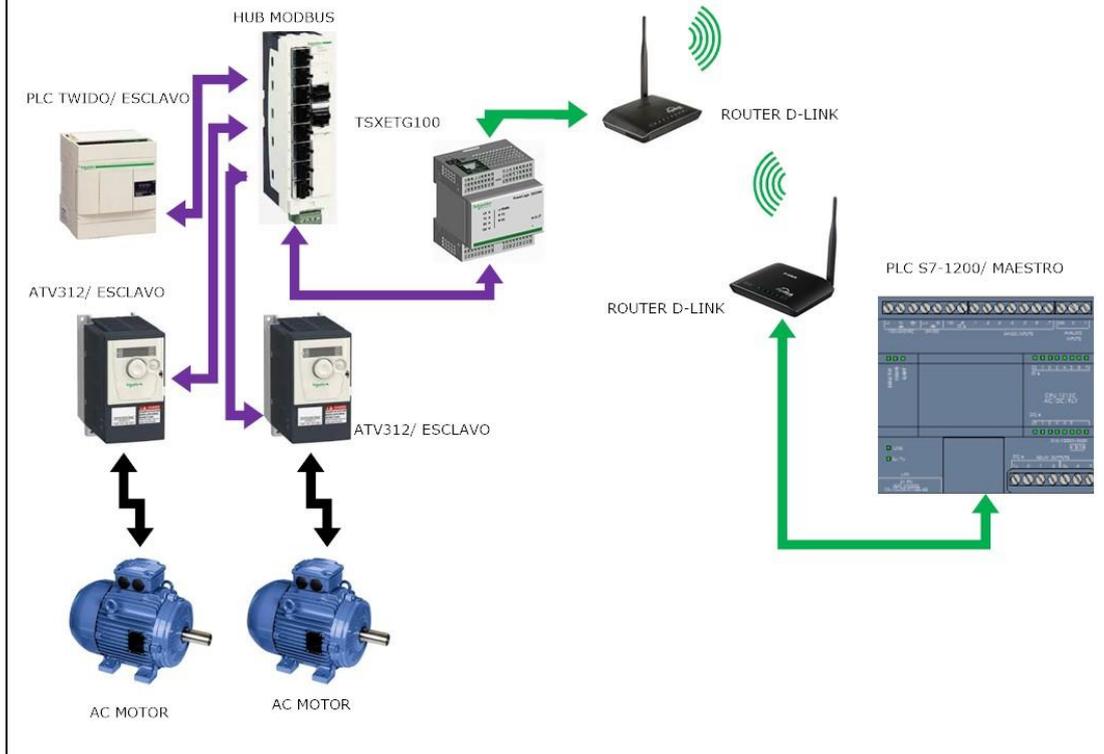
En esta práctica se estudia los distintos bloques de función que presta el software Tia PORTAL, como es el bloque de conversión de datos, también se hace la utilización de una herramienta digital como es el cuadro de estado, esto simplifica la programación para conocer el estado que presenta la simulación según los bits seleccionados.

La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

RECOMENDACIONES:

- Optimizar el diagrama de estado del PLC cliente.
- Incrementar las variables para la conexión de un motor adicional.
- La frecuencia de envío de datos del bloque Mb_Client, debe ser al menos el doble del tiempo de respuesta del esclavo.
- Verificar el estado de los cables de conexión de los equipos.
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.

- Se recomienda seguir el siguiente diagrama de conexión para los equipos utilizados, no se detalla la dirección de red de los equipos ya que esta pueda ser cambiada de acuerdo al programador.

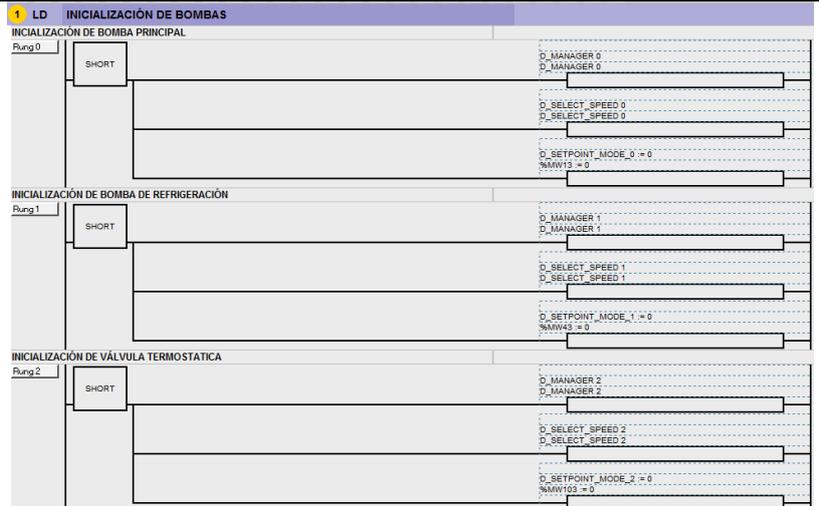


Docente / Técnico Docente: _____

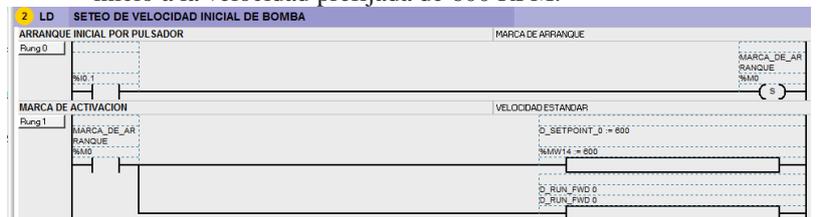
Firma: _____

4.5. Práctica 5

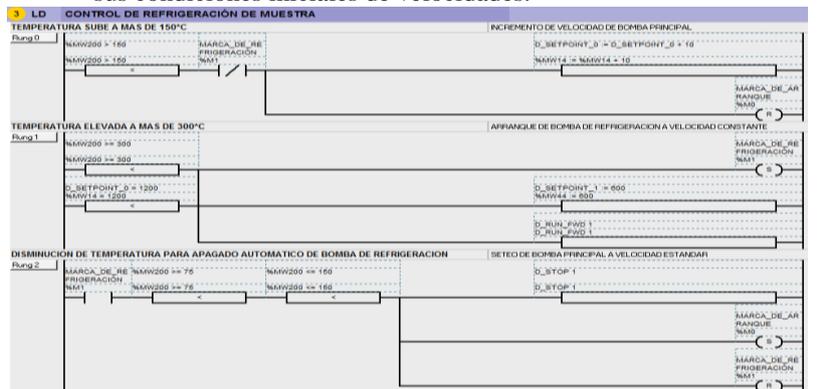
		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III	
NRO. PRÁCTICA:	5	TÍTULO PRÁCTICA: Simulación de sistema de enfriamiento para procesos de laboratorio con un equipo recirculador CHILLER.
OBJETIVOS:		
<ul style="list-style-type: none"> • Simulación de proceso de enfriamiento. • Simulación de estados de funcionamiento de acuerdo a la temperatura. • Manejo de señal análoga. • Monitorización de temperatura del servidor. 		
INSTRUCCIONES:	<p>1. Introducción de funcionamiento del programa. Un recirculador chiller, es un equipo que hace circular agua a bajas temperaturas a través de un sistema, con el fin de mantener una temperatura fija para un proceso específico. Esta práctica describe el funcionamiento del circuito de frío de un recirculador y simula dos condiciones que se presentan por temperatura baja o alta en el reservorio o en el sistema que se está enfriando. Este sistema está constituido por dos bombas, una principal y otra secundaria, la activación de estas se da mediante un pulsador y solo enciende la bomba principal a una velocidad, a medida que la temperatura va subiendo, la bomba principal recirculara más rápido hasta una velocidad tope, donde posteriormente si la temperatura sigue subiendo arrancara la bomba secundaria a una velocidad hasta volver a enfriar el sistema y esta volverá a apagarse y la bomba primaria regresara a la velocidad inicial. Una segunda condición de funcionamiento de este equipo es respecto a la temperatura del agua del reservorio, la cual debe mantenerse hasta una temperatura límite de frío y si esta baja de cierto nivel de temperatura, se activara una válvula termostática que es simulada por un motor y permitirá el paso temporizado del calor generado en el lado de alta presión del compresor para mantener esta temperatura al límite. La simulación de los datos de temperatura se realiza mediante la entrada analógica del PLC Cliente, del cual se transmite esta información y se monitorea las condiciones del sistema.</p>	
	<p>2. Configuración de direccionamiento RTU de los esclavos variadores (2, 3 y 4), que son controlados por medio de Macro Dirve de Twido.</p>	
	<p>3. Configuración y programación de esclavo Twido (4).</p> <ul style="list-style-type: none"> • Configuración de hardware Twido y parámetros de comunicación Modbus. • Inicialmente se seleccionan y configuran los macro drive a utilizarse, que en esta aplicación son 3 y se introduce el inicio del rango de dirección de las palabras a utilizar por macro y la dirección del ATV312 en la red. Este procedimiento se realiza para las 3 macro drive utilizadas, cada una con su respectiva dirección de red. • En el primer LD se establece la configuración inicial para cada macro drive, como son: el bloque de inicio del macro, la velocidad inicio del VDF y el bloque de selección de velocidad. • Cada Rung del LD 1, tiene su respectiva identificación para la simulación que se ejecutara luego de la programación. 	



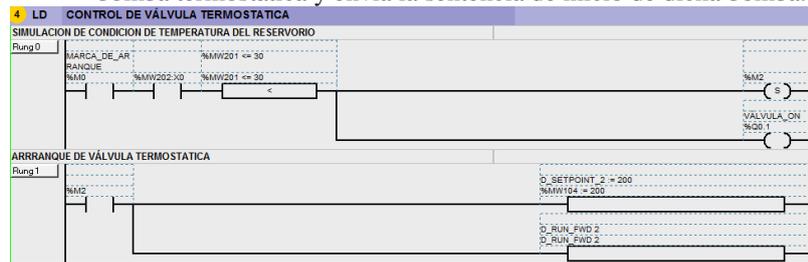
- En el siguiente ladder, se programa el seteo de velocidad inicial de la bomba principal del recirculador, la velocidad inicial es fija y el arranque del sistema se da mediante un pulsador %I0.1, el cual da inicio a la velocidad prefijada de 600 RPM.



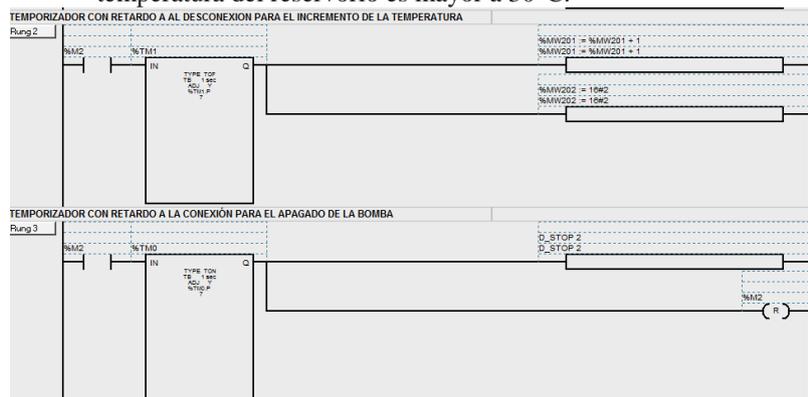
- En el ladder 3 se programa las condiciones de funcionamiento para la simulación del control de temperatura elevada del sistema que está siendo enfriado, la temperatura se escribe en la palabra MW200 y si esta supera 150°C, la velocidad de la bomba de refrigeración primaria empieza aumentar hasta una velocidad límite de 1200 RPM.
- Adicional cuando la temperatura alcanza 300°C o la bomba primaria ya ha alcanzado su máxima velocidad, se inicia el arranque de la bomba secundaria, esta gira a una velocidad fija independientemente de que la velocidad siga subiendo.
- En cuanto las dos bombas estén encendidas, la temperatura deberá empezar a bajar y cuando esta empiece a bajar entre 75°C y 150°C se detendrá la bomba secundaria y la bomba primaria regresara a sus condiciones iniciales de velocidades.



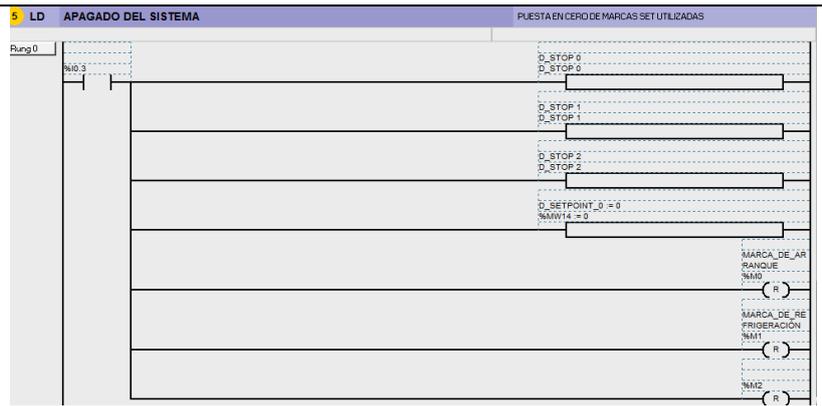
- En el siguiente bloque se programa la condición de temperatura del reservorio, esta respecta a la temperatura máxima que debe mantenerse el agua de enfriamiento. Cabe recalcar que este funcionamiento se da mediante un compresor de alta presión.
- Para la activación de este estado, se envía la orden desde el cliente hacia el bit 0 de la palabra MW202 y la temperatura se escribe en la palabra MW201.
- Si la temperatura del reservorio baja de 30°C se encenderá un indicador luminoso en el servidor, el cual mostrara que la válvula se ha activado.
- Posteriormente la marca set M2 activa la velocidad seteada a la bomba termostática y envía la sentencia de inicio de dicha bomba.



- En el mismo ladder se programan dos temporizadores, el primer temporizador con retardo a la desconexión (TOF), se activara mediante la marca set M2 que hará que la temperatura actual del reservorio aumente en una unidad durante 7 segundo, también con este bloque se envía un dato de lectura de la palabra MW202 para que el cliente deje de enviar el dato de temperatura baja hacia el servidor, ya que el incremento de la temperatura se hace en la misma variable escrita desde el cliente.
- El segundo temporizador con retardo a la conexión (TON), permite que en 7 segundos la válvula termostática se desactive ya que la temperatura del reservorio es mayor a 30°C.



- En el ladder 5 se programa el apagado del sistema, reset de todas la bobinas seteadas y detención de las bombas en movimiento.

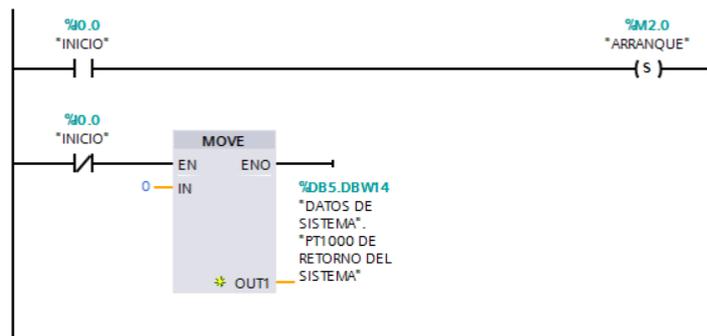


3. Configura y programación PLC cliente.

- En el software se configura el PLC correspondiente a usar (S7-1200) y la serie que este describe en la carcasa.
- En la configuración del PLC se activa las señales de marca del sistema (%MB1) y marcas de ciclo (%MB0).
- Se crean los OB's correspondientes, uno para los datos de estados de bloques de instrucciones de comunicación y otro para la holding registers leídos y escritos del servidor.
- Se crea un FC para la programación de envío de datos hacia el servidor y este bloque se copió en el programa principal del cliente.
- En la programación de los datos de envío (FC) se programa la secuencia de activación de los motores y se hace la simulación de los dos posibles estados que presenta el circuito de frio de recirculador.
- La activación de la simulación de estados se da mediante un pulsador en el cliente y este activa una bobina set que activara el bloque de conversión para la entras IW64.

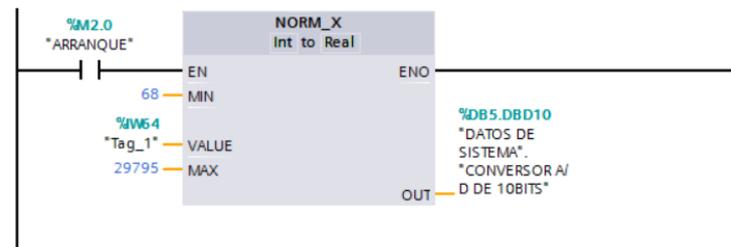
Segmento 1: INICIO DE RECIRCULADOR

Comentario



Segmento 2: CONVERSION A/D

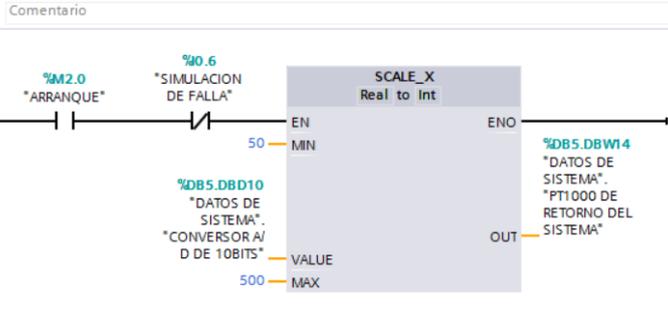
Comentario



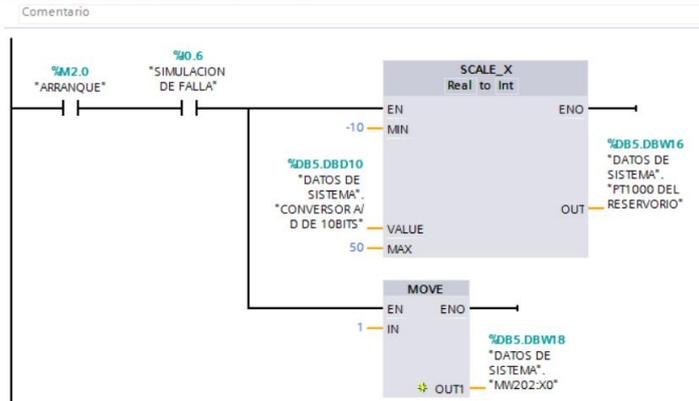
- En el siguiente segmento se programa el escalado para cada tipo de estados o falla que puede ocurrir en el sistema, esto se logra

mediante un switch de dos estados en donde dependiendo del estado actual de la entrada IO.6, se escalara el valor de la entrada analógica dependiendo a los rangos admisibles para cada estado posible.

Segmento 3: ESCALAMIENTO DE TEMPERATURA DEL SISTEMA ENFRIADO



Segmento 4: ESCALADO DE TEMPERATURA DEL RESERVORIO



- En el segmento 4 se envía un valor de arranque a la palabra MW202 del servidor, esto para dar inicio a la simulación de temperatura del reservorio del chiller, se debe revisar y constatar este estado en el programa del esclavo.

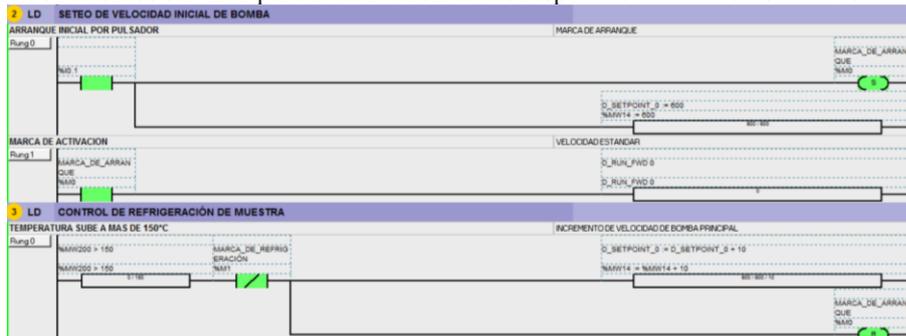
ACTIVIDADES POR DESARROLLAR

1. Verificar los valores enviados en los bloques Macro Drive del esclavo.
2. Utilizar dos entradas analógicas en el PLC cliente.
3. Modificar la aplicación enviando dos datos analógicos al mismo tiempo al servidor.

RESULTADO(S) OBTENIDO(S):

En esta práctica se obtiene el valor escrito de temperatura que se envía desde la entrada análoga del cliente.

Los valores recibidos servirán para la activación de los respectivos indicadores en el esclavo.



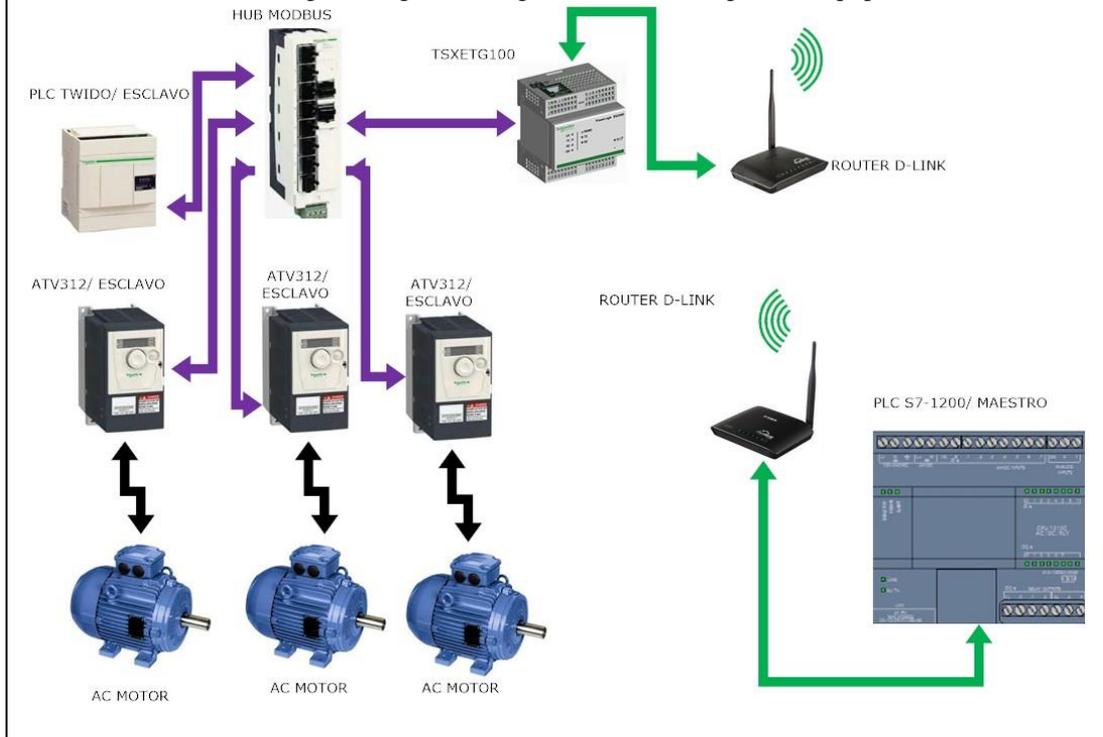
CONCLUSIONES:

En esta práctica se hace la utilización de los bloques de acondicionamiento y escalamiento de señales analógicas, esto para simular dos posibles estados que pueden presentarse en este tipo de equipos recirculadores, la simulación de este equipo solo refiere al circuito de frío y sus elementos que mantienen estable la temperatura de enfriamiento y la del reservorio.

La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

RECOMENDACIONES:

- Utilizar un sensor real (Pt100, Pt1000 o termocupla) conectado a las entradas digitales del cliente, para poder simular un valor más real.
- La frecuencia de envío de datos del bloque Mb_Client, debe ser al menos el doble del tiempo de respuesta del esclavo.
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.
- Se recomienda seguir el siguiente diagrama de conexión para los equipos utilizados.

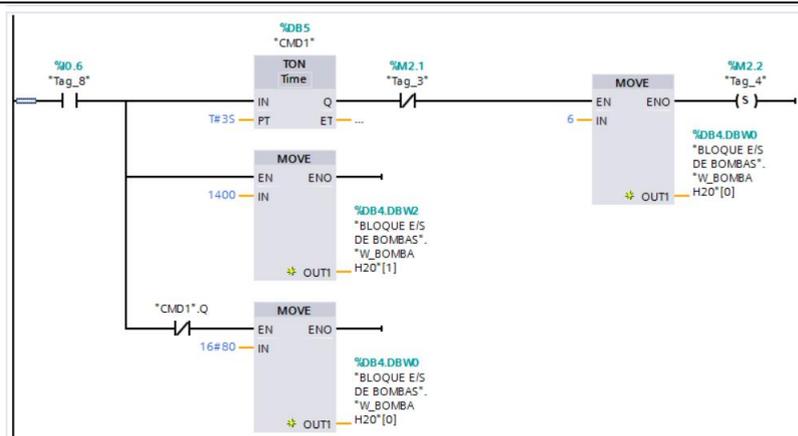


Docente / Técnico Docente: _____

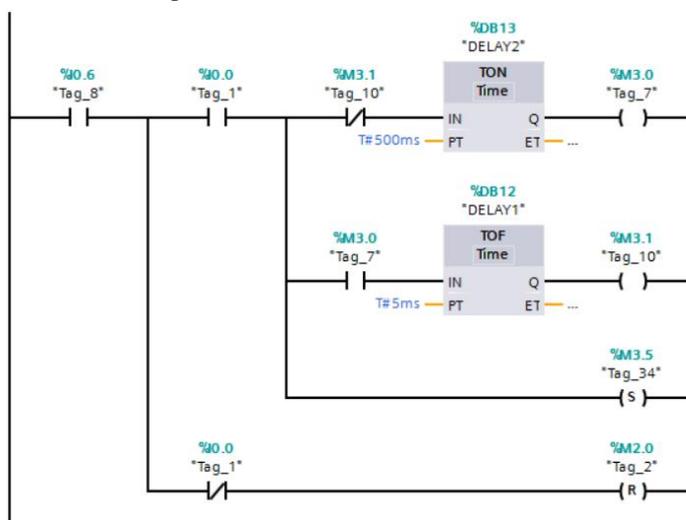
Firma: _____

4.6. Práctica 6

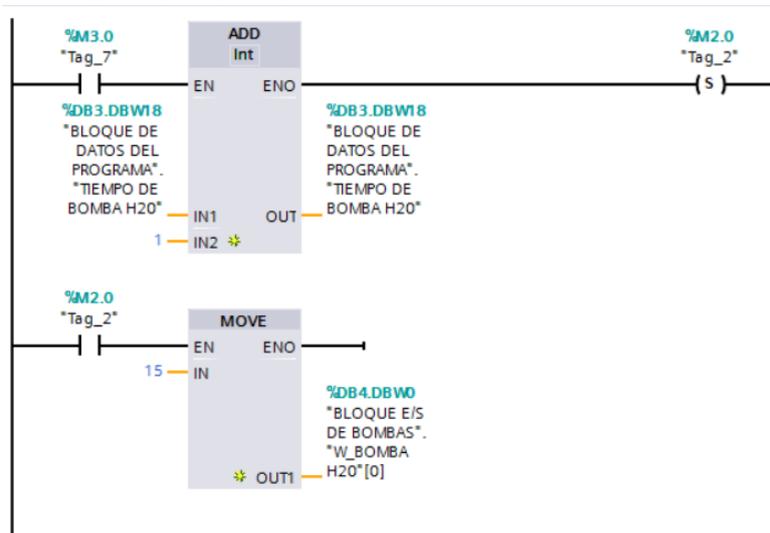
		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III		
NRO. PRÁCTICA:	6	TÍTULO PRÁCTICA: Simulación de sistema de dosificación de líquidos.	
OBJETIVOS:			
<ul style="list-style-type: none"> • Simulación de proceso de dosificación. • Comunicación directa y control de 3 variadores de frecuencia. • Almacenamiento de datos de tiempo. • Monitorización de datos escritos. 			
INSTRUCCIONES:	1. Introducción de funcionamiento del programa. Un sistema de dosificación de líquidos consta principalmente de un sistema de control y bombas peristálticas o de diafragma. Esta práctica pretende aplicar el funcionamiento del sistema de control de dosificadores que básicamente funciona por el tiempo de activación de la bomba, es decir, el nivel de líquidos dosificado es proporcional al tiempo que pasa activada la bomba. El flujo de dosificación es constante por lo que el nivel de líquido dosificado depende tan solo del tiempo de activación de la bomba. En la práctica estas bombas se calibran de acuerdo a la viscosidad del líquido que se desee dosificar y el nivel es controlado dependiendo del tiempo de activación. Las mezclas de diferentes líquidos o soluciones permiten generar reacciones diversas a una muestra de algún proceso, la presente práctica simula 3 bombas con soluciones distintas para análisis de proteínas en muestras de balanceados, se cuenta con una bomba de agua, bomba de ácido bórico y una bomba de soda. La orden de activación y calibraciones de estas bombas se envían desde el PLC cliente y dichas bombas son simuladas por 3 motores.		
	2. Configuración de direccionamiento RTU de los esclavos variadores (1, 2 y 3).		
	3. Configuración y programación PLC cliente. <ul style="list-style-type: none"> • Se selecciona el PLC con serie y versión disponible. • Se selecciona las marcas de ciclo del PLC, la cuales corresponden a la dirección de la palabra MB0. Y las marcas del sistema en la palabra MB1. • Se crean los OB's correspondientes, uno para los datos de estados de bloques de instrucciones de comunicación y otro para la holding registers leídos y escritos del servidor. • Se crean 3 FC's para cada bomba de dosificación, la programación de los tres bloques es idéntica, la diferencia entre los mismos son los diferentes direccionamientos utilizados. • En el primer segmento se programa el modo stand by del motor con una velocidad fija, esto enviando los valores correspondientes a la palabra de control del variador (revisar manual de ATV312), esta sección es activada por medio de la entrada %I0.6 para los tres motores. 		



- Se programa un segmento de retraso del conteo de tiempo de la calibración de la bomba, esto para que se retenga un valor pequeño mientras se pulsa el botón.



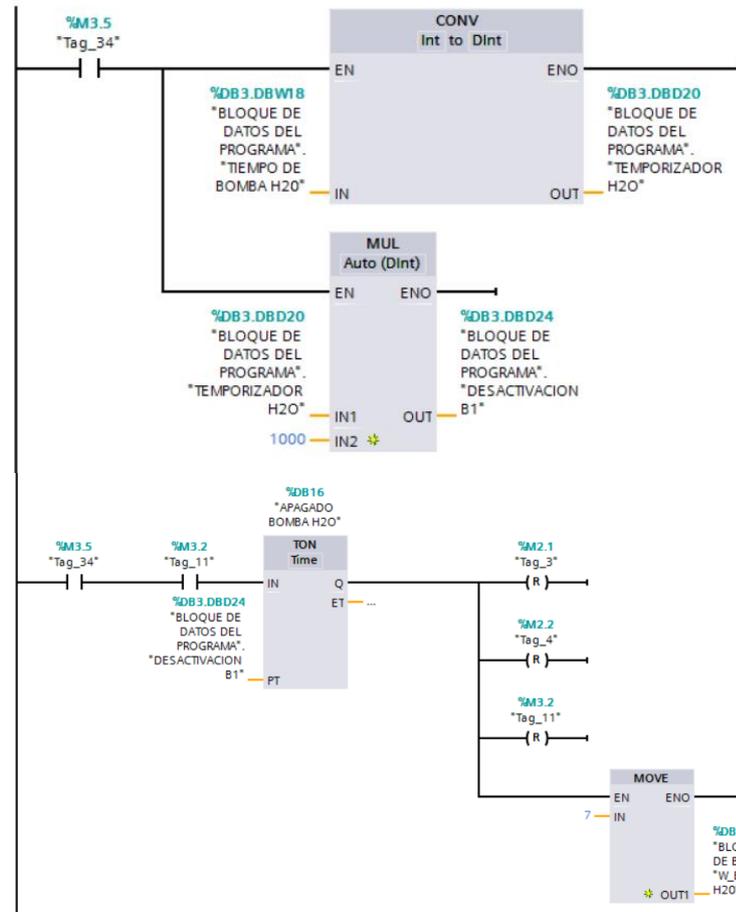
- En el siguiente bloque se programa la calibración de la bomba con un sumador, el cual es activado con una marca proveniente del delay del conteo y este valor se almacena en OB designado para transmitir este valor al servidor.



- El arranque para la verificación del volumen de dosificación calibrado se lo hace mediante el pulsador I0.3 que activara la bomba el tiempo que haya sido almacenado en el OB correspondiente.



- El último segmento consta del tiempo de apagado de la bomba una vez que se envía a la dispensación de los líquidos.



ACTIVIDADES POR DESARROLLAR

1. Verificar los valores enviados en los bloques Macro Drive del esclavo.
2. Incluir controles de funcionamiento de los motores desde el HMI del cliente.
3. Monitorear los valores de frecuencia y velocidad de funcionamiento por medio de la palabra 3201 de los ATV312.

RESULTADO(S) OBTENIDO(S):

En esta práctica se obtiene los datos de tiempo escritos en los variadores, velocidad y estado de funcionamiento de los variadores (encendido o apagados).

BLOQUE E/S DE BOMBAS				
Nombre	Tipo de datos	Offset	Valor de arranq...	
Static				
W_BOMBA H2O	Array [0..1] of Word	0.0		
W_BOMBA H2O[0]	Word	0.0	16#0	
W_BOMBA H2O[1]	Word	2.0	16#0	
R_BOMBA H2O	Array [0..1] of Word	4.0		
R_BOMBA H2O[0]	Word	0.0	16#0	
R_BOMBA H2O[1]	Word	2.0	16#0	
W_BOMBA NaOH	Array [0..1] of Word	8.0		
W_BOMBA NaOH[0]	Word	0.0	16#0	
W_BOMBA NaOH[1]	Word	2.0	16#0	
R_BOMBA NaOH	Array [0..1] of Word	12.0		
R_BOMBA NaOH[0]	Word	0.0	16#0	
R_BOMBA NaOH[1]	Word	2.0	16#0	
W_BOMBA H3B3	Array [0..1] of Word	16.0		
W_BOMBA H3B3[0]	Word	0.0	16#0	
W_BOMBA H3B3[1]	Word	2.0	16#0	
R_BOMBA H3B3	Array [0..1] ...	20.0		
R_BOMBA H3B3[0]	Word	0.0	16#0	
R_BOMBA H3B3[1]	Word	2.0	16#0	

CONCLUSIONES:

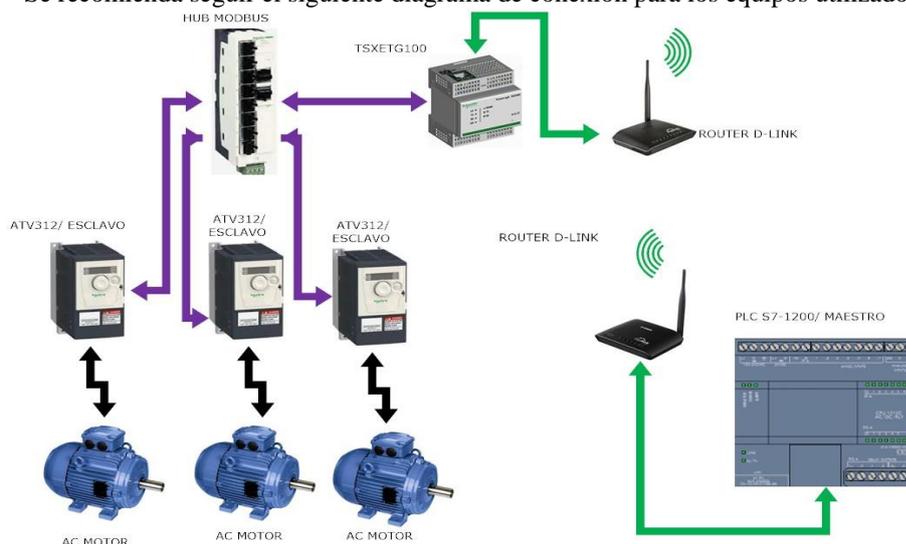
En esta práctica se realiza el control de directo de los ATV312 para la simulación de un proceso de dosificación. El programa presentado no tiene un límite de tiempo máximo de calibración para las bombas, pero sin embargo en la aplicación real dependiendo de las muestras los tiempos no son mayores a 1 minuto.

La programación presentada en este documento refiere a una bomba dosificadora de agua, en la aplicación son 3 bombas distintas pero la programación es la misma con excepción que las marcas utilizadas en cada una son diferentes.

La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

RECOMENDACIONES:

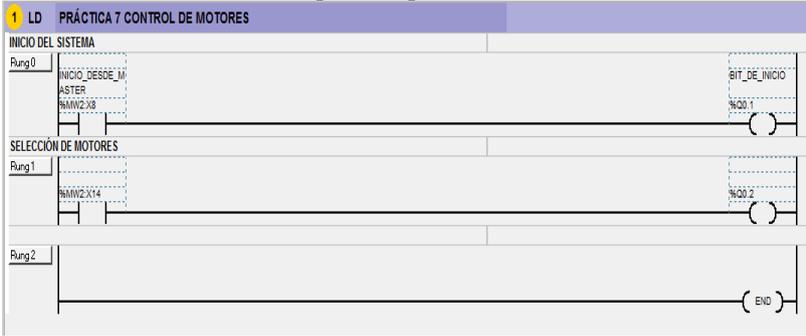
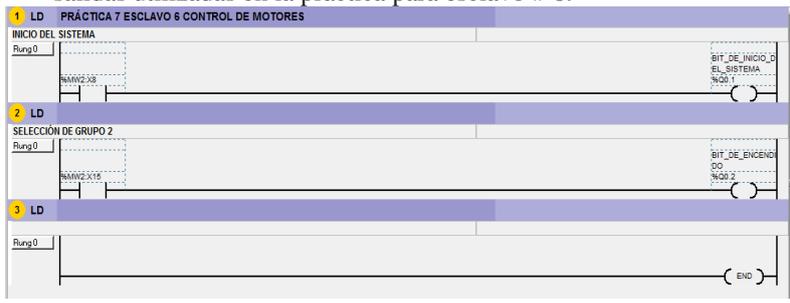
- Verificar el estado de los cables de conexión Modbus.
- Setear el tiempo de requerimiento de los bloques Mb_Client de acuerdo a la aplicación.
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.
- Se recomienda seguir el siguiente diagrama de conexión para los equipos utilizados.



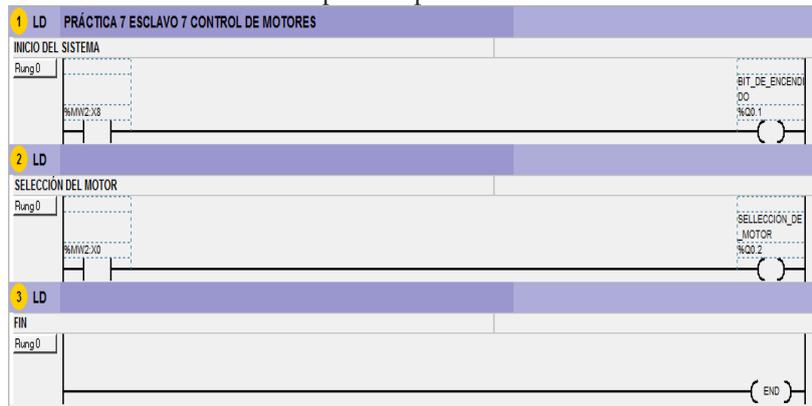
Docente / Técnico Docente: _____

Firma: _____

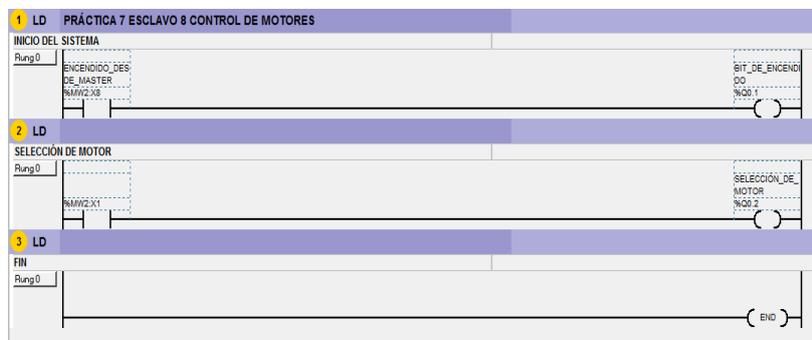
4.7. Práctica 7

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: Ing. Electrónica		ASIGNATURA: Informática Industrial/ Redes III	
NRO. PRÁCTICA:		7	TÍTULO PRÁCTICA: Activación y configuración de velocidades de cuatro variadores de frecuencia ATV312 mediante un PLC SIEMENS y un HMI SIEMENS KTP600
OBJETIVOS: <ul style="list-style-type: none"> • Entrenar al estudiante con los bloques Modbus TCP Siemens. • Comprender el mapeo de registro de ambas marcas de PLC's. • Monitoreo de datos enviados desde el cliente hacia los esclavos. • Manejo de envío/recepción de palabras. • Manejo y gestión de macro drive de PLC Twido. 			
INSTRUCCIONES:		1. Conexión de los dispositivos esclavos al HUB Modbus del módulo convertidor.	
		2. Mediante el menú de pantalla del ATV312 se configura dirección, velocidad de transmisión, paridad y tiempo de comunicación esto se realiza para cada uno de los dispositivos con las direcciones correspondientes	
		3. Configuración y programación de esclavos. <ul style="list-style-type: none"> • Configuración de hardware Twido. • Selección de PLC compacto TWDAA24DRF. • Selección de cartucho de memoria RTC. • Selección de puerto auxiliar de comunicación serie. • Configuración de parámetros de macro drives como, inicio de palabras de control desde Twido y máxima valor de palabras escritas. • Programación de circuito ladder. • PLC TWIDO 1: Inicialización del sistema y selección de marcas y salidas utilizadas en la práctica para esclavo # 5. 	
			
		<ul style="list-style-type: none"> • PLC TWIDO 2: Inicialización del sistema y selección de marcas y salidas utilizadas en la práctica para esclavo # 6. 	
			

- **PLC TWIDO 3:** Inicialización del sistema y selección de marcas y salidas utilizadas en la práctica para esclavo # 7.

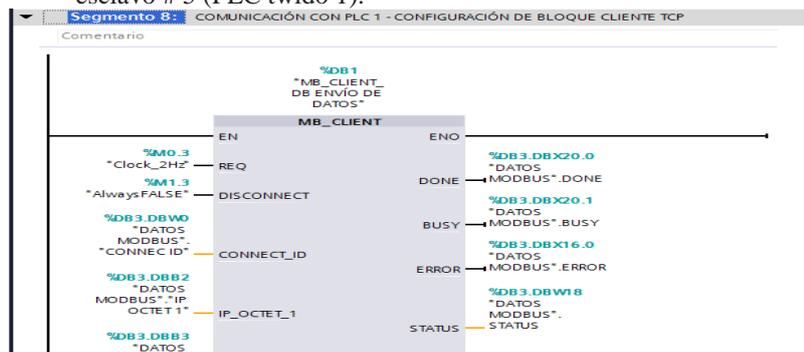


- **PLC TWIDO 4:** Inicialización del sistema y selección de marcas y salidas utilizadas en la práctica para esclavo # 8.

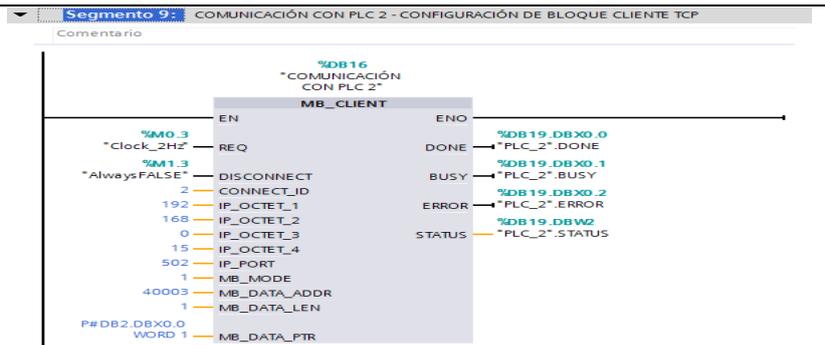


4. Configura y programación PLC cliente.

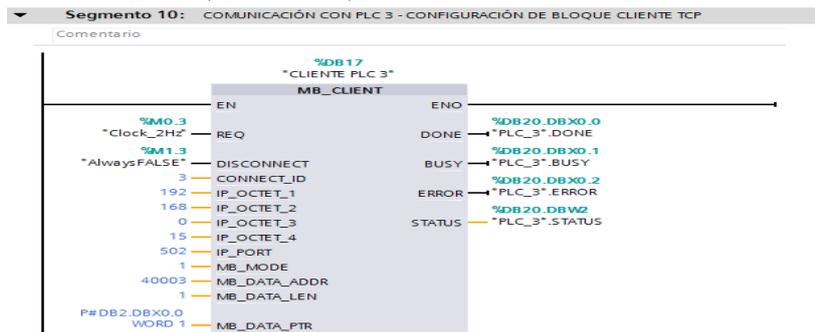
- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 5 (PLC twido 1).



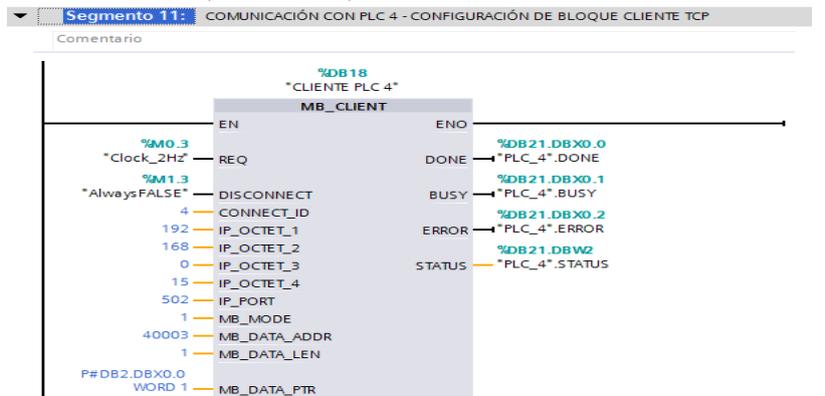
- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 6 (PLC twido 2).



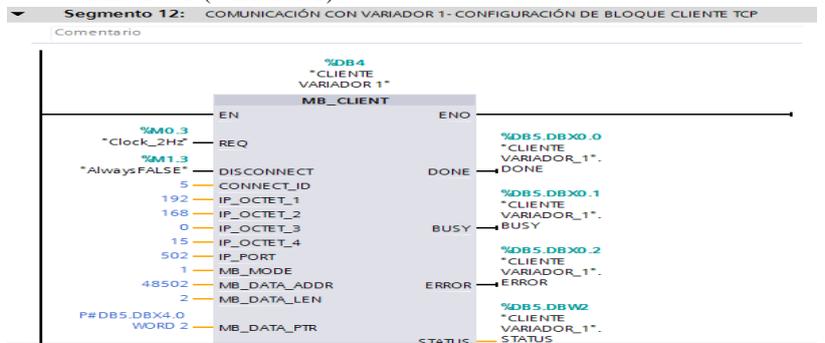
- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 7 (PLC twido 3).



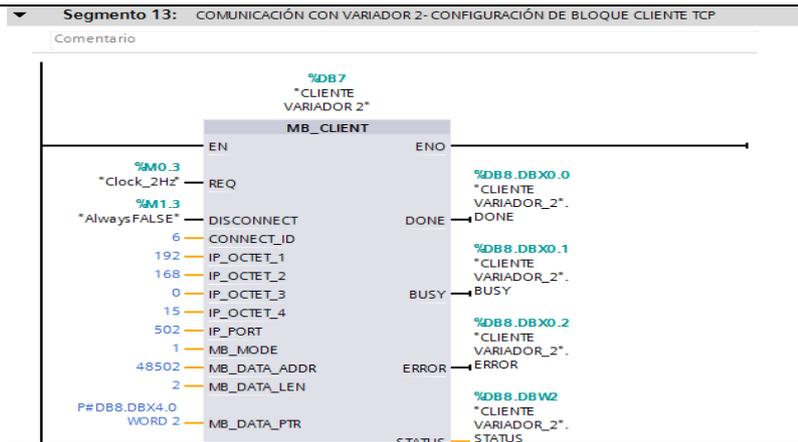
- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 8 (PLC twido 4).



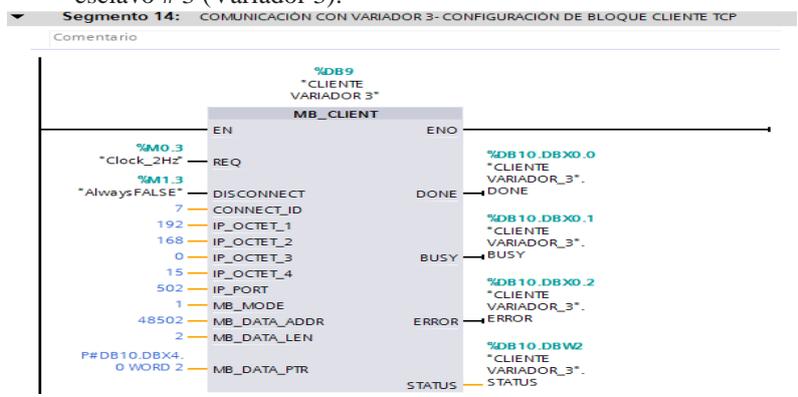
- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 1 (Variador 1).



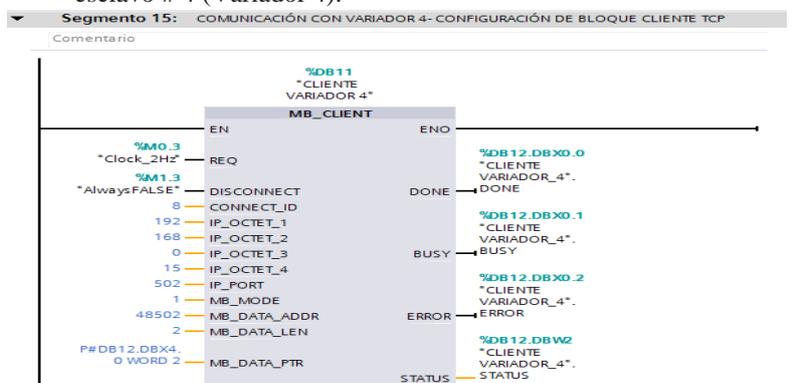
- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 2 (Variador 2).



- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 3 (Variador 3).



- Configuración de datos de bloque MB_Cliente de S7-1200 para esclavo # 4 (Variador 4).



5. Programación de HMI.

- Pantalla de presentación.



- Pantalla principal de configuración de velocidad de motores.



- Se programa botones de encendido y apagado del sistema.
- El sistema debe estar en ON para comenzar a transmitir.
- Se programan los botones de seteo de velocidad.
- Tanto en la imagen raíz como en la imagen AUTORES se puede acceder a la imagen denominada “PANTALLA PRINCIPAL” pulsando e Sobre la pantalla principal tenemos el botón “ON” por medio de cual encendemos el sistema.
- Una vez encendido el sistema se procede a seleccionar el motor que se desea poner en funcionamiento con los botones “MOTOR 1”, “MOTOR 2”, “MOTOR 3” Y “MOTOR 4”.
- Luego de seleccionar el motor se procede a seleccionar una velocidad en los campos numéricos en un rango de 0-1600 rpm (un valor por encima de 1600 no es aceptado por el sistema y el campo numérico se mantendrá en cero).
- Una vez seleccionada la velocidad de trabajo, se procede a presionar el botón mostrado en la parte inferior para el accionamiento del drive.
- Al presionar las teclas F1 para el motor 1, F2 para el motor 2, F3 para el motor 3 y F4 para el motor 4, el PLC maestro (SIEMENS S7 1200) enviará una señal para los respectivos dispositivos esclavos (PLC’S TWIDO Y VARIADORES ATV312).
- En cualquier pantalla se observa un botón denominado “REGRESAR” con el cual podemos acceder a la pantalla inmediatamente anterior.
- El botón “OFF” apaga todo el sistema el botón con el mismo nombre.

ACTIVIDADES POR DESARROLLAR

1. Verificar los valores enviados en los bloques Macro Drive del esclavo.
2. Gestionar el control del variador individualmente con cada PLC Twido.
3. Implementar un control directo entre el PLC cliente y los variadores de frecuencia, sin necesidad de Macro Drive.
4. Monitorear las condiciones de funcionamiento de los motores.

RESULTADO(S) OBTENIDO(S):

En esta práctica se obtiene los datos de tiempo escritos en los variadores, velocidad y estado de funcionamiento de los variadores (encendido o apagados).

VARIABLES DEL SISTEMA				
	Nombre	Tipo de datos	Valor de arranq...	Valor de observación
	Static			
	VARIABLE	Array[0..5] of Word		
	VARIABLE[0]	Word	0	16#1F00
	VARIABLE[1]	Word	16#0	16#0000
	VARIABLE[2]	Word	16#0	16#0000
	VARIABLE[3]	Word	16#0	16#0000
	VARIABLE[4]	Word </td <td>16#0</td> <td>16#0000</td>	16#0	16#0000
	VARIABLE[5]	Word	16#0	16#0000

CONCLUSIONES:

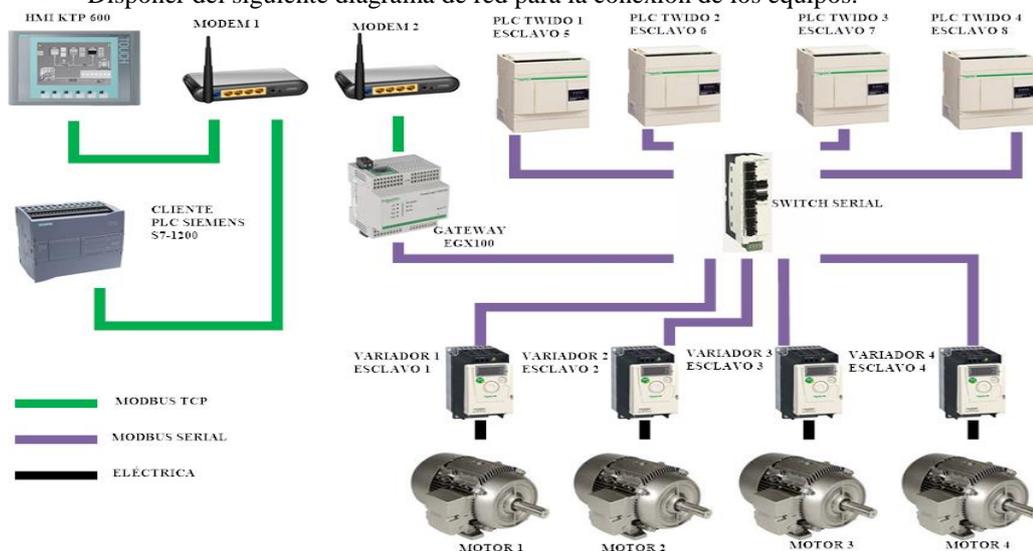
En esta práctica se muestra la programación de una pantalla HMI para controlar el encendido, configuración de velocidades y apagado de cuatro motores trifásicos por medio de una red de comunicación Modbus TCP inalámbrica, teniendo en cuenta que los equipos controlados trabajen bajo una red Modbus RTU.

Se demuestra la configuración de los bloques correspondientes para la comunicación y la aplicación de subrutinas en el caso del PLC Twido.

La configuración del direccionamiento del esclavo RTU se encuentra en el anexo 3 de este documento.

RECOMENDACIONES:

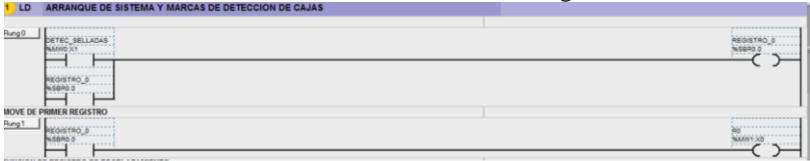
- La frecuencia de envío de datos del bloque Mb_Client, debe ser al menos el doble del tiempo de respuesta del esclavo.
- Revisar el estado de los cables que no se encuentren con falsos contactos.
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.
- Disponer del siguiente diagrama de red para la conexión de los equipos.

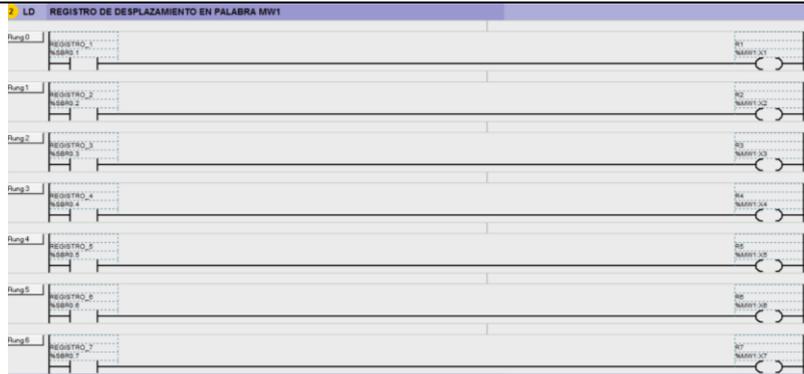


Docente / Técnico Docente: _____

Firma: _____

4.8. Práctica 8

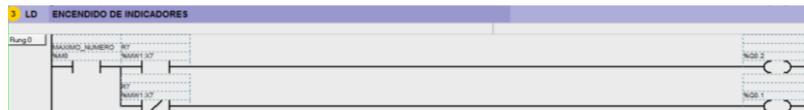
		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: Ing. Electrónica		ASIGNATURA: Informática Industrial/ Redes III	
NRO. PRÁCTICA:		8	TÍTULO PRÁCTICA: Contador e identificación de cajas.
OBJETIVOS: <ul style="list-style-type: none"> • Utilización de bloque SBR del PLC Twido. • Escritura y lectura de datos de control y bits desplazados. • Simulación de proceso de conteo y clasificación de cajas. • Monitorización de datos escritos. • Control de proceso desde HMI. 			
INSTRUCCIONES:		1. Introducción de funcionamiento del programa. Este programa hace uso de un bloque de registro de desplazamiento por parte del esclavo, el valor actual que lleve el bloque de registro de desplazamiento (SBR) del Twido, será leído por el maestro y mostrado en la pantalla HMI. Se simula un programa de desplazamiento de cajas, en donde se dejan caer de un deposito dos tipos de cajas (abiertas y cerradas), las cuales son captadas por sensores, uno que se encarga de contar las cajas y otro que reconoce el tipo de caja que es (abierta o cerrada). El registro de desplazamiento se hace cuando se detecta una caja cerrada ya que esto envía un valor de 1 hacia el bloque SBR y cada vez que se pulse el botón desplazar, el pistón se mueve y la caja se desplaza en HMI y el valor de 1 se desplaza hacia el siguiente bit del bloque SBR. Las cajas abiertas representan el valor de 0, por lo que al desplazar estas cajas tan solo se moverá el valor cero en los bits del bloque SBR, este bloque tiene un conteo máximo de 16 bits por lo que el HMI mostraba un máximo de 16 cajas contadas en formato decimal.	
		2. Configuración y programación de esclavo. <ul style="list-style-type: none"> • Configuración de hardware Twido y bus de comunicación Modbus. • En la pestaña de programación se ingresa el primer LD, en donde se programa la instancia de inicio de la palabra de envío del cliente en el bit MW0:X1 para la detección de cajas cerradas, esto activara el primer bit del registro de desplazamiento SBR0.0 y esta señal es auto enclavada con una marca del mismo registro.  <ul style="list-style-type: none"> • Adicional se incluye un rung en donde se transfiere el dato del SBR0.0 a la palabra MW1:X0 para que esta palabra pueda ser leída por el cliente. • Cada bit del registro de desplazamiento es transferido al bit correspondiente de la palabra MW1 y que este dato pueda ser leído por el cliente. 	



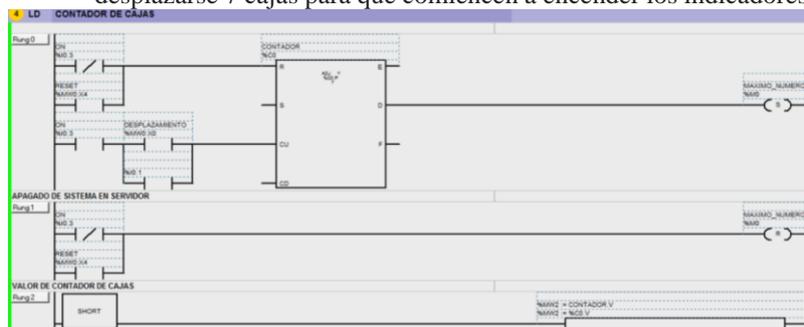
- El bloque de registro de desplazamiento consta de marcas que son escritas desde el HMI del cliente y de un selector %I0.3 que hace permitirá que el bloque SBR se active.



- Posteriormente se programa una sentencia para la activación de indicadores de luces, que mostraran el tipo de caja que está siendo detectadas.



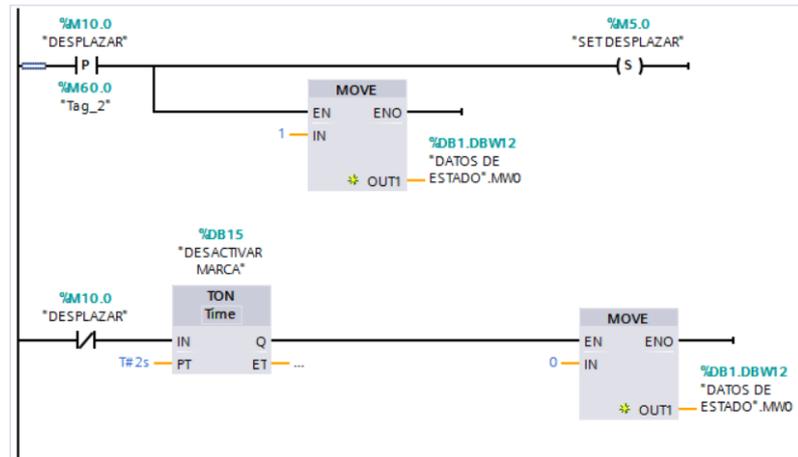
- Adicional se incluye un contador de cajas el cual cumple dos funciones, la primera es almacenar y que el cliente pueda leer el número de cajas que han sido detectadas (hasta 16), y la segunda es que al momento de contabilizar 7 cajas, la salida de este contador activara las marca M0 que permitirá que se pueda accionar los indicadores luminosos, esto a que al inicio del programa deben desplazarse 7 cajas para que comiencen a encender los indicadores.



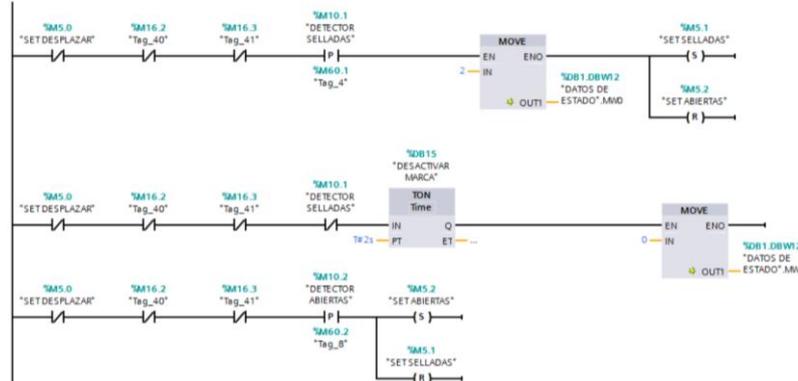
3. Configura y programación PLC cliente.

- Se selecciona el PLC con serie y versión disponible.
- Se selecciona las marcas de ciclo del PLC, la cuales corresponden a la dirección de la palabra MB0. Y las marcas del sistema en la palabra MB1.

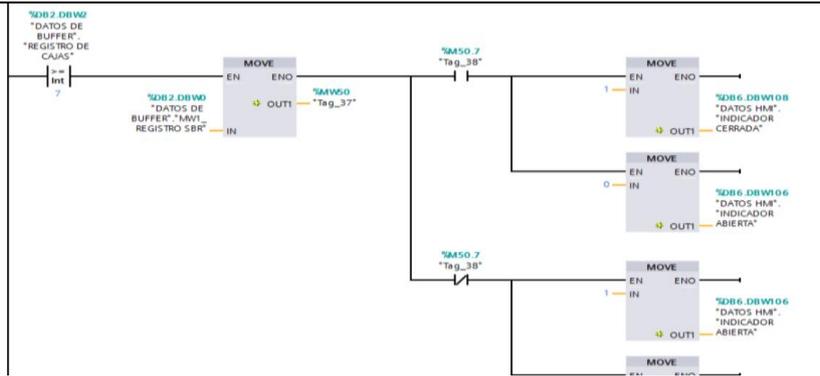
- La programación del cliente consta del envío de datos de control por medio del HMI para la activación del bloque de registro de desplazamiento del servidor.
- Se crea un FC para las funciones de envío/recepción de control.
- En el primer segmento se envía la palabra de desplazamiento del pistón del HMI, este mismo pulso activa el desplazamiento del bloque SBR del servidor, en esta misma línea se realiza la desactivación de la marca del HMI.



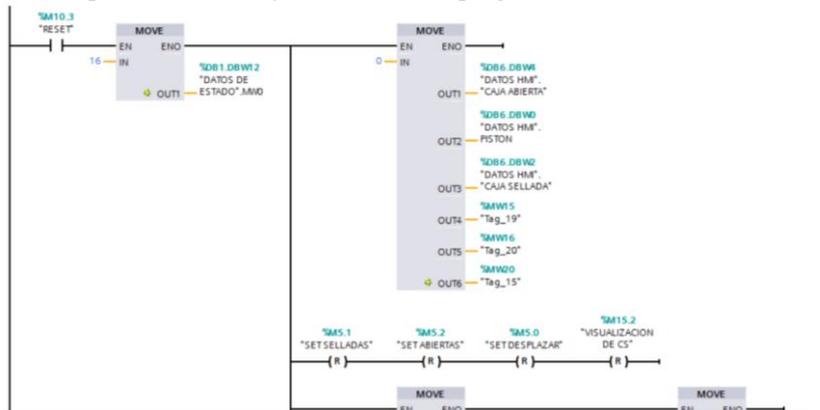
- En el segundo segmento se programan los pulsos para dejar caer las cajas abiertas y cerradas, cada una con su respectiva marca de activación.



- La marca de activación 1 (caja cerrada) del bloque %SBR0.0 del servidor se realiza mediante la marca %M10.1 del cliente, mientras que la otra marca deja caer las cajas abiertas (%M10.2).
- En el siguiente segmento se programa el encendido de los indicadores correspondiente de acuerdo al tipo de caja detectada, esto se realiza leyendo el estado del bit MW1:X7 del servidor.
- El bloque de lectura, está programado para leer toda la palabra MW1 por lo que el valor leído se transfiere a la palabra MW50 en donde dependiendo del estado del bit M50.7 se determina el estado del bit.



- En el último segmento se hace el reset y puesta en cero de todas las palabras escritas y utilizadas en la programación.



- Se crea otro FC para la programación de animaciones en el HMI, en este bloque se programa el movimiento del pistón y de las cajas, se muestra el número de cajas y el desplazamiento de las mismas.

4. Programación de HMI.

- Se programa la pantalla de presentación.



- Se programan 3 botones para el inicio del proceso de desplazamiento y dos cuadros de valores leídos del bloque de desplazamiento del servidor.
- Cada botón cumple su función independiente, uno para dejar caer cajas cerradas, otro botón para cajas abiertas y un tercer botón para dar movimiento al pistón de desplazamiento.

- Estos botones están unidos al bloque de transferencia de datos, ya que al pulsar uno de ellos, su función se verá reflejada en el servidor, es decir se transfiere el dato que se visualiza en el HMI, sea de número de caja, orden de desplazamiento y detección del tipo de caja.
- Se incluye adicionalmente un botón de reset para poner el contador y registros en cero.

ACTIVIDADES POR DESARROLLAR

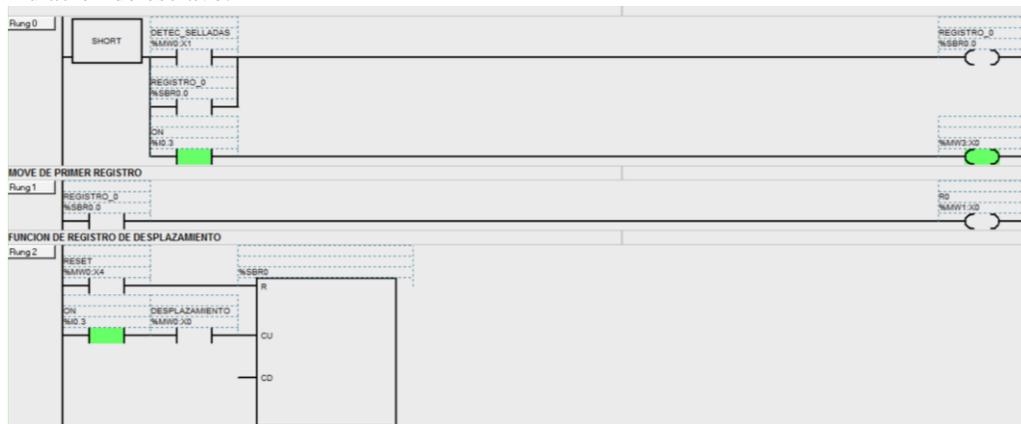
1. Generar un HMI con un registro de hasta 16 cajas.
2. Programar dos bloques SBR en cascada, en el servidor y que estos valores sean leídos por el cliente.
3. Diseñar un proceso de conteo y desplazamiento con dos PLC esclavos.

RESULTADO(S) OBTENIDO(S):

En esta práctica se obtienen la animación de cada registro desplazado en el bloque SBR del servidor, adicional la simulación de un proceso de conteo de cajas y detección del tipo de caja. Se muestra en la pantalla un valor entero correspondiente al número de cajas y valores booleanos como indicación del tipo de caja.

DATOS DE BUFFER			
Nombre	Tipo de datos	Offset	Valor de arranq...
Static			
MW1_REGISTRO SBR	Word	0.0	16#0
REGISTRO DE CAJAS	Word	2.0	16#0
BIT %I0.3	Bool	4.0	false

Simulación de esclavo.



CONCLUSIONES:

En esta práctica se hace la utilización de un bloque de registro de desplazamiento del PLC Twido, y este valor poder leerlo en una pantalla HMI y simular el desplazamiento de los bits por medio de las cajas.

La sincronización entre la simulación del HMI y el PLC esclavo se pudo lograr con el ajuste preciso de la velocidad de transmisión de datos del bloque Mb_Cliente.

No se adjunta imágenes de la programación del HMI, ya que esta práctica no pretende explicar la programación de la animación de la pantalla.

RECOMENDACIONES:

- Verificar el estado de los cables de conexión de los equipos Modbus.
- Se recomienda realizar un diagrama de direccionamiento, para un mejor entendimiento del mapeo de palabras entre maestro-esclavo.

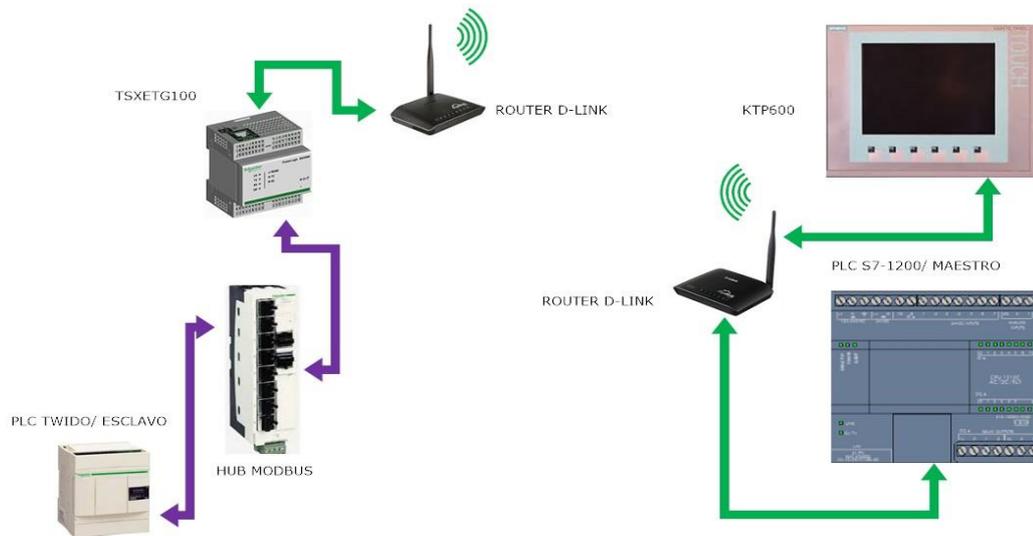
PALABRA DE CONTROL		
S7-1200	TWIDO	ESTADO

%M10.0	%MW0:X0	DESPLAZAMIENTO
%M10.1	%MW0:X1	CAJA CERRADA
%M10.3	%MW0:X4	RESET

REGISTRO DE DESPLAZAMIENTO		
S7-1200	TWIDO	ESTADO
%DB2.DBX 0.0	%MW1:X0	SBR0.0
	%MW1:X1	SBR0.1
	%MW1:X2	SBR0.2
	%MW1:X3	SBR0.3
	%MW1:X4	SBR0.4
	%MW1:X5	SBR0.5
	%MW1:X6	SBR0.6
	%MW1:X7	SBR0.7

CONTADOR DE CAJAS		
S7-1200	TWIDO	ESTADO
%DB2.DBX 4.0	%MW2	CONTADOR DE CAJAS

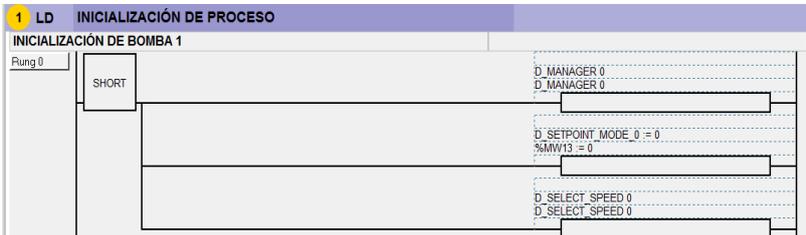
- Se recomienda seguir la referencia de conexión del siguiente diagrama de red de los equipos.

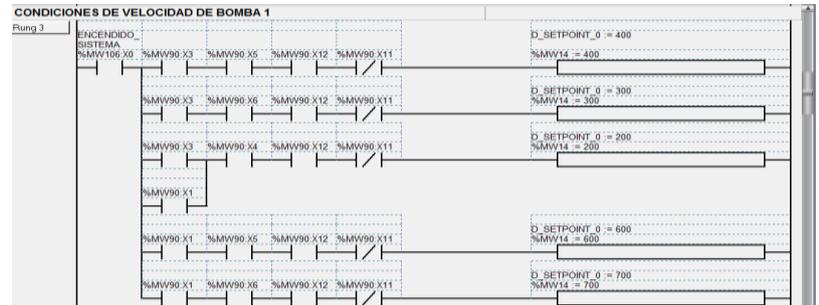


Docente / Técnico Docente: _____

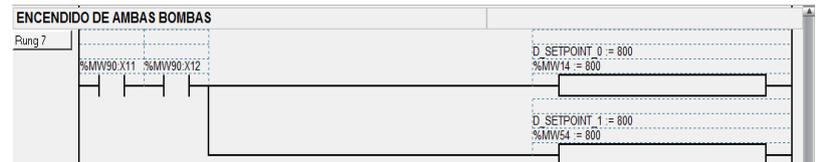
Firma: _____

4.9. Práctica 9

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III	
NRO. PRÁCTICA:	9	TÍTULO PRÁCTICA: Control de sistema hidráulico de llenado de un tanque
OBJETIVOS: <ul style="list-style-type: none"> • Entrenar al estudiante con la simulación de procesos industriales • Comprender el mapeo de registro de ambas marcas de PLC's. • Monitoreo de datos enviados desde el cliente hacia los esclavos. • Manejo de envío/recepción de palabras. • Manejo y gestión de macro drive de PLC Twido. • Configuración de equipos de comunicación inalámbricos. • Configuración de Gateway de protocolos de comunicación industrial. 		
INSTRUCCIONES:	1. Conexión del PLC Twido al HUB Modbus del módulo convertidor.	
	2. Configuración de variadores esclavos RTU. <ul style="list-style-type: none"> • Mediante el menú de pantalla del ATV312 se configura dirección, velocidad de transmisión, paridad y tiempo de comunicación esto se realiza para cada uno de los dispositivos con las direcciones correspondientes. 	
	2. Configuración y programación de esclavo. <ul style="list-style-type: none"> • Configuración de hardware Twido. • Selección de PLC compacto TWDLCAA24DRF. • Selección de cartucho de memoria RTC. • Selección de puerto auxiliar de comunicación serie TWDNAC485T. • Programación de circuito ladder. Esclavo Twido # 5. • LD1: Inicialización de parámetros de control para VDF's ATV312. Sentencias obligadas que deben activarse para la comunicación con macro drives. 	
		
<ul style="list-style-type: none"> • LD1: Encendido del sistema (Rung 2). Condiciones de velocidad del grupo de bombeo (Rung 3 – Rung 4). Arranque de motor de grupo de bombeo (Rung 7 – Rung 8). 		

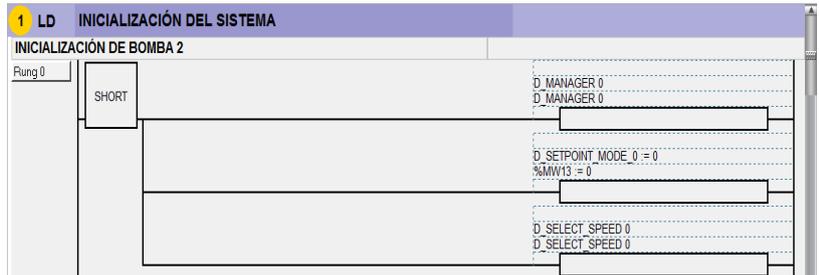


- LD2: detención de motor mediante marcas del sistema y límite máximo de velocidad establecida en el programa. Se finaliza el programa con la sentencia **END** para poder establecer una subrutina para la escritura de incremento de velocidad.

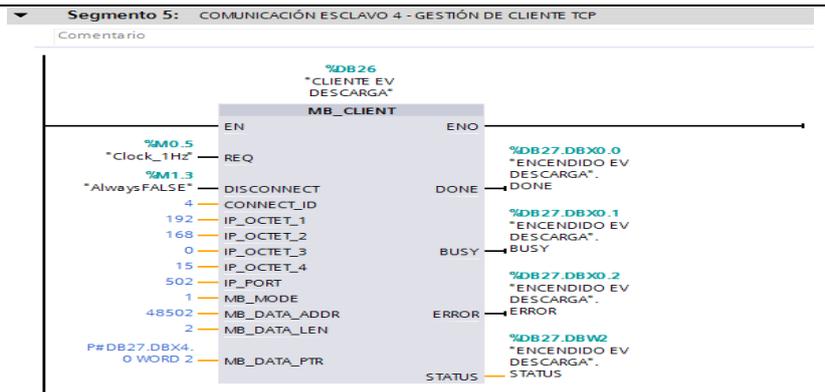


Esclavo Twido # 4.

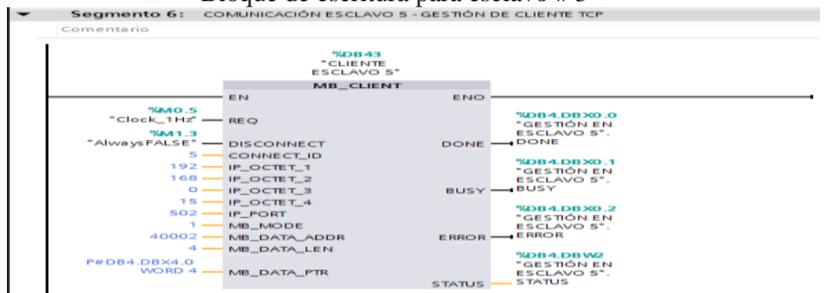
- LD1: Inicialización de parámetros de control para VDF's ATV312. Sentencias obligadas que deben activarse para la comunicación con macro drives.



- LD1: Encendido del sistema (Rung 1). Condiciones de velocidad del grupo de bombeo (Rung 2 – Rung 3). Arranque de motor de grupo de bombeo (Rung 4 – Rung 5). Parada de motor mediante marcas del sistema y límite máximo de velocidad establecida en el programa (Rung 6). Se finaliza el programa con la sentencia **END** para poder establecer una subrutina para la escritura de incremento de velocidad (Rung 7).



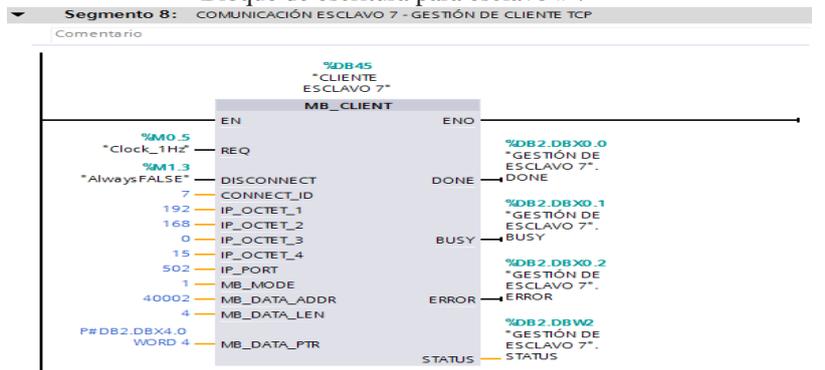
- Bloque de escritura para esclavo # 5



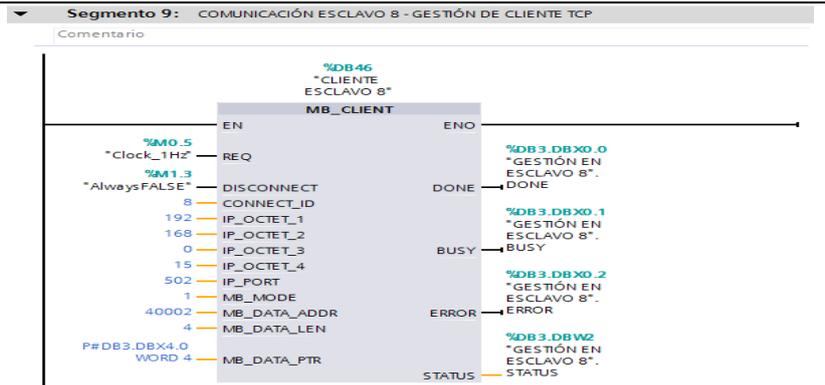
- Bloque de escritura para esclavo # 6



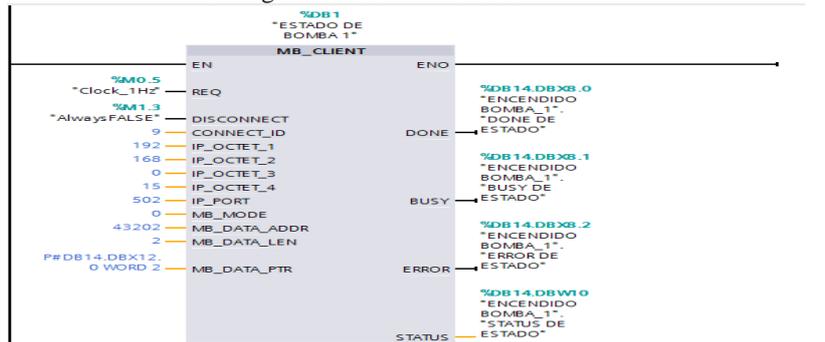
- Bloque de escritura para esclavo # 7



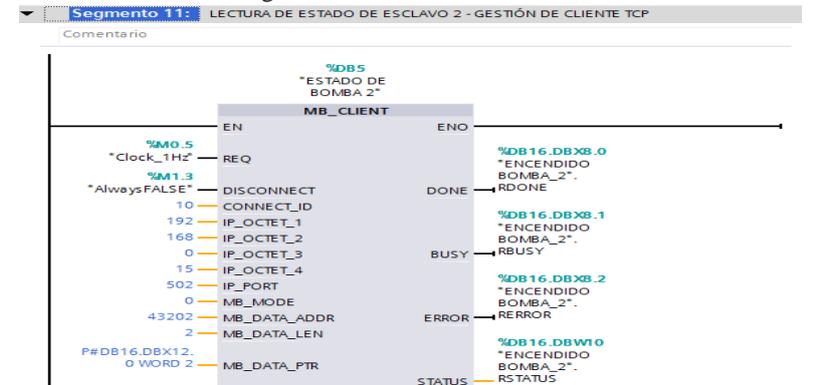
- Bloque de escritura para esclavo # 8



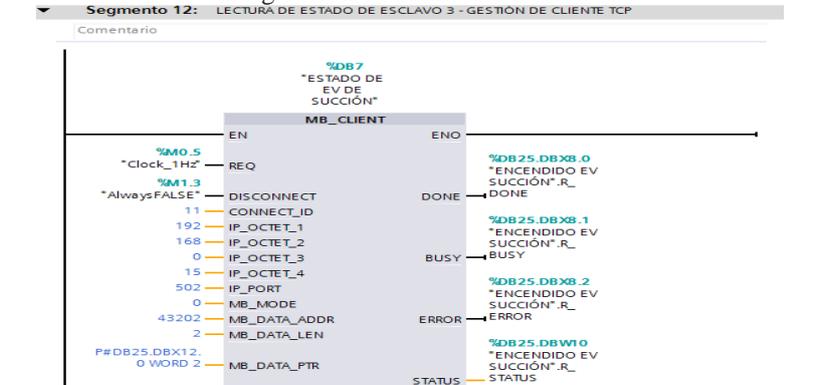
- Configuración de monitoreo de esclavos # 1



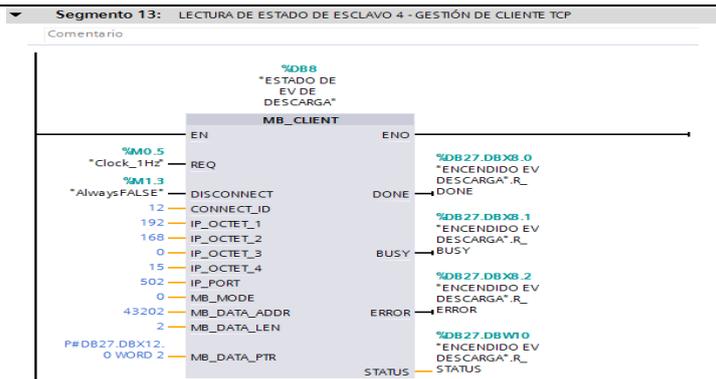
- Configuración de monitoreo de esclavos # 2



- Configuración de monitoreo de esclavos # 3



- Configuración de monitoreo de esclavos # 4



- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 1

Proyecto1 ▶ PLC_1 [CPU 1214C AC/DC/Rly] ▶ Bloques de program

ENCENDIDO BOMBA_1

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	DATOS DE CONTROL	Array[0..1] of Word	
7	DONE DE ESTADO	Bool	false
8	BUSY DE ESTADO	Bool	false
9	ERROR DE ESTADO	Bool	false
10	STATUS DE ESTADO	Word	16#0
11	ESTADO DE OPERACIÓN	Array[0..1] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 2

Proyecto1 ▶ PLC_1 [CPU 1214C AC/DC/Rly] ▶ Bloques de program

ENCENDIDO BOMBA_2

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	DATOS DE CONTROL	Array[0..1] of Word	
7	RDONE	Bool	false
8	RBUSY	Bool	false
9	RERROR	Bool	false
10	RSTATUS	Word	16#0
11	LECTURA DE ESTADO	Array[0..1] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 3

Proyecto1 > PLC_1 [CPU 1214C AC/DC/Rly] > Bloques de programaci...

ENCENDIDO EV DESCARGA

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	DATOS DE CONTROL	Array[0..1] of Word	
7	R_DONE	Bool	false
8	R_BUSY	Bool	false
9	R_ERROR	Bool	false
10	R_STATUS	Word	16#0
11	LECTURA DE ESTADO	Array[0..1] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 4

Proyecto1 > PLC_1 [CPU 1214C AC/DC/Rly] > Bloques de programaci...

ENCENDIDO EV SUCCIÓN

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	DATOS DE CONTROL	Array[0..1] of Word	
7	R_DONE	Bool	false
8	R_BUSY	Bool	false
9	R_ERROR	Bool	false
10	R_STATUS	Word	16#0
11	LECTURA DE ESTADO	Array[0..1] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 5

Proyecto1 > PLC_1 [CPU 1214C AC/DC/Rly] > Bloques de programaci...

GESTIÓN EN ESCLAVO 5

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	ESCRITURA EN ESCLAV...	Array[0..3] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 6

Proyecto1 > PLC_1 [CPU 1214C AC/DC/Rly] > Bloques de programaci...

GESTIÓN EN ESCLAVO 6

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	ESCRITURA EN ESCLAV...	Array[0..3] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 7

Proyecto1 ▶ PLC_1 [CPU 1214C AC/DC/Rly] ▶ Bloques de program

	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	ESCRITURA EN ESCLAV...	Array[0..3] of Word	

- Creación de bloque de datos y direccionamiento de datos escritos en esclavo # 8

Proyecto1 ▶ PLC_1 [CPU 1214C AC/DC/Rly] ▶ Bloques de program

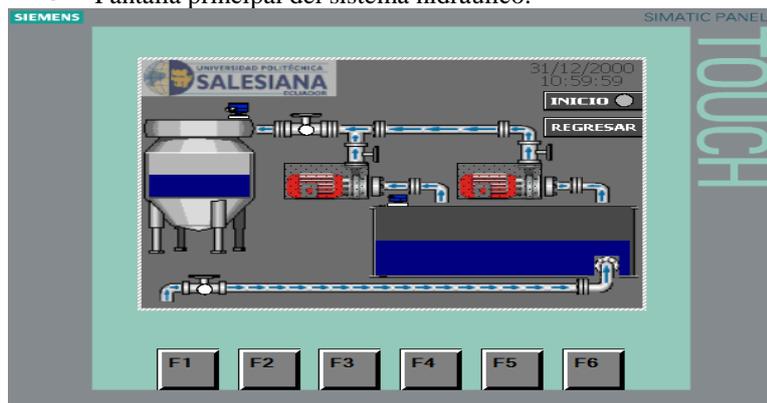
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	DONE	Bool	false
3	BUSY	Bool	false
4	ERROR	Bool	false
5	STATUS	Word	16#0
6	ESCRITURA EN ESCLAV...	Array[0..3] of Word	

4. Programación de HMI.

- Pantalla de presentación.



- Pantalla principal del sistema hidráulico.



- Pantalla de estado de grupo de bombeo # 1



- Pantalla de estado de grupo de bombeo # 2



- Al presionar sobre el botón "AUTORES" se activa la imagen denominada con el mismo nombre.
- Tanto en la imagen PÁGINA DE INICIO como en la imagen AUTORES se puede acceder a la imagen denominada "PANTALLA PRINCIPAL" pulsando el botón con el mismo nombre. En el caso de la imagen AUTORES se puede acceder a la imagen PANTALLA PRINCIPAL presionando también la tecla F1 y se puede regresar a la imagen anterior presionando el botón "REGRESAR" o la tecla F6.
- Sobre la pantalla principal tenemos el botón "INICIO" por medio de cual encendemos el sistema.
- Una vez encendido el sistema se puede apreciar el funcionamiento del mismo que consiste en la variación de nivel y accionamiento de los grupos de bombeo.
- Al hacer un clic sobre los grupos de bombeo se nos va a activar una ventana en la cual podremos apreciar la información de cada grupo de bombeo, así como también podremos habilitar y deshabilitar los grupos y también resetear la fallas presentes en las bombas.
- En cualquier pantalla se observa un botón denominado "REGRESAR" con el cual podemos acceder a la pantalla inmediatamente anterior.

ACTIVIDADES POR DESARROLLAR

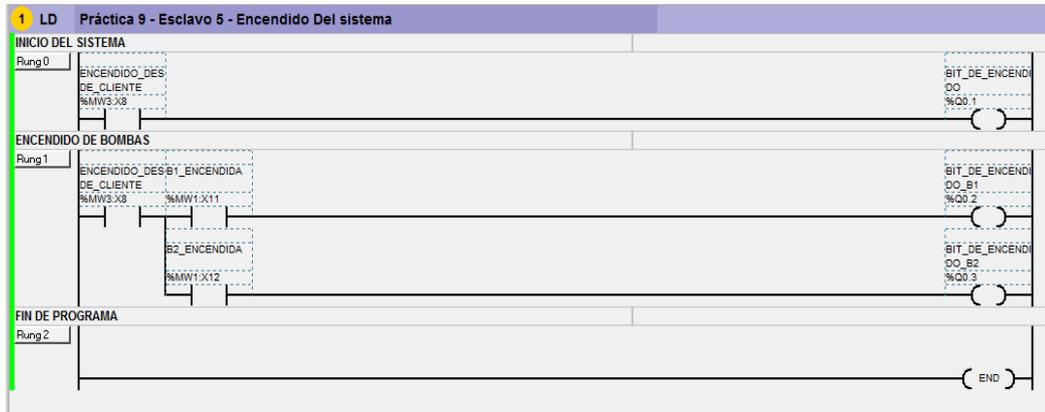
1. Generar un HMI con un registro de hasta 16 cajas.
2. Programar dos bloques SBR en cascada, en el servidor y que estos valores sean leídos por el cliente.
3. Diseñar un proceso de conteo y desplazamiento con dos PLC esclavos.

RESULTADO(S) OBTENIDO(S):

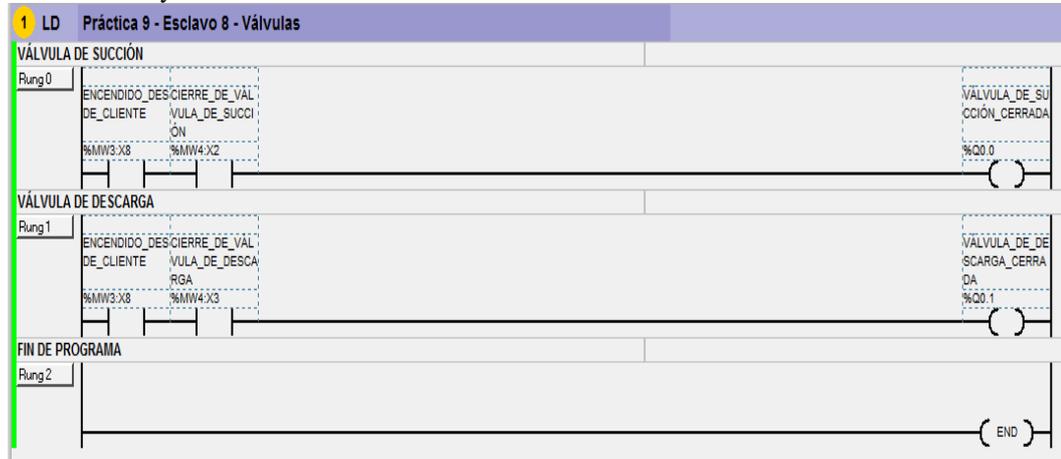
Escritura de Holding registers desde HMI Siemens KTP600.
 Recepción de Holding registers en dos PLC Twido conectados como esclavo.
 Simulación de un sistema hidráulico de llenado automático.

LECTURA DE BIT	Array[0..3] of Word	
LECTURA DE BIT[0]	Word	16#0
LECTURA DE BIT[1]	Word	16#0
LECTURA DE BIT[2]	Word	16#0
LECTURA DE BIT[3]	Word	16#0
LECTURA ESCLAVO 4	Array[0..3] o...	
LECTURA ESCLAVO...	Word	16#0

Encendido del sistema



Modo Stand by de válvulas



CONCLUSIONES:

En esta práctica se muestra la programación de una pantalla HMI y un PLC Siemens S71200 para controlar el encendido, configuración de velocidades y apagado de motores trifásicos por medio de una red de comunicación Modbus TCP inalámbrica, teniendo en cuenta que los equipos controlados trabajan bajo una red Modbus RTU.

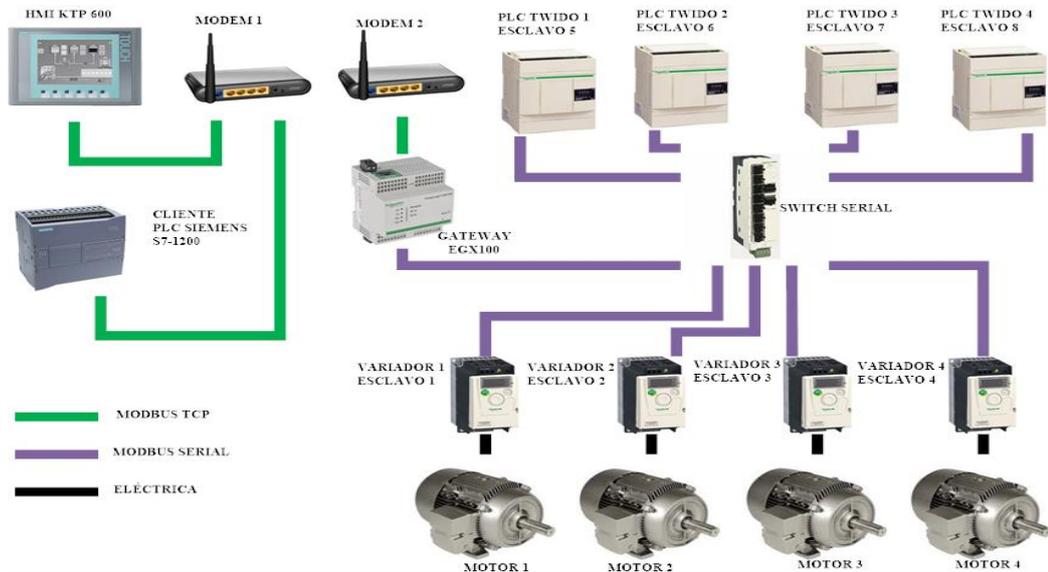
Se demuestra la configuración de los bloques correspondientes para la comunicación y la aplicación de subrutinas en el caso del PLC Twido.

RECOMENDACIONES:

- Se recomienda disponer del diagrama de direccionamiento adjunto.

VARIABLES UTILIZADAS EN PLC MAESTRO SIEMENS S7 1200					
ENTRADAS	NOMBRES EN PROGRAMA	DESCRIPCIÓN	SALIDAS	NOMBRES EN PROGRAMA	DESCRIPCIÓN
I0.0	ON	ENCENDIDO DEL SISTEMA	Q0.0	OPERACIÓN BOMBA 1	OPERACIÓN BOMBA 1
I0.1	OFF	APAGADO DEL SISTEMA	Q0.1	OPERACIÓN BOMBA 2	OPERACIÓN BOMBA 2
I0.2	NO USADA	NO USADA			
I0.3	NO USADA	NO USADA			
I0.4	NO USADA	NO USADA			
I0.5	NO USADA	NO USADA			
I0.6	SIMULACIÓN FALLA B1	FALLA BOMBA 1			
I0.7	SIMULACIÓN FALLA B2	FALLA BOMBA 2			
I1.0	SIMULACIÓN DESH B1	DESHABILITA BOMBA 1			
I1.1	SIMULACIÓN DESH B2	DESHABILITA BOMBA 2			
I1.2	HAB B1	HABILITAR BOMBA 1			
I1.3	HAB B2	HABILITAR BOMBA 2			
I1.4	ABRIR / CERRAR VALVULA DE SUCCIÓN	ABRIR / CERRAR VALVULA DE SUCCIÓN			
I1.5	ABRIR / CERRAR VALVULA DE DESCARGA	ABRIR / CERRAR VALVULA DE DESCARGA			
I1.6	PARO DE EMERGENCIA	PARO DE EMERGENCIA			
I1.7	NO USADA	NO USADA			

- Verificar el estado de los cables de conexión de los equipos Modbus RTU.
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.
- Se recomienda seguir la referencia de conexión del siguiente diagrama de red de los equipos.



Docente / Técnico Docente: _____

Firma: _____

4.10. Práctica 10

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES
CARRERA: Ing. Electrónica	ASIGNATURA: Informática Industrial/ Redes III	
NRO. PRÁCTICA:	10	TÍTULO PRÁCTICA: Simulación de proceso de dosificación volumétrica.
OBJETIVOS: <ul style="list-style-type: none"> • Utilización de varios esclavos RTU. • Manejo de Macro Drive para control de variadores ATV312. • Direccionamiento de bloques de instrucción desde el cliente. • Animación y control por HMI. • Monitorización de datos escritos. • Manejo de multi esclavos. • Simulación de proceso de dosificación volumétrica. 		
INSTRUCCIONES:	<p>1. Introducción de funcionamiento de la simulación. En los procesos de plantas procesadoras de alimentos y/o balanceados, existen sistemas automatizados para el mezclado de materia prima y llenado de producto final en sacos, los mecanismos usados son tolvas para la dosificación del producto y el llenado puede ser volumétrico o gravimétrico. Están especialmente diseñados para productos granulados y en polvo, adaptándose a gran variedad de densidades y granulometrías.</p> <p>En la dosificación volumétrica generalmente se usan medios mecánicos como sinfines, aunque también pueden ser válvulas rotativas cintas u otros elementos unidos a una tolva fija. Consiste en dosificar el producto a partir de un control por volumen del mismo. Esto se consigue a partir de una velocidad de giro constante que permite el avance constante del material. Si el producto es cohesivo la tolva puede incluir un agitador para favorecer la descarga.</p> <p>Esta simulación comprende el proceso de mezclado de materia prima y la dosificación del producto mediante la técnica de dosificación volumétrica, al dar inicio la aplicación una pantalla principal muestra dos opciones una de servicio y otra para operario, en la ventana de operario se hace el ingreso de parámetros tales como el volumen de llenado de cada producto y el número de saco que se desea llenar.</p> <p>La cantidad de llenado de productos (metros cúbicos) es independiente, es decir, cada producto puede tener un volumen distinto, menor a 2000 y mayor a 300, esto quiere decir que si se introduce un valor menor a 300 el programa no correrá, lo mismo pasa en el límite máximo ya que la pantalla HMI permite ingresar un valor máximo de 999, por lo que será lo máximo que se ingrese aunque la capacidad del tanque sea mayor.</p> <p>En la pantalla en donde corre la simulación se representan válvulas, mezcladoras, sensores y motores.</p> <p>Para el caso de los motores existen tres motores para cada materia prima y un motor para un reactivo que se adiciona para la mezcla con cada materia prima y con ayuda de las mezcladoras se genera el producto correspondiente.</p> <p>Al dar inicio al proceso, arranque el blower de reactivo, el cual se ve visualizado en el HMI y tiene un indicador de encendido a un costado, este blower arranca durante 7 segundos, seguido de esto arranca el motor 1 el cual se indica en el HMI por flechas que se dirigen desde la válvula de dos vías hacia el primer tanque, en esta misma secuencia se accionan las válvulas, en el caso cuando está operando el blower reactivo, se acciona un paso de la válvula de dos vías, en cuanto este se detiene la válvula se cierra y se activa en el otro paso para que se deje caer la materia prima A que es bombeada por el motor 1 y así mismo se enciende la válvula de materia</p>	

prima del tanque 1 ya que es el producto que se está generando, este proceso se repite para los dos productos siguientes de la misma forma, primero arranca el blower reactivo con la indicación de la respectiva válvula y seguidamente arranca el motor de materia prima correspondiente con las válvulas respectivas de los tanques que se estén llenando.

El llenado de los tanques se realiza mediante caída libre y se simula con una variable acumuladora que va sumando una unidad hasta llegar al valor seteado.

En cuanto ya se tienen los tanques con el nivel seteado, se da inicio al llenado de los sacos para el producto final, esto se visualiza en el HMI mediante un botón de activación de una válvula de llenado que es quien controla la cantidad de caída del producto para que llegue al valor de llenado (100 metros cúbicos), esta válvula abre durante el tiempo que se está llenado el saco y se cierra cuando se ha terminado de llenar, luego espera un tiempo a que se ponga un nuevo saco para volver a abrirse, esto lo repite en los tres productos y con la cantidad de sacos que se haya seteado llenar.

En la pantalla de servicio se programa un bloque de pruebas, esto puede ser utilizado como un método de diagnóstico de los componentes conectados, ya que mediante un mando de control (tablero de entradas del maestro), se puede arrancar los motores, mezcladores, válvulas y enviar señal analógica hacia el PLC de los sensores.

2. Configuración de variadores esclavos RTU.

- Mediante el menú de pantalla del ATV312 se configura dirección, velocidad de transmisión, paridad y tiempo de comunicación esto se realiza para cada uno de los dispositivos con las direcciones correspondientes.

2. Configuración y programación de esclavos.

- Se configura el hardware de los esclavos y la red de comunicación Modbus en la que transmitirán.
 - PLC_Sensores

Los datos son transferidos desde el cliente hacia las palabras MW2, MW3 y MW4 respectivamente según el tanque del producto que se esté llenado con materia prima, el valor del nivel de llenado es controlado por el cliente y el encendido de los indicadores de los sensores se realiza en el esclavo, mediante comparaciones.



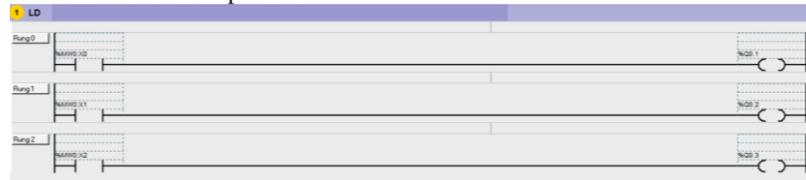
➤ PLC_Mezcladores

Este PLC maneja una sola palabra y los datos que maneja es en función de los bits activados de esta palabra, el cliente envía un valor tipo Word y de acuerdo al valor de la palabra se activan los bits correspondientes para encender los indicadores de los mezcladores.



➤ PLC_Actuadores (Válvulas)

En este bloque se transfieren los datos hacia el esclavo Twido que maneja los indicadores de los actuadores, estos bloques move son activados por la variable HMI correspondiente.



3. Configura y programación PLC cliente.

- Se selecciona el PLC con serie y versión disponible.
- Se selecciona las marcas de ciclo del PLC, la cuales corresponden a la dirección de la palabra MB0. Y las marcas del sistema en la palabra MB1.
- La programación del cliente consta del envío de datos de control por medio del HMI para la activación de la simulación del proceso.
- Se crea un FB para la programación de animación del HMI.
- Se crean 4 FC correspondientes para cada uno de los esclavos RTU.
- Se crean OB's para el manejo de datos del HMI y envío/recepción.
- Adicional se agregan dos FC para las funciones del HMI para la pantalla del operador y la pantalla de servicio. Se programa la animación del proceso y el manejo de muestra de datos en la pantalla, el mapeo de estos datos se encuentran en la tabla inicial.

➤ Bloque Operario

En este bloque se programa toda la comunicación y simulación con el HMI, se utilizan las marcas y palabras variables para la gestión de datos durante la simulación.

➤ Bloque de Servicio

En este bloque se programa un control de prueba de todos los dispositivos utilizados en esta programación, tras pulsar el botón de inicio de servicio, mediante un panel de control (entradas) se simulan datos de activación de los componentes conectados (sensores, mezcladores, motores y válvulas).

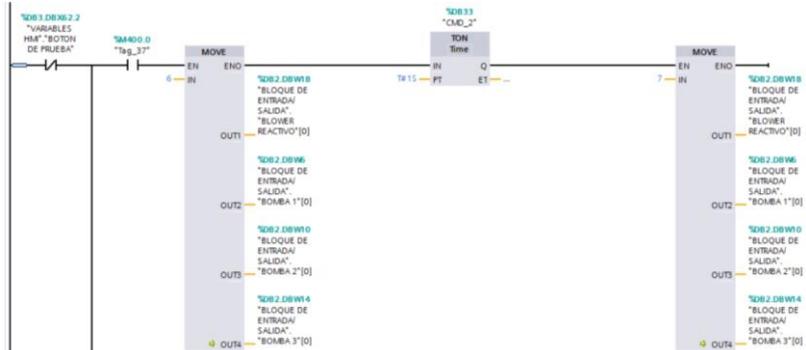
ENTRAD A DE PLC	FUNCION	DIRECCIÓN	ACCION EN SERVIDOR
%I0.6	ACTIVACION DE PRUEBA DE MOTORES	%I0.0	MOTOR 1
		%I0.1	MOTOR 2
		%I0.2	MOTOR 3
		%I0.3	MOTOR 4
%I0.7	ACTIVACION DE SENSORES	%I1.0 - %IW64	SENSOR 1
		%I1.1 - %IW64	SENSOR 2
		%I1.2 - %IW64	SENSOR 3
%I1.0	MEZCLADO RES	%I0.0	MEZCLADOR 1
		%I0.1	MEZCLADOR 2
		%I0.2	MEZCLADOR 3

%I1.1	VÁLVULAS	%I0.0	VÁLVULA 2 VIAS - ESTADO 1
		%I0.1	VÁLVULA 2 VIAS - ESTADO 2
		%I0.2	VÁLVULA DE MATERIA PRIMA
		%I0.3	VÁLVULA DE LLENADO DE SACO

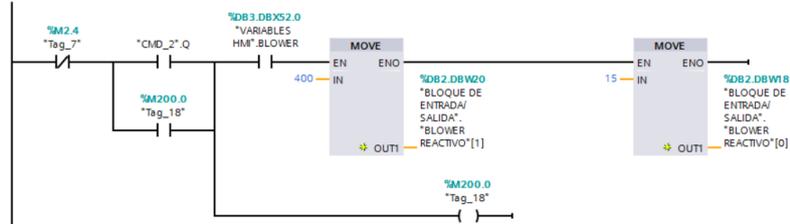
- El FC de servicio contiene bloques de transferencia directa hacia los esclavos y son accionados mediante pulsadores que permiten la activación de los dispositivos conectados en la red, tal como motores, válvulas y mezcladores.
- El bloque FC se nombran de acuerdo al PLC que se vaya enviar los datos, estos son: Sensores, actuadores y válvulas.
- Para el caso de los motores, estos se envían los datos de control directamente a los variadores, por lo que tan solo se controla el envío de la secuencia de activación y el arranque correspondiente.
- En los respectivos bloques FC se programan solo las marcas de que activan los bloques MOVE para transferir los datos hacia el cliente, todo el procesamiento de la simulación se realiza en el bloque el FB del HMI.

➤ Bloque FC Variadores

En este bloque se programa la secuencia de activación de los motores y la correspondientes marcas de activación de las bombas en el HMI, permiten la activación de los bloques move que transfieren los valores de activación y velocidad a los 4 ATV312, inicialmente se envía la secuencia de activación de los cuatro variadores.



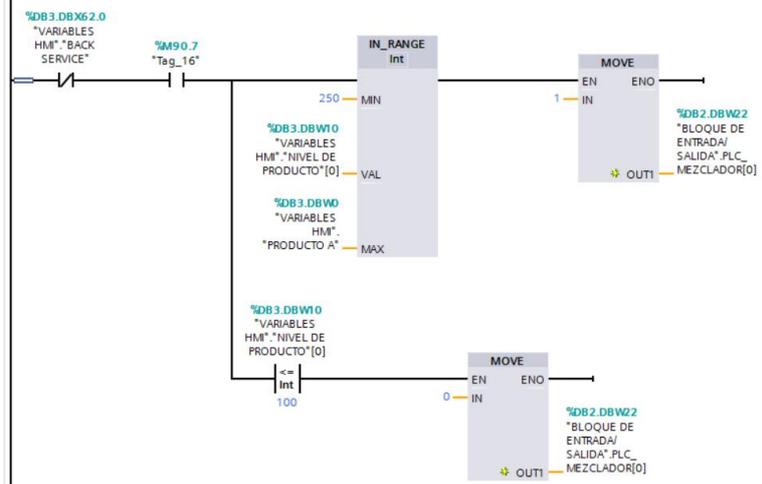
Se programa el arranque del blower reactivo (variador 4).



Se programa el arranque del motor 1 (variador 1).

➤ Bloque PLC_Mezcladores

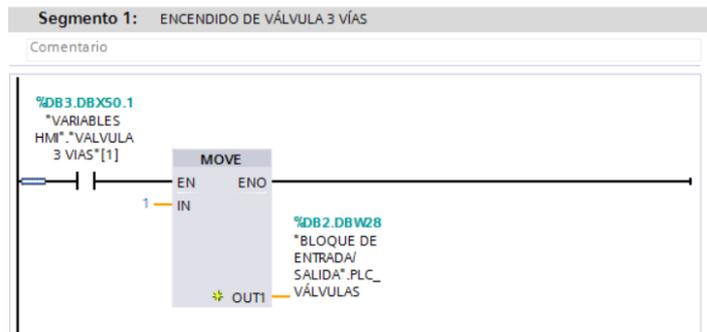
En este bloque se programan los bloques move para transferencia de datos hacia el esclavo para encender los indicadores luminosos por cada agitador, estos bloques de transferencia se activan de acuerdo a la simulación del HMI, es decir, si en la simulación el mezclador se activa, la misma variable permitirá que se transfiera el dato de activación de los indicadores en el esclavo.



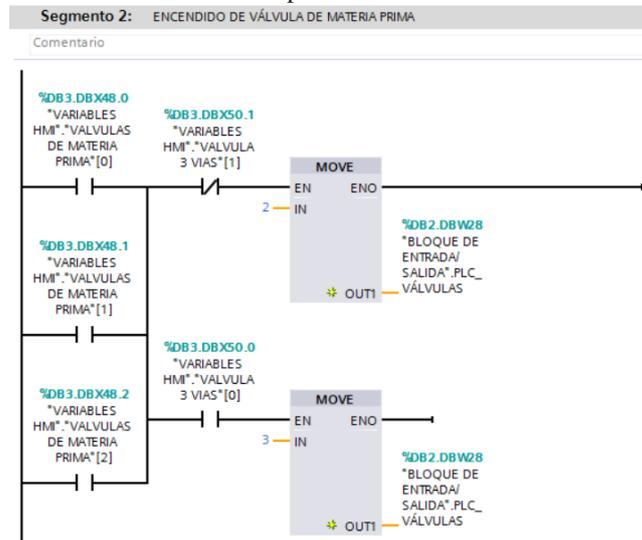
➤ PLC_Válvulas

En este bloque se transfieren los datos de activación hacia el PLC esclavo que maneja los indicadores de las válvulas, estos bloques son activados por la variable HMI correspondiente de la simulación.

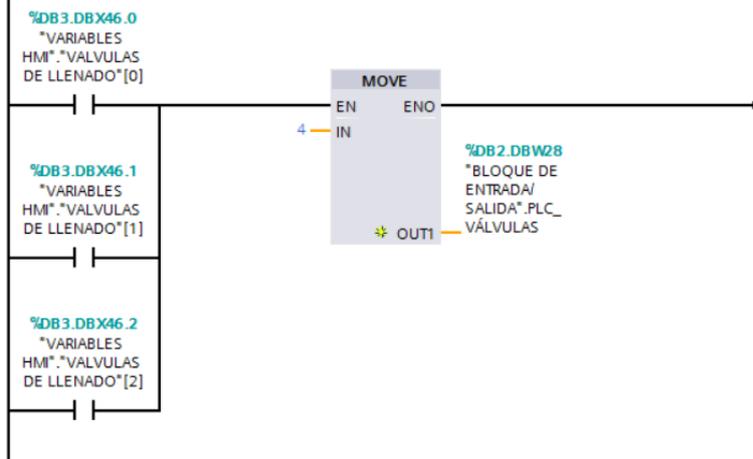
Encendido de válvula 2 vías.



Encendido de válvula de materia prima.



Encendido de válvulas de llenado.



4. Programación de HMI.

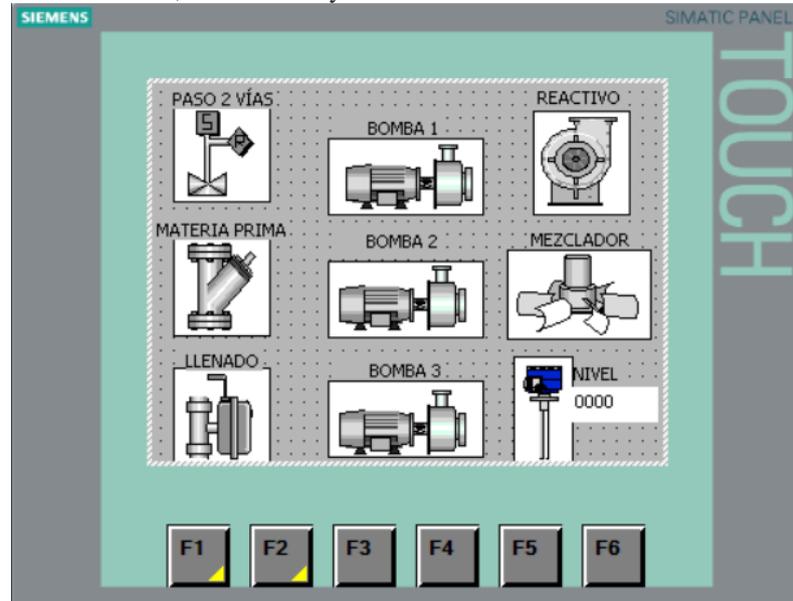
- Se programan las pantallas de simulación y de ingreso de datos.



- Se incluyen dos pantallas principales, una para servicio y otro para operario.
- La pantalla operario recolecta los datos ingresados por el operador y en función a esos datos, se llenara los tanques de productos y los sacos de producto final.



- La pantalla de servicio permitirá a un técnico o ingeniero, dar un mantenimiento preventivo de los equipos conectados, en donde mediante un panel de control desde el maestro, mandara a encender motores, mezcladores y válvulas.



- En esta pantalla se ha programado la tecla F2 para la puesta en cero de las variables HMI y envío de valor cero a las palabras de los esclavos.
- Todas las variables HMI usadas permiten la activación de los bloques de transferencia hacia los esclavos.
- En todas las pantallas HMI, la tecla F1 está programada para retroceder a la pantalla anterior.

ACTIVIDADES POR DESARROLLAR

1. Generar una señal analógica para la activación de los indicadores de nivel.
2. Monitorear los valores de condiciones de funcionamiento de los motores.
3. Implementar un bloque de simulación de fallas para verificar la respuesta del sistema.

RESULTADO(S) OBTENIDO(S):

Se obtiene como datos los valores escritos en las palabras de los esclavos y las condiciones actuales en las cuales estos se encuentran, ya que estos solo están programados para mostrar las condiciones del proceso, simulando los diferentes componente que se han agregado en el HMI, en la siguiente pantalla se muestra los valores enviados a los esclavos motores, el motor 3 se encuentra activado (valor enviado F).

BLOQUE DE ENTRADA/SALIDA					
Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación	
Static					
PLC_SENSORES - NIVEL	Array [0..2] of Word	0.0			
PLC_SENSORES - NI...	Word	0.0	16#0	16#0320	
PLC_SENSORES - NI...	Word	2.0	16#0	16#0320	
PLC_SENSORES - NI...	Word	4.0	16#0	16#0020	
BOMBA 1	Array [0..1] of Word	6.0			
BOMBA 1[0]	Word	0.0	16#0	16#0007	
BOMBA 1[1]	Word	2.0	16#0	16#0000	
BOMBA 2	Array [0..1] of Word	10.0			
BOMBA 2[0]	Word	0.0	16#0	16#0007	
BOMBA 2[1]	Word	2.0	16#0	16#0000	
BOMBA 3	Array [0..1] of Word	14.0			
BOMBA 3[0]	Word	0.0	16#0	16#000F	
BOMBA 3[1]	Word	2.0	16#0	16#0578	
BLOWER REACTIVO	Array [0..1] of Word	18.0			
BLOWER REACTIVO...	Word	0.0	16#0	16#0007	
BLOWER REACTIVO...	Word	2.0	16#0	16#0000	
PLC_MEZCLADOR	Array [0..2] of Word	22.0			
PLC_MEZCLADOR[0]	Word	0.0	16#0	16#0001	
PLC_MEZCLADOR[1]	Word	2.0	16#0	16#0001	
PLC_MEZCLADOR[2]	Word	4.0	16#0	16#0000	
PLC_VÁLVULAS	Word	28.0	16#0	16#0003	

En la siguiente figura se muestra los valores enviados para los mezcladores, sensores y válvulas, esto durante el proceso de llenado de saco, ya que al enviar el valor 4 hacia el esclavo de control de válvulas, se acciona la válvula de llenado de sacos.

BLOQUE DE ENTRADA/SALIDA				
Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
Static				
PLC_SENTORES - NIVEL	Array [0..2] ...	0.0		
PLC_SENTORES - NI...	Word	0.0	16#0	16#0258
PLC_SENTORES - NI...	Word	2.0	16#0	16#0258
PLC_SENTORES - NI...	Word	4.0	16#0	16#033F
BOMBA 1	Array [0..1] of Word	6.0		
BOMBA 1[0]	Word	0.0	16#0	16#0000
BOMBA 1[1]	Word	2.0	16#0	16#0000
BOMBA 2	Array [0..1] of Word	10.0		
BOMBA 2[0]	Word	0.0	16#0	16#0000
BOMBA 2[1]	Word	2.0	16#0	16#0000
BOMBA 3	Array [0..1] of Word	14.0		
BOMBA 3[0]	Word	0.0	16#0	16#0000
BOMBA 3[1]	Word	2.0	16#0	16#0000
BLOWER REACTIVO	Array [0..1] of Word	18.0		
BLOWER REACTIVO...	Word	0.0	16#0	16#0000
BLOWER REACTIVO...	Word	2.0	16#0	16#0000
PLC_MEZCLADOR	Array [0..2] of Word	22.0		
PLC_MEZCLADOR[0]	Word	0.0	16#0	16#0001
PLC_MEZCLADOR[1]	Word	2.0	16#0	16#0001
PLC_MEZCLADOR[2]	Word	4.0	16#0	16#0001
PLC_VÁLVULAS	Word	28.0	16#0	16#0004

CONCLUSIONES:

En esta práctica se realiza la comunicación con 8 dispositivos esclavos RTU, los cuales sirven para mostrar las condiciones de la simulación, es decir si está activado algún motor, válvula o indicador. En esta documento no se ha incluido la programación de la animación HMI, ya que el objetivo de este proyecto no es enseñar la programación de los distintos elementos más bien de las aplicaciones que tiene el protocolo con el que se está trabajando (Modbus TCP). Los valores obtenidos se muestran en formato Hexadecimal.

RECOMENDACIONES:

- Se recomienda realizar un diagrama de direccionamiento, para un mejor entendimiento del mapeo de palabras entre maestro-esclavo.

CLIENTE - PLC MOTORES		
S7-1200	ATV312	ESTADO
%M2.1	W8501	BOMBA MATERIA PRIMA A (Motor 1)
%M2.2	W8501	BOMBA MATERIA PRIMA B (Motor 2)
%M2.3	W8501	BOMBA MATERIA PRIMA C (Motor 3)
DB3 "VARIABLES HMI".BLOWER	W8501	BLOWER REACTIVO (Motor 4)

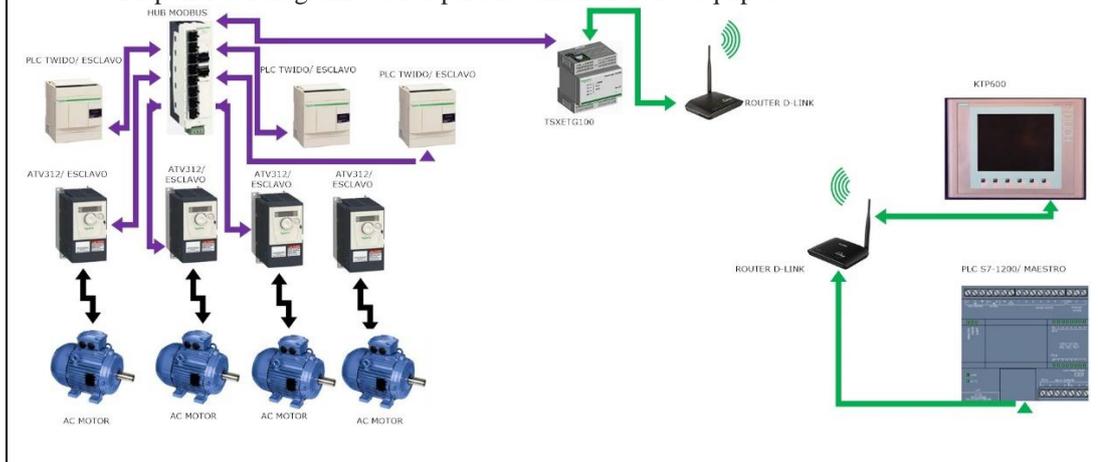
CLIENTE - PLC SENSORES		
S7-1200	TWIDO	ESTADO
DB3 "NIVEL PRODUCTO 0"	%MW2	SENSORES DE NIVEL
DB3 "NIVEL PRODUCTO 1"	%MW3	SENSORES DE NIVEL
DB3 "NIVEL PRODUCTO 2"	%MW4	SENSORES DE NIVEL

CLIENTE - PLC AGITADORES		
S7-1200	TWIDO	ESTADO
%M3.1	%MW0	INDICADOR DE AGITADOR A
%M3.2	%MW1	INDICADOR DE AGITADOR B
%M3.5	%MW2	INDICADOR DE AGITADOR C

CLIENTE - PLC VÁLVULAS		
S7-1200	TWIDO	ESTADO

%M2.0	%MW0:X0	VÁLVULA DE 2 VÍAS
%M3.0 – M3.1 – M3.2	%MW0:X1	VÁLVULA DE PASO
DB3 “VÁLVULAS DE LLENADO 0, 1 Y 2”	%MW0:X2	VÁLVULAS DE LLENADO

- Verificar el estado de los cables de conexión de los equipos Modbus.
- Se recomienda de disponer el uso del manual del ATV312 para la correcta configuración en red Modbus.
- Disponer del diagrama de red para la conexión de los equipos.



Docente / Técnico Docente: _____

Firma: _____

5. Análisis Y Resultados

5.1 Resultados Obtenidos

Para el desarrollo de este proyecto de titulación se obtuvo como resultado la implementación de un módulo de conversión de protocolos de comunicación industrial Modbus TCP, el cual permite la comunicación de un PLC maestro con comunicación Ethernet TCP con esclavos en Modbus RTU, las etapas de elaboración del proyecto se resumen en:

- Conexión de equipos del módulo convertidor.
- Conexión de la estación maestro con el Router inalámbrico.
- Pruebas de comunicación entre las estaciones maestro – esclavo.
- Elaboración de prácticas estudiantiles.



Figura 28. Montaje de equipos del módulo convertidor Modbus TCP (Autores, 2017)

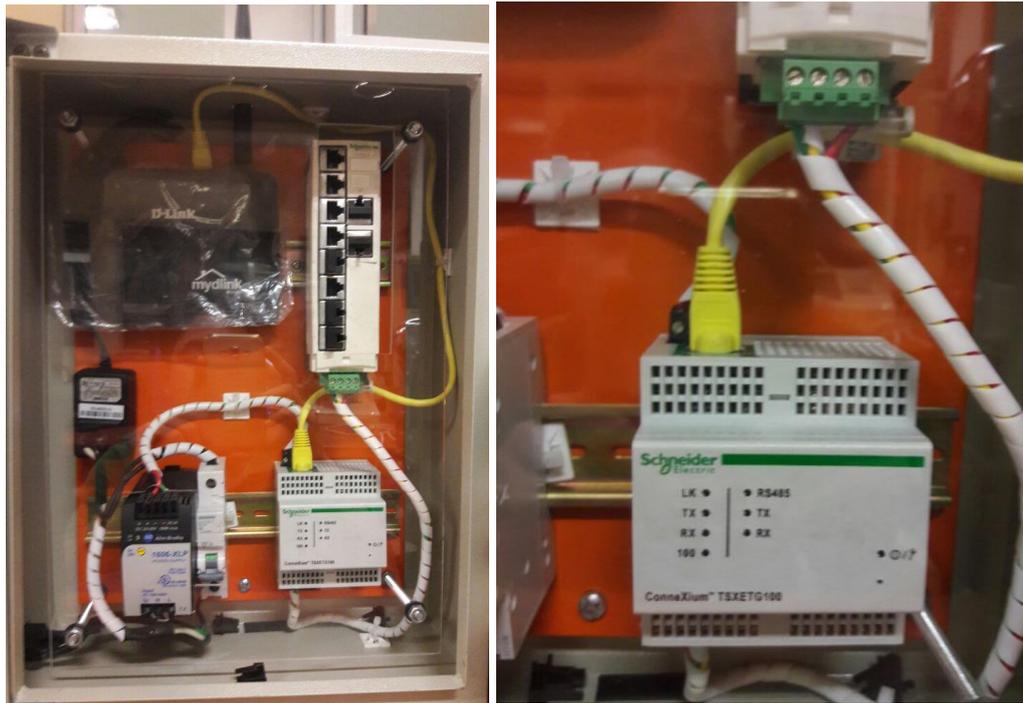


Figura 29. Conexión de los equipos de comunicación y ajuste de cables (Autores, 2017)

Durante las pruebas realizadas y las prácticas desarrolladas se visualizan los valores enviados y recibido por parte del módulo, las conexiones físicas de los dispositivos esclavos se realizan mediante cables UTP categoría 5E, se asegura que las conexiones no sean intermitentes, no presenten sulfataciones ni cortocircuitos en los conectores RJ45, esto para garantizar una velocidad de respuesta rápida por parte de los esclavos.



Figura 30. Conexión de los dispositivos esclavos en el HUB Modbus (Autores, 2017)

Los resultados de los valores obtenidos en las prácticas están adjuntos en el capítulo cuatro según la práctica correspondiente, la conexión de los equipos se realiza fijando bien los plus RJ45 en el terminal HUB Modbus del módulo y en su otro extremo asegurando el buen contacto con el dispositivo correspondiente de acuerdo al que se utilice en la práctica.

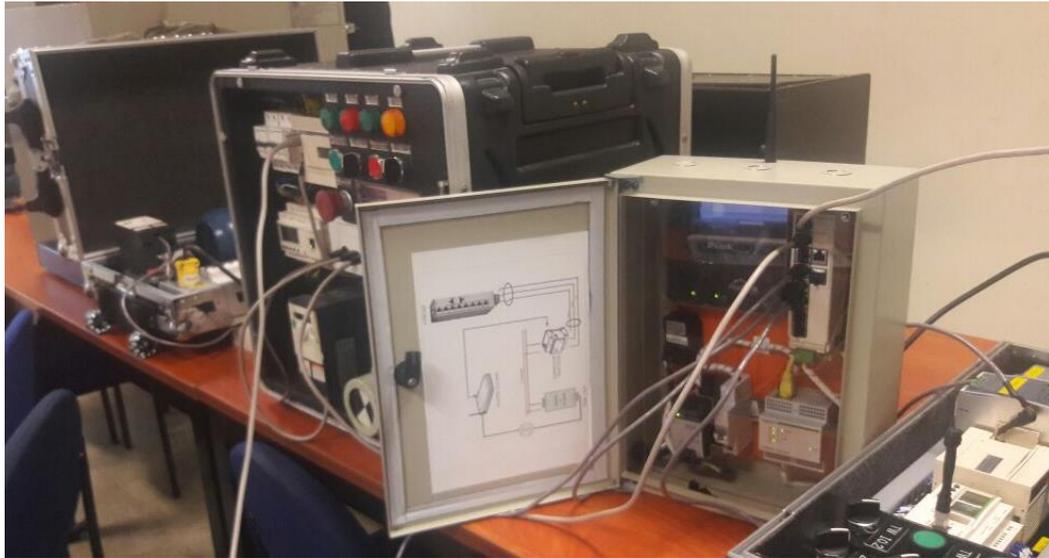


Figura 31. Conexión de 3 maletas Modbus – Práctica 9 (Autores, 2017)



Figura 32. Conexión de una maleta didáctica Modbus – Práctica 1 (Autores, 2017)

5.2 Análisis de Resultados

Para obtener los resultados correctos en las prácticas y pruebas realizadas se debe analizar las condiciones idóneas del módulo convertidor y accesorios utilizados.

- Conexión de equipos del módulo Modbus TCP:

Los cables deben estar conectados correctamente y estos no deben presentar falsos contactos, sulfataciones ni cortocircuitos, una mala conexión entre los equipos ocasiona que la comunicación sea lenta en el tiempo de respuesta.

- Conexión de la estación maestro con el Router inalámbrico:

La conexión del módulo maestro se encuentra conectada a un Router D-Link, el cual está enlazado al módulo convertidor Modbus TCP, la distancia de separación entre los Routers no debe superar los 7 metros y evitar obstáculos intermedios que harían más lento el envío/recepción de los datos inalámbricos.

- Pruebas de comunicación entre las estaciones maestro – esclavo:

Para la obtención de una correcta comunicación entre las estaciones, se debe realizar la correcta configuración y seteo de los equipos.

En el caso del bloque de control Mb_Client del maestro, este debe tener la frecuencia de requisición correcta para una sincronización de envío/recepción de datos, también establecer el número de dirección del esclavo RTU con el cual se gestionaran los datos de dicho bloque.

Para los esclavos, en el caso de los variadores se debe seguir las instrucciones descritas en el manual y configurar los parámetros de paridad, baudiaje, dirección de red, tiempo

de respuesta, etc, esto en el menú de comunicación. Para los PLC's Twido se debe configurar mediante el software los parámetros de red y el tiempo de respuesta.

- Elaboración de prácticas estudiantiles:

Los resultados obtenidos se deben a la correcta manipulación de los equipos y sus accesorios, esto para garantizar que los datos son enviados y recibidos.

Las prácticas desarrolladas se realizan en base a aplicaciones prácticas y simulación de proceso que permiten un mejor entendimiento del protocolo implementado.

CONCLUSIONES

El presente documento de titulación no pretende enseñar al lector la programación de los módulos utilizados, ya que es y ha sido responsabilidad del lector tener conocimientos previos de programación de PLC y redes industriales como son las descritas y utilizadas en este proyecto (Modbus y Ethernet Industrial), más bien tiene como objetivo demostrar la utilización de la variante Modbus sobre TCP en procesos industriales para una mejor supervisión de procesos, monitoreo de datos y velocidad de información en tiempo real.

El diseño de la red Modbus TCP se realizó con éxito con ayuda del módulo convertidor, el cual permite la comunicación entre equipos SIEMES y SCHNEIDER que manejan tipos de comunicaciones diferentes.

Las prácticas propuestas fueron probadas e implementadas con los equipos disponibles en las maletas didácticas, los procesos industriales que se simulan representan una ayuda para las aplicaciones didácticas por parte del docente, en este documento se pretende incentivar al lector, profesor o alumno a que realice nuevas prácticas, simulaciones o modificaciones con el modulo convertidor para hacer operar al Gateway ETG100 en modo esclavo.

Se puede realizar el cambio de configuración del módulo ETG100 esto para invertir el funcionamiento de los módulos y que el equipo funcione como maestro serie y esclavo TCP. Esto puede ser considerado para futuras prácticas o nuevo desarrollo de un módulo como comunicación Modbus TCP inversa.

RECOMENDACIONES

El módulo de conversión MODBUS – TCP permite el intercambio de datos entre el PLC maestro y las 4 estaciones de trabajo esclavos SCHNEIDER y se logra la comunicación inalámbrica entre las estaciones mencionadas con un alcance de hasta 7 metros sin pérdidas de datos.

Se recomienda la correcta manipulación del módulo convertidor desarrollado y la comprensión de los manuales técnicos de los equipos utilizados ya que con este módulo se puede realizar una conversión de datos Modbus RTU a TCP o de TCP a Modbus RTU.

Se debe tomar en consideración la distancia entre el modulo convertidor y la estación del cliente ya que si supera el alcance máximo de la señal del Router se ocasionaría perdida de la señal o respuestas más lenta por parte de los esclavos.

Disponer del manual del convertidor TSXETG100 para setear los parámetros del equipo y configurar el equipo en modo esclavo.

CRONOGRAMA

ACTIVIDAD	2015					2016												2017	
	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	
1																			
2																			
3																			
4																			
5																			
6																			
7																			
8	INVESTIGACIÓN DE PLATAFORMA TWIDOSUITE Y PLC TWIDO COMPACTO																		
9		ESTUDIO DE MALETAS DIDACTICAS DE COMUNICACIÓN MODBUS																	
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24	INVESTIGACIÓN DE VARIADOR ATV 312	PEDIDO DE IMPORTACIÓN DE EQUIPOS DE MODULO CONVERTIDOR																	
25																			
26																			
27																			
28																			
29																			
30																			

PRESUPUESTO

DETALLES DE GASTO – MODULO CONVERTIDOR				
ITEM	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	TOTAL
1	CONVERTIDOR TSXETG100	1	410.53	410.53
2	REPARTIDOR HUB LU9C-GC3	1	96.60	96.60
3	FUENTE ALLEN BRADLEY 24VDC	1	75.50	75.50
4	ROUTER DLINK DIR900L	1	34.25	34.25
5	ROUTER DLIN DIR610	1	23.40	23.40
6	RIEL DE MONTAJE DE EQUIPOS	1	1.20	1.20
7	PLANCHA DE ACRILICO	1	30.00	30.00
8	CAJA METÁLICA 40X30X15CM	1	35.25	32.25
9	CABLE AWG #14 Y 18 (1 METRO)	2	1.80	3.60
10	CABLE CONCENTRICO 3X12	1	2.50	2.50
11	PROTECTOR ESPIRAL PARA CABLE, SELLADOR TERMICO, ESPADINES	1	6.70	6.70
12	MATERIALES ELÉCTRICOS PARA DISEÑO Y MONTAJE DE EQUIPOS EN CAJA METÁLICA	1	21.56	21.56
13	DISEÑO Y CUBIERTA DE VINIL EN PLANCHA DE ACRILICO	1	16.89	16.89
14	CABLE BELDEN 3 HILOS COLOR MORADO (METRO)	12	0.75	9.00
15	CABLE BELDEN DE 6 HILOS COLOR VERDE (METRO)	4	0.75	3.00
16	CONECTOR RJ45 CAT. 6A	15	1.25	18.75
		SUBTOTAL		785.73
		IVA 14%		110.00
		TOTAL		895.73

BIBLIOGRAFÍA

- Schneider Electric, Guía de Soluciones de automatización – capítulo 9
- Universidad de Oviedo (2006). Comunicaciones Industriales
- Stallings William (2000). Comunicaciones y redes de computadoras edición 6. Pearson Education.
- Alfredo Rosado (2009). Sistemas Industriales Distribuidos. España: Universidad de Valencia, recuperado de: www.uv.es/rosado/course/sid/sid.html.
- Jose Hurtado (2015). Introducción a las redes de Comunicación Industrial, recuperado de:
<https://josemariahurtadotorres.wordpress.com/2015/10/25/introduccion-a-las-redes-de-comuniacion-industrial>.
- D. Olmos y Fco. Ramos (2008). Redes de computadoras “Ethernet Industrial”, recuperado de,
<https://profesores.elo.utfsm.cl/~agv/elo322/1s08/project/BarrosOlmos.doc>
- Andrés Ruiz Oyala (2002). Tesis: Implementación de una red Modbus/TCP. Santiago de Cali: Universidad del Valle.
- Schneider Electric (2011). TwidoSuite V2.3 Programming Guide, recuperado de:
http://download.schneiderelectric.com/files?p_File_Id=27603565&p_File_Name=35011386_K01_000_05.pdf
- Antonio Barragán Piña (2013). Representación de datos, recuperado de:
<http://uhu.es/antonio.barragan/content/representacion-datos>
- Paul Gálvez (2011). Estructura de Tablas, recuperado de:
<http://es.slideshare.net/PaulGlvez/modbus-39533348>

- Antonio Creus Sole (2005). Instrumentación Industrial. México: Marcombo.
- Barragán, J. (2013). Tutorial Virtual, recuperado de:
<http://www.tolaemon.com/docs/modbus.htm>
- Distefano, I. M. (2008). Electrónica General y Aplicada, Comunicaciones Industriales, recuperado de:
http://www.infoplcn.net/files/documentacion/infoPLC_net_ComunicacionesEntornos_Industriales.pdf
- Kryon_Ingenieria (2014). Protocolo Modbus, recuperado 2015.
- SIEMENS (2015). Siemens Automation Cooperates with Education, recuperado de:
http://www.swe.siemens.com/spain/web/es/industry/automatization/sce_educacion/e_education/pages/default.aspx
- SIEMENS_AG (2010). Modulos_Modbus RTU RS485, recuperado de:
https://cache.automation.siemens.com/dnl_iss/TE/TE2MjQwNwAA_44880906_HB/USMII_MODBUS_AOM_OI_ES_es-ES.pdf
- SIEMENS_AG (2008). HMI Device KTP400 Basic, KTP600 Basic, KTP1000 Basic, TP1500 Basic. Operating Instructions, recuperado de:
https://www.automatyka.siemens.pl/docs/docs_ia/HMI_KTP400_KTP600_KTP1000_TP1500.pdf
- Schneider (2012). Connexium Ethernet Gateway TSXETG100, User Guide
- Siemens (2009). Manual del Sistema S7 Controlador Programable S7-1200.

ANEXOS

ANEXO 1

Configuración y habilitación de Macros de Comunicación

Ir pestaña general de “Programa” → “Configurar” → “Configurar datos”, aparecerá la ventana de configuración de objetos, se muestra en la figura 27.

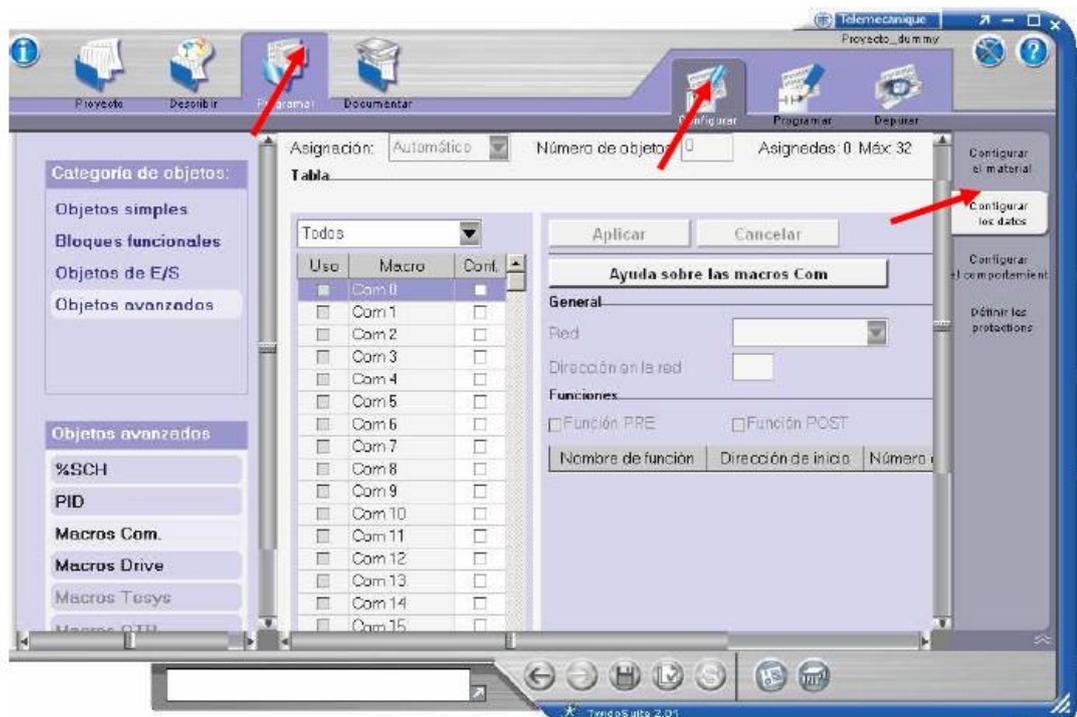


Figura 33. Paso 1 para seleccionar Macros (SCHNEIDER ELECTRIC, 2008)

Se selecciona la opción de “Objetos avanzados” y dentro “Macros Com.”, se muestra en la figura 28.

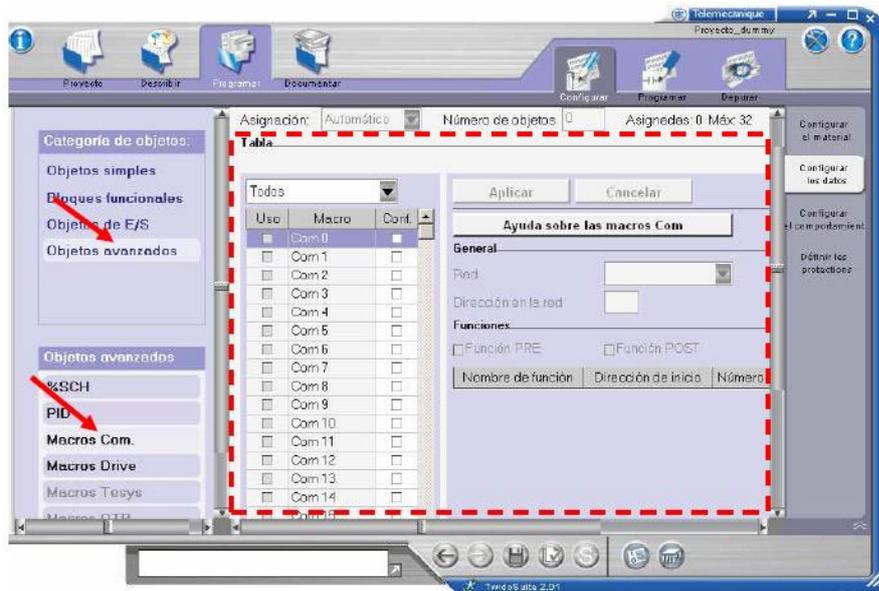


Figura 34. Paso 2 para seleccionar Macros (SCHNEIDER ELECTRIC, 2008)

Para configurar la macro, seleccionar entre las disponibles el número de macro deseada. En la casilla de “Red”, se selecciona el puerto donde se ha configurado la red Modbus y en la casilla de “Dirección de red” el número del esclavo que se ha configurado anteriormente en la pestaña “Describir”.

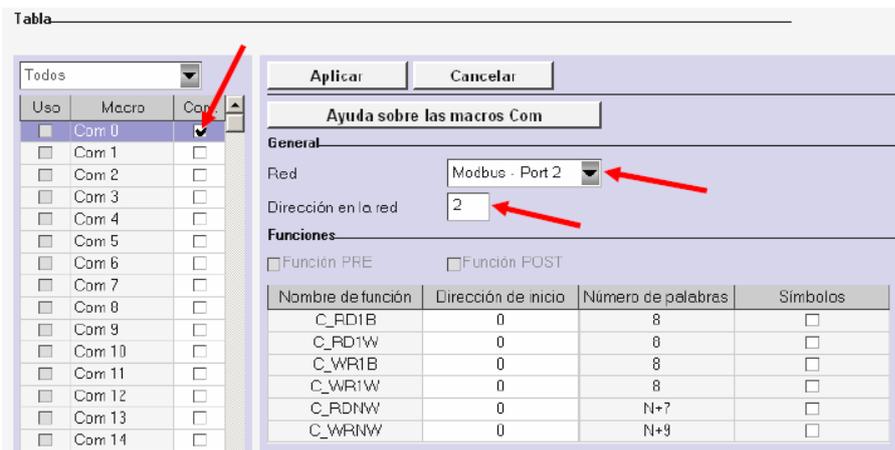


Figura 35. Paso 3 Selección de Puertos y Número de esclavo (SCHNEIDER ELECTRIC, 2008)

Ahora se tiene que seleccionar la función de comunicación que se quiere realizar sobre el dispositivo (**Lectura o Escritura**) y el volumen de datos (**byte, Word o N Words**).

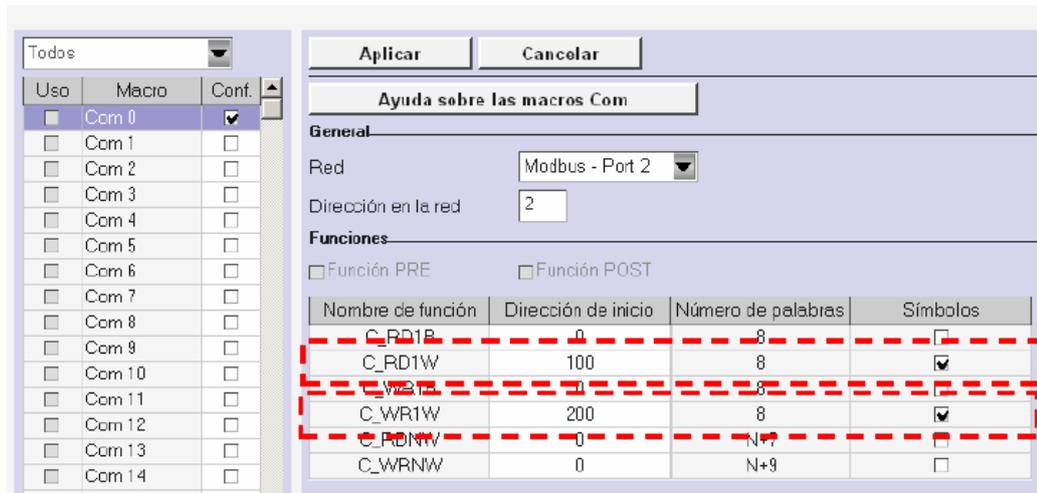


Figura 36. Paso 4 Selección de función Macro (SCHNEIDER ELECTRIC, 2008)

ANEXO 2

Configuración de Gateway ETG100 y Routers

La pasarela ETG100 permite una fácil configuración de los parámetros como el tipo conexión, puerto de comunicación y monitoreo en tiempo de real de equipo conectado, etc. Para lo cual solo debe acceder por medio de un explorador web a la puerta de enlace predeterminada 169.254.0.10 y acceder a la opción de **Configuración** en la barra de menús, en donde se seteara los diferentes parámetros iniciales del equipo como son:

Tabla 23. Configuraciones iniciales de ETG100

Opción	Descripción	Valor
Formato de trama	Permite seleccionar el formato en el que se envían los datos a través de una conexión Ethernet.	Ethernet II, 802.3 SNAP Predeterminado: Ethernet II
Tipo de medio	Sirve para definir la conexión física a Ethernet o el tipo de medio.	<ul style="list-style-type: none">• 10T/100Tx Auto• 10BaseT-HD• 10BaseT-FD• 100BaseTX-HD• 100BaseTX-FD Predeterminado: 10T/100Tx Auto
Dirección IP	Sirve para introducir la dirección IP estática de la ETG.	Valor predeterminado: 169.254.0.10
Máscara de subred	Permite introducir la dirección de la máscara de subred de la red.	Valor predeterminado: 255.255.0.0
Pasarela predeterminada	Sirve para introducir la dirección IP de la pasarela (router) que se usa en las comunicaciones con la red de área extendida (WAN).	Valor predeterminado: 0.0.0.0

Se describe el tipo de trama, tipo de medio, dirección IP, etc. (Schneider, 2012)

Luego de la configuración de parámetros iniciales, se debe realizar la configuración de la conexión del puerto serial como el baudiaje, paridad, modo, etc.



Figura 37. Página de configuración Gateway (Autores, 2016)

Una vez tenido configurado todos parámetros el equipo automáticamente se reinicia con la nueva configuración cargada y para acceder a las configuraciones nuevamente se debe ingresar por medio de la dirección IP asignada (192.168.0.15).

Para la comunicación inalámbrica se ha utilizado un Router Cloud D-Link modelo DL900, el cual ha sido debidamente configurado en la red que MODBUS/TCP. La conexión física del equipo se realizó conectando el puerto de servicio del Router al puerto de conexión Ethernet del Gateway ETG100. La configuración del Router se realizó por medio del servidor ingresando por una página web a la dirección predefinida (192.168.0.1), en donde se accedió a la configuración de red LAN para asignar la nueva IP del Router, la configuración por default se encuentra en el manual del equipo.

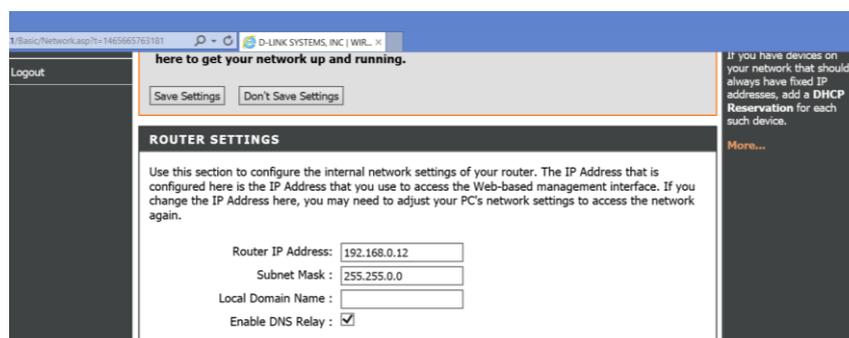


Figura 38. Página de configuración LAN Router (Autores, 2016)

Luego se accede a los parámetros de WLAN en donde se configura el nombre la red wireless.

Time and Date	WI-FI PROTECTED SETUP (ALSO CALLED WCN 2.0 IN WINDOWS VISTA)	Enabling Hidden Mode is another way to secure your network. With this option enabled, no wireless clients will be able to see your wireless network when they scan to see what's available. For your wireless devices to connect to your router, you will need to manually enter the Wireless Network Name on each device. If you have enabled Wireless Security, make sure you write down the Key or Passphrase that you have configured. You will need to enter this information on any wireless device that you connect to your wireless network. More...
Parental Control Rules	Enable: <input checked="" type="checkbox"/> Current PIN: 96119433 <input type="button" value="Generate New PIN"/> <input type="button" value="Reset PIN to Default"/> Wi-Fi Protected Status: Enable /Not Configured <input type="button" value="Reset to Unconfigured"/> Disable WPS-PIN Method: <input type="checkbox"/> <input type="button" value="Add Wireless Device with WPS"/>	
Logout	WIRELESS NETWORK SETTINGS	
	Wireless Mode: <input type="text" value="Wireless Router"/> <input type="button" value="v"/> Enable Wireless: <input checked="" type="checkbox"/> Wireless Network Name (SSID): <input type="text" value="Modbus/TCP"/> (Also called the SSID) Enable Auto Channel Selection: <input checked="" type="checkbox"/> Wireless Channel: <input type="text" value="6"/> <input type="button" value="v"/> Transmission Rate: <input type="text" value="Best (automatic)"/> (Mbit/s) <input type="button" value="v"/> WMM Enable: <input checked="" type="checkbox"/> (Wireless QoS) Enable Hidden Wireless: <input type="checkbox"/> (Also called the SSID Broadcast)	
WIRELESS SECURITY MODE		
Security Mode: <input type="text" value="Disable Wireless Security (not recommended)"/> <input type="button" value="v"/>		
<input type="button" value="Save Settings"/> <input type="button" value="Don't Save Settings"/>		
WIRELESS		

Figura 39. Página de configuración WLAN Router (Autores, 2016)

ANEXO 3

Configuración de bloque de instrucción Modbus TCP de PLC Siemens

Los bloques de instrucción Mb_Client y Mb_Server son únicamente para transferir datos bajo el protocolo Modbus TCP ya que su direccionamiento es por medio de dirección IP, estos valores de transferencia son por default del fabricante pero estos bloques pueden ser usados también cuando se cuenta con un convertidor Modbus TCP y se tiene un esclavo o servidor RTU que requiere una dirección entre 1 a 247.

En esta sección se explica el parámetro que debe modificarse para direccionar al esclavo RTU mediante un convertidor, dicha instrucción se encuentra ingresando al bloque de registros de parámetro del OB que se crea al insertar el bloque de instrucción.

La función MB_UNIT_ID, en donde se muestra el valor hexadecimal del esclavo RTU, es donde se introduce el valor número del esclavo entre 1 a 247.

	Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
25	▶ TCON_PARAM	TCON_Param		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	PF_FREQUENCY	Real	0.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
27	BLOCKED_PROC_TIME...	Real	3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
28	RCV_TIMEOUT	Real	2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
29	CURRENT_TIME_VALUE	DWord	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30	SAVED_TIME_VALUE	DWord	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
31	XMT_TIME	DWord	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
32	SAVED_CONNECT_ID	Word	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33	SAVED_IP_PORT	Word	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
34	SAVED_MODE	Byte	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
35	SAVED_IP1	Byte	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
36	SAVED_IP2	Byte	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
37	SAVED_IP3	Byte	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
38	SAVED_IP4	Byte	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
39	SAVED_DATA_ADDR	DWord	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40	SAVED_DATA_LEN	Word	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
41	MB_STATE	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
42	COMM_SENT_COUNT	Word	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
43	BYTE_COUNT	Word	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
44	BYTE_COUNTB	Byte	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
45	SAVED_START_ADDR	Word	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
46	MB_TRANSACTION_ID	Word	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
47	MB_UNIT_ID	Word	16#0001	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
48	RETRIES	Word	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
49	INIT_OK	Bool	false	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
50	ACTIVE	Bool	false	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
51	CONNECTED	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
52	SAVED_MA_REQ	Bool	false	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 40. Direccionamiento RTU del bloque MB_CLIENT de SIEMENS (Autores, 2016)