UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA:

INGENIERÍA ELECTRÓNICA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO ELECTRÓNICO

PROYECTO TÉCNICO

DISEÑO E IMPLEMENTACIÓN DE UNA RED MODBUS/RTU ENTRE DOS AUTÓMATAS PROGRAMABLES S7-1200 BASADO EN EL ESTÁNDAR RS485

AUTORES:

RENÉ EDUARDO ZAMBRANO MONSERRATE
CHRISTIAN PATRICIO CABALLERO DE LA TORRE

TUTOR:

ING. BYRON XAVIER LIMA CEDILLO Msc.

GUAYAQUIL – ECUADOR

2018

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

La responsabilidad del contenido de este proyecto de titulación, nos corresponde exclusivamente a nosotros:

- René Eduardo Zambrano Monserrate portador de C.I: 0921095071.
- Christian Patricio Caballero de la Torre portador de C.I: 0923399760.

Alumnos de la UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL.

Guayaquil, julio del 2018

_____

(f) René Eduardo Zambrano Monserrate

C.I: 0921095071

_____

(f) Christian Patricio Caballero de la Torre

C.I: 0923399760

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Nosotros, René Eduardo Zambrano Monserrate con documento de identificación 0921095071 y Christian Patricio Caballero de la Torre con documento de identificación 0923399760, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado: "**Diseño e implementación de una red MODBUS/RTU entre dos autómatas programables S7-1200 basado en el estándar RS485**", 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 hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, julio del 2018

_____

(f) René Eduardo Zambrano Monserrate

C.I: 0921095071

_____

(f) Christian Patricio Caballero de la Torre

C.I: 0923399760

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

En calidad de Tutor del trabajo de titulación proyecto técnico "**Diseño e implementación de una red MODBUS/RTU entre dos autómatas programables S7-1200 basado en el estándar RS485**", presentado por los señores René Eduardo Zambrano Monserrate y Christian Patricio Caballero de la Torre para optar por el título de Ingeniero Electrónico, certifico que el mencionado proyecto fue realizado bajo mi dirección.

_____

Tutor del trabajo de titulación

C.C: Ing. Byron Xavier Lima Cedillo Msc.

DEDICATORIA

Este proyecto no hubiese sido posible sin la ayuda y bendición de Dios, por brindarme fortaleza y sabiduría a lo largo de mi carrera profesional.

Quiero dedicar este logro a Dios y a mi amada familia que estuvieron apoyándome siempre en mis decisiones y desempeño académico. A mis padres René Zambrano Andrade y Betty Monserrate Zamora que han velado por mi integridad y bienestar toda una vida.

A mis hermanos Luis Andrés, María Fernanda y José Eduardo quienes me alentaron en los momentos difíciles y creyeron en mí en todo momento.

El sacrificio de horas interminables de trabajo y dedicación ha dado su recompensa, con mucha honestidad y humildad, como un salesiano de corazón.

René Eduardo Zambrano Monserrate

DEDICATORIA

Con todos mis sentimientos dedico este trabajo de titulación a mis padres Samuel y Marieta, quienes me apoyaron en toda situación y me ayudaron con sus consejos a continuar y concluir este objetivo personal.

Christian Patricio Caballero De la Torre

AGRADECIMIENTOS

Agradezco de primera mano a Dios por darme la bendición de crecer profesional y humanamente, sin su directriz no hubiese sido posible este acontecimiento.

A mi familia por haberme impulsado a estudiar y fomentar los valores morales y humanos hacia el prójimo, por haberme educado con rectitud, disciplina y resaltar la honestidad, la verdad y la justicia.

Agradezco a mi tutor el Ing. Byron Lima por brindarme las herramientas e ideas que han hecho posible este proyecto. Cabe recalcar que gracias a su constancia, consejos y buenos haberes hemos concretado este valioso reto.

René Eduardo Zambrano Monserrate

Agradezco a Dios por haberme permitido comenzar y concluir este trabajo con humildad y sencillez.

A los docentes, compañeros, administrativos y amigos por su enseñanza, colaboración y ayuda. Al Ing. Byron Lima tutor de este trabajo, quien con su dirección técnica nos dio el aporte como guía desde la concepción hasta el término de este proyecto.

Christian Patricio Caballero De la Torre

RESUMEN

AÑO	ALUMNOS	TUTOR DEL PROYECTO	TEMA DE PROYECTO
2017/2018	Zambrano Monserrate René Eduardo. Caballero de la Torre Christian Patricio.	Ing. Byron Xavier Lima Cedillo Msc.	Diseño e implementación de una red MODBUS/RTU entre dos autómatas programables S7-1200 basado en el estándar RS485

El proyecto técnico planteado tiene como objetivo principal enseñar eficientes protocolos y técnicas de comunicación entre diferentes sistemas de control y automatización, en nuestro caso damos a conocer el protocolo MODBUS/RTU, el cual detallaremos su uso, aplicación, diseño, configuración, desempeño y ventajas sobre otros métodos de comunicación industrial.

Esto se logrará a través de dos módulos didácticos equipados con controladores lógicos programables (PLC) SIEMENS SIMATIC S7-1200 CPU 1214C DC/AC/Rly, con sus respectivas interfaces de entrada y salida de datos, incluido el puerto RS485 el cual servirá de medio de comunicación entre dos o más módulos de prácticas. También los módulos cuentan con una fuente de poder tipo SIEMENS PM 1207, un Switch SIEMENS CSM 1277 SIMATIC NET, un Touch Panel SIEMENS KTP600 Basic Color PN, una salida analógica Signal Board SB 1232 AQ, contarán también con dos puertos Ethernet de tipo RJ45, un puerto DB9 y un DB25.

Los módulos servirán de gran ayuda para el personal docente y estudiantes ya que contarán con dos equipos repotenciados para el desarrollo de las diferentes prácticas propuestas, he incentivarán nuevo material académico para los estudiantes.

Se utilizó el software TIA PORTAL V12.0 para la programación de las prácticas propuestas, el mismo que se utiliza en la carrera de ingeniería electrónica para la programación y configuración de los diferentes equipos utilizados. Este software procesa y transmite datos a las diferentes salidas, como focos, motores, variadores de frecuencia, HMI, plantas, etc.

Palabras claves: MODBUS/RTU**,** RS485, PLC, HMI.

ABSTRACT

YEAR	STUDENTS	PROJECT TUTOR	TOPIC
2017/2018	Zambrano Monserrate René Eduardo. Caballero de la Torre Christian Patricio.	Eng. Byron Xavier Lima Cedillo Msc.	Design and implementation of a network Modbus/RTU between two programmable automats S7-1200 based on the standard RS485

The technical raised project has as principal aim teach efficient protocols and technologies of communication between different systems of control and automation, in our case we announce the protocol MODBUS/RTU, which we will detail his use, application, design, configuration, performance and advantages on other methods of industrial communication.

This we are going to achieve across two didactic modules equipped with logical programmable controllers (PLC) SIEMENS SIMATIC S7-1200 CPU 1214C DC/AC/Rly, with his respective interfaces of entry and exit of information, included the port RS485 which will use as way of communication between two or more modules of practices. Also the modules possess a source of is able type SIEMENS PM 1207, one Switch SIEMENS CSM 1277 SIMATIC NET, a Touch Panel SIEMENS KTP600 Basic Color PN, an analogical exit Signal Board SB 1232 AQ, they will rely on also two Ethernet ports of type RJ45, a port DB9 and one DB25.

The modules will use as great help for the educational personnel and students since they will possess two equipment's re-promoted for the development of the different proposed practices, I have they will stimulate new academic material for the students. The software was in use TIA PORTAL V12.0 for the programming of the

proposed practices, the same one that is in use in the career of electronic engineering for the programming and configuration of the different used equipment's. This software processes and transmits information to the different exits, as areas, engines, variators of frequency, HMI, plants, etc.

Key Words: Modbus/RTU, RS485, PLC, HMI.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN i

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR ii

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN iii

DEDICATORIA iv

AGRADECIMIENTO v

RESUMEN vii

ABSTRACT ix

ÍNDICE GENERAL xi

ÍNDICE DE FIGURAS xvi

ÍNDICE DE TABLAS xxxi

INTRODUCCIÓN 1

1. EL PROBLEMA 2

1.1. ANTECEDENTES 2

1.2. IMPORTANCIA Y ALCANCE 2

1.3. DELIMITACIÓN 3

1.3.1. Espacial 3

1.3.2. Temporal 3

1.3.3. Sectorial 4

1.3.4. Académica 4

1.4. JUSTIFICACIÓN DEL PROYECTO 4

1.4.1. Objetivo general 5

1.4.2. Objetivos específicos 5

2. FUNDAMENTACIÓN TEÓRICA 6

2.1. Evolución de Procesos Industriales 6
2.2. Sistemas industriales de Control 7
2.2.1. Control centralizado 7
2.2.2. Control distribuido 8
2.2.3. Control híbrido 8
2.3. La pirámide CIM 9
2.4. Nivel de E/S (Nivel actuador/sensor) 10
2.5. Nivel de Control (Nivel de célula) 10
2.6. Nivel de Campo y Proceso 11
2.7. Nivel de gestión (Nivel de fábrica) 11
2.8. Protocolo MODBUS 11
2.9. MODBUS RTU 12
2.10. Transmisión en MODBUS/RTU 13
2.11. MODBUS ASCII 14
2.12. MODBUS TCP/IP 14
2.13. Descripción del protocolo MODBUS 15
2.14. Trama del protocolo Modbus 17
2.14.1. Trama Maestro 17
2.14.2. Trama Esclavo 17
2.14.3. Tamaño de la trama 18
2.14.4. Funciones del protocolo 18
2.14.5. Funciones más comunes del protocolo Modbus 19
2.14.6. Read coils 20
2.14.7. Read discrete inputs 20
2.14.8. Read holding register 20
2.14.9. Read input register 20
2.15. Norma física RS485 21

2.16. Medio Físico 25
2.17. ¿Cómo se conectan las interfaces RS485/RS422 de los módulos SIMATIC y SIPLUS para la comunicación serie? 25
2.18. Motor eléctrico trifásico 26
2.18.1. Tipos de motores trifásicos 27
2.19. Variador de Frecuencia 27
2.19.1. Módulo adaptador Modbus RTU FMBA-01 28
3. MARCO METODOLÓGICO 29
3.1. Diseño del módulo de prácticas. 29
3.2. Desarrollo de la estructura del tablero. 29
3.3. Diagrama de conexiones 31
3.4. Cableado de los elementos del módulo 33
3.5. Acabado y detalles de los módulos 34
3.6. Fabricación de cable macho – macho RS485 35
3.7. Adaptación de puerto RS485 a planta de control de nivel 36
3.8. Adaptación de puerto RS485 a módulo variador de frecuencia 36
3.9. Siemens S7-1200 37
3.10. Signal Board de salidas analógicas 39
3.11. Switch CSM 1277. 39
3.12. Fuente de alimentación PM 1207 40
3.13. Relés 40
3.14. Módulo de comunicación CM1241 41
3.15. KTP600 Basic PN 41
3.16. Pulsantes 42
3.17. Luces piloto 43
3.18. Selectores 43
3.19. Voltímetro 43
3.20. Potenciómetros 44

3.21. Borneras tipo banana (hembra) 44
3.22. Cable Convertidor USB – RS485 45
3.23. Características del cable convertidor USB a RS485 46
3.24. Especificaciones del cable convertidor USB a RS485 46
3.25. Fuente de voltaje variable 47
3.26. TIA Portal V12 48
3.26.1. Tareas del TIA Portal 49
3.26.2. Ventajas del TIA Portal 49
3.27. Simulador Modbus Poll 49
3.27.1. Variantes de protocolo compatibles: 50
3.27.2. Datos y características técnicas de MODBUS POLL: 50
3.27.3. Funciones de Modbus soportadas: 51
3.27.4. Visión de conjunto. 51
3.27.5. Cambiar la definición de lectura / escritura 52
3.28. Simulador Modbus Slave. 53
3.28.1. Variantes de protocolo compatibles: 54
3.28.2. Características y características técnicas del esclavo MODBUS: 54
3.28.3. Funciones de Modbus soportadas: 54
4. PRÁCTICAS DE LABORATORIO 55
4.1. Práctica Nº 1 56
4.2. Práctica Nº 2 93
4.3. Práctica Nº3 102
4.4. Práctica Nº4 116
4.5. Práctica Nº 5 138
4.6. Práctica Nº 6 150
4.7. Práctica Nº 7 158
4.8. Práctica Nº 8 172

4.9. Práctica Nº 9 189
4.10. Práctica 10..... 204
5. RESULTADOS 220
5.1. ANÁLISIS DE RESULTADOS..... 221
6. CONCLUSIONES..... 226
7. RECOMENDACIONES..... 227
8. BIBLIOGRAFÍA 228

ÍNDICE DE FIGURAS

Figura 1 Universidad Politécnica Salesiana sede Guayaquil 3
Figura 2: Evolución de procesos industriales [1]. 6
Figura 3: Instalación industrial con control centralizado [1]. 9
Figura 4: Instalación industrial con control distribuido [1]. 9
Figura 5: Evolución de procesos industriales [1]. 10
Figura 6: Configuración de MODBUS en RS485 [2]. 12
Figura 7: Trama MODBUS/RTU [3]. 13
Figura 8: Trama MODBUS ASCII [3]. 14
Figura 9: Estructura de la trama de un maestro Modbus [4]. 17
Figura 10: Estructura de la trama del esclavo [4]. 18
Figura 11: Tamaño de cada campo de la trama del protocolo Modbus [4]..... 18
Figura 12: Funciones públicas disponibles en Modbus [4]..... 19
Figura 13: Señales eléctricas por la interface RS-485 [6]. 21
Figura 14: Transmisión diferencial aplicada a la norma RS-485 [6]. 22
Figura 15: Ejemplo de una red Profibus con diferentes componentes [6]. 22
Figura 16: Configuración multifilar de una red en la norma RS-485 [6]..... 23
Figura 17: Circuitos que integran la norma RS-485 [6]..... 24
Figura 18: Módulo CM1241 RS485 [7]. 25
Figura 19: Configuración de pines para puerto DB9 en RS485 [8]. 26
Figura 20: Partes del motor trifásico [9]. 27
Figura 21: Variador de frecuencia ABB ACS355 [10]. 28
Figura 22: Módulo FMBA-01 [10]. 28
Figura 23: Estructura física 29
Figura 24: Diseño CAD de los módulos didácticos. 30

Figura 25: Estructura metálica. 30
Figura 26: Montaje de elementos del tablero. 31
Figura 27: Entradas digitales [11]. 31
Figura 28: Salidas digitales [11]. 32
Figura 29: Entradas y salidas analógicas [11]. 32
Figura 30: Cableado de elementos del módulo. 33
Figura 31: Conexionado entre elementos internos del tablero. 33
Figura 32: Conexionado de PLC. 34
Figura 33: Marquillado de los elementos del tablero. 34
Figura 34: Estética de los módulos terminados. 35
Figura 35: Fabricación de cable macho – macho RS485. 35
Figura 36: Instalación de puerto RS485 para planta de control de nivel. 36
Figura 37: Instalación de puerto RS485 para módulo de variador. 37
Figura 38: Estructura metálica 38
Figura 39: Signal board. 39
Figura 40: Switch CSM 1277. 39
Figura 41: Fuente de alimentación PM 1207. 40
Figura 42: Relés tipo electromagnético. 40
Figura 43: Modulo de comunicación CM1241. 41
Figura 44: Parte frontal del KTP600 Basic PN. 41
Figura 45: Parte posterior del KTP600 Basic PN. 42
Figura 46: Pulsante. 42
Figura 47: Luz piloto. 43
Figura 48: Selector. 43
Figura 49: Voltímetro. 44
Figura 50: Potenciómetro de precisión. 44

Figura 51: Borneras banana hembra. 45
Figura 52: Cable convertidor USB a RS485. 45
Figura 53: Diagrama de conexión de fuente regulada [14]. 47
Figura 54: Fuente regulada a 10 VDC. 47
Figura 55: Distribución de proceso [16]. 48
Figura 56: Interfaz de conexión. 51
Figura 57: Cuadro de diálogo edición esclavo. 52
Figura 58: Barra de herramientas. 52
Figura 59: Ventana de configuración. 53
Figura 60: Elementos a utilizar en las prácticas de laboratorio. 55
Figura 61: Arquitectura de red. 57
Figura 62: Diagrama de conexiones. 57
Figura 63: Conexión del cable de poder 120 VAC. 58
Figura 64: Conexión del cable Ethernet en el puerto P1. 58
Figura 65: Interfaz de inicio. 59
Figura 66: Interfaz primeros pasos. 59
Figura 67: Interfaz Agregar dispositivo. 60
Figura 68: Árbol del proyecto. 60
Figura 69: Asignación de Signal Board. 61
Figura 70: Asignación de Módulo de comunicación. 61
Figura 71: Configuración del puerto RS485. 62
Figura 72: Interfaz para agregar al PLC SLAVE. 63
Figura 73: Interfaz para agregar HMI. 63
Figura 74: Conexiones de PLC. 64
Figura 75: Formato de imagen. 64
Figura 76: Avisos. 65

Figura 77: Botones. 65
Figura 78: Vista topológica. 66
Figura 79: Direcciones Ethernet. 66
Figura 80: Activación de marcas de sistema. 67
Figura 81: Creación de imagen Menú. 68
Figura 82: Diseño de imagen Menú en HMI_1_MASTER. 68
Figura 83: Agregar elementos a la imagen. 69
Figura 84: Asignación de gráfico para el botón. 69
Figura 85: Selección del evento para Botón_1. 70
Figura 86: Diseño de imagen Proceso. 70
Figura 87: Lista de textos PRACTICAS. 71
Figura 88: Asignación de variable para Campo E/S simbólico. 71
Figura 89: Diseño simulado de la imagen Práctica 1. 72
Figura 90: Diseño de la imagen Práctica 1. 72
Figura 91: Variables de la práctica 1. 73
Figura 92: Asignación de variables para los campos de la práctica 1. 73
Figura 93: Propiedades del botón Ejecutar. 74
Figura 94: Eventos del botón Ejecutar. 74
Figura 95: Eventos del botón Sí. 74
Figura 96: Eventos del botón No. 75
Figura 97: Agrupación de elementos del mensaje. 75
Figura 98: Diseño terminado de imagen Título 1. 75
Figura 99: Animación del botón Habilitar. 76
Figura 100: Asignación de variable de botón Habilitar. 77
Figura 101: Eventos del botón Habilitar. 77
Figura 102: Animación del botón Deshabilitar. 77

Figura 103: Eventos del botón Deshabilitar. 78
Figura 104: Descripción de botones Habilitar y Deshabilitar. 78
Figura 105: Grupo creado para el mensaje de alerta. 78
Figura 106: Propiedades de grupo de mensaje de alerta. 79
Figura 107: Animaciones de grupo de mensaje de alerta. 79
Figura 108: Eventos del grupo de mensaje de alerta 79
Figura 109: Ubicación del interfaz de las instrucciones MODBUS. 81
Figura 110: Instrucción PUERTO COM. 81
Figura 111: Configuración de instrucción PUERTO_COM MASTER. 84
Figura 112: Configuración de instrucción PUERTO_COM SLAVE. 84
Figura 113: Instrucción MASTER. 85
Figura 114: Creación del bloque de datos PARAMETERS. 85
Figura 115: Variables de configuración del puerto MASTER práctica 1. 86
Figura 116: Creación del bloque de datos CONFIGURACION. 86
Figura 117: Variables de habilitación. 87
Figura 118: Configuración de instrucción MASTER. 88
Figura 119: Creación del bloque de datos PRACTICAS – práctica 1. 90
Figura 120: Señal de reloj para las instrucciones MODBUS. 90
Figura 121: Configuración Modbus Slave. 91
Figura 122: Configuración de la conexión. 91
Figura 123: Datos enviados desde el simulador. 92
Figura 124: Datos recibidos en el PLC. 92
Figura 125: Arquitectura de red. 94
Figura 126: Diagrama de conexiones. 94
Figura 127: Parámetros de la práctica 2. 95
Figura 128: Variables de registro del PLC MASTER – práctica 2. 95

Figura 129: Variables de registro del PLC MASTER – práctica 2. 95
Figura 130: Configuración del bloque de escritura práctica 2. 96
Figura 131: Configuración del bloque de lectura práctica 2. 96
Figura 132: Configuración del bloque SLAVE práctica 2. 97
Figura 133: Bloques MOVE de la práctica 2. 97
Figura 134: Creación de la imagen Titulo 2. 98
Figura 135: Diseño de la imagen Practica 2. 98
Figura 136: Propiedades de la entrada ES_1. 99
Figura 137: Propiedades de la entrada ES_2. 99
Figura 138: Propiedades de la entrada ES_3. 100
Figura 139: Propiedades de la entrada ES_4. 100
Figura 140: Diseño simulado de la práctica 2. 101
Figura 141: Arquitectura de red. 103
Figura 142: Diagrama de conexiones. 103
Figura 143: Parámetros de la práctica 3 MASTER. 104
Figura 144: Parámetros de la práctica 3 SLAVE. 104
Figura 145: Variables de registro para el PLC MASTER – práctica 3. 104
Figura 146: Variables de registro para el PLC SLAVE – práctica 3. 105
Figura 147: Configuración del bloque MB_MASTER/ PLC MASTER práctica 3. 105
Figura 148: Configuración del bloque MB_SLAVE_DB/ PLC MASTER práctica 3. 106
Figura 149: Configuración del bloque MB_SLAVE_DB/ PLC SLAVE práctica 3 106
Figura 150: Configuración del bloque MB_MASTER/ PLC SLAVE práctica 3. ... 107
Figura 151: Diseño de circuito de control para el PLC MASTER. 107
Figura 152: Diseño de circuito de control para el PLC SLAVE. 108
Figura 153: Creación de la imagen Titulo 3. 109

Figura 154: Diseño de imagen Practica 3. 109
Figura 155: Composición de animación Grupo 1 – Caldero..... 110
Figura 156: Variable asignada para la animación de motor..... 110
Figura 157: Variable asignada para la animación del caldero. 111
Figura 158: Variable asignada para la salida del motor. 111
Figura 159: Propiedades del botón para selección de modo. 112
Figura 160: Eventos del botón para selección de modo..... 112
Figura 161: Asignación de variable de Botón_8 de selección de modo. 113
Figura 162: Asignación de variable y animación de Botón_11 de selección de modo. 113
Figura 163: Asignación de variable para el evento del botón Start. 114
Figura 164: Asignación de variable para el evento del botón Stop..... 114
Figura 165: Diseño del HMI_1_SLAVE de la práctica 3. 115
Figura 166: Arquitectura de red. 117
Figura 167: Diagrama de conexiones..... 117
Figura 168: Creación del PLC PLANTA y bloques de organización. 118
Figura 169: Parámetros del PLC MASTER, práctica 4. 118
Figura 170: Variables de registro del PLC MASTER, práctica 4. 119
Figura 171: Variables de registro del PLC PLANTA, práctica 4. 119
Figura 172: Configuración de instrucción MB_COMM_LOAD para PLANTA. ... 120
Figura 173: Bloque Slave de práctica 4. 120
Figura 174: Señal de sensor ultrasónico. 121
Figura 175: Ecuación de la recta. 121
Figura 176: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 4. 122
Figura 177: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 4. 122

Figura 178: Selección de modo para el PLC MASTER, práctica 4. 123
Figura 179: Marcas de apagado de controles manual / automático. 124
Figura 180: Activación manual de bomba y electroválvula. 124
Figura 181: Registro de nivel en variable NIVEL PLANTA. 125
Figura 182: Control de Segmento 1, PLANTA. 125
Figura 183: Activación y desactivación de la Bomba. 126
Figura 184: Activación y desactivación de la Electroválvula. 127
Figura 185: Llenado automático del tanque. 128
Figura 186: Vaciado automático del tanque. 128
Figura 187: Variable de bloqueo de nivel. 128
Figura 188: Comparación de nivel máximo del tanque TK102. 129
Figura 189: Diseño de imagen Título 4. 129
Figura 190: Creación de imagen Práctica 4. 130
Figura 191: Representación del grupo Sistema. 130
Figura 192: Animación de botón Automático. 131
Figura 193: Evento de botón Automático. 131
Figura 194: Animación de botón Manual. 131
Figura 195: Evento de botón Manual. 131
Figura 196: Animación de botón Paro. 132
Figura 197: Evento de botón Paro. 132
Figura 198: Animación de botón X. 132
Figura 199: Evento de botón X. 132
Figura 200: Creación de la Barra de nivel para el tanque TK102. 133
Figura 201: Asignación de variable NIVEL PLANTA. 133
Figura 202: Configuración de botón electroválvula. 134
Figura 203: Animación de botón electroválvula. 134

Figura 204: Animación de botón bomba. 134
Figura 205: Configuración de botón bomba. 135
Figura 206: Representación de sensor de boya. 135
Figura 207: Representación de sensor de nivel alto. 136
Figura 208: Representación de sensor de nivel bajo. 136
Figura 209: Arquitectura de red. 139
Figura 210: Circuito de fuerza. 139
Figura 211: Parámetros del PLC MASTER, práctica 5. 140
Figura 212: Parámetros del PLC SLAVE, práctica 5. 140
Figura 213: Variables de registro para el PLC MASTER – práctica 5. 141
Figura 214: Variables de registro para el PLC SLAVE – práctica 5. 141
Figura 215: Bloque Master de parámetro 5302. 142
Figura 216: Bloque Master de parámetro 5310. 143
Figura 217: Bloque Master de parámetro 1102. 143
Figura 218: Bloque Máster de parámetro 1102. 144
Figura 219: Parámetros de control del variador ACS355 [18]. 145
Figura 220: Configuración de la comunicación Modbus 5302 [18]. 145
Figura 221: Parámetros de comunicación Modbus 5310 [18]. 146
Figura 222: Parámetros de comunicación Modbus 1102 [18]. 146
Figura 223: Parámetros necesarios para motores síncronos de imanes permanentes [18]. 147
Figura 224: Puesta a cero de los controles de visualización. 147
Figura 225: Creación de imagen Práctica 5. 147
Figura 226: Creación de imagen Parametros_VLT. 148
Figura 227: Animación y variable de lista de parámetros. 148
Figura 228: Eventos y variable de lista de parámetros. 149

Figura 229: Arquitectura de red. 151
Figura 230: Diagrama de conexiones. 151
Figura 231: Parámetros del PLC MASTER, práctica 6. 152
Figura 232: Parámetros del PLC SLAVE, práctica 6. 152
Figura 233: Variables de registro para el PLC MASTER – práctica 6. 152
Figura 234: Variables de registro para el PLC SLAVE – práctica 6. 153
Figura 235: Configuración del bloque escritura MB_SLAVE/ PLC MASTER práctica 6. 153
Figura 236: Configuración del bloque escritura MB_SLAVE/ PLC SLAVE práctica 6. 154
Figura 237: Diseño de imagen Título 6, práctica 6. 154
Figura 238: Diseño de la imagen Practica 6. 155
Figura 239: Configuración de campos de E/S para PLC MASTER, práctica 6. 155
Figura 240: Configuración de campos de E/S para PLC SLAVE, práctica 6. 156
Figura 241: Modbus Poll configurado para la práctica 6. 156
Figura 242: Configuración en modo lectura, práctica 6. 157
Figura 243: Parámetros de conexión del Modbus Poll. 157
Figura 244: Arquitectura de red. 159
Figura 245: Circuito de fuerza. 159
Figura 246: Parámetros del PLC MASTER, práctica 7. 160
Figura 247: Parámetros del PLC SLAVE, práctica 7. 160
Figura 248: Variables de registro para el PLC MASTER – práctica 7. 161
Figura 249: Variables de registro para el PLC MASTER – práctica 7. 161
Figura 250: Bloque de escritura para relé, práctica 7. 162
Figura 251: Bloque de escritura para velocidad, práctica 7. 163
Figura 252: Bloque de lectura para frecuencia, corriente y potencia, práctica 7. 163

Figura 253: Marcha y Paro del sistema, práctica 7. 164
Figura 254: Creación de bloque SIGNAL_ANALOGA. 164
Figura 255: Aplicación de bloque SIGNAL_ANALOGA. 164
Figura 256: Incremento y reducción manual de la velocidad del motor. 165
Figura 257: Bloqueo para la velocidad máxima. 166
Figura 258: Mensajes de avisos de límite superior de velocidad del motor. 166
Figura 259: Mensajes de avisos de límite inferior de velocidad del motor. 166
Figura 260: Lectura de frecuencia, corriente y potencia del motor. 167
Figura 261: Diseño de la imagen Titulo 7. 167
Figura 262: Diseño de la imagen Practica 7. 168
Figura 263: Eventos del botón Marcha, practica 7. 168
Figura 264: Eventos del botón Marcha, practica 7. 168
Figura 265: Propiedades del campo E/S 3. 169
Figura 266: Propiedades del grupo 3. 170
Figura 267: Propiedades de visor de gráfico. 170
Figura 268: Animación de motor, práctica 7. 171
Figura 269: Animación del campo E/S 2. 171
Figura 270: Arquitectura de red. 173
Figura 271: Diagrama de conexiones. 174
Figura 272: Parámetros del PLC MASTER, práctica 8. 174
Figura 273: Variables de registro del PLC MASTER, práctica 8. 175
Figura 274: Variables de registro del PLC PLANTA, práctica 8. 175
Figura 275: Bloque Slave de práctica 8. 175
Figura 276: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 8. 176

Figura 277: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 8. 177
Figura 278: Selección de modo para el PLC MASTER, práctica 8. 177
Figura 279: Marcas de apagado de controles manual / automático. 178
Figura 280: Activación manual de bomba y electroválvula. 178
Figura 281: Registro de nivel en variable NIVEL PLANTA. 178
Figura 282: Control de Segmento 1 manual, PLANTA. 179
Figura 283: Control de Segmento 1 automático, PLANTA. 179
Figura 284: Activación y desactivación de la Bomba. 180
Figura 285: Activación y desactivación de la Electroválvula. 181
Figura 286: Llenado automático del tanque. 182
Figura 287: Vaciado automático del tanque. 182
Figura 288: Variable de bloqueo de nivel 182
Figura 289: Comparación de nivel máximo del tanque TK102. 183
Figura 290: Envío de bits de E/S. 183
Figura 291: Diseño de imagen de presentación SCADA. 184
Figura 292: Diseño de imagen Practica # 8 SCADA. 184
Figura 293: Configuración botón salir de la pantalla Presentación. 185
Figura 294: Configuración botón Next de la pantalla Presentación. 185
Figura 295: Configuración de Barra_1. 186
Figura 296: Configuración de Campo ES_1 nivel. 186
Figura 297: Configuración de Grupo_16 electroválvula manual. 187
Figura 298: Configuración de Grupo_8 bomba manual. 187
Figura 299: Arquitectura de red. 190
Figura 300: Diagrama de conexiones. 190
Figura 301: Parámetros del PLC MASTER, práctica 9. 191

Figura 302: Variables de registro del PLC MASTER, práctica 9. 191
Figura 303: Variables de registro del PLC PLANTA, práctica 9. 192
Figura 304: Bloque Slave de práctica 9. 192
Figura 305: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 9. 193
Figura 306: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 8. 194
Figura 307: Activación de sistema vía SCADA. 194
Figura 308: Almacenamiento de datos de Histéresis y Setpoint. 195
Figura 309: Bloqueos del sistema. 195
Figura 310: Operación matemática de histéresis. 196
Figura 311: Bloqueo para límite superior de setpoint. 197
Figura 312: Bloqueo para límite superior de histéresis. 197
Figura 313: Control de encendido de bomba y electroválvula. 198
Figura 314: Visualización de Histéresis y Setpoint. 198
Figura 315: Activación / desactivación de electroválvula. 199
Figura 316: Diseño de imagen Practica 9. 199
Figura 317: Configuración de Grupo, botón Star / Stop. 200
Figura 318: Configuración de barra, tanque TK2. 200
Figura 319: Visor de curvas para histéresis. 201
Figura 320: Asignación de variables para gráfico de histéresis. 201
Figura 321: Visualización de resultados de histéresis. 202
Figura 322: Visualización de resultados de histéresis. 202
Figura 323: Arquitectura de red. 205
Figura 324: Diagrama de conexiones. 205
Figura 325: Parámetros del PLC MASTER, práctica 10. 206

Figura 326: Variables de registro del PLC MASTER, práctica 10. 207

Figura 327: Variables de registro del PLC PLANTA, práctica 10. 207

Figura 328: Bloque Slave de práctica 10. 208

Figura 329: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 10. 208

Figura 330: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 10. 209

Figura 331: Bloque de organización para PID. 209

Figura 332: Programación del Segmento 1 PID Compact. 210

Figura 333: Programación de Segmento 2. 210

Figura 334: Registro de Ti, Td, Kp. 211

Figura 335: Inicio de PID. 211

Figura 336: Recepción y bloqueo de setpoint. 212

Figura 337: Registro de parámetros PID. 212

Figura 338: Controles de bloqueo y desbloqueo para HMI y SCADA. 213

Figura 339: Registro de datos en variables del PLC MASTER. 213

Figura 340: Borrado de registros PID. 214

Figura 341: Diseño de imagen Titulo 10. 214

Figura 342: Creación de gráfico de la curva PID. 215

Figura 343: Diseño de imagen Practica 10. 215

Figura 344: Curva a lazo abierto, Ziegler - Nichols 216

Figura 345: Recta tangente para puntos de inflexión. 216

Figura 346: Comportamiento de grafica PD. 217

Figura 347: Comportamiento de grafica PI. 218

Figura 348: Comportamiento de gráfica PID. 218

Figura 349: Tiempos de gráfica PID. 219

Figura 350: Exposición en seminario de S7-1200. 221
Figura 351: Presentación de diapositivas en seminario de S7-1200. 222
Figura 352: Exposición en seminario de Variadores. 222
Figura 353: Presentación de diapositivas en seminario de Variadores. 223

ÍNDICE DE TABLAS

Tabla 1: Comparación de tramas ASCII-RTU [5]. 16

Tabla 2: Señales del estándar [6]. 21

Tabla 3: Características de estándares de comunicación [6]. 24

Tabla 4: Partes de un PLC S7-1200. 38

Tabla 5: Partes del KTP600 Basic PN. 42

Tabla 6: Rangos de direcciones del dispositivo y MODBUS [17]. 53

Tabla 7: Parámetros de la instrucción MB_COMM_LOAD [15]. 81

Tabla 8: Parámetros de la instrucción MB_MASTER [15]. 87

Tabla 9: Funciones de configuración para el DATA_ADDR [15]. 89

Tabla 10: E/S de la planta de control de nivel. 123

Tabla 11: Resultados de tiempos de gráficas. 219

INTRODUCCIÓN

En este proyecto se implementó con el objetivo de equipar al laboratorio de automatización industrial con dos módulos repotenciados para la UPS-G, el mismo que permitirá a los estudiantes conocer técnicas bastante utilizadas en la industria.

El protocolo RS485 ha estado hábil en los módulos anteriores pero no se ha explotado completamente las bondades y funciones que brinda de forma experimental.

Los módulos permiten realizar prácticas de diferentes sistemas de control automatizados, sistemas SCADA, control PID, procesos y circuitos simulados. El sistema consta de variables de entrada, procesamiento de datos, y obtención de resultados a través de las salidas. Estos módulos son capaces de concatenar diferentes tipos de dispositivos los cuales son herramientas fundamentales para las prácticas de laboratorio y a nivel industrial. Una de las funciones importantes es la configuración de maestro / esclavo, el cual sirve para comandar varios elementos de un sistema gracias a la transmisión de datos del maestro. Este proyecto va ligado a un software el cual servirá para la programación de los elementos según lo que se desee diseñar.

Este proyecto ofrece nuevo material de trabajo para los docentes con la premisa de que los estudiantes adquieran nuevos conocimientos mejorando su formación académica y profesional y destacándose en las materias de la malla curricular.

Adicionalmente se ha realizado la migración de los PLC S7-1200 del laboratorio de automatización industrial a unas nuevas y mejoradas estructuras, añadiendo el puerto DB9 el cual se utilizó como vía de comunicación con los demás módulos a través del estándar RS485.

1. EL PROBLEMA

1.1. ANTECEDENTES

El desarrollo de proyectos en la carrera de Electrónica de la Universidad Politécnica Salesiana Sede Guayaquil permite a cada estudiante potencializar los conocimientos adquiridos, es por ello que se da paso a que se generen nuevas propuestas con el fin de lograr mejoras académicas y por consiguiente futuros profesionales que lideren el campo de la Electrónica. Por este motivo se **plantea el diseño, implementación de una red MODBUS/RTU entre dos autómatas programables S7-1200 basado en el estándar RS485, para el laboratorio de automatización industrial de la UPS-GYE,** que actualmente cuenta con módulos didácticos para el desarrollo de prácticas con nuevas herramientas tecnológicas para las futuras generaciones de estudiantes.

Es relevante indicar que los módulos existentes permiten experimentar con redes PROFINET y ETHERNET, pero también se puede explotar las bondades técnicas y sus ventajas de comunicación con otros protocolos, con esto se ha dejado a un lado un importante recurso que en su momento hubiera permitido a los alumnos obtener un conocimiento amplio sobre los tipos de comunicaciones industriales.

1.2. IMPORTANCIA Y ALCANCE

El desarrollo del proyecto se inició con la implementación de una red industrial capaz de comunicar los módulos de automatización con eficiencia para la transmisión y recepción de datos.

Este proyecto va dirigido esencialmente para alumnos y docentes de la Universidad Politécnica Salesiana de Guayaquil (UPS-G) especialmente para la carrera de ingeniería electrónica mención sistemas industriales.

Esta herramienta de comunicación impulsará el aprendizaje de nuevos protocolos de comunicación industrial en futuras nuevas prácticas con los módulos que posee la

universidad. A su vez partiendo de la necesidad de conocer nuevos protocolos de comunicación se puede indicar que este proyecto permitirá incentivar a los futuros estudiantes a desarrollar nuevas propuestas que generen oportunidades a otros estudiantes.

1.3. DELIMITACIÓN

1.3.1. Espacial

El proyecto se implementó en el laboratorio de automatización industrial del bloque B de la Universidad Politécnica Salesiana sede Guayaquil (Véase Figura 1).

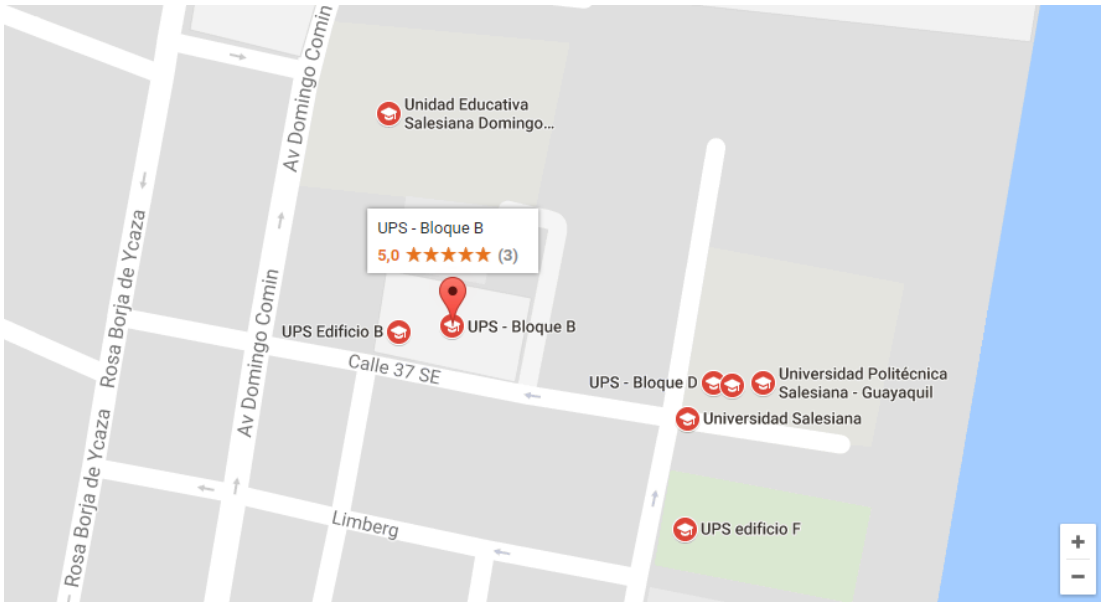


Figura 1 Universidad Politécnica Salesiana sede Guayaquil

Recuperado de: https://www.google.com.ec/maps

1.3.2. Temporal

El proyecto técnico se efectuó tomando como base de estudio el periodo comprendido entre los meses de abril del 2017 hasta abril del año 2018.

1.3.3. Sectorial

El presente proyecto técnico tuvo su desarrollo, en el sector sur del Cantón Guayaquil de la provincia del Guayas, específicamente en las calles Av. Domingo Comín y Chambers.

1.3.4. Académica

El alcance del proyecto técnico, respecto al marco académico se ha ejecutado en función de la aplicación de las asignaturas que se han estudiado a lo largo de la malla universitaria tales como: electrónica digital, electrónica analógica, automatización industrial, sistemas microprocesados, sensores y transductores, electrónica de potencia e instalaciones industriales.

1.4. JUSTIFICACIÓN DEL PROYECTO

En la Universidad Politécnica Salesiana sede Guayaquil se han realizado varios proyectos estudiantiles que han dado paso al desarrollo de mejoras de equipos eléctricos y electrónicos; es por ello que se ha considerado aportar a la universidad una propuesta innovadora que consiste en la aplicación de la red de comunicación MOBBUS /RTU.

Es importante mencionar que se debe resaltar la utilización de los equipos de comunicación MOBBUS/RTU vía RS485 ya que con la aplicación y uso de los mismos contaremos con módulos capaces de interactuar entre sí y realizar prácticas de automatización industrial no realizadas antes, con mayor eficiencia en transferencia de datos. En la actualidad el uso de una red MODBUS/RTU es muy usada para ayudar a concatenar procesos en plantas industriales, como comunicar dos autómatas y variadores de frecuencia que integran de serie el puerto RS485 y con esto tener lectura y escritura en el accionamiento de elementos.

OBJETIVOS

1.4.1. Objetivo general

- Implementar una red de comunicación industrial Maestro-Esclavo vía MODBUS/RTU entre dos autómatas programables S7-1200 utilizando el módulo RS485, mejorando su capacidad de transmisión de datos para prácticas de control y automatización industrial.

1.4.2. Objetivos específicos

- Verificar el correcto funcionamiento de dos módulos SIEMENS S7 1200 del laboratorio de automatización industrial otorgando las condiciones mínimas para la realización de comunicaciones entre sí.
- Ejecutar 10 prácticas de comunicaciones MODBUS/RTU utilizando el programa TIA PORTAL V.12
- Reemplazar las carcasas de dos módulos SIEMENS S7-1200 actuales a un diseño mejorado y simplificado.
- Diseñar e implementar una estructura y controlador tipo PID para una planta didáctica de control de nivel con sensor de entrada análoga al PLC.
- Parametrizar un variador de frecuencia para el control y manejo del PID.
- Realizar comparaciones PI, PD, PID, etc.

2. FUNDAMENTACIÓN TEÓRICA

2.1. Evolución de Procesos Industriales

[1] Hasta el siglo xx, el crecimiento tecnológico había llegado de la zarpa de la revolución artificial en la que los procedimientos automáticos eran obras clave para la pericia de los desarrollos artificiales. Pero es en el siglo xx cuando los sistemas de mecanización surgen con más redundancia, potenciados por los sistemas eléctricos basados en relés electromagnéticos en la primera mitad del siglo, y de los sistemas electrónicos de estado denso en su segunda parte. Así, a más de sus objetivos iniciales, el criterio de automatización se extiende a la rebaja de desarrollo, deducción de contratiempos gremiales, rebaja de precios, etc. La gestación de los autómatas programables marcó un proceso en la habilidad de acoplamiento y líneas de fabricación. La definición de línea de obtención automatizada apareció entonces como un hecho real. Este pequeño mecanismo permitió que los métodos de procesos industriales fuesen desde entonces más capaces, precisos, y lo que es más importante, reprogramables, eliminando el gran costo que se producía al reemplazar el difícil método de cuidado asentado en relés y contactores, en dimensiones y duración.

Figura 2: Evolución de procesos industriales [1].

La evolución en los procesos de automatización pasa actualmente por el avance de las redes de comunicación. La comunicación de sistemas y procesos industriales no

es un tema nuevo, ya que es grandemente conocido en el uso de sistemas como IEEE-488 y RS485/422 que durante más de 20 años han sido responsables de ofrecer las bondades minimas en las instalaciones de baja y media complejidad en cuanto a las capacidades de intercomunicación se refiere. Este tipo de vínculos entre métodos se ha invertido esencialmente para equipos de instrumentación y sistemas de automatización, adonde es necesaria una baja tasa de transferencia de numeros entre equipos, sin embargo que en gran espectáculo de eventos en la actualidad en momento ahora no puede reponer a las carencias de intercomunicación entre artefactos que se demandan. Por ello, redes de comunicación como Profibus y Profinet han obtenido ser indispensables en un círculo de trabajo adonde cada momento es más necesaria la integración total [1].

2.2. Sistemas industriales de Control

Existen tres clases de sistemas de control industrial los cuales son: control centralizado, control híbrido y control distribuido. La gran importancia de las funciones a desarrollar, o la opción de distribuir los trabajos de control del proceso o conjunto de máquinas en esas funciones autónomas, indicará en muchos casos la elección de un tipo de control [1].

2.2.1. Control centralizado

Este proceso es el que se sigue en caso de sistemas poco dificultosos, donde un sistema puede ser verificado directamente con un único elemento de control responsable de ejecutar todas las tareas del proceso de producción y que podría incluir un sistema de monitorización y supervisión. En base a las necesidades de producción han requerido mayor desarrollo, una tendencia ha sido la de usar elementos de control más elaborados y potentes, respaldando todo el control del proceso, con la dificultad que ello supone, ya que se hace necesario hacer llegar todas las señales de sensores y conexionar todos los actuadores donde estén. Como virtud de esta metodología se tiene que no es necesario implementar un sistema de comunicación entre procesos, ya que todas las señales están enlazadas por el mismo sistema. Por otro lado, para sistemas poco dificultosos, posee un menor gasto económico. Al contrario, posee muchas desventajas, ya que si el proceso falla, toda el sistema queda paralizado, siendo necesario un proceso repetidor para evitar estas

eventualidades. También es necesario el uso de unidades de control (mayormente PLC) de superior capacidad de proceso, dada la complejidad de los problemas que debe abarcar y las restricciones de tiempo límite que son acostumbrados a nivel industrial. De otra manera, el conexionado puede incrementar notablemente debido a la gran distancia que puede haber entre los sensores, actuadores y la unidad de control, aunque este inconveniente se pueda reducir en cierta medida debido al uso de buses de datos [1].

2.2.2. Control distribuido

El control distribuido necesita considerar procesos, grupos de procesos o áreas de trabajo susceptibles para ser definidas por un algoritmo de control, y que de pueda ejecutar de forma autónoma. A cada unidad se destinará un autómata dimensionado de acuerdo con los requerimientos del sistema que se empleen. Debido a la similitud que existe entre las operaciones que hay en cada proceso, se debe considerar que es necesario enlazar los autómatas entre ellas mediante entradas y salidas digitales, o a su vez a una red de comunicaciones para intercambio de datos y estados. De por sí, el autómata o controlador designado debe permitir las comunicaciones. Con este concepto de control es posible que cada unidad funcional se base en un proceso relativamente sencillo comparado con el proceso general, disminuyendo la posibilidad de errores en la programación y otorgando el empleo de unidades de control (PLC principalmente) más fáciles y económicas. De igual manera, la existencia de fallos en unidades de control no radica totalmente en la paralización de todos los procesos que se den en el interior de la planta. Como desventaja, es necesario realizar un estudio de implantación previo, ya que se necesita identificar los procesos automáticos, asignar elementos a cada proceso y diseñar el modelo de conexionado para resolver las necesidades del proceso planificado [1].

2.2.3. Control híbrido

El control híbrido no está muy bien conceptualizado, ya que este tipo de gestión de planta puede considerarse a cualquier proceso de distribución de elementos de control mezclado entre el control distribuido y el centralizado. En muchas ocasiones no resulta fácil separar los procesos de manera cien por ciento automática, por lo que se debe acudir al desarrollo de varios procesos desde una misma unidad de control,

porque la dificultad de la separación es mayor que la dificultad que supone su gestión conjunta. De plano general, una estrategia de este tipo también termina a una gestión estructurada, de manera que hay elementos de control de nivel superior que supervisan y enlazan los procesos automáticos más fáciles, siendo los responsables de gestionar la información común. Para esta estructura también es necesario el uso de redes de comunicación [1].

Figura 3: Instalación industrial con control centralizado [1].

Figura 4: Instalación industrial con control distribuido [1].

2.3. La pirámide CIM

La idea de fabrica totalmente automatizada (Computer Integrated Manufacturing) se denota como una pirámide en la que en los niveles inferiores se hallan los sensores y actuadores; en los niveles medios se entrelazan estos componentes para trabajar en conjunto ejecutando funciones sincronizadas y por último, en el nivel superior

aparece la red informática técnico-administrativa donde se recopilan datos de estado, registros históricos, datos de partida, consignas, etc. [1].

Figura 5: Evolución de procesos industriales [1].

2.4. Nivel de E/S (Nivel actuador/sensor)

También conocido como nivel de instrumentación. Está compuesto por los elementos de medida "sensores" y mando "actuadores" configurados en una línea de producción. Son los elementos más evidentemente ligados con el proceso productivo. De esta manera, los actuadores son los responsables de ejecutar las instrucciones de los elementos de control para cambiar el proceso productivo y los sensores miden variables en el proceso de producción. Ejemplos de sensores son aquellos que permiten medir nivel de líquidos, caudal, temperatura, presión, posición, etc. Para ejemplos de actuadores se mencionan los motores, válvulas, calentadores, taladros, cizallas, etc. Como característica adicional, los sensores y actuadores suelen ser dispositivos que necesitan ser controlados por otros elementos [1].

2.5. Nivel de Control (Nivel de célula)

Todos los dispositivos de control existentes en planta es posible monitorearlos si existe un sistema de comunicación diseñado, capaz de comunicar estos elementos

con otros tipo de dispositivos dedicados a la gestión y supervisión, que normalmente están elaborados por computadores o sistemas de visualización tales como pantallas industriales. En este nivel es posible observar cómo se están llevando a cabo los procesos de planta y, a través de sistemas SCADA (Supervisión, Control y Adquisición de Datos), poseer una imagen virtual de la planta de manera que ésta se puede verificar de forma detallada, o a su vez mediante pantallas que son capaces de disponer de un "panel virtual" donde se detallen las posibles alarmas, fallos o alteraciones en cualquiera de los procesos que se realice [1].

2.6. Nivel de Campo y Proceso

Aquí se sitúan los elementos capaces de controlar los actuadores y sensores del nivel anterior, tales como autómatas programables o equipos de aplicación específica basados en microprocesador como robots, máquinas herramienta o controladores de motor. Estos elementos permiten que los actuadores y sensores funcionen de forma recíproca para ser capaces de controlar el proceso industrial deseado. Los elementos de este nivel de control, junto con los del nivel inferior de entrada/salida, poseen capacidad absoluta para llevar a cabo procesos productivos por su propia cuenta [1].

2.7. Nivel de gestión (Nivel de fábrica)

Este nivel estará principalmente conformado por computadoras, ya que se encuentra más apartado de los procesos productivos. Es más, en este nivel no es importante el estado y la supervisión de los procesos de planta, al contrario, sí gana importancia toda la información relativa a la producción y su gestión asociada, es decir, a través del nivel de control es posible obtener información general de todos los niveles inferiores de una o varias plantas [1].

2.8. Protocolo MODBUS

[2] MODBUS es un protocolo de comunicaciones industriales de capa 7 del modelo OSI, relacionado en la arquitectura maestro/esclavo, fue creada en 1979 por MODICON para un lote de PLC. Hoy en día es un protocolo estándar dentro de la industria que tiene mayor disponibilidad para el enlace de artefactos industriales.

Un proceso del tipo maestro/esclavo tiene un nodo maestro cuya función es de enviar los comandos explícitos a cada uno de los nodos esclavos los cuales procesaran la

respuesta solicitada. Una ventaja de este tipo de bus de campo es que los nodos no transmiten información sin una petición del nodo maestro y además no se enlazan con los demás nodos esclavos dentro de la red [3].

En el protocolo de comunicación MODBUS hay varias formas de intercambiar la información con los elementos de la red, se mencionan a continuación:

- Modbus RTU
- Modbus ASCII
- Modbus TCP

La selección del modo de comunicación RTU o ASCII solo se refiere a las redes estándar de MODBUS; esta selección asigna como será empaquetada la información dentro de las tramas de MODBUS y al mismo tiempo la decodificación [3].

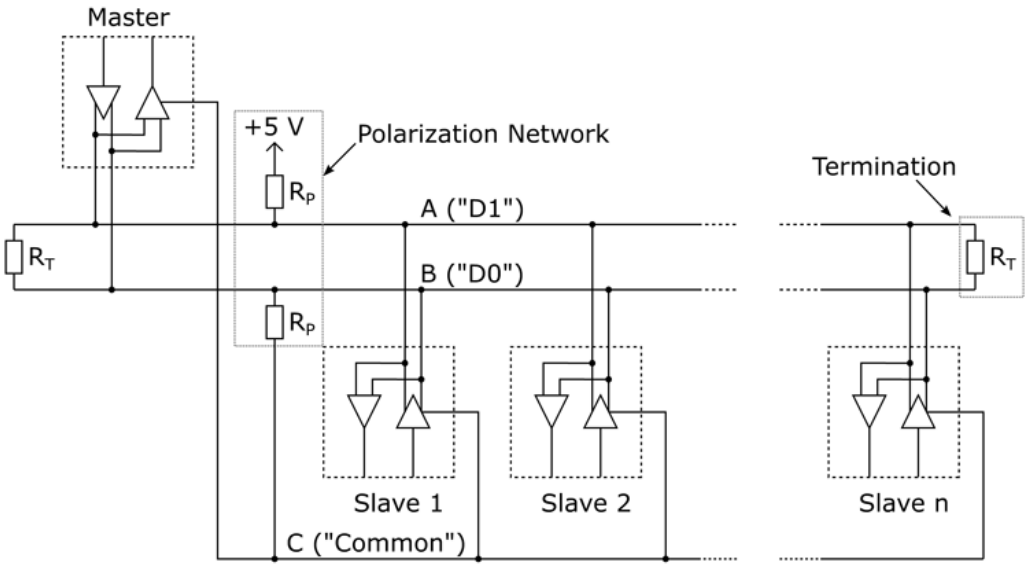


Figura 6: Configuración de MODBUS en RS485 [2].

2.9. MODBUS RTU

[4] MODBUS RTU (Remote Terminal Unit) es conocida y su principal función es que los bytes se envían en su codificación binaria plana, sin ningún tipo de conversión. Está inicialmente creado para comunicaciones en bus serie. Como ventaja general tiene excelente aprovechamiento del canal de comunicación, mejorando la velocidad de la transmisión de los datos. El inconveniente es que

necesita una gestión de tiempos entre bytes recibidos para saber cuándo empiezan y terminan las tramas.

Dentro de la trama MODBUS RTU, la delimitación de la misma se realiza por intervalos de tiempo de caracteres de silencio. Un carácter de silencio tiene la duración de un byte de datos enviado por el medio físico, pero no transporta datos, y su duración (T) depende de la velocidad (Vt) y del número bits que se utilicen para su codificación (N) entonces T=N/Vt. Según MODBUS, cuando se utilicen velocidades de hasta 19.200 bps, el tiempo entre tramas debe ser como mínimo 3,5 veces la duración de un carácter, y para velocidades superiores se recomienda un tiempo constante de 1,75ms. Por ejemplo, para una configuración del puerto serie de 19.200bps, con un bit de parada y un bit de paridad (11 bits en total, sumando el de inicio y 8 de datos) se tiene: 3,5*11/19.200= 2ms.

La trama MODBUS RTU incorpora un código Cyclical Redundancy Check (CRC) de 16 bits para comprobación errores, que tiene que ser determinado por el emisor a raiz de todos los bytes de la trama enviados antes del CRC, exceptuando los delimitadores. Para aquello se utiliza un algoritmo puntual, bien concreto en la especificación de MODBUS serie. El receptor debe volver a calcular el código de igual forma que el emisor, y comprobar que el dato obtenido del cálculo es igual al dato presente en la trama para poder corroborar los datos [3].

2.10. Transmisión en MODBUS/RTU

Cuando se designa la configuración de los controladores en modo RTU (terminal de unidad remota), cada 8 bits (1 byte) en un mensaje, contiene dos caracteres hexadecimales de 4 bits. La mayor característica de este protocolo es que a mayor densidad de caracteres otorga un mejor rendimiento que los datos en ASCII para la misma velocidad de transmisión (Véase Figura 7).

- Trama RTU:

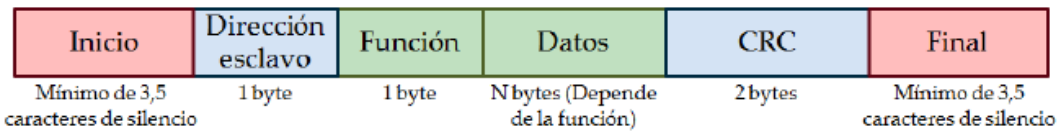


Figura 7: Trama MODBUS/RTU [3].

2.11. MODBUS ASCII

Los datos se codifican como caracteres ASCII entre el "0" (16#30) y el "9" (16#39) y entre "A" (16#41) y "F" (16#46). Por ejemplo, si se necesita enviar el byte de valor 16#FF, se debe de enviar la cadena "FF", por lo que realmente se enviarían dos bytes: 16#46 y 16#46. Además se usan 3 caracteres especiales. El carácter ":" (16#3A) se utiliza para marcar el inicio de la trama y el par de caracteres no imprimibles "CRLF" (16#0D, retorno de carro, y 16#0A, salto de línea) se usan como delimitador del fin de la trama. Este protocolo tiene dos grandes ventajas. Primero, ofrece una facilidad de detección del principio y del fin de trama gracias a los campos de inicio y fin (caracteres ":" y "CRLF"), con independencia de los tiempos de la transmisión del canal de comunicación. Segundo, logra trabajar con equipos de procesamiento lento sin tener que reducir la velocidad de comunicación siempre que tengan buffers de almacenamiento de los datos recibidos. Los problemas son que necesitan un mayor ancho de banda que MODBUS RTU para el envío de la misma petición o respuesta, o visto de otra forma, para el mismo ancho de banda, el envío de una trama con ASCII es más lento que con RTU [3].

- Trama ASCII:

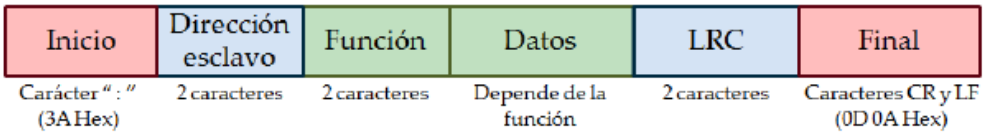


Figura 8: Trama MODBUS ASCII [3].

2.12. MODBUS TCP/IP

Es una evolución natural de las comunicaciones seriales como Modus RTU y Modbus ASCII. En Modbus TCP/IP como lo nombra la Modbus Organization toda la estructura de la trama se mantiene como lo que se ha creado hasta el momento, pero se realizan algunos ajustes propios de las nuevas capas del modelo OSI que se están incluyendo.

La configuración de los equipos son de tipo Cliente y Servidor. Se puede enlazar un paralelo para entender el comportamiento de los equipos con el tradicional Modbus RTU, los maestros son llamados clientes y los esclavos son los servidores [4].

En Modbus TCP en la capa física se estandariza con la norma EIA/TIA 568, que define entre otros aspectos el código de colores del cable, se utiliza un conector RJ45 que conexionan 8 señales, en donde 4 son dedicadas únicamente a la transmisión y recepción de datos (TX+, TX-, RX+, RX-).

Al encapsularse sobre TCP/IP viene en la trama un campo nuevo MBAP Header (MODBUS Aplication Protocolo header). El header tiene un tamaño de 7 bytes con el siguiente contenido:

[Transaction Identifier][Protocol Identifier][Length][Unit Identifier]

[2 Bytes][2 Bytes][2 Bytes][1 Byte]

Transaction Identifier: Identificación de una petición o respuesta de Modbus

Protocolo Indentifier: Es igual a 0 (cero) para el protocolo Modbus

Length: Número de Bytes

Unit Identifier: Identificación del esclavo (En caso de pasar a redes Modbus RTU)

Entonces la configuración inicial básica que debe ejecutar un usuario cuando utiliza Modbus TCP es asignar la dirección IP del equipo y adicionalmente la dirección del esclavo con el que se va a comunicar [4].

2.13. Descripción del protocolo MODBUS

[5] El protocolo MODBUS trabaja siempre en configuración maestro-esclavo (cliente - servidor), siendo el maestro (cliente) quien controla siempre las comunicaciones con los esclavos que pueden ser hasta 247 (@1d a @247d). Los esclavos (servidores) se limitan a devolver los datos solicitados o a ejecutar la acción dicha por el maestro. La comunicación del maestro hacia los esclavos puede ser de dos tipos:

- “peer to peer”: la comunicación es de tipo “maestro - esclavo”, el maestro pide información y el esclavo lo otorga.

-“broadcast”: la comunicación es de tipo “maestro - todos los esclavos”, el maestro manda un dato a todos los esclavos de la red sin esperar respuesta (se envía a la dirección @0d). Como se puede suponer, la secuencia básica en las comunicaciones MODBUS se basa siempre en una trama de pregunta, seguida de la trama de respuesta:

- Pregunta: con el código de operación que establece al esclavo que trabajo ha de ejecutar, y los bytes necesarios (datos y comprobación) para su desarrollo.
- Respuesta: con la recepción o datos resultantes del desarollo de la función.

Se pueden dar ciertos casos, en que hay más de una trama de respuesta para una trama de pregunta, como por ejemplo cuando el maestro comanda una función cuya respuesta toma tiempo para que el esclavo la procese. Por consiguiente el esclavo envía una primera respuesta detallando que aún no tiene los datos y tardará cierto tiempo en obtenerlos, y otra segunda con los datos o confirmación de la función.

También las comunicaciones MODBUS se pueden ejecutar en modo ASCII o en modo RTU. En modo ASCII los bytes se envían codificados en ASCII, es decir, que por cada byte a transmitir se envían dos caracteres ASCII (2 bytes) con su representación hexadecimal (puede leer las tramas con un simple codificador de texto). En modo RTU se envían en binario, tal cual. En el modo ASCII las tramas comienzan por 3AH (carácter ':'), y terminan en 0DH-0AH (CRLF Carrier Return Line Feed) y cada byte se envía como dos caracteres ASCII. En modo RTU no se utiliza indicador de inicio y final de trama [5].

Tabla 1: Comparación de tramas ASCII-RTU [5].

Parámetro	**Modo ASCII**	**Modo RTU**
Caracteres	ASCII ‘0’…’9’,’A’….’F’	Binario 0…255
Comprob. Error	LRC Longitudinal Redundancy check	CRC Cyclic Redundancy Check
Inicio de trama	Carácter ‘:’	3.5 veces t de carácter
Final de trama	Carácter CR/CL	3.5 veces t de carácter
Distancia max. entre caracteres	1 seg	1.5 veces t de carácter
Bit de inicio	1	1
Bits de datos	7	8
Paridad	Par / Impar / Ninguna	Par / Impar / Ninguna
Bits de parada	1 si hay paridad 2 si ninguna	1 si hay paridad 2 si ninguna

2.14. Trama del protocolo Modbus

La trama es una estructura definida en el protocolo en donde los dispositivos intercambian datos cumpliendo este formato. Dentro de la trama se definen los campos y el tamaño asignado para cada uno [4].

[ID][FUNCIÓN][DATO][CRC]

La estructura anterior muestra de forma general la trama del protocolo, los campos son:

ID: Dirección del esclavo, número enteros desde 1, hasta 255.

FUNCIÓN: Tipo de solicitud que se le hace al esclavo, codificada de manera numérica, normalmente números enteros que empiezan desde 1, 2, 3, 4, etc.

DATO: Se usa para enviar información suplementaria a la solicitud hecha al esclavo, o también para responder a la solicitud, por consiguiente el campo de dato tiene una estructura diferente cuando el mensaje lo envía el maestro o cuando es la respuesta de un esclavo [4].

2.14.1. Trama Maestro

La trama del maestro manifiesta que el campo dato está conformado por dos subcapas, la dirección y la longitud. En la dirección se porporciona al esclavo en qué dirección debe encontrar lo que se ha pedido por medio de la función y la longitud indicada, a partir de esa dirección cuantos elementos se deben tomar [4].

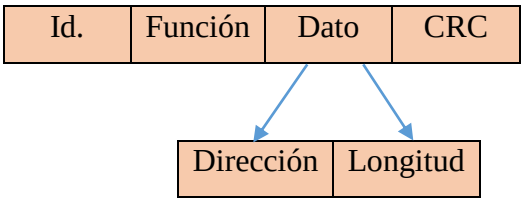


Figura 9: Estructura de la trama de un maestro Modbus [4].

2.14.2. Trama Esclavo

La estructura del esclavo se mantiene pero nuevamente la diferencia está en el campo de Dato, pues aparecen dos subcapas que son: número de bytes para dar la respuesta y la respuesta en sí.

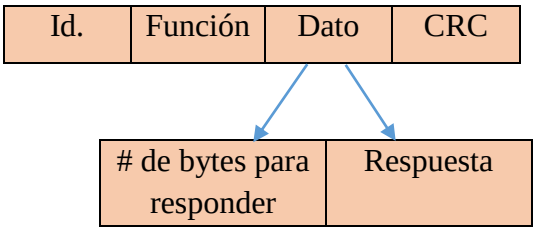


Figura 10: Estructura de la trama del esclavo [4].

2.14.3. Tamaño de la trama

En el tamaño de la trama depende de qué tipo de dato se va a enviar, los bits, bytes o words, para el caso del protocolo Modbus se organizan bloques de tamaño mínimo de bytes como lo muestra la siguiente estructura.

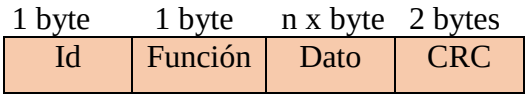


Figura 11: Tamaño de cada campo de la trama del protocolo Modbus [4].

El primer recuadro es la dirección del esclavo (slave), con un byte se pueden obtener valores entre 0 y 255 para poner la dirección (Id) del equipo. El segundo recuadro es la función y con un byte se tiene también la posibilidad de colocar valores entre 0 y 255 pero se debe tener en cuenta que no existen 256 funciones dentro del protocolo, son menos. El tercer cuadro es el campo de dato y es flexible dependiendo de que si es un maestro o un esclavo. Y el recuadro final es designado al CRC (Chequeo de Redundancia Cíclica) [4].

2.14.4. Funciones del protocolo

La forma de hacer solicitudes al esclavo se realiza por medio de la función la cual tiene asignado un byte, realmente las funciones del protocolo Modbus van desde el #1 hasta el #127 y se clasifican en tres grupos: funciones públicas, funciones definidas por el usuario y funciones reservadas. Las funciones más importantes y utilizadas son las que utilizan todos los dispositivos que tienen implementado el protocolo y son aquellos códigos que son públicos. Ahora se mostrará los códigos disponibles según la Modbus Organization [4].

				Function Codes			
				code	Sub code	(hex)	Section
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
			Write Multiple Coils	15		0F	6.11
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03	6.3
			Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access		Read File record	20		14	6.14
			Write File record	21		15	6.15
Diagnostics			Read Exception status	07		07	6.7
			Diagnostic	08	00-18,20	08	6.8
			Get Com event counter	11		OB	6.9
			Get Com Event Log	12		0C	6.10
			Report Server ID	17		11	6.13
			Read device Identification	43	14	2B	6.21
Other			Encapsulated Interface Transport	43	13,14	2B	6.19
			CANopen General Reference	43	13	2B	6.20

Figura 12: Funciones públicas disponibles en Modbus [4].

2.14.5. Funciones más comunes del protocolo Modbus

Las siguientes 8 funciones son muy utilizadas, pero no quiere decir que los fabricantes las implementan todas en un equipo, se puede dar el caso que de las mencionadas anteriormente sólo se desarrollen dos o tres:

01 ->Read Coils

02->Read Discrete Inputs

03->Read Holding Register

04->Read Input Register

05->Write Single Coil

06->Write Single Register

15->Write Multiple Coils

16->Write Multiple Registers

2.14.6. Read coils

Su función es leer bobinas, suena raro la verdad para aquellos que no conocen un PLC, pero es importante anotar que los equipos tienen diferentes tipos de salidas entre las que se encuentran los relés y de allí el nombre de bobinas ya que el elemento que se debe energizar para activar una salida es la bobina. Este nombre del código proviene de los primero PLC´s ya que este tipo de salida es usual e inició en los primeros Modicon. Su funcionamiento se basa en que el maestro solicita a un esclavo la función [01] le está pidiendo información del estado de sus salidas, indiferente si son tipo digital, tipo relé, etc. [4]

2.14.7. Read discrete inputs

El concepto se basa en leer entradas discretas, con este código el maestro solicita al esclavo información del estado de sus entradas, no importa el tipo de entrada que tenga. Un ejemplo de respuesta del esclavo puede ser “entrada 1 en alto, la entrada 3, la entrada 7, etc.” [4].

2.14.8. Read holding register

Se define para leer registros de retención, el maestro le indica al esclavo por medio de este código que entregue información de una posición de memoria, este tipo de memorias son de lectura y escritura. Es una de las funciones más importantes del protocolo Modbus pues por medio de estos registros se puede obtener información de todo lo que ocurre en los dispositivos, es solamente mover la información que se quiere obtener al registro de retención [4].

2.14.9. Read input register

Su tarea corresponde a leer registros de entrada, es decir que el maestro le da instrucciones al esclavo a través de este código, le entrega la información de una posición de memoria determinada, este tipo de memorias son de sólo lectura. Cuantiosos dispositivos los usan para intercambiar datos de sus entradas analógicas en caso de que las integren [4].

2.15. Norma física RS485

[6] La EIA definió un nuevo estándar: la RS485. Creada en el año 1983, es una versión optimizada de la RS422. Se establece como interface multipunto y permite la comunicación de hasta 32 dispositivos emisores/receptores en un bus de datos único, ya que mantiene un tercer estado (habilitación) para que no existan interferencias en el canal de comunicación.

La norma RS485 se basa en un sistema diferencial que permite suprimir los posibles ruidos que se puedan incorporar en el canal de comunicación. A efectos eléctricos, son absolutamente compatibles con la norma RS422. Los niveles lógicos en mención a los eléctricos son:

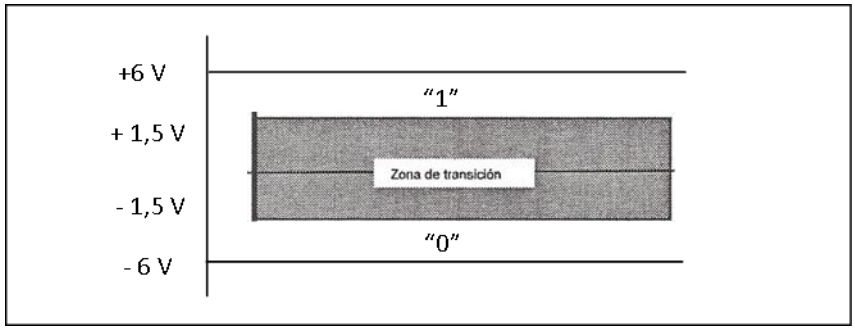


Figura 13: Señales eléctricas por la interface RS-485 [6].

Las señales que incorpora esta norma son las siguientes:

Tabla 2: Señales del estándar [6].

Señal	Definición
A o D+	Señal de emisión/recepción no invertida transmitida al canal de comunicaciones
B o D-	Señal de emisión/recepción invertida transmitida al canal de comunicaciones
FG	Masa de protección

En el estándar RS-485 se usa el tipo de transmisión diferencial Half-Duplex que se caracteriza por:

- Usar dos hilos o cables, referenciado a masa.
- El dato se obtiene de la diferencia de la señal eléctrica entre los dos hilos que conforman el canal de comunicación.

- La condición de que el tipo de transmisión sea Half-Duplex indica que cada equipo puede enviar y recibir, pero no al mismo tiempo.
- El estándar RS485 proyecta velocidades que llegan a 10 Mbps y distancias que pueden alcanzar los 1.200 metros.
- El estándar RS485 otorga un enlace en red de hasta 32 dispositivos.

La norma RS485 contiene un tercer estado que permite que un dispositivo se pueda quedar en estado de alta impedancia, y de por sí no genere lectura, es como si se encontrara desconectado de la red.

Generalmente la habilitación se encuentra en estado de recepción "0". Si se quiere transmitir, se debe poner un "1" en la señal de habilitación correspondiente [6].

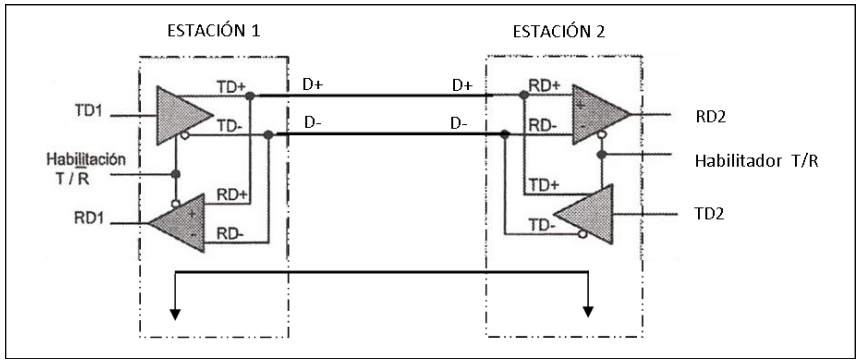


Figura 14: Transmisión diferencial aplicada a la norma RS-485 [6].

Existen varias aplicaciones dentro de las comunicaciones industriales en las que la norma RS485 es la utilizada para transmisión de datos, como es el caso de PROFIBUS.

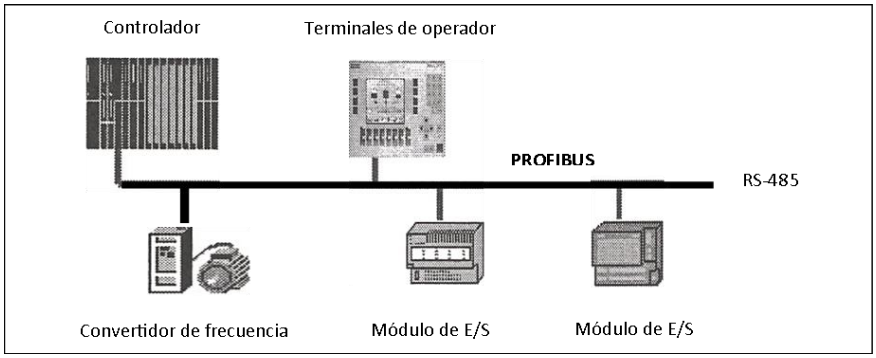


Figura 15: Ejemplo de una red Profibus con diferentes componentes [6].

Las principales ventajas de la norma es que la longitud máxima del conductor es de aproximadamente 1.200 metros con velocidad de transmision de hasta 90 kbps, pues su maxima velocidad de enlace alcanza los 10 Mbps. Comparando con otras redes de comunicaciones, la velocidad y longitud del enlace están inversamente ligadas: si se requiere hayar la máxima velocidad, el cable deberá ser más corto y viceversa.

El estándar establece que el número máximo de dispositivos será de 32, pero con receptores de alta impedancia se pueden llegar a los 256 equipos. Los adaptadores RS485 utilizan una fuente de alimentación de 5 voltios para sus circuitos.

En la figura 16 se puede evidenciar que cada dispositivo del enlace dispone de un adaptador con las líneas TD/RD y habilitación (T - /R) para manipular el modo de funcionamiento del dispositivo:

- Cuando la entrada tiene un "0", el equipo se encuentra en modo recepción y puede leer el tráfico en la red.
- Cuando la entrada de control se pone en "1", el dispositivo se pondrá en modo transmisión y es cuando éste está a disposición de poder enviar datos a la red.

En una configuración en red trabajando bajo el estándar RS485 cualquier dispositivo puede comunicarse con el resto, lo que representa una ventaja con respecto a la RS422, pero de otra manera complica el control de acceso a la red [6].

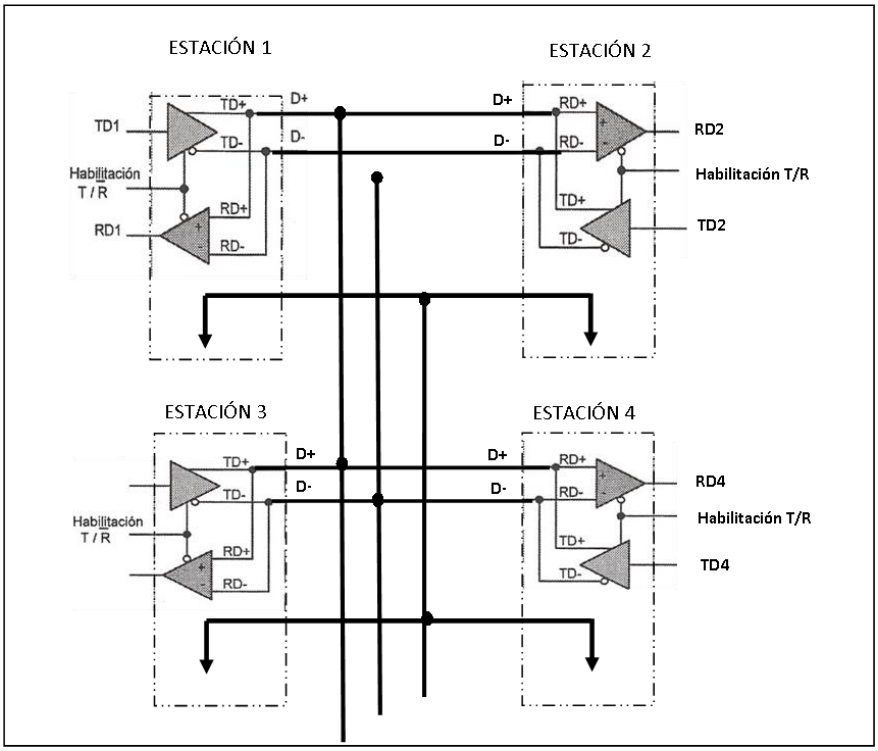


Figura 16: Configuración multifilar de una red en la norma RS-485 [6].

Al igual que ocurria en la norma RS422, y como también pasa en la mayoría de redes, se deberán colocar resistencias terminadoras (Rt) en los extremos del canal de comunicaciones para obtener en todo momento la impedancia en la línea y que es del valor de 120 ohmios. [6]

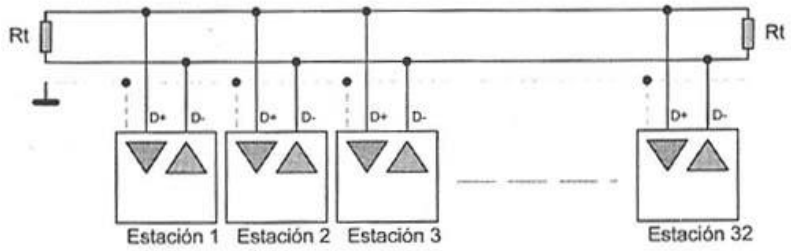


Figura 17: Circuitos que integran la norma RS-485 [6].

A continuación se presenta una tabla, de las diferentes normas físicas:

Tabla 3: Características de estándares de comunicación [6].

Parámetros		RS-232C	RS-422-A	RS-485
Modo de trabajo		Unipolar	Diferencial	Diferencial
Número de emisiones y receptores		1emisor 1 receptor	1 emisor 32 receptores	32 emisores 32 receptores
Longitud máxima del cable		15 metros	1.200 metros	1.200 metros
Velocidad de transmisión máxima		20 kbps	Hasta 10 mbps	Hasta 10 mbps
Número de líneas		Hasta 25 (datos y control)	Hasta 4 (datos y control por software)	2 (datos y control por software)
Tipo de cable		Cable especifico(hilos tantos como señales)	Par trenzado (2 pares)	Par trenzado (1 par)
Topología que admiten		Punto a punto	Multipunto (punto a punto desde principal a secundarias) Anillo	Punto a punto Multipunto Anillo Bus
Simultaneidad en la transmisión		Simplex Half Duplex Full Duplex	Full Duplex	Half Duplex
Tensión de salida del emisor	Sin cargar Cargado	+/- 5 V +/- 15 V	+/- 2 V +/- 6 V	+/- 1,5 V +/- 6 V

2.16. Medio Físico

[7] El medio físico para establecer un enlace consiste en un bus semi dúplex (half dúplex) (RS485 o fibra óptica) o full dúplex (RS422, BC 0- 20 mA o fibra óptica).

El tipo de comunicación es asíncrona y en cuanto a la velocidad de transmisión va desde los 75 baudios hasta 19.200 baudios. La distancia entre estaciones cambia dependiendo del medio físico, pudiendo llegar hasta los 1200 metros sin necesidad de repetidores.

[4] El estándar RS485 define un bus para la transmisión serie multipunto, donde en un instante puede existir un dispositivo transmitiendo y varios recibiendo. La comunicación es semi dúplex, de manera que un dispositivo puede enviar y recibir, pero no a la misma vez. El cableado y topología consiste en un par de hilos trenzados de cobre donde se transmite una señal diferencial para generar los bits de datos, donde es inmune a las interferencias y soporta largas extensiones de terreno. Adicional del par trenzado para datos, puede implementarse líneas de 0V y 5V para alimentar dispositivos del bus. Los bits se envían por medio de una trama asíncrona.

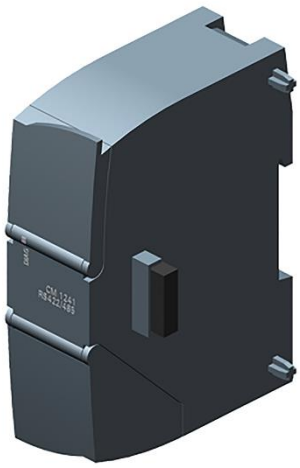


Figura 18: Módulo CM1241 RS485 [7].

2.17. ¿Cómo se conectan las interfaces RS485/RS422 de los módulos SIMATIC y SIPLUS para la comunicación serie?

[8] En los siguientes módulos se puede implementar la comunicación serie utilizando interfaces RS485 y RS422. La disponibilidad del conexionado está libre para los gráficos habientes.

Si se utiliza un cable de conexión RS485 o RS422 de más de 50 metros, se debe parar el segmento de bus en ambos extremos con un resistor de cierre de al menos 330Ω. Excepciones para este caso son los módulos CM 1241 y CB 1241 que tienen otras resistencias de cierre diferentes. Sólo se cierran los 2 extremos de la red RS485. Los equipos situados entre ambos equipos finales no se cierran.

La figura 19 muestra la disposición de la conexión para tener una comunicación RS485. En el módulo CM 1241 se pueden conectar conectores PROFIBUS de 9 pines de Siemens.
Para tener una configuración propia del conector hay que considerar los siguientes puntos:

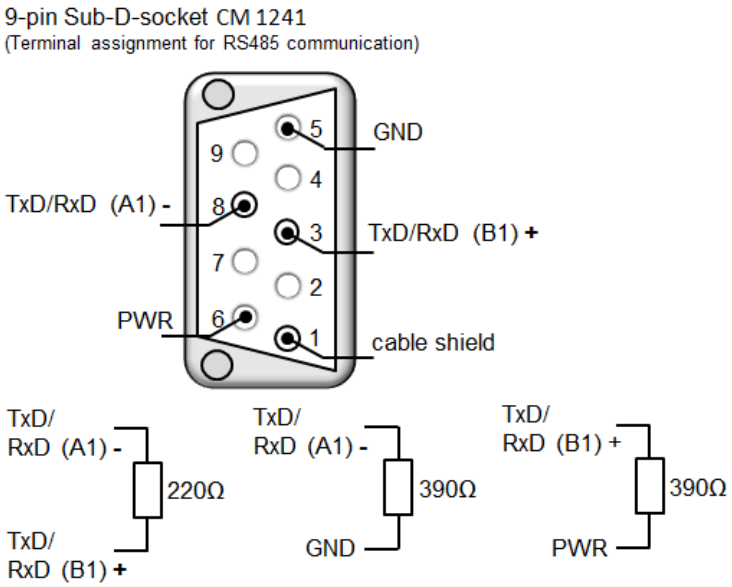


Figura 19: Configuración de pines para puerto DB9 en RS485 [8].

2.18. Motor eléctrico trifásico

[9] Se trata de un componente eléctrico que transforma energía eléctrica en energía mecánica a base de pulsos electromagnéticos. Varios motores eléctricos son reversibles, pueden transformar energía mecánica en eléctrica.

La energía eléctrica trifásica genera campos magnéticos rotativos en el bobinado primario correspondiente al estator, lo que hace que con el arrancar de aquellos motores no requiera circuito secundario, son más reducidos y simplificados que un

monofásico de inducción con la misma potencia, ya que por su elaboración es de menor valor económico.

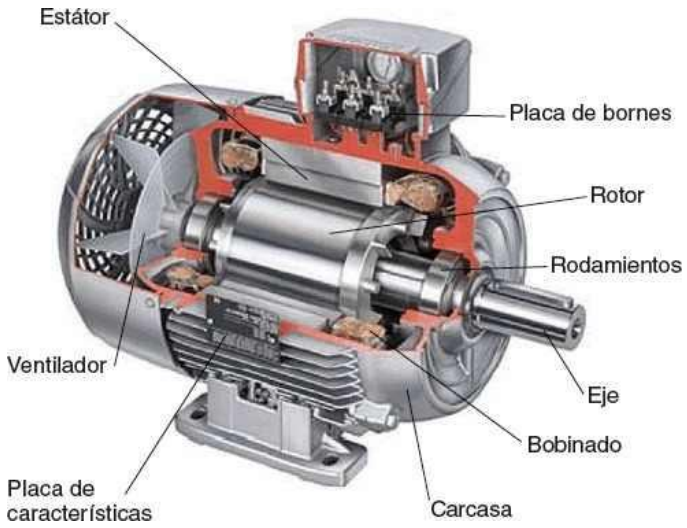


Figura 20: Partes del motor trifásico [9].

2.18.1. Tipos de motores trifásicos

[9] Cuando el rotor tiene similar velocidad de giro que la del campo magnético rotativo, se establece que el motor es síncrono. Pero por el contrario, el rotor tiene una velocidad de giro superior o inferior que aquel campo magnético rotativo, el motor se define como asíncrono de inducción.

Los motores eléctricos trifásicos están clasificados por dos grupos:

1. Motores Síncronos
2. Motores Asíncronos

2.19. Variador de Frecuencia

[10] El variador de frecuencia modula la velocidad de los motores eléctricos para que la electricidad que llega al motor se ajuste a la demanda real de del sistema, disminuyendo el consumo energético del motor entre un 20 y un 70%.

Los variadores de frecuencia como concepto general es un modulador industrial que se manifiesta entre la alimentación energética y el motor. La energía de la red pasa por el variador y la regula antes de que ésta ingrese al motor para posteriormente ajustar la frecuencia y la tensión en función de los requerimientos del sistema.

Figura 21: Variador de frecuencia ABB ACS355 [10].

Los variadores aminoran la potencia de salida de un elemento, como una **bomba o un ventilador**, por medio del control de la velocidad del motor, asegurando que no funcione a una velocidad mayor a la necesitada [10].

2.19.1. Módulo adaptador Modbus RTU FMBA-01

El FMBA-01 serie F es un módulo adaptador Modbus RTU que conecta el convertidor a una red Modbus-RTU RS-485. Los códigos de función de lectura/escritura común y de registrador múltiple son compatibles.

Figura 22: Módulo FMBA-01 [10].

3. MARCO METODOLÓGICO

3.1. Diseño del módulo de prácticas.

Los módulos de prácticas cuentan con controladores lógicos programables de marca Siemens, cuentan con módulos de comunicación y conmutadores de redes y panel HMI de la misma marca. Adicionalmente los módulos están equipados con luces pilotos, potenciómetros de precisión, selectores, relés, pulsadores y fuentes de poder. Los puertos de comunicación que incorporan son RJ45, DB25, DB9.

Figura 23: Estructura física

3.2. Desarrollo de la estructura del tablero.

La estructura del módulo está conformada de un chasis metálico y resistente, en forma de una caja triangular la cual incorpora un compuerta que abre hacia arriba, y en ella se encuentran incrustados distintos elementos como luces pilotos, pulsadores, interruptores, pantalla HMI, potenciómetros y borneras.

Al interior del módulo encontraremos el PLC, relés, switches, módulo de comunicación, bornes, breaker, fuente de poder. A continuación se muestran los planos con las respectivas medidas a escala.

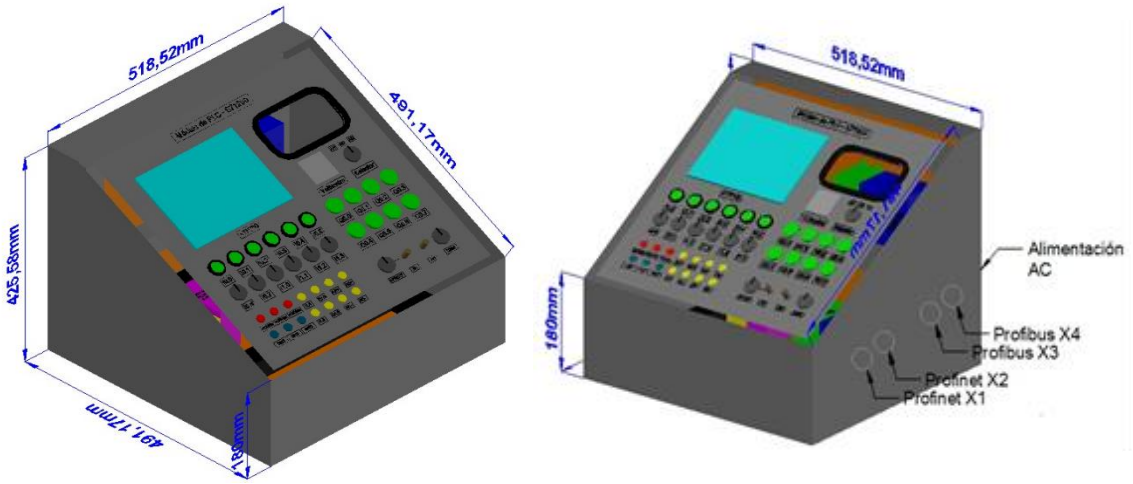


Figura 24: Diseño CAD de los módulos didácticos.

Luego del diseño de los planos procedemos a la elaboración de las estructuras metálicas, con la ayuda del proveedor se realiza el ajuste mecánico de las estructuras, que comprende cortes, perforaciones, limado, lijado y pintado de las mismas. Así mismo se instaló las rieles din donde irán montados los elementos de control.

Figura 25: Estructura metálica.

Ahora se procede con el ensamblaje de las partes y elementos que conforman los módulos didácticos, entre ellos están los exteriores como las borneras, luces piloto, potenciómetros, puertos de comunicación, terminales de poder, voltímetros, botones e interruptores. Al interior de los módulos irán los PLC`s, switches, relés, puentes, breakers, y fuente de voltaje.

Figura 26: Montaje de elementos del tablero.

3.3. Diagrama de conexiones

Los módulos didácticos cuentan con alimentación 120 VAC lo cual es suministrado por un interruptor que conduce hasta la fuente de voltaje PM 1207, a partir de allí la fuente entrega 24 VDC hacia todos sus componentes como relés, luces piloto, botones, interruptores y HMI. Los potenciómetros y voltímetros son alimentados por 10 VDC lo cual obtenemos con una tarjeta electrónica de conversión 24-10 VDC. A continuación presentamos los diagramas de conexión.

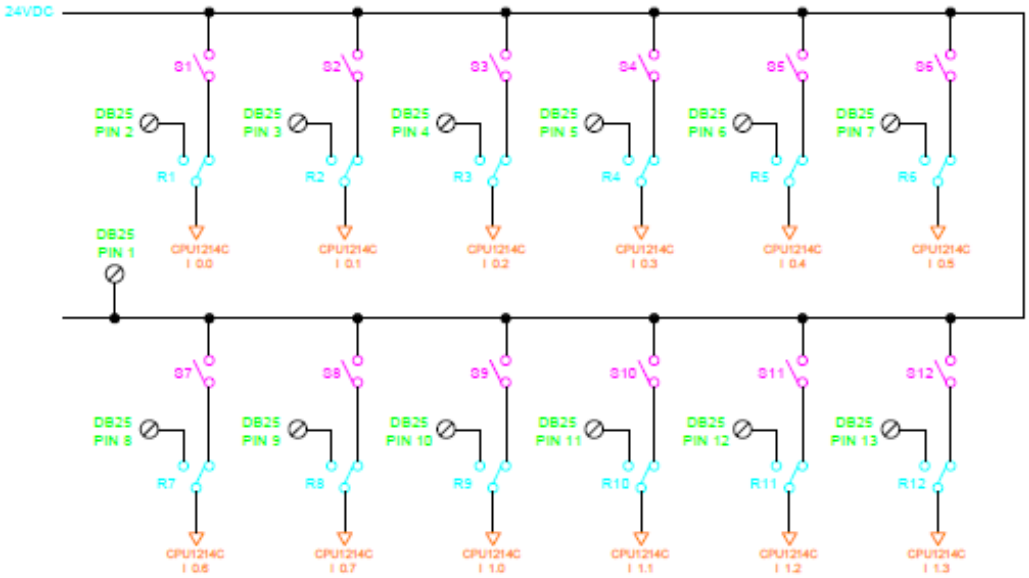


Figura 27: Entradas digitales [11].

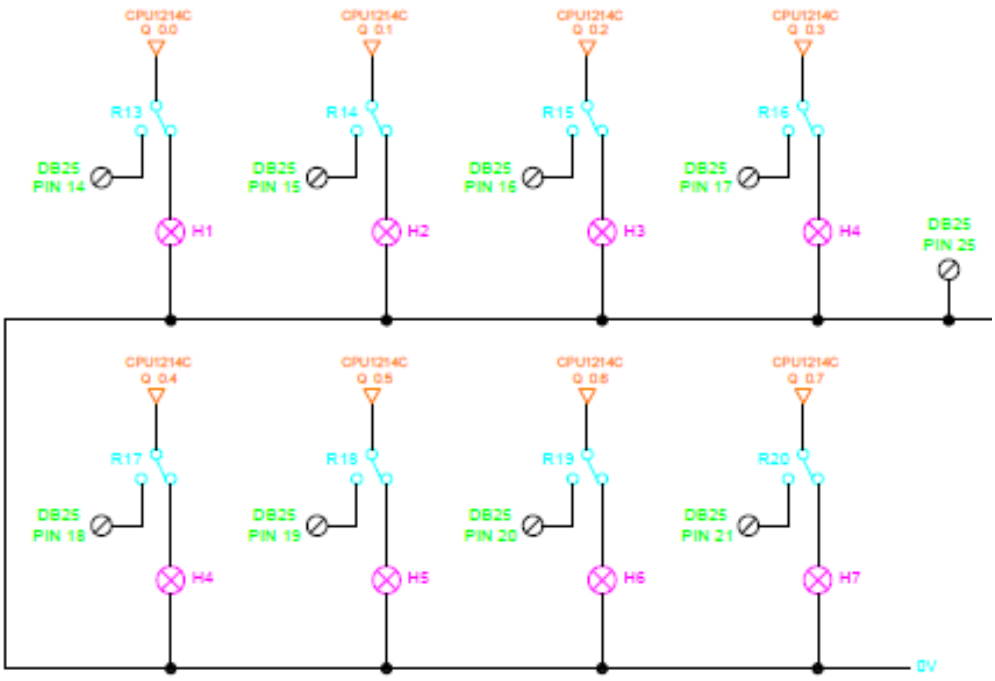


Figura 28: Salidas digitales [11].

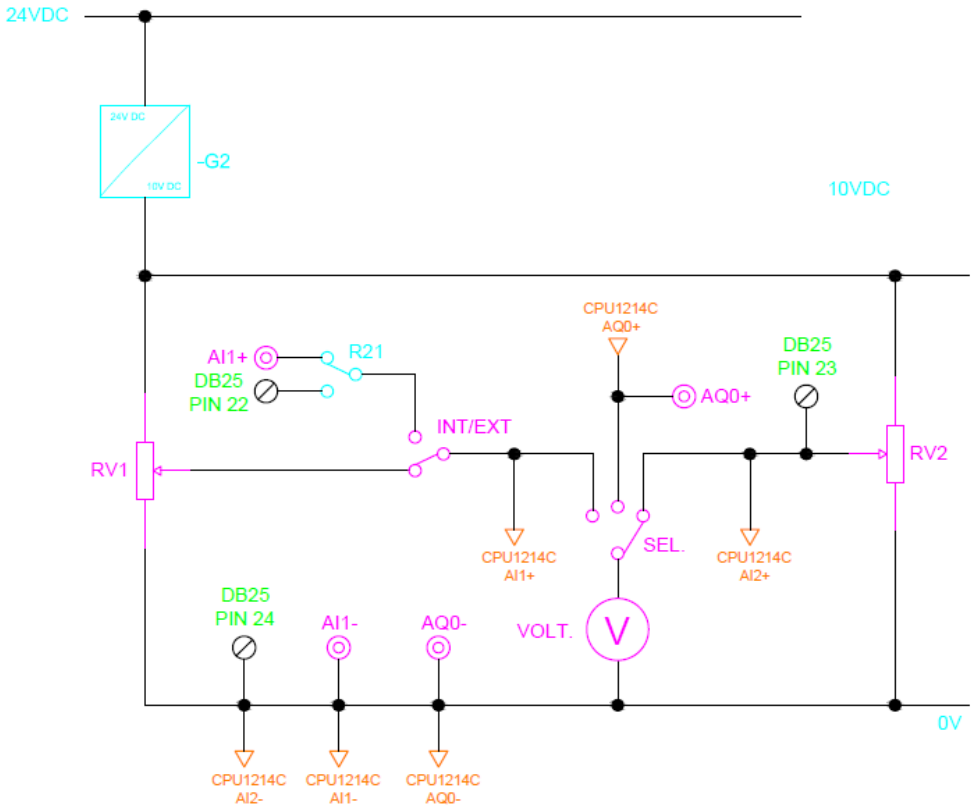


Figura 29: Entradas y salidas analógicas [11].

3.4. Cableado de los elementos del módulo

Luego del montaje de los elementos a la estructura se realizó la conexión de los elementos del módulo por medio de cable flexible Nº 18, para esto se siguió los planos de acuerdo al diseño eléctrico de los mismos. Una vez realizadas las conexiones se mejoró el ruteado de forma ordenada.

Figura 30: Cableado de elementos del módulo.

Ahora se muestra el proceso de conexionado de los relés con las luces piloto e interruptores, para esto se instalaron las rieles din y canaletas de 10 x 15 mm. Además se utilizó tres colores de cables diferentes para su identificación.

Figura 31: Conexionado entre elementos internos del tablero.

Finalmente conectamos el PLC con sus diferentes entradas y salidas, pantalla HMI, switches de comunicación Ethernet, y fuente de poder.

Figura 32: Conexionado de PLC.

Una vez hechas las conexiones procedemos a colocar las marquillas en los cables para su correcta identificación y seguimiento en las líneas.

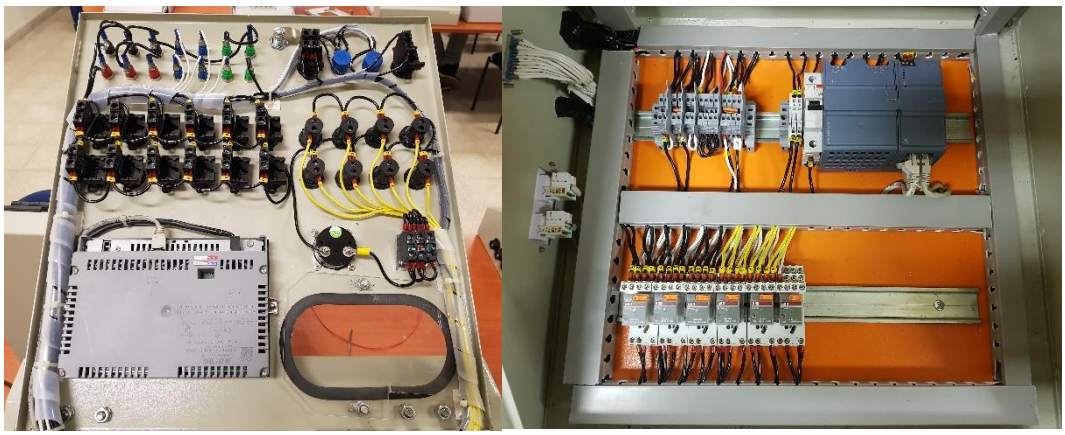


Figura 33: Marquillado de los elementos del tablero.

3.5. Acabado y detalles de los módulos

Para terminar el proceso de ensamblaje procedemos a colocar amarras plásticas en el cableado, también topes de fijación, espirales aislantes y las plaquillas para la identificación de los elementos externos de los módulos de automatización.

Figura 34: Estética de los módulos terminados.

3.6. Fabricación de cable macho – macho RS485

Para la comunicación entre plantas y dispositivos es necesaria la utilización de un cable de tipo macho – macho, para esto se utilizó cable para audio, dos puertos machos DB9. Se debe soldar el pin 3 (Tx+) de un extremo con el pin 3 del otro extremo, de igual forma se debe soldar el pin 8 (Rx-) de un extremo con el pin 8 consecutivo. La puesta a tierra debe estar soldada al chasis del puerto.

Figura 35: Fabricación de cable macho – macho RS485.

3.7. Adaptación de puerto RS485 a planta de control de nivel

Como los datos se van a enviar vía comunicación Modbus RTU es necesario agregar un puerto de comunicación RS485, para esto se perforó la estructura de la planta de control de nivel para enlazar con el CM1241 del PLC Planta. Se agregó un cable hembra macho para finalizar el enlace.

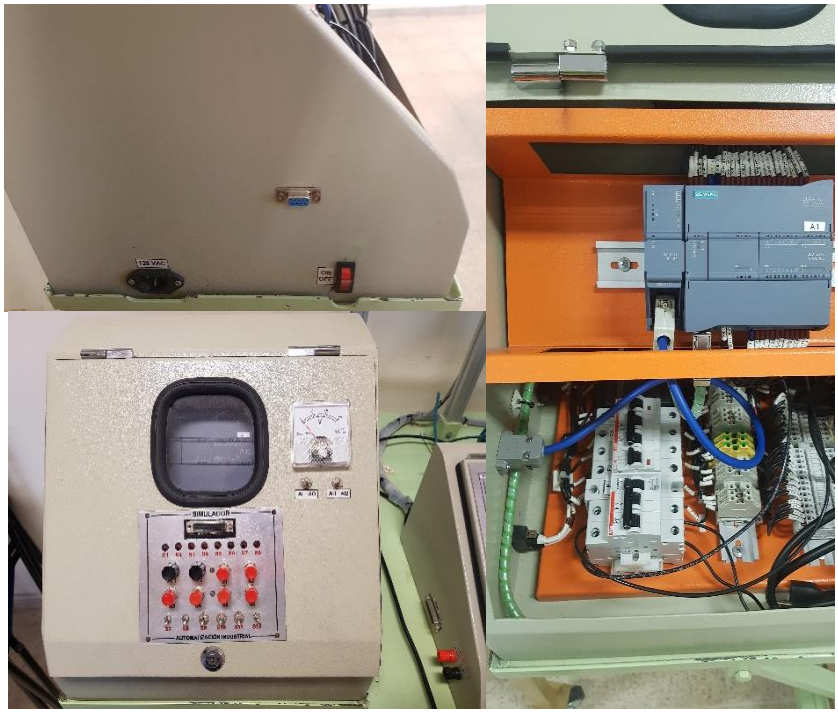


Figura 36: Instalación de puerto RS485 para planta de control de nivel.

3.8. Adaptación de puerto RS485 a módulo variador de frecuencia

Para comunicar el motor trifásico con el PLC se debe agregar una interfaz Modbus RTU a través de la tarjeta FMBA-01 del variador de frecuencia. Para aquello es necesario perforar la estructura metálica del módulo del variador para agregar un puerto RS485 hembra, el cual será ruteado hasta la tarjeta FMBA-01 por medio de dos terminales. El pin 3 (Tx+) se conecta en el pin 2 de la tarjeta FMBA-01, por último el pin 8 (Rx-) se conecta en el pin 3 de la tarjeta FMBA-01.

Figura 37: Instalación de puerto RS485 para módulo de variador.

3.9. Siemens S7-1200

Simatic S7-1200 otorga una extensa gama de bondades técnicas entre las cuales sobresalen las siguientes:

- Superior capacidad de procesamiento. Cálculo de 64 bits.
- Interfaz Ethernet / PROFINET integrado.
- Entradas analógicas integradas.
- Bloques de función para control de ejes conforme a PLC open.

Programación mediante la herramienta de software STEP 7 Basic v12 para la configuración y programación no sólo del S7-1200, también los paneles de la gama Simatic Basic Panels.

El modelo de CPU a utilizar en el presente proyecto es el 1214C AC/DC/Rly (6ES7214-1BE30-0XB0), que cuenta una memoria de trabajo de 50 KB, la cual utiliza una alimentación de corriente alterna que va desde 120V a 240V, con una frecuencia de 47Hz a 63Hz.

Este controlador cuenta con una capacidad de 14 entradas digitales a 24 VDC, de las cuales seis entradas digitales son empleadas para funciones tecnológicas (Contadores de Alta Velocidad HSC), posee 10 salidas digitales tipo relé, con carga resistiva máxima hasta 2A, tiene 2 entradas analógicas de 0-10 VDC.

En la Figura 16 y en la Tabla 1 se especifica las partes notables del controlador lógico programable S7-1200 [12].

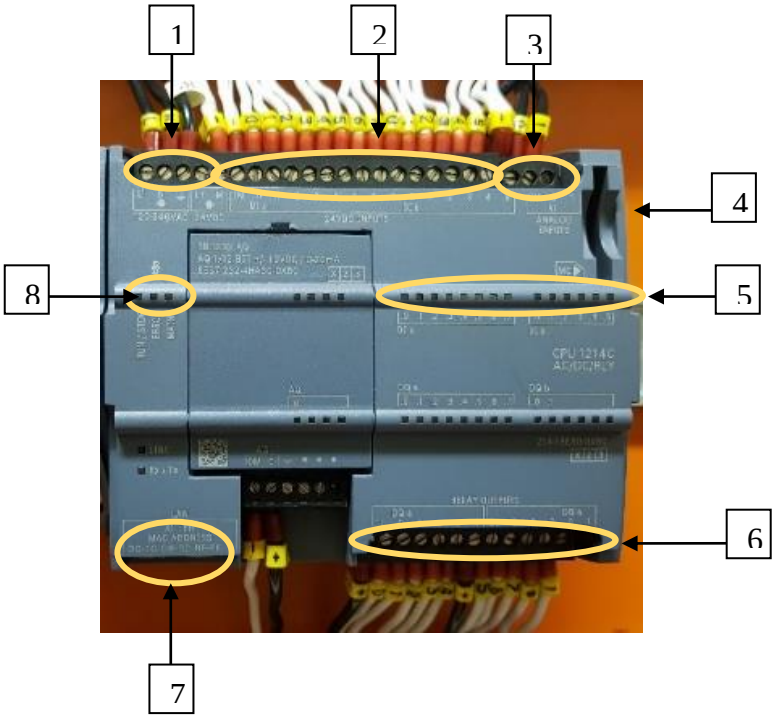


Figura 38: Estructura metálica

Tabla 4: Partes de un PLC S7-1200.

N°	Componentes S7-1200
1	Entradas de alimentación AC.
2	Entradas digitales.
3	Entradas analógicas.
4	Ranura de memoria externa.
5	LEDs de estado para E/S integradas
6	Salidas digitales.
7	Puerto Ethernet.
8	LEDs de run/stop del PLC.

3.10. Signal Board de salidas analógicas

El signal board SB 1232 AQ (6ES7232-4HA30-0XB0) de salidas analógicas, es un módulo de expansión de autómata programable para Siemens S7-1200, la cual se conecta directamente a la CPU.

Este módulo contiene 1 salida analógica, con rangos de salida de tensión de -10V a +10V y rangos de salida intensidad de 0mA a 20mA.

A continuación en la Figura 17 se puede apreciar el modulo [12].

Figura 39: Signal board.

3.11. Switch CSM 1277.

Para la comunicación Ethernet entre los diferentes dispositivos se utilizará el Switch CSM 1277, el cual cuenta con cuatro conectores hembra RJ45.

Figura 40: Switch CSM 1277.

3.12. Fuente de alimentación PM 1207

La fuente de alimentación PM 1207 es un elemento cuya función es de entregar al módulo la tensión nominal de 24 VDC a una corriente de 5A para un correcto funcionamiento.

La alimentación corresponde a una tensión nominal monofásica de 120 VAC o bifásica de 230 VAC a 50/60 Hz.

Figura 41: Fuente de alimentación PM 1207.

3.13. Relés

Los relés que se utilizarán para este proyecto son de tipo electromagnético de la marca ABB. Utiliza una configuración de contactos 4PDT, su tensión nominal de alimentación es monofásica 12-240 VAC a 50/60 Hz.

Figura 42: Relés tipo electromagnético.

3.14. Módulo de comunicación CM1241

Los Módulos de Comunicación RS485 y RS232 son capaces de funcionar con conexión punto a punto en serie, relacionadas en caracteres. Esta comunicación se configura y programa con bloques de instrucciones, también con las funciones de librerías para protocolo maestro y esclavo USS Drive y Modbus RTU, que están dentro del sistema de ingeniería SIMATIC STEP 7 Basic.

Figura 43: Modulo de comunicación CM1241.

3.15. KTP600 Basic PN

El KTP600 Basic PN es una pantalla de 6¨ TFT, la cual utiliza una interfaz Ethernet RJ45 10/100 Mbit/s, programmable en WinCC Flexible 2008 sp2 compact/ WinCC Basic / Step7 Basic, contiene SW Open Source.

Su voltaje de alimentación es 24 VDC a 50/60 Hz, utiliza una configuración de contactos 4PDT, su tensión nominal de alimentación es monofásica 120 VAC a 50/60 Hz.

Figura 44: Parte frontal del KTP600 Basic PN.

Figura 45: Parte posterior del KTP600 Basic PN.

Tabla 5: Partes del KTP600 Basic PN.

N°	Partes del KTP600 Basic PN
1	Pantalla táctil.
2	Botones programables.
3	Terminal de alimentación.
4	Terminal de comunicación Profinet.

3.16. Pulsantes

Los módulos cuentan con 6 pulsantes on/off NA para cada uno, los cuales están interconectados para ejecutarse como entradas hábiles del PLC S7-1200.

Figura 46: Pulsante.

3.17. Luces piloto

El proyecto cuenta con 8 luces piloto rojas, las cuales actuarán como salidas digitales del PLC. Estas luces cuentan con una tensión de entrada de 12 VDC.

Figura 47: Luz piloto.

3.18. Selectores

Los módulos cuentan con 6 selectores de dos posiciones los cuales se encargan de actuar como entradas digitales del PLC S7-1200. Adicionalmente contienen un selector de dos posiciones el cual servirá para habilitar el puerto de comunicación DB25. También los módulos tendrán un selector para habilitar una entrada analógica. Finalmente los módulos incluyen un selector de tres posiciones los mismos que proporcionaran el cambio de lecturas entre el voltímetro, entradas y salidas analógicas.

Figura 48: Selector.

3.19. Voltímetro

Cada módulo de automatización cuenta con un voltímetro analógico el cual tendrá una lectura de voltaje de entrada desde 0 – 24 VDC.

Figura 49: Voltímetro.

3.20. Potenciómetros

Ambos módulos de automatización cuentan con dos potenciómetros de precisión 10 K 10 vueltas, los mismos que servirán de entradas analógicas al PLC.

Figura 50: Potenciómetro de precisión.

3.21. Borneras tipo banana (hembra)

Los módulos de automatización contarán con 14 borneras las cuales están seccionadas y distribuidas de la siguiente manera:

- 3 borneras (rojo) asignado a 24 VDC.
- 3 borneras (negro) asignado a tierra.
- 2 borneras (azul) asignado a entrada digital.
- 2 borneras (azul) asignado a salida digital.
- 1 bornera (verde) asignado a entrada analógica positivo.
- 1 bornera (verde) asignado a entrada analógica negativo.
- 1 bornera (verde) asignado a salida analógica positivo.
- 1 bornera (verde) asignado a salida analógica negativo.

Figura 51: Borneras banana hembra.

3.22. Cable Convertidor USB – RS485

Para enlazar la comunicación vía puerto serie desde la PC hasta el PLC se requiere de un hardware el cual sea capaz de codificar los datos y enviar – recibir en ocasiones, para esto el convertidor USB a RS485 de Manhattan transforma cualquier puerto USB full-duplex estándar en un canal de comunicación dúplex y semi-dúplex con un alcance de hasta 1200 m (3900 pies). Detectado e instalado automáticamente como un puerto Com nativo sin configuración IRQ o DMA, proporciona conectividad USB con Plug and Play y capacidades de intercambio en caliente para dispositivos heredados y los conecta a PC más nuevas con menos conflictos de dispositivos. Su compatibilidad con Windows / 7/8, su potencia de bus y su robusta construcción ayudan a proporcionar un rendimiento confiable sin una fuente de alimentación externa o configuraciones complicadas [13].

Figura 52: Cable convertidor USB a RS485.

3.23. Características del cable convertidor USB a RS485

- Agrega rápidamente una conexión RS485 a un solo puerto USB.
- Convierte fácilmente buses industriales RS485 en una interfaz USB.
- Admite múltiples módulos cuando se instala en un concentrador USB.
- Configuración sin complicaciones de redes dúplex a dos hilos de hasta 1200 metros.
- Compatible con Windows / 7/8: plug and play; intercambiables en caliente.

3.24. Especificaciones del cable convertidor USB a RS485

Estándares y certificaciones:

- CE
- FCC
- USB 1.0
- USB 1.1
- Conexiones USB 2.0
- Puerto USB; tipo A macho
- D Sub; Macho de 9 pines
- D Sub; Hembra de 9 pines (en el adaptador)
- Puerto RS485; Hembra de 4 pines (en el adaptador)
- Contenedor de recepción de 384 bytes
- Control de transmisión automática para 128 bytes
- Velocidad de transmisión: 300 bps a 1 Mbps
- Alcance de red máximo: 1200 m
- Bus accionado
- Protección contra sobrecorriente: 600 W entre el puerto USB y RS485
- Chipset FT232RL
- Longitud del cable: 80 cm (31.5 in)
- Peso: 59.5 g (2.1 oz.)

3.25. Fuente de voltaje variable

[14] La fuente de voltaje variable con LM317T se caracteriza por entregar en la salida un voltaje variable entre 1.5 V a 15 voltios, con la capacidad de entregar una corriente continua de hasta de 1.5 amperios.

Utilizando el transistor LM317 se obtiene 500 mA a la salida, rango suficiente para trabajo en la mayoría de circuitos, pero en este caso utilizamos el LM317T ya que otorga más corriente.

El elemento tiene protección contra sobre corrientes y evita el integrado se recaliente provocado por un corto circuito. La tensión de salida depende de la posición que tenga la perilla del potenciómetro de 5 KΩ (kilohmios), perilla que se conecta al pin de ajuste del integrado. (COM).

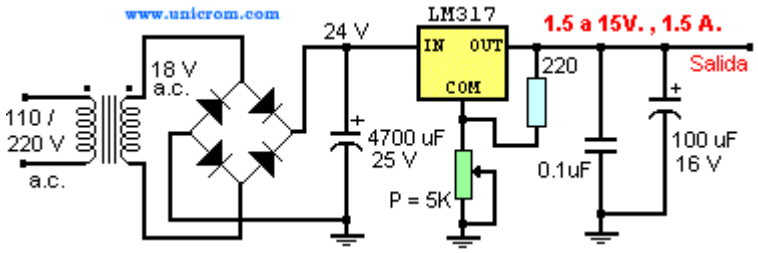


Figura 53: Diagrama de conexión de fuente regulada [14].

Esta fuente fue implementada con el fin de obtener 10 VDC para suministrar al voltímetro analógico y potenciómetros, la cual fue diseñada en una tarjeta perforada de 10*4 cm. y colocada al interior de las canaletas recubierto por una pequeña caja, como en la figura 40 se detalla:

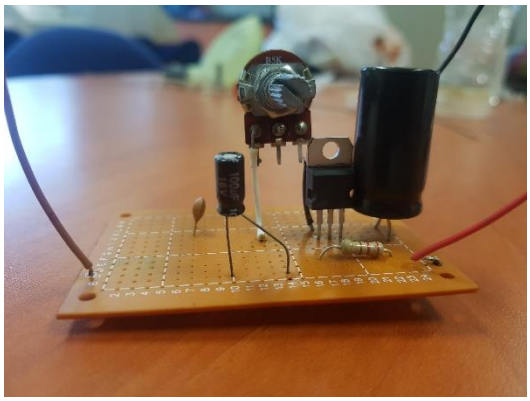


Figura 54: Fuente regulada a 10 VDC.

3.26. TIA Portal V12

El software optimiza todos sus sistemas de procesamiento, operación de máquinas y diseño. Posee una intuitiva interfaz de usuario, el fácil acceso de sus funciones y la completa transferencia de datos permite un fácil uso y manipulación. [15]

TIA Portal es el sofisticado sistema de ingeniería que permite elaborar de forma intuitiva y eficiente todos los procesos de planificación y desarrollo. El sistema convence por su funcionalidad probada y por ofrecer un entorno de ingeniería unificado para todas las tareas de control, visualización y accionamiento.

El TIA Portal conlleva varias actualizaciones de Software de Ingeniería SIMATIC STEP 7, WinCC y Startdrive para el diseño, desarrollo y diagnóstico de todos los controladores SIMATIC, pantallas HMI y accionamientos SINAMICS de última tecnología. [16]

Una solución de automatización típica contiene lo siguiente:

- Un controlador que configura el programa con la ayuda del software.
- Una pantalla de operador con el que se manipula y visualiza el proceso.

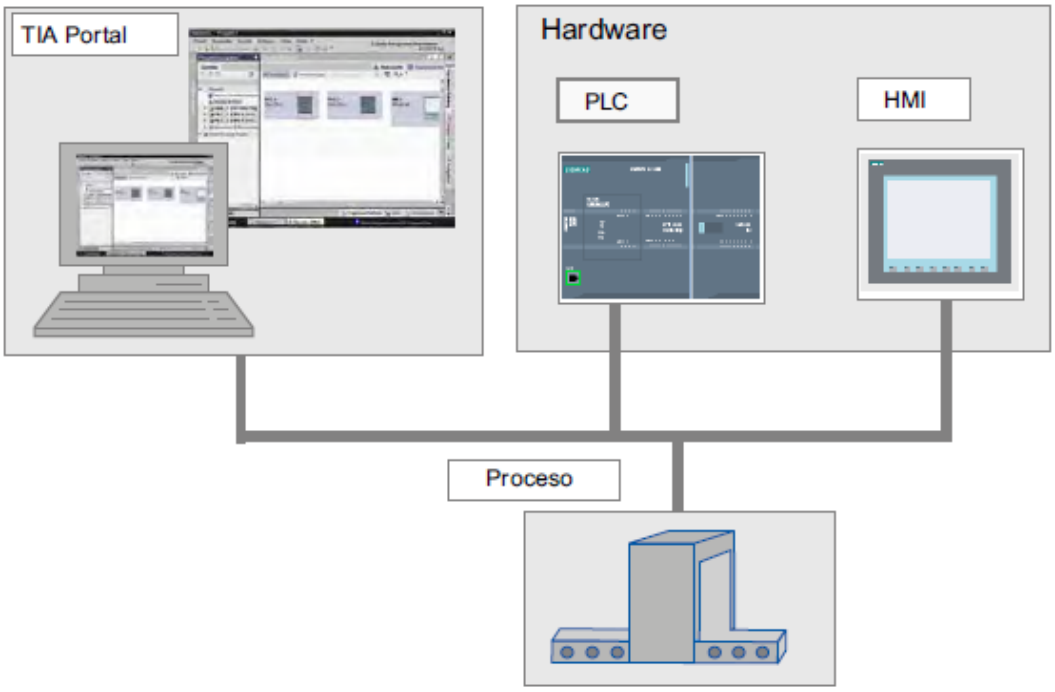


Figura 55: Distribución de proceso [16].

3.26.1. Tareas del TIA Portal

El TIA Portal ayuda a plantear soluciones de automatización. Los principales pasos de configuración son:

- Creación del proyecto
- Adaptación del hardware
- Conexión en red de los equipos
- Programación del controlador
- Configuración de la visualización
- Carga de datos de configuración
- Manejo de funciones online y diagnóstico

3.26.2. Ventajas del TIA Portal

El TIA Portal detalla las siguientes soluciones:

- Gestión unificada de los datos
- Manejo independiente de programas, los datos de configuración y los datos de visualización
- Sencillo modo de edición con la función Drag & Drop
- Facilidad de carga de datos en los equipos
- Manejo independiente
- Configuración y diagnóstico asistidos gráficamente

3.27. Simulador Modbus Poll

[17] El software Modbus Poll es un simulador maestro Modbus creado fundamentalmente para aportar a los desarrolladores de dispositivos esclavos Modbus u otros que necesiten probar y simular el protocolo Modbus. Con la interfaz de múltiples accesos, se logra revisar varios esclavos Modbus y/o áreas de datos al mismo tiempo. Por cada ventana, se necesita indicar la ID del esclavo Modbus, la función, la dirección, el tamaño y la tasa de sondeo. También es capaz de leer y escribir registros y bobinas desde cualquier ventana. Si se requiere cambiar un solo registro, se debe dar doble clic en el valor. O puede cambiar múltiples registros/bobinas. Los múltiples formatos de datos como float, double y long con intercambio de orden de palabras están a disposición.

Los errores generados en el programa se hayan en la línea de estado.

Si se utiliza un dispositivo de tipo esclavo, se puede componer y enviar propias cadenas de prueba en el "centro de prueba" y comprobar el resultado del esclavo en valores hexadecimales.

3.27.1. Variantes de protocolo compatibles:

- Modbus RTU
- Modbus ASCII
- Modbus TCP/IP
- Modbus RTU sobre TCP/IP
- Modbus ASCII sobre TCP/IP
- Modbus UDP/IP
- Modbus RTU sobre UDP/IP
- Modbus ASCII sobre UDP/IP

3.27.2. Datos y características técnicas de MODBUS POLL:

- Automatización OLE para una interfaz sencilla para Excel usando el lenguaje de macros VBA
- Lectura/escritura de hasta 125 registros
- Lectura/escritura de hasta 2000 entradas/bobinas
- Test Center (Componga sus propias cadenas de prueba)
- Imprimir e imprimir vista previa
- Monitoreo del tráfico de datos en serie
- Registro de datos en archivo de texto
- Registro de datos directo a Excel
- Ayuda sensible al contexto
- 28 Formatos de visualización como flotación, doble, etc.
- Base de dirección ajustable (0 o 1)
- Fuente y selección
- Selección de color condicional
- Transmisión (ID de esclavo 0)
- Control sencillo de convertidores RS-485 con interruptor RTS

- Modo ENRON / DANIEL
- 6 dígitos de direcciones

3.27.3. Funciones de Modbus soportadas:

- 01: Leer el estado de la bobina
- 02: estado de entrada de lectura
- 03: Leer registros de tenencia
- 04: leer los registros de entrada
- 05: bobina única de fuerza
- 06: registro único preestablecido
- 15: Forzar múltiples bobinas
- 16: registros múltiples pre ajustados
- 17: Informar la identificación del esclavo
- 22: Registro de escritura de máscara
- 23: Lectura / escritura de registros

3.27.4. Visión de conjunto.

Modbus Poll utiliza una interfaz de usuario de múltiples ventanas. Eso significa que puede abrir varias ventanas que muestran diferentes áreas de datos o datos de diferentes ID esclavos al mismo tiempo. Puede escribir cualquier texto en las celdas Alias.

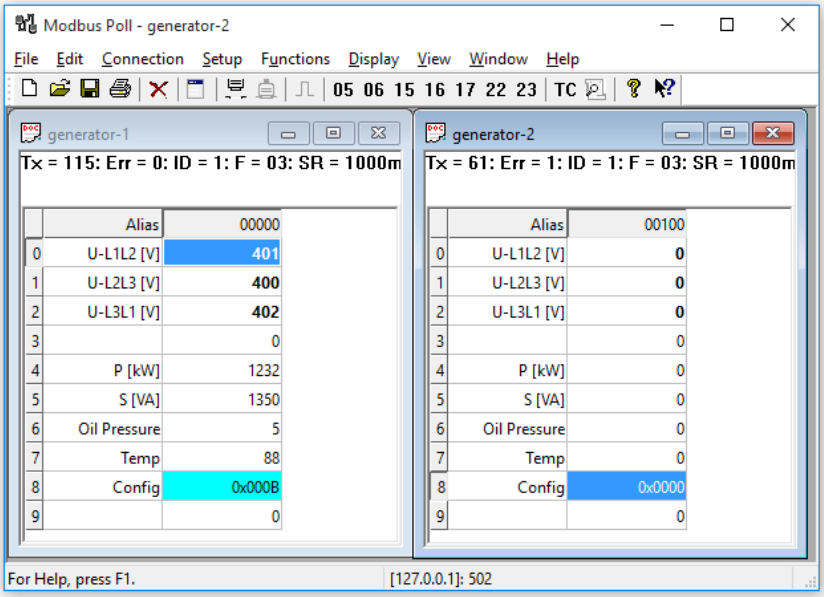


Figura 56: Interfaz de conexión.

Esta imagen muestra dos ventanas abiertas, una lectura de 10 registros de retención de la dirección 0 y una lectura de 10 registros de la dirección 100.

Si su dispositivo esclavo le permite cambiar un registro de retención, haga doble clic en la celda o simplemente comience a escribir un nuevo valor en la celda. Luego se muestra un cuadro de diálogo de edición.

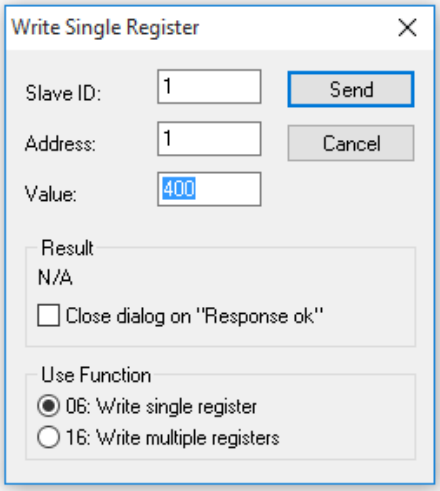


Figura 57: Cuadro de diálogo edición esclavo.

3.27.5. Cambiar la definición de lectura / escritura

Para cambiar la definición de lectura / escritura de una ventana, puede presionar F8 o seleccionar "definición de lectura / escritura" en el menú Configuración.

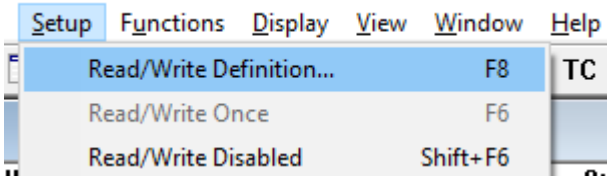


Figura 58: Barra de herramientas.

Aquí se define los datos a mostrar en la ventana. Esta configuración muestra cómo leer 10 registros de mantenimiento desde la dirección 0. Dirección 40001 en algunas descripciones de protocolo. Se debe tener en cuenta que Modbus Poll utiliza direcciones Modbus que siempre cuentan desde 0.

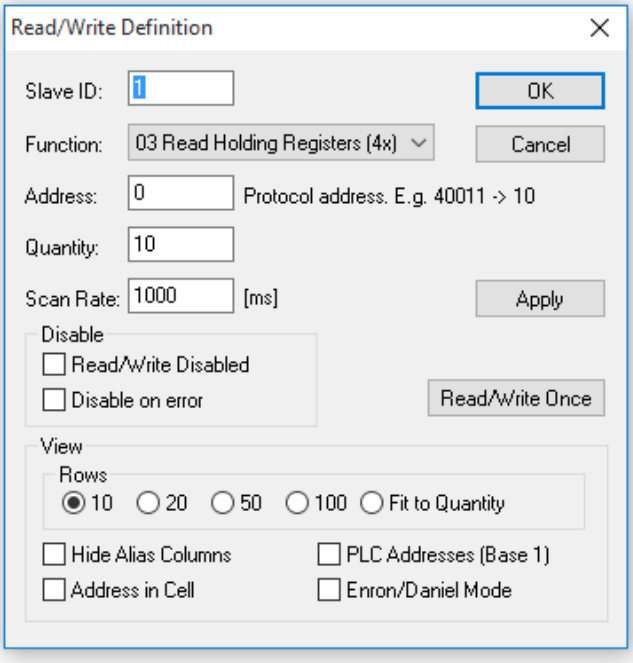


Figura 59: Ventana de configuración.

Tabla 6: Rangos de direcciones del dispositivo y MODBUS [17].

Dirección del dispositivo	Dirección MODBUS	Descripción	Función	R / W
1 ... 10000 *	Dirección 1	Bobinas (salidas)	01	Leer escribir
10001 ... 20000 *	dirección - 10001	Entradas discretas	02	Leer
40001 ... 50000 *	dirección - 40001	Registros de mantenimiento	03	Leer escribir
30001 ... 40000 *	dirección - 30001	Registros de entrada	04	Leer

3.28. Simulador Modbus Slave.

[17] El software Modbus Slave se utiliza para simular hasta 32 equipos esclavos en 32 ventanas. Una fuerte ventaja es que acelera la programación de los PLC con las herramientas de simulación. Los datos contenidos con cualquier documento abierto son accesibles para la aplicación maestra. La interfaz de usuario que Modbus Poll utiliza es similar a Modbus Slave. Función de soporte 01, 02, 03, 04, 05, 06, 15, 16, 22 y 23. Supervisión del tráfico en serie.

3.28.1. Variantes de protocolo compatibles:

- Modbus RTU
- Modbus ASCII
- Modbus TCP/IP
- Modbus RTU sobre TCP/IP
- Modbus UDP/IP
- Modbus RTU sobre UDP/IP

3.28.2. Características y características técnicas del esclavo MODBUS:

- Automatización OLE para una interfaz sencilla para Excel usando el lenguaje de macros VBA
- Imprimir e imprimir vista previa
- Monitoreo del tráfico de datos en serie
- Ayuda optimizada al contexto
- 12 Formatos de visualización como flotación, doble, etc.
- Base de dirección ajustable (0 o 1)
- Selección de fuente y color
- Transmisión (ID de esclavo 0)
- Control sencillo de convertidores RS-485 con interruptor RTS

3.28.3. Funciones de Modbus soportadas:

- 01: Leer el estado de la bobina
- 02: estado de entrada de lectura
- 03: Leer registros de tenencia
- 04: leer los registros de entrada
- 05: bobina única de fuerza
- 06: registro único preestablecido
- 15: Forzar múltiples bobinas
- 16: registros múltiples pre ajustados
- 22: Registro de escritura de máscara
- 23: Lectura / escritura de registros

4. PRÁCTICAS DE LABORATORIO

En esta sección se detallará paso a paso el desarrollo, diseño e implementación de cada una de las 10 prácticas propuestas, para lo cual se contará con herramientas adicionales a los dos módulos implementados, los cuales son:

- Software Modbus Poll / Slave.
- Cable convertidor de USB a RS485.
- Módulo de variador de frecuencia, el cual consta de un variador de frecuencia de marca ABB modelo ACS355, tarjeta de comunicación FMBA-01 y un motor trifásico marca ABB.
- Planta didáctica de tanques de agua marca Festo con PLC Siemens, tomado de una anterior tesis de la UPS.

Todo esto con la finalidad de demostrar la amplia gama de equipos y sistemas los cuales pueden ser compatibles y a la vez comunicados a través de la tecnología Modbus RTU, se mencionarán las diferentes conexiones y configuraciones para cada caso.

Figura 60: Elementos a utilizar en las prácticas de laboratorio.

4.1. Práctica Nº 1

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización I</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>1</td><td>TÍTULO PRÁCTICA: “Comunicación MODBUS entre un maestro y un esclavo.”</td></tr>
<tr><td colspan="4">OBJETIVOS:

• Conocer el funcionamiento de la comunicación MODBUS RTU.
• Comprobar las entradas y salidas del módulo, configurando el PLC y el HMI.
• Diseñar un proyecto en Tia Portal V12 el cual comunique un maestro PC y un esclavo PLC.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Instalar correctamente los softwares a utilizar con sus respectivos drivers.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:

• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• PC con programa Modbus Slave.
• Cable convertidor USB a RS485.

DESCRIPCIÓN:

Se debe enviar 10 datos desde la PC hacia el PLC donde se deben guardar los registros enviados, para esto se debe configurar la PC como esclavo y al PLC como maestro. Se podrá enviar la información a cualquiera de los dos módulos, uno a la vez pero con una señal de habilitación para establecer la comunicación Modbus RTU a través del cable RS485.

La figura 61 muestra la arquitectura de red a implementar.</td></tr>
</table>

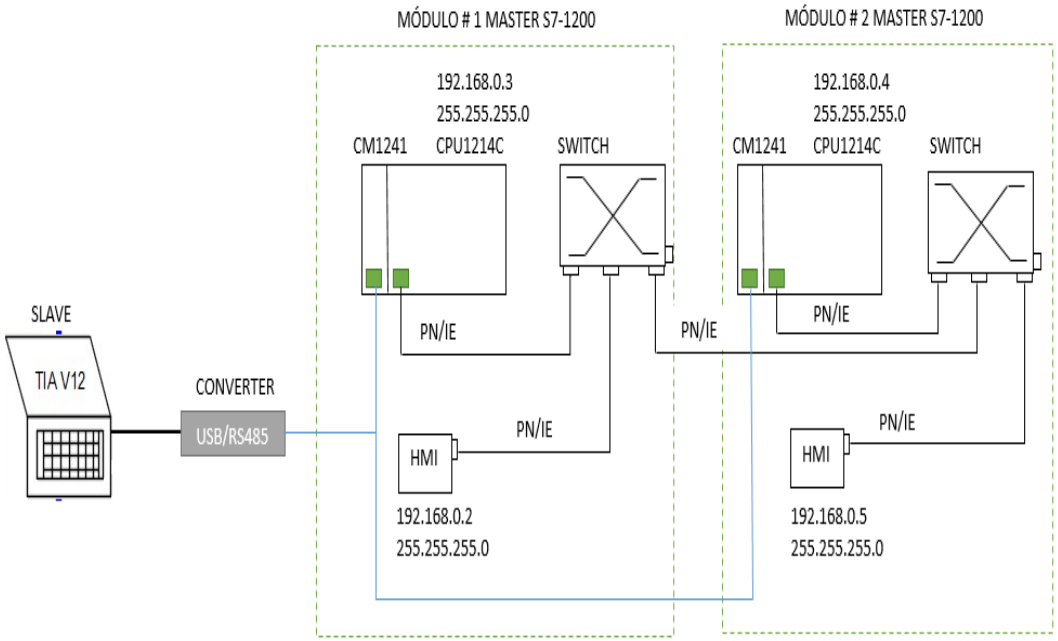


Figura 61: Arquitectura de red.

En figura 62 se muestra el diagrama de conexiones a implementar.

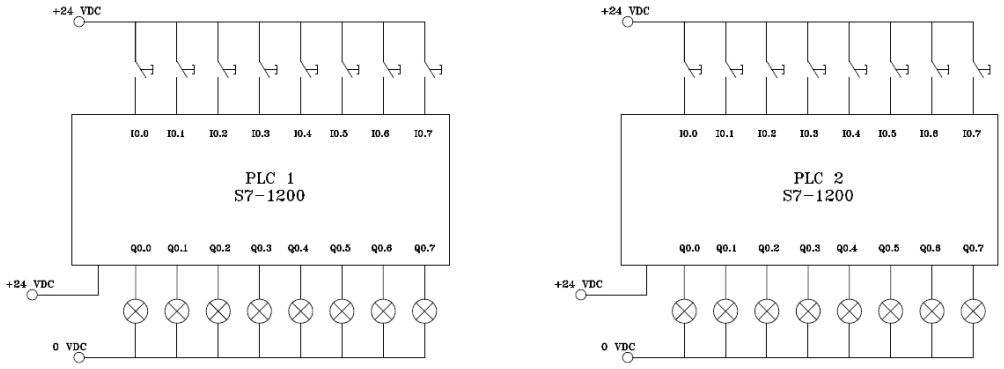


Figura 62: Diagrama de conexiones.

Paso 1.

Comenzamos energizando los módulos didácticos por medio del cable de poder a 120 VAC, luego se deberá poner en **ON** el **Interruptor** que se encuentra a lado de la entrada de 120 VAC, así mismo como el breaker de protección.

Figura 63: Conexión del cable de poder 120 VAC.

Paso 2.

Se procede a conectar con el cable Ethernet la laptop o PC hacia el módulo de prácticas en el puerto P1. El indicador LED del puerto LAN del PLC S7-1200 parpadeará.

Figura 64: Conexión del cable Ethernet en el puerto P1.

Paso 3.

Procedemos a ejecutar el software TIA PORTAL V.12 desde el escritorio, luego damos clic en **Iniciar**, luego clic en **Crear proyecto**, aquí se deberá llenar los campos de **Nombre proyecto**, **Ruta** que es donde desea guardar el proyecto, **Autor** y **Comentario.**
Para finalizar la creación del archivo del proyecto debemos dar clic en el botón **Crear**, ubicado en la parte inferior derecha de la pantalla.

Cabe indicar que para el proyecto a implementar se utilizará un mismo archivo para las 10 prácticas propuestas, es decir no se crearan proyectos independientes por cada práctica. Esto facilitará el modo de operación y control de las prácticas establecidas y a su vez mejora el interfaz el cual es más agradable a la vista del operador.

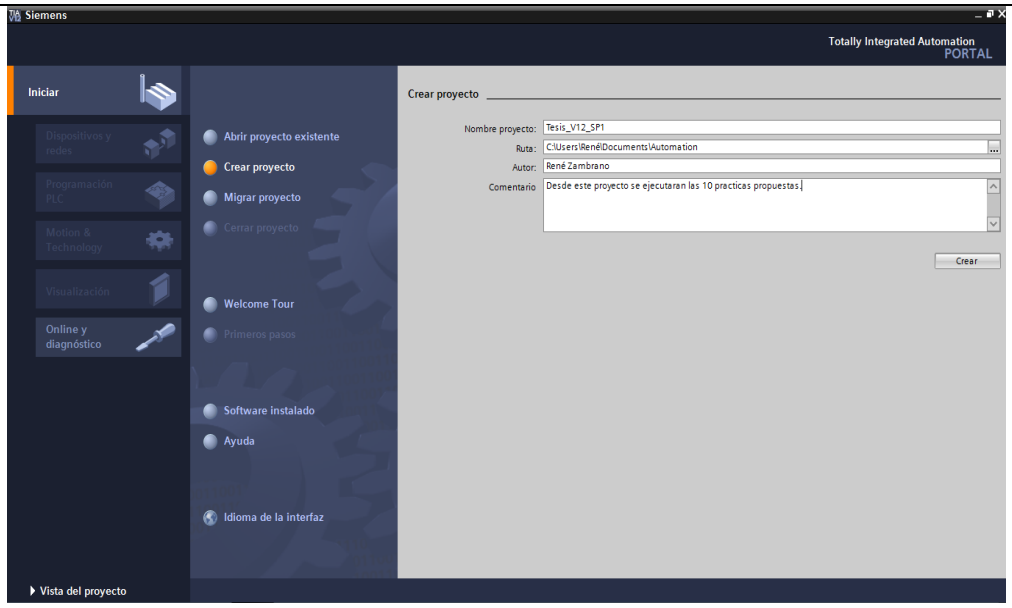


Figura 65: Interfaz de inicio.

Paso 4.

A continuación aparece la pantalla **Primeros pasos**, procedemos a dar clic en la etiqueta **Dispositivos y redes, Configurar un dispositivo.**

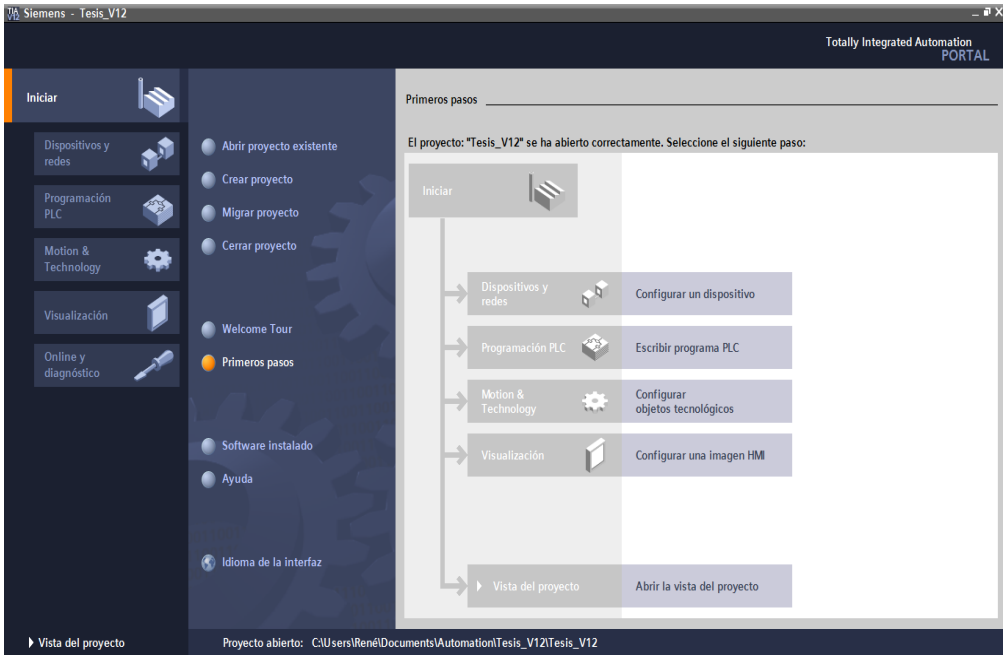


Figura 66: Interfaz primeros pasos.

Paso 5.

En este paso vamos a agregar los dispositivos que vamos a utilizar en la práctica 1 por lo pronto son 2 PLC y 2 pantallas HMI, para esto hacemos lo siguiente:

En la ventana siguiente dar clic en la opción **Agregar dispositivo**, luego en **Controladores**. A continuación, dar clic en las siguientes carpetas **Controladores**, **SIMATIC S7-1200**, **CPU**, **CPU 1214C AC/DC/Rly** donde seleccionaremos el

modelo **6ES7 214-1BE30-0XB0**.

En la pestaña **Versión**, seleccionar **V2.2**, y dar clic en el botón **Agregar.**

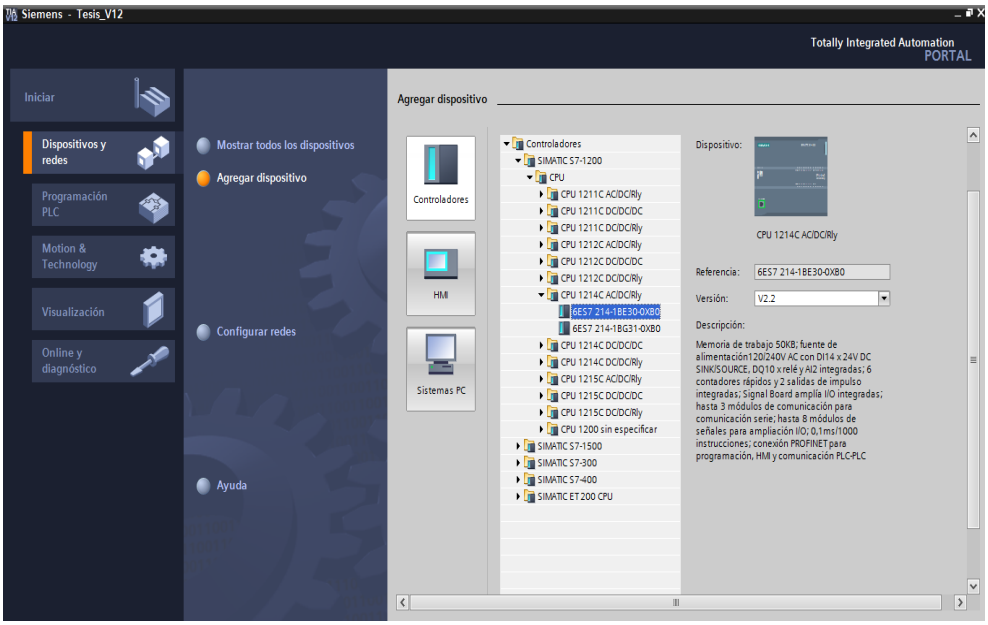


Figura 67: Interfaz Agregar dispositivo.

Paso 6.

Una vez agregado el PLC_1 le cambiaremos el nombre a “MASTER”, para esto nos dirigimos al **Árbol del proyecto**, damos clic en el nombre del proyecto **Tesis_V12_SP1**, se despliegan las opciones y seleccionamos el controlador **PLC_1**, damos clic derecho y seleccionamos la opción **Cambiar nombre** donde vamos a escribir **MASTER.**

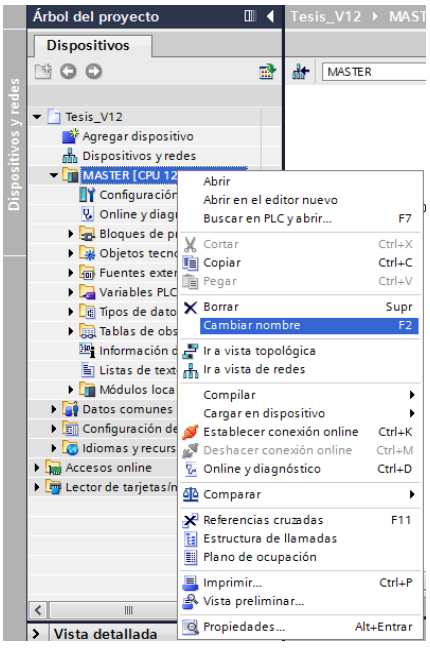


Figura 68: Árbol del proyecto.

Paso 7.

Ahora procederemos a agregar los módulos de comunicación y tarjetas de interfaz al PLC MASTER, esto lo utilizaremos para la comunicación MODBUS RTU y para el uso de entradas y salidas analógicas con la Signal board.

Damos doble clic **Configuración de dispositivo** donde se despliega la vista de redes y dispositivos. Nos vamos al lado derecho de la pantalla y damos clic en la pestaña **Catálogo de hardware**. Luego seleccionamos **Signals Boards**, luego en **AQ**, seguido de **AQ1 x 12 bits** y finalmente damos doble clic en **6ES7 232-4HA30-0XB0** o simplemente seleccionamos y arrastramos hacia el PLC.

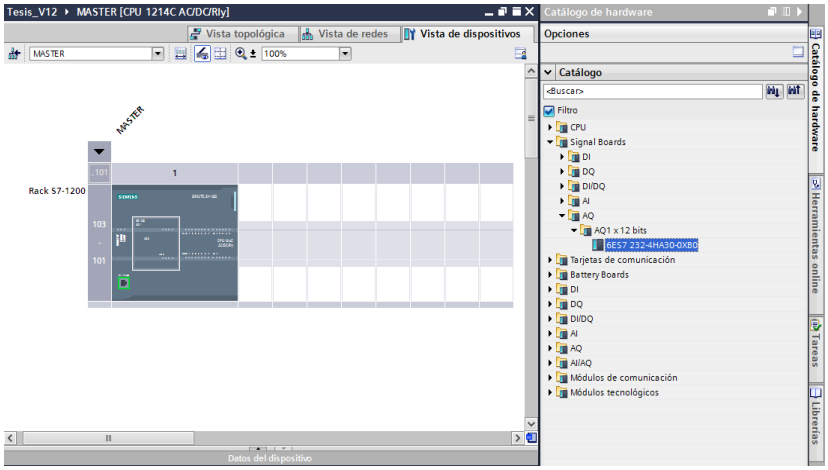


Figura 69: Asignación de Signal Board.

Paso 8.

Para agregar el módulo de comunicación damos clic en **Catálogo de hardware**. Luego clic en **Módulos de comunicación**, luego clic en **Punto a punto**, seguido de **CM1241 (RS485)** y finalmente damos doble clic en **6ES7 241-1CH30-0XB0** o simplemente seleccionamos y arrastramos hacia el PLC.

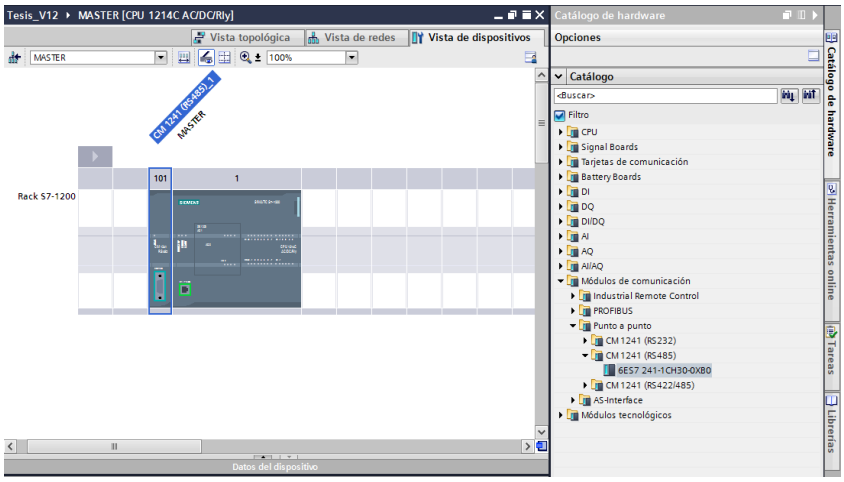


Figura 70: Asignación de Módulo de comunicación.

Procedemos a configurar los valores de la trama MODBUS RTU perteneciente al módulo recientemente agregado. Damos doble clic sobre el puerto DB9 del CM1241 donde se despliega la **Interfaz RS485,** seleccionamos **Configuración del puerto** donde registramos lo siguiente:

- Velocidad de transferencia: 19.2 kbits
- Paridad: Sin paridad
- Bit de datos: 8bits por carácter
- Bit de parada: 1

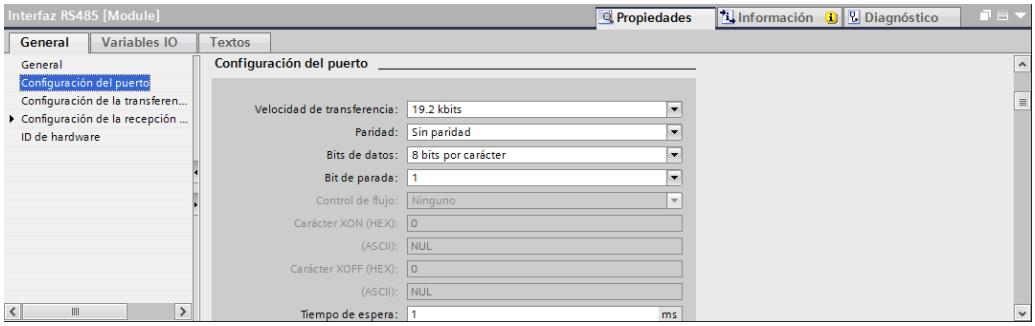


Figura 71: Configuración del puerto RS485.

Observación: Se debe configurar el puerto RS485 del CM1241 perteneciente al PLC SLAVE con los mismos valores.

Paso 9.

Una vez agregado el PLC MASTER procedemos a agregar el PLC_2 al proyecto el cual asignaremos el nombre de "SLAVE", para esto nos dirigimos al **Árbol del proyecto** y damos doble clic en **Agregar dispositivo**, **Controladores**, **SIMATIC S7-1200**, **CPU**, **CPU 1214C AC/DC/Rly,** modelo **6ES7 214-1BE30-0XB0**.

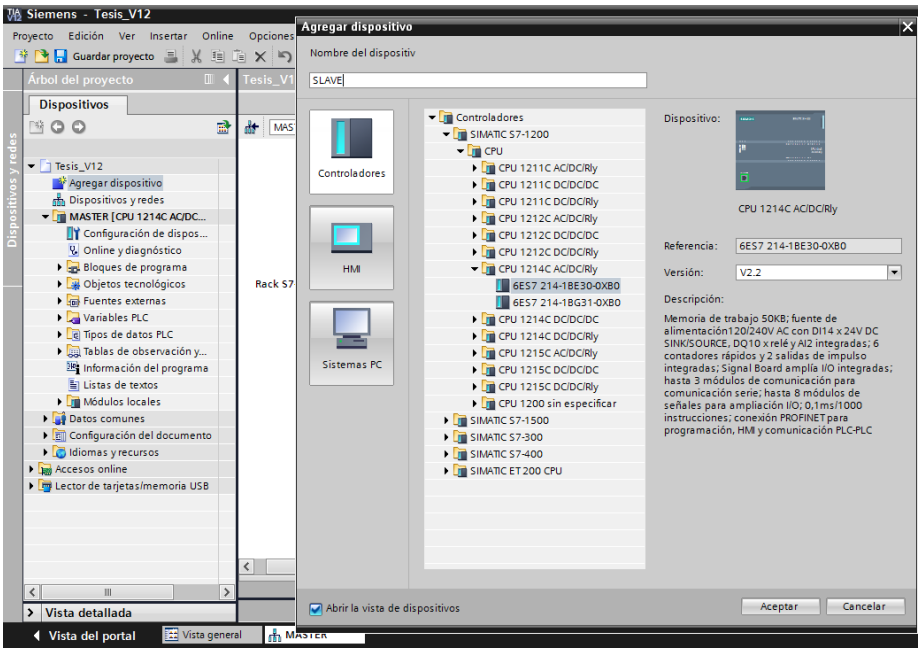


Figura 72: Interfaz para agregar al PLC SLAVE.

Al PLC SLAVE se deben agregar las tarjetas y módulos al igual que el PLC MASTER, para esto se deben repetir los pasos 7 y 8.

Paso 10.

Al estar creados los PLC maestro y esclavo con sus respectivas tarjetas y módulos, procedemos a crear las pantallas HMI_1 y HMI_2 para cada uno respectivamente.
Para agregar el HMI_1_MASTER dirigirse a **Árbol del proyecto** y escoger **Agregar dispositivo**.
En la pantalla **Agregar dispositivo**, escoger la opción **HMI**, y abrir las siguientes pestañas: **HMI**, **SIMATIC Basic Panel**, **6" Display,** luego **KTP600 Basic color PN**.
Dirigirse a la pestaña **Versión** y escoger **12.0.0.0**, y dar clic en el botón **Aceptar.**

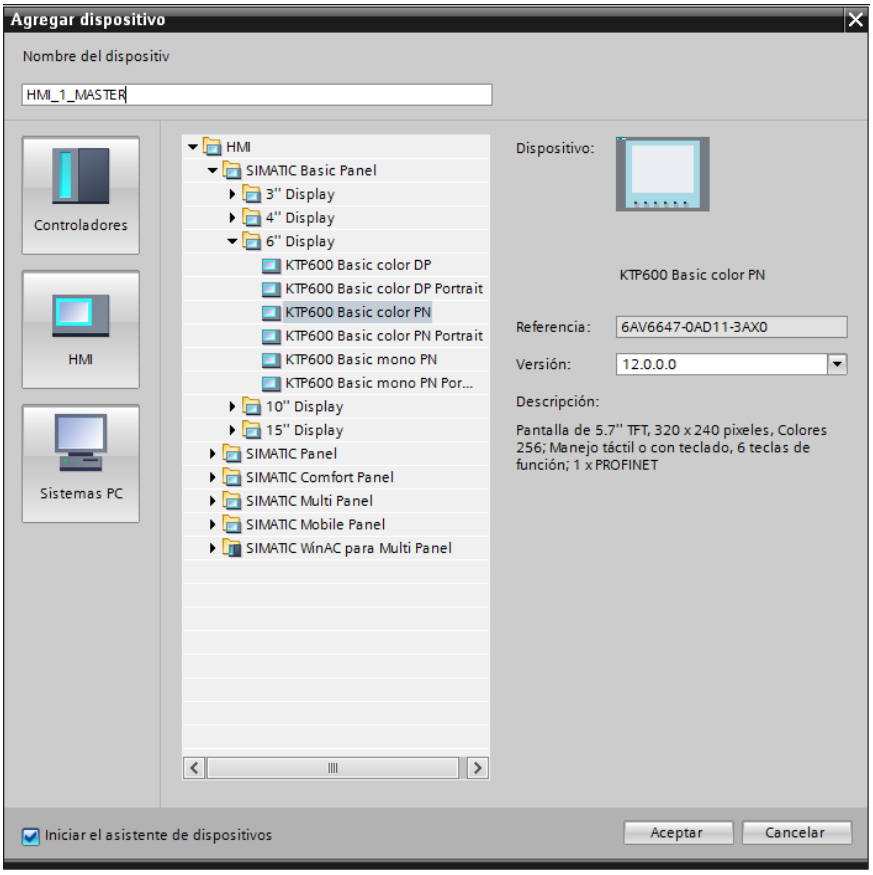


Figura 73: Interfaz para agregar HMI.

Paso 11.

Se mostrará la pantalla **Asistente del panel de operador: KTP600 Basic color PN**, en la pantalla **Conexiones del PLC** damos clic en **Examinar** y seleccionamos al PLC **MASTER**. Debe aparecer una línea verde indicando la estabilización de la conexión entre el PLC **MASTER** y la pantalla **HMI_1_MASTER**. El driver de comunicación será SIMATIC S7-1200 y la interfaz será PROFINET (X1). A continuación damos clic en **Siguiente**.

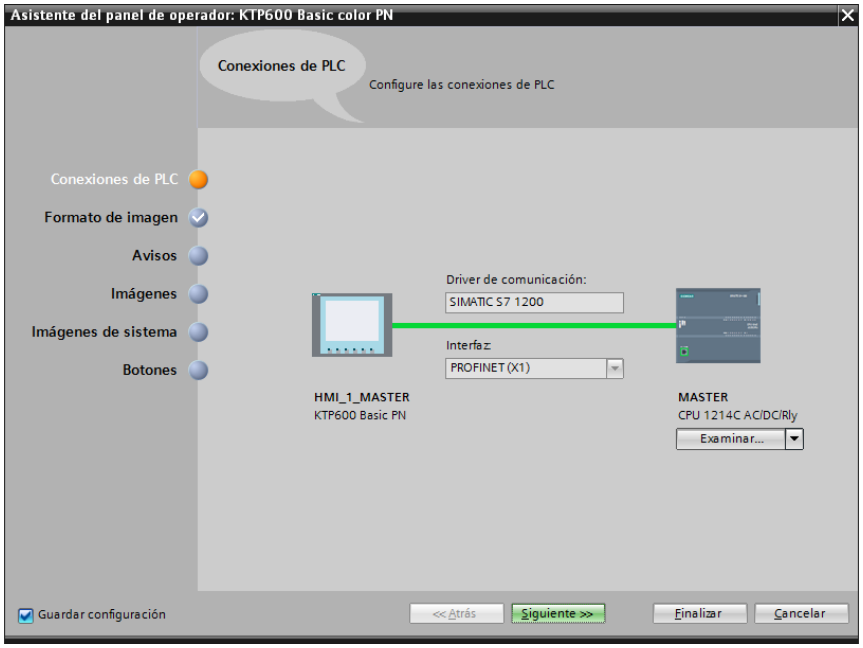


Figura 74: Conexiones de PLC.

Damos clic en siguiente donde nos muestra la ventana **Formato de imagen**, procedemos a desmarcar el **Encabezado**.

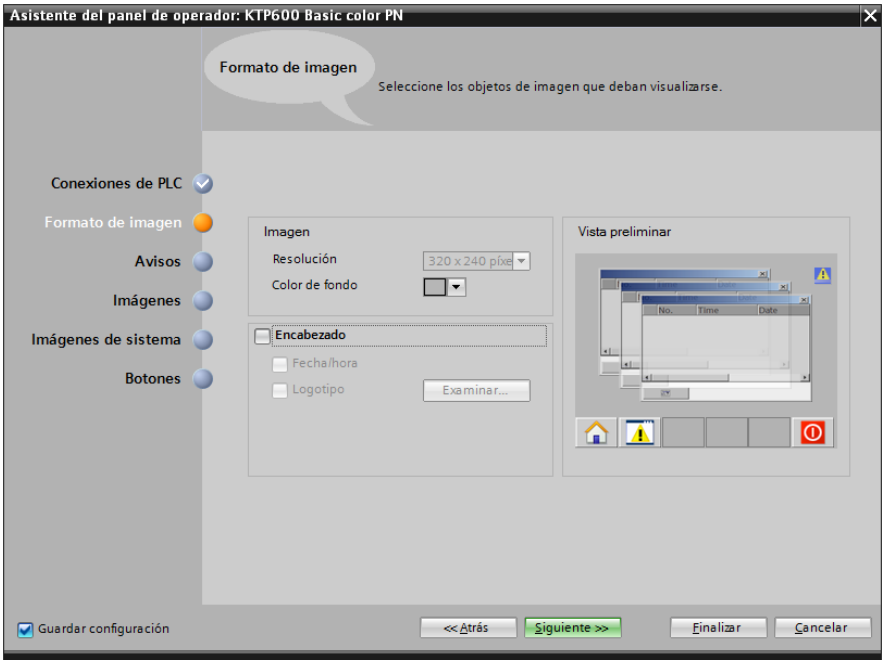


Figura 75: Formato de imagen.

Presionamos siguiente y nos arroja a la ventana **Avisos**, procedemos a desmarcar las siguientes opciones: **Avisos no acusados**, **Avisos pendientes** y **Avisos de sistema pendientes,** como lo muestra la figura 75.

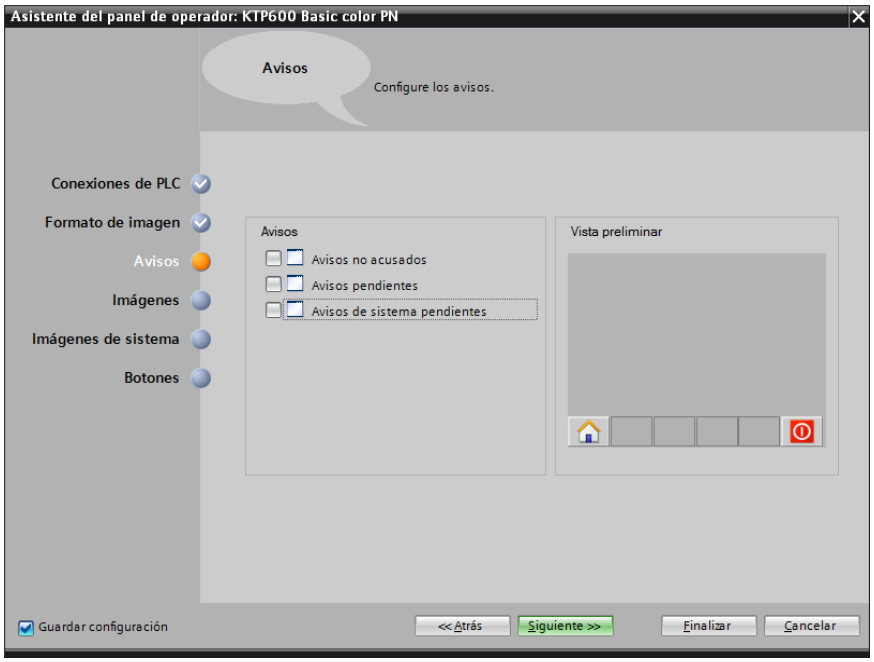


Figura 76: Avisos.

Luego damos clic en **Siguiente** hasta llegar a la opción **Botones**, donde desmarcaremos la casilla **Abajo** y **Finalizar**.

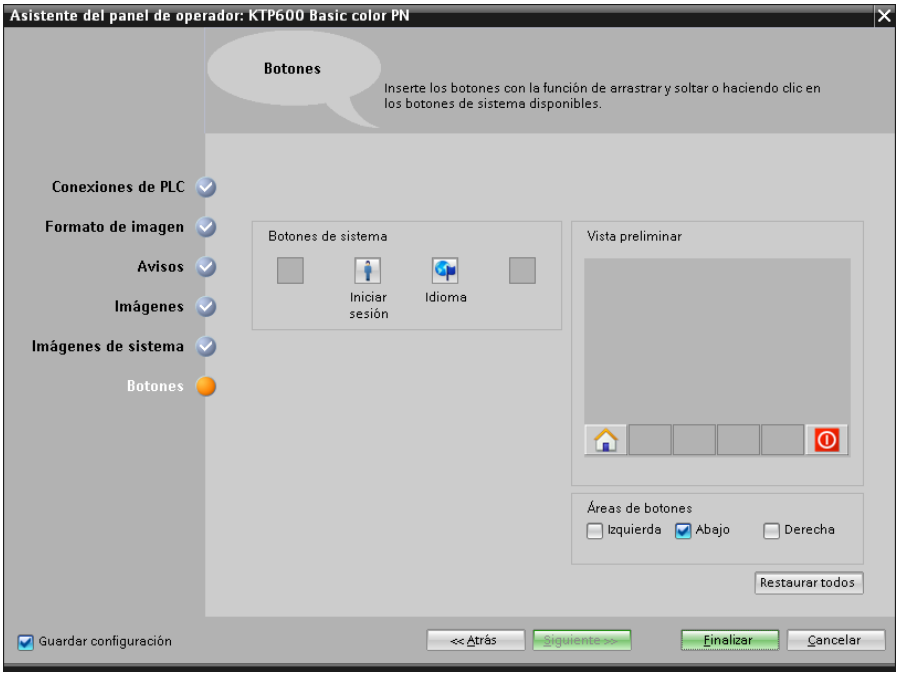


Figura 77: Botones.

Para crear el HMI para el PLC SLAVE se debe repetir el paso 10 pero esta vez

seleccionando SLAVE en la opción PLC.

Paso 12.

En este paso vamos a interconectar en red de tipo Profinet los dispositivos agregados al proyecto, para esto nos dirigimos a **Árbol del proyecto**, luego **Dispositivos y redes**, seguidamente **Vista de redes**. Se procede a dar clic en los puertos Ethernet y arrastrar el puntero hacia los demás puertos en serie.

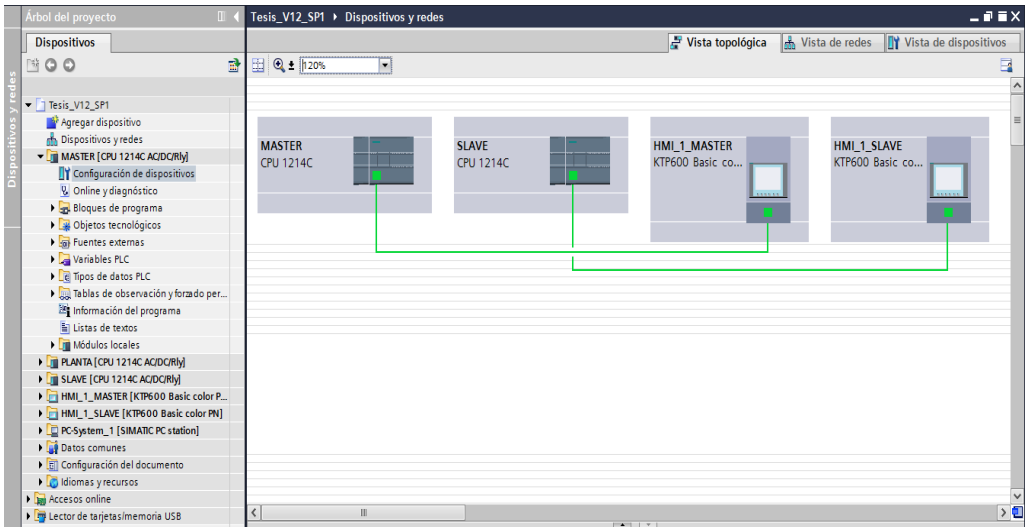


Figura 78: Vista topológica.

Paso 13.

A continuación vamos a realizar la **Configuración de puertos Ethernet** para asegurar la comunicación Profinet entre los dispositivos agregados recientemente. Para esto vamos a establecer la dirección IP para cada dispositivo los mismos que se utilizarán para todas las prácticas.

Debemos dar doble clic en el puerto Ethernet del PLC MASTER, se abrirá una ventana con las propiedades del puerto. Una vez abierta la ventana **Propiedades** del puerto **Interfaz PROFINET_1**, damos clic en **Direcciones Ethernet**, dentro de esta nos dirigimos a **Protocolo IP** y modificamos la **Dirección IP** de acuerdo a nuestro diseño.

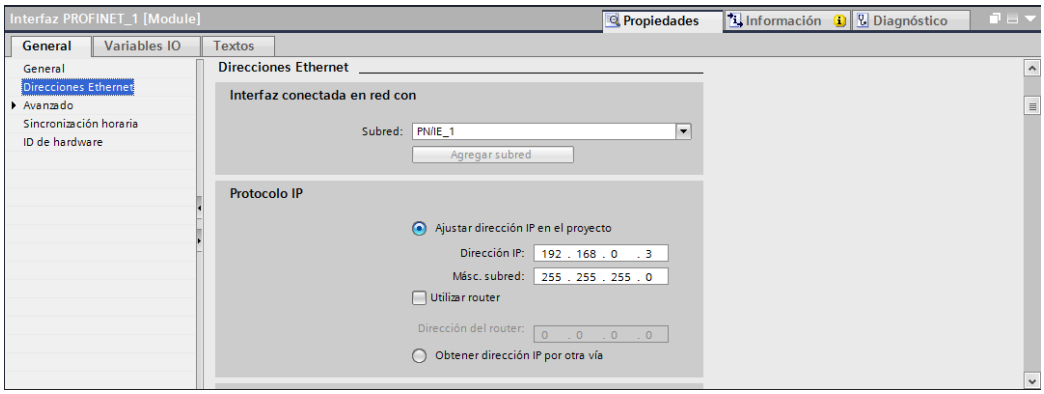


Figura 79: Direcciones Ethernet.

Las direcciones IP a configurar en cada dispositivo son las siguientes:

- Dirección IP PLC MASTER: 192.168.0.1
- Dirección IP PLC SLAVE: 192.168.0.4
- Dirección IP HMI_1_MASTER: 192.168.0.2
- Dirección IP HMI_1_SLAVE: 192.168.0.5

Paso 14.

En este paso activaremos las marcas del sistema con el fin de utilizarlas a lo largo de las prácticas y configuración de bloques. Hacemos clic derecho en el PLC **MASTER**, en donde se abrirá un menú y presionamos **Propiedades**, aparecerá el menú **General**. Seleccionamos la opción **Marcas del sistema y de ciclo,** nos ubicamos en **Bits de marcas de sistema** y marcamos la casilla **Activar la utilización del byte de marcas de sistema.**

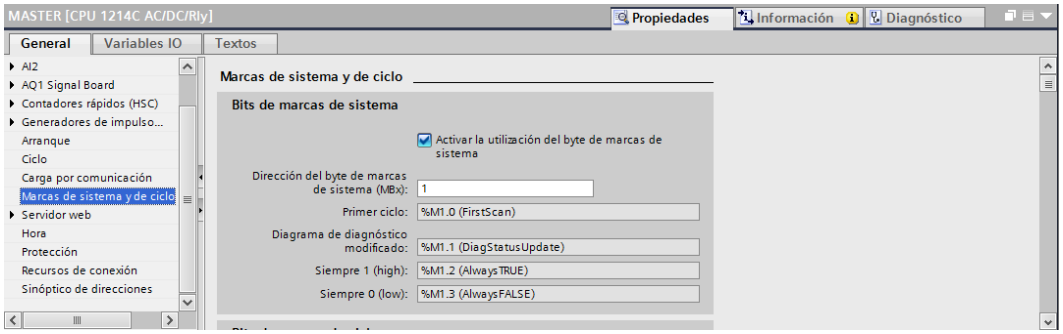


Figura 80: Activación de marcas de sistema.

Paso 15.

Ahora nos daremos paso para la creación de las imágenes del HMI, las cuales serán la presentación del proyecto, al igual que la imagen de proceso y selección de la práctica 1.

Para crear la imagen de presentación accedemos al **Árbol del proyecto**, damos clic en la pestaña **HMI_1_MASTER**, luego en **Imágenes** y después hacer doble clic en **Agregar imagen**.
Notamos la nueva imagen creada donde le damos clic derecho y escogemos **Propiedades**.
En la **Lista de propiedades** seleccionamos **General**, aquí es donde le dará un nombre a la imagen creada, en este caso será **Menú**.
Definimos a la imagen **Menú** como imagen inicial y de presentación del proyecto, para esto nos vamos a **Árbol del proyecto**, **HMI_1, Imágenes**, clic derecho sobre la imagen y seleccionamos **Definir como imagen inicial**.

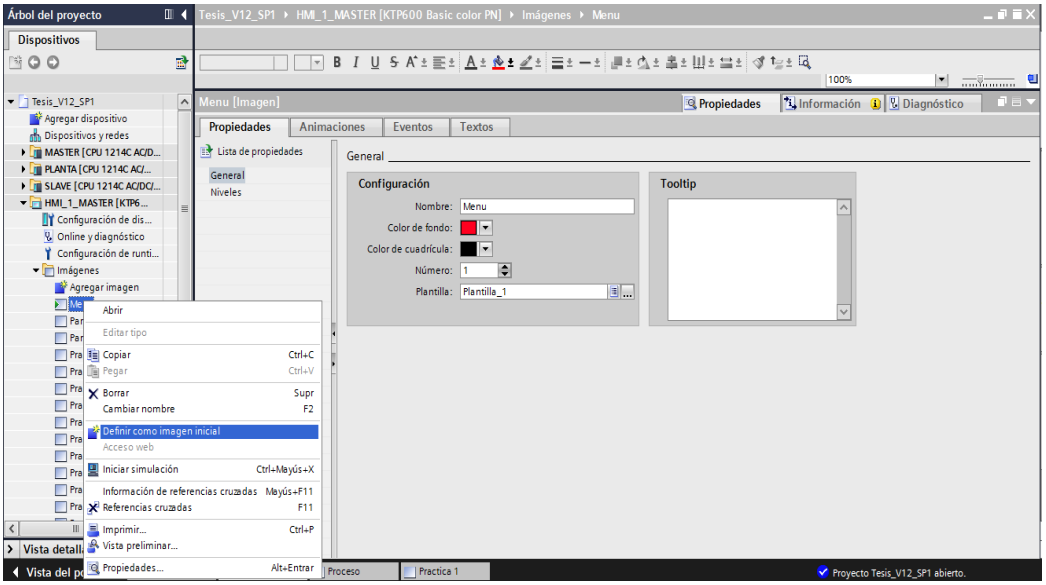


Figura 81: Creación de imagen Menú.

Paso 16.

En este paso procedemos a diseñar la imagen **Menú** la misma que fue asignada como presentación del proyecto y servirá para las demás prácticas.

Figura 82: Diseño de imagen Menú en HMI_1_MASTER.

El botón ubicado en la parte inferior derecha de **Menú** sirve para direccionarnos a la imagen de selección de prácticas llamado **Proceso**. Para añadir del botón con la flecha hacia la derecha seleccionamos del lado derecho de la pantalla la opción **Herramientas, Elementos, Botón.**

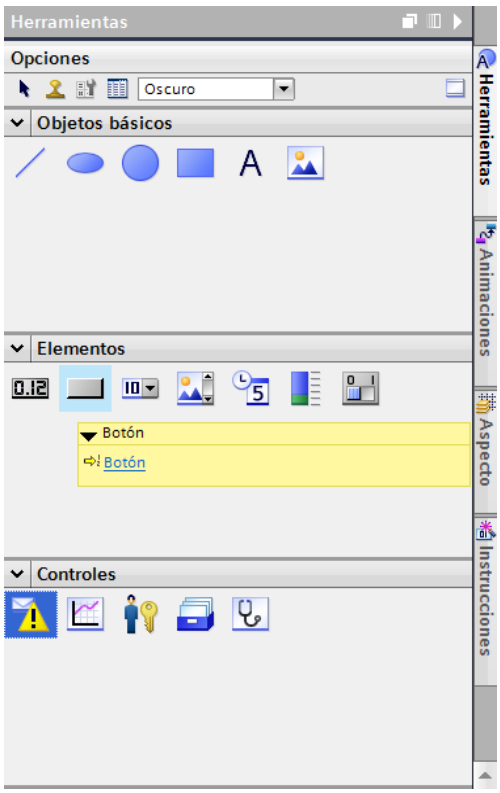


Figura 83: Agregar elementos a la imagen.

Trazamos el tamaño del **Botón_1** en la imagen y damos doble clic, en la pestaña **Propiedades**, **General,** y en **Modo** marcamos **Gráfico**, en la opción **Gráfico** seleccionamos igualmente Gráfico. Tanto en la forma "no pulsado" como "pulsado" vamos a seleccionar la imagen **Right_Arrow** suministrada por el software.

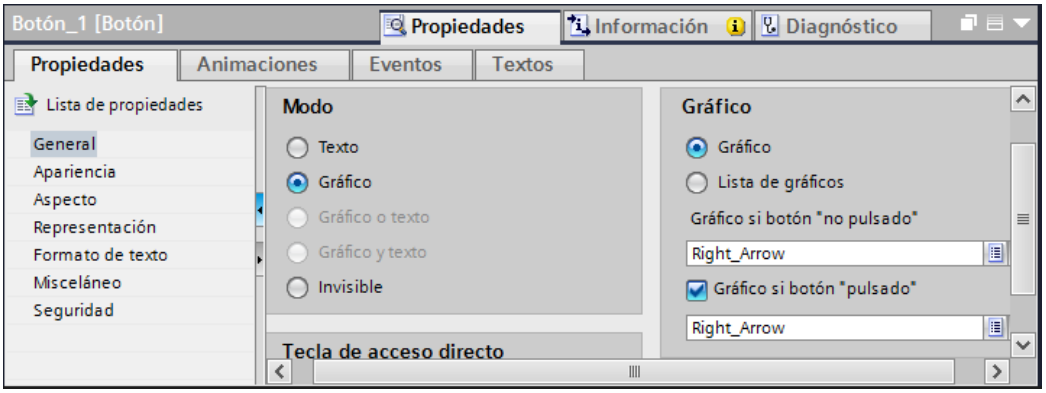


Figura 84: Asignación de gráfico para el botón.

Pasamos a la pestaña **Eventos, Hacer clic, Agregar función.** Seleccionamos la imagen **Proceso** la cual al presionar el botón nos dirigirá hasta allí.

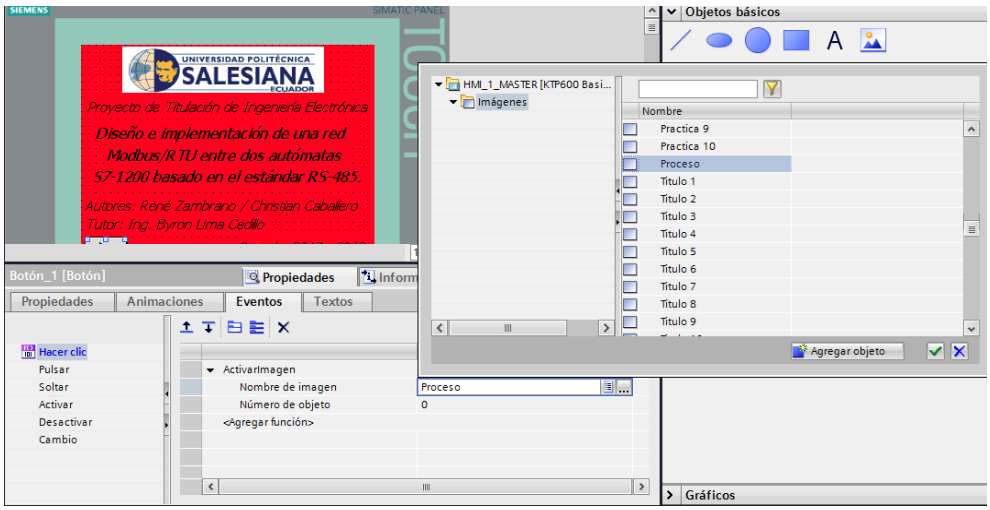


Figura 85: Selección del evento para Botón_1.

Nota: la imagen **Menú** del HMI_1_MASTER será el mismo diseño para el HMI_1_SLAVE.

Paso 17.

Procederemos a crear la imagen **Proceso** con la cual seleccionaremos las prácticas a ejecutar, para esto usaremos controles que se describirán a continuación. La imagen de proceso quedará como en la figura 75.

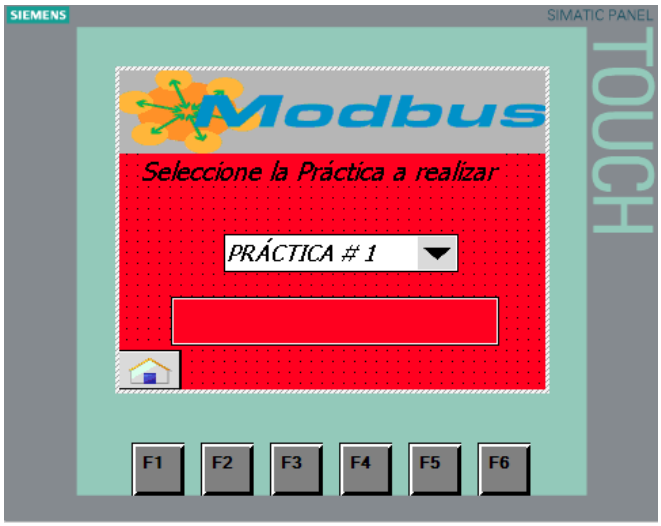


Figura 86: Diseño de imagen Proceso.

Nota: la imagen **Menú** del HMI_1_MASTER será el mismo diseño para el HMI_1_SLAVE.

Para el siguiente paso vamos a crear una **Lista de texto** para que al momento de presionar el botón **Campo ES simbólico** se desplieguen los textos desde la práctica 1 hasta la práctica 10. En **Lista de texto** damos doble clic en Agregar e ingresamos el

nombre **PRACTICAS**, luego vamos a **Selección** y elegimos **Rango (…-…).** Pasamos a la ventana **Entradas de la lista de textos** donde llenamos las 10 prácticas una a una.

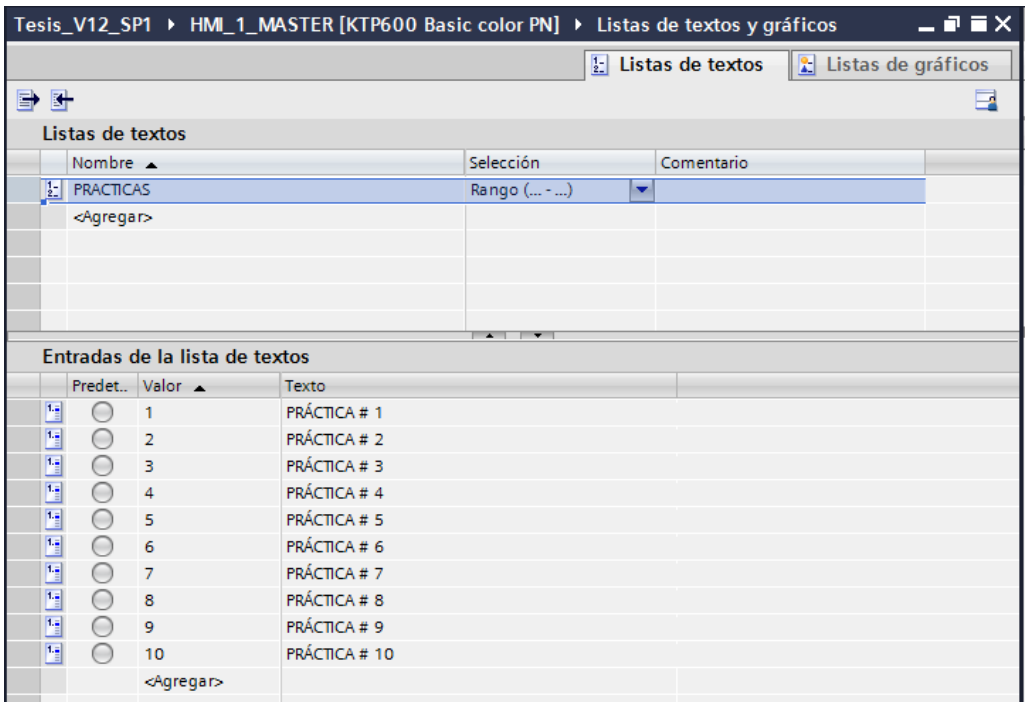


Figura 87: Lista de textos PRACTICAS.

Para desplegar las prácticas a realizar se deben agregar a la imagen **Proceso** el control **Campo E/S simbólico,** damos doble clic sobre el control y nos vamos a **Animaciones**, **Conexiones de variables**, **Valor de proceso**, dentro del cuadro **Proceso** se define la variable para la animación, presionamos el botón de buscar y elegimos la variable "CONFIGURACION_VARIABLES.COMBO BOX".
A su vez seleccionamos **Propiedades, Contenido, Lista de textos,** damos clic en la flecha verde y seleccionamos la lista **PRACTICAS**.

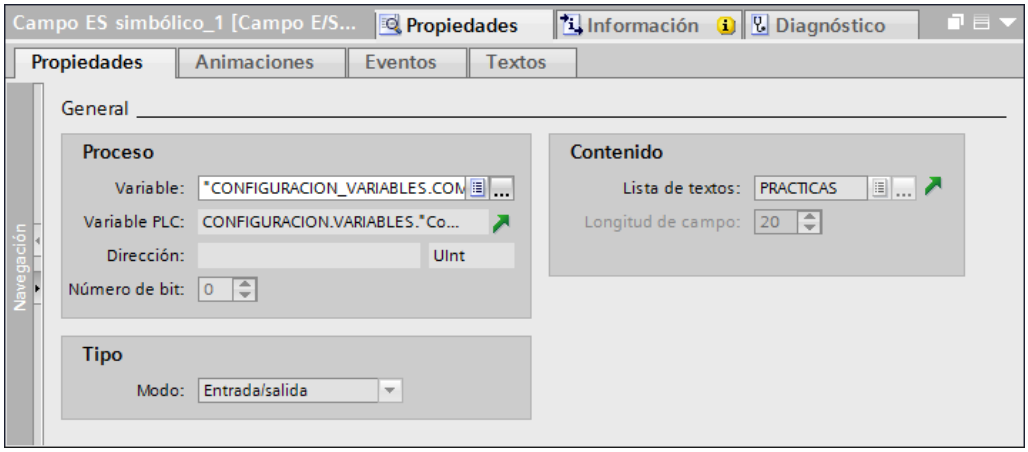


Figura 88: Asignación de variable para Campo E/S simbólico.

Nota: la imagen **Proceso** del HMI_1_MASTER será el mismo diseño para el

HMI_1_SLAVE.

Paso 18.

A continuación detallaremos la creación de la imagen de la **Práctica 1**.

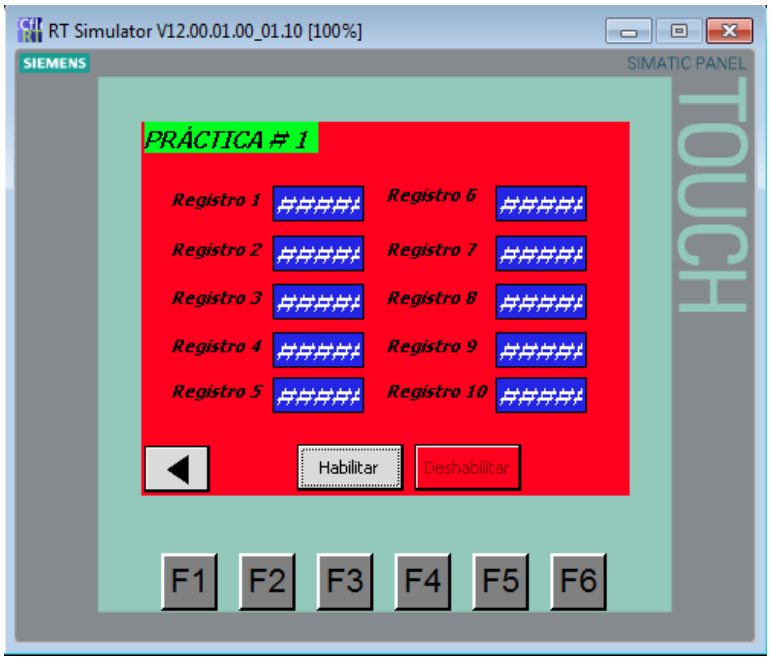


Figura 89: Diseño simulado de la imagen Práctica 1.

Se ingresan 10 **campos de E/S** y se asignan a las variables donde se van a guardar los datos enviados como se ve en la figura 79 y 80.

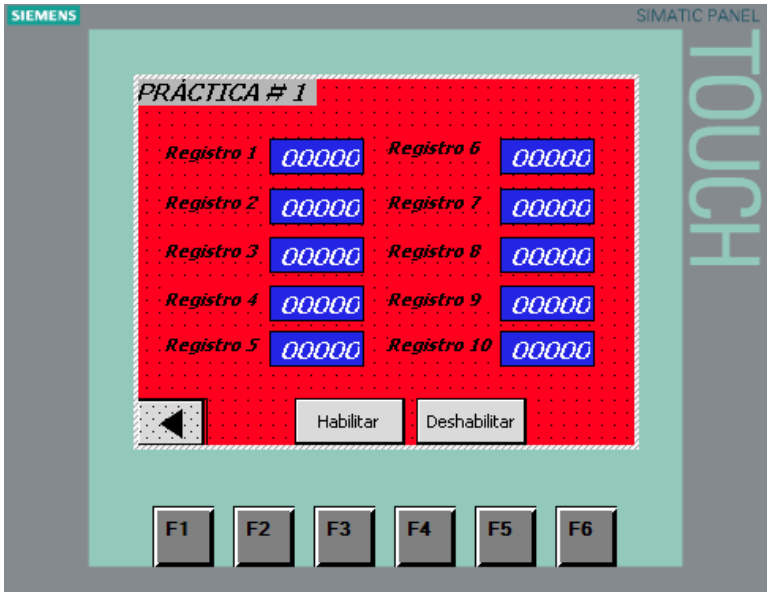


Figura 90: Diseño de la imagen Práctica 1.

Se necesita crear 10 variables para guardar los datos recibidos del maestro, se detalla a continuación la creación de las variables para los registros de la Práctica 1 dentro de un bloque de datos llamado **PRACTICAS [DB4].**

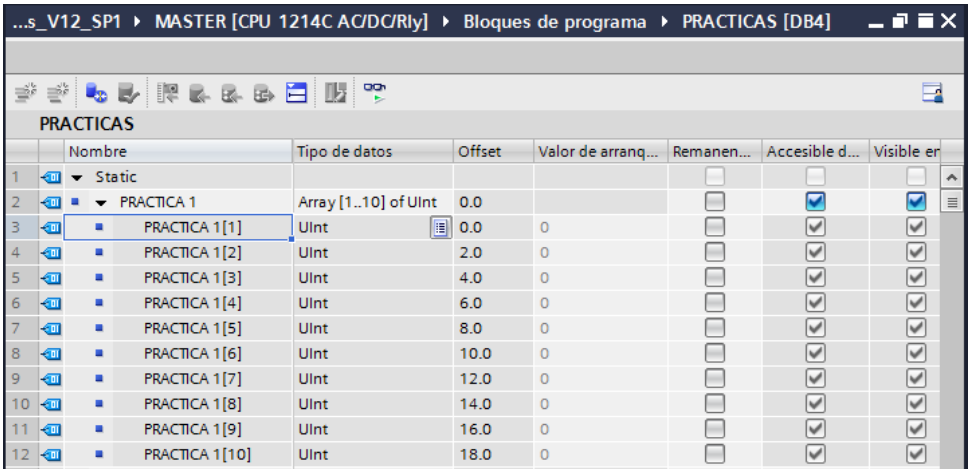


Figura 91: Variables de la práctica 1.

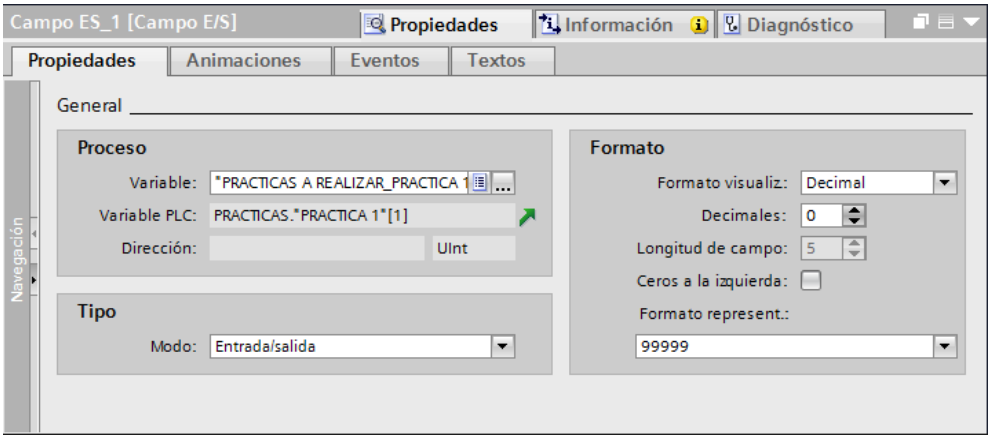


Figura 92: Asignación de variables para los campos de la práctica 1.

Paso 19.

En este paso detallaremos el diseño de la imagen **Título 1** la cual presenta el tema de la práctica 1, además se agregó un cuadro de dialogo el cual servirá para proseguir a la imagen **Práctica 1**.

Creamos un botón el cual llamaremos **Ejecutar**, nos vamos a las **Propiedades** del botón, **General** donde llenamos los siguientes campos:

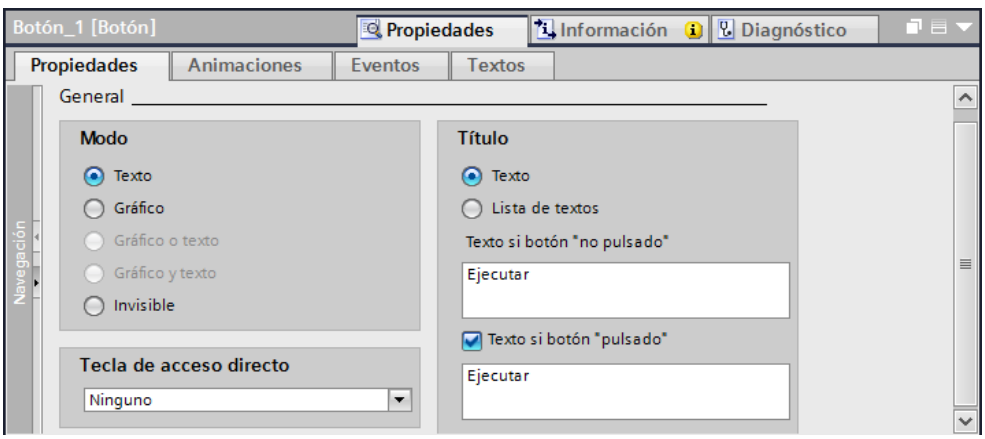


Figura 93: Propiedades del botón Ejecutar.

Luego asignamos la variable en la pestaña **Eventos**, esto para que al dar clic en el botón aparezca una ventana de aviso que explicaremos más adelante. Ingresamos a la opción **Hacer Clic**, **Agregar función** y escribimos **Activar bit** donde asignaremos la variable F36.

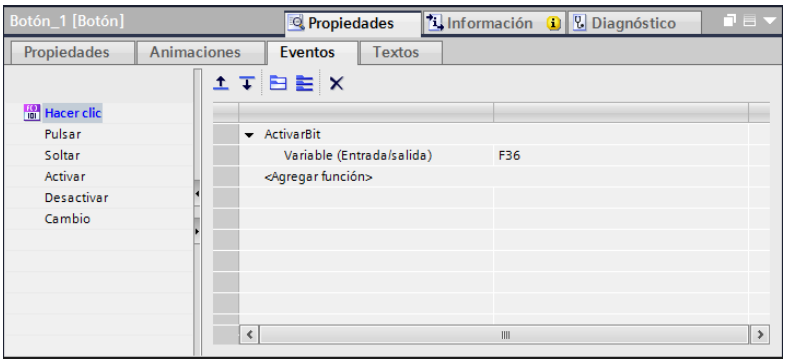


Figura 94: Eventos del botón Ejecutar.

Después de esto crearemos un mensaje de alerta el cual nos preguntará si deseamos ir a la práctica luego de presionar el botón ejecutar, si la respuesta es **Si** nos enviará a la imagen **Práctica 1** y si se escoge **No** nos enviara de vuelta a la imagen **Proceso**.
El mensaje está conformado por un **Rectángulo**, un **Cuadro de texto** y dos **Botones** los cuales serán agregados a la imagen actual.

El botón **Si** está configurado de la siguiente manera:

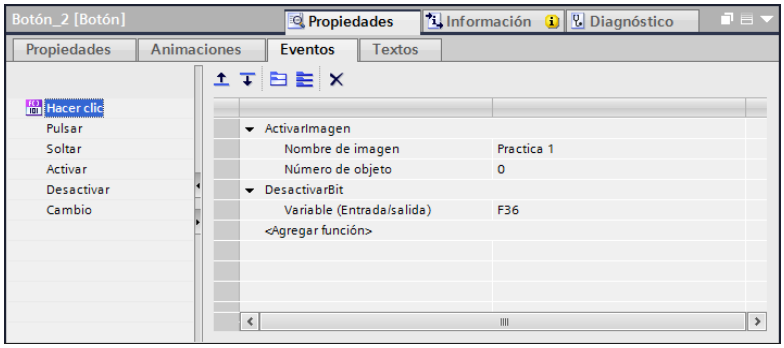


Figura 95: Eventos del botón Sí.

El botón **No** será configurado de la siguiente manera:

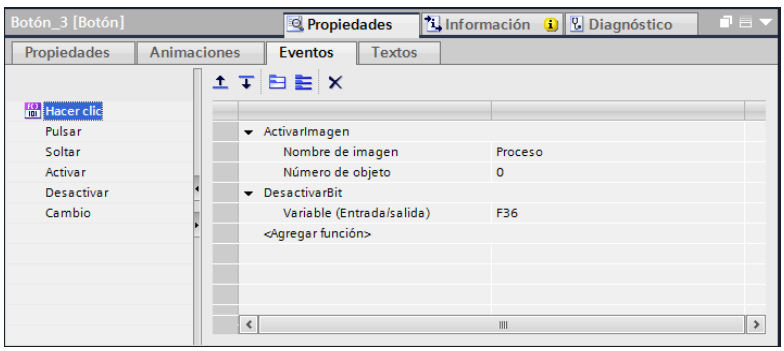


Figura 96: Eventos del botón No.

Finalmente con el clic izquierdo seleccionamos los elementos del mensaje damos clic derecho y luego presionamos **Agrupar**, como se muestra en la figura 86.

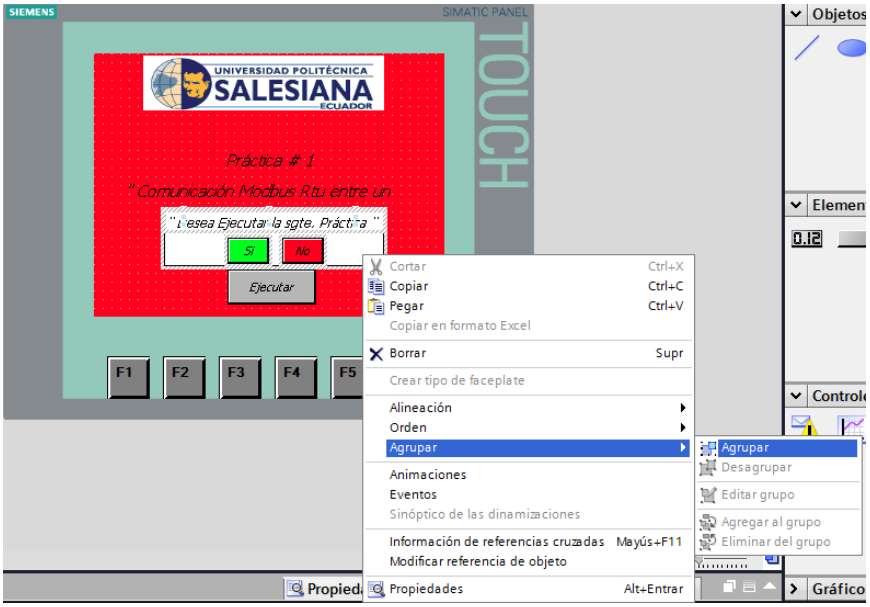


Figura 97: Agrupación de elementos del mensaje.

Nota: el mensaje de alerta de la imagen **Titulo 1** del HMI_1_MASTER será el mismo diseño para las imágenes de las demás prácticas (prácticas del 1 al 10) solamente se debe cambiar la variable para un correcto funcionamiento. Estas alertas sirven para el HMI_1_SLAVE.

Figura 98: Diseño terminado de imagen Título 1.

Paso 20.

En la imagen Práctica 1 debemos agregar 2 botones los cuales servirán para enlazar la comunicación, es decir, el botón **Habilitar** nos brindará la comunicación

MODBUS RTU y el botón **Deshabilitar** la cortará. Es importante indicar que las prácticas solo funcionarán con la comunicación habilitada.
Nos dirigimos a **Elementos**, **Botón** donde dibujamos en la imagen el tamaño del botón e ingresamos el nombre **Habilitar**.
Damos clic derecho en el botón **Habilitar** y luego **Propiedades**, nos dirigimos a la pestaña **Animaciones, Visualización, Agregar animación, Apariencia.**

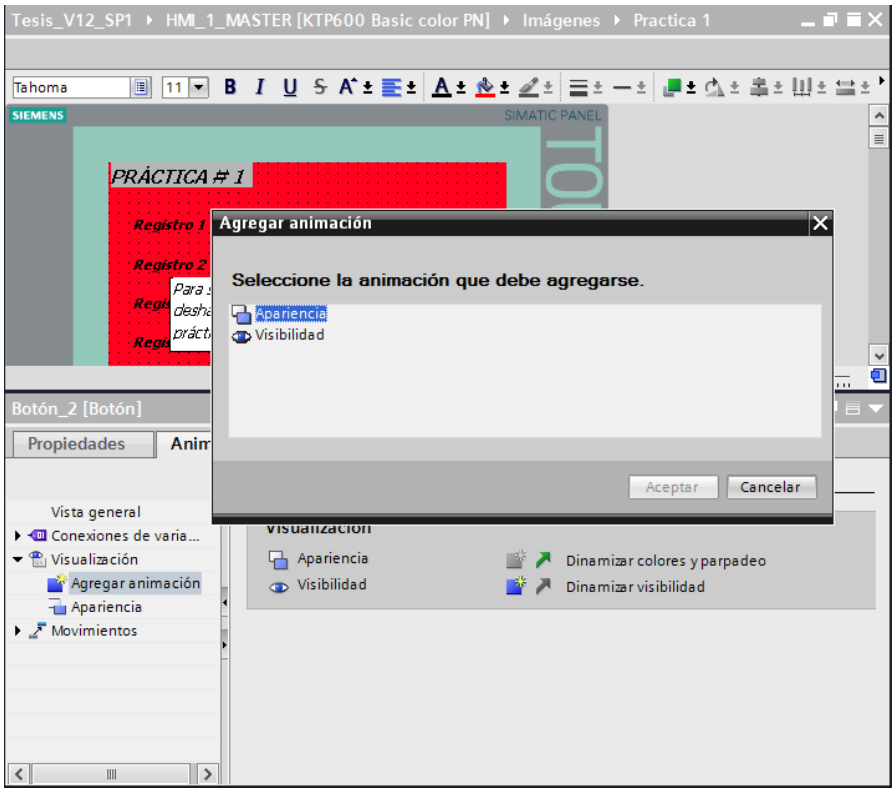


Figura 99: Animación del botón Habilitar.

Es aquí donde agregaremos un parpadeo al botón al momento de dar clic, realizamos la siguiente configuración: dentro de **Apariencia** vamos a **Variable**, Nombre donde seleccionamos la variable que utilizaremos para este botón la cual se llamaremos "CONFIGURACION_VARIABLES.Sistema_Hab[1]". En **Tipo** marcamos **Rango** y finalmente en el **Rango** damos doble clic en **Agregar**. Le damos el valor de “1”, seleccionamos un **Color de fondo** y un **Color de primer plano**. En la opción **Parpadeo** le damos Sí.

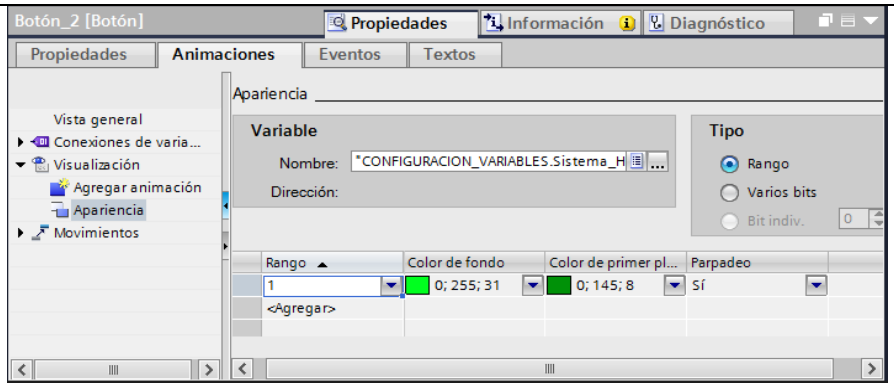


Figura 100: Asignación de variable de botón Habilitar.

Luego de agregar la animación nos dirigimos a la pestaña **Eventos**, **Pulsar**, **Agregar función**, es donde vamos a escribir el comando **ActivarBitMientrasTeclaPulsada.** Esto servirá para que al presionar una sola vez el botón permanezca encendido. Asignamos en **Variable (Entrada/Salida)** la variable "CONFIGURACION_VARIABLES.Hab_Prac[0]".

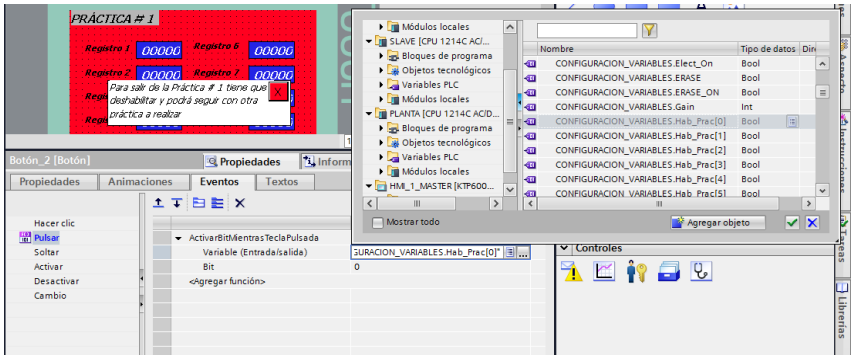


Figura 101: Eventos del botón Habilitar.

Para el botón **Deshabilitar** se deben realizar los mismos pasos del botón **Habilitar**, a diferencia que asignamos las variables correspondientes a esta función, cambiaremos los colores del botón a rojo y vino para mejorar la interfaz y el parpadeo sea más vistoso.

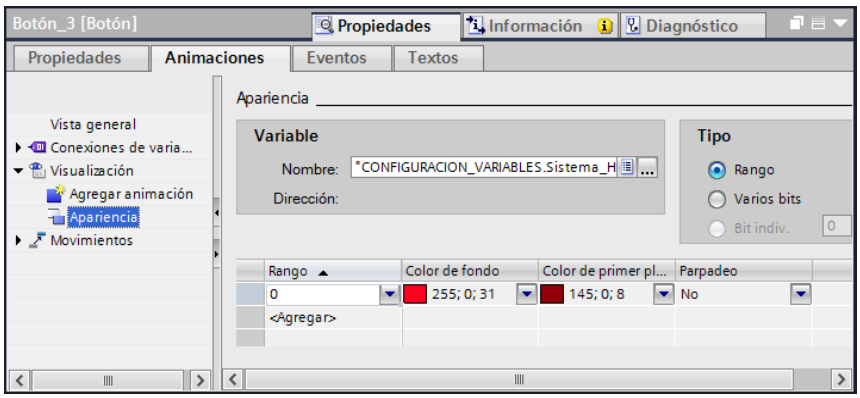


Figura 102: Animación del botón Deshabilitar.

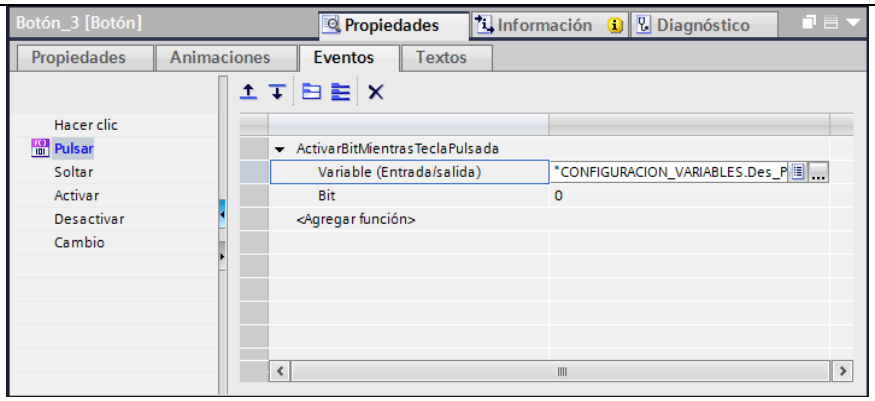


Figura 103: Eventos del botón Deshabilitar.

La presentación de la imagen Práctica 1 quedará de la siguiente manera:

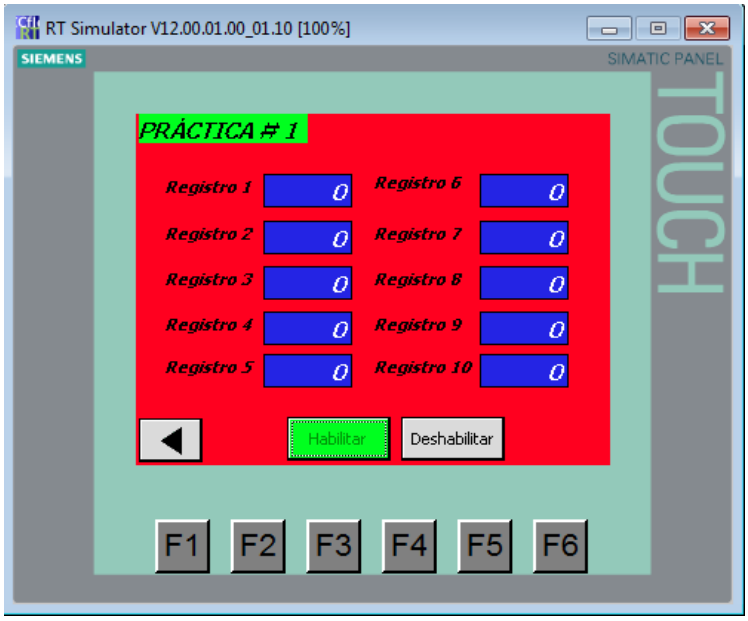


Figura 104: Descripción de botones Habilitar y Deshabilitar.

Nota: la imagen **práctica 1** del HMI_1_MASTER será el mismo diseño para el HMI_1_SLAVE, incluyendo variables y bloque de datos.

Adicionalmente se diseñó un mensaje de alerta el cual aparecerá en el momento que se salga de la práctica sin deshabilitar la misma. Para esto utilizaremos un **Grupo** conformado por un **Rectángulo**, tres **Cuadros de textos** y un **Botón**.

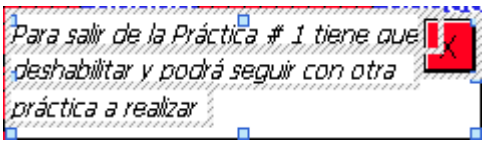


Figura 105: Grupo creado para el mensaje de alerta.

La configuración del grupo se muestra a continuación:

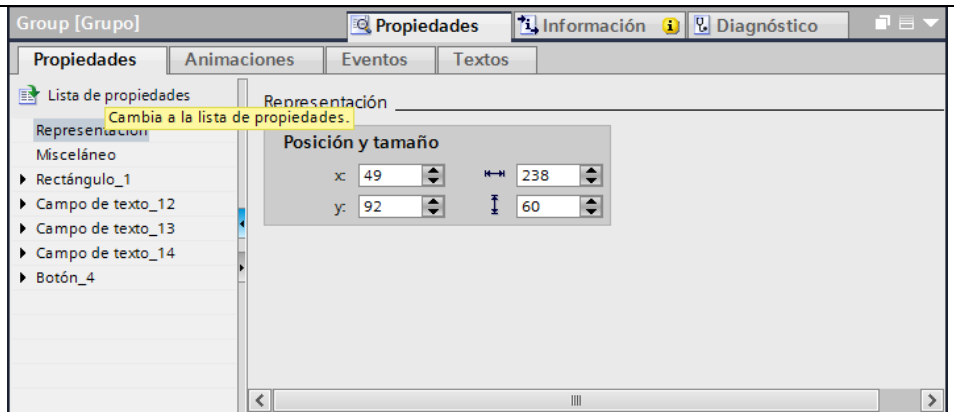


Figura 106: Propiedades de grupo de mensaje de alerta.

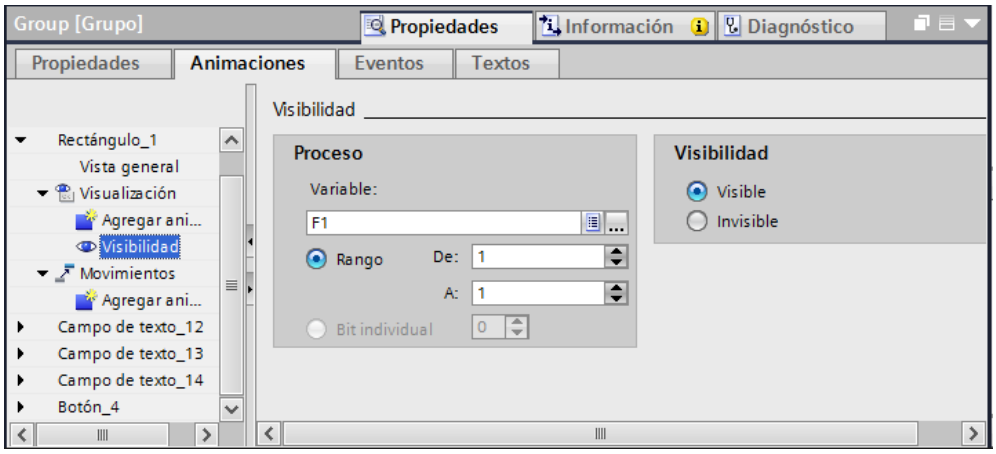


Figura 107: Animaciones de grupo de mensaje de alerta.

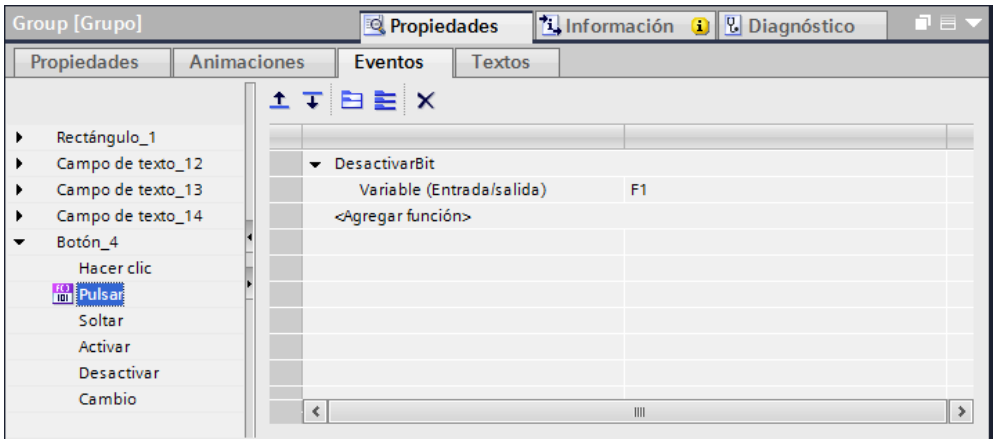


Figura 108: Eventos del grupo de mensaje de alerta

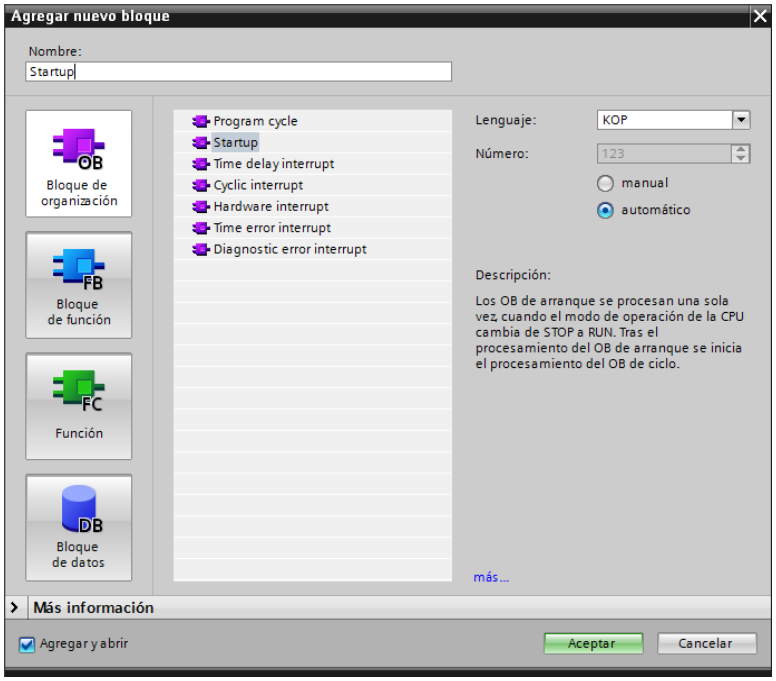
Paso 21.

A continuación detallaremos la programación de la práctica 1. Nos dirigimos al **Árbol de proyecto, Tesis_V12_SP1, MASTER, Bloque de programas** y damos doble clic en **Main [OB1]**.

El OB1 lo renombraremos como **MAESTRO**. Luego vamos a agregar un segundo bloque de organización que llamaremos **Startup** el cual arrancará primero que todos los bloques de programación.

Esto se utilizará con el fin de arrancar los bloques **LOAD** para los maestros y esclavos a utilizar en las prácticas. Nos dirigimos a **Árbol de proyecto, Tesis_V12_SP1, MASTER, Bloque de programas** y **Agregar nuevo bloque**.

Luego seleccionamos **Bloque de organización** y en **Nombre** escribimos **Startup**, en la opción **Lenguaje** escogemos **KOP** y **automático**. Finalmente presionamos el botón Aceptar.



Ahora nos dirigimos a la parte derecha de la pantalla en la pestaña **Instrucciones**, **Comunicación**, **Procesador de comunicaciones**, **MODBUS**, y damos doble clic en la instrucción **MB_COMM_LOAD**.

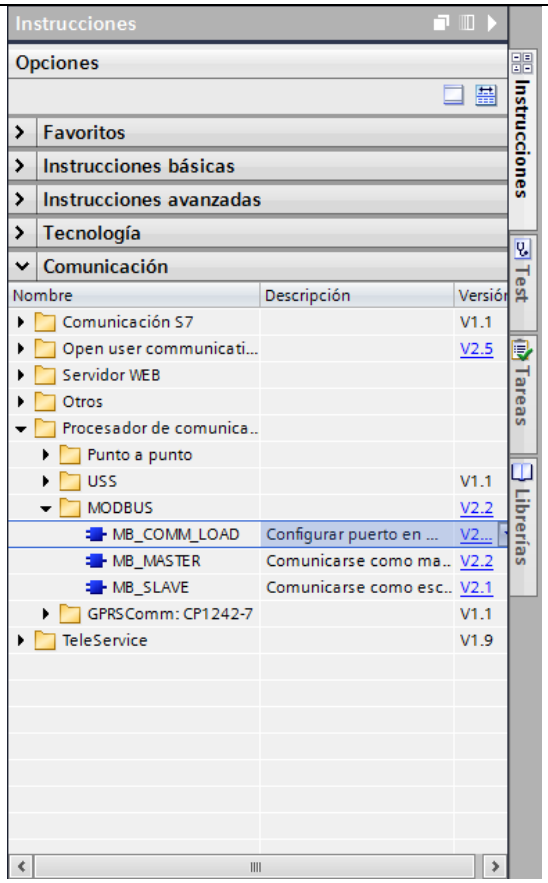


Figura 109: Ubicación del interfaz de las instrucciones MODBUS.

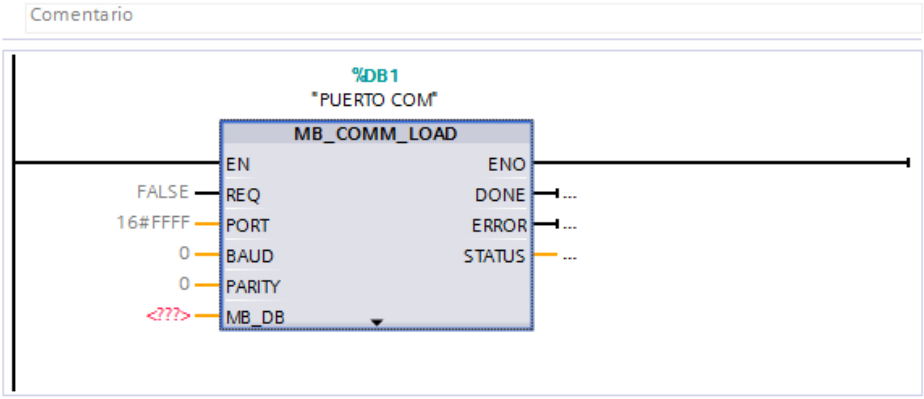


Figura 110: Instrucción PUERTO COM.

Una vez agregada la instrucción **MB_COMM_LOAD** explicaremos los parámetros que deben configurarse para la comunicación, en la tabla siguiente se detallará las funciones de los parámetros:

Tabla 7: Parámetros de la instrucción MB_COMM_LOAD [15].

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L	Ejecución de la instrucción con flanco ascendente.

PORT	Input	PORT	I, Q, M, D, L o constante	Identificación del puerto de comunicación: Tras haber insertado el módulo de comunicación en la configuración de dispositivos, la identificación de puerto aparece en la lista desplegable de la conexión del cuadro PORT. Esta constante también se puede referenciar en la tabla de variables de la ficha "Constantes".
BAUD	Input	UDINT	I, Q, M, D, L o constante	Selección de la velocidad de transferencia: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 Todos los demás valores no son válidos.
PARITY	Input	UINT	I, Q, M, D, L o constante	Selección de la paridad: • 0 – Ninguna • 1 – Impar • 2 – Par
FLOW_CTRL	Input	UINT	I, Q, M, D, L o constante	Selección del control de flujo: • 0 – (ajuste predeterminado) Sin control de flujo • 1 – Control de flujo por hardware con RTS siempre ON (no se aplica a los puertos RS485)
RTS_ON_DLY	Input	UINT	I, Q, M, D, L o constante	Selección de retardo a la conexión de RTS: • 0 – (ajuste predeterminado) Sin retardo de "RTS activo" hasta que se envía el primer carácter del aviso. • de 1 a 65535 – Retardo en milisegundos de "RTS activo" hasta que se envía el primer carácter del aviso (no se aplica a los puertos RS-485). Los retardos de RTS deben aplicarse independientemente de la selección de FLOW_CTRL.

RTS_OFF_DLY	Input	UINT	I, Q, M, D, L o constante	Selección de retardo a la desconexión de RTS: • 0 – (ajuste predeterminado) Sin retardo después de la transferencia del último carácter hasta "RTS inactivo" • de 1 a 65535 – Retardo en milisegundos desde la transferencia del último carácter hasta "RTS inactivo" (no se aplica a los puertos RS-485). Los retardos de RTS deben aplicarse independientemente de la selección de FLOW_CTRL.
RESP_TO	Input	UINT	I, Q, M, D, L o constante	Timeout de respuesta: Tiempo en milisegundos que MB_MASTER espera una respuesta del esclavo. Si el esclavo no responde a tiempo, MB_MASTER repite la solicitud o la finaliza con un error, siempre que se haya enviado el número de repeticiones indicado. de 5 ms a 65535 ms (estándar=1000 ms).
MB_DB	Input	MB_BASE	D	Una referencia al bloque de datos instancia de las instrucciones "MB_MASTER" o "MB_SLAVE". Después de haber insertado "MB_SLAVE" o "MB_MASTER" en el programa, la identificación de DB está disponible en la lista desplegable en la conexión del cuadro MB_DB.
DONE	Output	BOOL	I, Q, M, D, L	Ejecución de la instrucción finalizada sin errores.
ERROR	Output	BOOL	I, Q, M, D, L	Error: • 0 – No se ha detectado ningún error • 1 – Indica que se ha detectado un error. En el parámetro STATUS se devuelve un código de error.
STATUS	Output	WORD	I, Q, M, D, L	Código de error de la configuración del puerto

Nuestro bloque de carga para la comunicación MODBUS RTU quedará configurado de la siguiente manera:

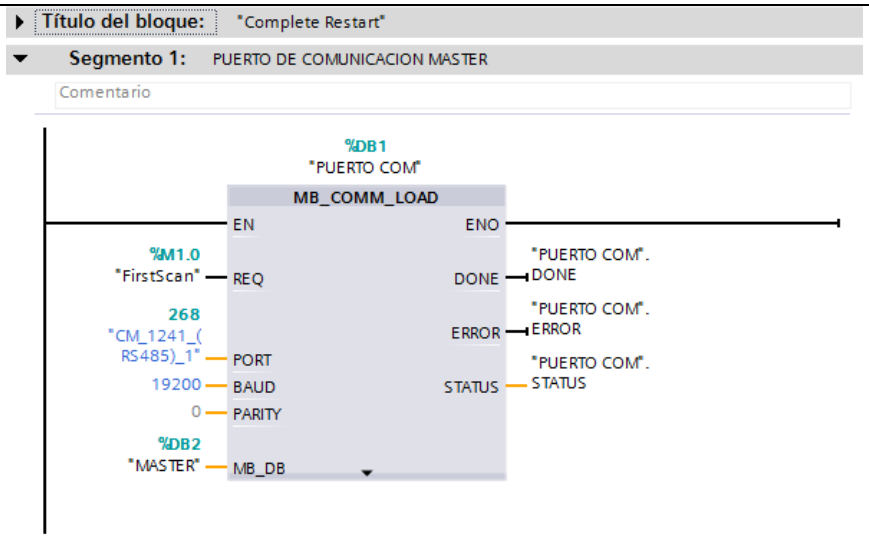


Figura 111: Configuración de instrucción PUERTO_COM MASTER.

El puerto COM debe estar configurado con los mismos datos tanto para los bloques del PLC MASTER como para los bloques del PLC SLAVE, es decir, deberán funcionar con la misma paridad, velocidad de transmisión y puerto de comunicación MOSBUS RTU.

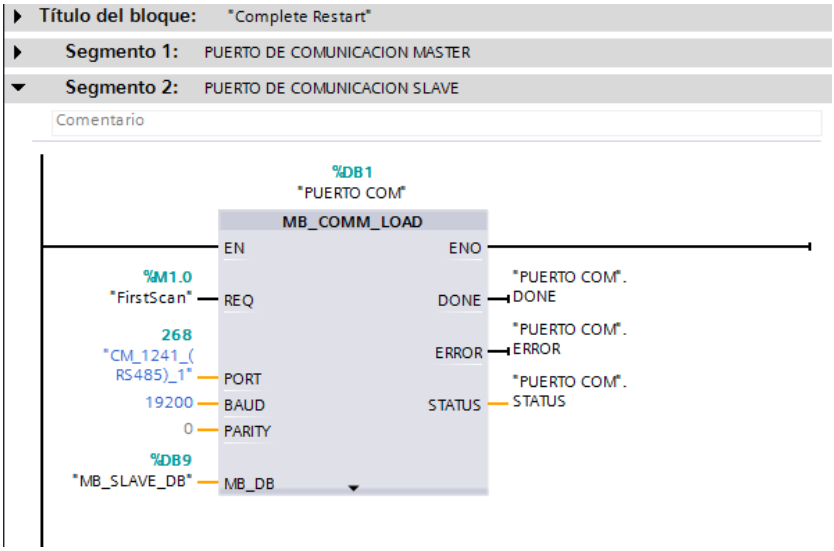


Figura 112: Configuración de instrucción PUERTO_COM SLAVE.

Observación: se debe repetir el paso 22 para el PLC SLAVE.

Paso 23.

Ahora vamos a agregar la instrucción **MASTER** el cual sirve para configurar al PLC como maestro, este bloque de comunicación lo vamos a crear en el **MAESTRO_OB1** para eso nos vamos a la pestaña **Instrucciones**, **Comunicación**, **Procesador de comunicaciones**, **MODBUS**, y damos doble clic en la instrucción **MB_MASTER** (ver figura 100).

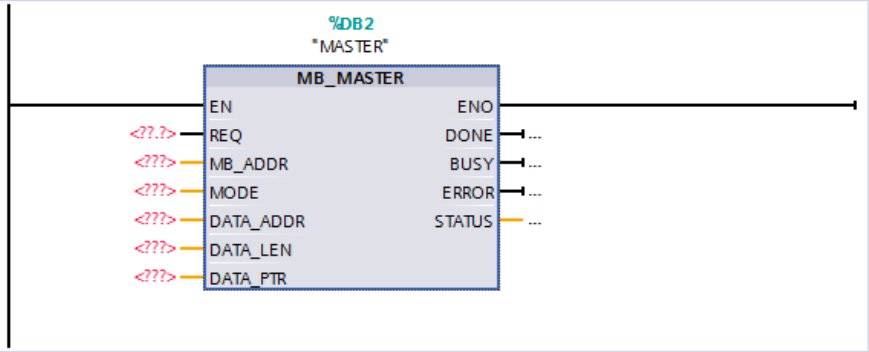


Figura 113: Instrucción MASTER.

Procederemos a configurar las respectivas instrucciones PUERTO_COM y MASTER. En nuestro programa crearemos un **Bloque de datos** que llamaremos **PARAMETERS** en el cual vamos a configurar cada uno de los parámetros o funciones de cada instrucción de comunicación, lo que incluye las instrucciones MB_COM_LOAD, MB_MASTER y MB_SLAVE. Nos dirigimos a **Árbol de proyecto, Tesis_V12_SP1, Master, Bloque de programas** y **Agregar nuevo bloque**. Seleccionamos **Bloque de datos** y en **Nombre** ponemos **PARAMETERS**.

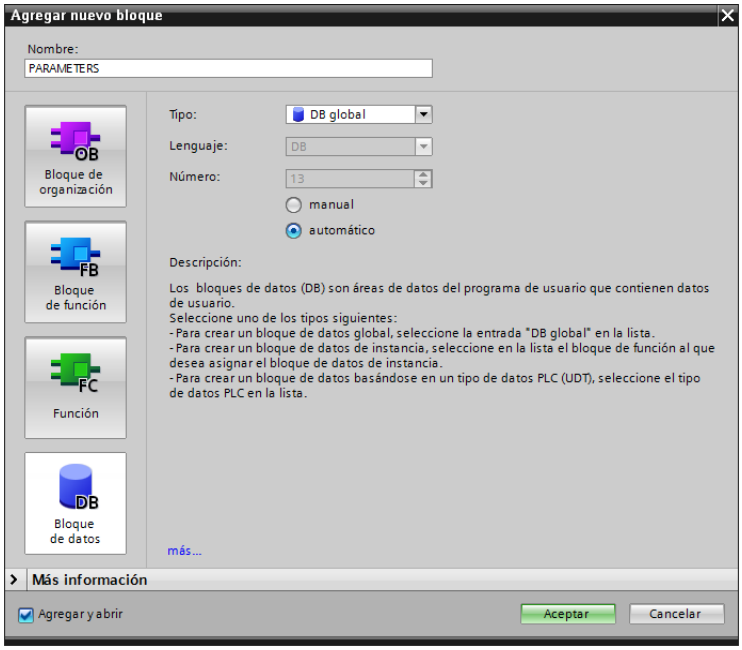


Figura 114: Creación del bloque de datos PARAMETERS.

Ahora crearemos las 5 variables que configuraran los puertos de comunicación, para esto insertamos una variable que llamaremos **P1** y le asignamos de tipo Struct, dentro de él generamos 5 variables, REQ, MB_ADRR, MODE, DATA_ ADDR, DATA_LEN, verificar la figura 102.

Figura 115: Variables de configuración del puerto MASTER práctica 1.

Nuestro programa contará con un sistema de habilitación el cual va a estar ligado a la lista de texto creada anteriormente en la imagen Proceso. Para esto crearemos variables de habilitación en cada práctica. Procedemos a crear un nuevo bloque de datos DB el cual llamaremos **CONFIGURACION** donde se hallarán todas las variables del proyecto.

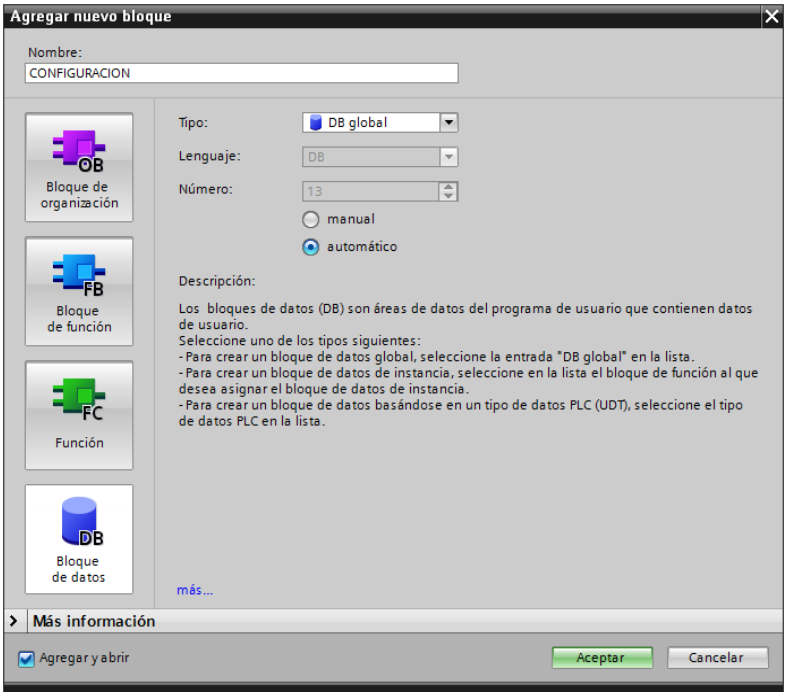


Figura 116: Creación del bloque de datos CONFIGURACION.

Dentro del bloque CONFIGURACION vamos a crear las variables que serán utilizadas por cada control en las pantallas, es decir, botones, selectores, marcas, etc. Procederemos a crear las variables para habilitación de las prácticas a través de la lista de texto:

Figura 117: Variables de habilitación.

Observación: se debe repetir el paso 23 para el PLC SLAVE.

Explicaremos a continuación la función de cada parámetro de la instrucción MASTER, para esto se detallará en la siguiente tabla:

Tabla 8: Parámetros de la instrucción MB_MASTER **[15]**.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L	Entrada de solicitud: • 0 – Ninguna solicitud • 1 – Solicitud de enviar datos a esclavo(s) Modbus
MB_ADDR	Input	UINT	I, Q, M, D, L o constante	Dirección de estación Modbus RTU: • Área de direccionamiento estándar: 0 a 247 • El valor "0" está reservado para la difusión general de un aviso a todos los esclavos Modbus. Para el broadcast sólo se soportan los códigos de función Modbus 05, 06, 15 y 16.
MODE	Input	USINT	I, Q, M, D, L o constante	Modo de selección: Indica el tipo de solicitud: lectura, escritura o diagnóstico En la tabla de funciones Modbus encontrará más detalles.
DATA_ADDR	Input	UDINT	I, Q, M, D, L o constante	Dirección inicial en el esclavo: indica la dirección inicial de los datos a los que se debe acceder en el esclavo Modbus. Las direcciones válidas se pueden encontrar en la tabla de funciones Modbus.
DATA_LEN	Input	UINT	I, Q, M, D, L o constante	Longitud de datos: indica el número de bits o palabras a los que debe acceder esta solicitud. Las longitudes válidas se pueden encontrar en la tabla de funciones Modbus.
DATA_PT	Input	VARIAN	M, D	Puntero hacia la dirección del DB o de la

R		T		marca de la CPU de los datos que se deben escribir o leer. En un DB, éste se debe crear con el tipo de acceso "Estándar: compatible con S7-300/400".
DONE	Output	BOOL	I, Q, M, D, L	• 0: Transacción no finalizada • 1: Transacción finalizada sin errores
BUSY	Output	BOOL	I, Q, M, D, L	• 0: No se está ejecutando ninguna transacción de MB_MASTER • 1: Transacción de MB_MASTE en ejecución
ERROR	Output	BOOL	I, Q, M, D, L	• 0: Ningún error • 1: Error; el código de error se visualiza en el parámetro STATUS
STATUS	Output	WORD	I, Q, M, D, L	Condición de ejecución

El bloque de instrucción **MASTER** quedará configurado en modo lectura, es decir que el **PLC MASTER** actuará como esclavo MODBUS y la PC actuará como maestro MODBUS. La PC utilizará el simulador Modbus Slave donde se configurará para escribir en el PLC. En la siguiente figura se muestra la configuración de la instrucción **MASTER** para modo lectura:

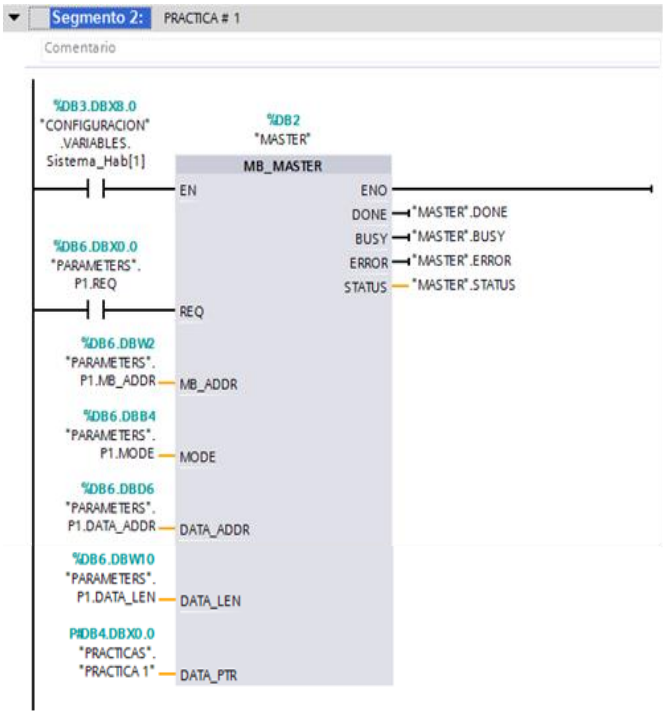


Figura 118: Configuración de instrucción MASTER.

Referente a la función MODBUS que vamos a designar para cada bloque se debe configurar el parámetro **DATA_ADDR** el cual detallaremos en la siguiente tabla las diversas funciones:

Tabla 9: Funciones de configuración para el DATA_ADDR **[15]**.

MODE	Función Modbus	Longitud de datos	Operación y datos	Dirección Modbus
0	01	de 1 a 2000 1 a 1992 (1)	Leer bits de salida: 1 a (1992 ó 2000) bits por consulta	de 1 a 9999
0	02	de 1 a 2000 1 a 1992 (1)	Leer bits de entrada: 1 a (1992 ó 2000) bits por consulta	10001 a 19999
0	03	de 1 a 125 1 a 124 (1)	Leer registro de parada: 1 a (124 ó 125) WORD por consulta	40001 a 49999 ó de 400001 a 465535
0	04	de 1 a 125 1 a 124 (1)	Leer WORD de entrada: 1 a (124 ó 125) WORD por consulta	30001 a 39999
1	05	1	Escribir un bit de salida: Un bit por consulta	de 1 a 9999
1	06	1	Escribir un registro de parada: 1 WORD por consulta	40001 a 49999 ó de 400001 a465535
1	15	de 2 a 1968 1 a 1960 (2)	Escribir varios bits de salida: 2 a (1960 ó 1968) bits por consulta	de 1 a 9999
1	16	de 2 a 123 1 a 122 (2)	Escribir varios registros de parada: 2 a (122 ó 123) WORD por consulta	40001 a 49999 ó de 400001 a 465535

Para el programa y las prácticas en general se necesita crear variables a las cuales se van a registrar los datos a recibir o enviar, la función **DATA_PRT** del bloque de instrucción **MASTER** guarda esos datos en las variables que nosotros designemos, para ello es necesario crear variables donde se guarden los datos enviados o a recibir. A continuación crearemos un Bloque de datos que denominaremos **PRACTICAS [DB4]** para guardar los 10 datos que usaremos en la práctica 1.

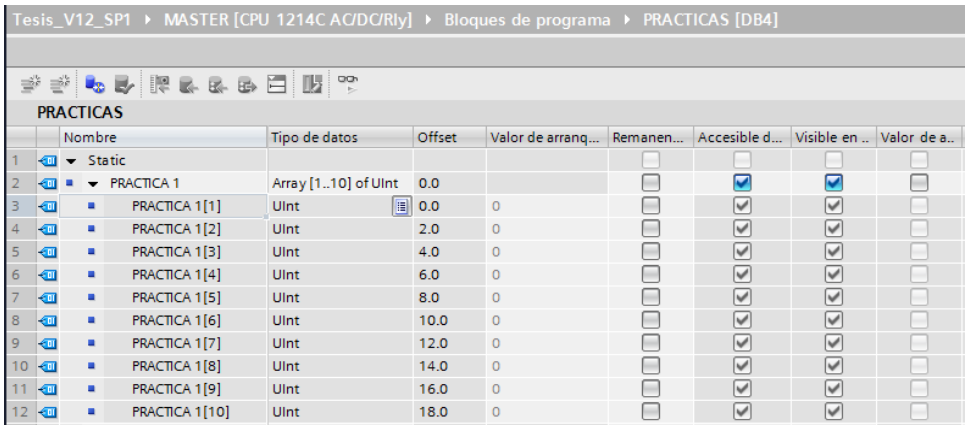


Figura 119: Creación del bloque de datos PRACTICAS – práctica 1.

Los bloques de comunicación necesitan una señal de reloj o retroalimentación en un lazo cerrado, esto se debe a que el sistema es cíclico. Con esta teoría diseñamos un pequeño control cíclico el cual consta de dos marcas y un temporizador, lo vamos a crear en el **Segmento 1** del bloque **MAESTRO [OB1]**:

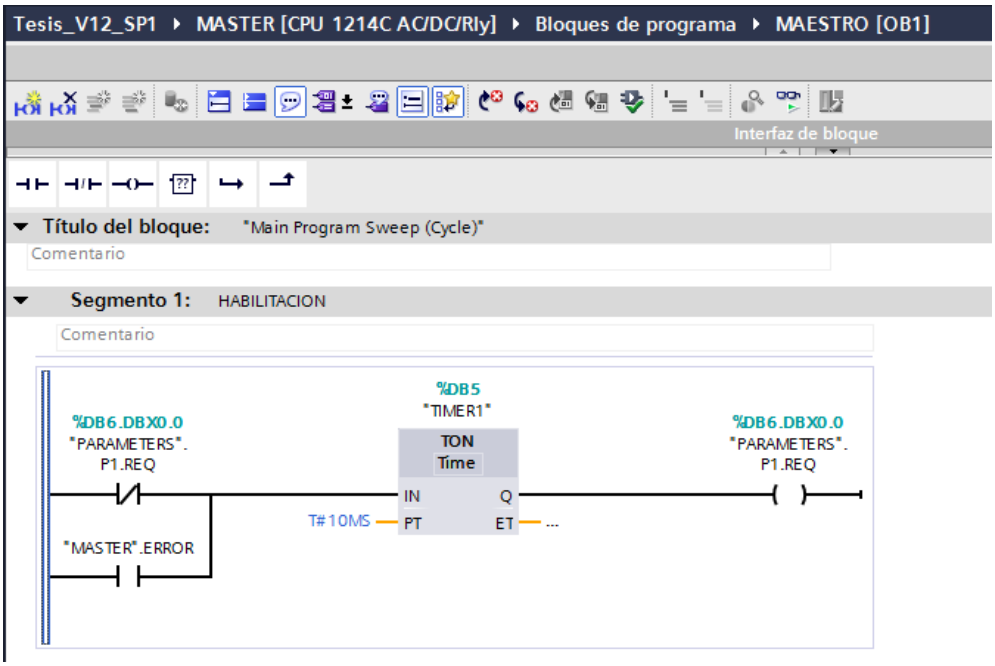


Figura 120: Señal de reloj para las instrucciones MODBUS.

Observación: se debe repetir el paso 23 para el PLC SLAVE.

Paso 24.

Tenemos listo el programa en el Tía Portal para lo cual el PLC MASTER y PLC SLAVE están configurados como esclavos MODBUS. Procedemos a ejecutar el simulador Modbus Slave y lo configuramos para modo escritura.

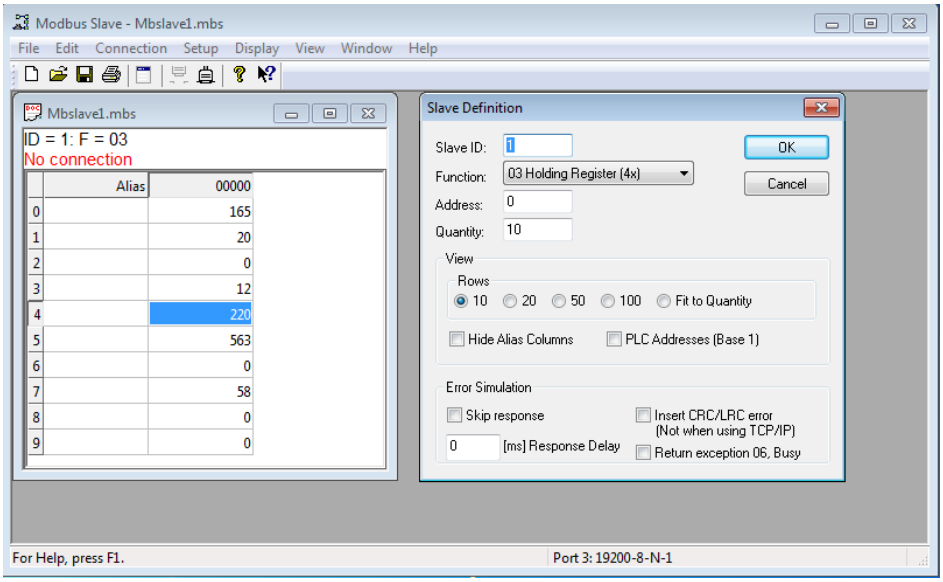


Figura 121: Configuración Modbus Slave.

Una vez realizada la configuración del simulador damos clic en **Connection**, **Connect.** Seleccionamos las siguientes opciones:

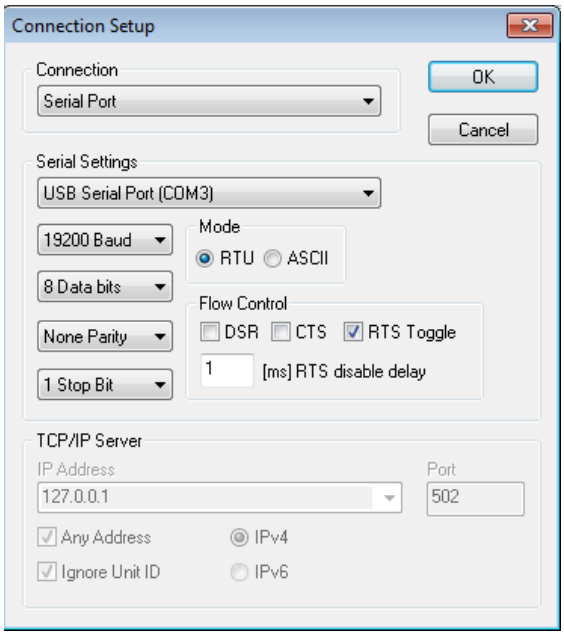


Figura 122: Configuración de la conexión.

Las configuraciones de la conexión como de los parámetros (maestro / esclavo) deben ser iguales que el programa Tía Portal.

Conectamos el cable driver USB – RS485 a la laptop, el puerto de comunicación se llama por defecto **USB Serial Port (COM 3)**. El otro extremo del cable lo conectamos en el puerto de comunicación **DB9-X2** del módulo del PLC **MASTER**.

Finalmente enlazamos el simulador con el PLC donde desde el simulador escribimos números enteros y se visualizarán en la pantalla HMI, así mismo se registraran los datos enviados en el PLC.

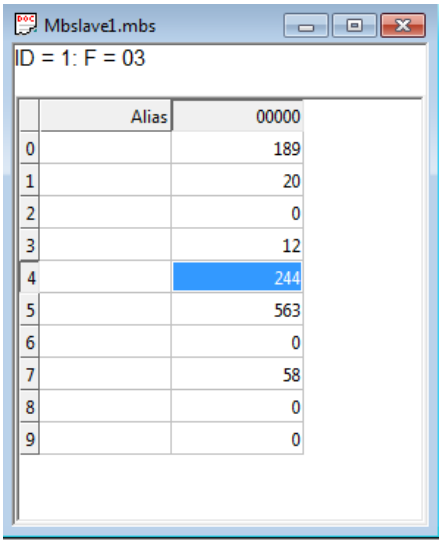


Figura 123: Datos enviados desde el simulador.

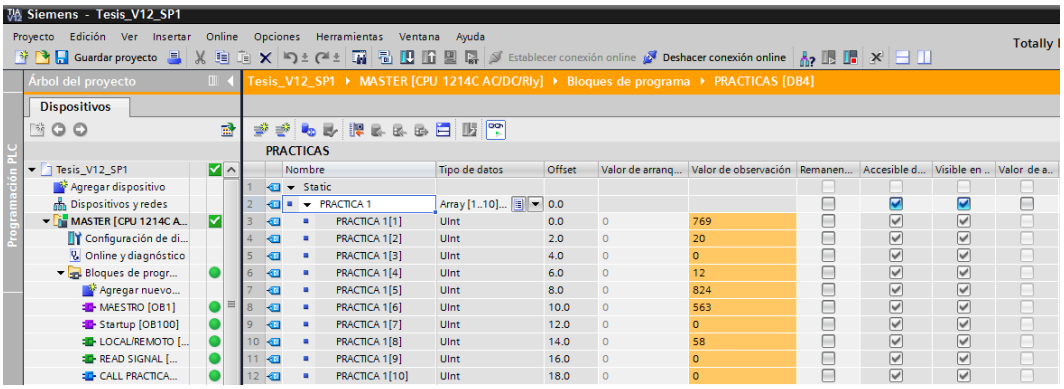


Figura 124: Datos recibidos en el PLC.

Recomendaciones:

- Configurar los puertos MODBUS correctamente para lo que se requiera elaborar, ya sea lectura o escritura.
- Revisar el tipo de dato a enviar, así mismo como la cantidad y longitud.
- Conectar los cables RS485 en el puerto correspondiente.
- Verificar la práctica con ambos módulos PLC.

4.2. Práctica Nº 2

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización I</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>2</td><td>TÍTULO PRÁCTICA: “Conexión Half-Duplex entre 2 autómatas S7-1200.”</td></tr>
<tr><td colspan="4">OBJETIVOS:

• Interconectar 2 módulos PLC vía MODBUS RTU.
• Conocer el funcionamiento maestro – esclavo entre 2 PLC S7-1200.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Instalar correctamente los softwares a utilizar con sus respectivos drivers.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:

• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• Dos módulos de comunicación CM1241.
• Un cable de comunicación RS485 macho - macho.

DESCRIPCIÓN:

Esta práctica deberá comunicar a dos PLC S7-1200 vía MODBUS RTU a través de un cable RS485 macho - macho, cada PLC será capaz tanto de enviar como de recibir datos pero uno a la vez. Los resultados se mostrarán en las pantallas HMI de cada módulo. El sistema contará con dos bloques de instrucción MB_MASTER los cuales se programarán en el PLC MASTER, uno servirá para escribir datos y el otro para leer. Así mismo contará con un bloque MB_SLAVE el cual se programará en el PLC SLAVE para recibir los datos enviados del MB_MASTER y almacenarlos con dos bloques MOVE en la variable PRACTICA 2. De esta manera se enviaran y recibirán datos desde ambos módulos didácticos.

La figura 125 muestra la arquitectura de red a implementar.</td></tr>
</table>

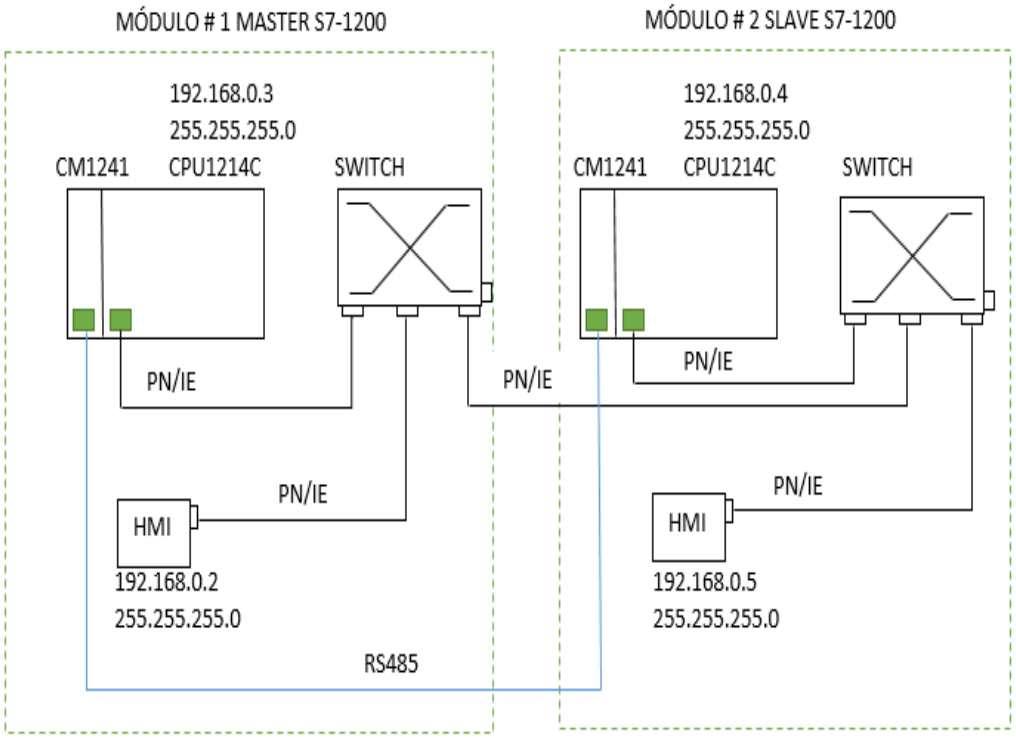


Figura 125: Arquitectura de red.

En figura 126 se muestra el diagrama de conexiones a implementar.

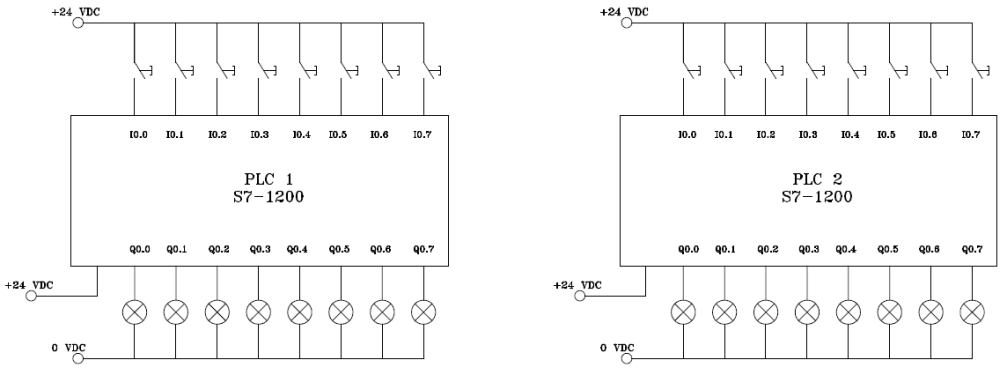


Figura 126: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear los parámetros para las instrucciones de la práctica 2, por consiguiente nos dirigimos al DB **PARAMETERS [DB6]** ubicado en el PLC **MASTER** e ingresamos las siguientes variables:

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB6]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	P1	Struct	0.0					
3	P2	Struct	12.0					
4	MB_ADDR	UInt	0.0	2				
5	MODE	USInt	2.0	1				
6	DATA_ADDR	UDInt	4.0	40001				
7	DATA_LEN	UInt	8.0	2				
8	MB_ADDR SLAVE	UInt	10.0	2				
9	MODE SLAVE	USInt	12.0	0				
10	DATA_ADDR SLAVE	UDInt	14.0	40003				
11	DATA_LEN SLAVE	UInt	18.0	2				

Figura 127: Parámetros de la práctica 2.

Adicionalmente tenemos que crear las variables donde se van a registrar los datos, para esto nos dirigimos hacia PRACTICAS [DB4]. La imagen a continuación mostrará las variables de registro de datos para el PLC **MASTER**.

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	ENVIAR 1	Array [1..2] of Int	0.0					
5	ENVIAR 1[1]	Int	0.0	0				
6	ENVIAR 1[2]	Int	2.0	0				
7	RECIBIR 1	Array [1..2] of Int	4.0					
8	RECIBIR 1[1]	Int	0.0	0				
9	RECIBIR 1[2]	Int	2.0	0				

Figura 128: Variables de registro del PLC MASTER – práctica 2.

De la misma forma generamos las variables para el PLC SLAVE.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	RECIBIR 1	Int	0.0	0				
5	RECIBIR 2	Int	2.0	0				
6	ENVIAR 1	Int	4.0	0				
7	ENVIAR 2	Int	6.0	0				

Figura 129: Variables de registro del PLC MASTER – práctica 2.

Paso 2.

Procedemos a agregar los bloques de instrucción MB_MASTER en el PLC MASTER, para esto nos dirigimos dentro del **MAESTRO [OB1]** y creamos el **Segmento 3** perteneciente a la PRACTICA 2.

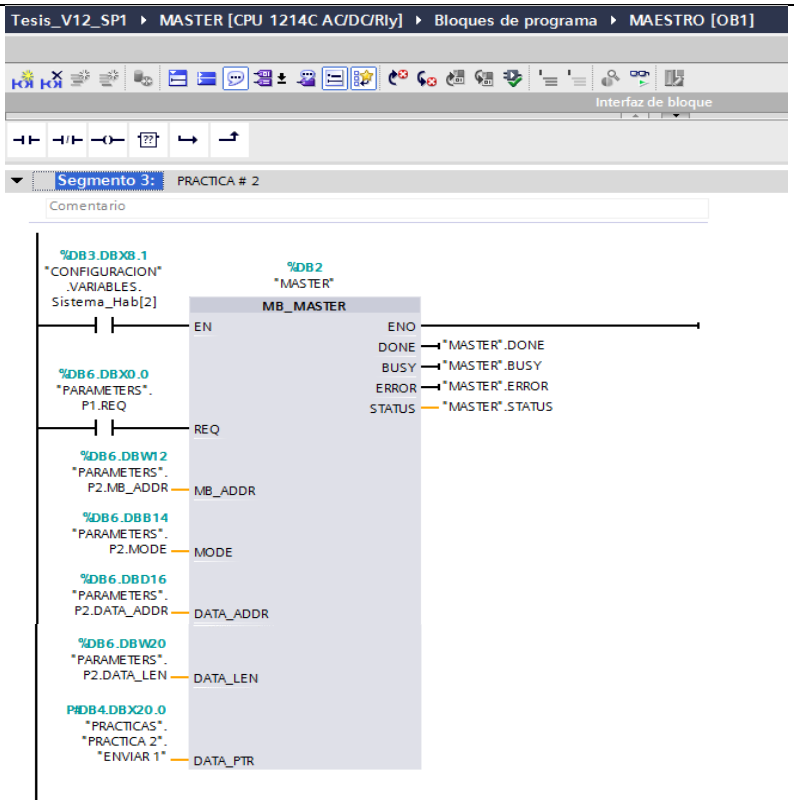


Figura 130: Configuración del bloque de escritura práctica 2.

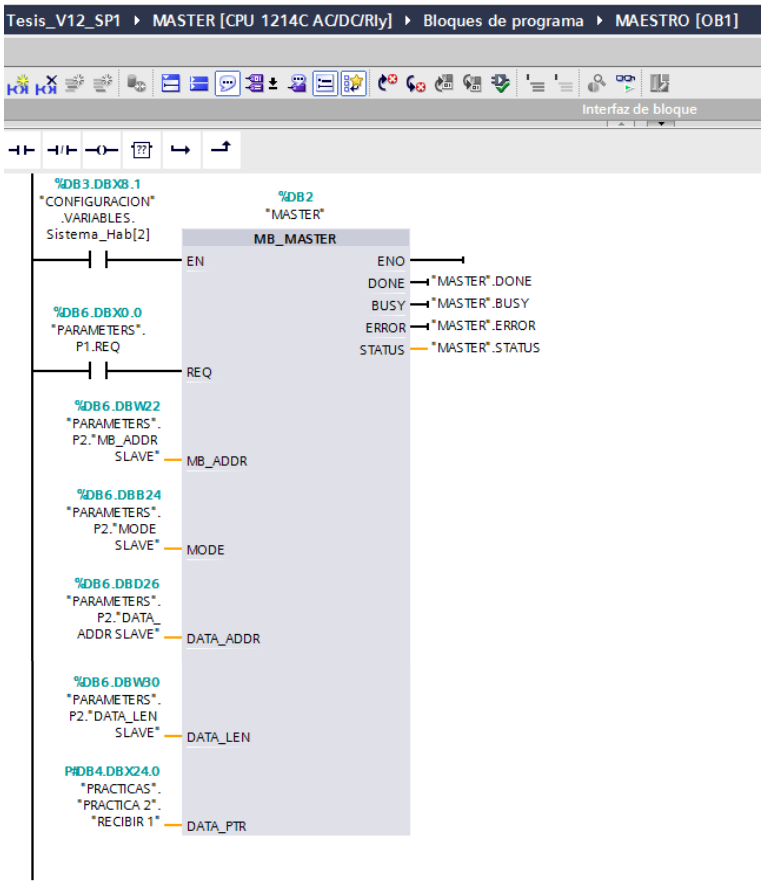


Figura 131: Configuración del bloque de lectura práctica 2.

Paso 3.

Ahora programaremos el bloque MB_SLAVE_DB para que se complete la comunicación. Nos vamos a **PLC SLAVE**, **Bloques de programa**, **Esclavo [OB1]** y creamos el **Segmento 3** donde le asignaremos el comentario PRACTICA # 2:

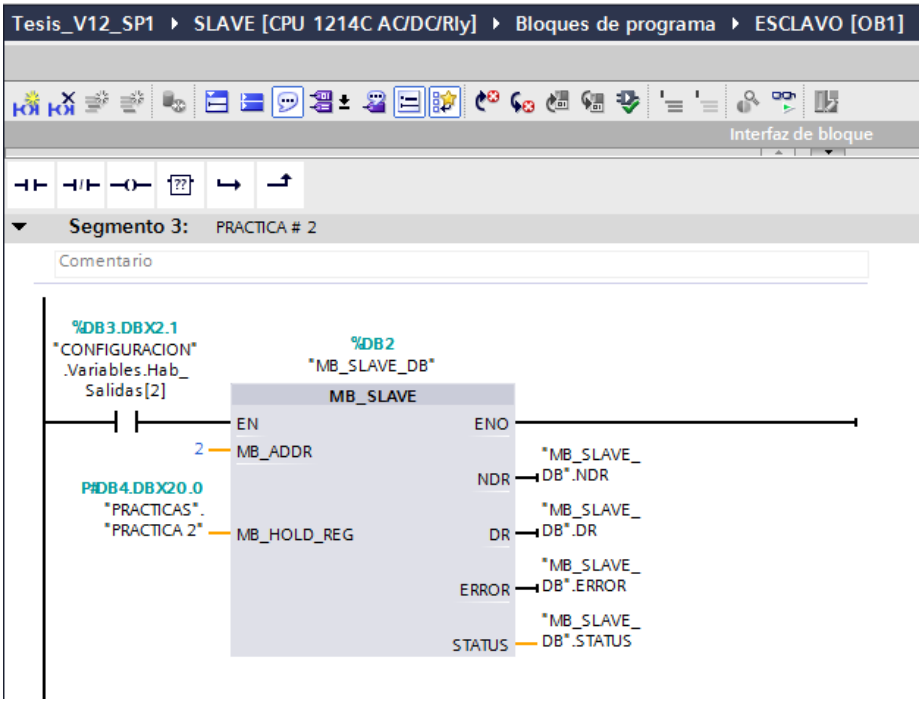


Figura 132: Configuración del bloque SLAVE práctica 2.

Paso 4.

Con dos bloques MOVE vamos a copiar el dato ingresado en los controles del HMI SLAVE y los vamos a guardar en las variables creadas para el bloque master de lectura, como observamos en la figura 118.

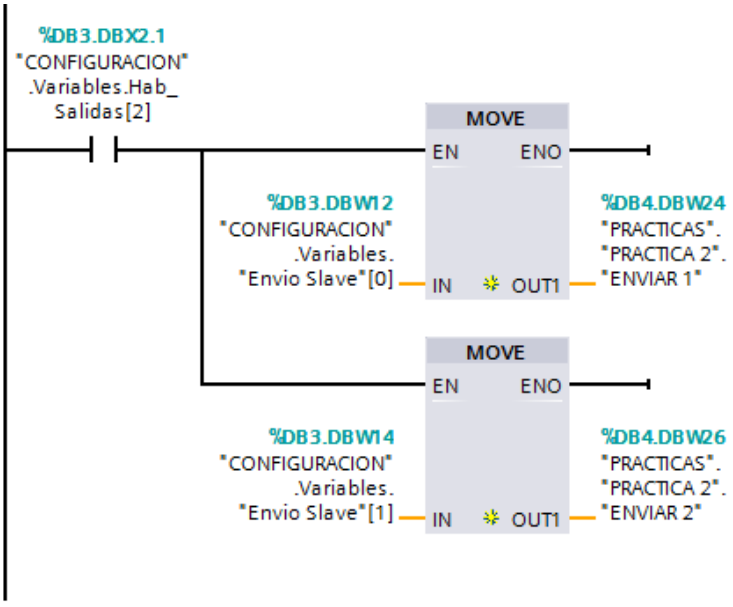


Figura 133: Bloques MOVE de la práctica 2.

Paso 5.

Ahora daremos paso al diseño de las imágenes del HMI, a continuación se mostrará la pantalla **Titulo 2** la misma que es la presentación de la práctica 2.

Figura 134: Creación de la imagen Titulo 2.

Observación: para la creación de la imagen **Titulo 2** se debe seguir el proceso del paso 19, a diferencia de escoger las variables correspondientes de la práctica 1.

La imagen **Practica 2** constará de los siguientes controles:

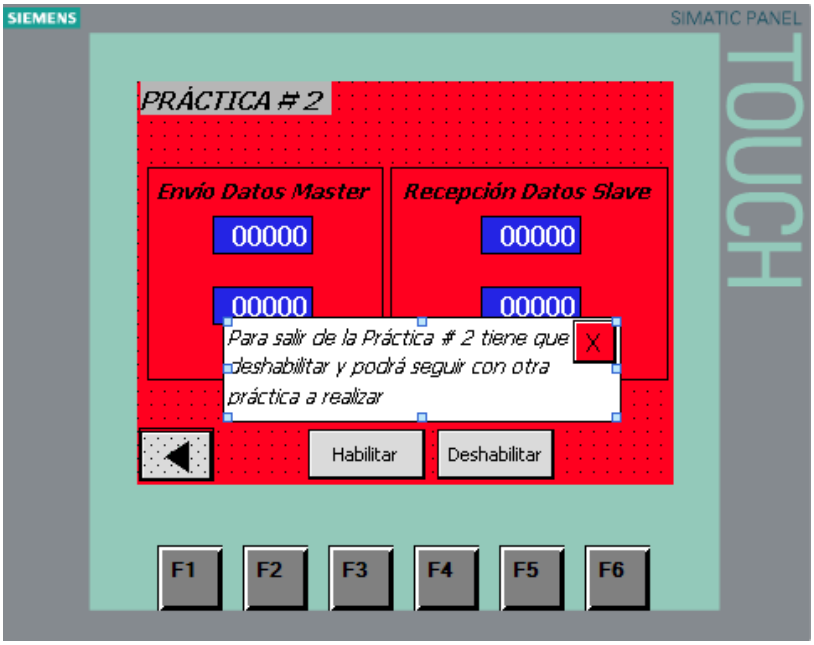


Figura 135: Diseño de la imagen Practica 2.

La asignación de variables para los campos de entrada, tanto de envío como recepción de datos queda de la siguiente manera:

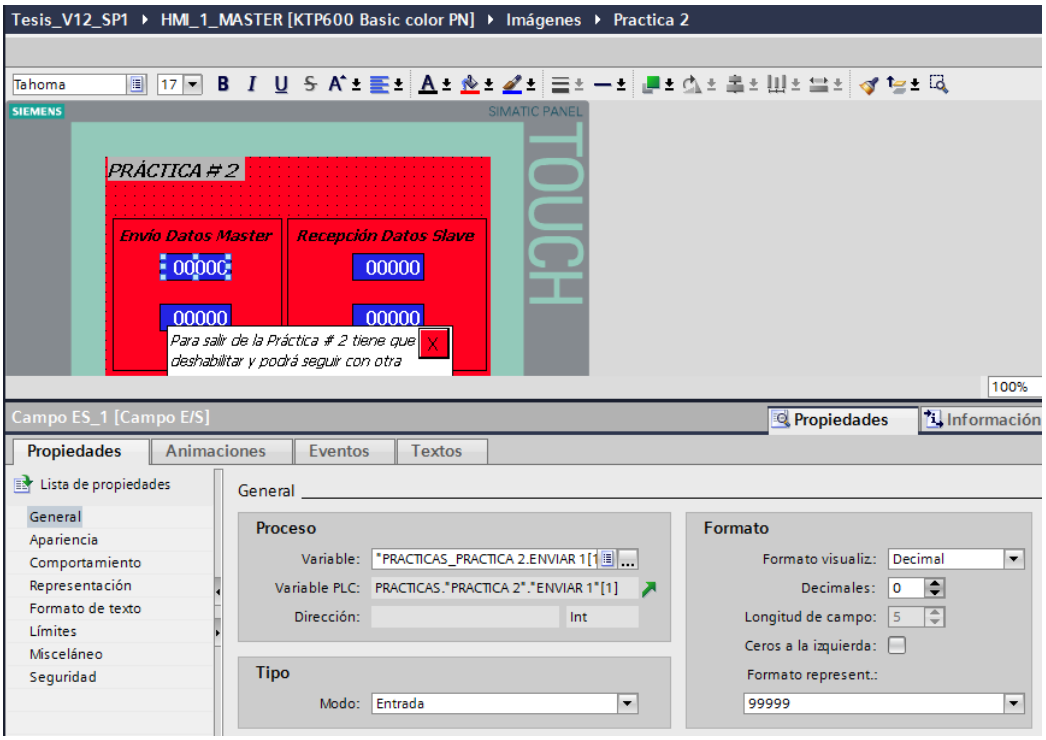


Figura 136: Propiedades de la entrada ES_1.

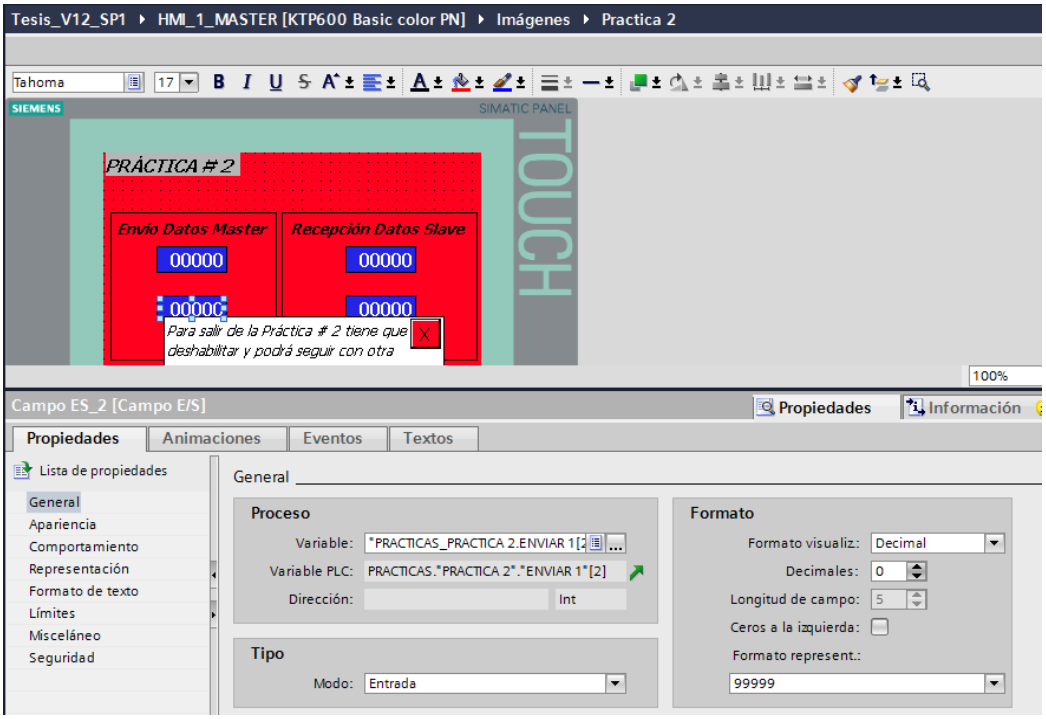


Figura 137: Propiedades de la entrada ES_2.

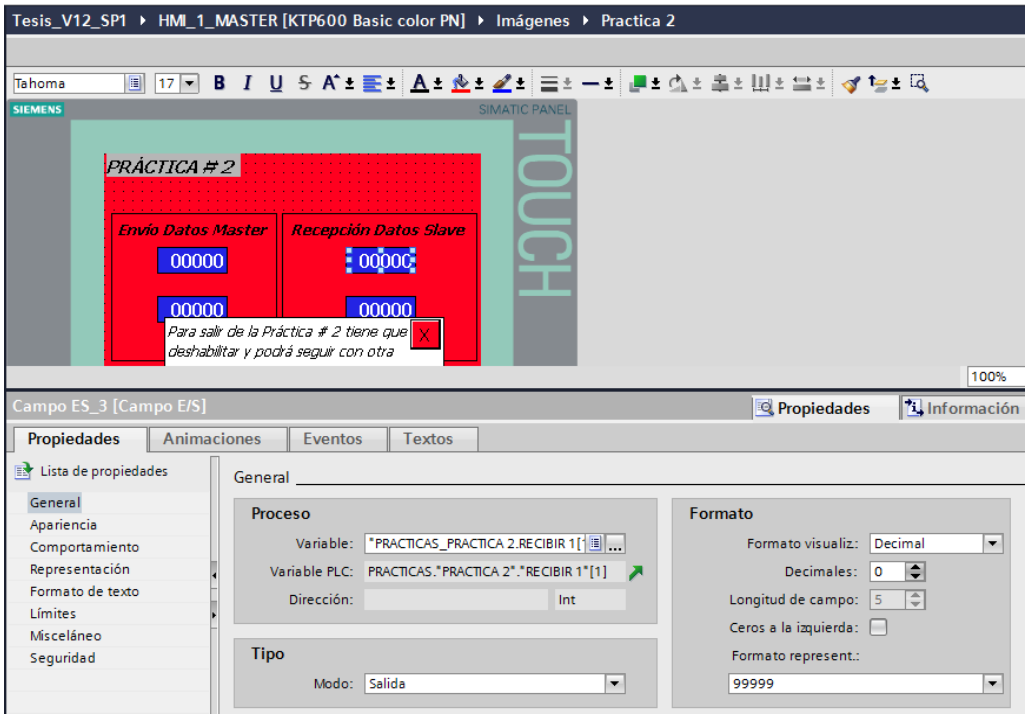


Figura 138: Propiedades de la entrada ES_3.

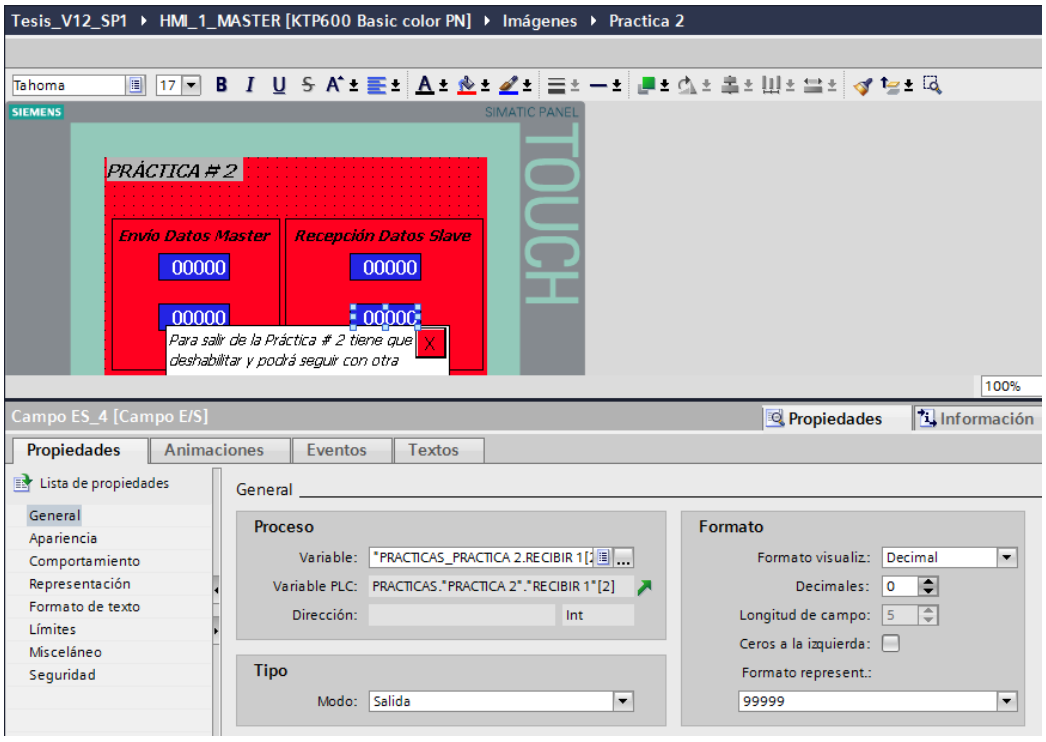


Figura 139: Propiedades de la entrada ES_4.

Respecto a los botones **Habilitar**, **Deshabilitar**, y el **mensaje de alerta** se debe seguir las instrucciones del **Paso 20** de la práctica 1.

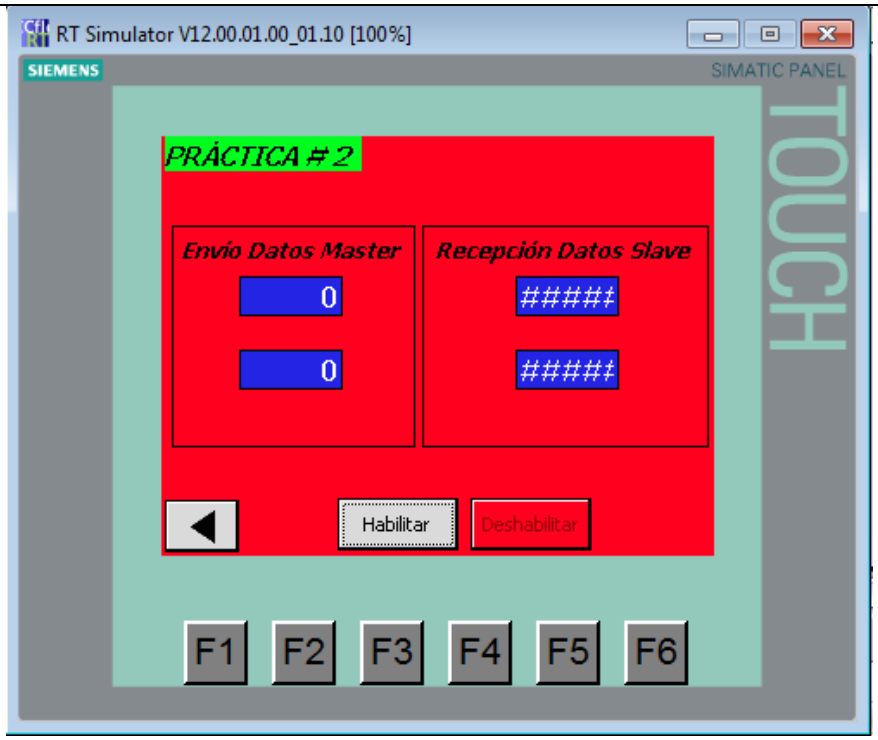


Figura 140: Diseño simulado de la práctica 2.

Observación: se debe implementar el mismo diseño de las imágenes HMI para el PLC SLAVE asignando las variables correspondientes al mismo.

Recomendaciones:

- Revisar el tipo de dato a enviar, así mismo como la cantidad y longitud.
- Comprobar que las variables de los controles del HMI estén correctamente asignadas.
- Registrar los parámetros de los bloques de instrucciones de comunicación de manera que se puedan escribir y leer datos de tipo entero.

4.3. Práctica Nº3

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>3</td><td>TÍTULO PRÁCTICA: “Manejo de un sistema de marcha y paro desde dos módulos S7 1200 conectados vía HMI.”</td></tr>
<tr><td colspan="4">OBJETIVOS:
Diseñar un sistema marcha y paro entre 2 PCL S7-1200.
Implementar el sistema vía MODBUS RTU.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:
Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
Dos módulos de comunicación CM1241.
Un cable de comunicación RS485 macho - macho.
Dos botones simulados en el HMI, uno para activar el motor del caldero y otro para arrancar el motor del silo.
Una luz piloto en cada módulo: Start (Q 0.0).
DESCRIPCIÓN:
Este sistema funciona de la siguiente manera: primero se debe habilitar la comunicación en ambos módulos con el botón Habilitar. Luego se debe configurar un PLC como maestro y el otro como esclavo, para esto se debe seleccionar con un botón en la pantalla HMI de cada módulo, que función quiere cumplir Master o Slave. Una vez configurados los módulos se enviará a activar la salida Q0.0 del módulo Slave, es decir al presionar el botón Start en el módulo Master se activará la luz piloto del módulo Slave y viceversa. El botón Stop apagará la luz piloto del Slave. El sistema no permitirá la transferencia de datos si ambos PLC son maestros o son esclavos al mismo tiempo, solamente funcionará cuando uno de los dos sea distinto del otro.

La figura 141 muestra la arquitectura de red a implementar.</td></tr>
</table>

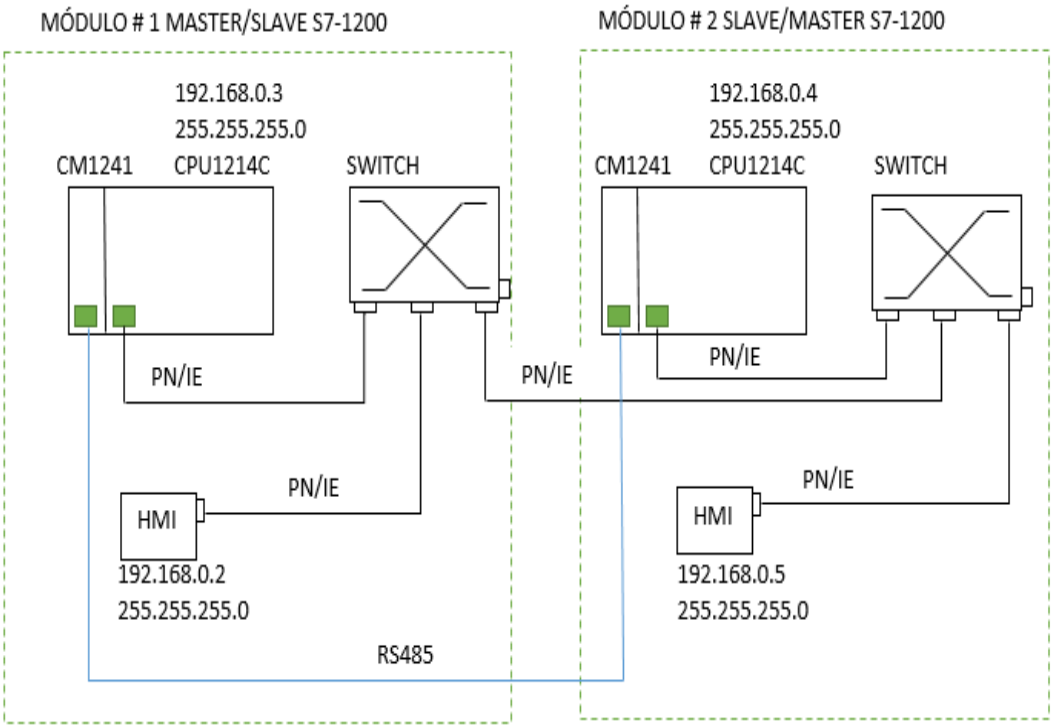


Figura 141: Arquitectura de red.

En figura 142 se muestra el diagrama de conexiones a implementar.

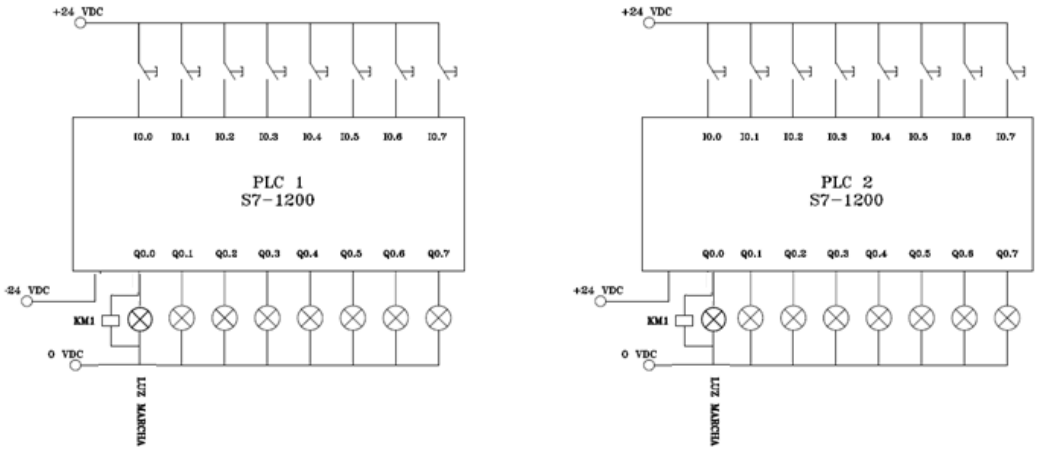


Figura 142: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear los parámetros para las instrucciones de la práctica 3, nos dirigimos a **PARAMETERS [DB6]** ubicado en el PLC **MASTER** y creamos las siguientes variables:

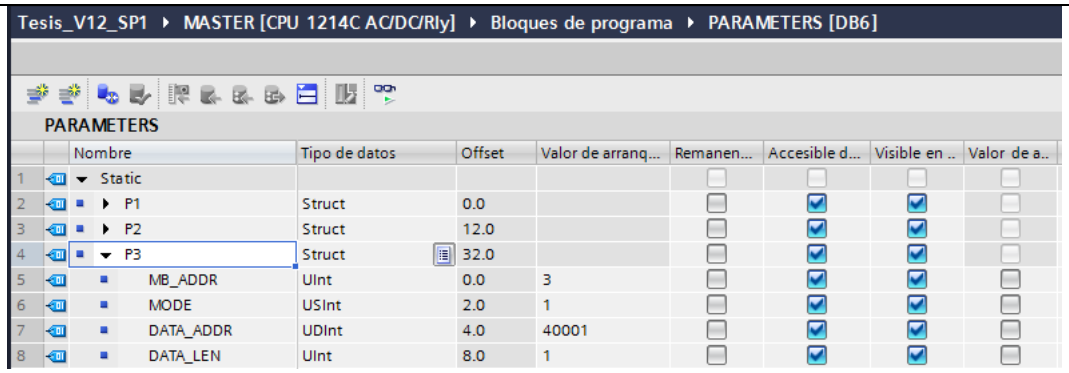


Figura 143: Parámetros de la práctica 3 MASTER.

A continuación realizamos los mismos pasos para el PLC **SLAVE**.

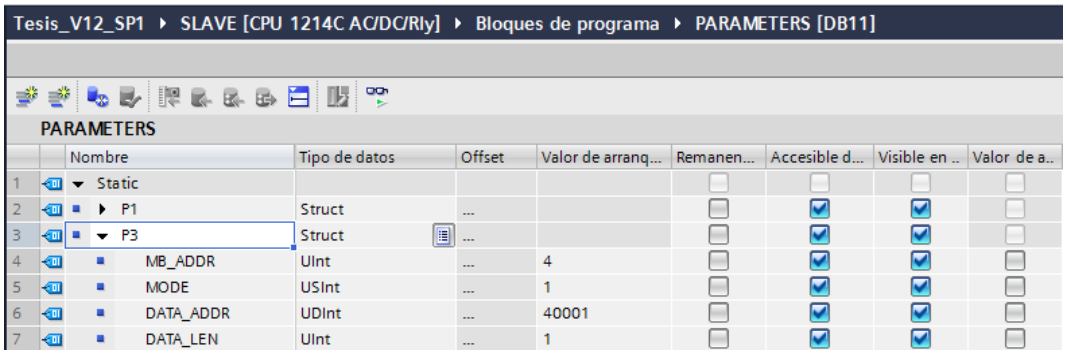


Figura 144: Parámetros de la práctica 3 SLAVE.

Los parámetros entre ambos bloques master es el mismo a diferencia del parámetro **MB_ADDR** el cual debe ser distinto del otro bloque master ya que son dos PLC a utilizar.

Paso 2.

Debemos guardar los datos de envío como recepción, por esto vamos a generar las variables en **PRACTICAS [DB4].** La imagen a continuación mostrará las variables de registro de datos para el PLC **MASTER**.

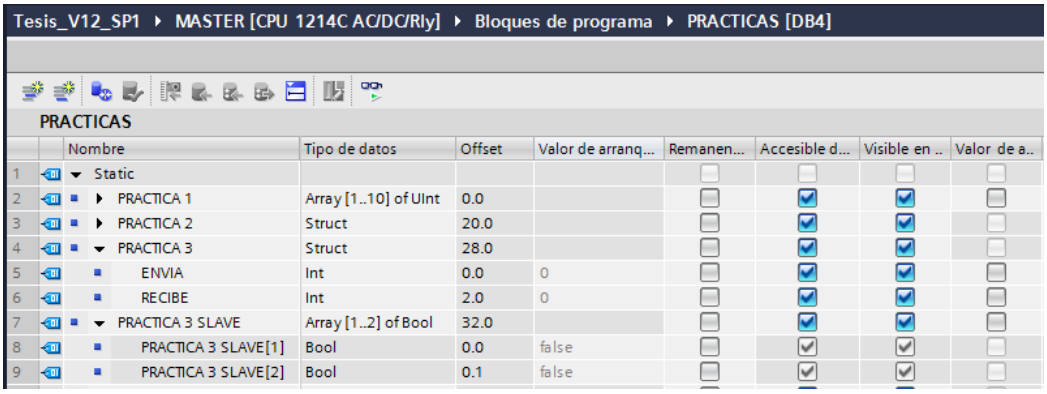


Figura 145: Variables de registro para el PLC MASTER – práctica 3.

Seguimos con la creación de las variables de registro para el PLC **SLAVE**.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	PRACTICA 3	Struct	28.0					
5	RECIBE	Int	0.0	0				
6	ENVIAR	Int	2.0	0				
7	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
8	PRACTICA 3 SLAVE[1]	Bool	0.0	false				
9	PRACTICA 3 SLAVE[2]	Bool	0.1	false				

Figura 146: Variables de registro para el PLC SLAVE – práctica 3.

Paso 3.

Procedemos a agregar los bloques de instrucción MB_MASTER en el PLC **MASTER**, para esto nos dirigimos al **MAESTRO [OB1]** y creamos el **Segmento 4** perteneciente a la PRACTICA 3. Se necesita configurar el bloque MB_MASTER en modo escritura para enviar el dato ingresado hasta el bloque slave del PLC SLAVE.

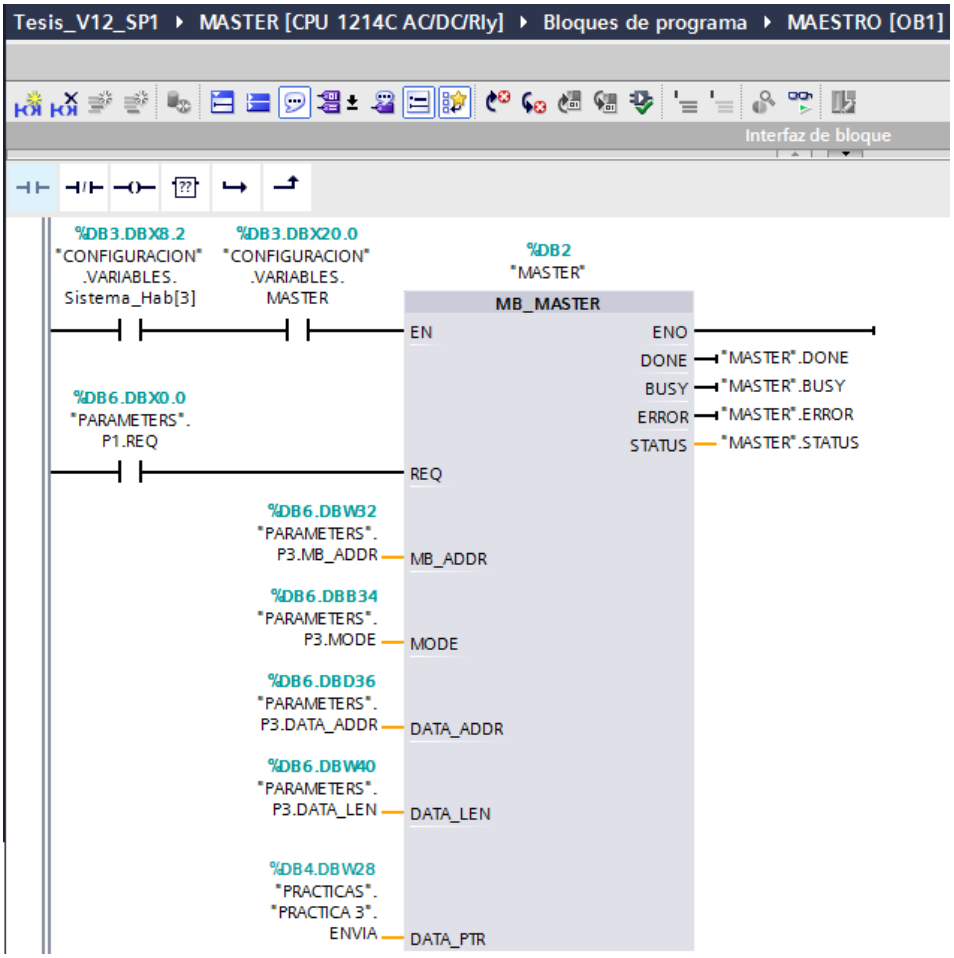


Figura 147: Configuración del bloque MB_MASTER/ PLC MASTER práctica 3.

Ahora agregaremos un bloque de instrucción SLAVE el cual receptará los datos enviados por el MB_MASTER del PLC SLAVE, la configuración es la siguiente:

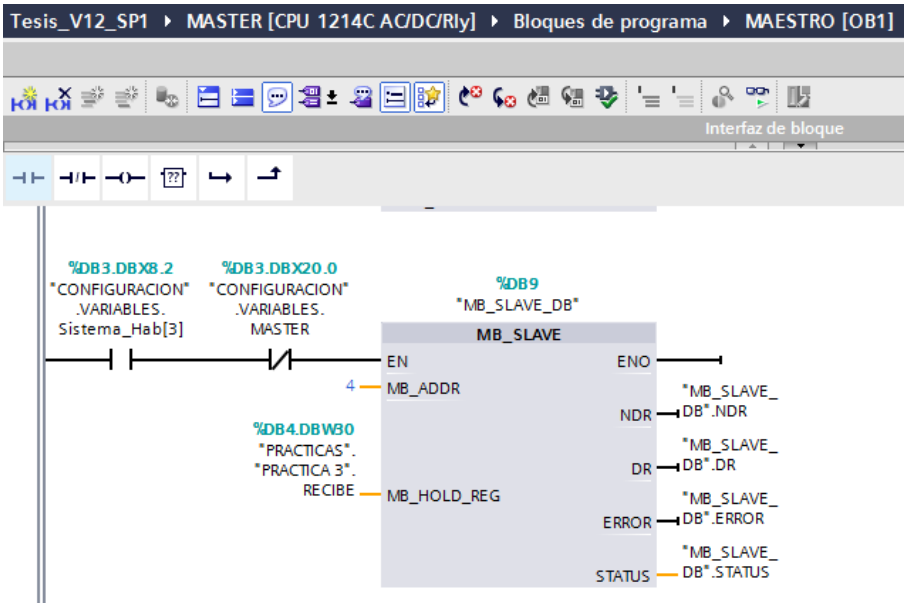


Figura 148: Configuración del bloque MB_SLAVE_DB/ PLC MASTER práctica 3.

Paso 4.

Ya creados los bloques de comunicación del PLC **MASTER** vamos a generar los bloques del PLC **SLAVE**. Nos vamos al **ESCLAVO [OB1]** e insertamos un bloque **MB_SLAVE_DB** el cual le asignamos las variables correspondientes:

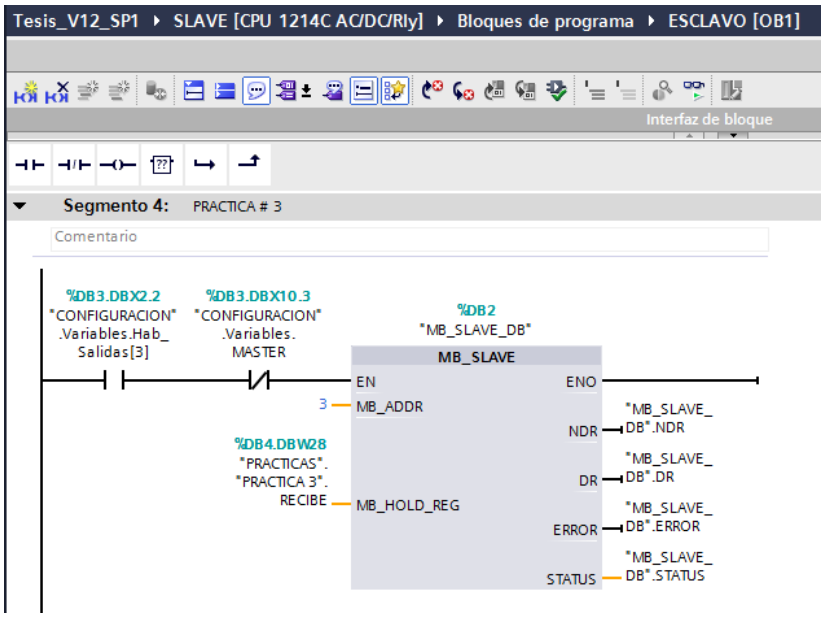


Figura 149: Configuración del bloque MB_SLAVE_DB/ PLC SLAVE práctica 3

Para terminar la etapa de comunicación insertamos un bloque **MB_MASTER** el cual es el encargado de escribir hacia el otro PLC.

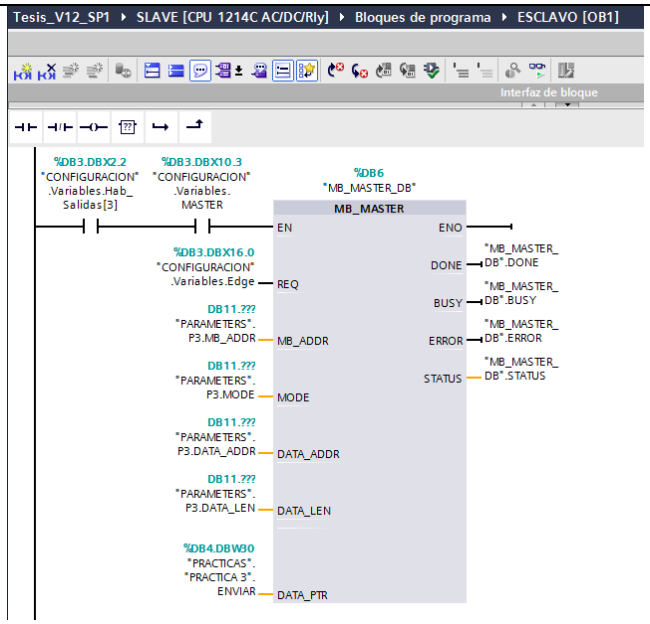


Figura 150: Configuración del bloque MB_MASTER/ PLC SLAVE práctica 3.

Paso 5.

Ahora elaboraremos la parte de control del PLC **MASTER** y **SLAVE**, para ellos hemos utilizado contactos abiertos / cerrados y marcas los cuales se ejecutaran como bloqueos, habilitaciones y acciones. Nuestra salida **Q0.0** va a reflejar la marcha en el PLC contrario donde se haya ejecutado la acción.

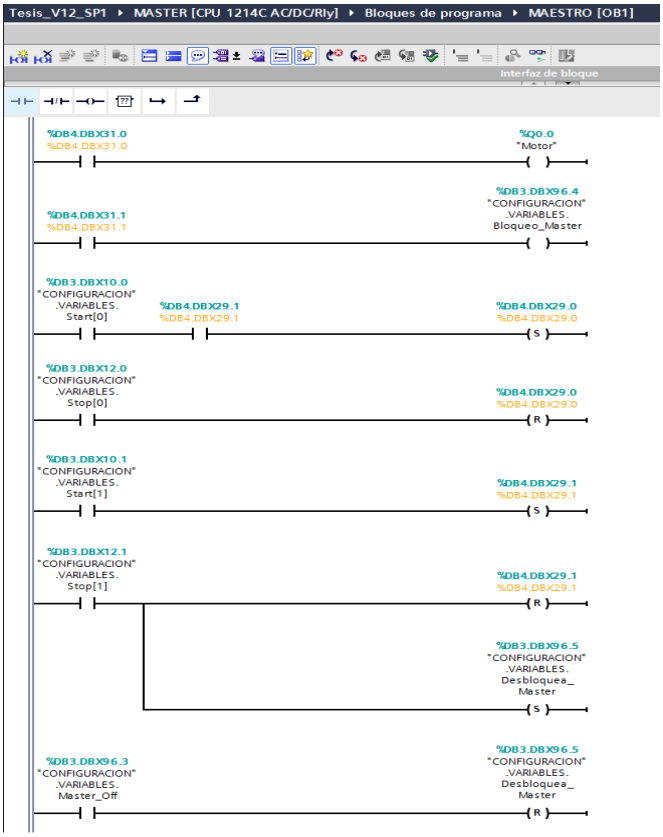


Figura 151: Diseño de circuito de control para el PLC MASTER.

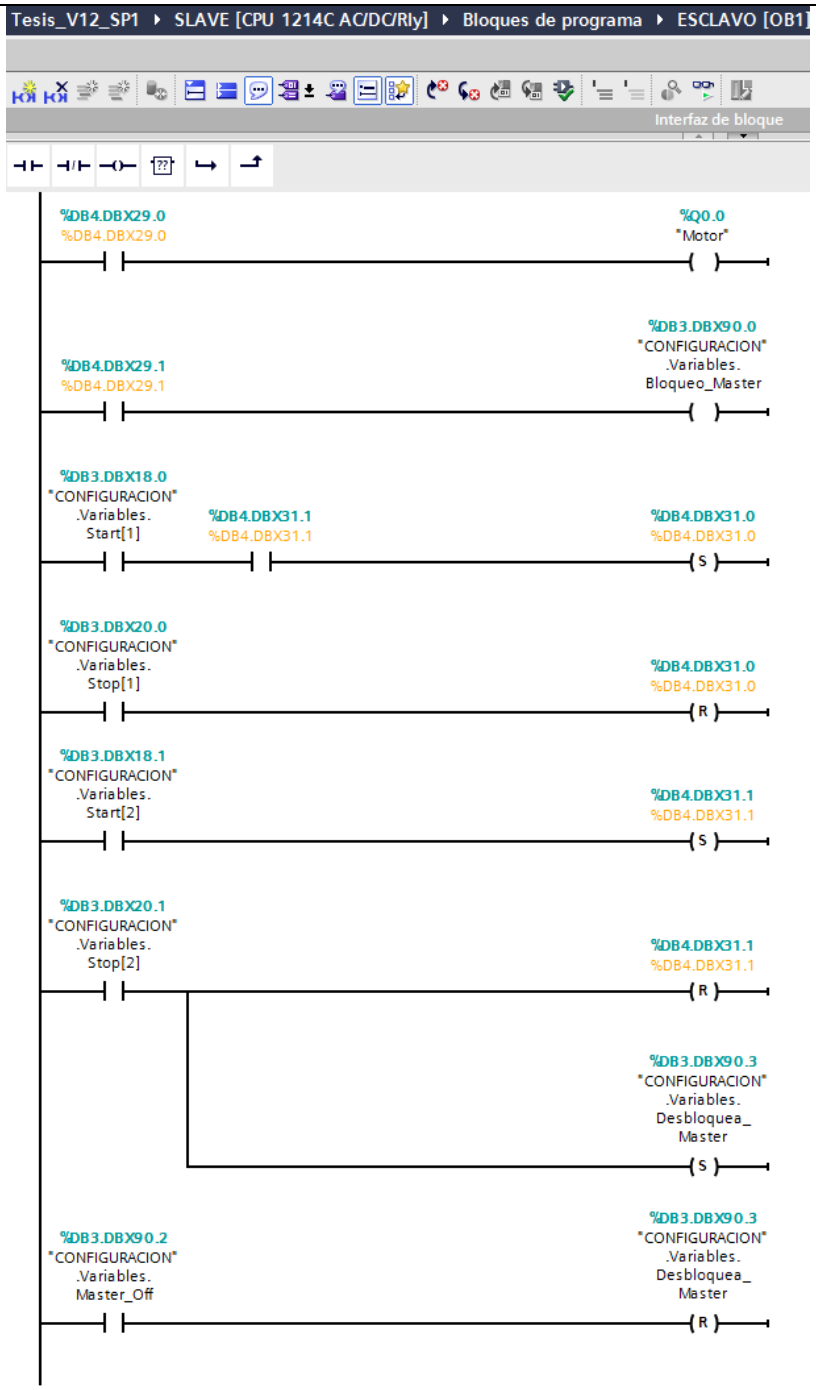


Figura 152: Diseño de circuito de control para el PLC SLAVE.

Paso 6.

Ahora procederemos con la creación de las imágenes para los HMI. Comenzaremos por el diseño de la imagen de presentación de la práctica 3 donde constara de una imagen con el logotipo de la Universidad Politécnica Salesiana, tres cuadros de texto con el número de la práctica, y el nombre de la práctica en dos filas consecutivas. También contará con un cuadro de diálogo con dos botones, uno para ingresar a la práctica y otra para salir al menú de selección.

Figura 153: Creación de la imagen Titulo 3.

Observación: para la creación de la imagen **Titulo 3** se debe seguir el proceso del paso 19 de la práctica 1, a diferencia de escoger las variables correspondientes de la práctica 3.
Se deberá generar una imagen de presentación para el PLC SLAVE con el mismo diseño.

La imagen **Practica 3** constará de los siguientes controles:

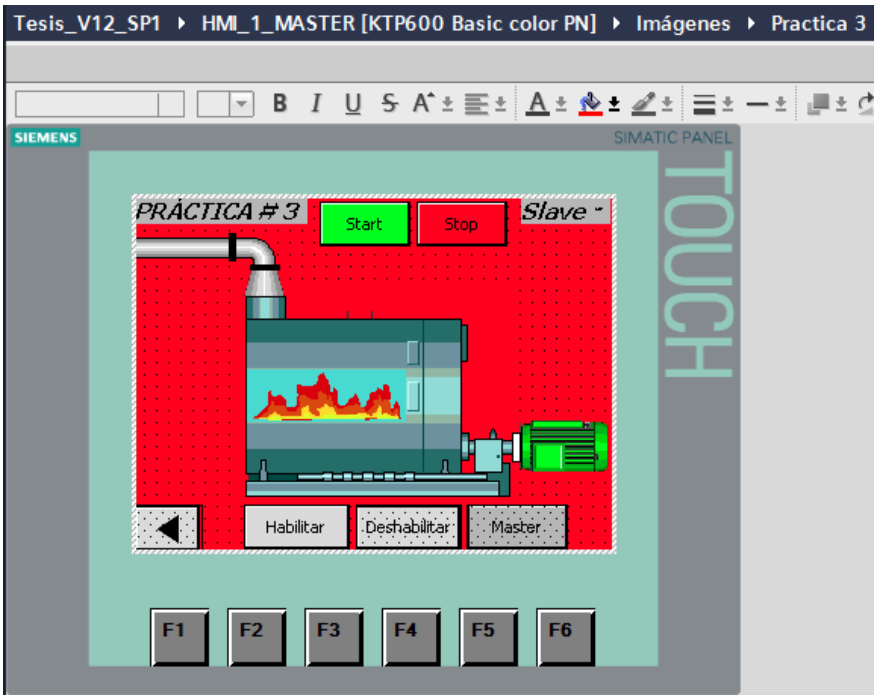


Figura 154: Diseño de imagen Practica 3.

Para la animación de la caldera se utilizó 5 imágenes que luego fueron agrupadas, las mismas que podemos evidenciar la configuración:

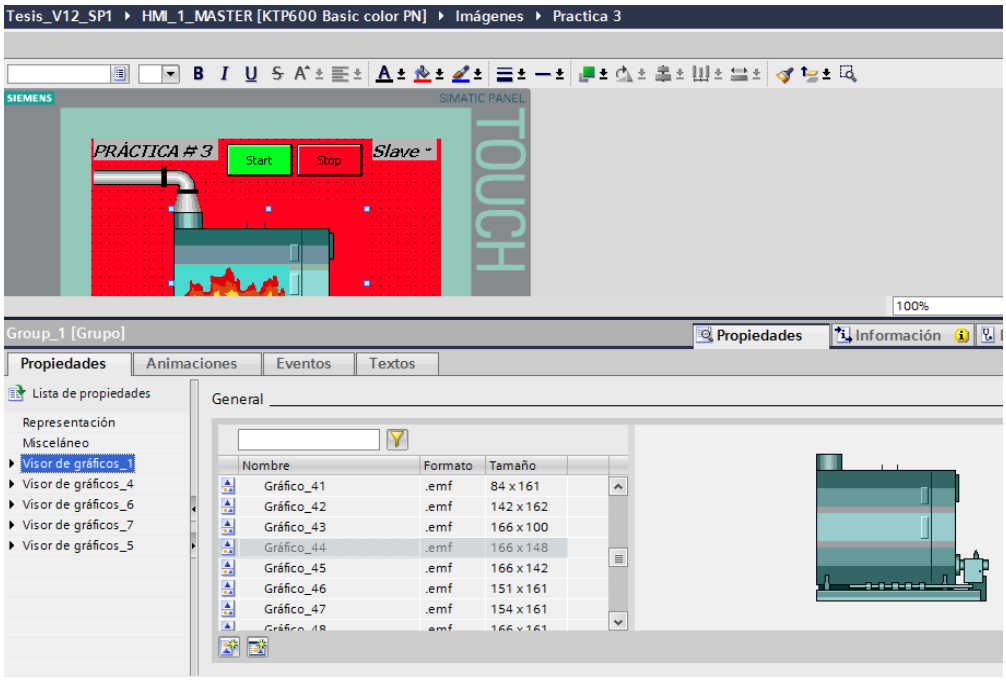


Figura 155: Composición de animación Grupo 1 – Caldero.

Respecto a la asignación de la variable a utilizar es la siguiente:

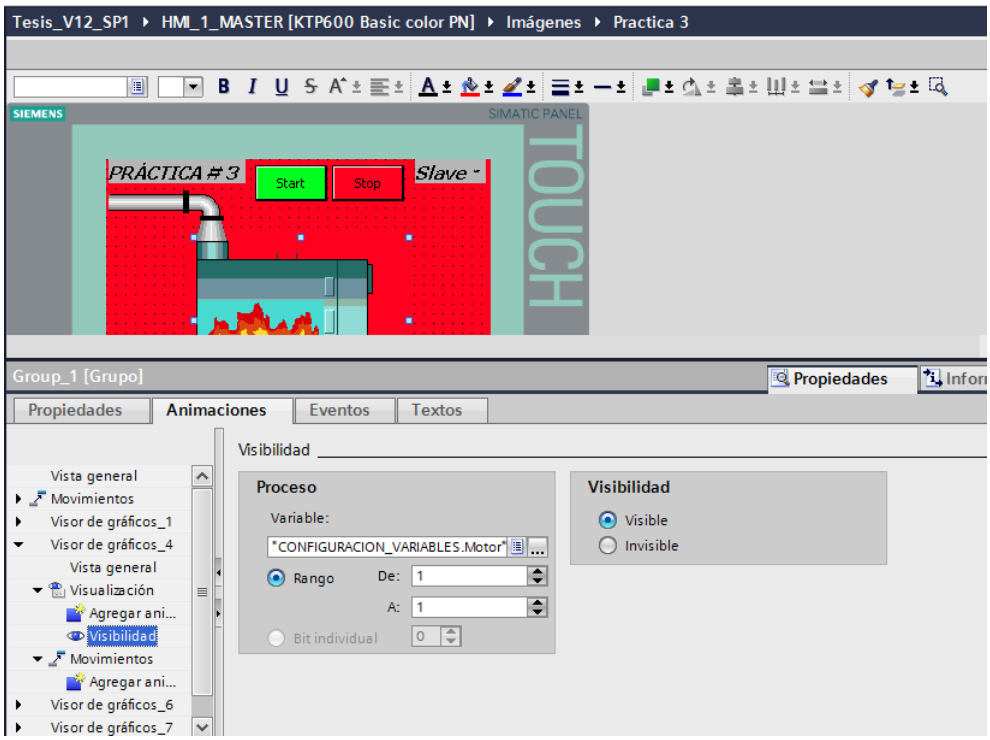


Figura 156: Variable asignada para la animación de motor.

Para realizar la transición del cambio de gráfico se debe agregar intervalos de

tiempo, esto lo hacemos en **Animaciones**, **Visor de gráficos** (se selecciona uno por uno), **Visibilidad** y finalmente seleccionamos el **Rango** de cada gráfico.

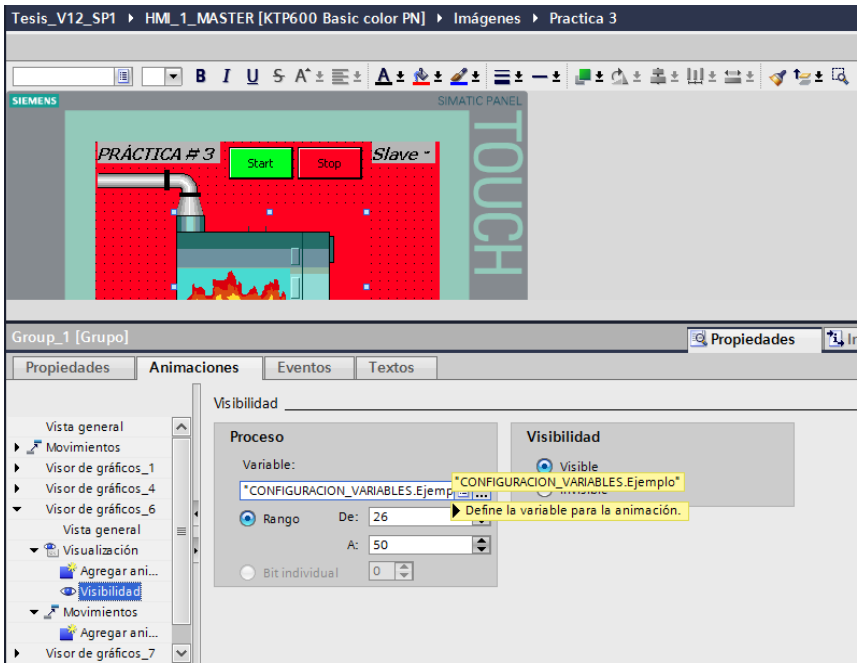


Figura 157: Variable asignada para la animación del caldero.

Referente al cambio de color del motor debemos configurar su apariencia.

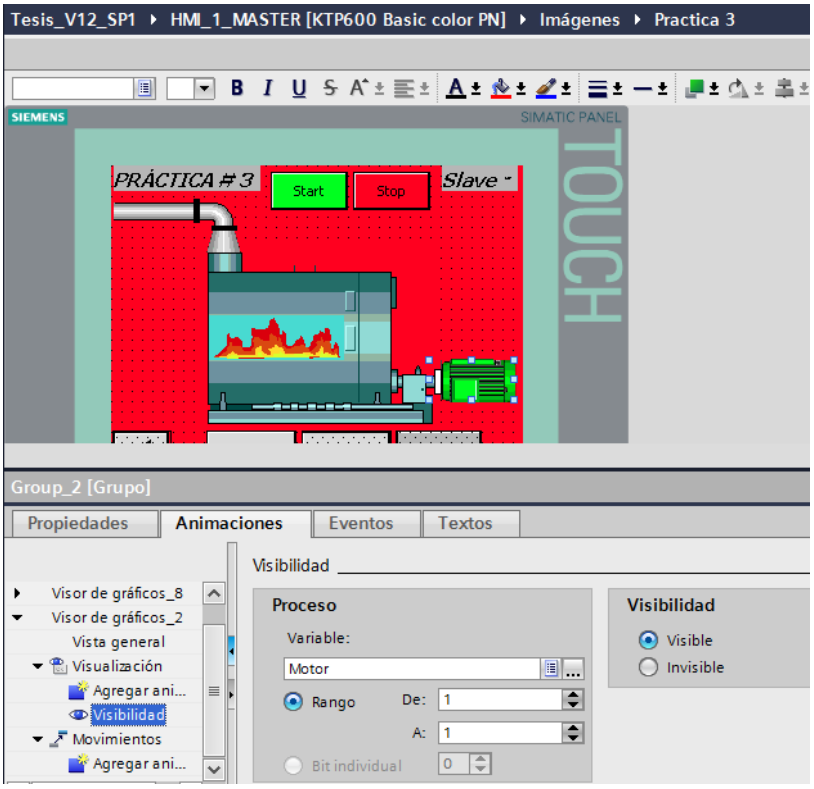


Figura 158: Variable asignada para la salida del motor.

Paso 7.

El botón de selección de modo, maestro o esclavo, es una agrupación de 4 botones, 2 visibles que se mostrarán en los estados “Master” y “Slave” (Botón_8, Botón_9), al pulsar el botón se cambiara de maestro a esclavo y viceversa. Los 2 botones invisibles funcionarán para el bloqueo en caso de que se seleccionen las mismas opciones por error (Botón_11, Botón_12).

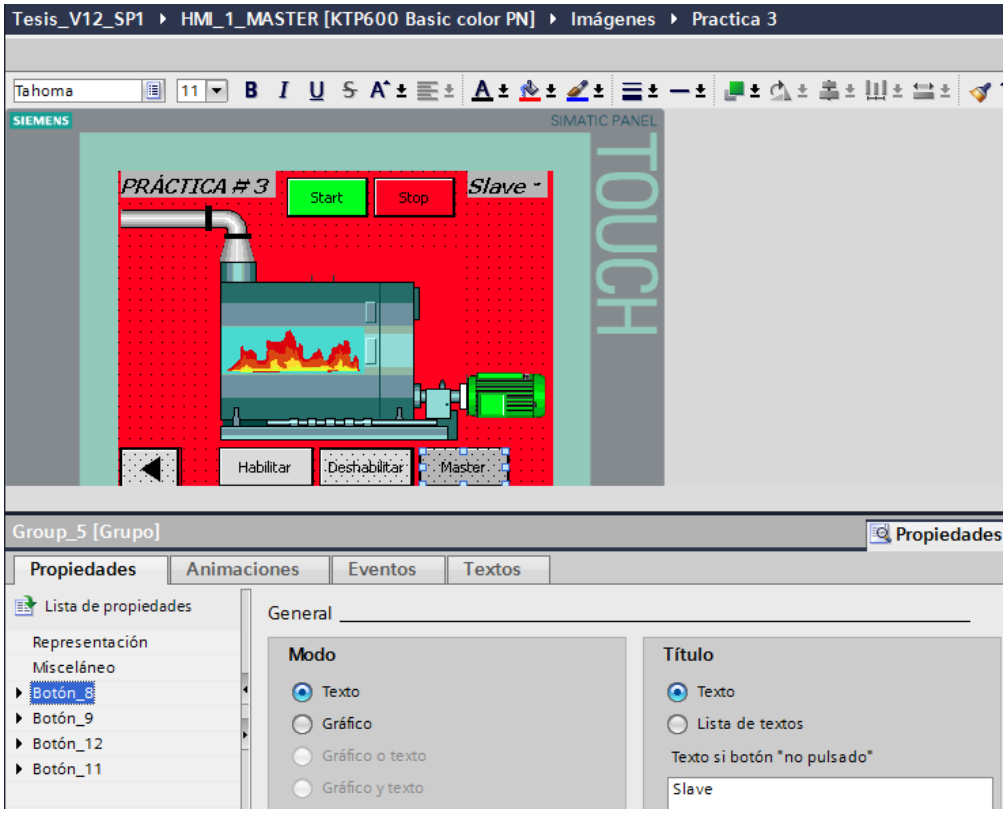


Figura 159: Propiedades del botón para selección de modo.

Figura 160: Eventos del botón para selección de modo.

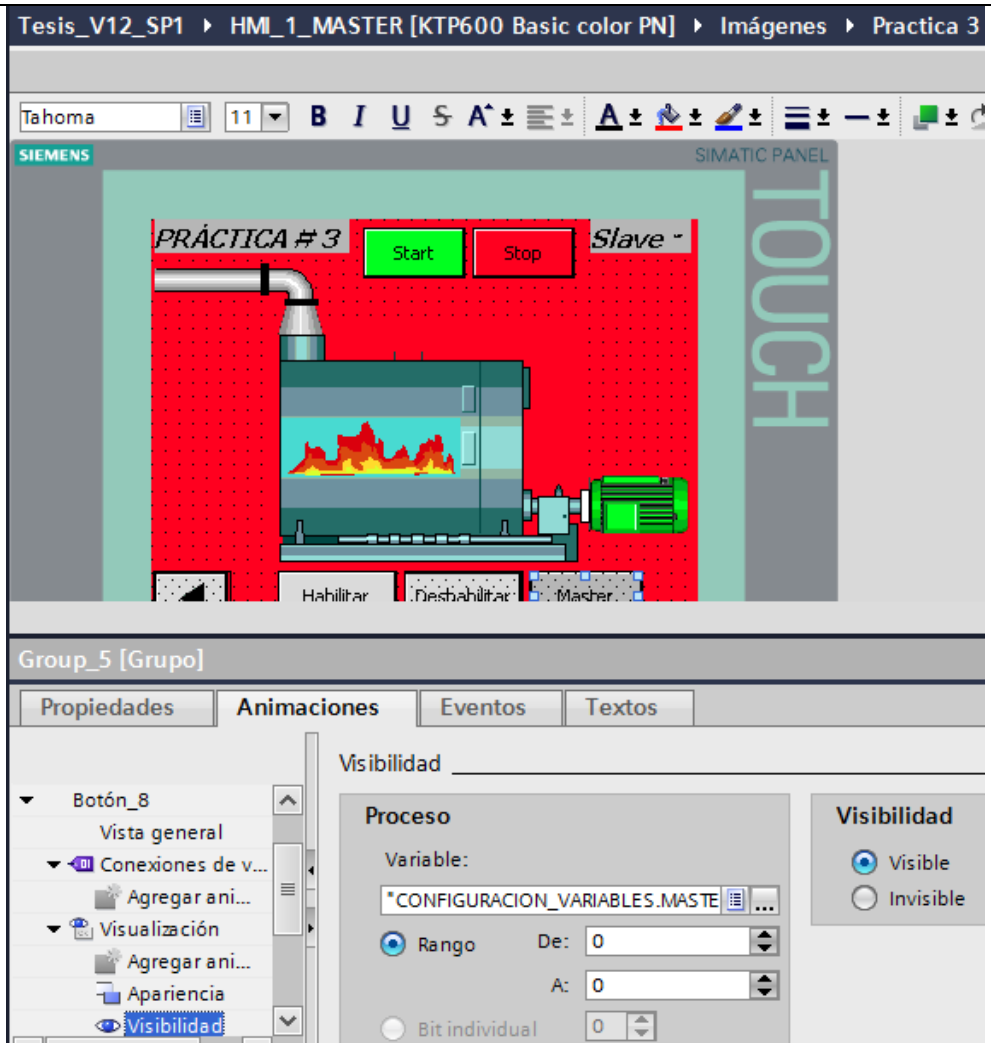


Figura 161: Asignación de variable de Botón_8 de selección de modo.

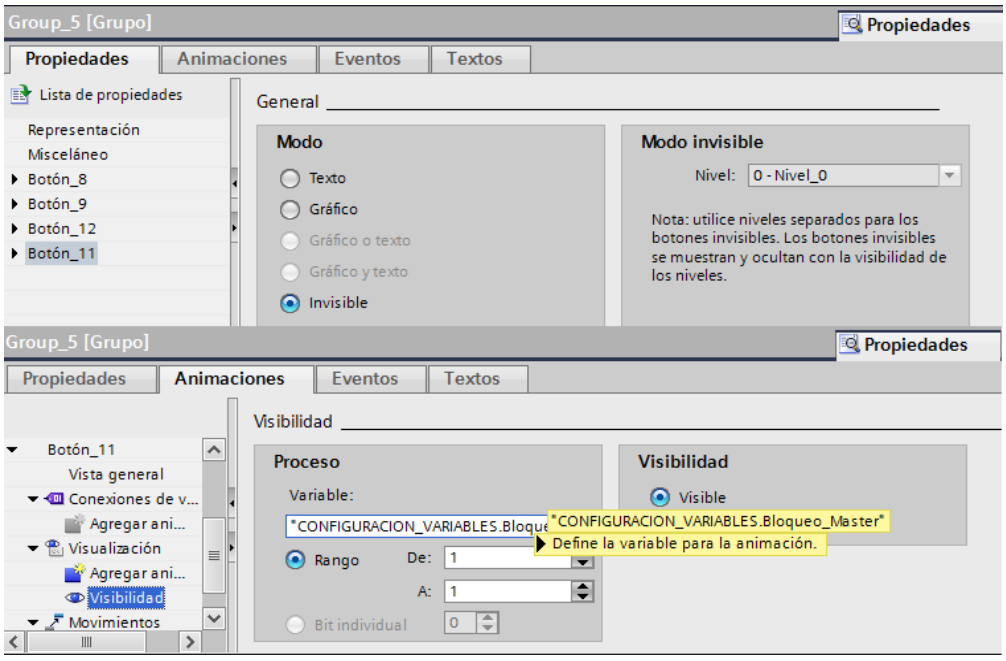


Figura 162: Asignación de variable y animación de Botón_11 de selección de modo.

Paso 8.

Ahora vamos a configurar los botones Start y Stop para lo cual se mostrarán las respectivas configuraciones de los controles.

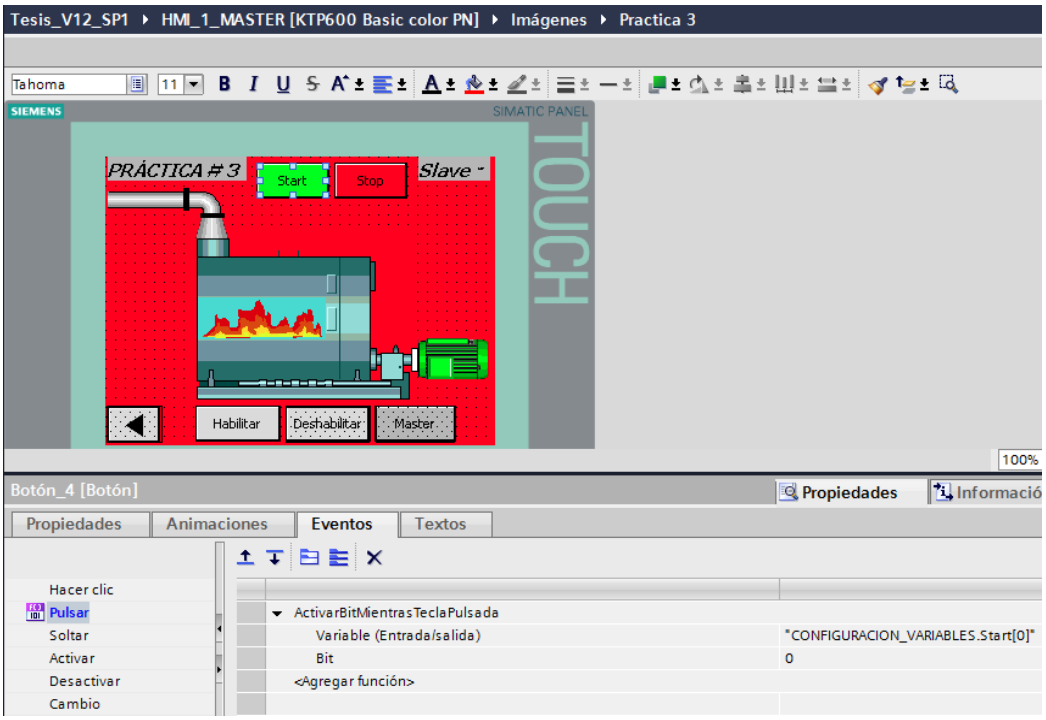


Figura 163: Asignación de variable para el evento del botón Start.

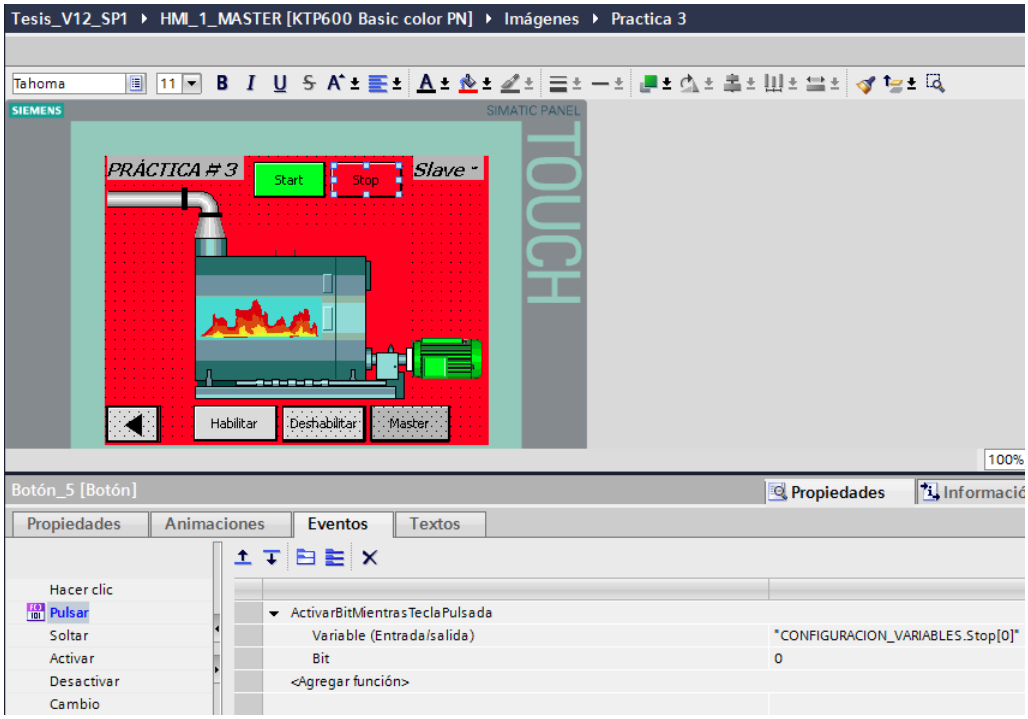


Figura 164: Asignación de variable para el evento del botón Stop.

Observaciones: Respecto a los botones **Habilitar**, **Deshabilitar**, y los **mensajes de alerta** se debe seguir las instrucciones del **Paso 20** de la **Práctica 1**.

Paso 9.

El proceso de creación de imágenes para el PLC SLAVE es el mismo que el PLC MASTER, pero diferenciando las variables a utilizar, a continuación se mostrará el diseño del silo animado que fue asignado al **HMI_1_SLAVE**.

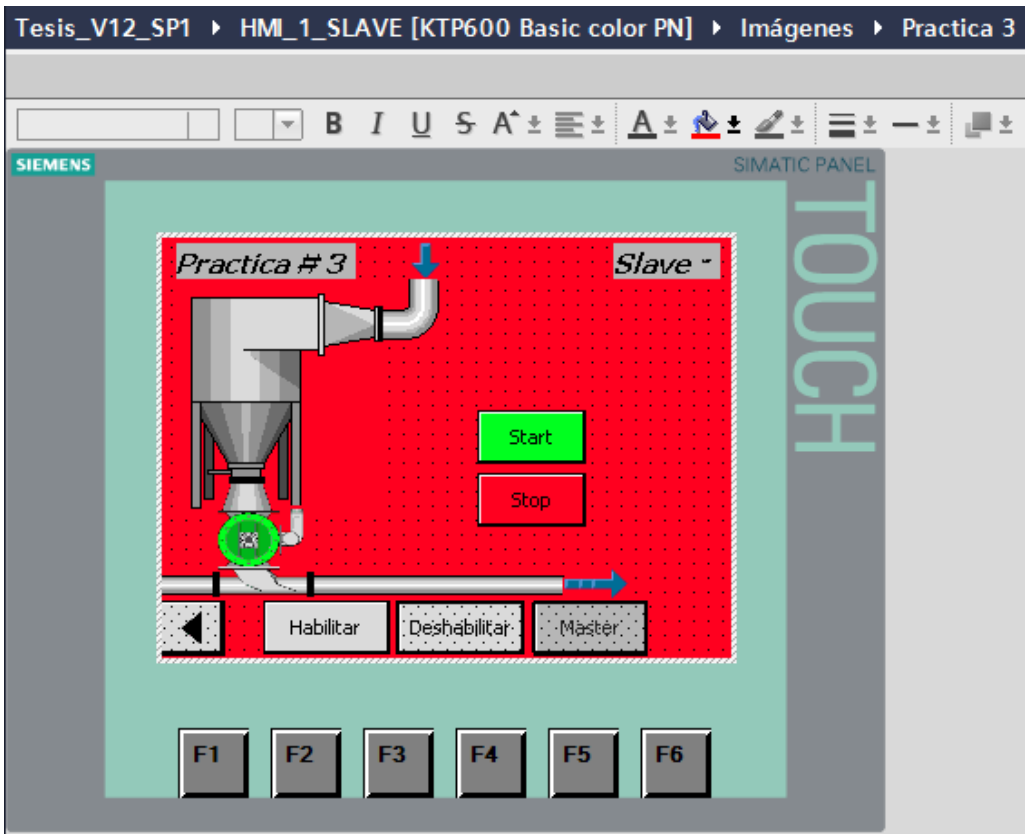


Figura 165: Diseño del HMI_1_SLAVE de la práctica 3.

Recomendaciones:

- Comprobar que las variables estén identificadas y con nombre para cada parte del proceso.
- Revisar que las variables de los controles del HMI estén correctamente asignadas.
- Verificar que los datos enviados y recibidos se guarden en las respectivas variables asignadas.

4.4. Práctica Nº4

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>4</td><td>TÍTULO PRÁCTICA: “Sistema de llenado a escala de un tanque controlado vía MODBUS RTU.”</td></tr>
<tr><td colspan="4">OBJETIVOS:
Diseñar un sistema de llenado manual y automático entre un maestro y un esclavo con 2 PCL S7-1200.
Implementar un control de nivel de una planta de agua vía comunicación MODBUS RTU.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:
Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
Dos módulos de comunicación CM1241.
Un cable de comunicación RS485 macho – macho.
Dos tanques de 10 litros de capacidad (TK1 / TK2).
Una electroválvula (SV101).
Una bomba centrifuga (P101).
Un sensor ultrasónico (TK102) LIC102.2
Un sensor tipo boya en tanque TK102 (LSL102.1).
Dos sensores de proximidad capacitivos en TK101 (LSH101.2 / LSL101.1)
Dos pulsadores, uno para la selección de la práctica (S1) y el otro para resetear la selección (S2).
DESCRIPCIÓN:
Este sistema funciona de la siguiente manera: primero se debe establecer la comunicación en el módulo didáctico con el botón Habilitar para después presionar el pulsante S1 en la planta para habilitar la práctica 4. Establecida la comunicación entre maestro (módulo didáctico) y esclavo (planta) seleccionamos el modo Manual o Automático. En modo manual se podrá llenar el tanque TK2 presionando la figura de la bomba P101 en el HMI, para vaciar el tanque TK2 se pulsa la figura de la</td></tr>
</table>

electroválvula **SV101**. Para cambiar de modo a **Automático** se debe presionar el botón **Paro** en el HMI. Seleccionamos el modo automático y sin importar el nivel en que se encuentre el tanque **TK2**, esta va a comenzar a llenar activado por la bomba hasta su capacidad máxima 30 cm. Al llegar a 30 cm. el nivel comienza a bajar al activarse la electroválvula hasta llegar a su capacidad mínima 3.5 cm. El ciclo se repite hasta dar **Paro** al sistema.

La figura 166 muestra la arquitectura de red a implementar.

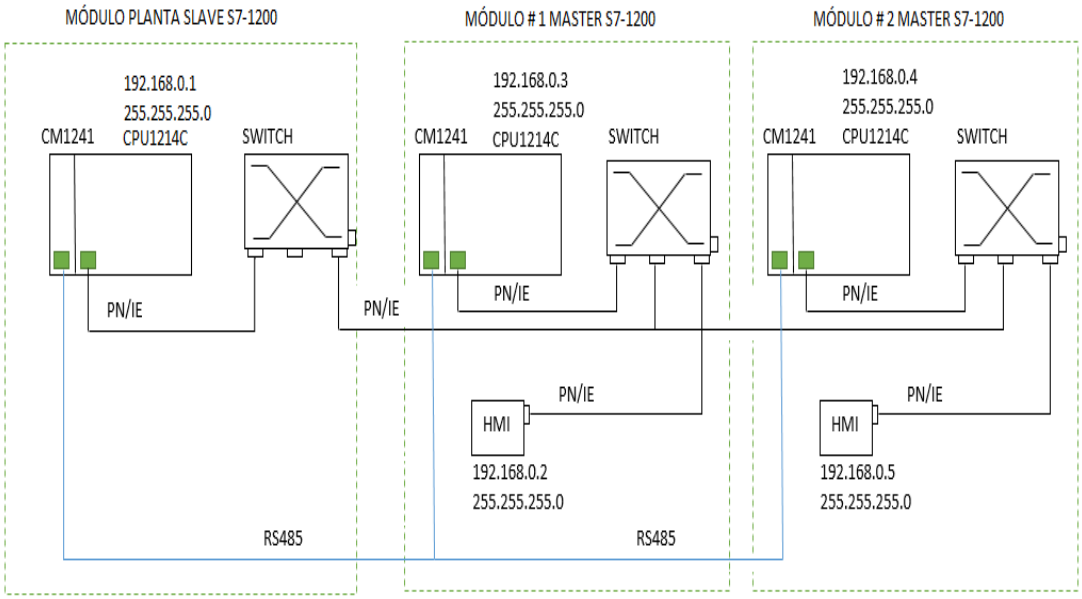


Figura 166: Arquitectura de red.

En figura 167 se muestra el diagrama de conexiones a implementar.

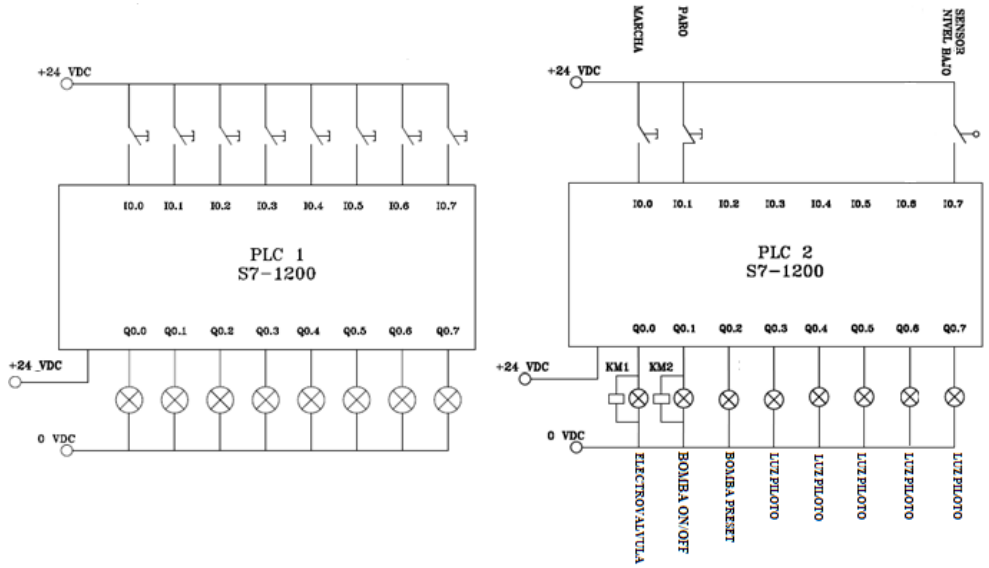


Figura 167: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear un nuevo PLC el cual llamaremos **PLANTA** dentro de nuestro proyecto **Tesis_V12_SP1**, para esto seguimos desde el Paso 5 al Paso 8 de la Práctica 1. Luego creamos el bloque de organización **PLANTA_OB1 [OB1]** donde programaremos la comunicación y control. Finalmente agregamos un segundo bloque de organización **Startup [OB100]** para nuestro bloque de carga.

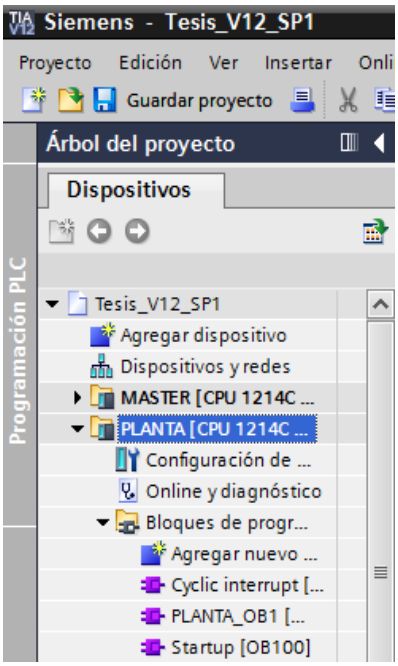
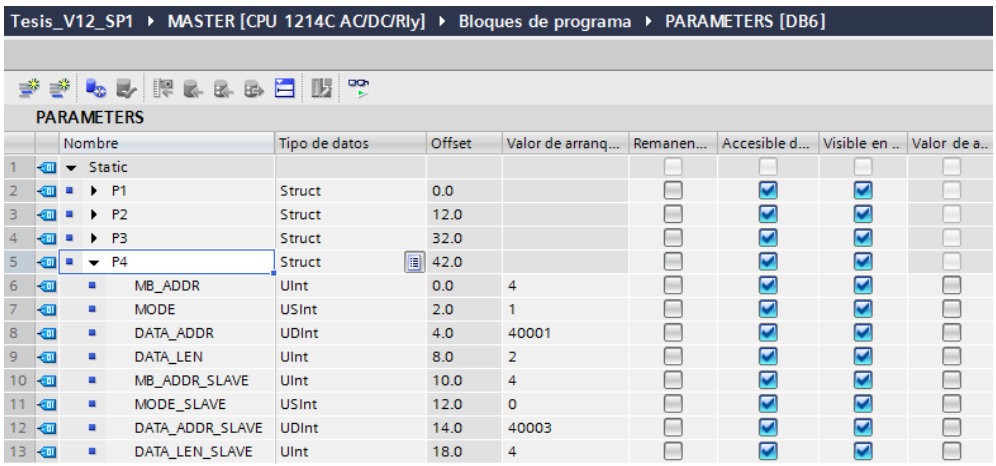


Figura 168: Creación del PLC PLANTA y bloques de organización.

Paso 2.

Como paso dos vamos a crear los parámetros para las instrucciones de la práctica 4, nos dirigimos a **PARAMETERS [DB6]** ubicado en el PLC **MASTER** y creamos las siguientes variables:



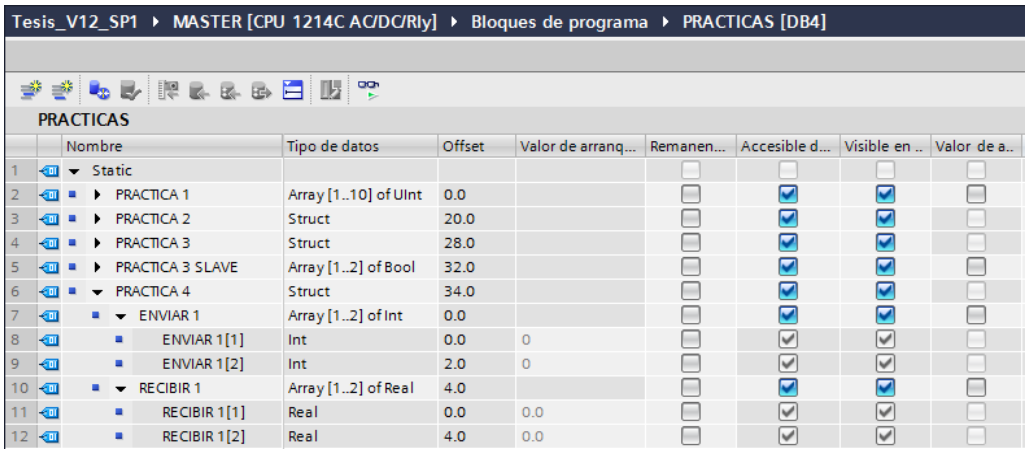
	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor de a...
1	Static				☐	☐	☐	☐
2	P1	Struct	0.0		☐	☑	☑	☐
3	P2	Struct	12.0		☐	☑	☑	☐
4	P3	Struct	32.0		☐	☑	☑	☐
5	P4	Struct	42.0		☐	☑	☑	☐
6	MB_ADDR	UInt	0.0	4	☐	☑	☑	☐
7	MODE	USInt	2.0	1	☐	☑	☑	☐
8	DATA_ADDR	UDInt	4.0	40001	☐	☑	☑	☐
9	DATA_LEN	UInt	8.0	2	☐	☑	☑	☐
10	MB_ADDR_SLAVE	UInt	10.0	4	☐	☑	☑	☐
11	MODE_SLAVE	USInt	12.0	0	☐	☑	☑	☐
12	DATA_ADDR_SLAVE	UDInt	14.0	40003	☐	☑	☑	☐
13	DATA_LEN_SLAVE	UInt	18.0	4	☐	☑	☑	☐

Figura 169: Parámetros del PLC MASTER, práctica 4.

Observación: También se necesita crear los parámetros para el PLC **SLAVE** los cuales van a ser los mismos que el **MASTER**. El PLC **PLANTA** no requiere de un bloque de parámetros ya que solo va a estar compuesto de bloques **MB_SLAVE_DB**, es decir la planta de nivel de agua funcionará como esclavo MODBUS.

Paso 3.

Ahora debemos generar las variables de registro para el PLC **MASTER** en **PRACTICAS [DB4]**. La imagen a continuación mostrará las variables de registro de datos.



	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	PRACTICA 3	Struct	28.0					
5	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
6	PRACTICA 4	Struct	34.0					
7	ENVIAR 1	Array [1..2] of Int	0.0					
8	ENVIAR 1[1]	Int	0.0	0				
9	ENVIAR 1[2]	Int	2.0	0				
10	RECIBIR 1	Array [1..2] of Real	4.0					
11	RECIBIR 1[1]	Real	0.0	0.0				
12	RECIBIR 1[2]	Real	4.0	0.0				

Figura 170: Variables de registro del PLC MASTER, práctica 4.

En la siguiente figura se mostrarán las variables de registro para el PLC PLANTA:

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
11	PRACTICA 4	Struct	34.0					
12	RECIBIR 1	Int	0.0	0				
13	RECIBIR 2	Int	2.0	0				
14	ENVIAR 1	Real	4.0	0.0				
15	ENVIAR 2	Real	8.0	0.0				

Figura 171: Variables de registro del PLC PLANTA, práctica 4.

Paso 4.

Vamos a generar los bloques de comunicación para el PLC **PLANTA**, nos dirigimos al bloque **Startup [OB100]** donde vamos agregar el bloque de carga:

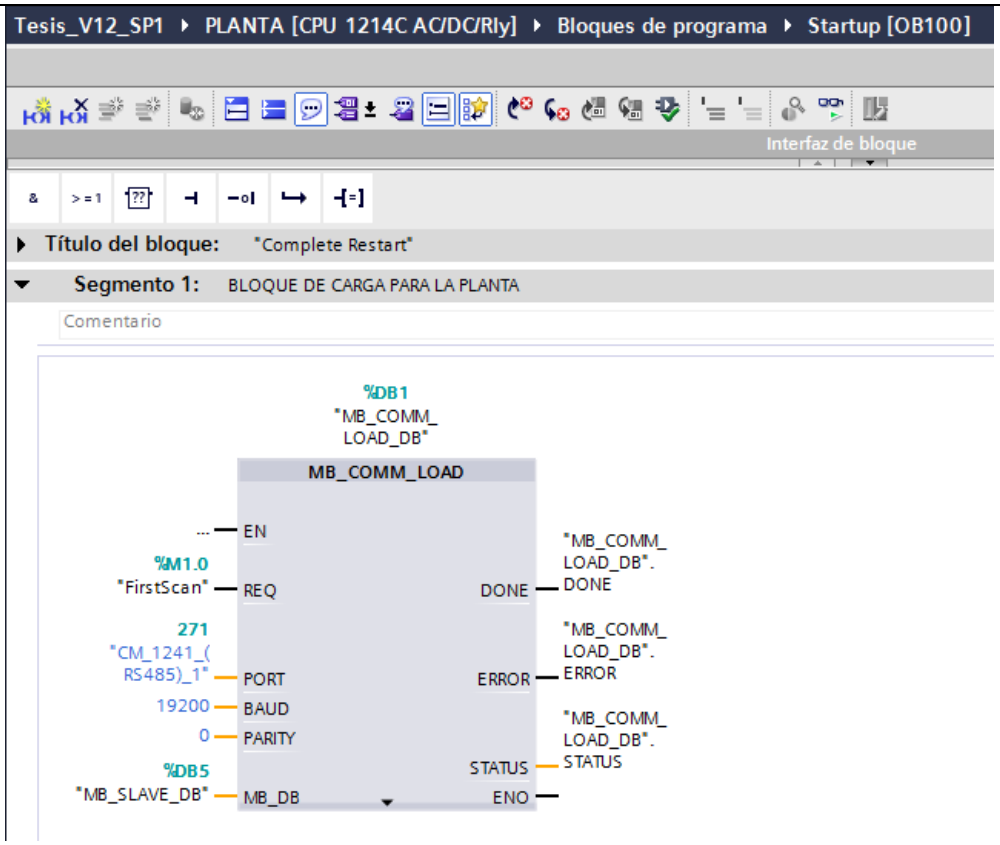


Figura 172: Configuración de instrucción MB_COMM_LOAD para PLANTA.

Nos dirigimos a **PLANTA_OB1** donde crearemos el bloque de instrucción **MB_SLAVE_DB** que receptará los datos de comunicación en la planta. Esto lo vamos a generar en el **Segmento 1**.

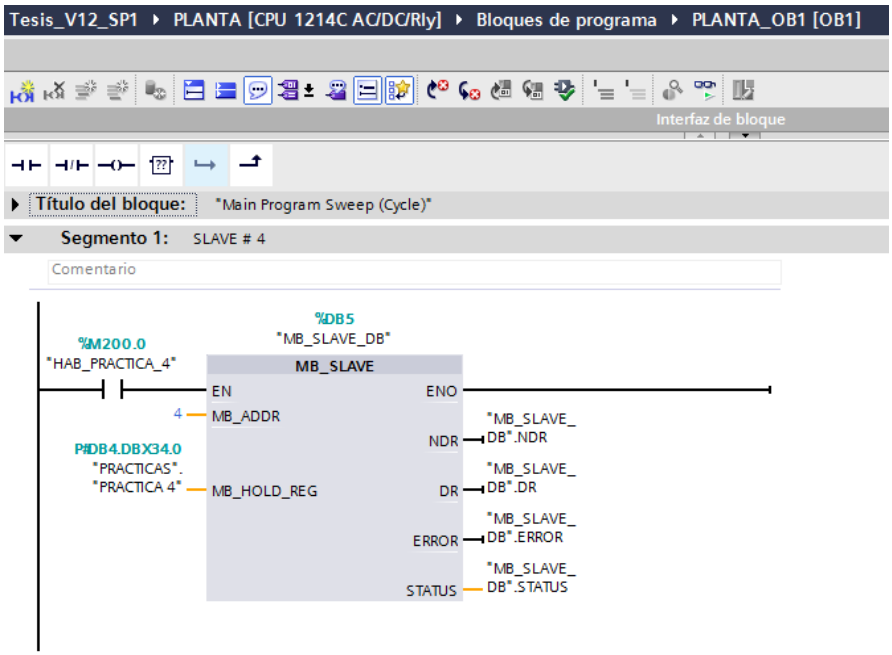


Figura 173: Bloque Slave de práctica 4.

Paso 5.

Ahora damos paso a la creación de lectura de datos del sensor ultrasónico el cual el dato se va a guardar en la variable **MD24**. Esto se guardará en las variables de cada práctica.

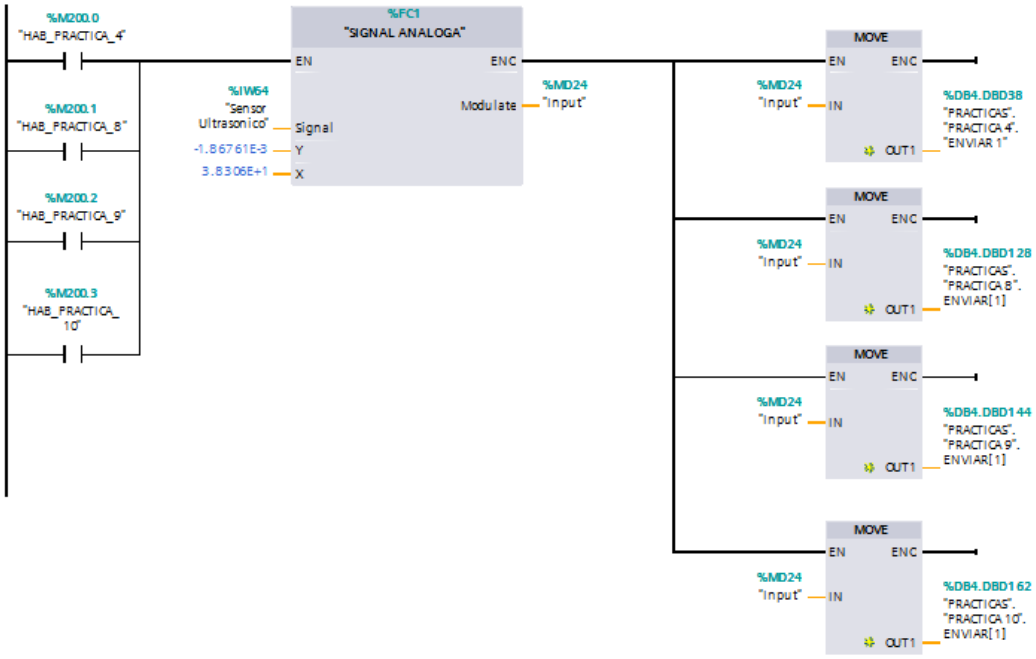


Figura 174: Señal de sensor ultrasónico.

Los valores de entrada de X y Y se obtienen de la ecuación de la recta, para esto generamos tabla con los datos obtenidos del sensor en cada centímetro del tanque.

	Valor sensor	Regla cm
2	18664	4
3	18135	5
4	17596	6
5	16979	7
6	16325	8
7	15901	9
8	15261	10
9	14730	11
10	14157	12
11	13778	13
12	13086	14
13	12652	15
14	12129	16
15	11559	17
16	11018	18
17	10534	19
18	10009	20

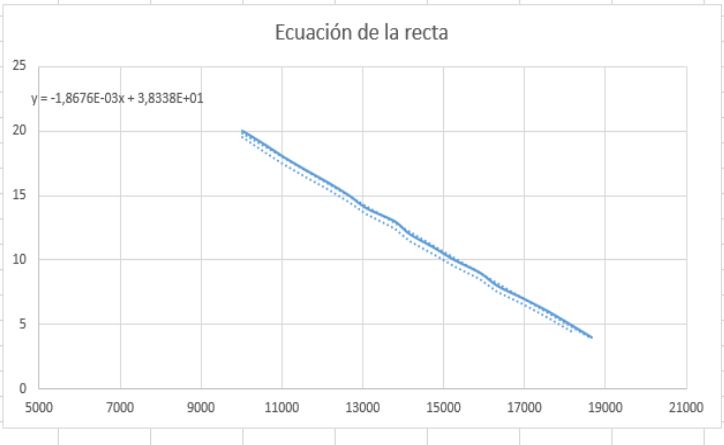


Figura 175: Ecuación de la recta.

Paso 6.

En este paso vamos a crear los bloques de comunicación para el PLC **MASTER** en el **Segmento 5,** aquí funcionaran dos bloques máster donde uno tendrá la labor de escribir datos y el segundo en leer:

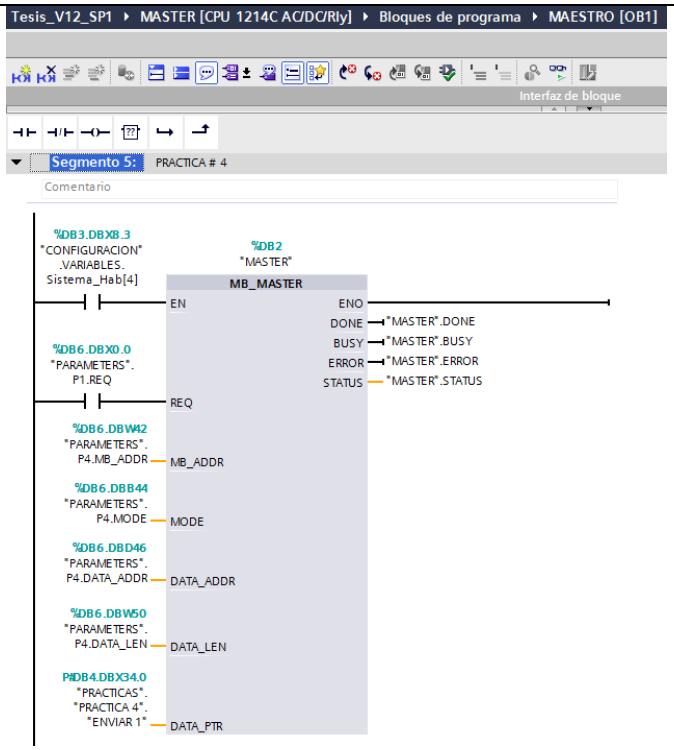


Figura 176: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 4.

El primer bloque de instrucción está funcionando en modo escritura, el cual enviará los datos al slave como activación de la bomba y electroválvula. Debemos agregar un segundo bloque máster el cual va a leer el sensor ultrasónico para detección de nivel de agua en los tanques.

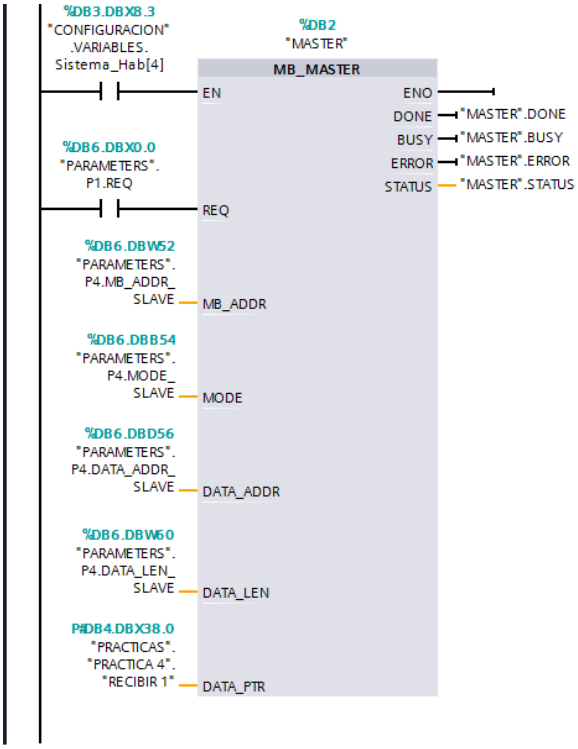


Figura 177: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 4.

Paso 6.

Antes de detallar la parte de control es necesario identificar las entradas y salidas a utilizar en el programa, cabe indicar que solo se ocuparan las entradas y salidas del PLC PLANTA ya que en el PLC MASTER / SLAVE el control de señales será por medio del HMI y no por entradas / salidas del PLC.

Las entradas y salidas a utilizar en la PLANTA están designadas de acuerdo al panel de control de la planta, la cual fue diseñada en un trabajo de titulación anterior, lo cual presentamos a continuación:

Tabla 10: E/S de la planta de control de nivel.

NOMBRE	DIRECCIÓN
PULSADOR_MARCHA	%I0.0
PULSADOR_PARO	%I0.1
BOTON_PARO_EMERGENCIA	%I0.2
BOTON_SEL_MANUAL	%I0.3
BOTON_SEL_AUTOMATICO	%I0.4
PULSADOR_JOG_VALVULA	%I0.5
PULSADOR_JOG_VARIADOR	%I0.6
NIVEL_BAJO_TANQUE102	%I0.7
VALVULA_ABIERTA	%I1.0
VALVULA_CERRADA	%I1.1
NIVEL_ALTO_TANQUE101	%I1.2
NIVEL_BAJO_TANQUE101	%I1.3
ELECTROVALVULA	%Q0.0
BOMBA_ON/OFF	%Q0.1
BOMBA_PRESET	%Q0.2
LUZ_PILOTO_(H4)	%Q0.3
LUZ_PILOTO_(H5)	%Q0.4
LUZ_PILOTO_(H6)	%Q0.5
LUZ_PILOTO_(H7)	%Q0.6
LUZ_PILOTO_(H8)	%Q0.7
ENTRADA_ANALOGICA_NIVEL_TANQUE_102	%IW64
ENTRADA_ANALOGICA_RESERVA	%IW66
SALIDA_ANALOGICA_VARIADOR	%QW80

Paso 7.

Ahora detallaremos el circuito de control para el PLC MASTER, en el **Segmento 6** tenemos la selección del modo, Manual o Automático:

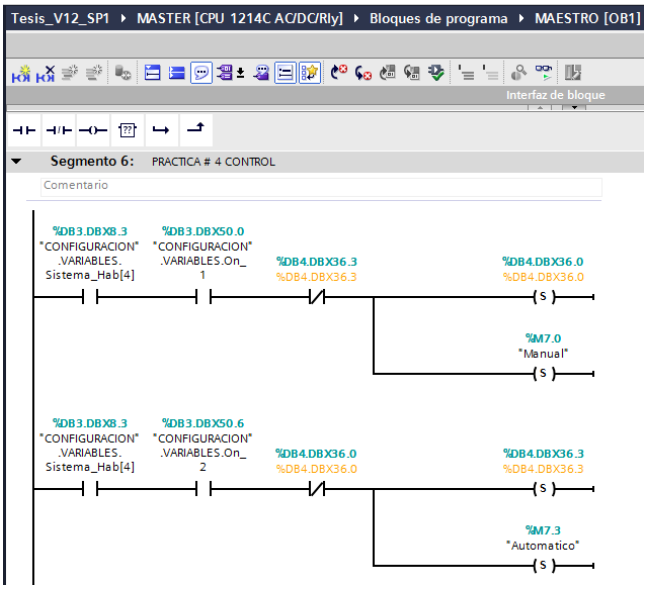


Figura 178: Selección de modo para el PLC MASTER, práctica 4.

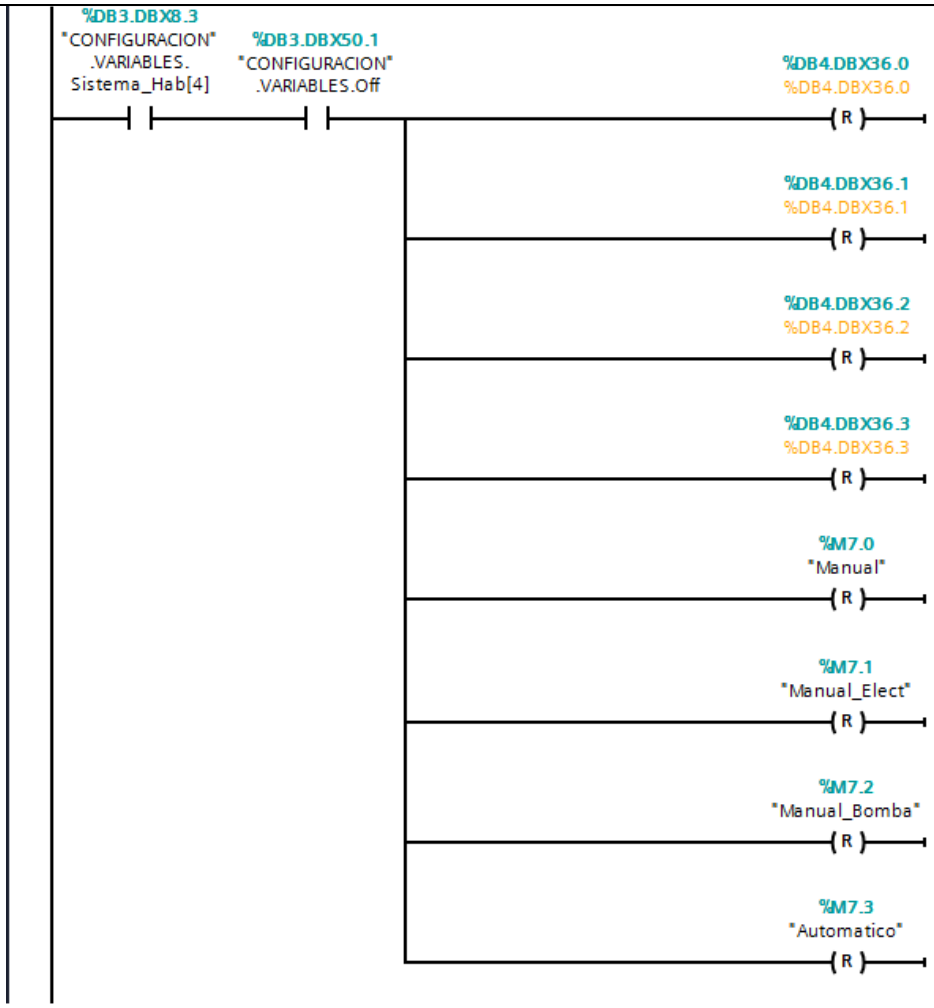


Figura 179: Marcas de apagado de controles manual / automático.

En esta sección (figura 180) se encenderá manualmente la bomba y electroválvula:

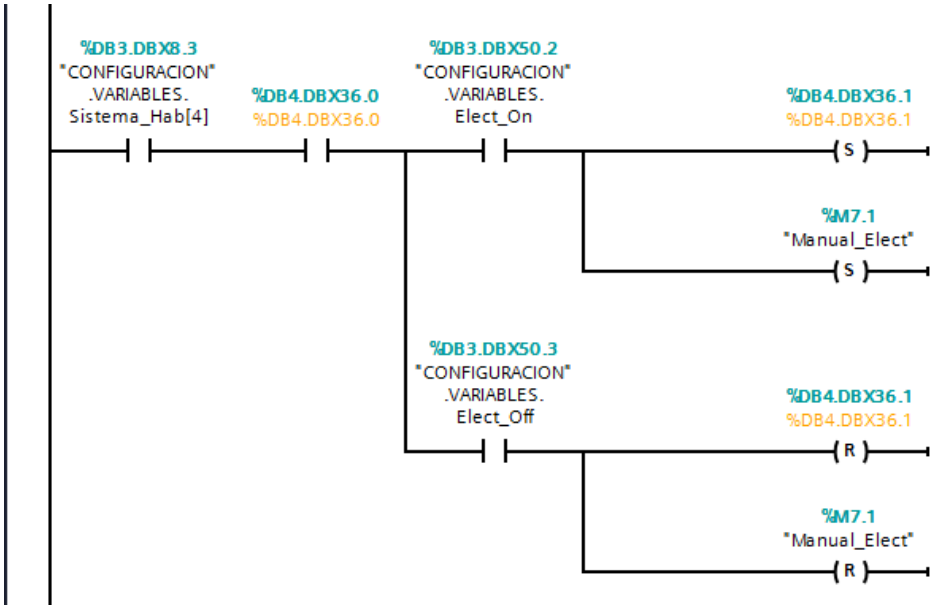


Figura 180: Activación manual de bomba y electroválvula.

En esta sección vamos a guardar el nivel del tanque en la variable de tipo real en la llamada NIVEL PLANTA.

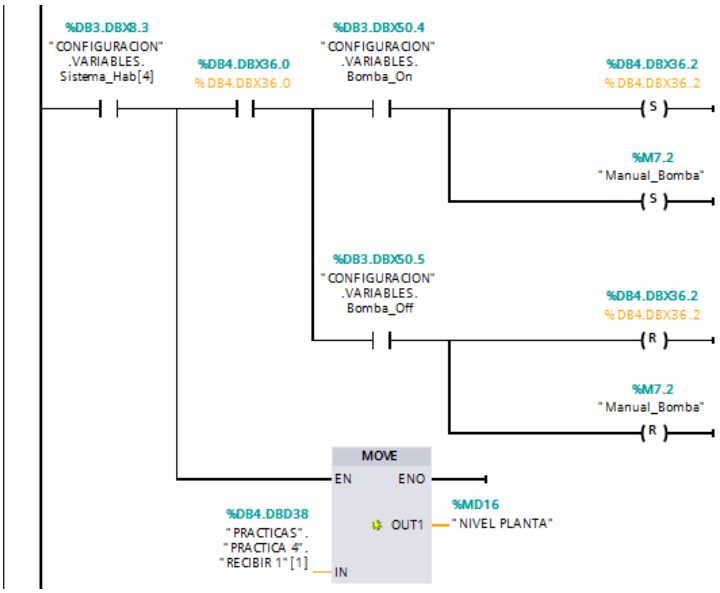


Figura 181: Registro de nivel en variable NIVEL PLANTA.

Paso 8.

Ahora vamos a programar el circuito de control para el PLC **PLANTA**, aquí vamos a diferenciar las diversas funciones del esclavo. Para esto se creará un bloque de función el cual llamaremos **PRACTICA # 4 [FC2]**.

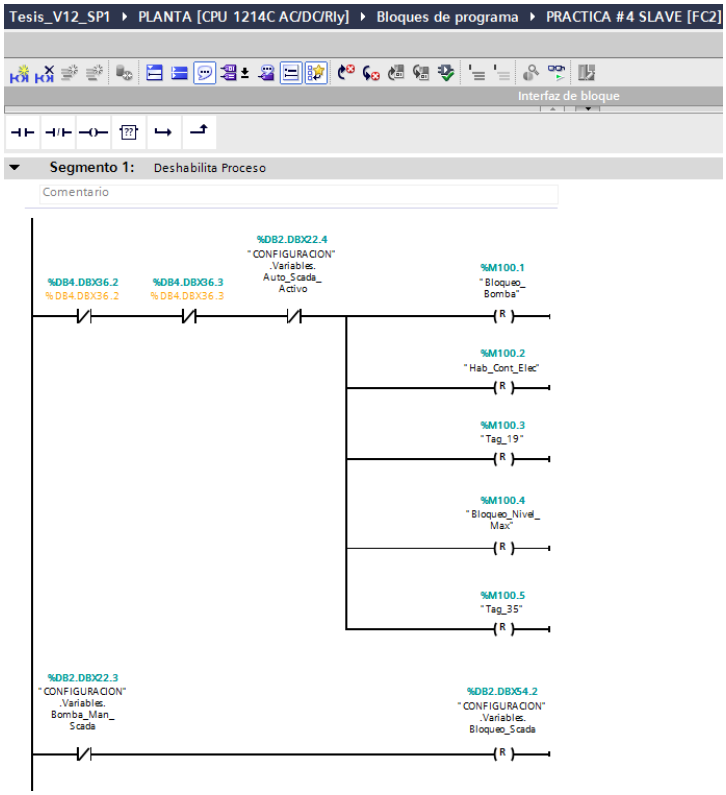


Figura 182: Control de Segmento 1, PLANTA.

En el **Segmento 2** realizaremos la activación y bloqueo de la Bomba (Q0.1).Se debe tomar en cuenta que para que la bomba P101 se desactive debe estar activada la marca Bloqueo_Bomba o Bloqueo_nivel_Max, es decir, cuando se rebase el nivel máximo permitido del tanque TK102 se activaran las marcas de bloqueo para desactivar la bomba, inmediatamente se activará la electroválvula SV101 para el vaciado del tanque.

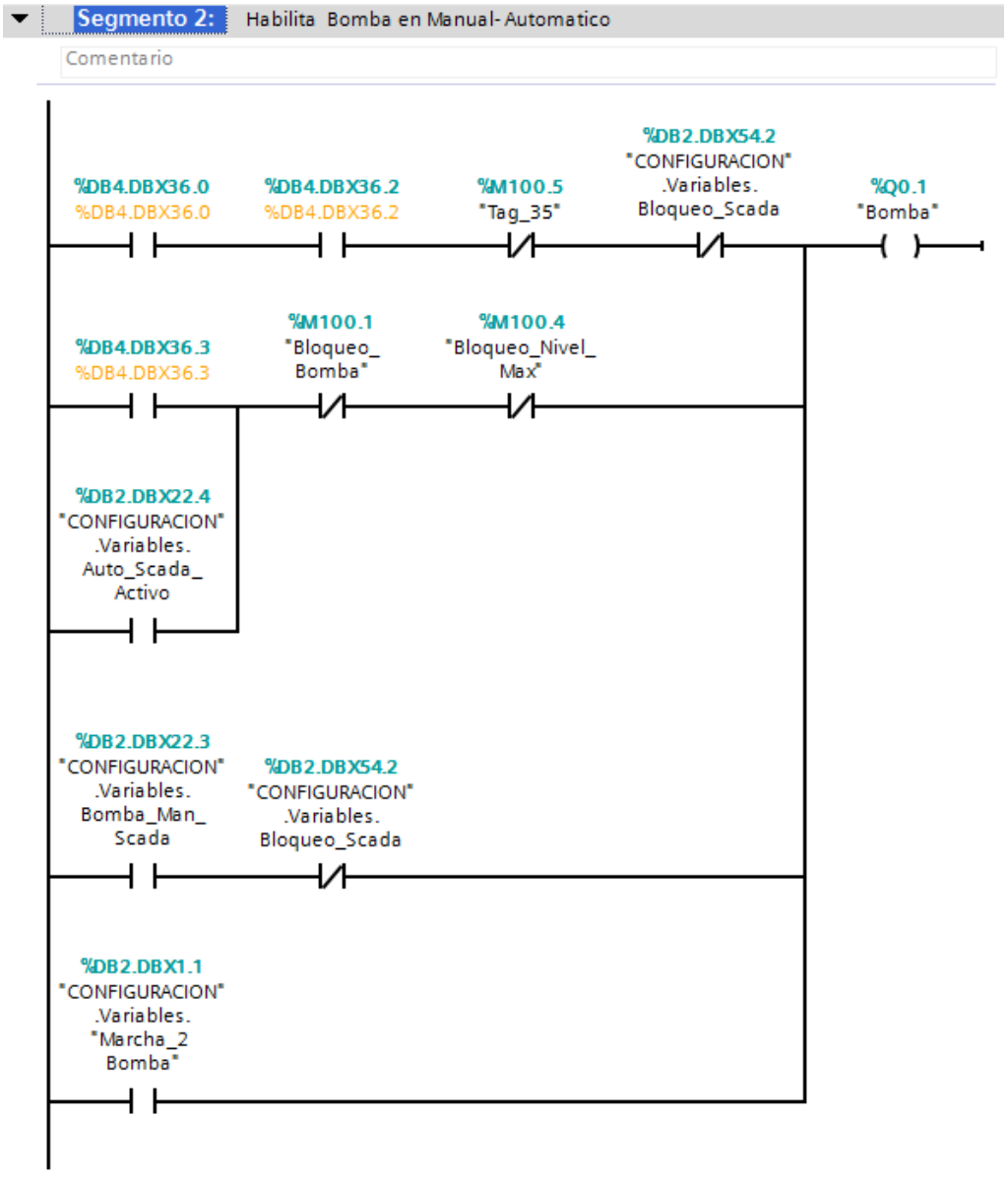


Figura 183: Activación y desactivación de la Bomba.

Una vez configurada la bomba pasaremos al diseño de activación y desactivación de la electroválvula, para esto lo generamos en el Segmento 3:

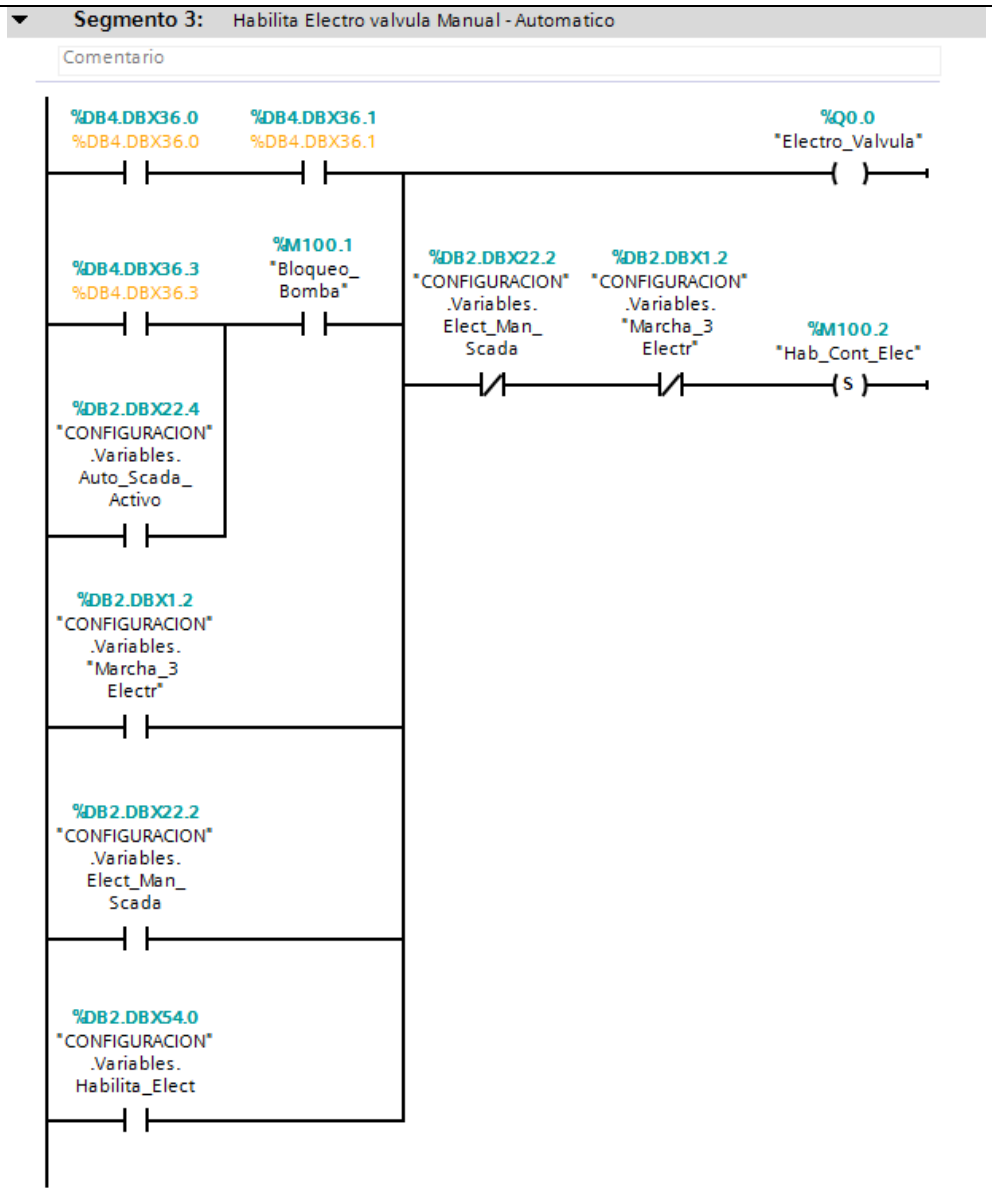


Figura 184: Activación y desactivación de la Electroválvula.

Al momento que se activa el modo Automático la bomba comenzará a trabajar y será controlada por dos temporizadores, cuando el nivel de agua sobrepase el sensor de boya LS102.1 del tanque TK2 se activará un temporizador que contará hasta 20 segundos, este tiempo fue calculado para que el nivel llegue al máximo (30 cm). Luego que pasan los 20 segundos se desactiva la bomba e inmediatamente se activa la electroválvula, a su vez también se activa un temporizador que contará 35 segundos que es el tiempo de vaciado. Luego el proceso se repite.

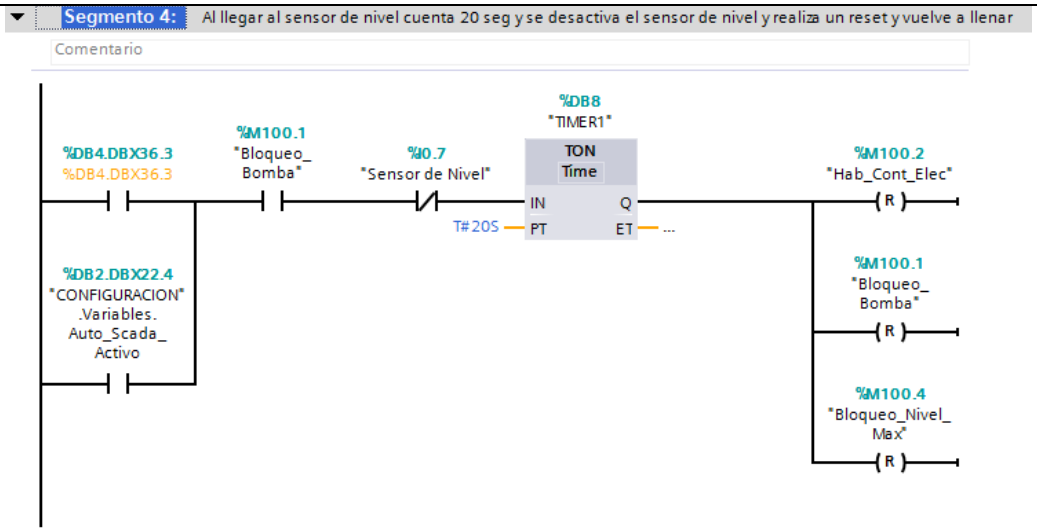


Figura 185: Llenado automático del tanque.

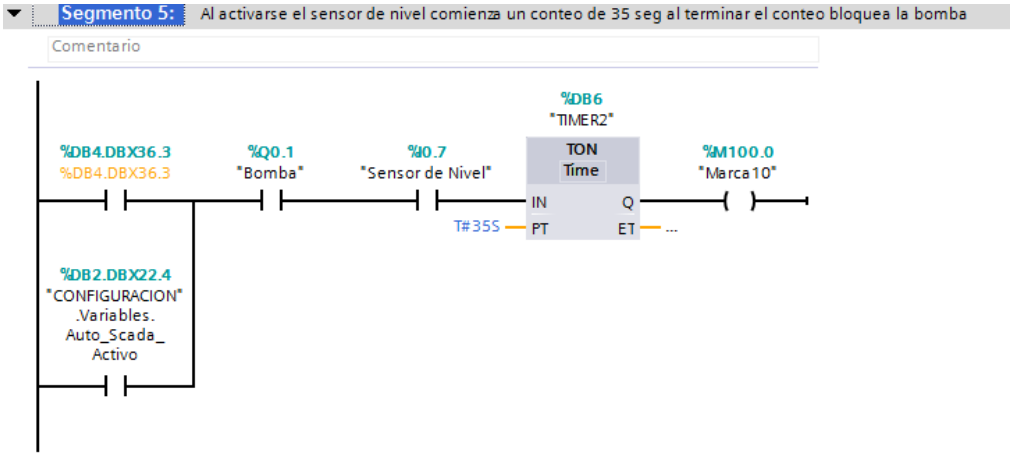


Figura 186: Vaciado automático del tanque.

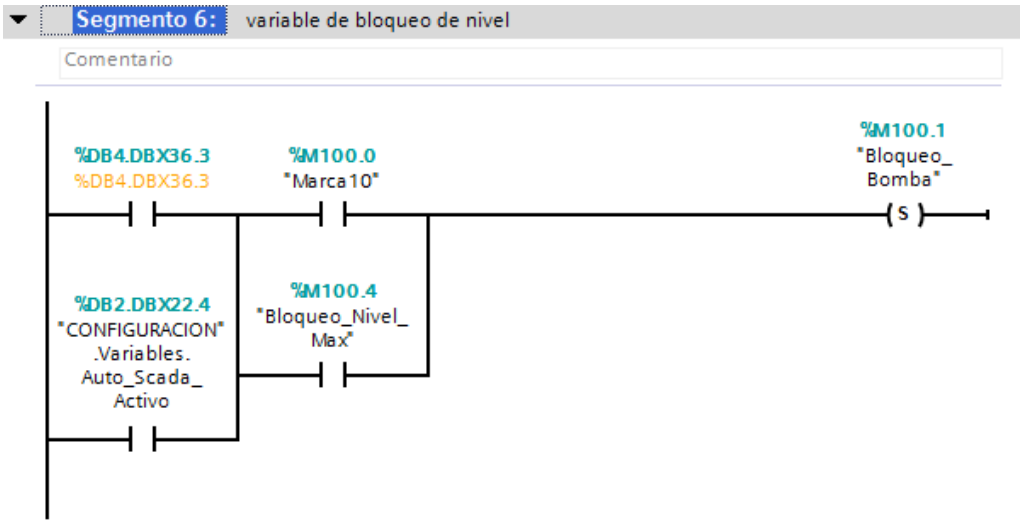


Figura 187: Variable de bloqueo de nivel.

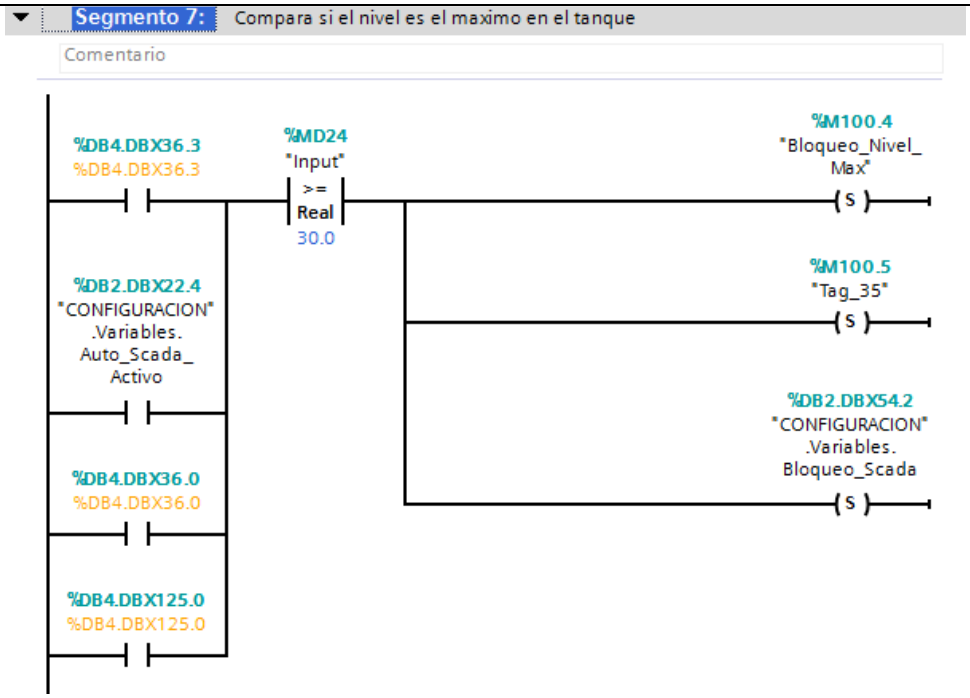


Figura 188: Comparación de nivel máximo del tanque TK102.

Paso 9.

Ahora daremos paso al diseño de las imágenes del HMI. Comenzaremos por el diseño de la imagen de presentación de la práctica 4, el diseño es similar a las demás prácticas.

Figura 189: Diseño de imagen Título 4.

Observación: para la creación de la imagen **Titulo 4** se debe seguir el proceso del paso 19 de la práctica 1, a diferencia de escoger las variables correspondientes de la práctica 4.
Se deberá generar una imagen de presentación para el PLC SLAVE con el mismo diseño.
La imagen **Practica 4** constará de los siguientes controles:

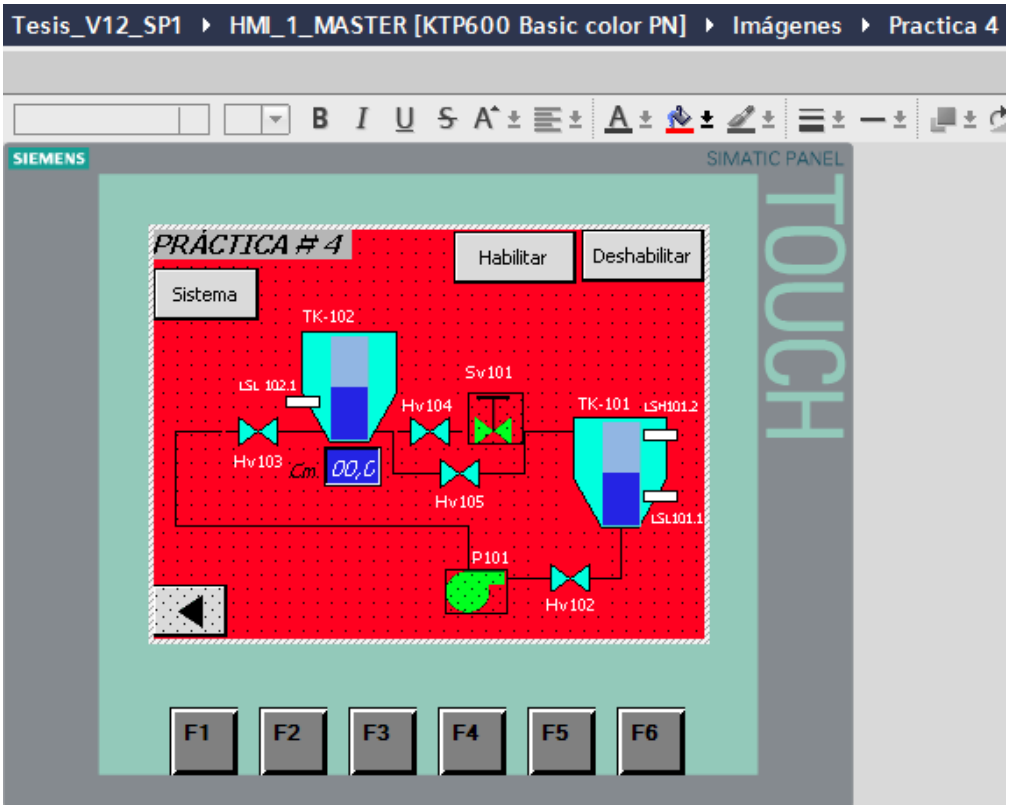


Figura 190: Creación de imagen Práctica 4.

Al presionar el botón **Sistema** no aparecerá una ventana secundaria la cual nos da a escoger si deseamos el sistema Manual o Automático. Para esto creamos un grupo el cual está compuesto por un rectángulo y cuatro botones.

Figura 191: Representación del grupo Sistema.

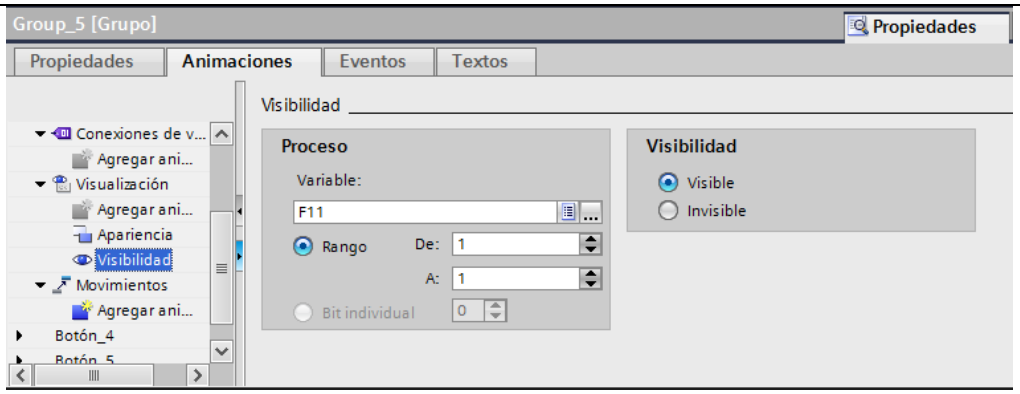


Figura 192: Animación de botón Automático.

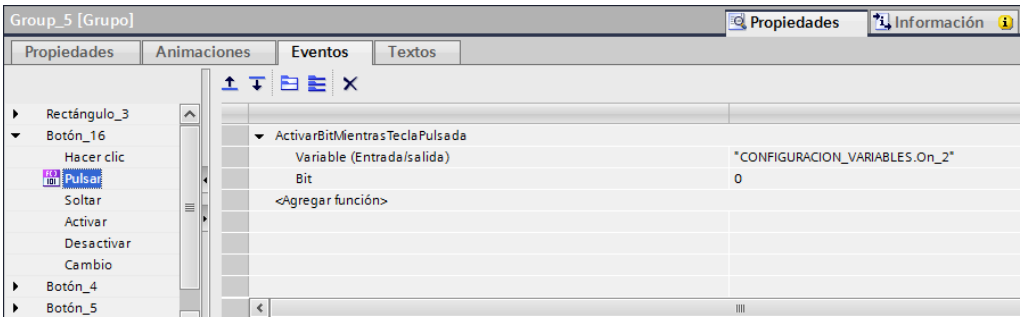


Figura 193: Evento de botón Automático.

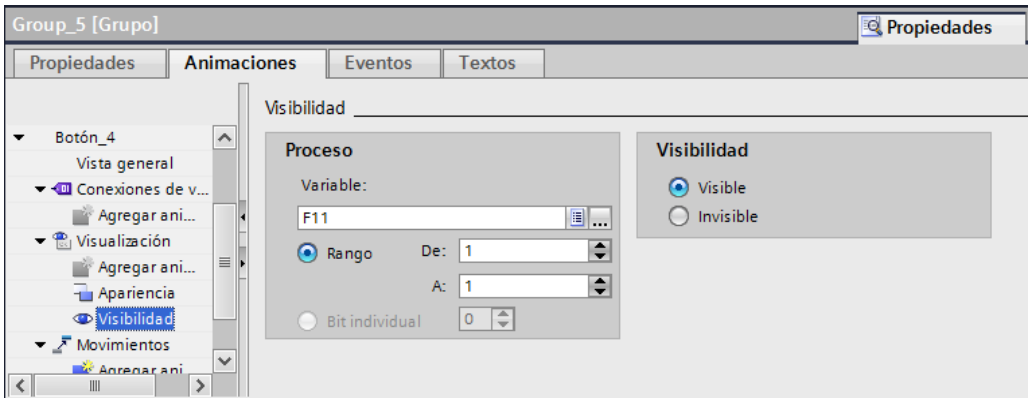


Figura 194: Animación de botón Manual.

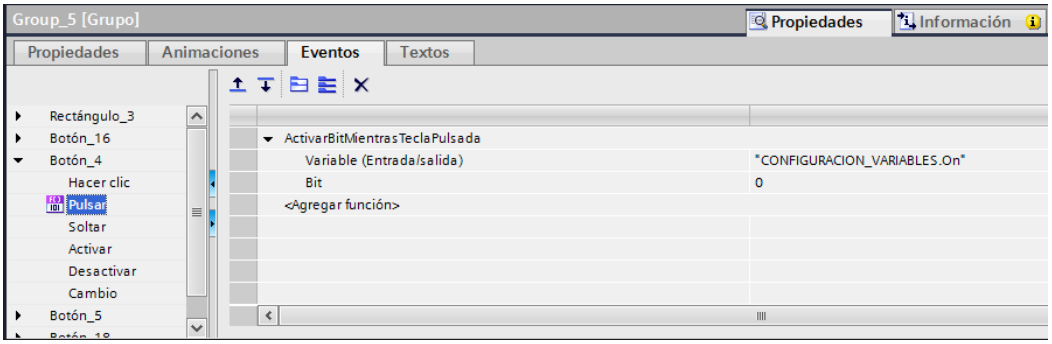


Figura 195: Evento de botón Manual.

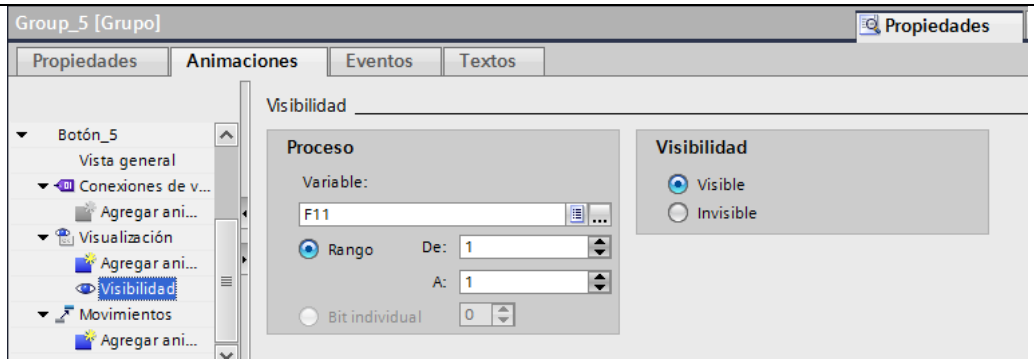


Figura 196: Animación de botón Paro.

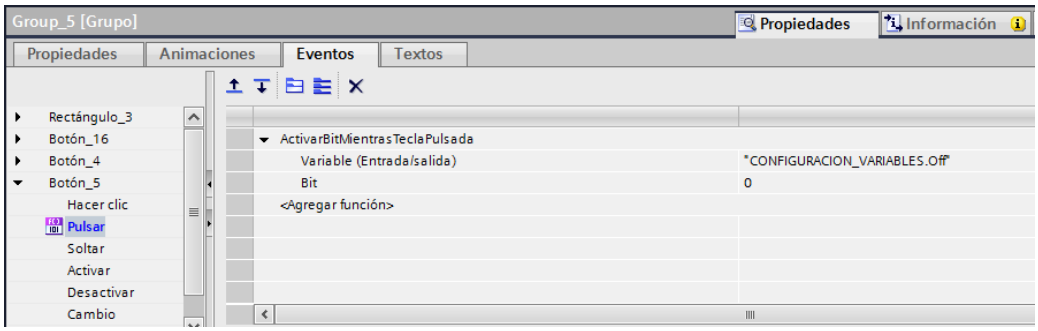


Figura 197: Evento de botón Paro.

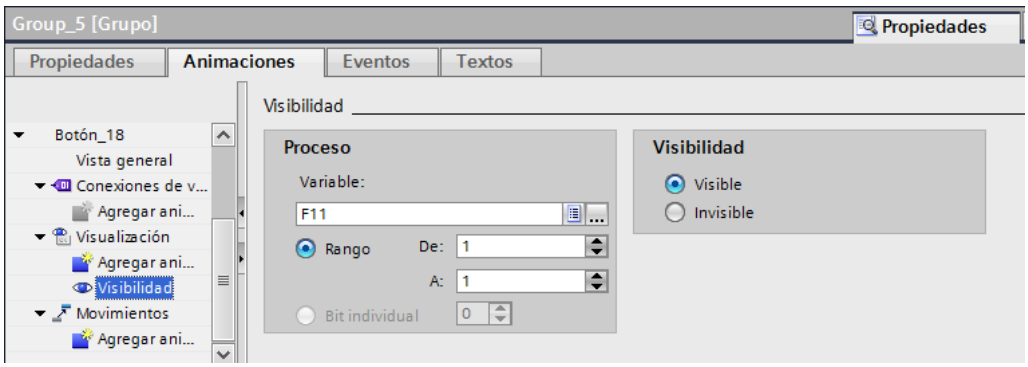


Figura 198: Animación de botón X.

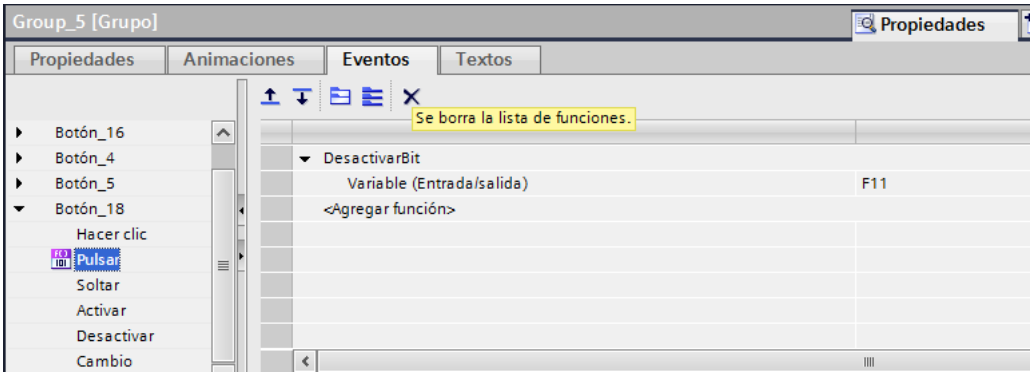


Figura 199: Evento de botón X.

Para la simulación de los tanques utilizaremos Barras las cuales estarán ligadas a las variables análogas, seleccionamos en rango de operación que es la capacidad del tanque TK102.

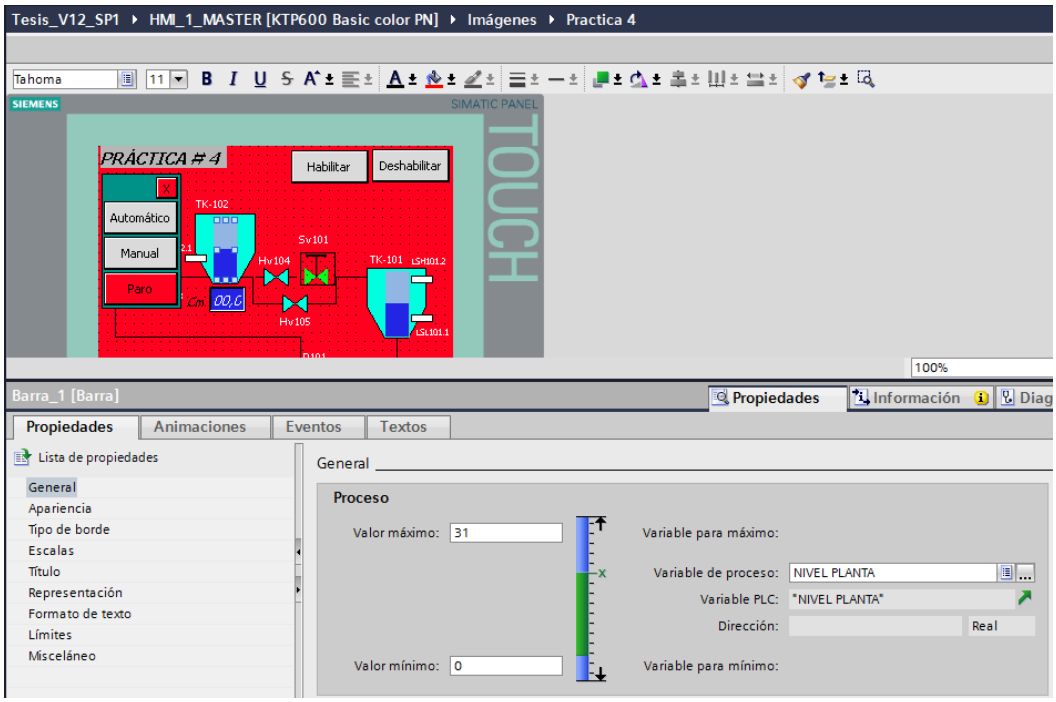


Figura 200: Creación de la Barra de nivel para el tanque TK102.

Ahora procedemos a asignar la variable al control.

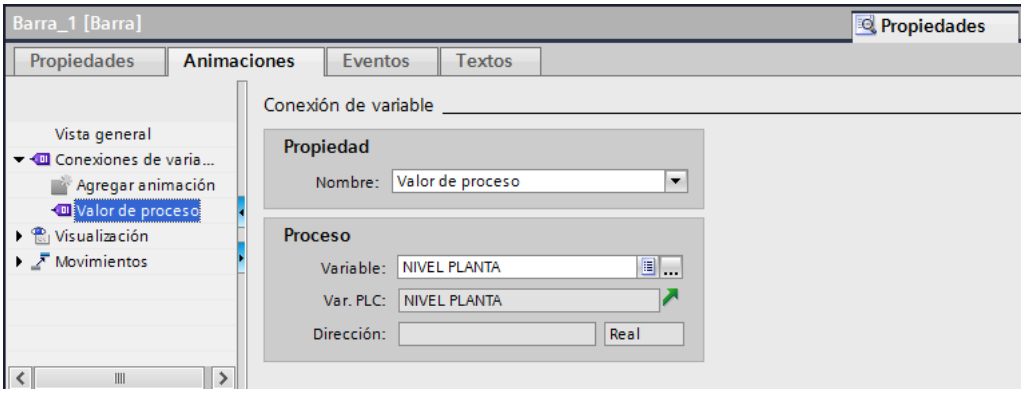


Figura 201: Asignación de variable NIVEL PLANTA.

Para el botón de la electroválvula presentamos la configuración correspondiente:

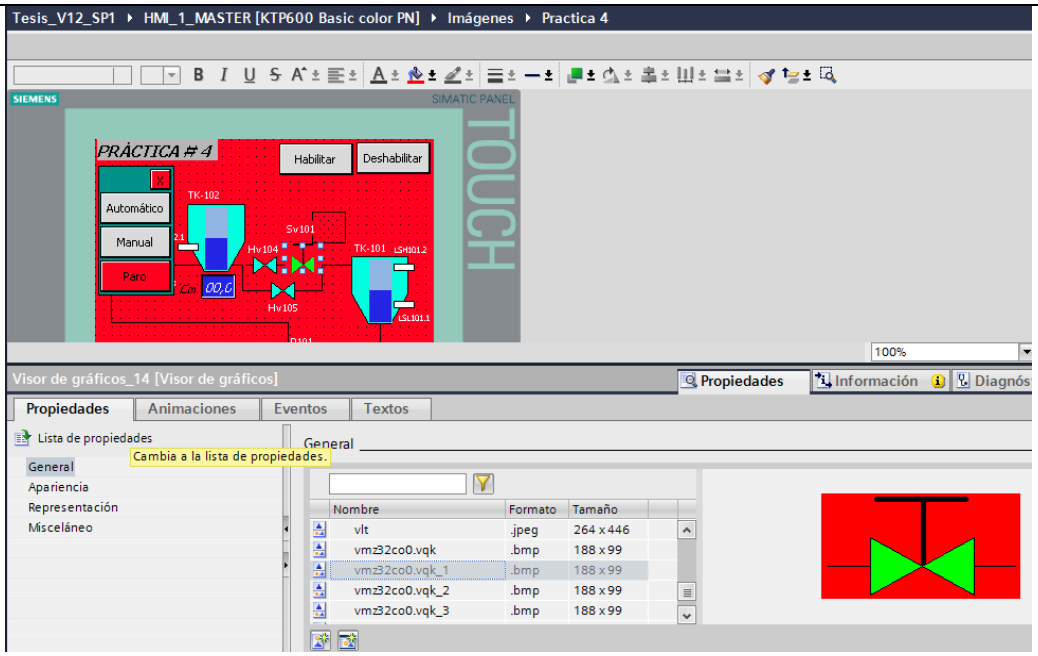


Figura 202: Configuración de botón electroválvula.

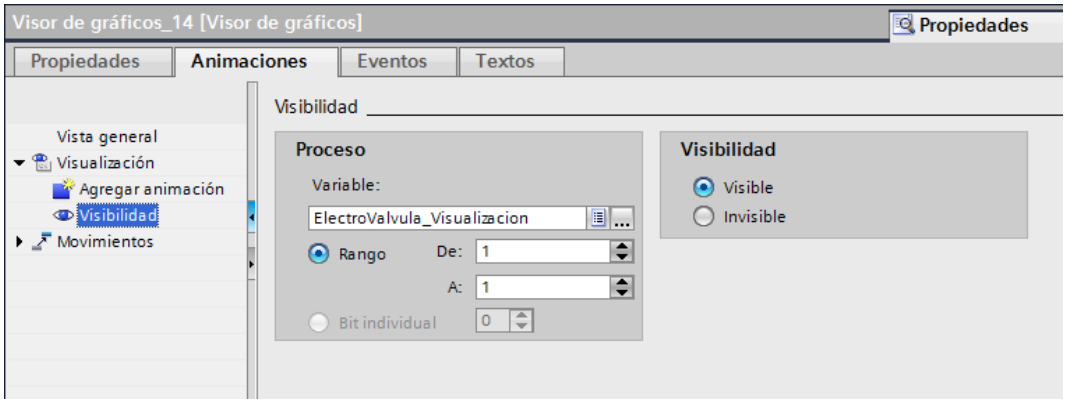


Figura 203: Animación de botón electroválvula.

Ahora detallaremos la configuración del botón para la bomba manual.

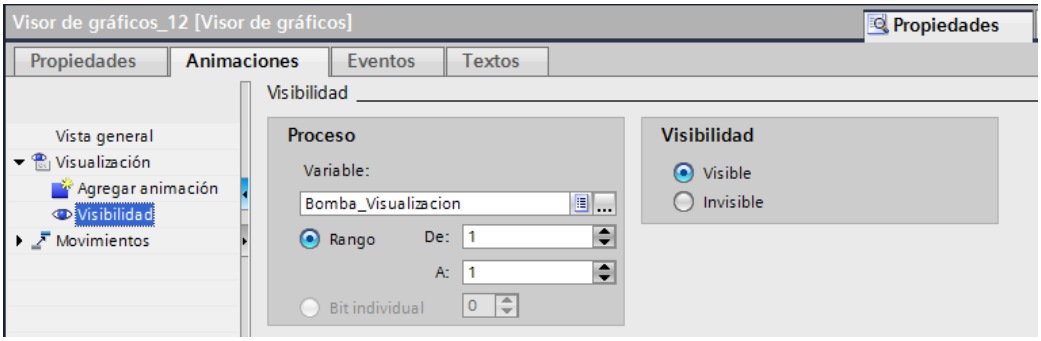


Figura 204: Animación de botón bomba.

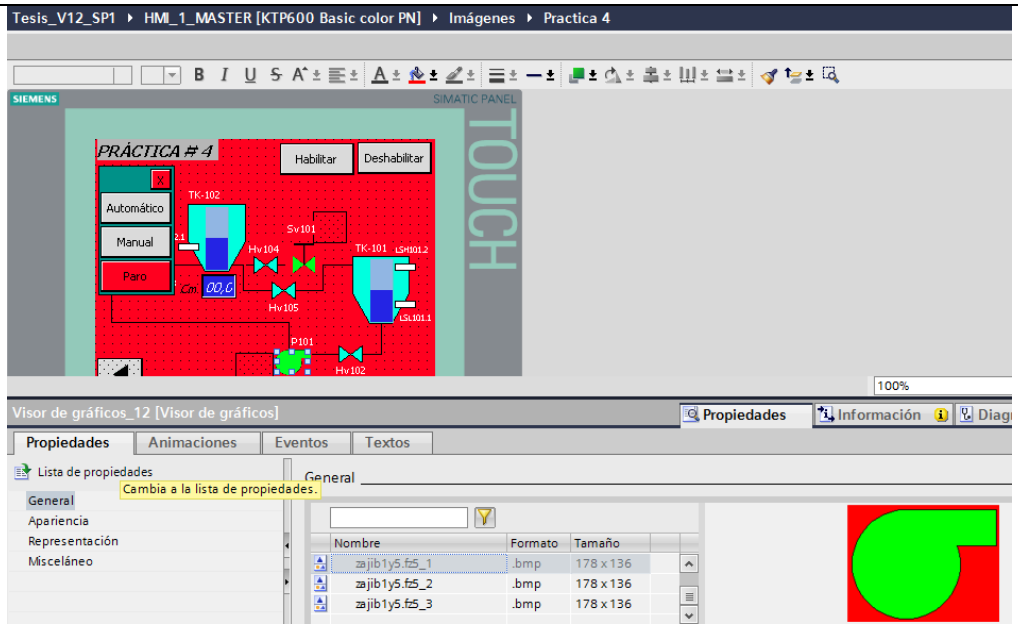


Figura 205: Configuración de botón bomba.

Se añadió también los sensores de nivel de la boya LSL102.1 del tanque TK102 y los sensores de nivel alto y bajo LSH101.2 / LSL101.1 del tanque TK101.

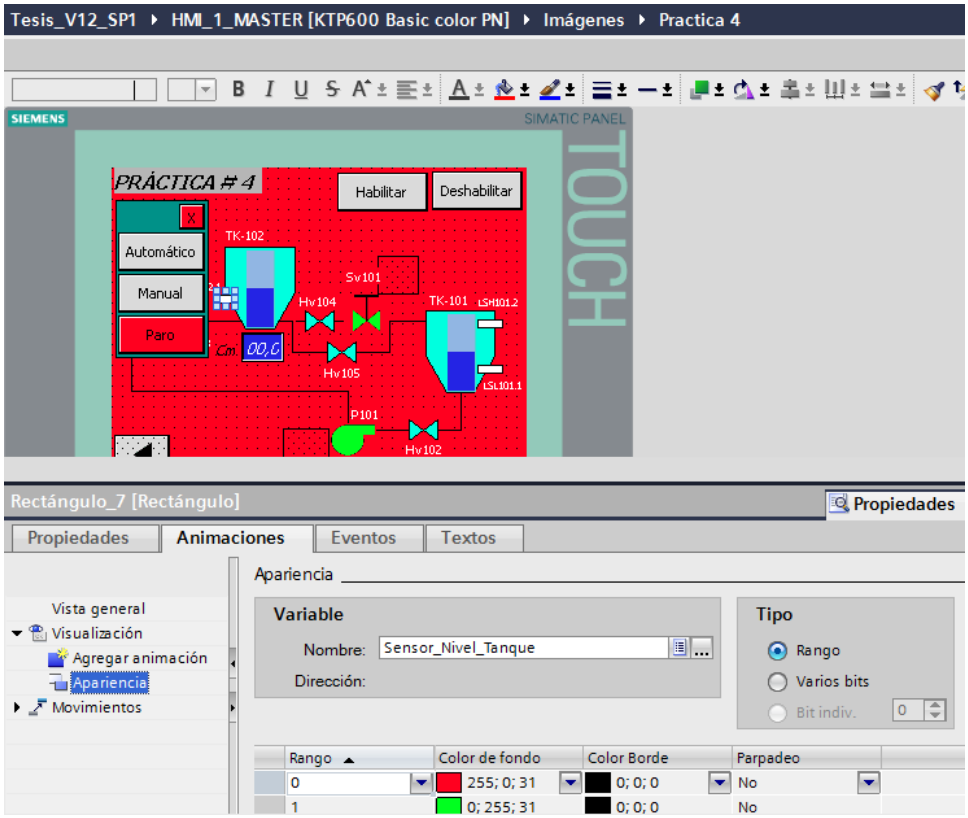


Figura 206: Representación de sensor de boya.

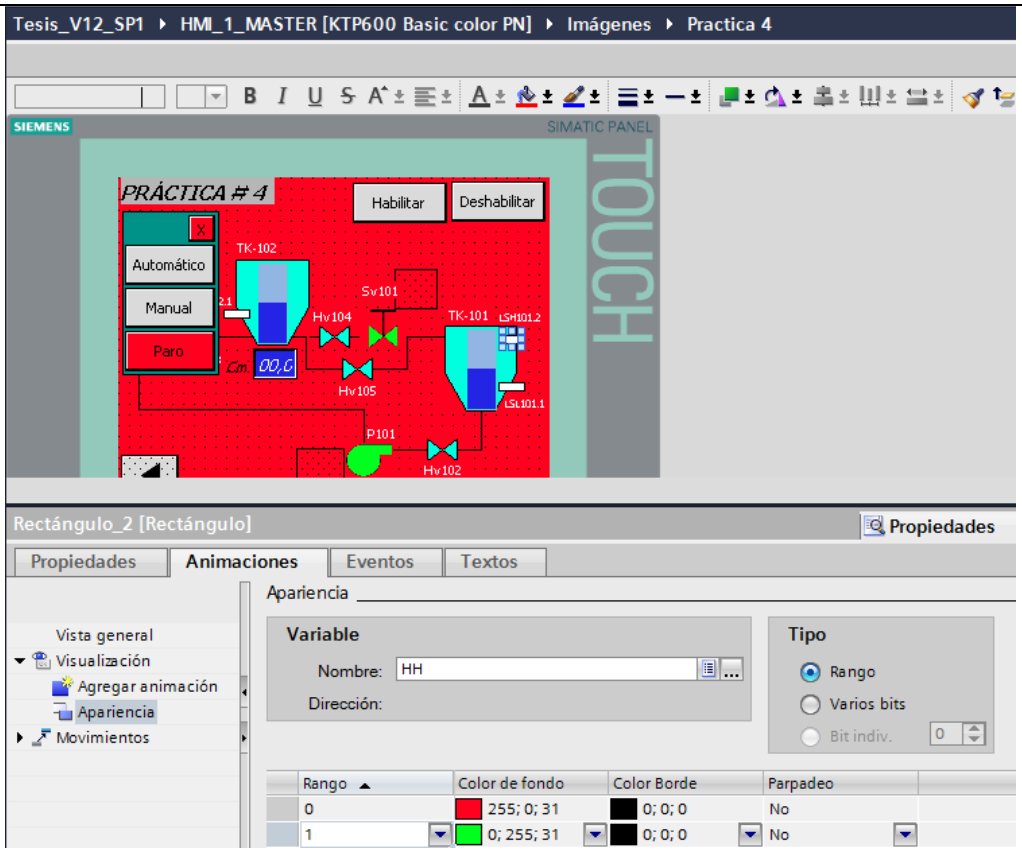


Figura 207: Representación de sensor de nivel alto.

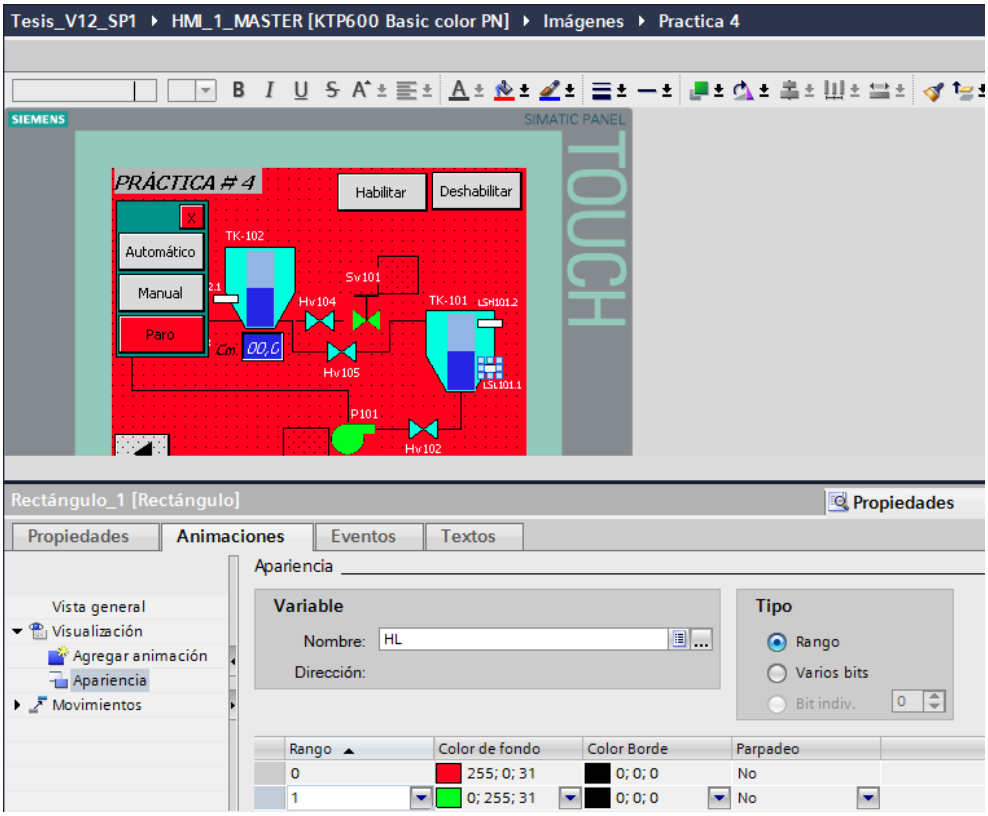


Figura 208: Representación de sensor de nivel bajo.

Observaciones: Respecto a los botones **Habilitar**, **Deshabilitar**, y los **mensajes de alerta** se debe seguir las instrucciones del **Paso 20** de la **Práctica 1**.
Recomendaciones: • Comprobar que las variables estén identificadas y con nombre para cada parte del proceso. • Revisar que las variables de los controles del HMI estén correctamente asignadas. • Verificar que los datos enviados y recibidos se guarden en las respectivas variables asignadas. • Conectar de forma adecuada la planta de nivel de agua, sensores, alimentación, entrada y salida.

4.5. Práctica Nº 5

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>5</td><td>TÍTULO PRÁCTICA: “Parametrización de un variador de frecuencia ABB ACS355.”</td></tr>
<tr><td colspan="4">OBJETIVOS:

• Diseñar un sistema de visualización para la parametrización de variadores de frecuencia ABB modelo ACS355.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 / 240 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:

• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• Dos módulos de comunicación CM1241.
• Un cable de comunicación RS485 macho – macho.
• Un módulo adaptador FMBA-01 para MODBUS RTU.
• Un variador de frecuencia ABB modelo ACS355.

DESCRIPCIÓN:

El sistema en mención está diseñado para visualizar los parámetros configurados en el variador de frecuencia, para esto se configuró al PLC como maestro y el variador de frecuencia como esclavo Modbus. La pantalla HMI cumplirá la función de mostrar los parámetros obtenidos del variador al PLC.

La figura 209 muestra la arquitectura de red a implementar.</td></tr>
</table>

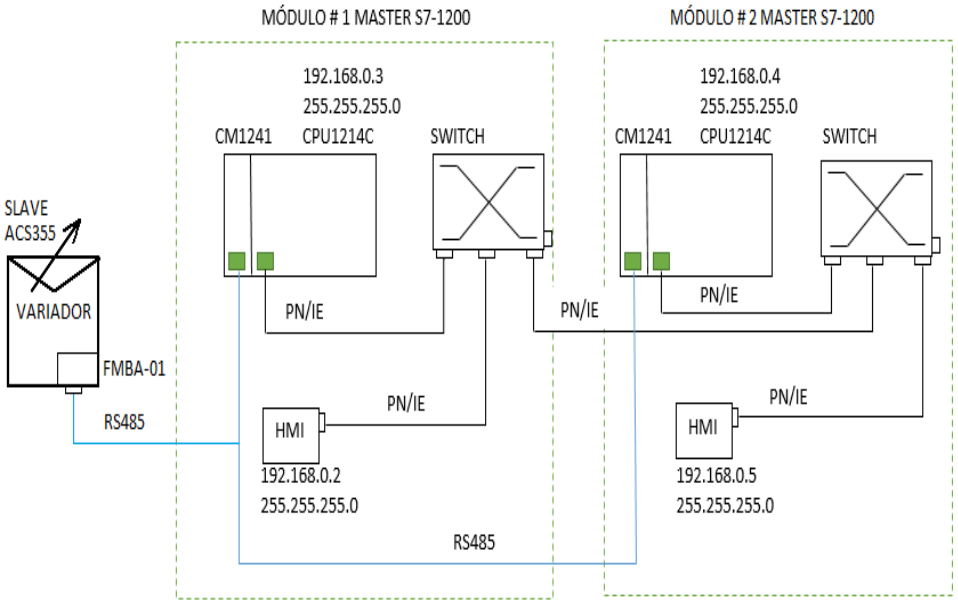


Figura 209: Arquitectura de red.

En la figura 210 se muestra el circuito de fuerza del variador y motor trifásico.

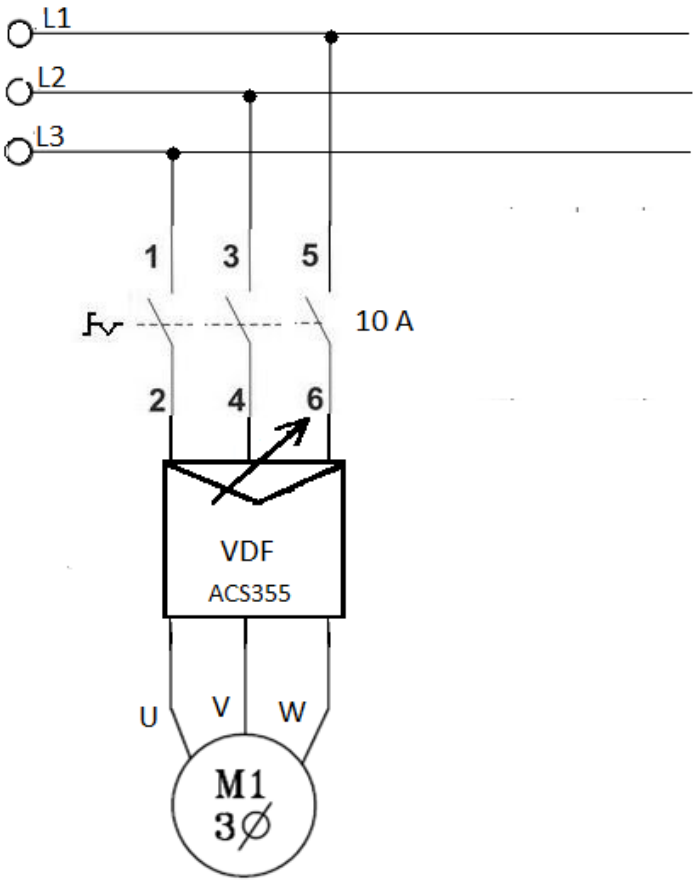


Figura 210: Circuito de fuerza.

Paso 1.

Procedemos a generar los parámetros para las instrucciones de la práctica 5, nos dirigimos a **PARAMETERS [DB6]** ubicado en el PLC **MASTER** y creamos las siguientes variables:

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB6]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor de a...
1	Static							
2	P1	Struct	0.0					
3	P2	Struct	12.0					
4	P3	Struct	32.0					
5	P4	Struct	42.0					
6	P5	Struct	62.0					
7	MB_ADDR 5202	UInt	0.0	10				
8	MODE 5202	USInt	2.0	0				
9	DATA_ADDR 5202	UDInt	4.0	45302				
10	DATA_LEN 5202	UInt	8.0	4				
11	MB_ADDR 5310	UInt	10.0	10				
12	MODE 5310	USInt	12.0	0				
13	DATA_ADDR 5310	UDInt	14.0	45310				
14	DATA_LEN 5310	UInt	18.0	8				
15	MB_ADDR 1102	UInt	20.0	10				
16	MODE 1102	USInt	22.0	0				
17	DATA_ADDR 1102	UDInt	24.0	41102				
18	DATA_LEN 1102	UInt	28.0	4				
19	MB_ADDR 9905	UInt	30.0	10				
20	MODE 9905	USInt	32.0	0				
21	DATA_ADDR 9905	UDInt	34.0	49905				
22	DATA_LEN 9905	UInt	38.0	4				

Figura 211: Parámetros del PLC MASTER, práctica 5.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB11]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor de a...
1	Static							
2	P1	Struct	...					
3	P3	Struct	...					
4	P4	Struct	...					
5	P5	Struct	...					
6	MB_ADDR 5202	UInt	...	10				
7	MODE 5202	USInt	...	0				
8	DATA_ADDR 5202	UDInt	...	45302				
9	DATA_LEN 5202	UInt	...	4				
10	MB_ADDR 5310	UInt	...	10				
11	MODE 5310	USInt	...	0				
12	DATA_ADDR 5310	UDInt	...	45310				
13	DATA_LEN 5310	UInt	...	8				
14	MB_ADDR 1102	UInt	...	10				
15	MODE 1102	USInt	...	0				
16	DATA_ADDR 1102	UDInt	...	41102				
17	DATA_LEN 1102	UInt	...	4				
18	MB_ADDR 9905	UInt	...	10				
19	MODE 9905	USInt	...	0				
20	DATA_ADDR 9905	UDInt	...	49905				
21	DATA_LEN 9905	UInt	...	4				

Figura 212: Parámetros del PLC SLAVE, práctica 5.

Paso 2.

Para guardar los datos de envío y recepción, creamos las variables en **PRACTICAS [DB4].** La imagen a continuación mostrará las variables de registro de datos para el PLC **MASTER**.

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5302[...	Int	0.0	0				
9	PRACTICA 5_5302[...	Int	2.0	0				
10	PRACTICA 5_5302[...	Int	4.0	0				
11	PRACTICA 5_5302[...	Int	6.0	0				
12	PRACTICA 5_5310	Array [1..8] of Int	54.0					
13	PRACTICA 5_5310[...	Int	0.0	0				
14	PRACTICA 5_5310[...	Int	2.0	0				
15	PRACTICA 5_5310[...	Int	4.0	0				
16	PRACTICA 5_5310[...	Int	6.0	0				
17	PRACTICA 5_5310[...	Int	8.0	0				
18	PRACTICA 5_5310[...	Int	10.0	0				
19	PRACTICA 5_5310[...	Int	12.0	0				
20	PRACTICA 5_5310[...	Int	14.0	0				
21	PRACTICA 5_1102	Array [1..4] of Int	70.0					
22	PRACTICA 5_1102[...	Int	0.0	0				
23	PRACTICA 5_1102[...	Int	2.0	0				
24	PRACTICA 5_1102[...	Int	4.0	0				
25	PRACTICA 5_1102[...	Int	6.0	0				
26	PRACTICA 5_9905	Array [1..4] ...	78.0					
27	PRACTICA 5_9905[...	Int	0.0	0				
28	PRACTICA 5_9905[...	Int	2.0	0				
29	PRACTICA 5_9905[...	Int	4.0	0				
30	PRACTICA 5_9905[...	Int	6.0	0				

Figura 213: Variables de registro para el PLC MASTER – práctica 5.

Continuamos con la creación de las variables de registro para el PLC **SLAVE**.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5302[...	Int	0.0	0				
9	PRACTICA 5_5302[...	Int	2.0	0				
10	PRACTICA 5_5302[...	Int	4.0	0				
11	PRACTICA 5_5302[...	Int	6.0	0				
12	PRACTICA 5_5310	Array [1..8] of Int	54.0					
13	PRACTICA 5_5310[...	Int	0.0	0				
14	PRACTICA 5_5310[...	Int	2.0	0				
15	PRACTICA 5_5310[...	Int	4.0	0				
16	PRACTICA 5_5310[...	Int	6.0	0				
17	PRACTICA 5_5310[...	Int	8.0	0				
18	PRACTICA 5_5310[...	Int	10.0	0				
19	PRACTICA 5_5310[...	Int	12.0	0				
20	PRACTICA 5_5310[...	Int	14.0	0				
21	PRACTICA 5_1102	Array [1..4] of Int	70.0					
22	PRACTICA 5_1102[...	Int	0.0	0				
23	PRACTICA 5_1102[...	Int	2.0	0				
24	PRACTICA 5_1102[...	Int	4.0	0				
25	PRACTICA 5_1102[...	Int	6.0	0				
26	PRACTICA 5_9905	Array [1..4] ...	78.0					
27	PRACTICA 5_9905[...	Int	0.0	0				
28	PRACTICA 5_9905[...	Int	2.0	0				
29	PRACTICA 5_9905[...	Int	4.0	0				
30	PRACTICA 5_9905[...	Int	6.0	0				

Figura 214: Variables de registro para el PLC SLAVE – práctica 5.

Paso 3.

Procedemos a agregar los bloques de instrucciones para la comunicación, para esta práctica se debemos utilizar 4 bloques master que estarán configurados en modo lectura. Cada bloque manejara distintos parámetros. A continuación se detalla la configuración de los bloques:

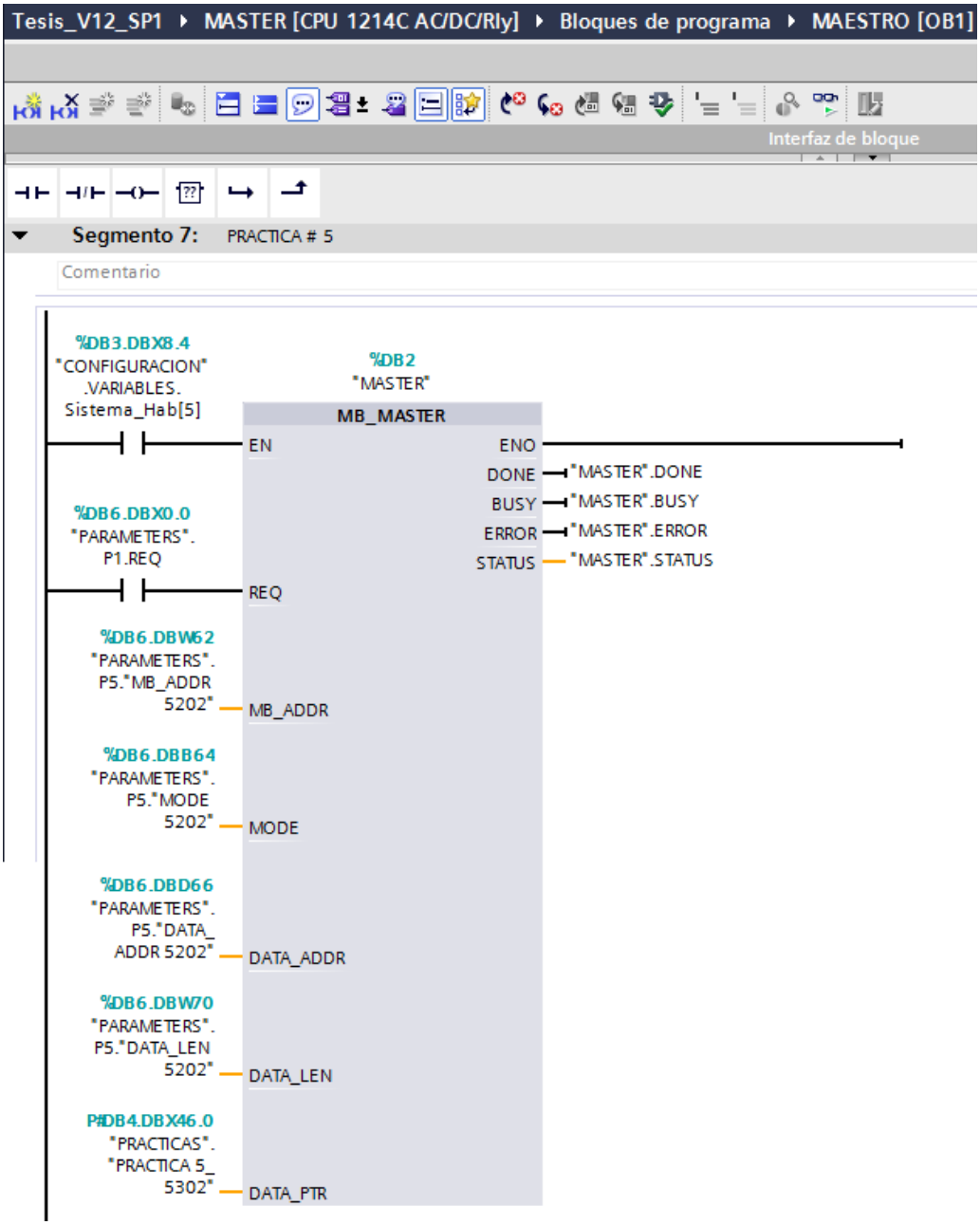


Figura 215: Bloque Master de parámetro 5302.

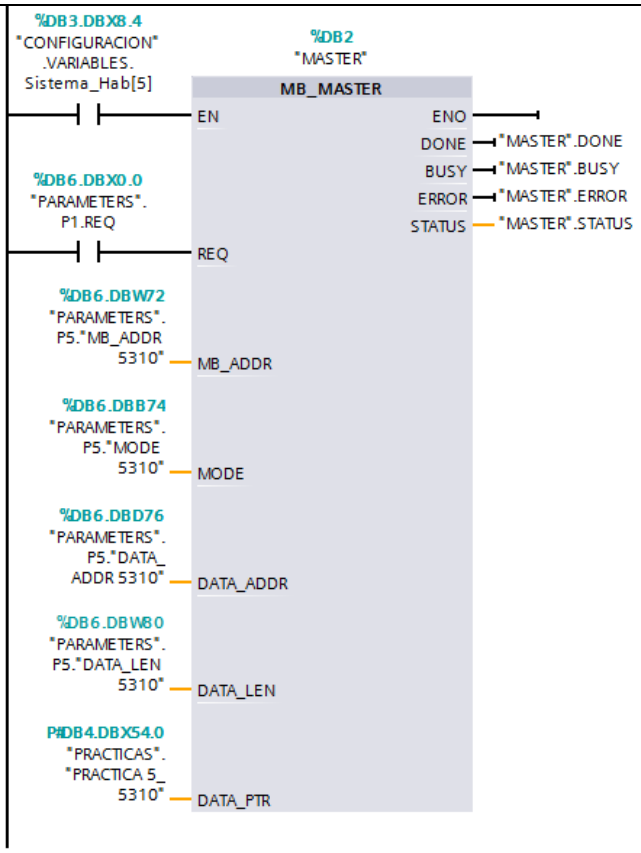


Figura 216: Bloque Master de parámetro 5310.

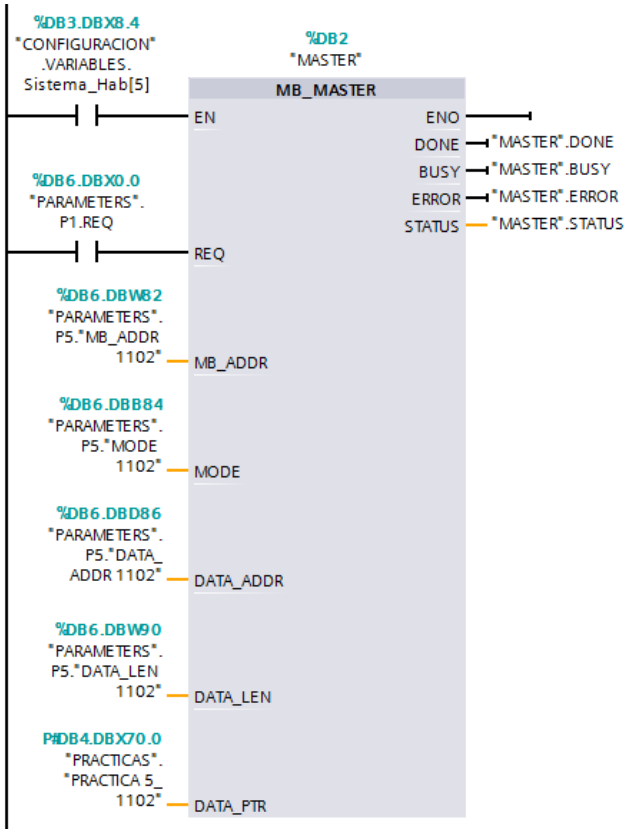


Figura 217: Bloque Master de parámetro 1102.

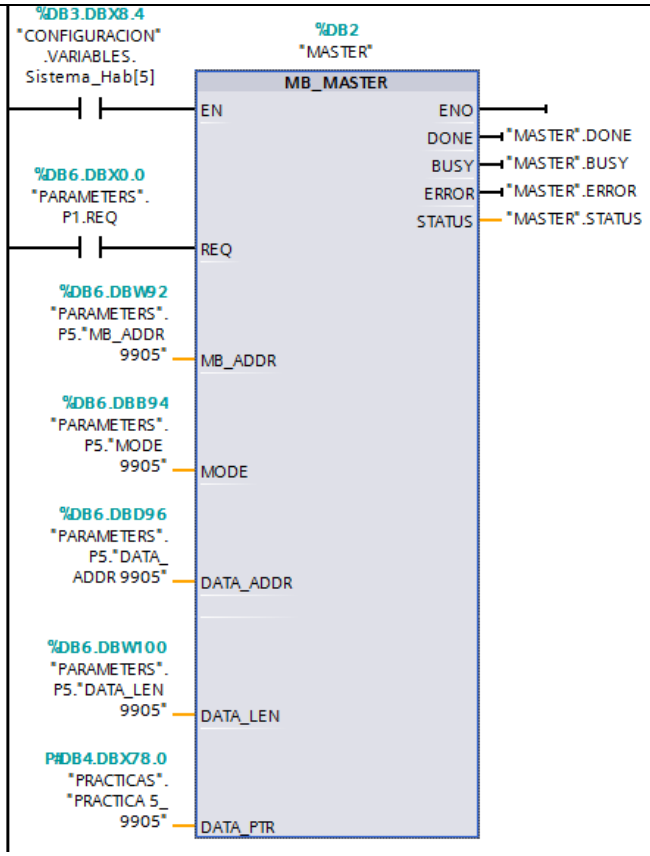


Figura 218: Bloque Máster de parámetro 1102.

Ahora detallaremos a que hace referencia cada parámetro configurado para la comunicación Modbus RTU en el variador de frecuencia ACS355, según el manual del fabricante ABB.

Todos los parámetros			
N.º	**Nombre/Valor**	**Descripción**	**Def./FbEq**
14 SALIDAS DE RELE		Información de estado indicada a través de las salidas de relé y las demoras de funcionamiento del relé. **Nota:** Las salidas de relé 2…4 sólo están disponibles si el módulo de salidas de relé MREL-01 está conectado al convertidor. Véase el *MREL-01 output relay module user's manual* (3AUA0000035974 [Inglés]).	
1401	SALIDA RELE SR1	Selecciona un estado del convertidor indicado mediante la salida de relé SR 1. El relé se excita cuando el estado coincide con el valor ajustado.	*FALLO (-1)*
	SIN SEL	Sin usar.	0
	LISTO	Listo para funcionar: Señal de Permiso de marcha activada, sin fallos, tensión de alimentación dentro del rango aceptable y señal de paro de emergencia desactivada.	1
1202	VELOC CONST 1	Define la velocidad constante 1 (o la frecuencia de salida del convertidor).	E: 5,0 Hz U: 6,0 Hz
	0,0…600,0 Hz / 0…30000 rpm	Velocidad en rpm. Frecuencia de salida (Hz) si el ajuste del parámetro *9904 MODO CTRL MOTOR* es *ESCALAR: FREC*.	1 = 0,1 Hz / 1 rpm
0101	VELOCIDAD & DIR	Velocidad calculada del motor en rpm. Un valor negativo indica dirección de retroceso.	1 = 1 rpm

Figura 219: Parámetros de control del variador ACS355 [18].

Parámetro	Ajustes alternativos	Ajuste para control por bus de campo	Función / información
5302 ID ESTACION BCI	0…247	Cualquiera	Define la dirección de la ID de estación del enlace RS-232/EIA-485. Dos estaciones en línea no pueden tener la misma dirección.
5303 VEL TRANSM BCI	1,2 kbit/s 2,4 kbit/s 4,8 kbit/s 9,6 kbit/s 19,2 kbit/s 38,4 kbit/s 57,6 kbit/s 115,2 kbit/s		Define la velocidad de comunicación del enlace RS-232/EIA-485.
5304 PARIDAD BCI	*8N1* *8N2* *8E1* *8O1*		Selecciona el ajuste de paridad. Deben utilizarse los mismos ajustes en todas las estaciones en línea.
5305 PERFIL CTRL BCI	*ABB DRV LIM* *DCU PROFILE* *ABB DRV FULL*	Cualquiera	Selecciona el perfil de comunicación utilizado por el convertidor. Véase el apartado *Perfiles de comunicación* en la página *351*.

Figura 220: Configuración de la comunicación Modbus 5302 [18].

Todos los parámetros		
N.º Nombre/Valor	**Descripción**	**Def./FbEq**
5310 PAR BCI 10	Selecciona un valor actual para correlacionarlo con el registro 40005 del Modbus.	0
0…65535	Índice de parámetro	1 = 1
5311 PAR BCI 11	Selecciona un valor actual para correlacionarlo con el registro 40006 del Modbus.	0
0…65535	Índice del parámetro	1 = 1
5312 PAR BCI 12	Selecciona un valor actual para correlacionarlo con el registro 40007 del Modbus.	0
0…65535	Índice del parámetro	1 = 1
5313 PAR BCI 13	Selecciona un valor actual para correlacionarlo con el registro 40008 del Modbus.	0
0…65535	Índice del parámetro	1 = 1
5314 PAR BCI 14	Selecciona un valor actual para correlacionarlo con el registro 40009 del Modbus.	0
5315 PAR BCI 15	Selecciona un valor actual para correlacionarlo con el registro 40010 del Modbus.	0
0…65535	Índice del parámetro	1 = 1
5316 PAR BCI 16	Selecciona un valor actual para correlacionarlo con el registro 40011 del Modbus.	0
0…65535	Índice del parámetro	1 = 1
5317 PAR BCI 17	Selecciona un valor actual para correlacionarlo con el registro 40012 del Modbus.	0

Figura 221: Parámetros de comunicación Modbus 5310 [18].

Parámetro		Ajuste para el control por bus de campo	Función / información	Dirección de registro Modbus	
1102	*SELEC EXT1/EXT2*	*COMUNIC*	Habilita la selección de EXT1/EXT2 mediante *0301 COD ORDEN BC 1*, bit 5 (*EXT2*); con el perfil ABB Drives *Código de control del perfil ABB Drives, parámetro 5319 PAR BCI 19 Código de control del perfil ABB Drives, parámetro 5319 PAR BCI 19*, bit 11 (*EXT CTRL LOC*).	40001 bit 11	40031 bit 5
1103	*SELEC REF1*	*COMUNIC COMUNIC+ EA1 COMUNIC* EA1*	La referencia de bus de campo REF1 se usa cuando se selecciona EXT1 como el lugar de control activo. Véase la sección *Referencias del bus de campo* en la página *343* para obtener información acerca de los ajustes alternativos.	40002 para REF1	
1106	*SELEC REF2*	*COMUNIC COMUNIC+ EA1 COMUNIC* EA1*	La referencia de bus de campo REF2 se usa cuando se selecciona EXT2 como el lugar de control activo. Véase el apartado *Referencias del bus de campo* en la página *343* para obtener información acerca de los ajustes alternativos.	40003 para REF2	

Figura 222: Parámetros de comunicación Modbus 1102 [18].

N.º	Nombre	Valor	Descripción
9903	*TIPO MOTOR*	2	Motor síncrono de imanes permanentes
9904	*MODO CTRL MOTOR*	1 2	VECTOR:VELOC VECTOR:PAR **Nota**: También se puede seleccionar el modo de control escalar (3) pero no se recomienda, porque en este modo de control, el motor síncrono de imanes permanentes puede volverse inestable y dañar el proceso, el motor o el convertidor.
9905	*TENSION NOM MOT*		**Nota**: Si no se dispone de la tensión de la fuerza contraelectromotriz del motor, ajuste el valor nominal aquí y realice una marcha de ID. Si la tensión se facilita como un valor proporcional, como 103 V/1000 rpm en un motor de 3000 rpm, ajuste a 309 V. A veces el valor se facilita como el valor pico. En este caso, divida el valor por la raíz cuadrada de 2 (1,41). **Nota**: Se recomienda utilizar la tensión de la fuerza contraelectromotriz. Si no se usa, deberá realizarse una marcha de ID completa.
9906	*INTENS NOM MOT*		Intensidad nominal del motor. No use el valor pico.
9907	*FREC NOM MOTOR*		Frecuencia eléctrica nominal del motor. Si la frecuencia no aparece en la placa de características del motor, puede calcularse con ayuda de la siguiente formula: frecuencia [Hz] = velocidad [rpm] x (número de pares de polos) / 60
9908	*VELOC NOM MOTOR*		Velocidad mecánica nominal del motor. Si ni se proporciona, puede calcularse con ayuda de la siguiente formula: velocidad [rpm] = frecuencia [Hz] x 60 / (número de pares de polos)
9909	*POT NOM MOTOR*		Potencia nominal del motor. Si ni se proporciona, puede calcularse con ayuda de la siguiente formula: potencia [kW] = par nominal [Nm] x 2 x pi x velocidad nominal [rpm] / 60000

Figura 223: Parámetros necesarios para motores síncronos de imanes permanentes [18].

Paso 5.

En este paso usamos la función MOVE_BLK la cual se encargará de borrar los parámetros una vez que se deshabilite la comunicación, la función MUL se encarga de multiplicar los baudios obtenidos del variador x 100.

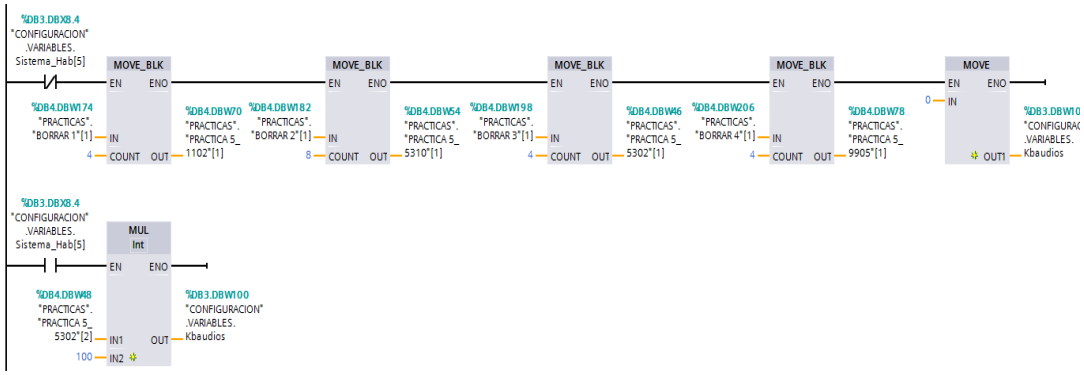


Figura 224: Puesta a cero de los controles de visualización.

Paso 6.

Daremos paso a la creación de las imágenes de la práctica 5. A continuación se detalla el diseño de la imagen **Practica 5**.

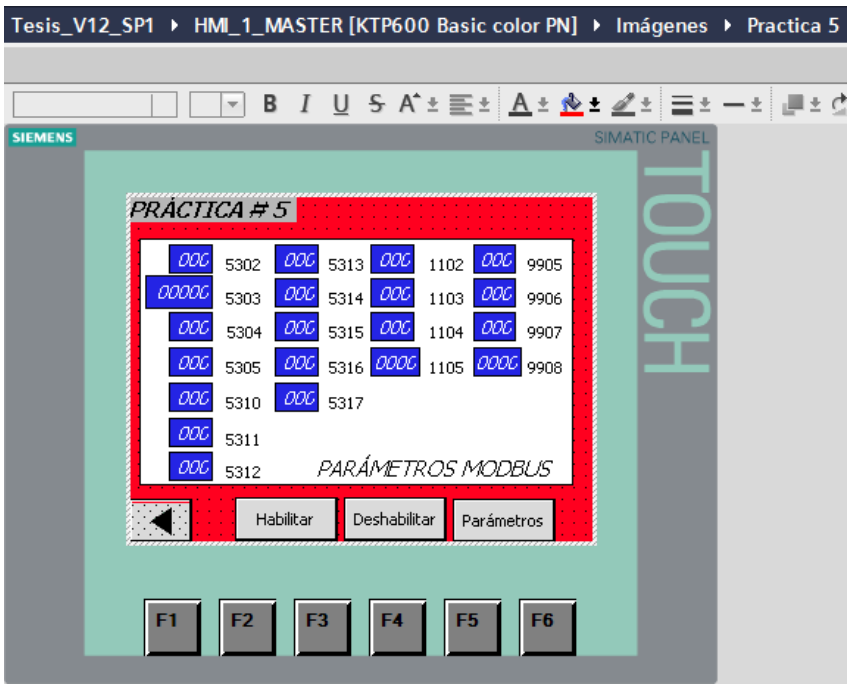


Figura 225: Creación de imagen Práctica 5.

Crearemos una segunda imagen donde constara una breve descripción de cada parámetro y que sea apreciable al usuario. La imagen **Parámetros_VLT** constará de 5 cuadros de texto que se desplazarán por medio de un botón.

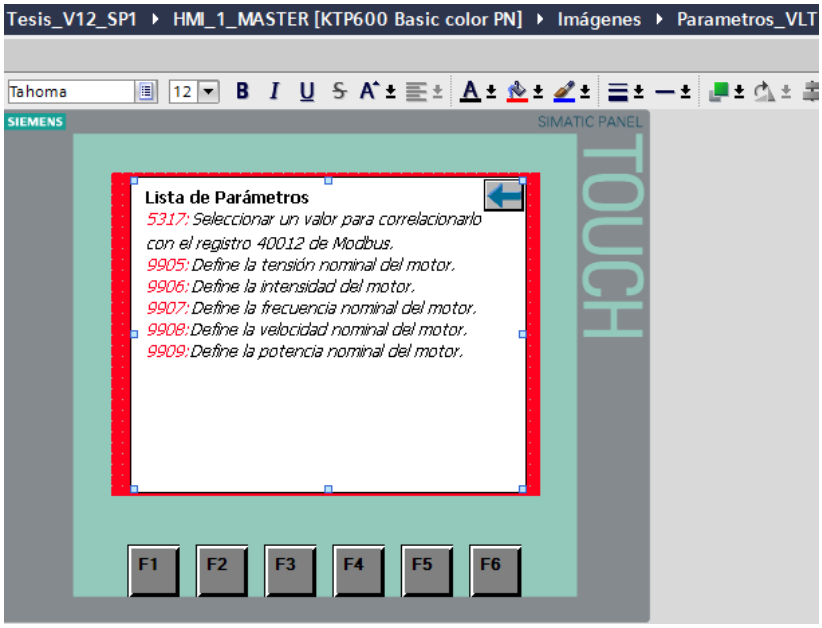


Figura 226: Creación de imagen Parametros_VLT.

La lista de parámetros es un grupo que está conformado por 3 rectángulos que al interior se conforma con cuadros de texto donde se describen los parámetros. Un botón en cada rectángulo para pasar de uno a otro. A continuación se muestra la asignación de variables y programación:

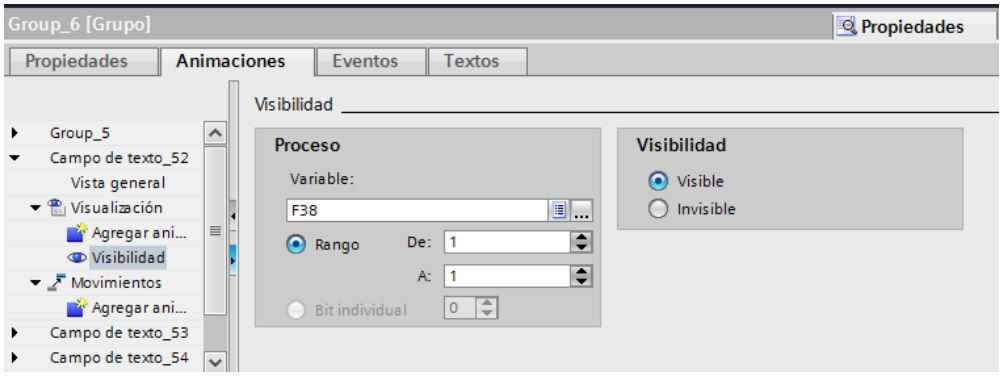


Figura 227: Animación y variable de lista de parámetros.

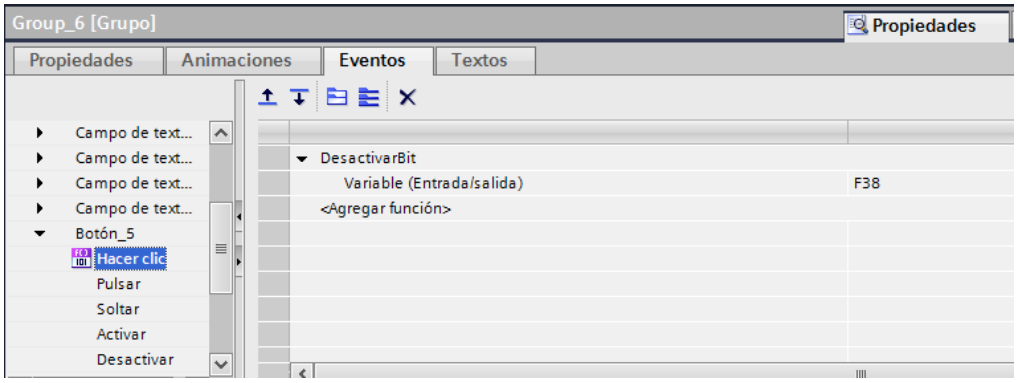


Figura 228: Eventos y variable de lista de parámetros.

Recomendaciones.

- Verificar los pines de comunicación entre el puerto DB9 del módulo CM124 y el módulo FMBA-01.
- Comprobar la correcta lectura de los parámetros del variador de frecuencia.
- Colocar en modo remoto el variador para una efectiva comunicación.

4.6. Práctica Nº 6

<table>
<tr><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td>CARRERA: Ingeniería Electrónica</td><td>ASIGNATURA: Automatización I</td></tr>
<tr><td>NRO. PRÁCTICA: 6</td><td>TÍTULO PRÁCTICA: “Configuración MODBUS RTU Esclavo – S7-1200 con CM 1241 RS485.”</td></tr>
<tr><td colspan="2">OBJETIVOS:

• Establecer la comunicación maestro – esclavo entre PLC y PC.
• Configurar al PLC como esclavo Modbus RTU.
• Adaptar el software Modbus Poll para la recepción de datos.</td></tr>
<tr><td rowspan="4">INSTRUCCIONES:</td><td>1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td>2. Instalar correctamente los softwares a utilizar con sus respectivos drivers.</td></tr>
<tr><td>3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td>4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="2">MATERIALES:

• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• PC con programa Modbus Poll.
• Cable convertidor USB a RS485.

DESCRIPCIÓN:

Se debe enviar 10 datos desde el PLC hacia la PC donde se deben guardar los registros enviados, para esto se debe configurar la PC como maestro y al PLC como esclavo. Se podrá recibir la información de cualquiera de los dos módulos, uno a la vez pero con una señal de habilitación para establecer la comunicación Modbus RTU a través del cable RS485.

La figura 229 muestra la arquitectura de red a implementar.</td></tr>
</table>

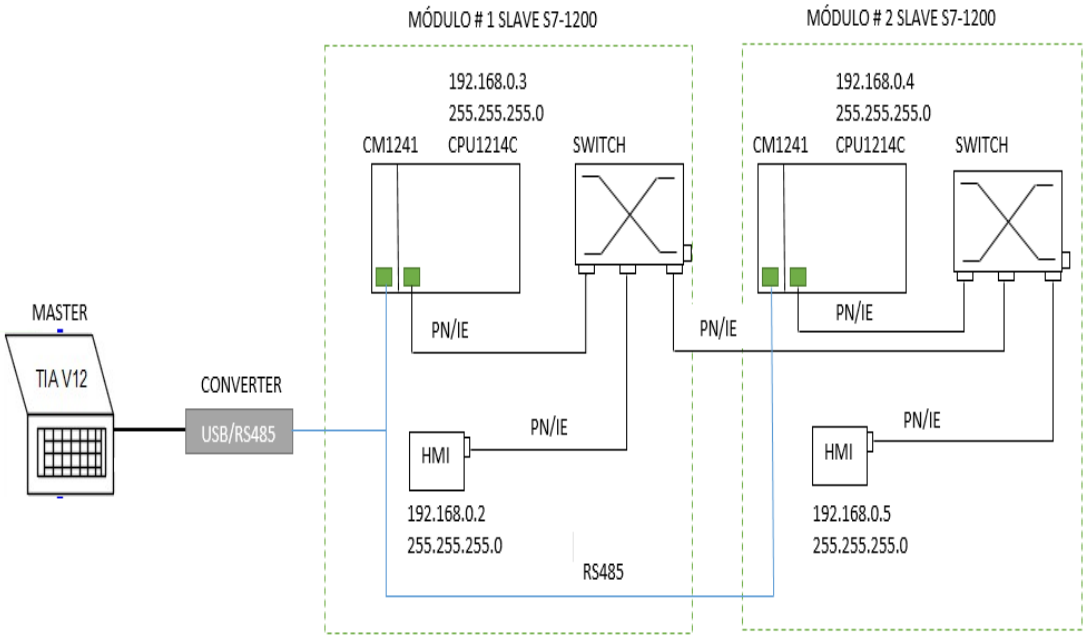


Figura 229: Arquitectura de red.

En figura 230 se muestra el diagrama de conexiones a implementar.

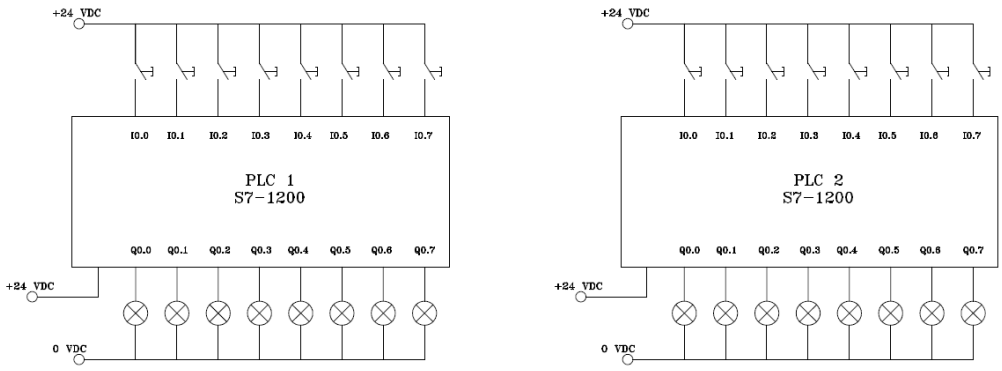


Figura 230: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear los parámetros para la práctica 6 para el PLC MASTER y el PLC SLAVE, donde vamos a configurar a los bloques maestros en modo escritura.

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB6]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	P1	Struct	0.0					
3	P2	Struct	12.0					
4	P3	Struct	32.0					
5	P4	Struct	42.0					
6	P5	Struct	62.0					
7	P6	Struct	102.0					
8	MB_ADDR	UInt	0.0	6				
9	MODE	USInt	2.0	1				
10	DATA_ADDR	UDInt	4.0	40001				
11	DATA_LEN	UInt	8.0	10				

Figura 231: Parámetros del PLC MASTER, práctica 6.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB11]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	P1	Struct	...					
3	P3	Struct	...					
4	P4	Struct	...					
5	P5	Struct	...					
6	P6	Struct	...					
7	MB_ADDR	UInt	...	6				
8	MODE	USInt	...	1				
9	DATA_ADDR	UDInt	...	40001				
10	DATA_LEN	UInt	...	10				

Figura 232: Parámetros del PLC SLAVE, práctica 6.

Paso 2.

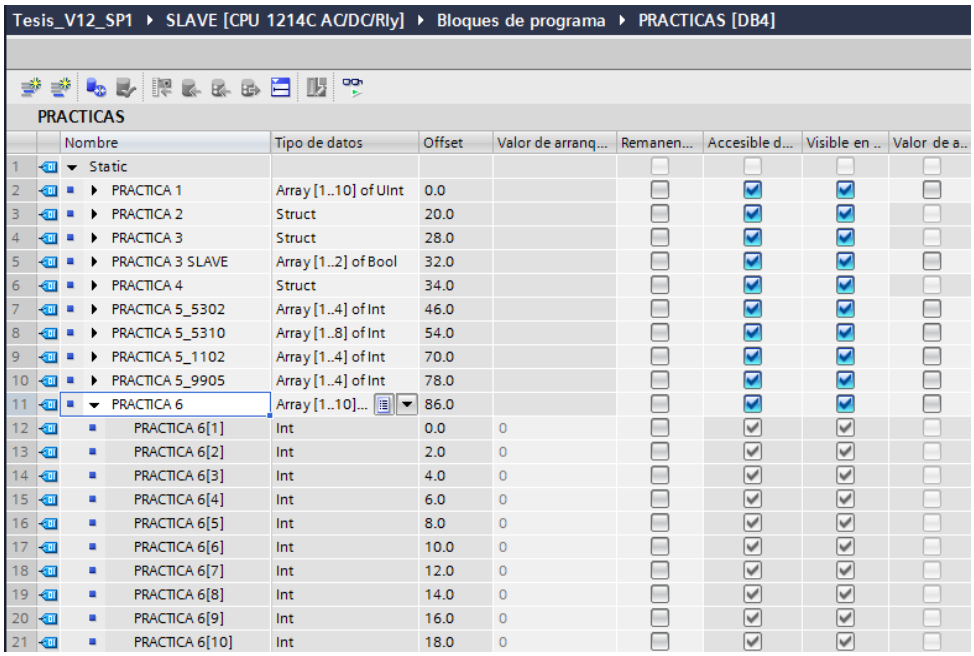
Ahora procedemos a crear las variables donde se registraran los 10 datos a enviar hacia la PC, se detalla a continuación.

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
4	PRACTICA 3	Struct	28.0					
5	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
6	PRACTICA 4	Struct	34.0					
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5310	Array [1..8] of Int	54.0					
9	PRACTICA 5_1102	Array [1..4] of Int	70.0					
10	PRACTICA 5_9905	Array [1..4] of Int	78.0					
11	PRACTICA 6	Array [1..10]...	86.0					
12	PRACTICA 6[1]	Int	0.0	0				
13	PRACTICA 6[2]	Int	2.0	0				
14	PRACTICA 6[3]	Int	4.0	0				
15	PRACTICA 6[4]	Int	6.0	0				
16	PRACTICA 6[5]	Int	8.0	0				
17	PRACTICA 6[6]	Int	10.0	0				
18	PRACTICA 6[7]	Int	12.0	0				
19	PRACTICA 6[8]	Int	14.0	0				
20	PRACTICA 6[9]	Int	16.0	0				
21	PRACTICA 6[10]	Int	18.0	0				

Figura 233: Variables de registro para el PLC MASTER – práctica 6.



	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	PRACTICA 3	Struct	28.0					
5	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
6	PRACTICA 4	Struct	34.0					
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5310	Array [1..8] of Int	54.0					
9	PRACTICA 5_1102	Array [1..4] of Int	70.0					
10	PRACTICA 5_9905	Array [1..4] of Int	78.0					
11	PRACTICA 6	Array [1..10]...	86.0					
12	PRACTICA 6[1]	Int	0.0	0				
13	PRACTICA 6[2]	Int	2.0	0				
14	PRACTICA 6[3]	Int	4.0	0				
15	PRACTICA 6[4]	Int	6.0	0				
16	PRACTICA 6[5]	Int	8.0	0				
17	PRACTICA 6[6]	Int	10.0	0				
18	PRACTICA 6[7]	Int	12.0	0				
19	PRACTICA 6[8]	Int	14.0	0				
20	PRACTICA 6[9]	Int	16.0	0				
21	PRACTICA 6[10]	Int	18.0	0				

Figura 234: Variables de registro para el PLC SLAVE – práctica 6.

Paso 3.

Procedemos a crear los bloques de instrucción para la comunicación, se necesita de un bloque MB_SLAVE por cada PLC, a continuación se muestra la configuración:

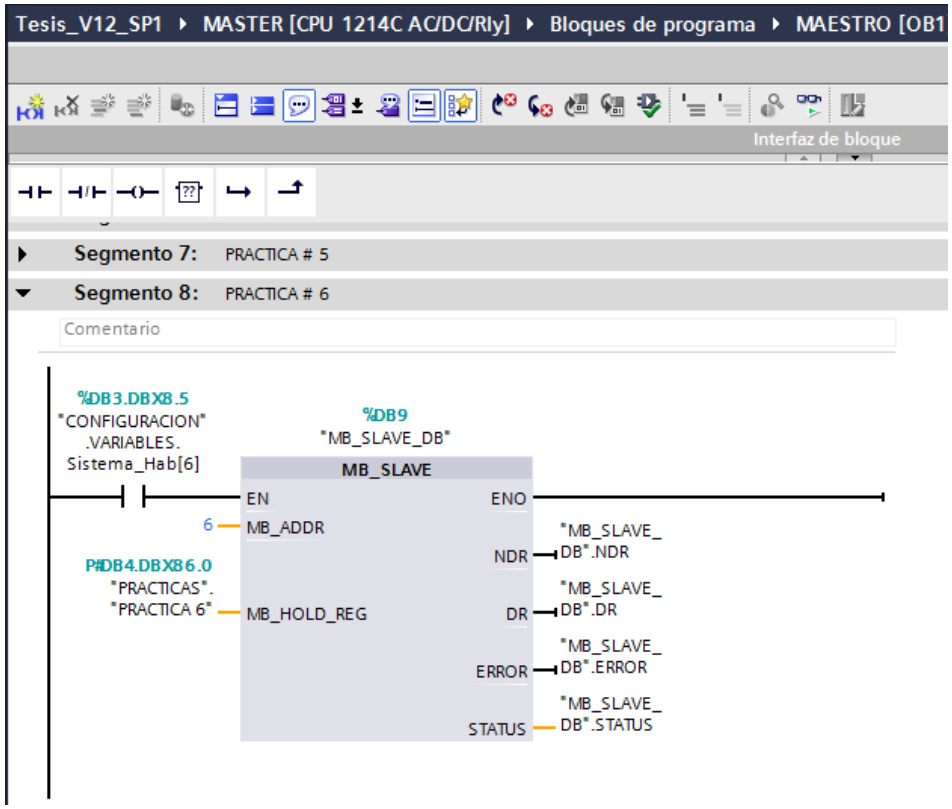


Figura 235: Configuración del bloque escritura MB_SLAVE/ PLC MASTER práctica 6.

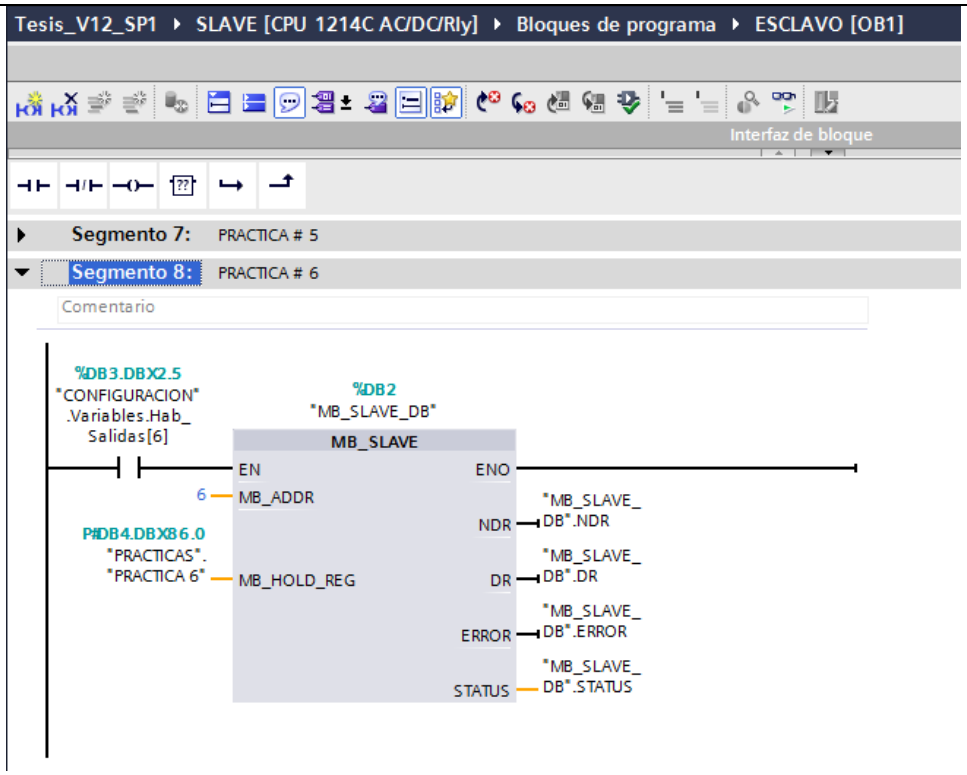


Figura 236: Configuración del bloque escritura MB_SLAVE/ PLC SLAVE práctica 6.

Paso 4.

Aquí debemos crear la imagen Título 6, ver figura 237.

Figura 237: Diseño de imagen Título 6, práctica 6.

Observación: para la creación de la imagen **Titulo 2** se debe seguir el proceso del

paso 19, a diferencia de escoger las variables correspondientes de la práctica 1.

Paso 5.

En este paso vamos a crear las imágenes de la práctica para la visualización en el HMI, agregamos los controles correspondientes:

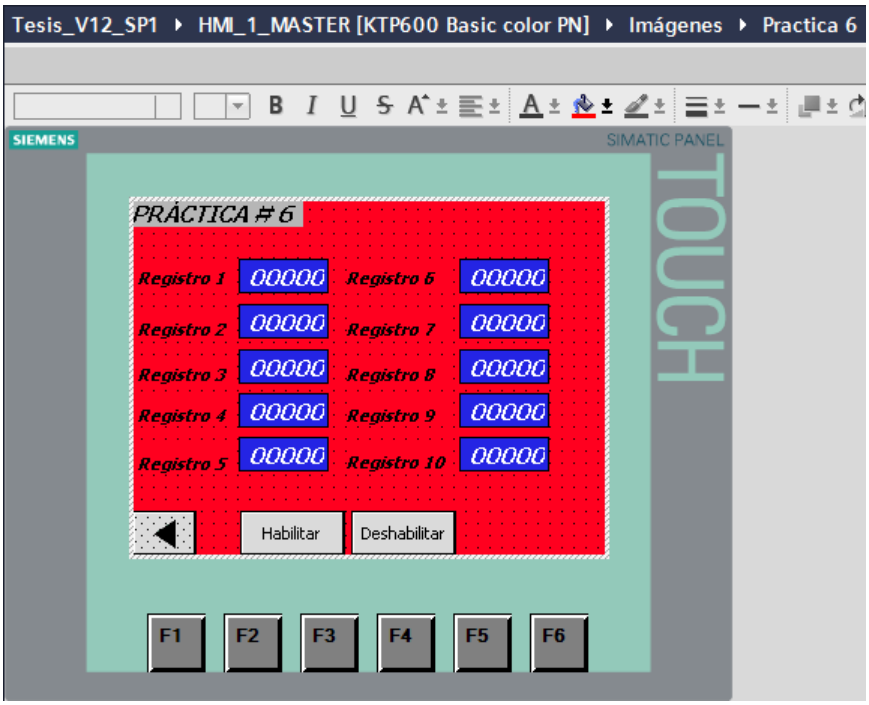


Figura 238: Diseño de la imagen Practica 6.

Procedemos a configurar cada campo de entrada / salida, asignando las variables correspondientes a los 10 campos anteriormente creados.

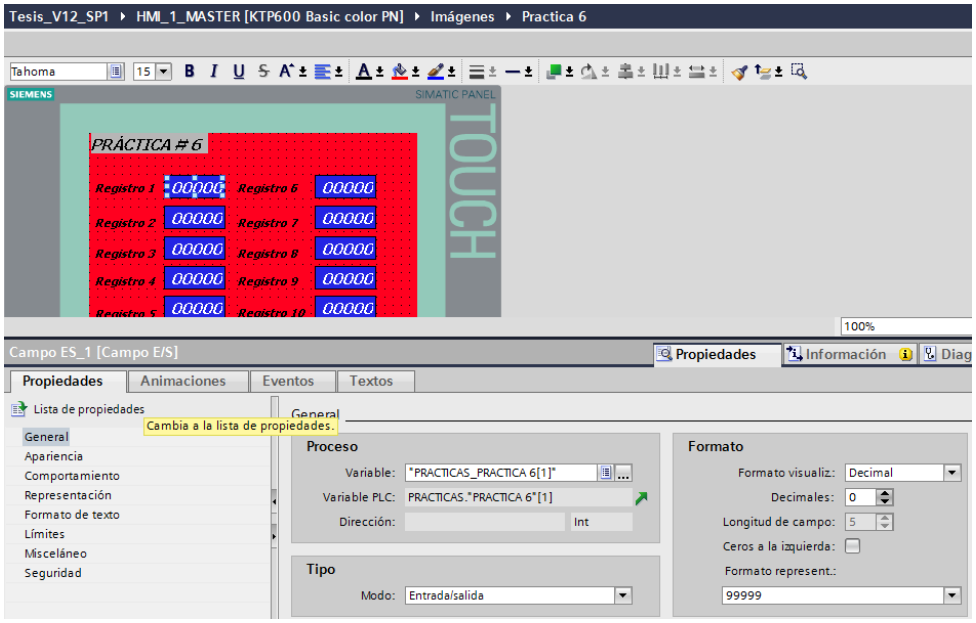


Figura 239: Configuración de campos de E/S para PLC MASTER, práctica 6.

Ahora se debe repetir el proceso con el PLC SLAVE.

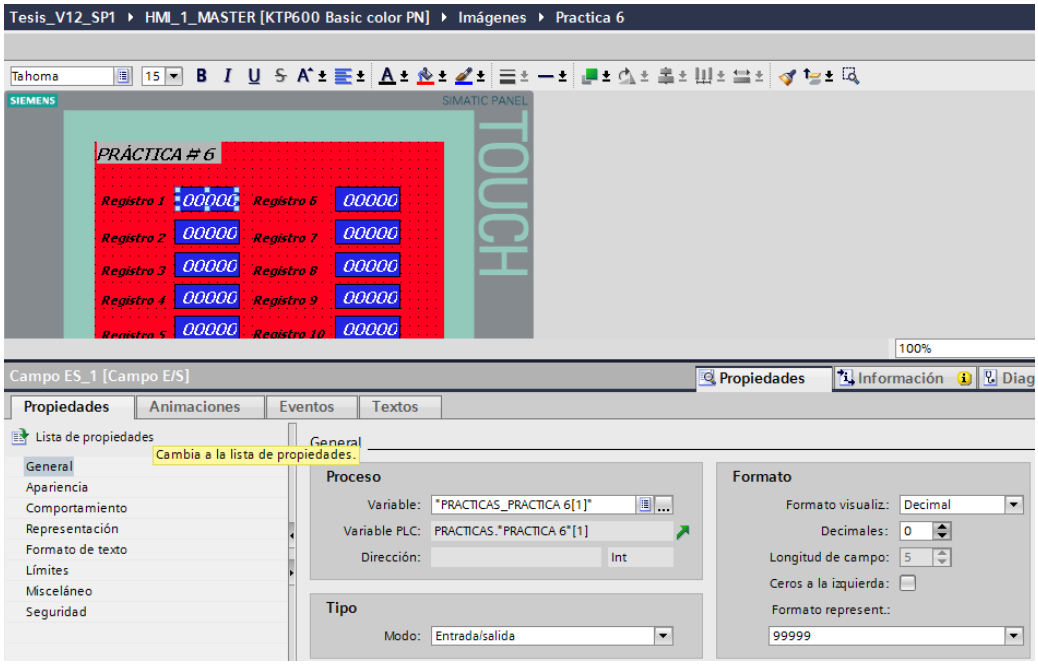


Figura 240: Configuración de campos de E/S para PLC SLAVE, práctica 6.

Observación: respecto a los botones **Habilitar**, **Deshabilitar**, y el **mensaje de alerta** se debe seguir las instrucciones del **Paso 20** de la práctica 1.

Paso 7.

Ahora procedemos a abrir el programa Modbus Poll y lo configuramos de manera que se reciban y guarden los datos enviados desde el PLC.

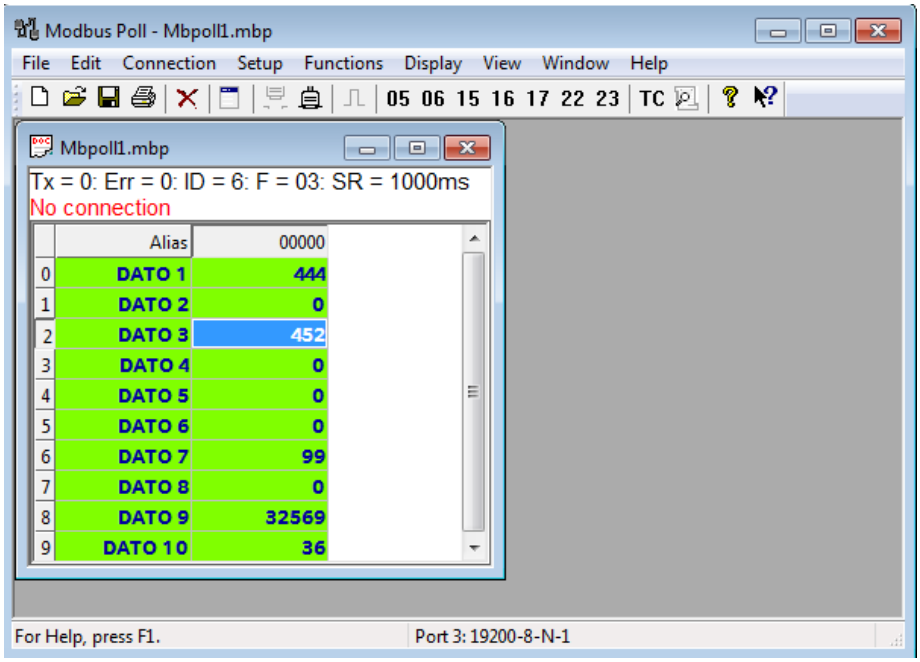


Figura 241: Modbus Poll configurado para la práctica 6.

Se detalla a continuación la configuración de conexión:

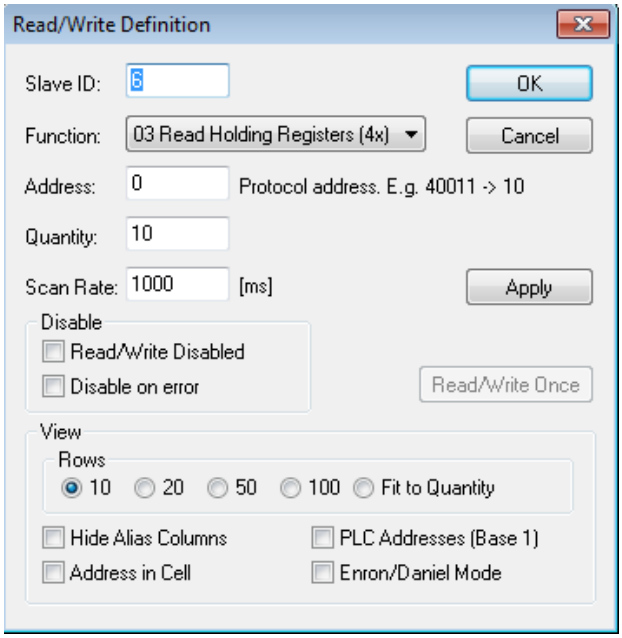


Figura 242: Configuración en modo lectura, práctica 6.

Los parámetros de conexión del simulador deben ser los mismos que el PLC.

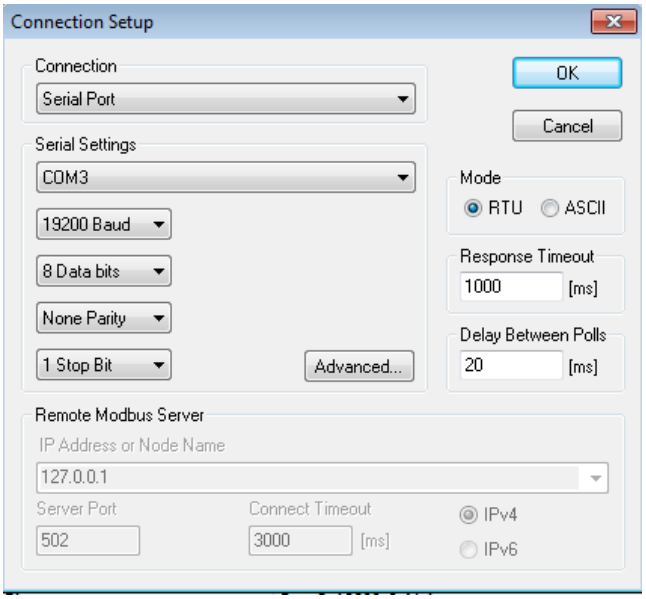


Figura 243: Parámetros de conexión del Modbus Poll.

Recomendaciones.

- Revisar el puerto de comunicación al momento de conectar el simulador.
- Comprobar la correcta lectura / escritura de datos.
- Se debe deshabilitar la práctica antes de salir del sistema.

4.7. Práctica Nº 7

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>7</td><td>TÍTULO PRÁCTICA: “Control y velocidad de un motor trifásico con un autómata Siemens S7-1200 y un TOUCH PANEL KTP600 PN.”</td></tr>
<tr><td colspan="4">OBJETIVOS:
Diseñar un sistema de marcha, paro y control de velocidad de un motor trifásico.
Parametrizar un variador de frecuencia ABB modelo ACS355.
Comunicar un PLC Siemens S7-1200 vía Modbus RTU con un variador ABB ACS355.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 / 240 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:
Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
Dos módulos de comunicación CM1241.
Un cable de comunicación RS485 macho – macho.
Un módulo adaptador FMBA-01 para MODBUS RTU.
Un variador de frecuencia ABB modelo ACS355.
Un motor trifásico ABB.
DESCRIPCIÓN:
Para arrancar el motor se debe presionar el botón Marcha desde el HMI donde el motor girará en sentido horario, se puede ajustar la velocidad antes de arrancar o después el motor. El sistema tendrá dos botones para incrementar o reducir la velocidad de giro. Así mismo el programa contará con tres pantallas gráficas de visualización de frecuencia, corriente y velocidad. A su vez en la pantalla se podrá visualizar los valores de frecuencia, amperaje y potencia par del motor en tiempo real, finalmente con el botón Paro se detendrá el sistema.

La figura 244 muestra la arquitectura de red a implementar.</td></tr>
</table>

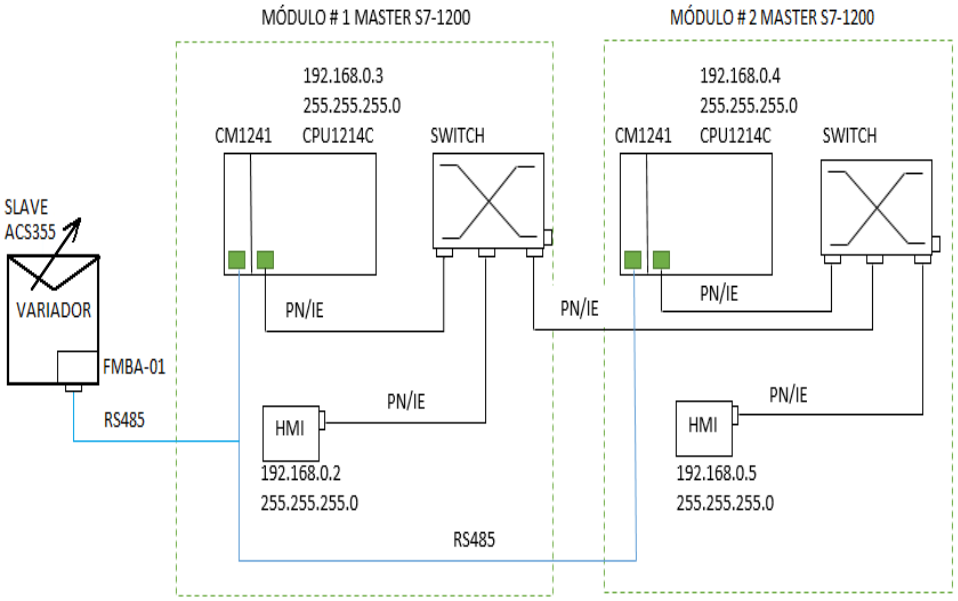


Figura 244: Arquitectura de red.

En la figura 245 se muestra el circuito de fuerza del variador y motor trifásico.

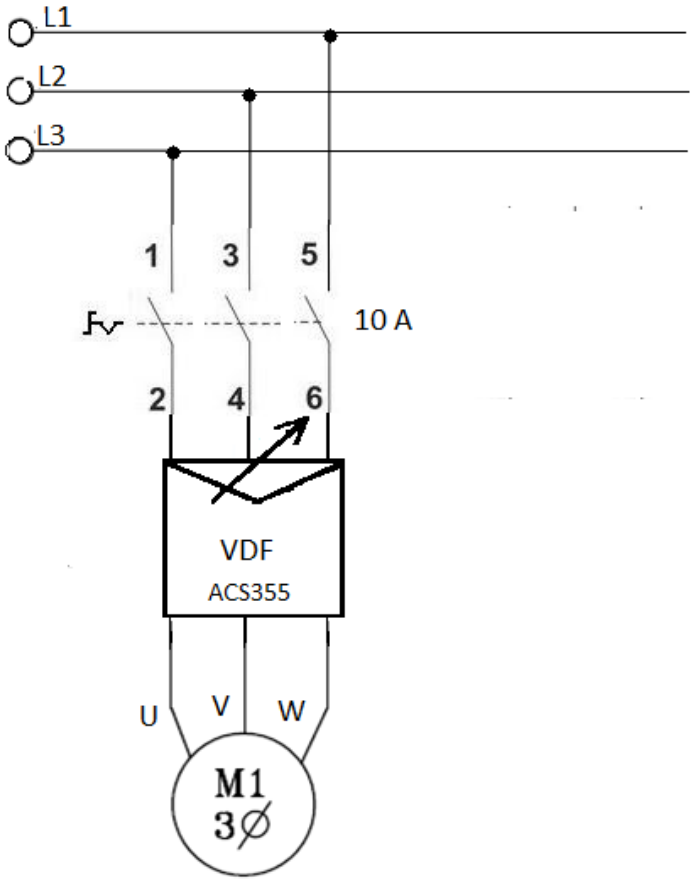


Figura 245: Circuito de fuerza.

Paso 1.

Como primer paso debemos parametrizar el variador de frecuencia, para esto seguimos el paso 3 de la práctica 5.

Paso 2.

Proseguimos con la creación de los parámetros a utilizar en los bloques de instrucciones de comunicación:

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB6]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	P1	Struct	0.0					
3	P2	Struct	12.0					
4	P3	Struct	32.0					
5	P4	Struct	42.0					
6	P5	Struct	62.0					
7	P6	Struct	102.0					
8	P7	Struct	112.0					
9	MB_ADDR	UInt	0.0	10				
10	MODE	USInt	2.0	1				
11	DATA_ADDR	UDInt	4.0	41401				
12	DATA_LEN	UInt	8.0	1				
13	MB_ADDR HZ	UInt	10.0	10				
14	MODE HZ	USInt	12.0	1				
15	DATA_ADDR HZ	UDInt	14.0	41202				
16	DATA_LEN HZ	UInt	18.0	1				
17	MB_ADDR SLAVE	UInt	20.0	10				
18	MODE SLAVE	USInt	22.0	0				
19	DATA_ADDR SLAVE	UDInt	24.0	40101				
20	DATA_LEN SLAVE	UInt	28.0	5				

Figura 246: Parámetros del PLC MASTER, práctica 7.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB11]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	P1	Struct	...					
3	P3	Struct	...					
4	P4	Struct	...					
5	P5	Struct	...					
6	P6	Struct	...					
7	P7	Struct	...					
8	MB_ADDR	UInt	...	10				
9	MODE	USInt	...	1				
10	DATA_ADDR	UDInt	...	41401				
11	DATA_LEN	UInt	...	1				
12	MB_ADDR HZ	UInt	...	10				
13	MODE HZ	USInt	...	1				
14	DATA_ADDR HZ	UDInt	...	41202				
15	DATA_LEN HZ	UInt	...	1				
16	MB_ADDR SLAVE	UInt	...	10				
17	MODE SLAVE	USInt	...	0				
18	DATA_ADDR SLAVE	UDInt	...	40101				
19	DATA_LEN SLAVE	UInt	...	5				

Figura 247: Parámetros del PLC SLAVE, práctica 7.

Paso 3.

Vamos a generar las variables de registro donde se guardarán los datos enviados y recibidos, se detalla a continuación las mismas:

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	PRACTICA 3	Struct	28.0					
5	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
6	PRACTICA 4	Struct	34.0					
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5310	Array [1..8] of Int	54.0					
9	PRACTICA 5_1102	Array [1..4] of Int	70.0					
10	PRACTICA 5_9905	Array [1..4] of Int	78.0					
11	PRACTICA 6	Array [1..10] of Int	86.0					
12	PRACTICA 7	Array [0..1] ...	106.0					
13	PRACTICA 7[0]	Int	0.0	0				
14	PRACTICA 7[1]	Int	2.0	0				
15	PRACTICA 7 HZ	Array [0..1] of Int	110.0					
16	PRACTICA 7 HZ[0]	Int	0.0	0				
17	PRACTICA 7 HZ[1]	Int	2.0	0				
18	PRACTICA 7 SLAVE	Array [1..5] of Int	114.0					
19	PRACTICA 7 SLAVE...	Int	0.0	0				
20	PRACTICA 7 SLAVE...	Int	2.0	0				
21	PRACTICA 7 SLAVE...	Int	4.0	0				
22	PRACTICA 7 SLAVE...	Int	6.0	0				
23	PRACTICA 7 SLAVE...	Int	8.0	0				

Figura 248: Variables de registro para el PLC MASTER – práctica 7.

A su vez creamos las variables de registro para el PLC SLAVE.

Tesis_V12_SP1 ▸ SLAVE [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	PRACTICA 3	Struct	28.0					
5	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
6	PRACTICA 4	Struct	34.0					
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5310	Array [1..8] of Int	54.0					
9	PRACTICA 5_1102	Array [1..4] of Int	70.0					
10	PRACTICA 5_9905	Array [1..4] of Int	78.0					
11	PRACTICA 6	Array [1..10] of Int	86.0					
12	PRACTICA 7	Array [0..1] of Int	106.0					
13	PRACTICA 7[0]	Int	0.0	0				
14	PRACTICA 7[1]	Int	2.0	0				
15	PRACTICA 7 HZ	Array [0..1] of Int	110.0					
16	PRACTICA 7 HZ[0]	Int	0.0	0				
17	PRACTICA 7 HZ[1]	Int	2.0	0				
18	PRACTICA 7 SLAVE	Array [1..5] ...	114.0					
19	PRACTICA 7 SLAVE...	Int	0.0	0				
20	PRACTICA 7 SLAVE...	Int	2.0	0				
21	PRACTICA 7 SLAVE...	Int	4.0	0				
22	PRACTICA 7 SLAVE...	Int	6.0	0				
23	PRACTICA 7 SLAVE...	Int	8.0	0				

Figura 249: Variables de registro para el PLC MASTER – práctica 7.

Paso 4.

Procedemos a agregar los bloques de instrucciones para la comunicación, en nuestro caso se necesita agregar 3 bloques master de los cuales 2 son de escritura y 1 de lectura. Es importante recalcar que los PLC ejercerán la función de maestro (uno a la

vez), mientras que el variador de frecuencia en conjunto con el motor trifásico permanecen como esclavo Modbus.

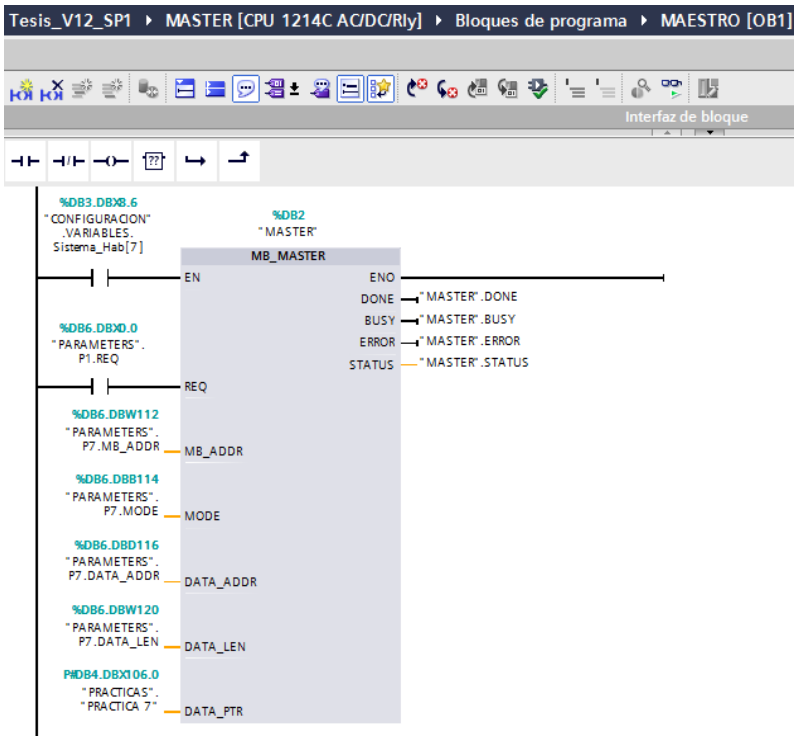


Figura 250: Bloque de escritura para relé, práctica 7.

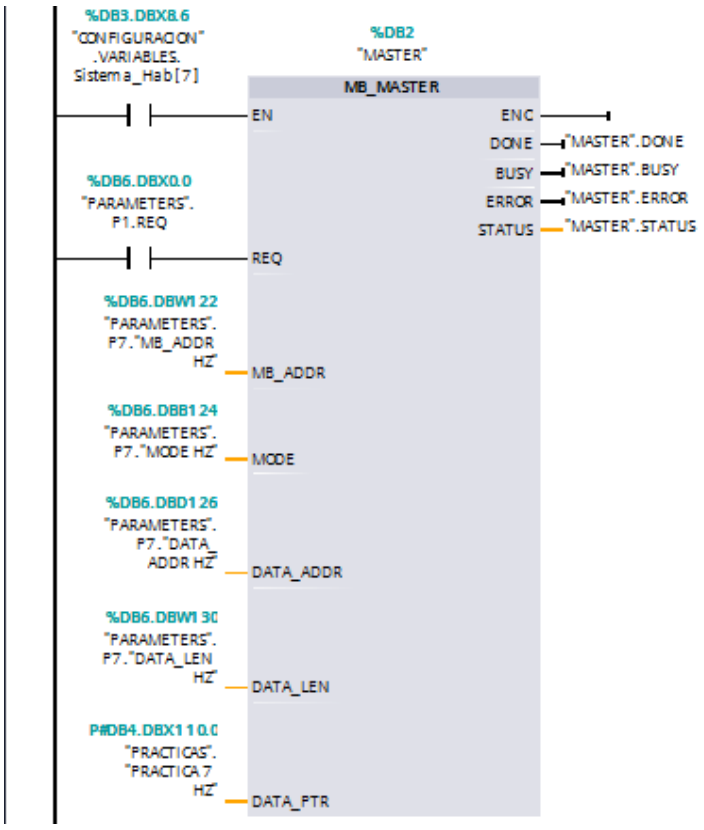


Figura 251: Bloque de escritura para velocidad, práctica 7.

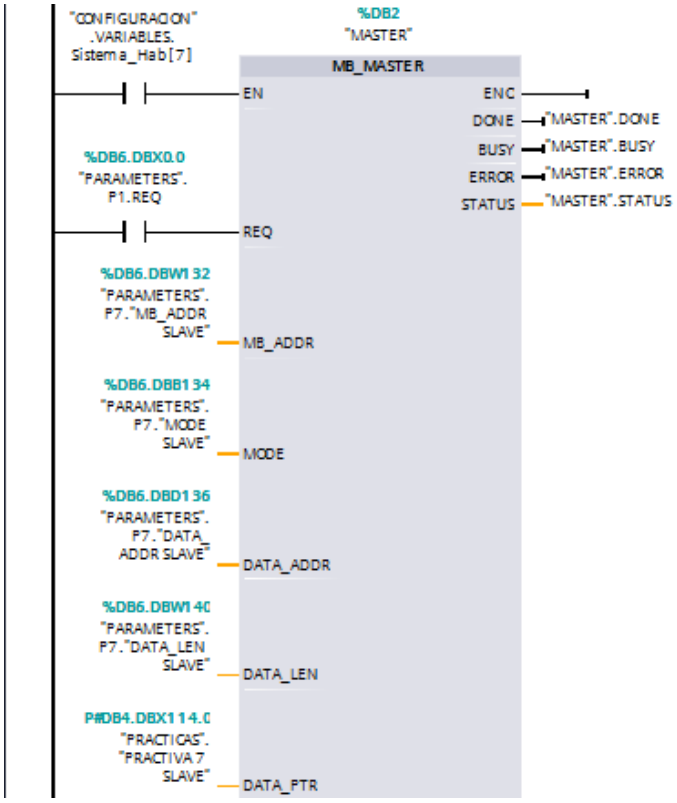


Figura 252: Bloque de lectura para frecuencia, corriente y potencia, práctica 7.

Paso 5.

En este paso vamos a programar el circuito de control del sistema en el **Segmento 10** del Maestro, para ello detallamos el encendido y apagado del motor con los botones **Marcha** y **Paro**.

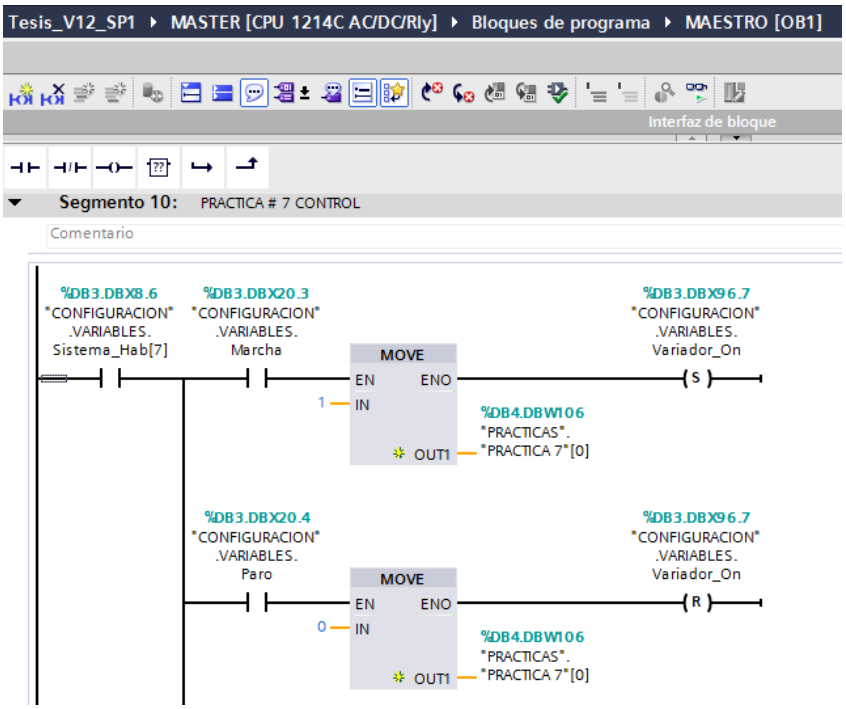


Figura 253: Marcha y Paro del sistema, práctica 7.

Paso 6.

Para visualizar los datos de frecuencia del variador es necesario utilizar las funciones **NORM_X** (Normalización) y **SCALE_X** (Escalado), esto nos sirve para normalizar una señal análoga (real) y escalarla para que se adapte al programa (entero). Para la normalización se fijan los limites inferior y superior, la cual va estar dado por los valores MIN = 0 y MAX = 27000; así mismo para el escalado va estar dado por los valores MIN = 0 y MAX = 27000.

Vamos a crear un bloque de función FC el cual obtendremos la señal modulada de la conversión, el bloque se llamará **SIGNAL_ANALOGA [FB3].**

El valor que da la salida del escalado **MODULATE** se lo va usar en el HMI.

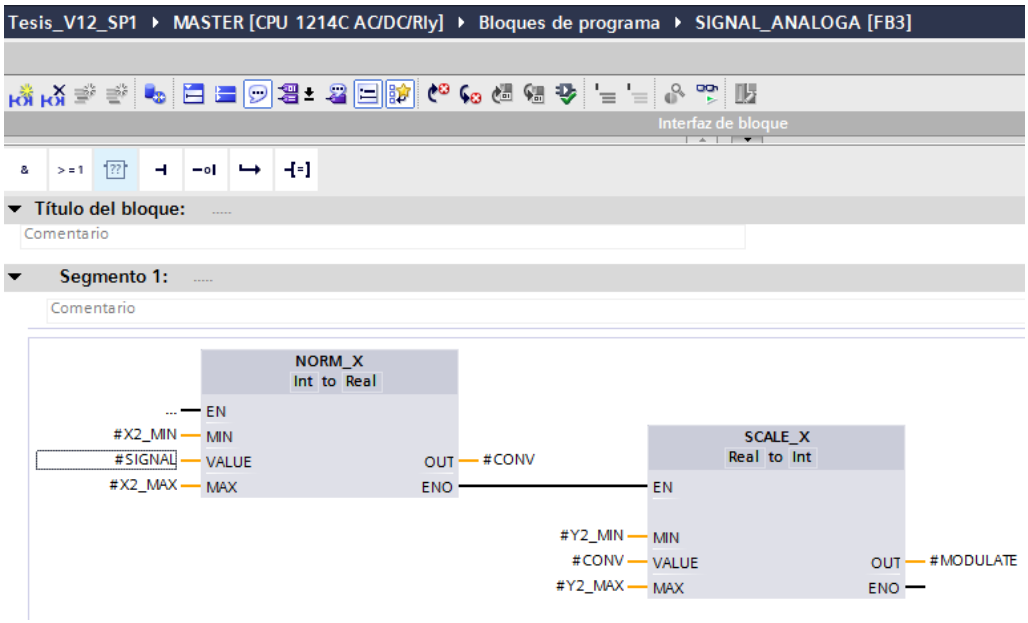


Figura 254: Creación de bloque SIGNAL_ANALOGA.

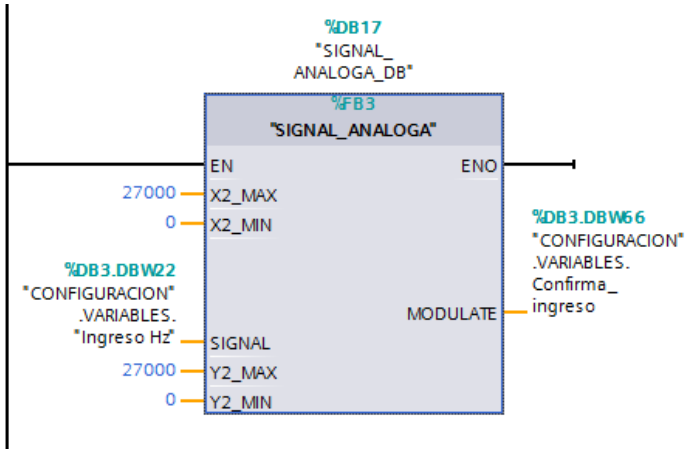


Figura 255: Aplicación de bloque SIGNAL_ANALOGA.

Paso 7.

Continuando con la parte de control vamos a generar la programación para los botones de subida y bajada de la velocidad del motor, los cuales de los valores que se ingresen en la variable **Ingreso Hz** se va a sumar o restar 10 unidades, con esto al presionar los botones existirá un incremento o reducción de la velocidad del motor en 10 unidades.

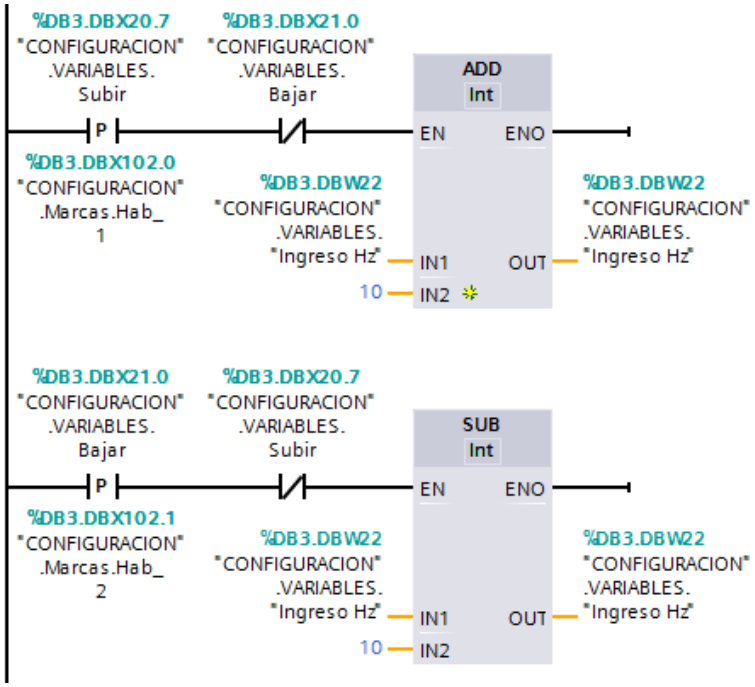


Figura 256: Incremento y reducción manual de la velocidad del motor.

Vamos a añadir un bloqueo para cuando se ingrese un valor que sobrepase la velocidad máxima del motor la cual es 1500 rpm.

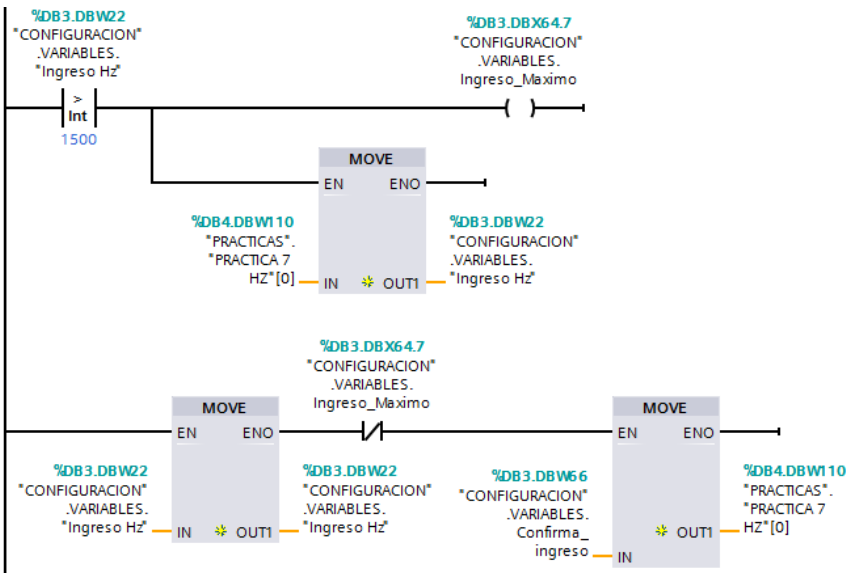


Figura 257: Bloqueo para la velocidad máxima.

Se colocaron mensajes de aviso para que el usuario se alerte al sobrepasar los límites del motor, es decir cuando sobrepase las 1500 rpm y menos de 0 rpm.

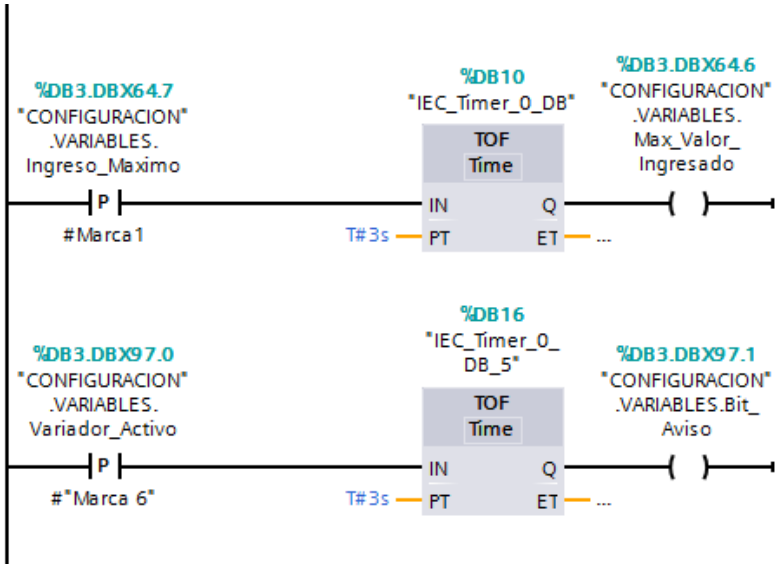


Figura 258: Mensajes de avisos de límite superior de velocidad del motor.

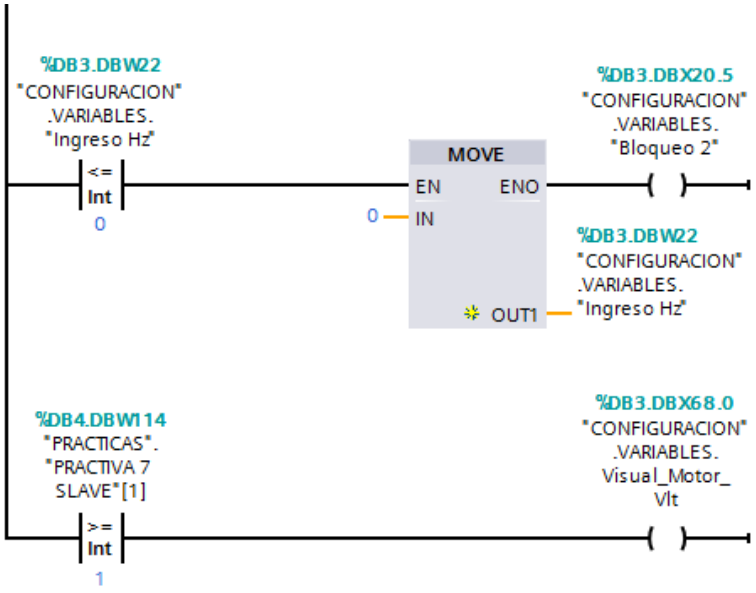


Figura 259: Mensajes de avisos de límite inferior de velocidad del motor.

Paso 8.

En nuestro programas se necesita tomar lectura del variador como por ejemplo corriente, potencia par del motor y frecuencia, para ello elaboramos en el Segmento 11 el monitoreo de los mismos para que sea visualizado en la pantalla HMI. Realizamos una conversión de entero a real para que en el HMI se visualicen los valores con decimales, se divide para 10 ya que la lectura del variador nos arroja un

valor alto con dos ceros adicionales.

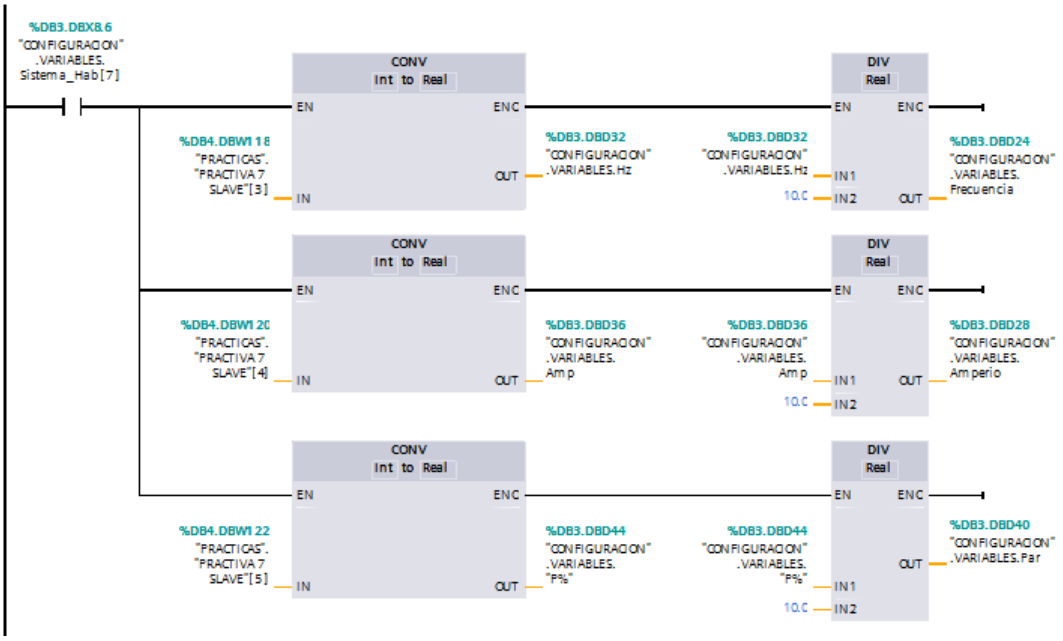


Figura 260: Lectura de frecuencia, corriente y potencia del motor.

Observación: se deben repetir los pasos del 4 al 8 para el PLC SLAVE pero seleccionando las variables adecuadas.

Paso 9.

Daremos paso a la creación de las imágenes en el HMI. Mostramos a continuación el diseño de la **Título 7**.

Figura 261: Diseño de la imagen Titulo 7.

Observación: se debe crear una imagen similar para el PLC SLAVE.

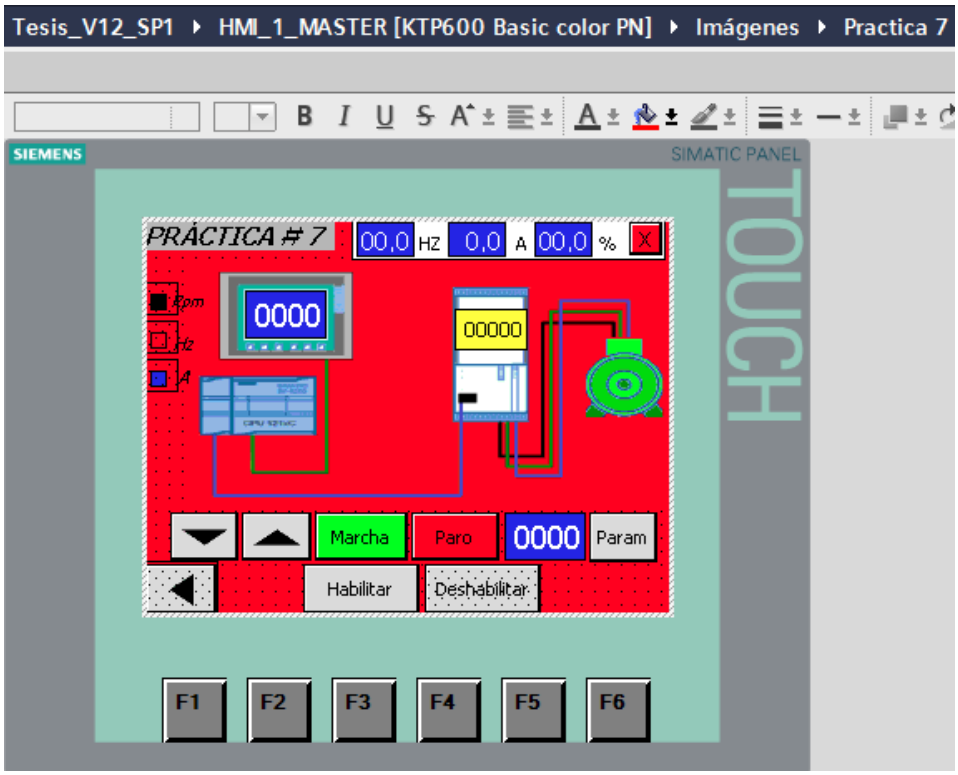


Figura 262: Diseño de la imagen Practica 7.

A continuación se muestra la configuración del botón marcha:

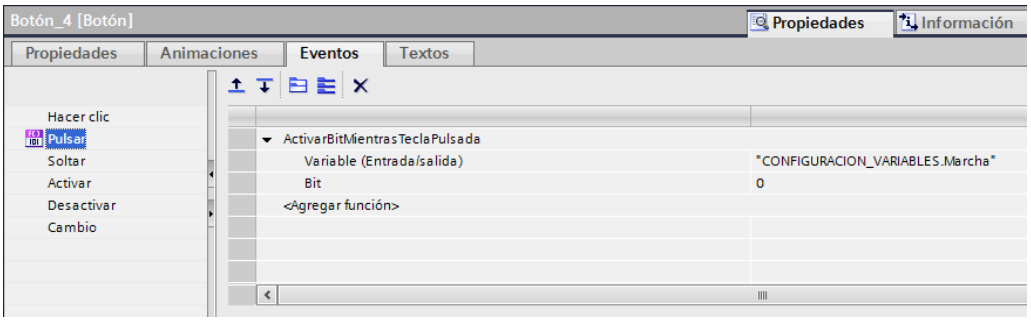


Figura 263: Eventos del botón Marcha, practica 7.

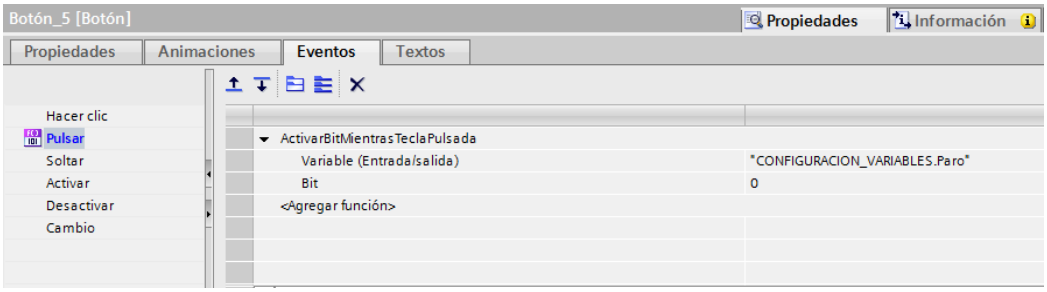


Figura 264: Eventos del botón Marcha, practica 7.

Ahora mostraremos la configuración de los campos de entrada/salida, se utilizará la mismas variable **Ingreso Hz** para dos controles, el que simula el HMI y el que aparece en la imagen 265:

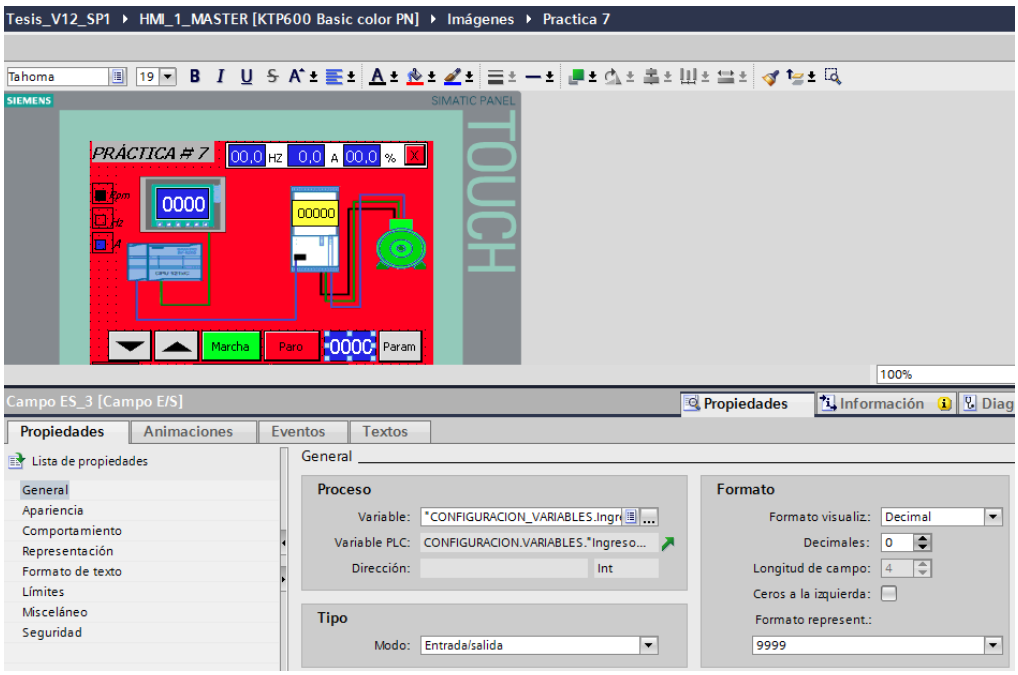


Figura 265: Propiedades del campo E/S 3.

El **Grupo_3** está conformado por 1 rectángulo, 3 campos de E/S (frecuencia, amperaje y porcentaje par), 3 cuadros de texto y 1 botón (X).

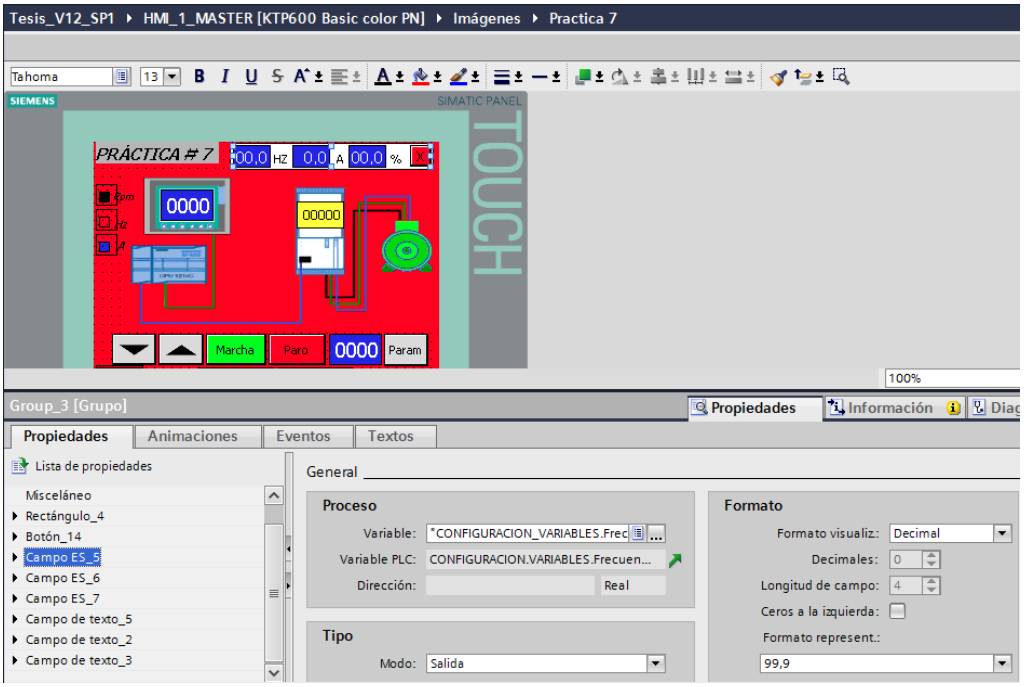


Figura 266: Propiedades del grupo 3.

El diseño de la pantalla también cuenta con un visor de gráficos donde obtendremos el valor de las rpm.

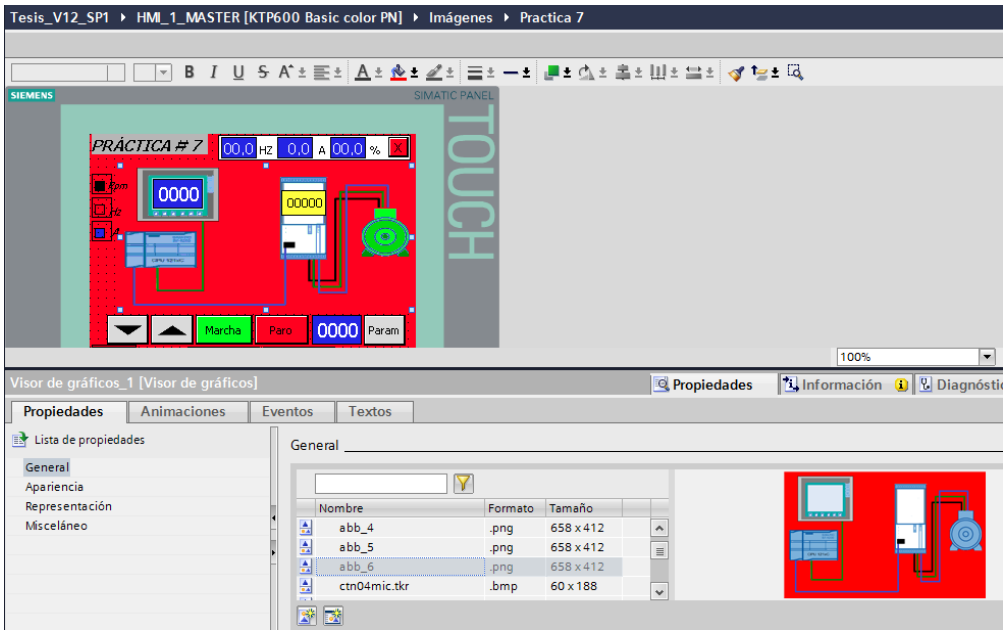


Figura 267: Propiedades de visor de gráfico.

Respecto a la iluminación del motor presentamos la configuración donde ligamos la imagen con la variable "CONFIGURACION_VARIABLES.Visual_Motor_Vlt" :

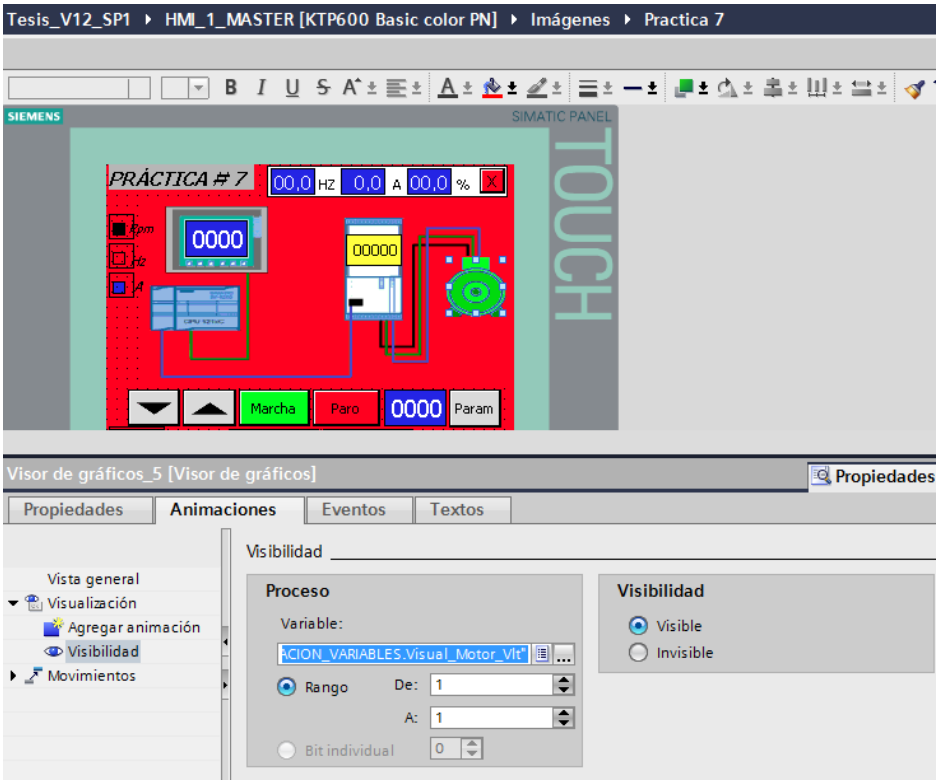


Figura 268: Animación de motor, práctica 7.

Respecto al campo de texto asignado al variador en el visor de gráficos, se detalla la configuración:

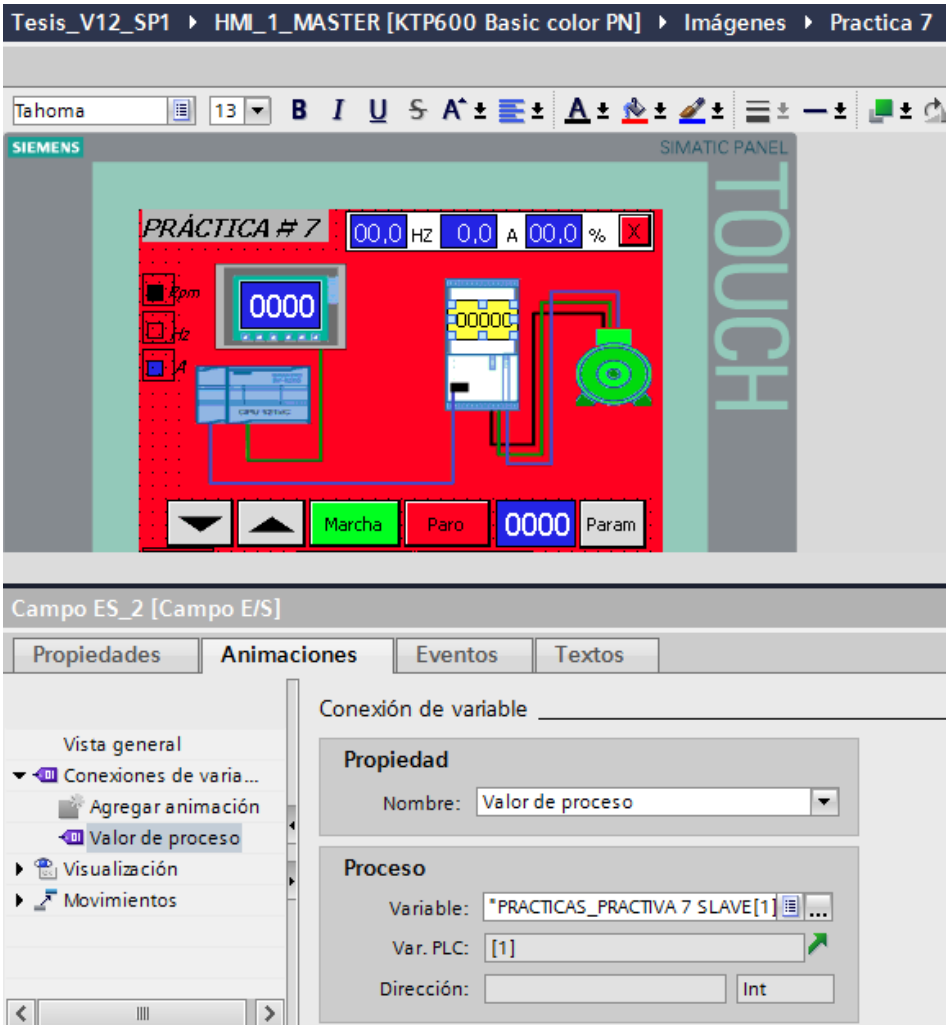


Figura 269: Animación del campo E/S 2.

Recomendaciones.

- Verificar los pines de comunicación entre el puerto DB9 del módulo CM124 y el módulo FMBA-01.
- Comprobar la correcta lectura de los parámetros del variador de frecuencia.
- Colocar en modo remoto el variador para una efectiva comunicación.

4.8. Práctica Nº 8

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>8</td><td>TÍTULO PRÁCTICA: “Comunicación MODBUS RTU con SCADA – S7-1200 a Wincc Rtu Advanced”</td></tr>
<tr><td colspan="4">OBJETIVOS:
• Diseñar un sistema de llenado manual y automático entre un maestro y un esclavo con 2 PCL S7-1200 vía SCADA.
• Implementar un control de nivel de una planta de agua vía comunicación MODBUS RTU.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:
• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• Dos módulos de comunicación CM1241.
• Un cable de comunicación RS485 macho – macho.
• Dos tanques de 10 litros de capacidad (TK1 / TK2).
• Una electroválvula (SV101).
• Una bomba centrifuga (P101).
• Un sensor ultrasónico (TK102) LIC102.2
• Un sensor tipo boya en tanque TK102 (LSL102.1).
• Dos sensores de proximidad capacitivos en TK101 (LSH101.2 / LSL101.1)
• Dos pulsadores, uno para la selección de la práctica (S1) y el otro para resetear la selección (S2).

DESCRIPCIÓN:
Este sistema funciona de la siguiente manera: primero se debe establecer la comunicación en el SCADA con el botón Habilitar para después presionar el pulsante S1 en la planta para habilitar la práctica 8. Establecida la comunicación entre maestro (módulo didáctico) y esclavo (planta) seleccionamos el modo Manual</td></tr>
</table>

o **Automático**. En modo manual se podrá llenar el tanque TK2 presionando la figura de la bomba **P101** en el SCADA, para vaciar el tanque TK2 se pulsa la figura de la electroválvula **SV101**. Para cambiar de modo a **Automático** se debe presionar el botón **Paro** en el SCADA. Seleccionamos el modo automático y sin importar el nivel en que se encuentre el tanque **TK2**, esta va a comenzar a llenar activado por la bomba hasta su capacidad máxima 30 cm. Al llegar a 30 cm. el nivel comienza a bajar al activarse la electroválvula hasta llegar a su capacidad mínima 3.5 cm. El ciclo se repite hasta dar **Paro** al sistema.

La figura 270 muestra la arquitectura de red a implementar.

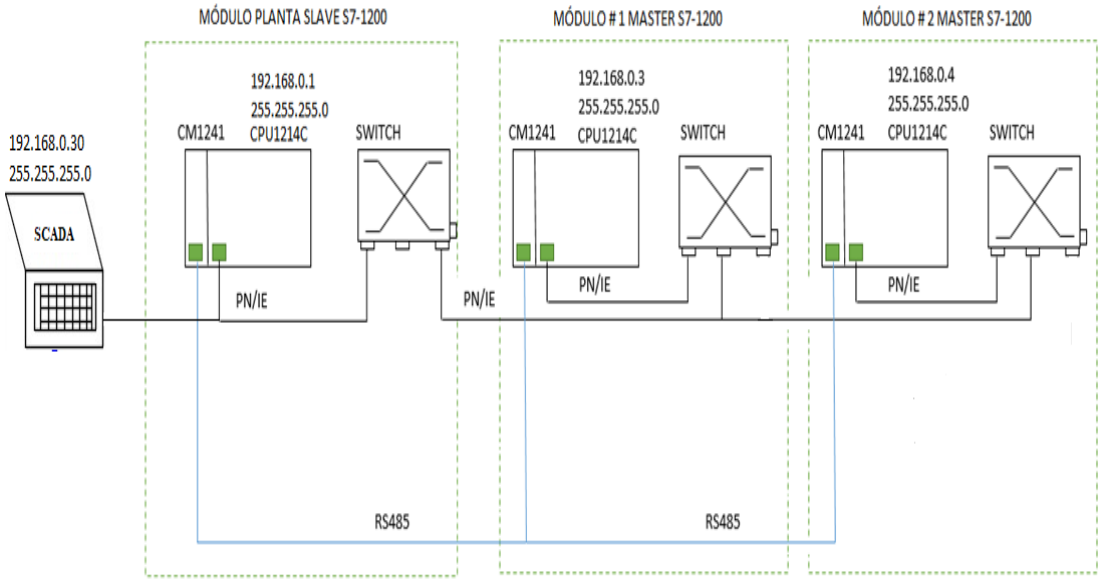


Figura 270: Arquitectura de red.

En figura 271 se muestra el diagrama de conexiones a implementar.

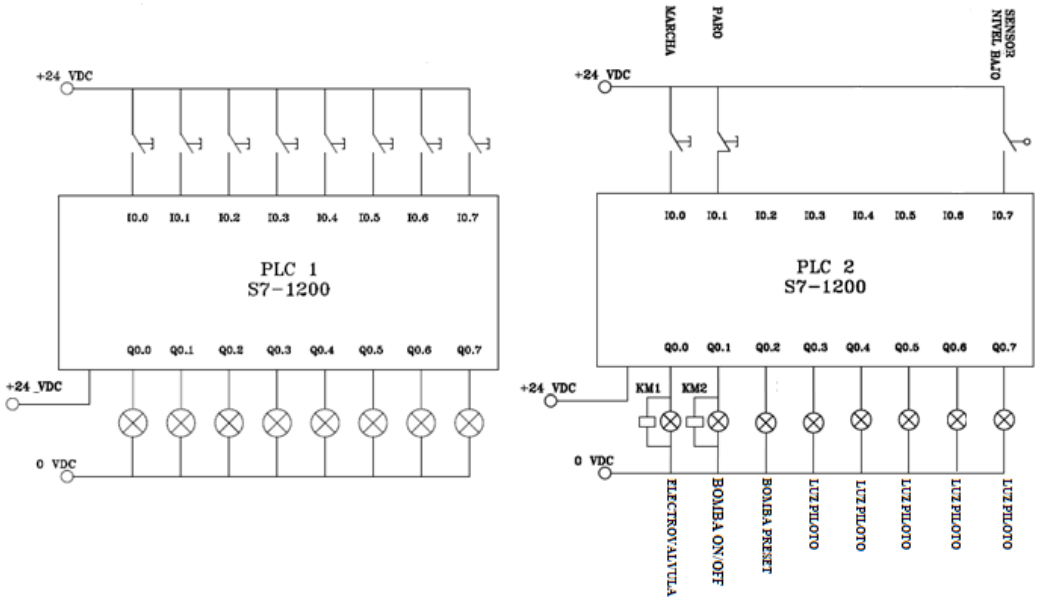
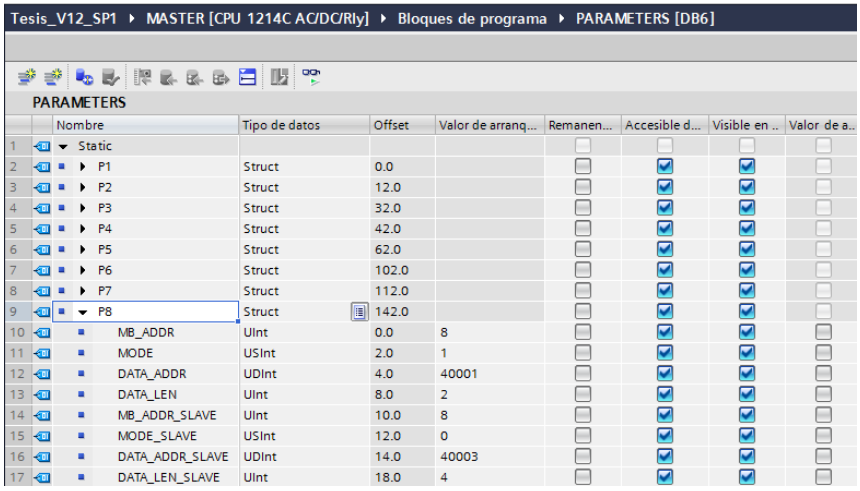


Figura 271: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear los parámetros para las instrucciones de la práctica 8, nos dirigimos a **PARAMETERS [DB6]** ubicado en el PLC **MASTER** y creamos las siguientes variables:



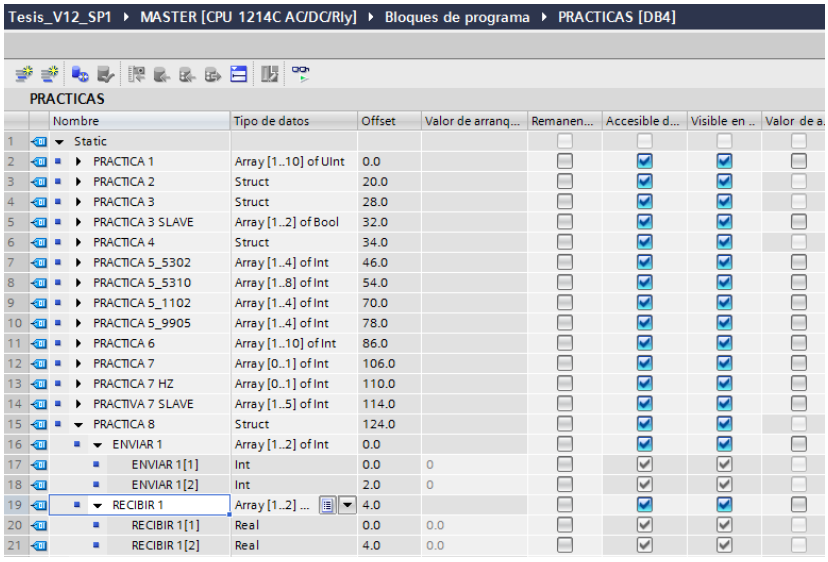
	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor de a...
1	Static							
2	P1	Struct	0.0					
3	P2	Struct	12.0					
4	P3	Struct	32.0					
5	P4	Struct	42.0					
6	P5	Struct	62.0					
7	P6	Struct	102.0					
8	P7	Struct	112.0					
9	P8	Struct	142.0					
10	MB_ADDR	UInt	0.0	8				
11	MODE	USInt	2.0	1				
12	DATA_ADDR	UDInt	4.0	40001				
13	DATA_LEN	UInt	8.0	2				
14	MB_ADDR_SLAVE	UInt	10.0	8				
15	MODE_SLAVE	USInt	12.0	0				
16	DATA_ADDR_SLAVE	UDInt	14.0	40003				
17	DATA_LEN_SLAVE	UInt	18.0	4				

Figura 272: Parámetros del PLC MASTER, práctica 8.

Observación: También se necesita crear los parámetros para el PLC **SLAVE** los cuales van a ser los mismos que el **MASTER**. El PLC **PLANTA** no requiere de un bloque de parámetros ya que solo va a estar compuesto de bloques **MB_SLAVE_DB**, es decir la planta de nivel de agua funcionará como esclavo MODBUS.

Paso 2.

Ahora debemos generar las variables de registro para el PLC **MASTER** en **PRACTICAS [DB4]**. La imagen a continuación mostrará las variables de registro de datos.



	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor de a...
1	Static							
2	PRACTICA 1	Array [1..10] of UInt	0.0					
3	PRACTICA 2	Struct	20.0					
4	PRACTICA 3	Struct	28.0					
5	PRACTICA 3 SLAVE	Array [1..2] of Bool	32.0					
6	PRACTICA 4	Struct	34.0					
7	PRACTICA 5_5302	Array [1..4] of Int	46.0					
8	PRACTICA 5_5310	Array [1..8] of Int	54.0					
9	PRACTICA 5_1102	Array [1..4] of Int	70.0					
10	PRACTICA 5_9905	Array [1..4] of Int	78.0					
11	PRACTICA 6	Array [1..10] of Int	86.0					
12	PRACTICA 7	Array [0..1] of Int	106.0					
13	PRACTICA 7 HZ	Array [0..1] of Int	110.0					
14	PRACTIVA 7 SLAVE	Array [1..5] of Int	114.0					
15	PRACTICA 8	Struct	124.0					
16	ENVIAR 1	Array [1..2] of Int	0.0					
17	ENVIAR 1[1]	Int	0.0	0				
18	ENVIAR 1[2]	Int	2.0	0				
19	RECIBIR 1	Array [1..2] ...	4.0					
20	RECIBIR 1[1]	Real	0.0	0.0				
21	RECIBIR 1[2]	Real	4.0	0.0				

Figura 273: Variables de registro del PLC MASTER, práctica 8.

En la siguiente figura se mostrarán las variables de registro para el PLC PLANTA:

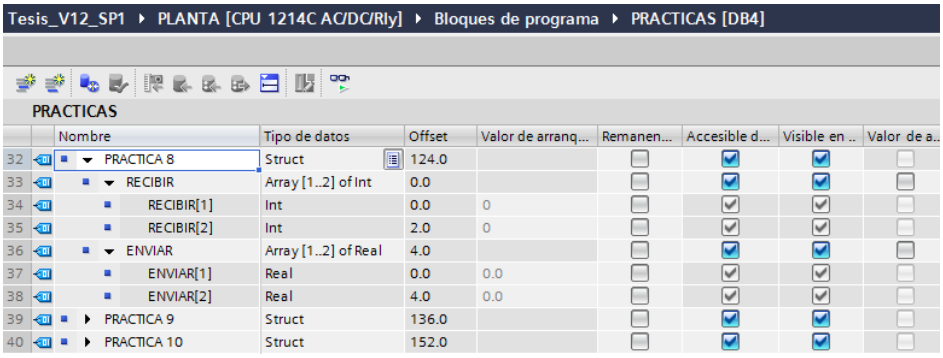


Figura 274: Variables de registro del PLC PLANTA, práctica 8.

Paso 3.

Nos dirigimos a **PLANTA_OB1** donde crearemos el bloque de instrucción **MB_SLAVE_DB** que receptará los datos de comunicación en la planta. Esto lo vamos a generar en el **Segmento 2**.

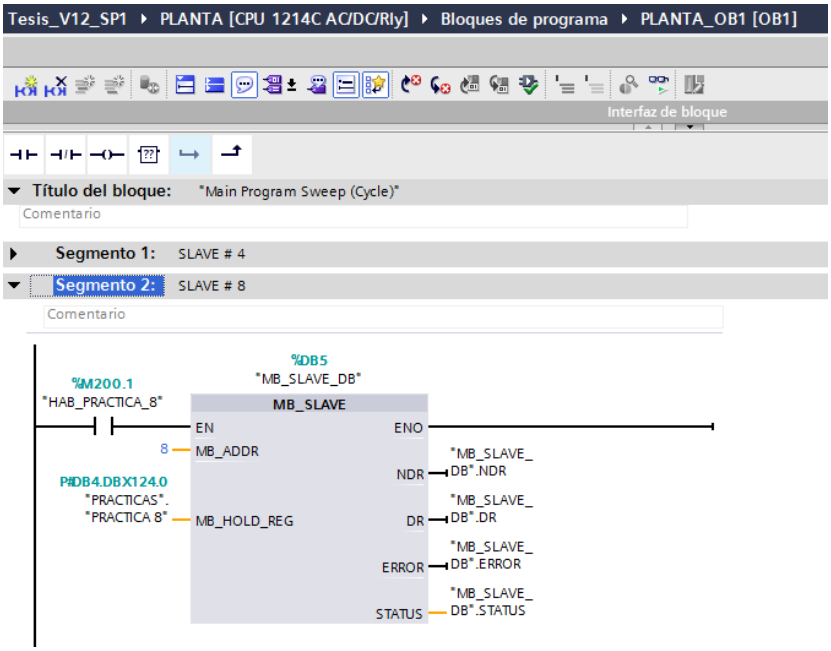


Figura 275: Bloque Slave de práctica 8.

Paso 4.

En este paso vamos a crear los bloques de comunicación para el PLC **MASTER** en el **Segmento 12,** aquí funcionaran dos bloques máster donde uno tendrá la labor de escribir datos y el segundo en leer:

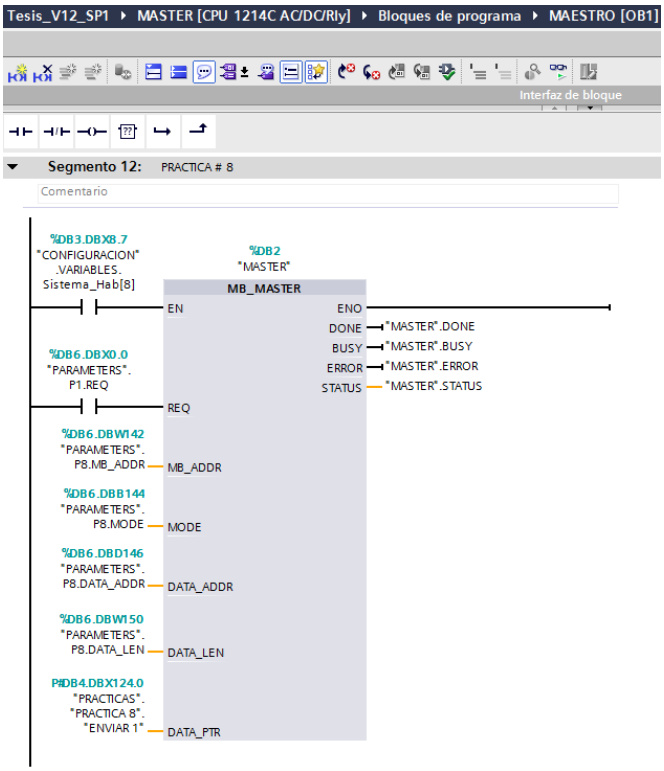


Figura 276: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 8.

El primer bloque de instrucción está funcionando en modo escritura, el cual enviará los datos al slave como activación de la bomba y electroválvula. Debemos agregar un segundo bloque máster el cual va a leer el sensor ultrasónico para detección de nivel de agua en los tanques.

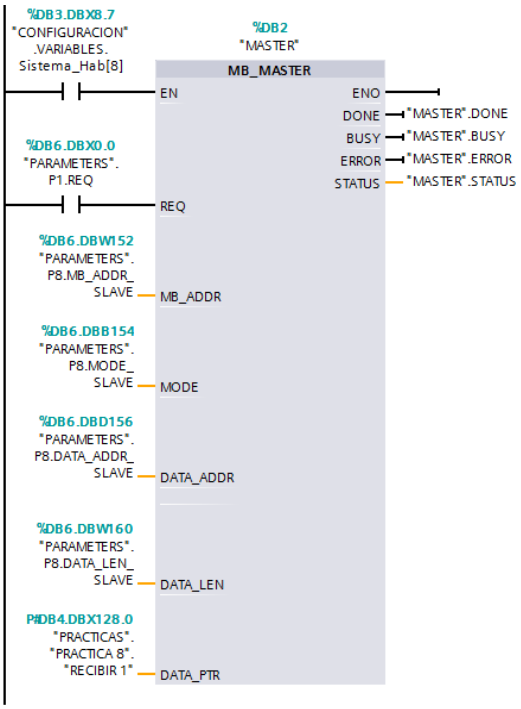


Figura 277: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 8.

Paso 7.

Ahora detallaremos el circuito de control para el PLC MASTER, en el **Segmento 13** tenemos la selección del modo, Manual o Automático:

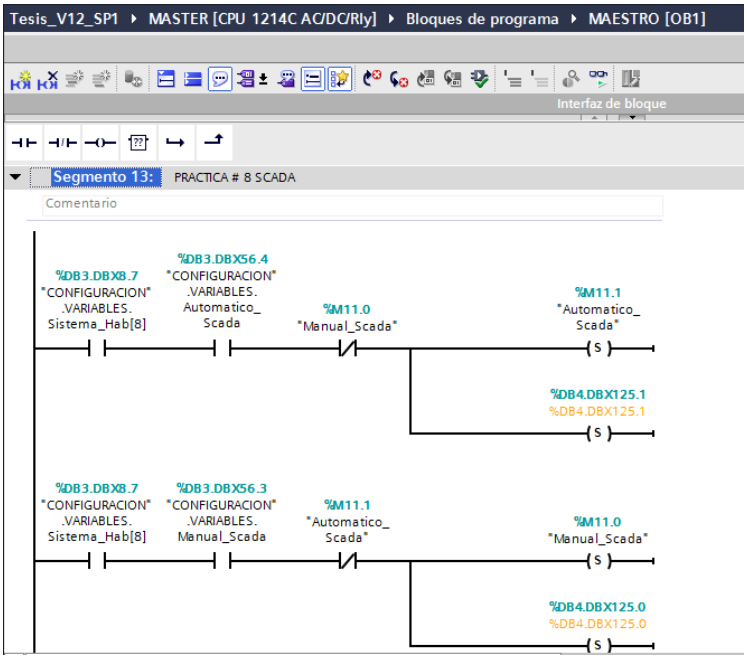


Figura 278: Selección de modo para el PLC MASTER, práctica 8.

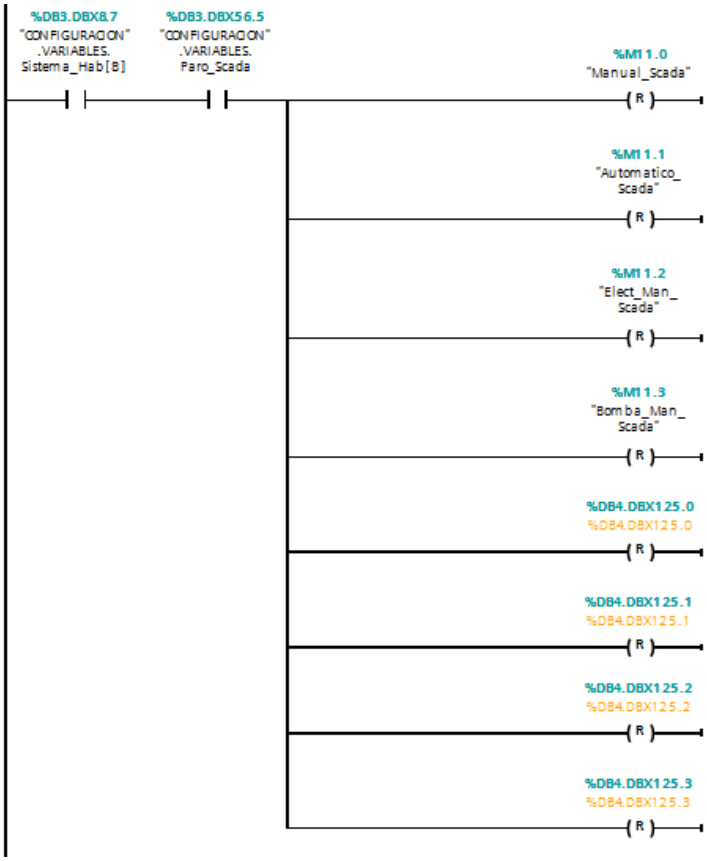


Figura 279: Marcas de apagado de controles manual / automático.

En esta sección (figura 280) se encenderá manualmente la bomba y electroválvula:

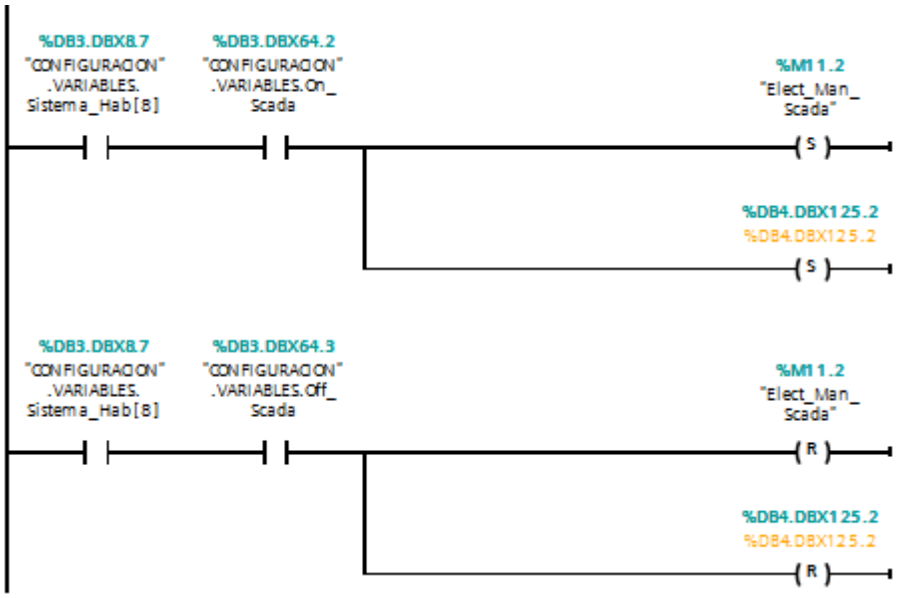


Figura 280: Activación manual de bomba y electroválvula.

En esta sección vamos a guardar el nivel del tanque en la variable de tipo real llamada NIVEL PLANTA.

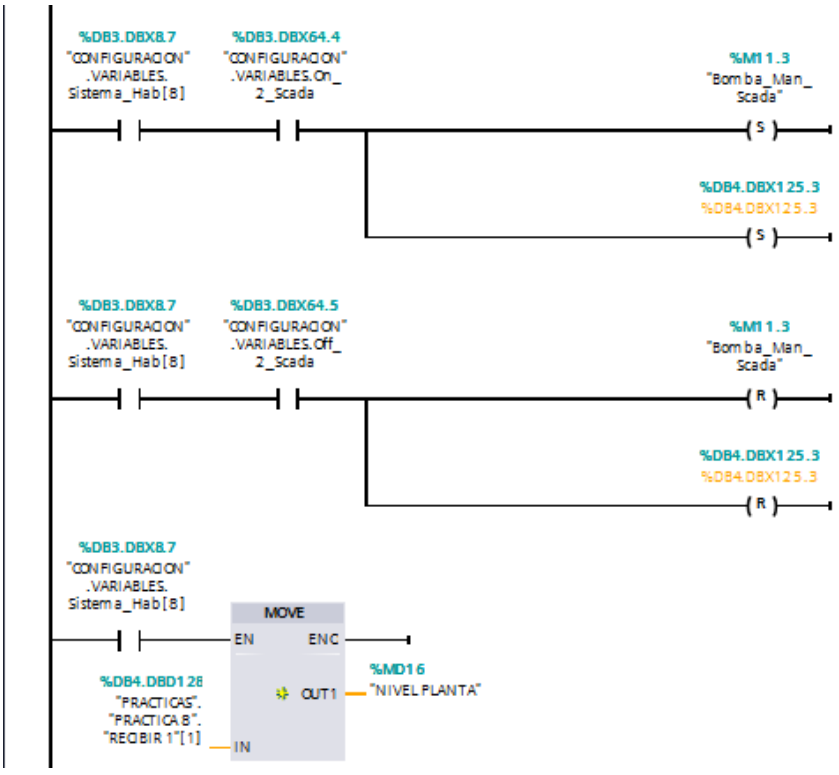


Figura 281: Registro de nivel en variable NIVEL PLANTA.

Paso 8.

Ahora vamos a programar el circuito de control para el PLC **PLANTA**, aquí vamos a diferenciar las diversas funciones del esclavo. Para esto se creará un bloque de función el cual llamaremos **PRACTICA # 8 SLAVE [FC5]**.

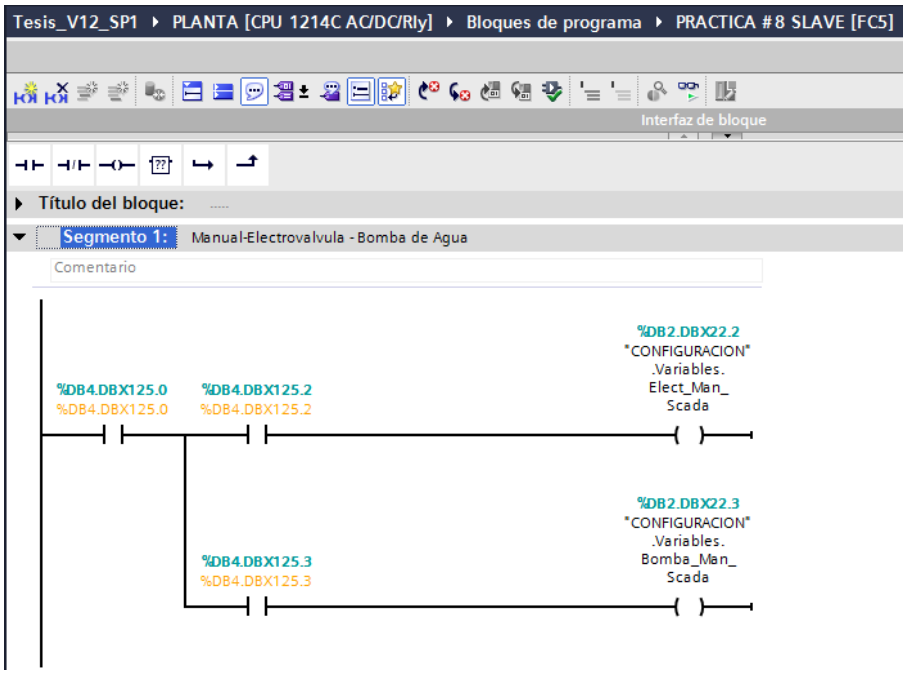


Figura 282: Control de Segmento 1 manual, PLANTA.

Figura 283: Control de Segmento 1 automático, PLANTA.

En el Segmento 2 realizaremos la activación y bloqueo de la Bomba (Q0.1).Se debe tomar en cuenta que para que la bomba P101 se desactive debe estar activada la marca Bloqueo_Bomba o Bloqueo_nivel_Max, es decir, cuando se rebase el nivel máximo permitido del tanque TK102 se activaran las marcas de bloqueo para desactivar la bomba, inmediatamente se activará la electroválvula SV101 para el vaciado del tanque.

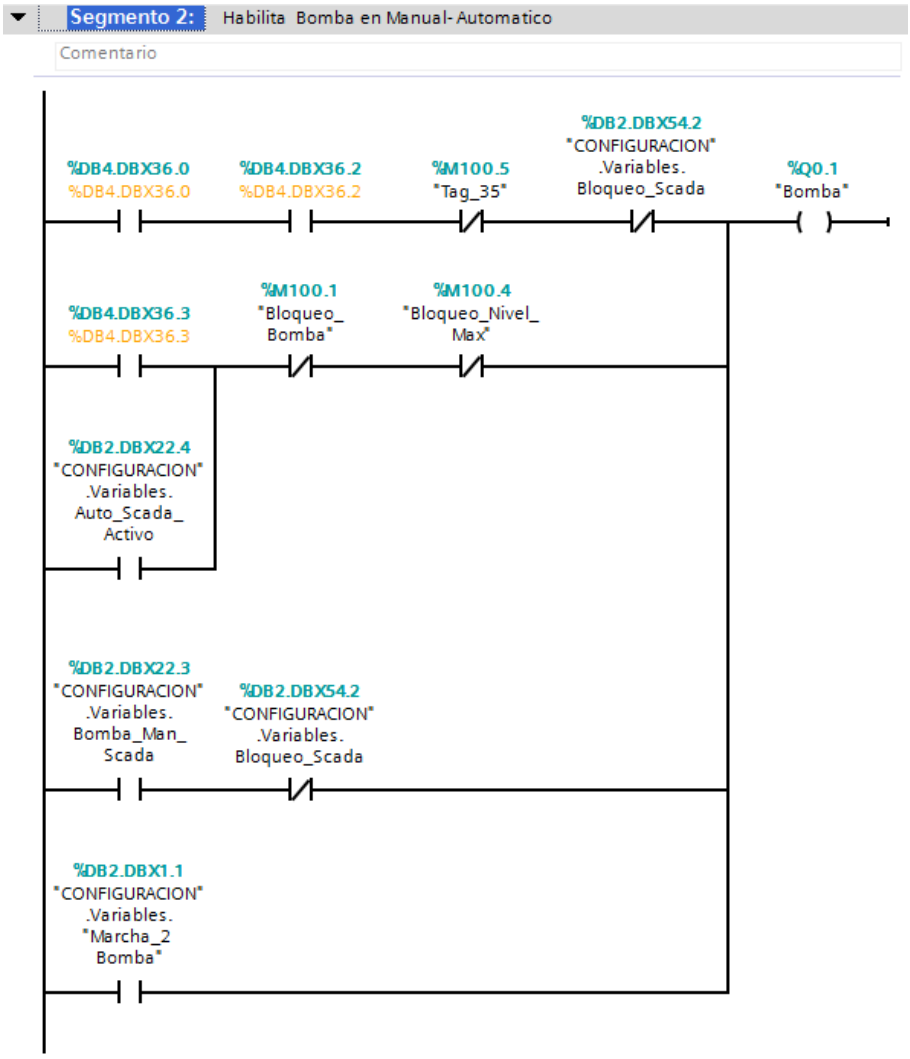


Figura 284: Activación y desactivación de la Bomba.

Una vez configurada la bomba pasaremos al diseño de activación y desactivación de la electroválvula, para esto lo generamos en el **Segmento 3**. El circuito de control consta de marcas y contactos de tipo set – reset para la activación y desactivación de la salida Q0.0, la cual es una salida digital que fue asignada a la electroválvula. Además para que se active la electroválvula se agregó un contacto abierto correspondiente a la bomba de llenado, es decir, solamente se podrá activar la electroválvula cuando esté apagada la bomba y viceversa.

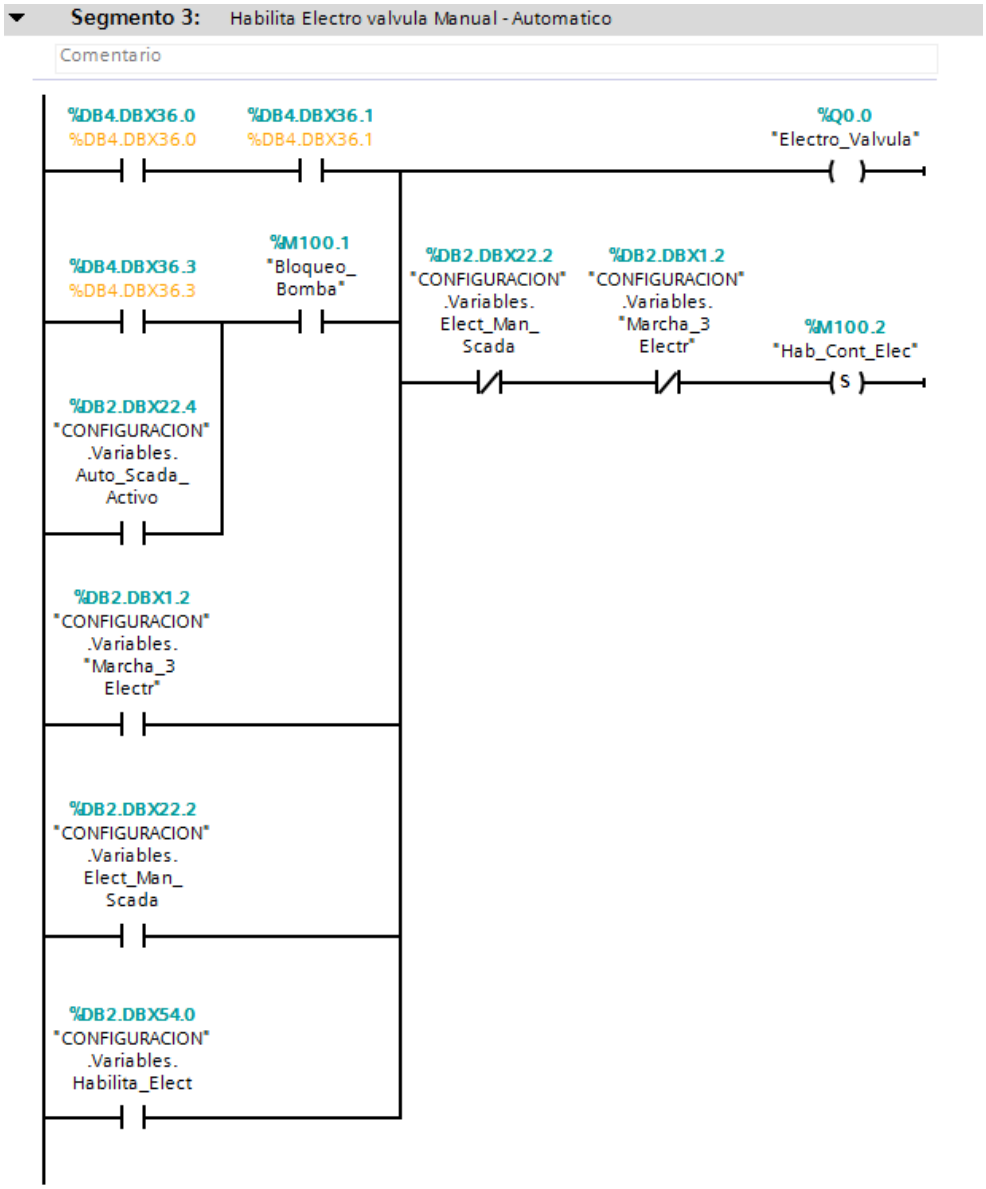


Figura 285: Activación y desactivación de la Electroválvula.

Al momento que se activa el modo Automático la bomba comenzará a trabajar y será controlada por dos temporizadores, cuando el nivel de agua sobrepase el sensor de boya LS102.1 del tanque TK2 se activará un temporizador que contará hasta 20 segundos, este tiempo fue calculado para que el nivel llegue al máximo (30 cm). Luego que pasan los 20 segundos se desactiva la bomba e inmediatamente se activa la electroválvula, a su vez también se activa un temporizador que contará 35 segundos que es el tiempo de vaciado. Luego el proceso se repite.

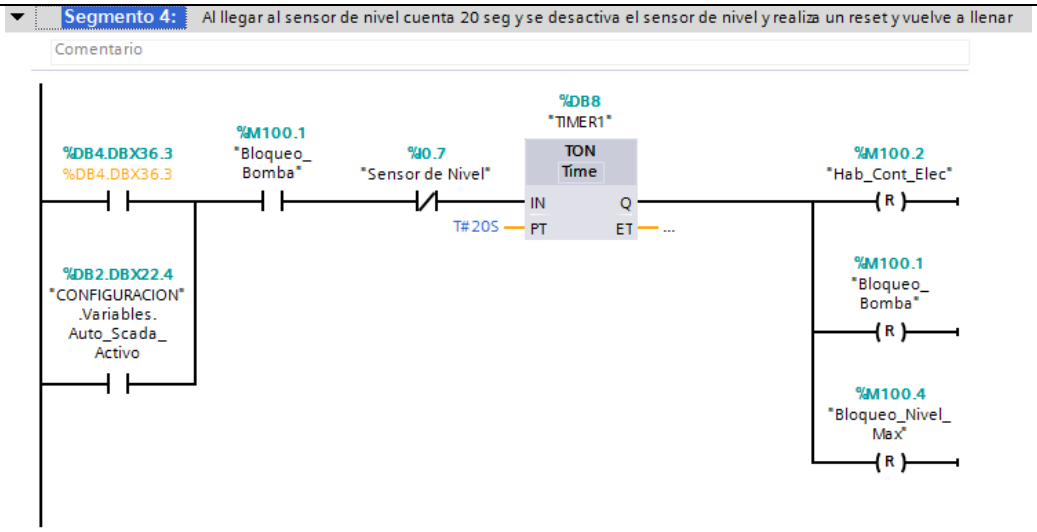


Figura 286: Llenado automático del tanque.

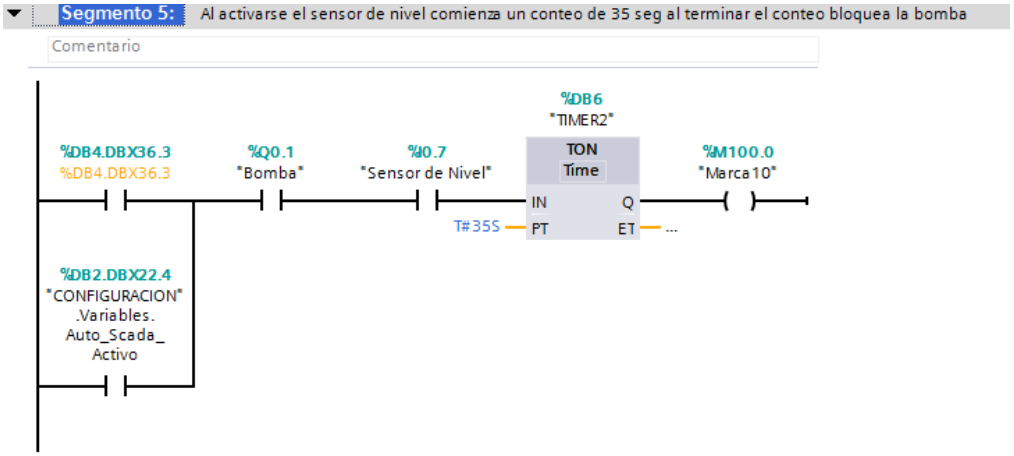


Figura 287: Vaciado automático del tanque.

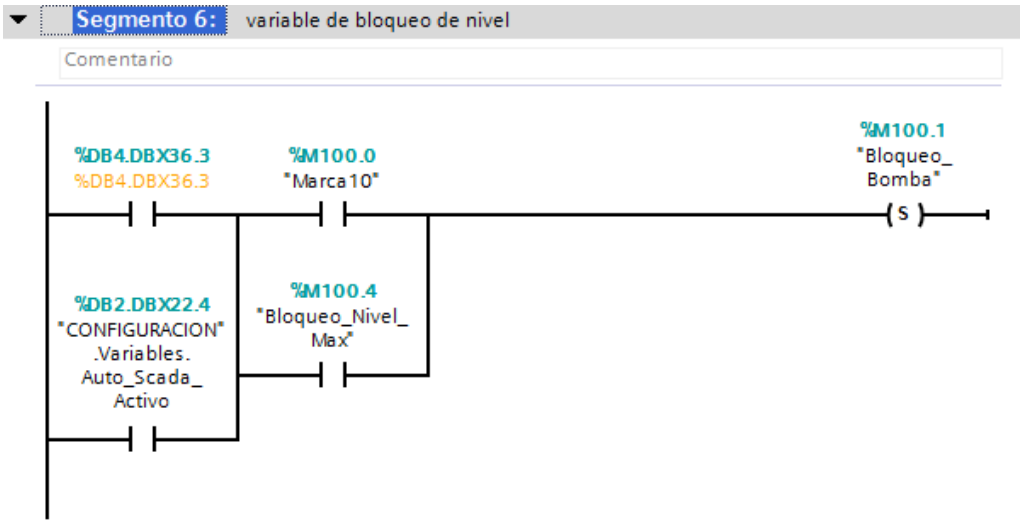


Figura 288: Variable de bloqueo de nivel

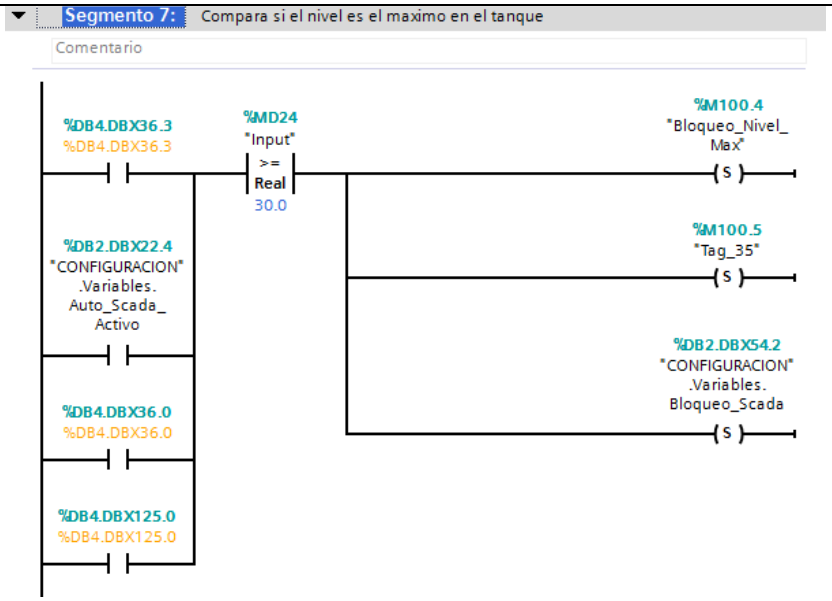


Figura 289: Comparación de nivel máximo del tanque TK102.

Paso 9.

Ahora vamos a crear en el Segmento 12 el Envío de bits, al momento que las entradas y salidas del sistema se activen también se activarán bits que hemos utilizado como marcas dentro del programa, a continuación se detalla cada uno:

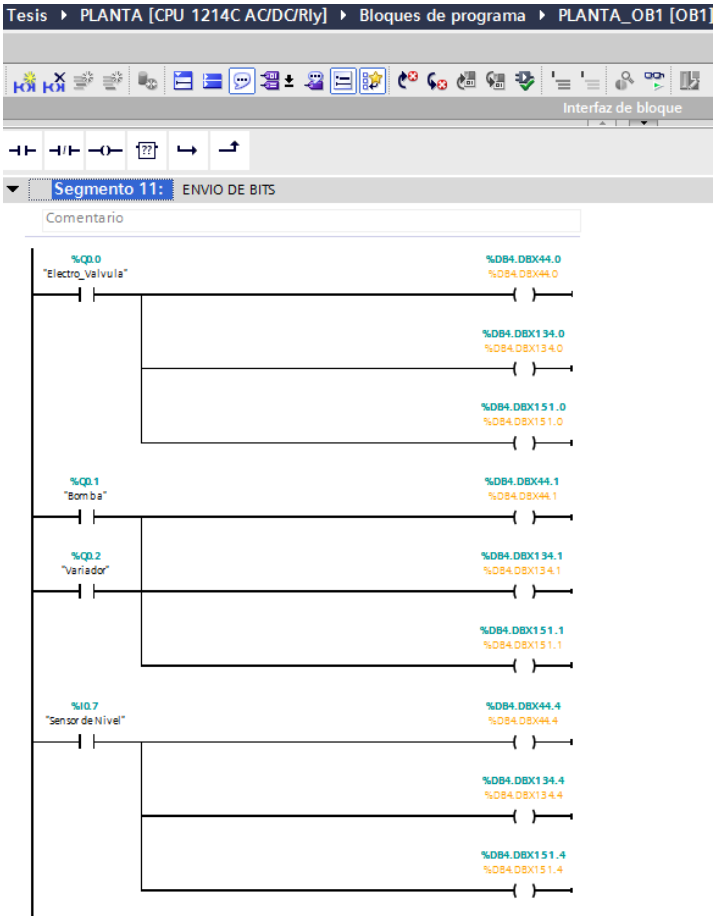


Figura 290: Envío de bits de E/S.

Paso 10.

Ahora se tendrá que crear las imágenes en el **PC-System_1 [SIMATIC PC station].** La imagen **PORTADA** debe quedar definida como imagen principal.

A continuación, se presenta la imagen de **PORTADA** en la Figura 291.

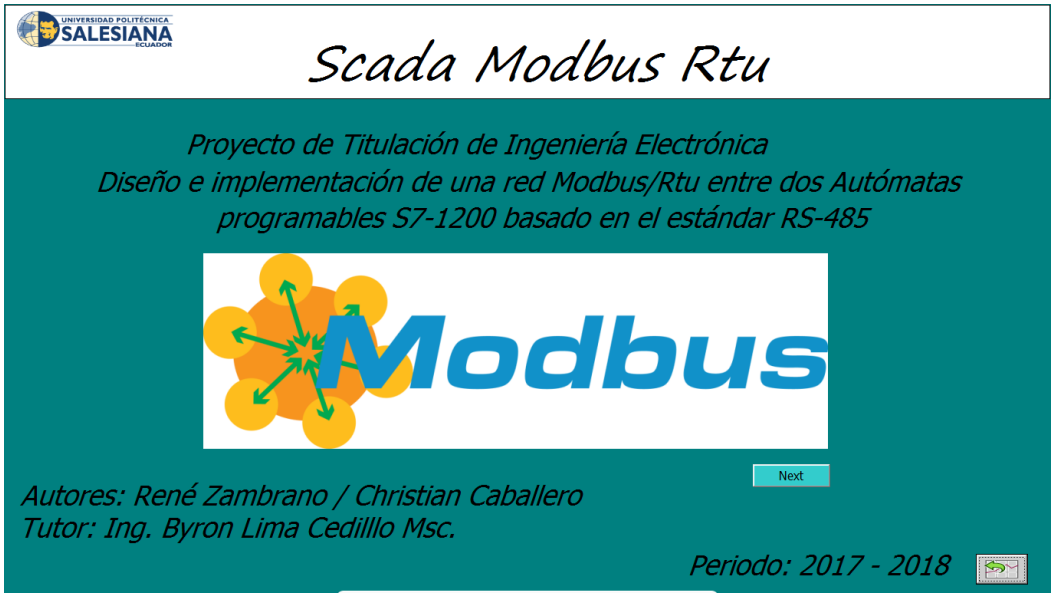


Figura 291: Diseño de imagen de presentación SCADA.

A continuación, se presenta la imagen de **Practica # 8** en la Figura 292.

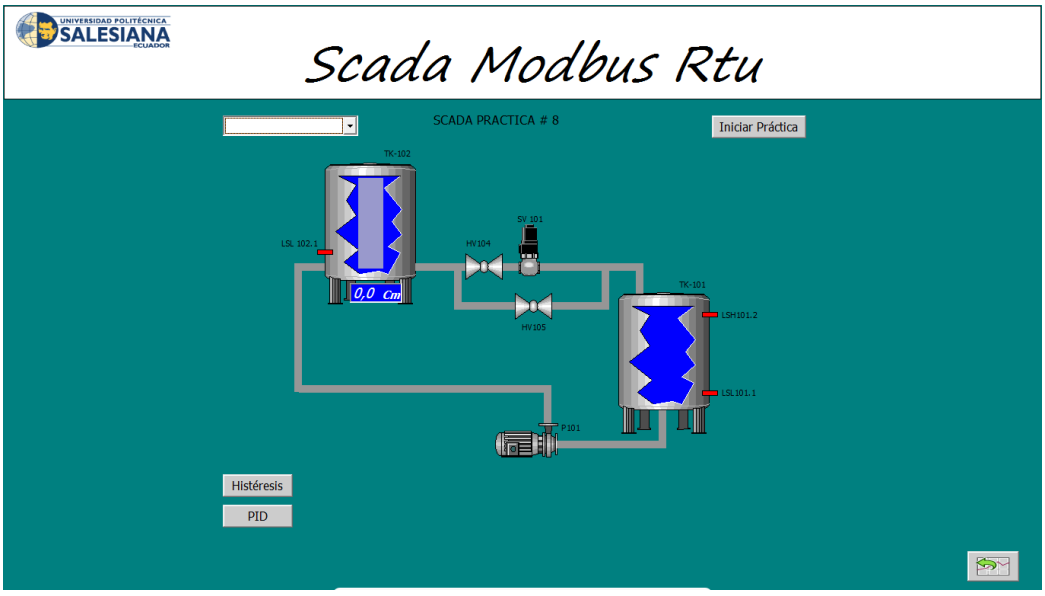


Figura 292: Diseño de imagen Practica # 8 SCADA.

Paso 11.

Ahora procederemos a configurar los controles de la imagen Presentación:

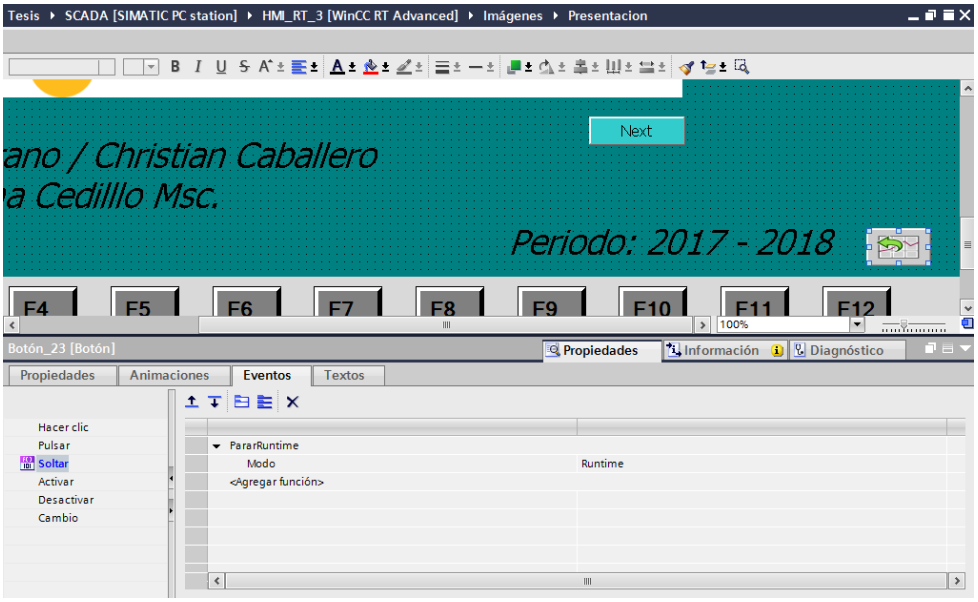


Figura 293: Configuración botón salir de la pantalla Presentación.

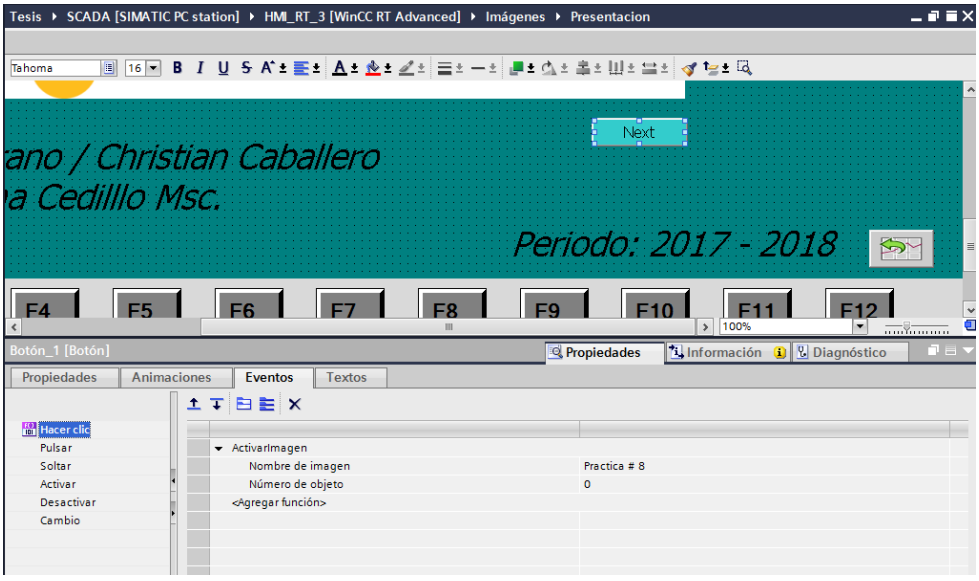


Figura 294: Configuración botón Next de la pantalla Presentación.

Paso 12.

Ahora procederemos a configurar los controles de la imagen **Practica # 8** donde vamos a agregar dos barras las cuales tendrá la función de simular el llenado y vaciado de ambos tanques en la pantalla HMI. Esto se realizara en tiempo real comparado con la planta didáctica. También se agregó un indicador de E/S para mostrar el valor numérico del nivel del tanque TK1:

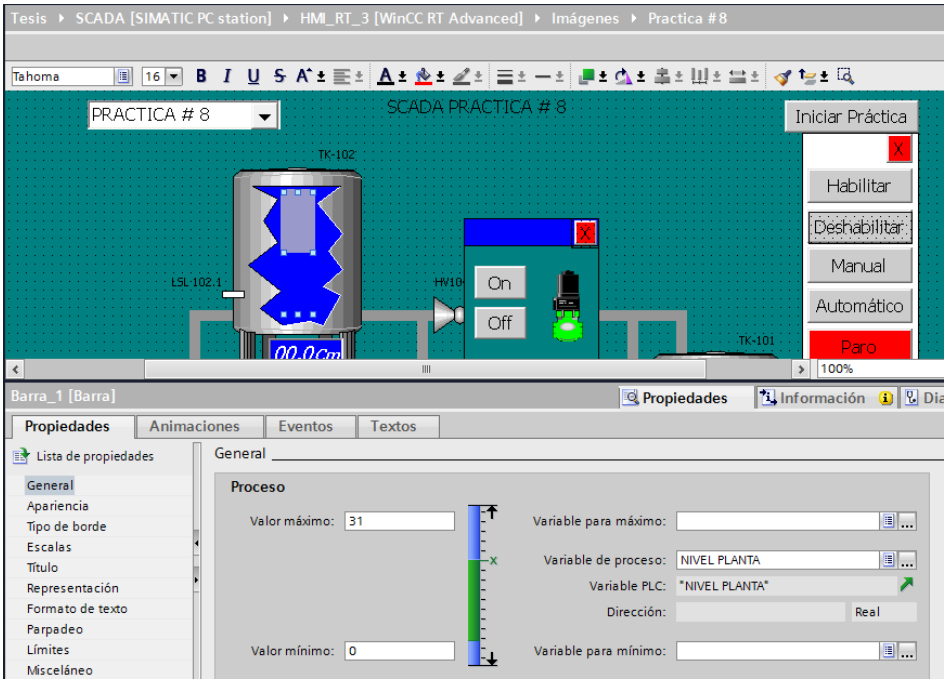


Figura 295: Configuración de Barra_1.

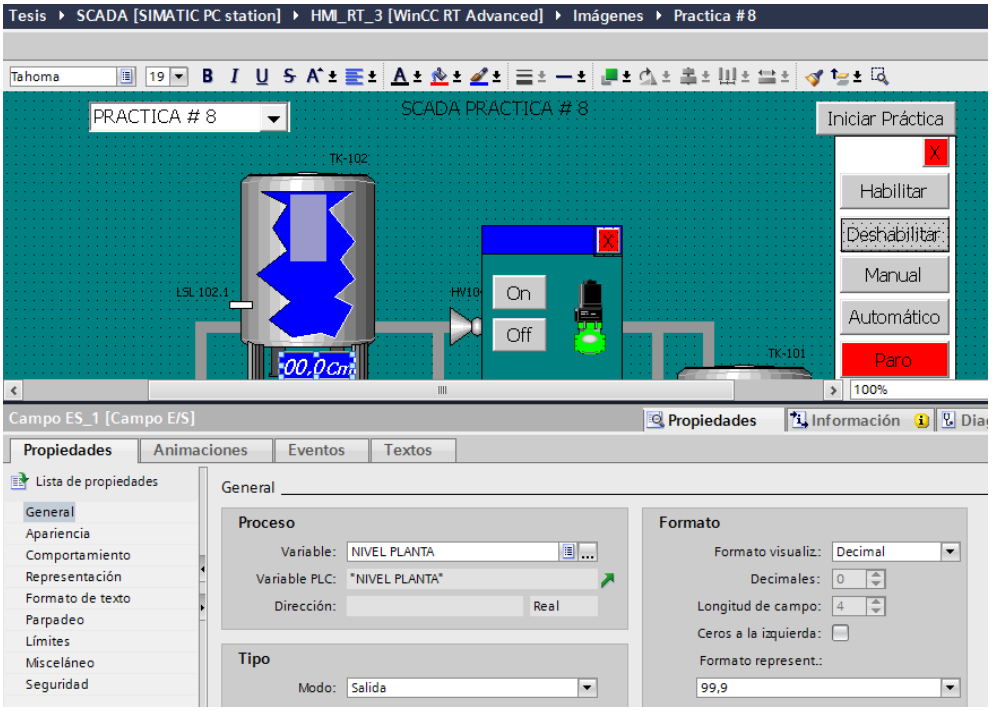


Figura 296: Configuración de Campo ES_1 nivel.

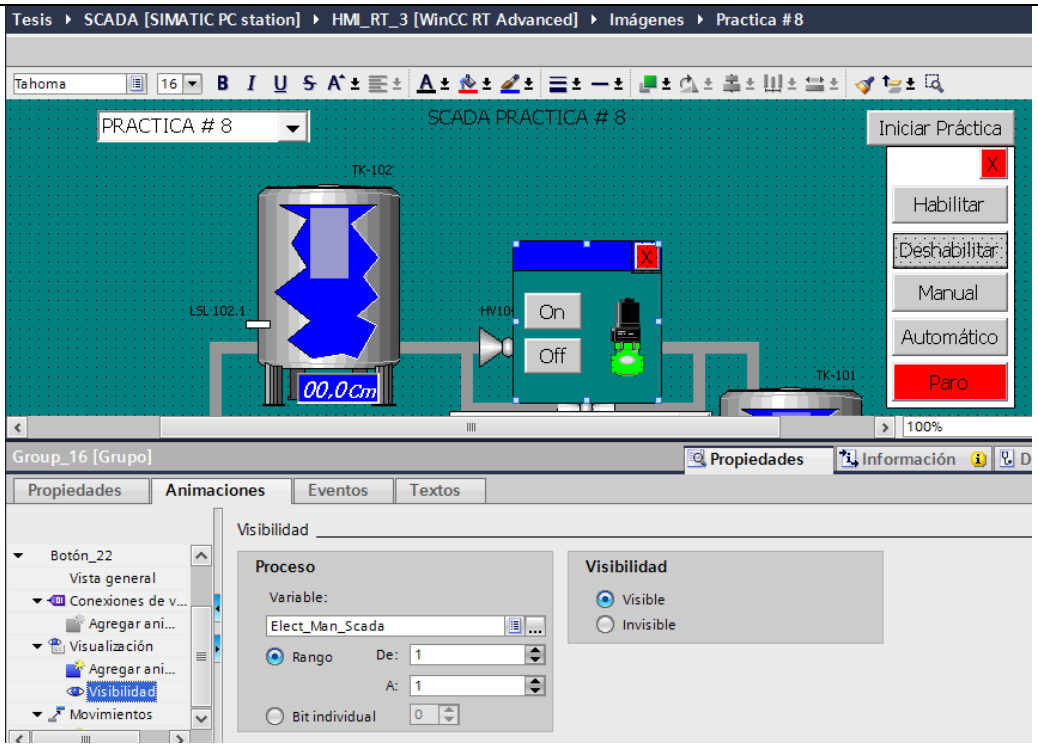


Figura 297: Configuración de Grupo_16 electroválvula manual.

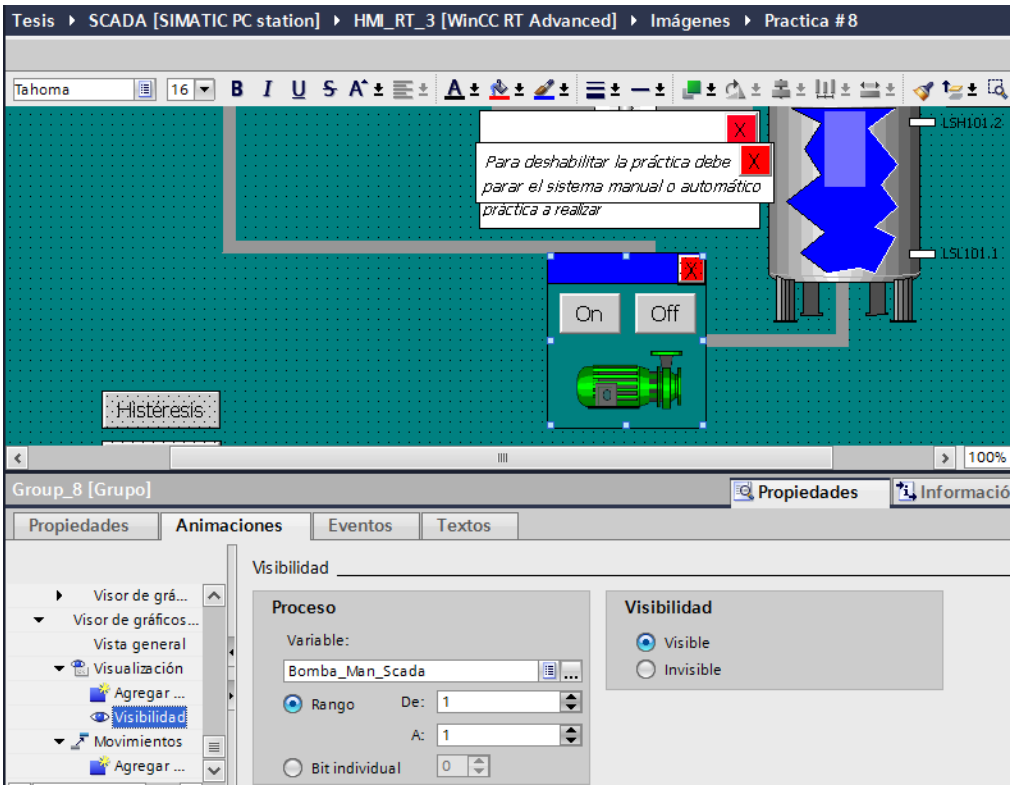


Figura 298: Configuración de Grupo_8 bomba manual.

Recomendaciones.

- Comprobar que las variables estén identificadas y con nombre para cada parte del proceso.
- Revisar que las variables de los controles del SCADA estén correctamente asignadas.
- Verificar que los datos enviados y recibidos se guarden en las respectivas variables asignadas.
- Conectar de forma adecuada la planta de nivel de agua, sensores, alimentación, entrada y salida.

4.9. Práctica Nº 9

<table>
<tr><td colspan="2"> </td><td colspan="2">GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td colspan="2">ASIGNATURA: Automatización</td></tr>
<tr><td colspan="2">NRO. PRÁCTICA:</td><td>9</td><td>TÍTULO PRÁCTICA: “Control de nivel tipo ON/OFF con histéresis para una planta didáctica.”</td></tr>
<tr><td colspan="4">OBJETIVOS:
• Diseñar un sistema de llenado de un tanque con histéresis entre un maestro y un esclavo.
• Implementar un control de nivel de una planta de agua vía comunicación MODBUS RTU.</td></tr>
<tr><td colspan="2" rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="4">MATERIALES:
• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• Dos módulos de comunicación CM1241.
• Un cable de comunicación RS485 macho – macho.
• Dos tanques de 10 litros de capacidad (TK1 / TK2).
• Una electroválvula (SV101).
• Una bomba centrifuga (P101).
• Un sensor ultrasónico (TK102) LIC102.2
• Un sensor tipo boya en tanque TK102 (LSL102.1).
• Dos sensores de proximidad capacitivos en TK101 (LSH101.2 / LSL101.1)
• Dos pulsadores, uno para la selección de la práctica (S1) y el otro para resetear la selección (S2).

DESCRIPCIÓN:
Este sistema funciona de la siguiente manera: primero se debe establecer la comunicación en el HMI con el botón Habilitar para después presionar el pulsante S1 en la planta para habilitar la práctica 9. Establecida la comunicación entre maestro (módulo didáctico) y esclavo (planta) seleccionamos el sistema donde vamos a ingresar el valor de Setpoint e Histéresis. Al presionar el botón Start se activa la bomba P101 hasta llegar al nivel deseado, luego por acción de la electroválvula el</td></tr>
</table>

nivel baja hasta el valor de histéresis seteado. El sistema va a estar oscilando en los niveles superior o inferior del setpoint hasta presionar **Paro**.

La figura 299 muestra la arquitectura de red a implementar.

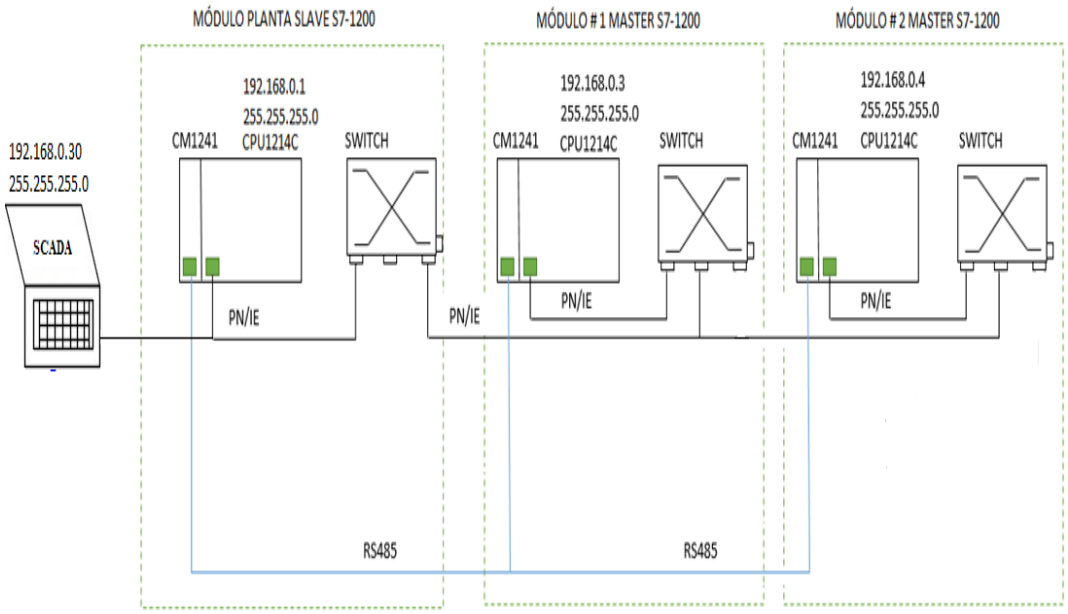


Figura 299: Arquitectura de red.

En figura 300 se muestra el diagrama de conexiones a implementar.

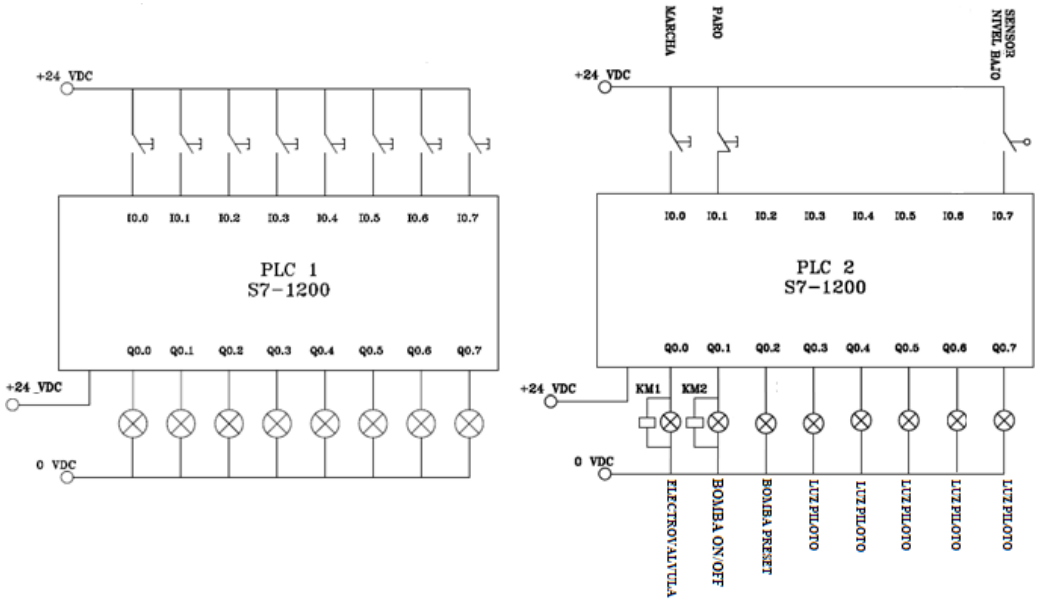
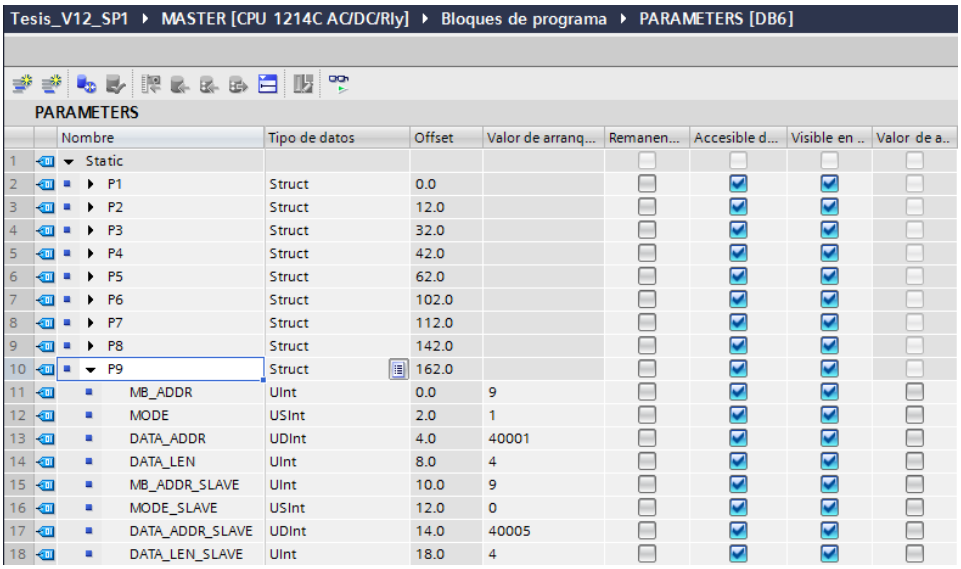


Figura 300: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear los parámetros para las instrucciones de la práctica 9, nos dirigimos a **PARAMETERS [DB6]** ubicado en el PLC **MASTER** y creamos las siguientes variables:



Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PARAMETERS [DB6]

PARAMETERS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
1	Static							
2	P1	Struct	0.0					
3	P2	Struct	12.0					
4	P3	Struct	32.0					
5	P4	Struct	42.0					
6	P5	Struct	62.0					
7	P6	Struct	102.0					
8	P7	Struct	112.0					
9	P8	Struct	142.0					
10	P9	Struct	162.0					
11	MB_ADDR	UInt	0.0	9				
12	MODE	USInt	2.0	1				
13	DATA_ADDR	UDInt	4.0	40001				
14	DATA_LEN	UInt	8.0	4				
15	MB_ADDR_SLAVE	UInt	10.0	9				
16	MODE_SLAVE	USInt	12.0	0				
17	DATA_ADDR_SLAVE	UDInt	14.0	40005				
18	DATA_LEN_SLAVE	UInt	18.0	4				

Figura 301: Parámetros del PLC MASTER, práctica 9.

Observación: También se necesita crear los parámetros para el PLC **SLAVE** los cuales van a ser los mismos que el **MASTER**. El PLC **PLANTA** no requiere de un bloque de parámetros ya que solo va a estar compuesto de bloques **MB_SLAVE_DB**, es decir la planta de nivel de agua funcionará como esclavo MODBUS.

Paso 2.

Ahora debemos generar las variables de registro para el PLC **MASTER** en **PRACTICAS [DB4]**. La imagen a continuación mostrará las variables de registro de datos.

Tesis_V12_SP1 ▸ MASTER [CPU 1214C AC/DC/Rly] ▸ Bloques de programa ▸ PRACTICAS [DB4]

PRACTICAS

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..
16	PRACTICA 9	Struct	136.0					
17	ENVIAR	Array [1..4] of Int	0.0					
18	ENVIAR[1]	Int	0.0	0				
19	ENVIAR[2]	Int	2.0	0				
20	ENVIAR[3]	Int	4.0	0				
21	ENVIAR[4]	Int	6.0	0				
22	RECIBIR	Array [1..2] of Real	8.0					
23	RECIBIR[1]	Real	0.0	0.0				
24	RECIBIR[2]	Real	4.0	0.0				
25	PRACTICA 10	Struct	152.0					

Figura 302: Variables de registro del PLC MASTER, práctica 9.

En la siguiente figura se mostrarán las variables de registro para el PLC PLANTA:

Figura 303: Variables de registro del PLC PLANTA, práctica 9.

Paso 3.

Nos dirigimos a **PLANTA_OB1** donde crearemos el bloque de instrucción **MB_SLAVE_DB** que receptará los datos de comunicación en la planta. Esto lo vamos a generar en el **Segmento 3**.

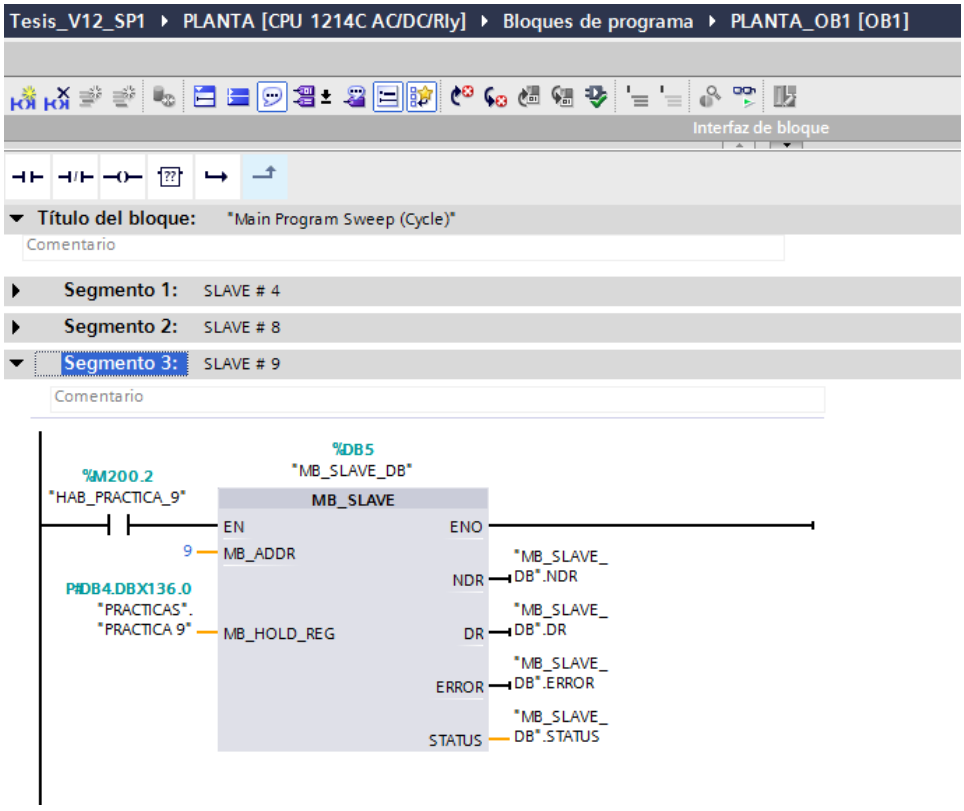


Figura 304: Bloque Slave de práctica 9.

Paso 4.

En este paso vamos a crear los bloques de comunicación para el PLC **MASTER** en el **Segmento 14,** aquí funcionaran dos bloques master donde uno tendrá la labor de escribir datos y el segundo en leer:

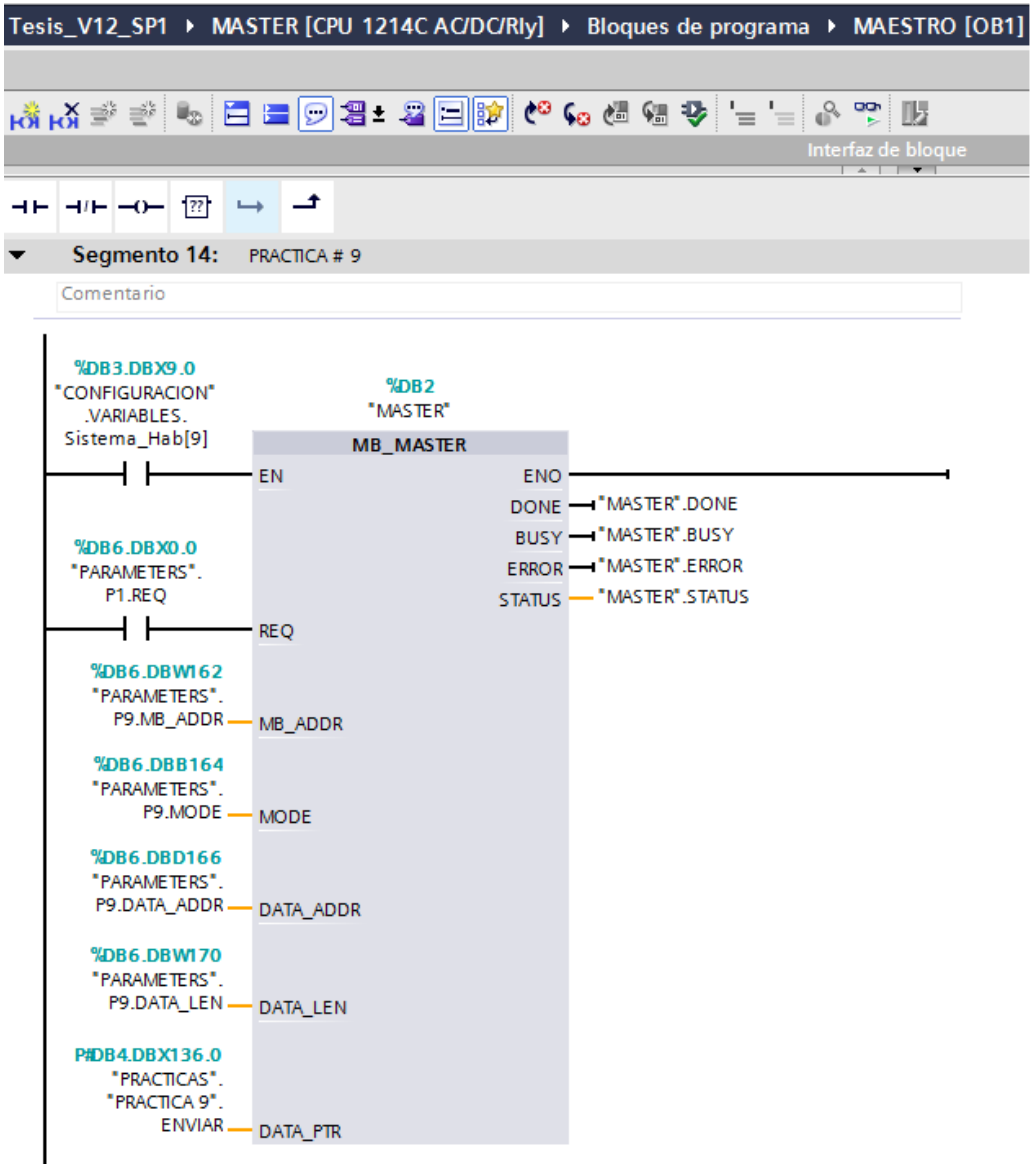


Figura 305: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 9.

El primer bloque de instrucción está funcionando en modo escritura, el cual enviará los datos al slave como activación de la bomba y electroválvula. Debemos agregar un segundo bloque master el cual va a leer el sensor ultrasónico para detección de nivel de agua en los tanques.

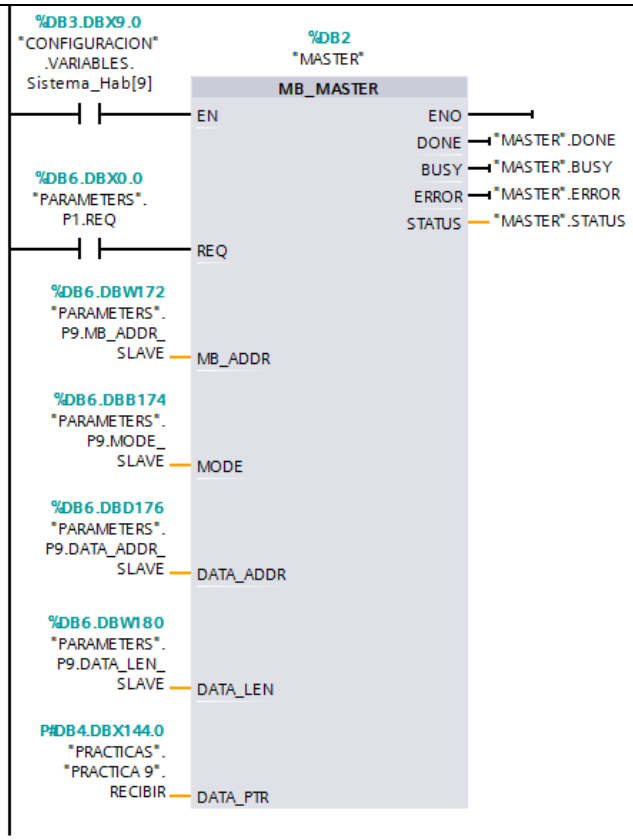


Figura 306: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 8.

Paso 5.

Ahora detallaremos el circuito de control para el PLC MASTER, en el **Segmento 15** tenemos la selección del modo, SCADA o HMI:

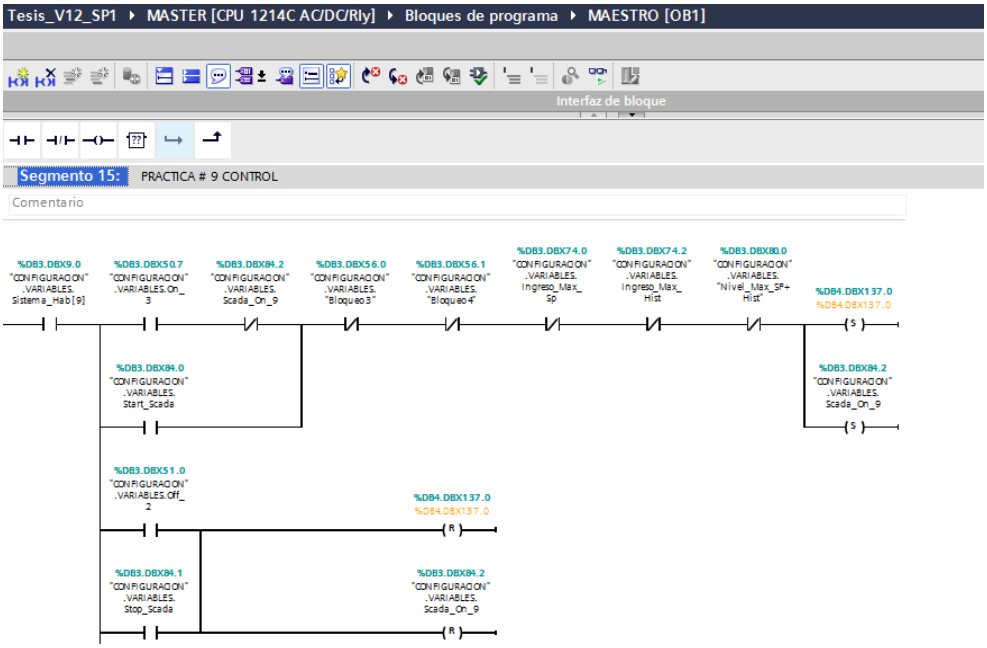


Figura 307: Activación de sistema vía SCADA.

A continuación guardamos los registros en las variables correspondientes:

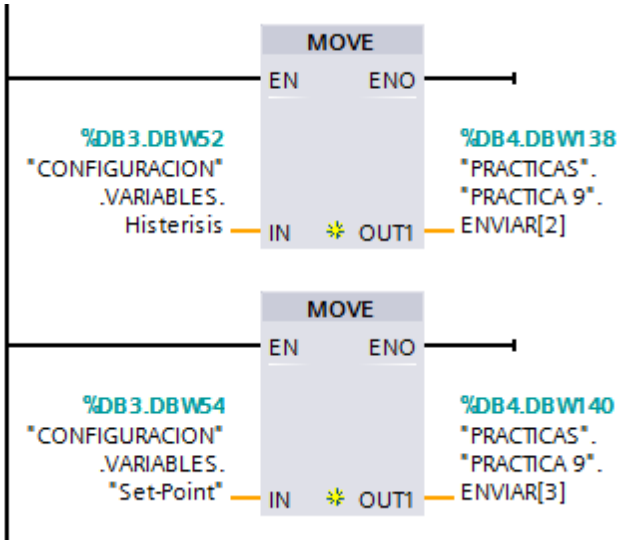


Figura 308: Almacenamiento de datos de Histéresis y Setpoint.

En caso de ingresar datos incoherentes o fuera de límite se establecen los mensajes de alerta con los bloqueos correspondientes.

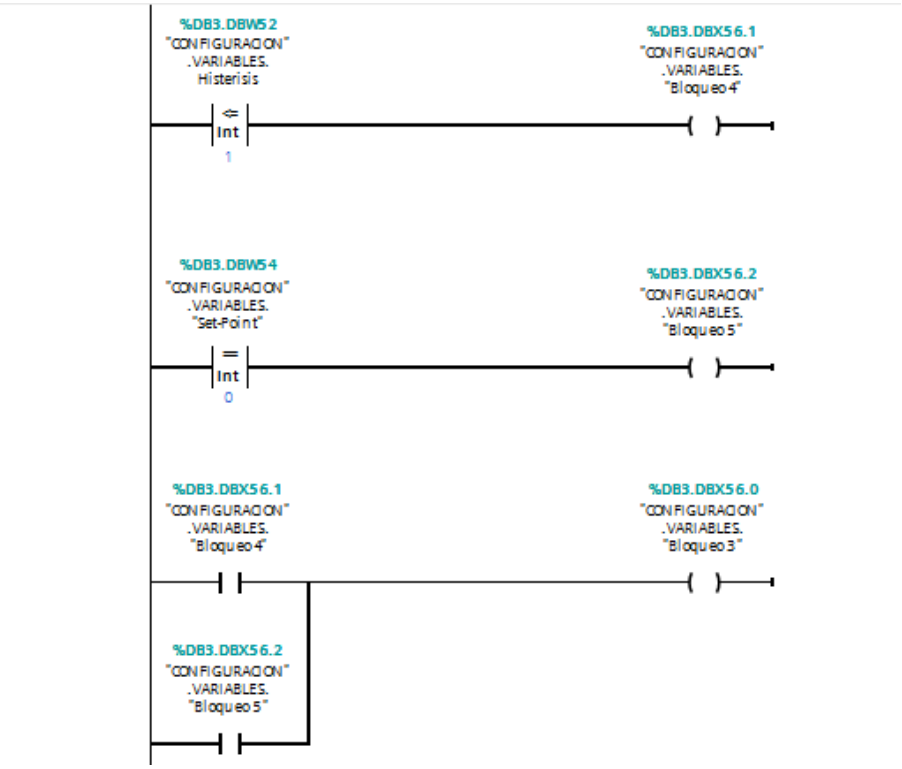


Figura 309: Bloqueos del sistema.

Ahora se muestra la asignación de histéresis en las variables, es decir el valor ingresado se divide para 2 y el resultado se suma y se guarda en la variable **Result_SP+Hist**, a su vez el resultado también se resta y se guarda en la variable **Result_SP-Hist.**

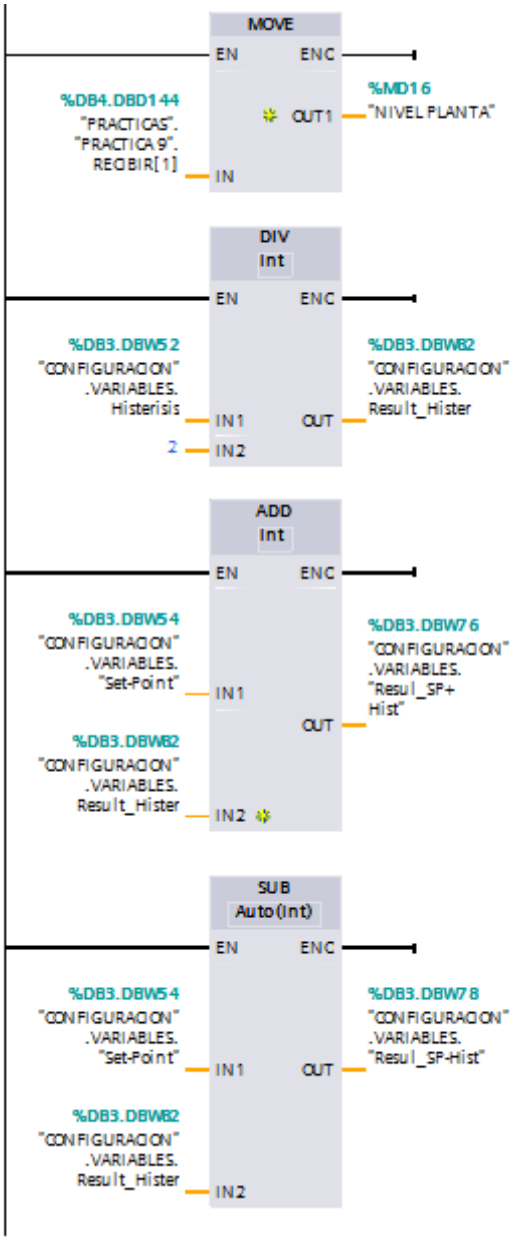


Figura 310: Operación matemática de histéresis.

Paso 6.

Ahora programaremos los bloqueos del sistema.

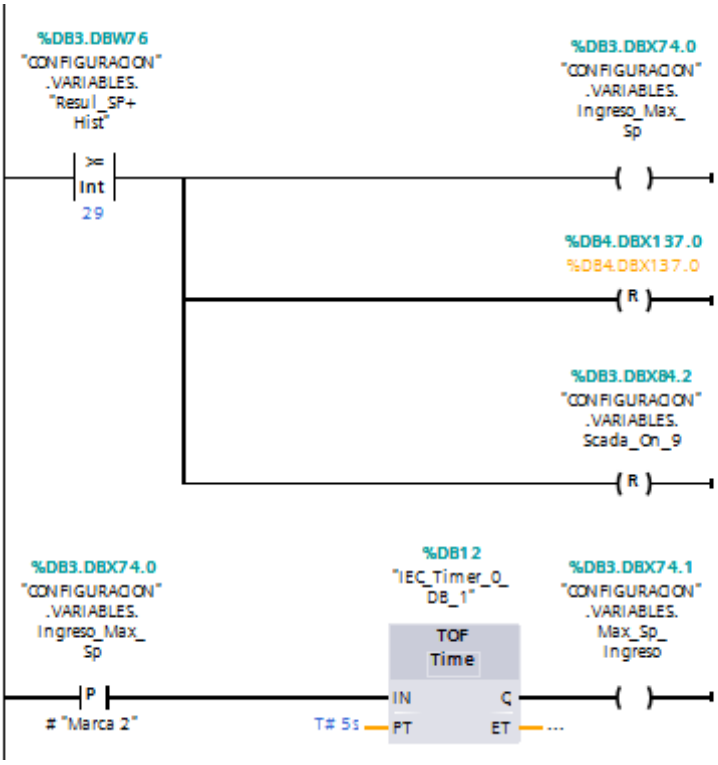


Figura 311: Bloqueo para límite superior de setpoint.

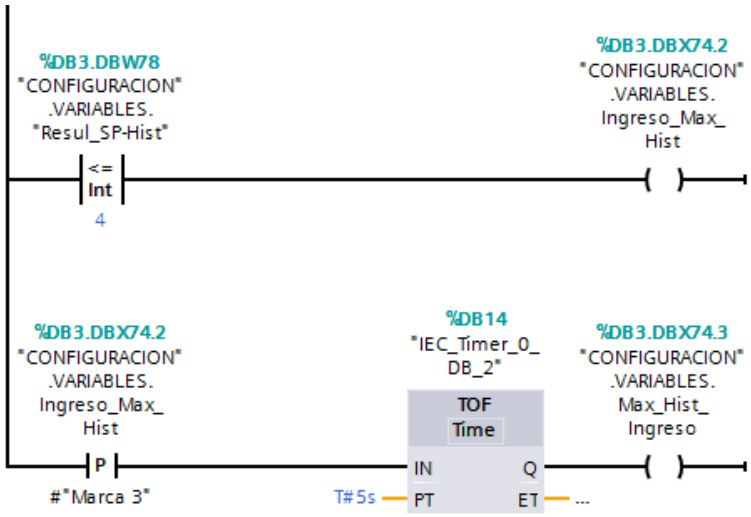


Figura 312: Bloqueo para límite superior de histéresis.

Paso 7.

Ahora detallaremos la programación del control de la PLANTA, creamos un bloque de función sin memoria FC llamada PRACTICA # 9 SLAVE [FC3].

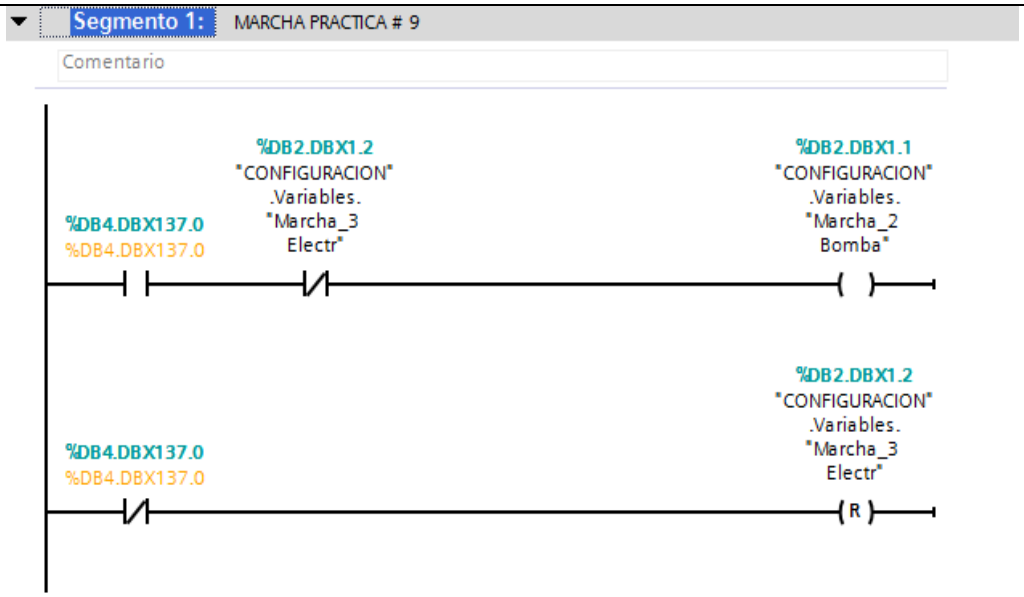


Figura 313: Control de encendido de bomba y electroválvula.

En el Segmento 2 realizaremos una conversión de entero a real para visualizar en el HMI y SCADA con valores decimales.

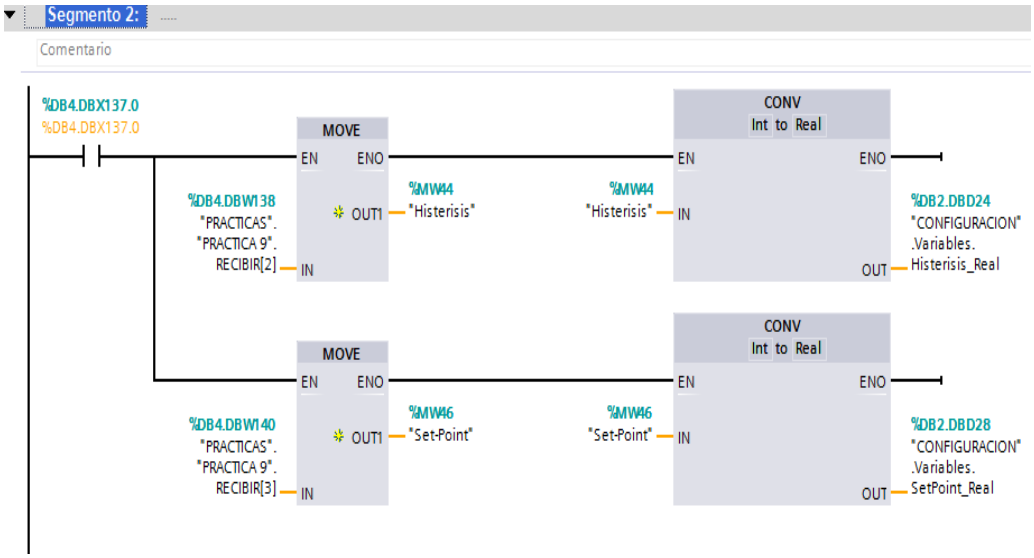


Figura 314: Visualización de Histéresis y Setpoint.

Para la activación de la electroválvula es necesario realizar una comparación, cuando se sobrepase el valor de histéresis + se activa la electroválvula. Cuando la entrada MD24 sea menor o igual al histéresis – se desactiva la electroválvula

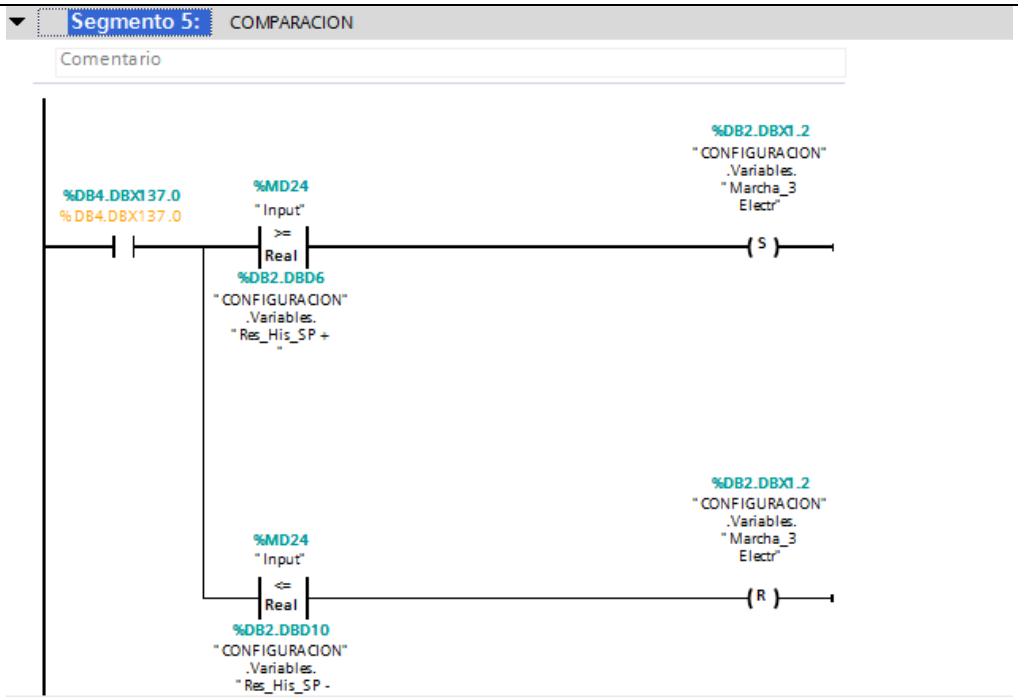


Figura 315: Activación / desactivación de electroválvula.

Paso 8.

Daremos paso a la creación de imágenes del proyecto.

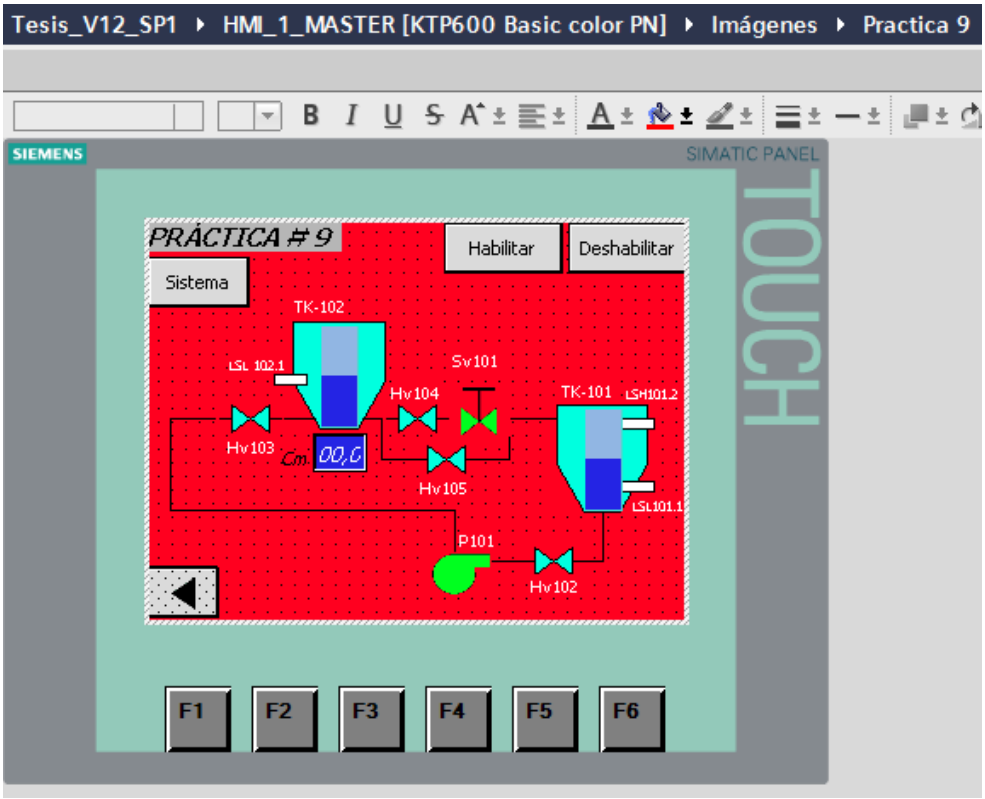


Figura 316: Diseño de imagen Practica 9.

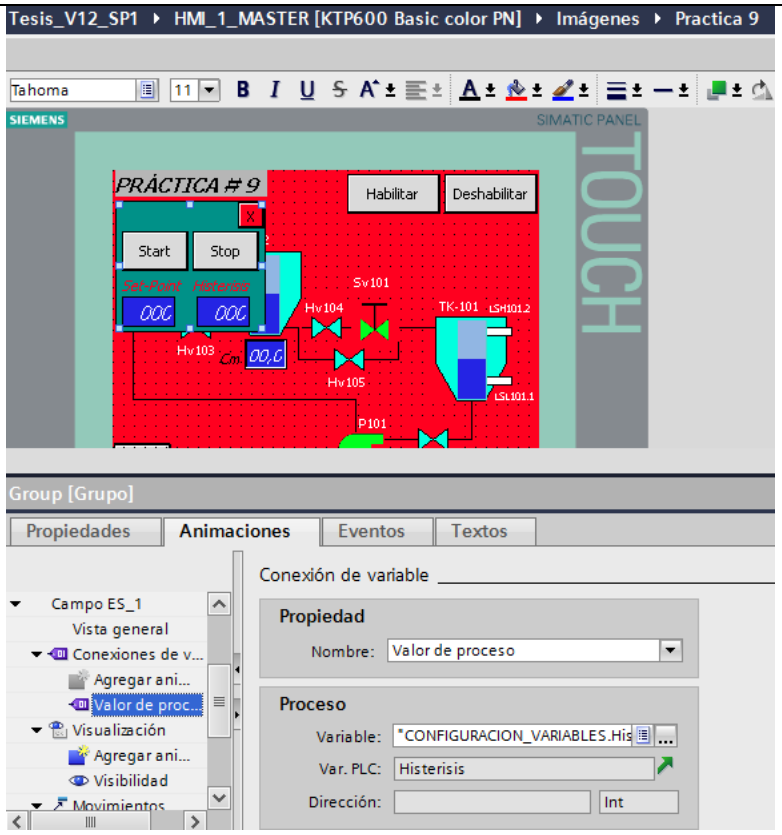


Figura 317: Configuración de Grupo, botón Star / Stop.

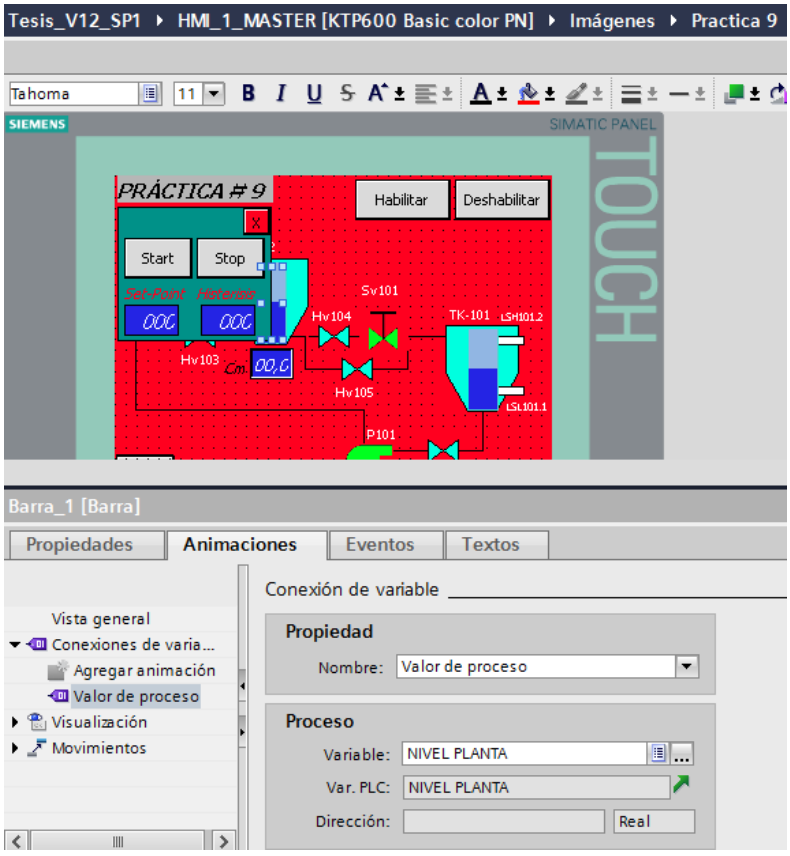


Figura 318: Configuración de barra, tanque TK2.

Paso 9.

Ahora se mostrarán las gráficas obtenidas del control de histéresis, para esto insertamos un visor de curvas el cual vamos a añadir las entradas set point, nivel, histéresis positiva y negativa, para que sean visualizadas en la pantalla HMI de ambos módulos:

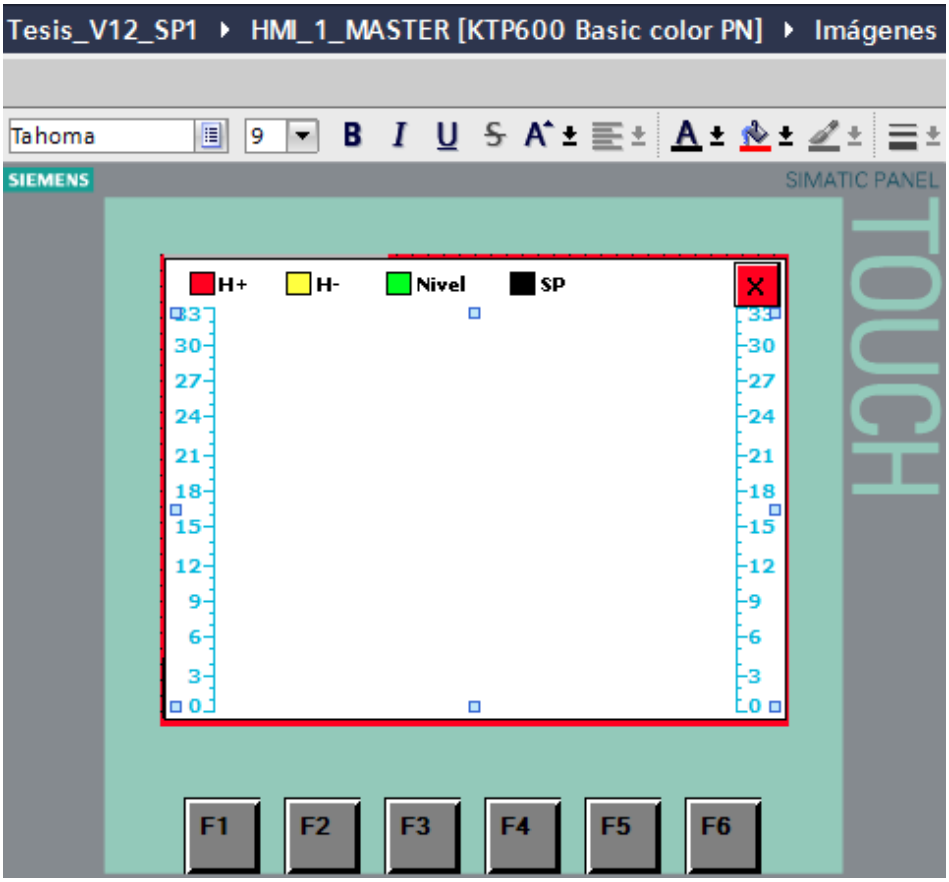


Figura 319: Visor de curvas para histéresis.

En la opción **Curva** agregamos cuatro elementos que serán asignadas a las variables anteriormente nombradas, es importante darle el **Estilo** adecuado para una mejor apreciación.

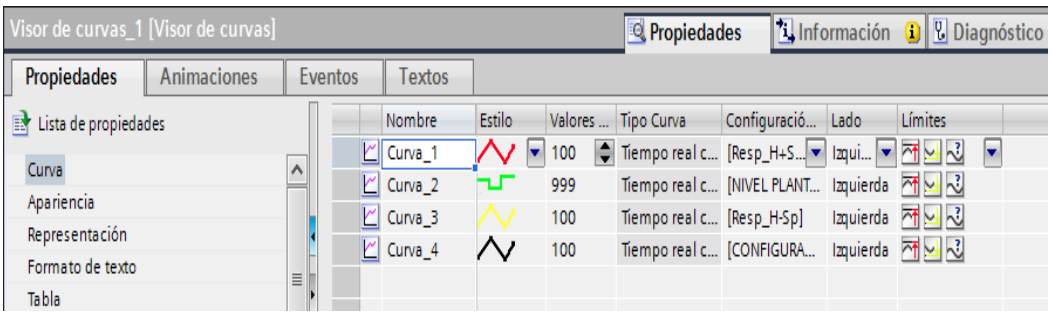


Figura 320: Asignación de variables para gráfico de histéresis.

Una vez configurado el visor de visualización procedemos a ejecutar el programa donde daremos diferentes valores de histéresis y set point para evidenciar el comportamiento de la gráfica.

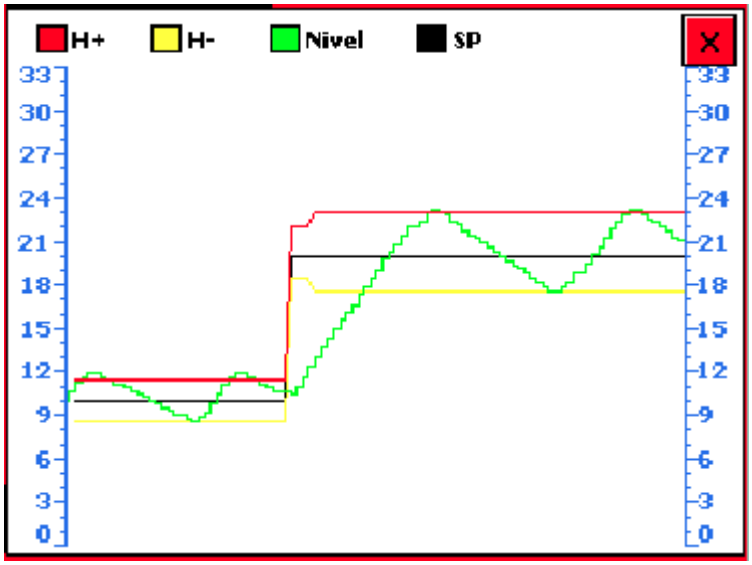


Figura 321: Visualización de resultados de histéresis.

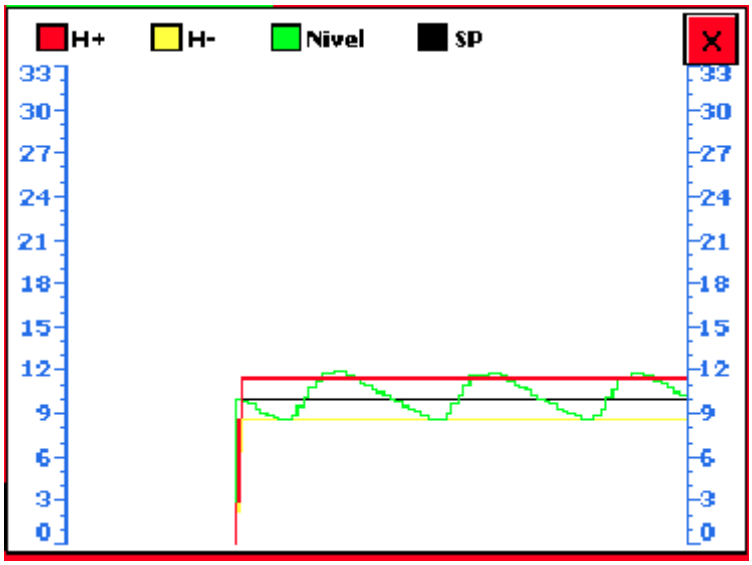


Figura 322: Visualización de resultados de histéresis.

Recomendaciones.

- Comprobar que las variables estén identificadas y con nombre para cada parte del proceso.

- Revisar que las variables de los controles del SCADA y HMI estén correctamente asignadas.
- Verificar que los datos enviados y recibidos se guarden en las respectivas variables asignadas.
- Conectar de forma adecuada la planta de nivel de agua, sensores, alimentación, entrada y salida.

4.10. Práctica 10

<table>
<tr><td colspan="2"> </td><td>GUÍA DE PRÁCTICA DE LABORATORIO</td></tr>
<tr><td colspan="2">CARRERA: Ingeniería Electrónica</td><td>ASIGNATURA: Automatización</td></tr>
<tr><td>NRO. PRÁCTICA:</td><td>10</td><td>TÍTULO PRÁCTICA: “Aplicación de un controlador PID (Método Experimental) utilizando una planta didáctica para control de nivel.”</td></tr>
<tr><td colspan="3">OBJETIVOS:
• Diseñar un sistema de llenado de un tanque con PID entre un maestro y un esclavo.
• Implementar un control de nivel de una planta de agua vía comunicación MODBUS RTU.</td></tr>
<tr><td rowspan="4">INSTRUCCIONES:</td><td colspan="2">1. Alistar los elementos a utilizar para la práctica, energizando con 120 V.</td></tr>
<tr><td colspan="2">2. Revisar la disponibilidad y funcionamiento de los elementos a utilizar.</td></tr>
<tr><td colspan="2">3. Comprobar las diferentes conexiones para un correcto funcionamiento de la práctica.</td></tr>
<tr><td colspan="2">4. Cumplir el procedimiento de la guía paso a paso de forma ordenada.</td></tr>
<tr><td colspan="3">MATERIALES:
• Dos módulos didácticos con PLC Siemens S7-1200 y HMI KTP-600.
• Dos módulos de comunicación CM1241.
• Un cable de comunicación RS485 macho – macho.
• Dos tanques de 10 litros de capacidad (TK1 / TK2).
• Una electroválvula (SV101).
• Una bomba centrifuga (P101).
• Un sensor ultrasónico (TK102) LIC102.2
• Un sensor tipo boya en tanque TK102 (LSL102.1).
• Dos sensores de proximidad capacitivos en TK101 (LSH101.2 / LSL101.1)
• Dos pulsadores, uno para la selección de la práctica (S1) y el otro para resetear la selección (S2).

DESCRIPCIÓN:
Este sistema funciona de la siguiente manera: primero se debe establecer la comunicación en el HMI con el botón Habilitar para después presionar el pulsante S1 en la planta para habilitar la práctica 10. Establecida la comunicación entre maestro (módulo didáctico) y esclavo (planta) seleccionamos el sistema donde vamos a ingresar el valor de Setpoint. Al presionar el botón Start se activa la bomba</td></tr>
</table>

P101 con salida analógica hasta llegar al nivel deseado, luego por inercia el nivel baja es cuando la bomba debe activarse para mantener el nivel deseado. El sistema va a estar perturbado por una válvula manual pero se debe mantener el setpoint, hasta presionar **Paro**.

La figura 323 muestra la arquitectura de red a implementar.

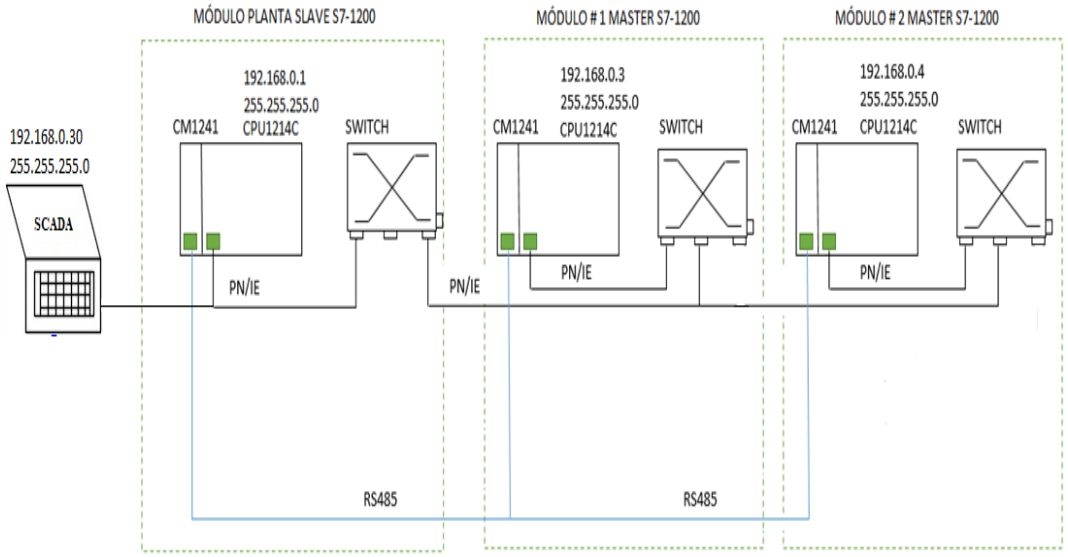


Figura 323: Arquitectura de red.

En figura 324 se muestra el diagrama de conexiones a implementar.

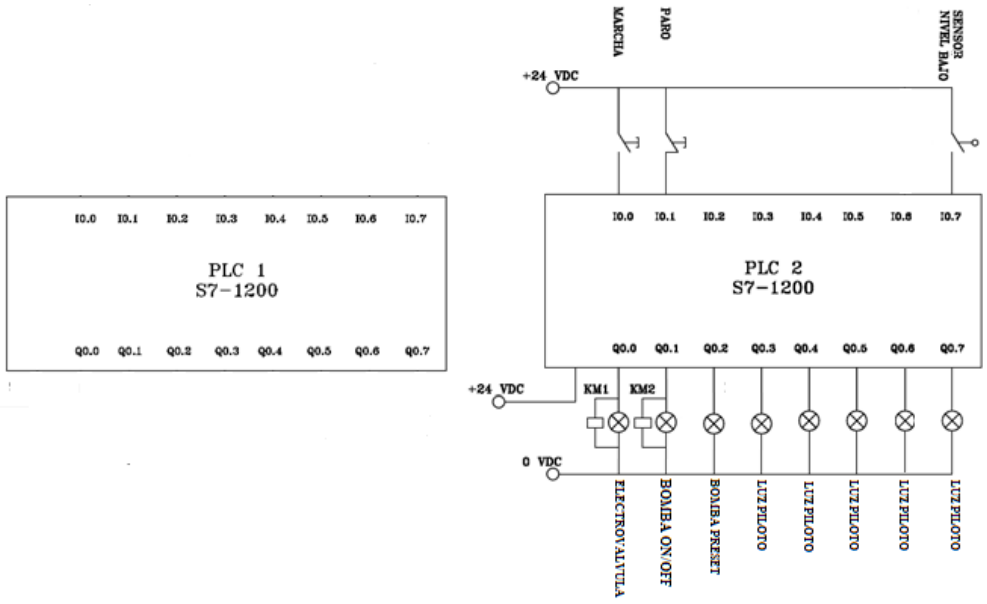
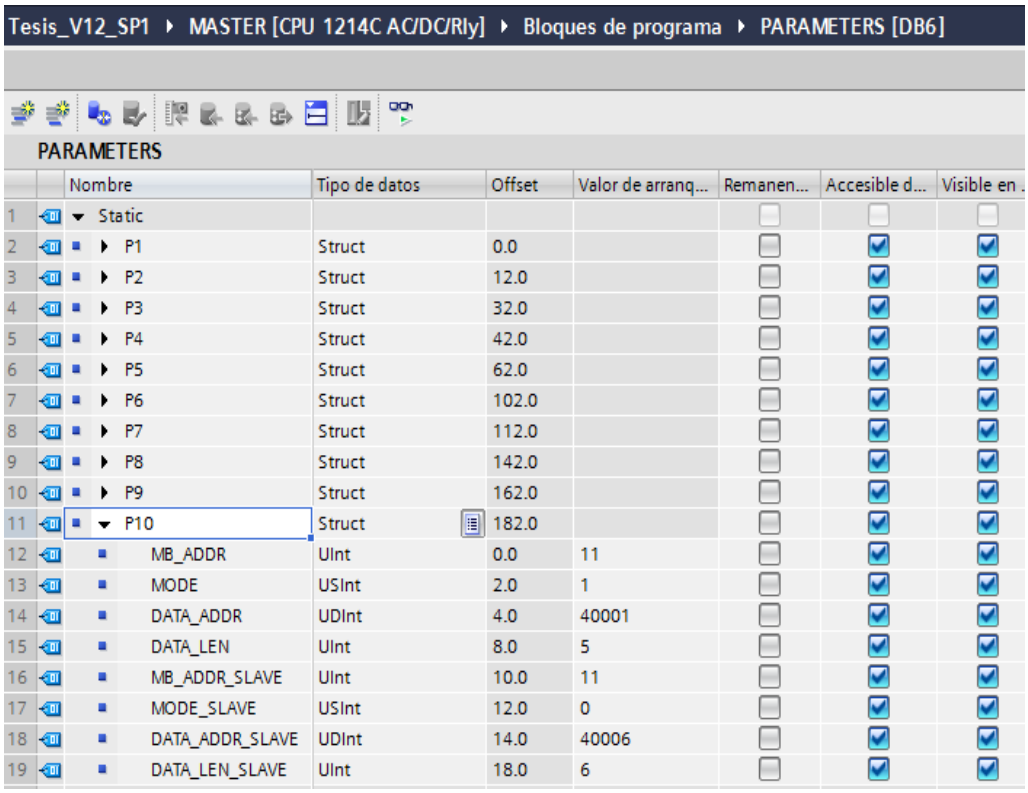


Figura 324: Diagrama de conexiones.

Paso 1.

Como primer paso vamos a crear los parámetros para las instrucciones de la práctica 10, nos dirigimos a **PARAMETERS [DB6]** ubicado en el PLC **MASTER** y creamos las siguientes variables:



	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en
1	Static				☐	☐	☐
2	P1	Struct	0.0		☐	☑	☑
3	P2	Struct	12.0		☐	☑	☑
4	P3	Struct	32.0		☐	☑	☑
5	P4	Struct	42.0		☐	☑	☑
6	P5	Struct	62.0		☐	☑	☑
7	P6	Struct	102.0		☐	☑	☑
8	P7	Struct	112.0		☐	☑	☑
9	P8	Struct	142.0		☐	☑	☑
10	P9	Struct	162.0		☐	☑	☑
11	P10	Struct	182.0		☐	☑	☑
12	MB_ADDR	UInt	0.0	11	☐	☑	☑
13	MODE	USInt	2.0	1	☐	☑	☑
14	DATA_ADDR	UDInt	4.0	40001	☐	☑	☑
15	DATA_LEN	UInt	8.0	5	☐	☑	☑
16	MB_ADDR_SLAVE	UInt	10.0	11	☐	☑	☑
17	MODE_SLAVE	USInt	12.0	0	☐	☑	☑
18	DATA_ADDR_SLAVE	UDInt	14.0	40006	☐	☑	☑
19	DATA_LEN_SLAVE	UInt	18.0	6	☐	☑	☑

Figura 325: Parámetros del PLC MASTER, práctica 10.

Observación: También se necesita crear los parámetros para el PLC **SLAVE** los cuales van a ser los mismos que el **MASTER**. El PLC **PLANTA** no requiere de un bloque de parámetros ya que solo va a estar compuesto de bloques **MB_SLAVE_DB**, es decir, la planta de nivel de agua funcionará como esclavo MODBUS.

Paso 2.

Ahora debemos generar las variables de registro para el PLC **MASTER** en **PRACTICAS [DB4]**. La imagen a continuación mostrará las variables de registro de datos donde contaremos con variables de tipo entero para el envío de datos, y variables de tipo real para recepción de datos.

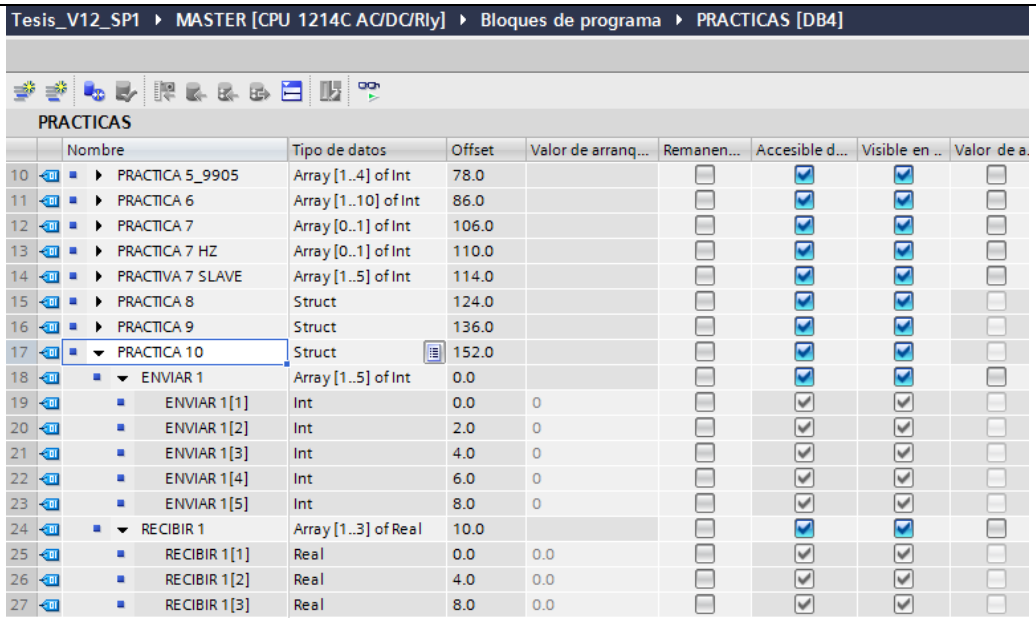


Figura 326: Variables de registro del PLC MASTER, práctica 10.

En la siguiente figura se mostrarán las variables de registro para el PLC PLANTA:

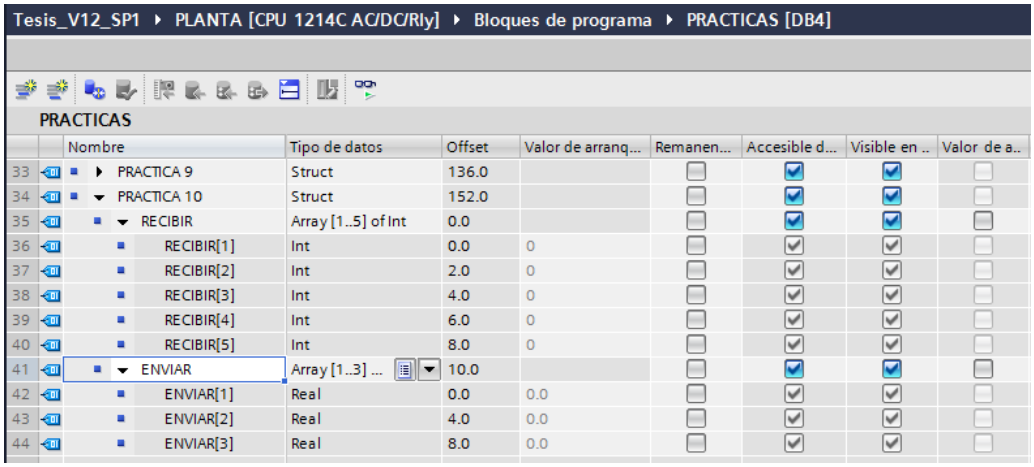


Figura 327: Variables de registro del PLC PLANTA, práctica 10.

Paso 3.

Nos dirigimos a **PLANTA_OB1** donde crearemos el bloque de instrucción **MB_SLAVE_DB** que receptará los datos de comunicación en la planta. Esto lo vamos a generar en el **Segmento 4**.

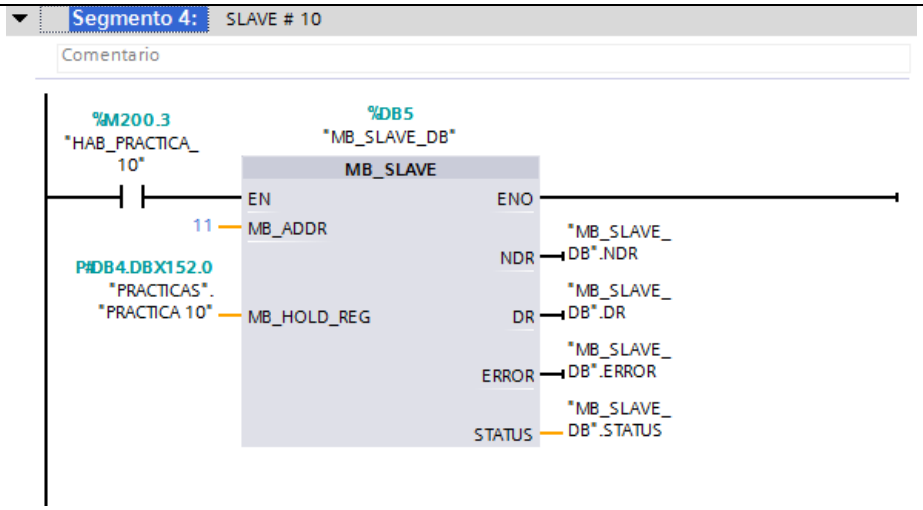


Figura 328: Bloque Slave de práctica 10.

Paso 4.

En este paso vamos a crear los bloques de comunicación para el PLC **MASTER** en el **Segmento 12,** aquí funcionaran dos bloques master donde uno tendrá la labor de escribir datos y el segundo en leer:

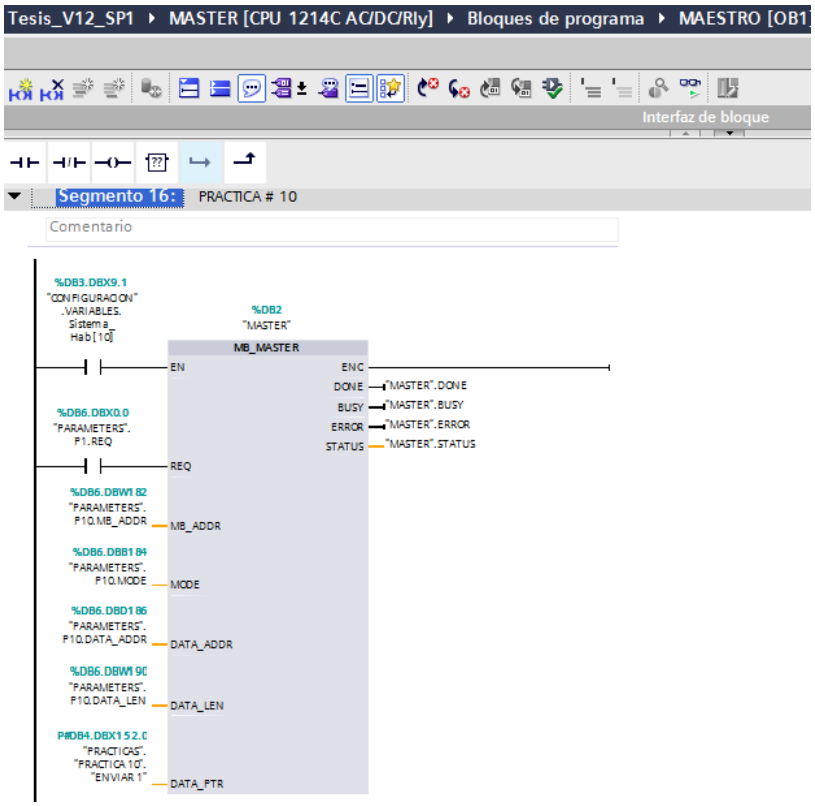


Figura 329: Configuración del bloque escritura MB_MASTER/ PLC MASTER práctica 10.

El primer bloque de instrucción está funcionando en modo escritura, el cual enviará los datos al slave como activación de la bomba y electroválvula. Debemos agregar un

segundo bloque master el cual va a leer el sensor ultrasónico para detección de nivel de agua en los tanques.

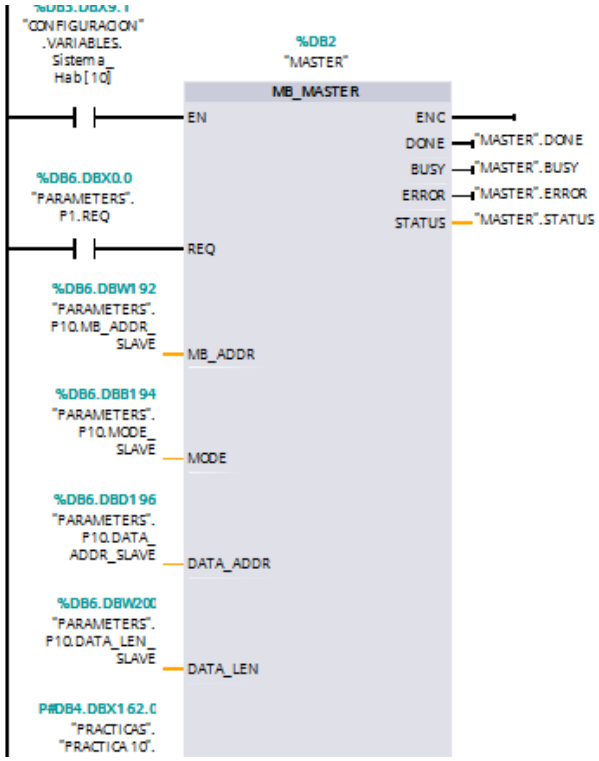


Figura 330: Configuración del bloque lectura MB_MASTER / PLC MASTER práctica 10.

Paso 5.

Ahora deberá agregar el bloque **Cyclic interrupt [OB30]** en el **PLC_2**, que servirá para crear el bloque **PID**.

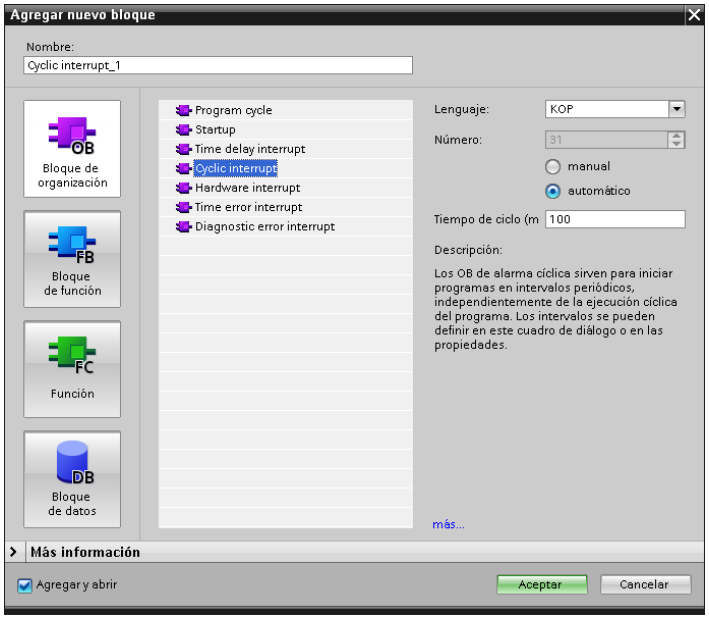


Figura 331: Bloque de organización para PID.

Ahora se deberá programar el bloque **Cyclic interrupt [OB30]**, comenzando con el **Segmento 1**, que es en donde se debe crear el bloque **PID** para el sensor de nivel y configurarlo de la siguiente manera:

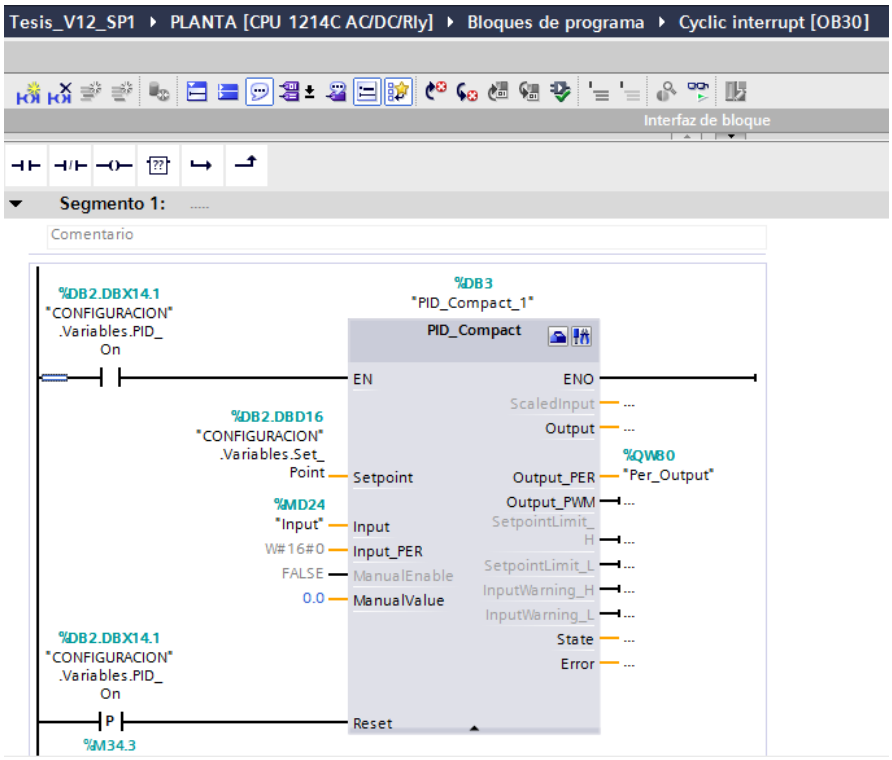


Figura 332: Programación del Segmento 1 PID Compact.

Paso 6.

Se procede a guardar la escala en la variable.

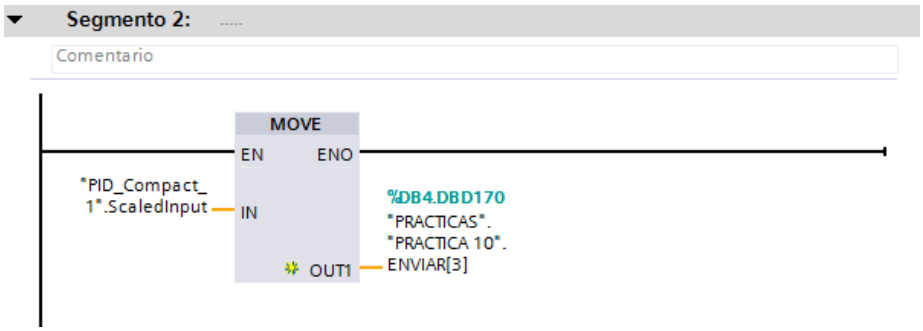


Figura 333: Programación de Segmento 2.

Paso 7.

Los valores de Td, Ti, y Kp se deben registrar para el correcto funcionamiento del PID es por esto que generamos variables donde se guardaran estos valores.

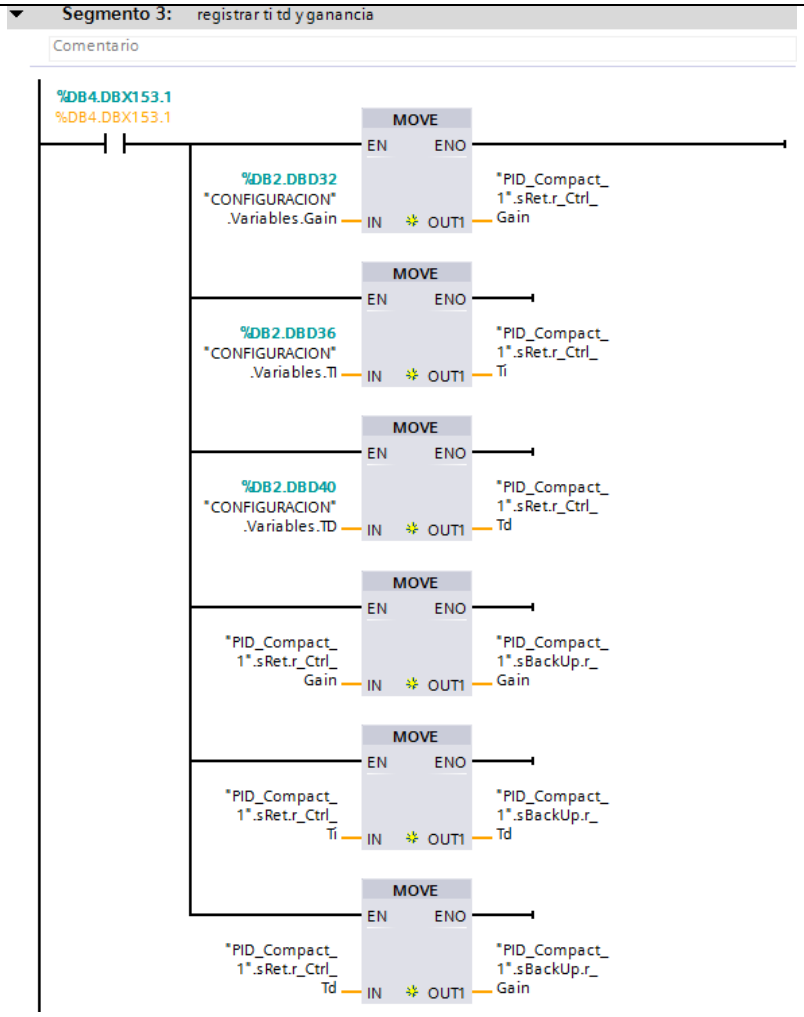


Figura 334: Registro de Ti, Td, Kp.

Paso 8.

A continuación el circuito de control.

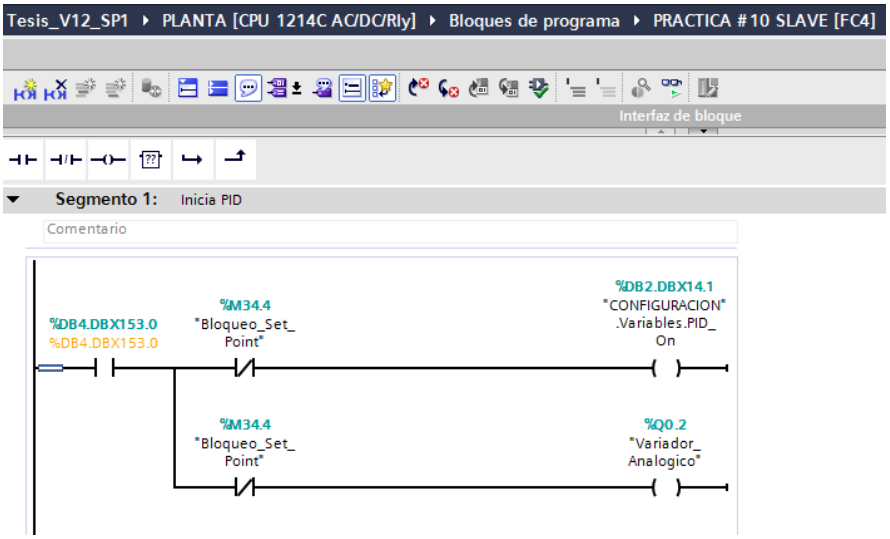


Figura 335: Inicio de PID.

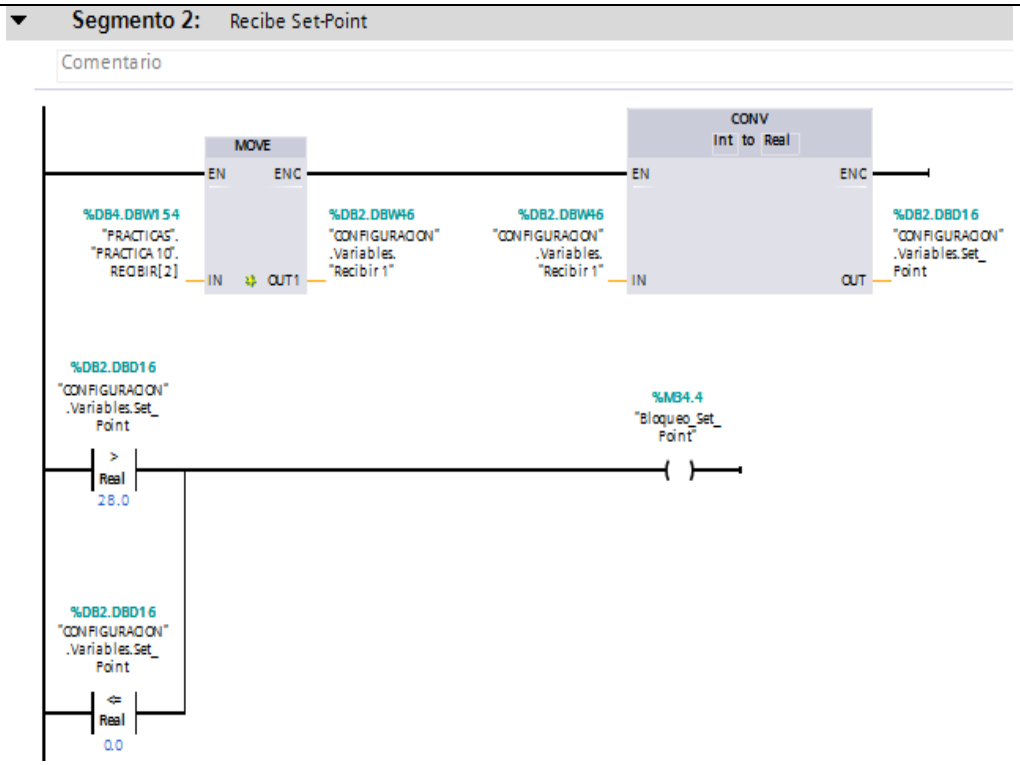


Figura 336: Recepción y bloqueo de setpoint.

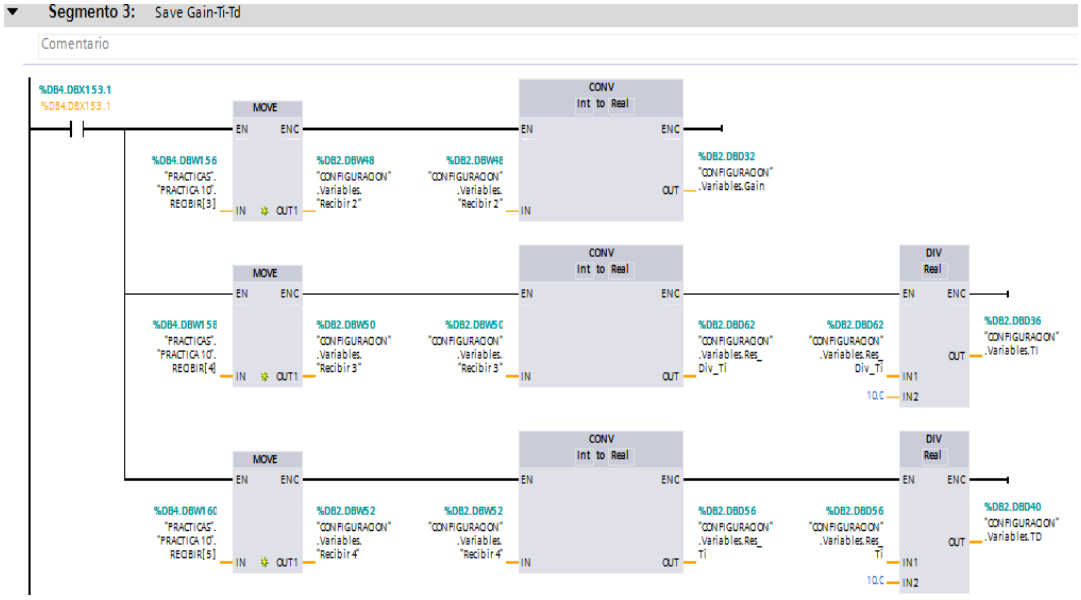


Figura 337: Registro de parámetros PID.

Paso 9.

Ahora se deberá programar el circuito de control para el PLC MASTER / SLAVE.

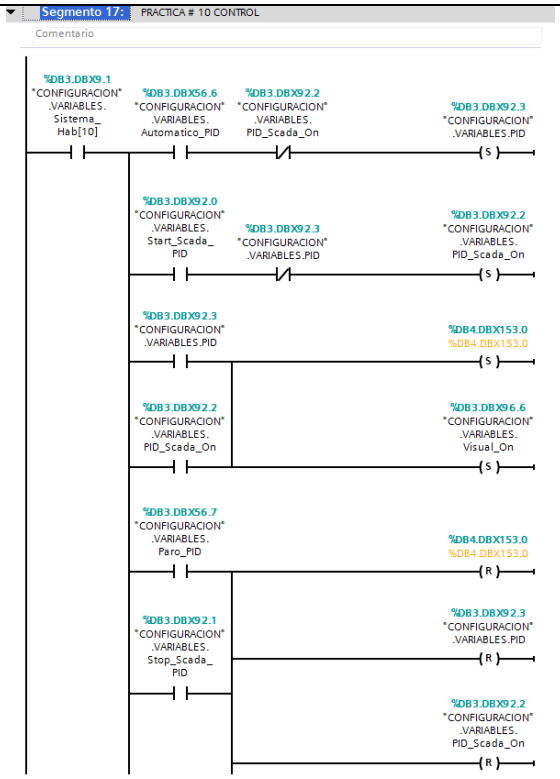


Figura 338: Controles de bloqueo y desbloqueo para HMI y SCADA.

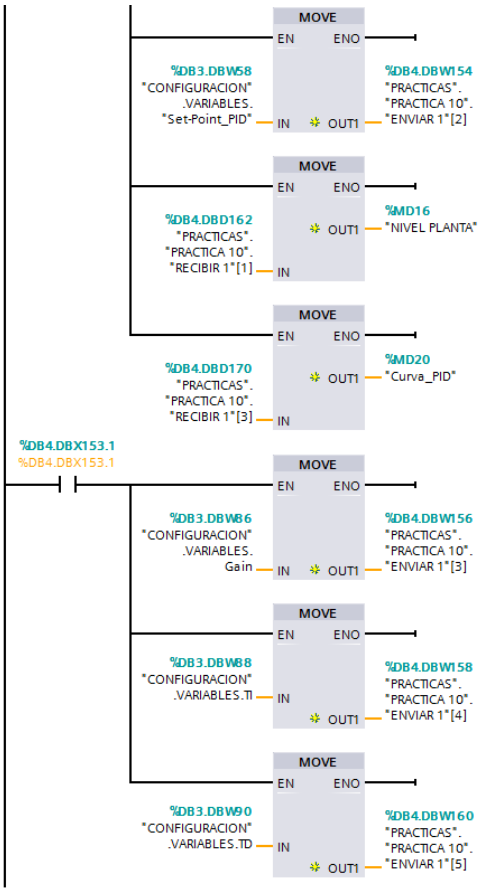


Figura 339: Registro de datos en variables del PLC MASTER.

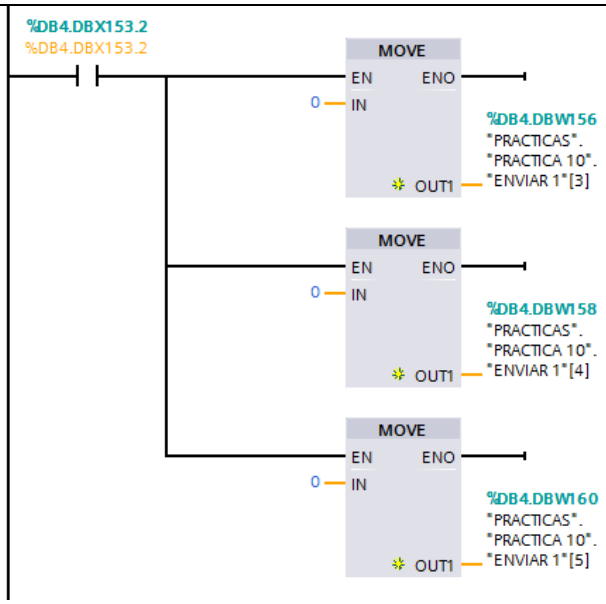


Figura 340: Borrado de registros PID.

Paso 10.

Daremos paso al diseño de las imágenes de la práctica 10.

Figura 341: Diseño de imagen Titulo 10.

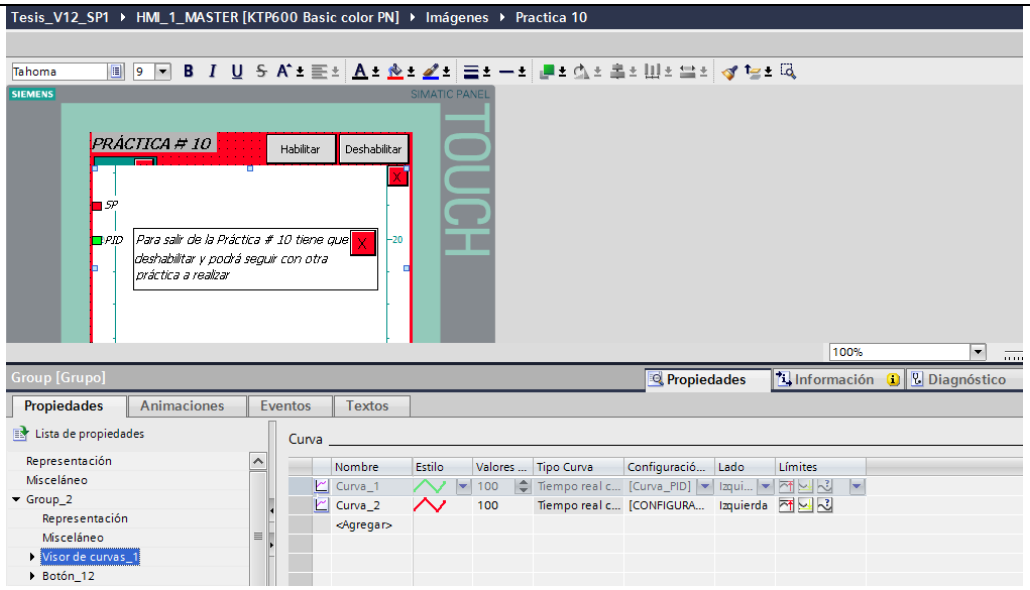


Figura 342: Creación de gráfico de la curva PID.

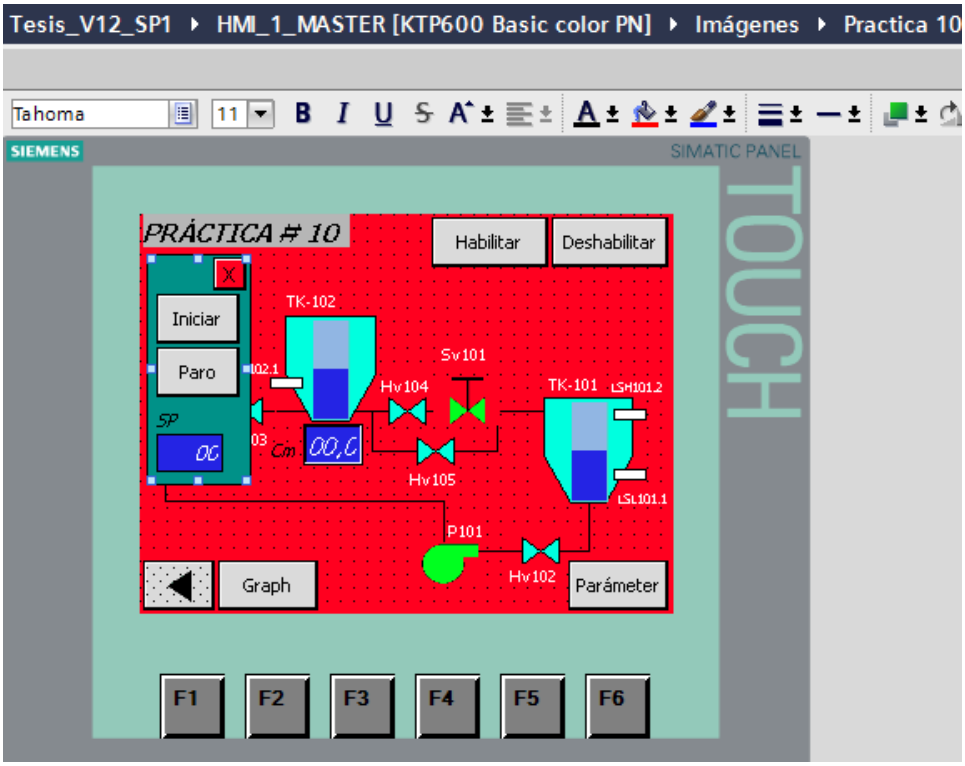


Figura 343: Diseño de imagen Practica 10.

Paso 11.

En este paso vamos a evaluar la obtención de las gráficas del PID, para esto se analizará el comportamiento del control PID mediante el método de sintonización Ziegler - Nichols aplicado para control de lazo abierto.

Procedemos a generar un cambio escalón (9 v.) en la entrada del sistema, seteando el nivel del tanque TK2 a 25 litros.

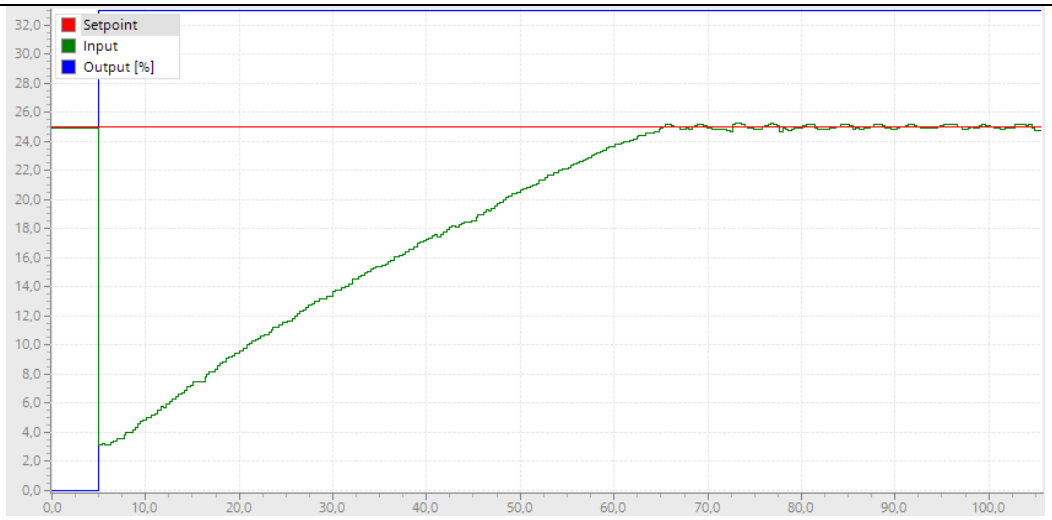


Figura 344: Curva a lazo abierto, Ziegler - Nichols.

Luego vamos a trazar una línea tangente al momento en que se realiza el despegue de la gráfica y realizamos un acercamiento.

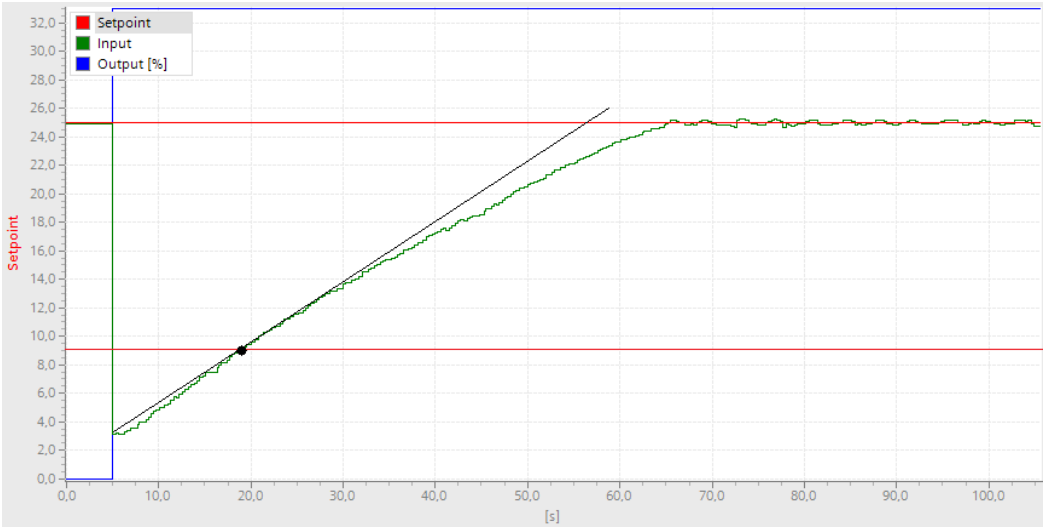


Figura 345: Recta tangente para puntos de inflexión.

Una vez obtenida la gráfica aplicamos la siguiente fórmula:

$$Ko = \frac{(dx \times T2)}{(dy \times T1)}$$

El primer paso lo obtenemos sacando T1 el cual es el tiempo muerto, es decir, el tiempo en que la gráfica empieza a subir hasta el punto de corte de la tangente con el valor inicial del tanque. O a su vez, el tiempo que el sistema se tarda en responder. En nuestro caso experimentalmente medimos T1=14 s.

El tiempo T2 es el tiempo de subida, y se lo mide desde que comienza a subir la señal desde el corte con la tangente hasta que llegue a estabilizarse en el set point. El

tiempo medido fue de T2= 66 – 14 = 52 s.
Dx es la señal de entrada suministrada, en nuestro caso 9 v. Dy es la distancia que hay desde el punto de partida hasta el set point, 25 cm – 3 cm = 22 cm.

Una vez obtenido los datos procedemos a reemplazarlos en la siguiente fórmula:

	Kp	Ti	Td
P	Ko		
PI	0.9 (Ko)	3.3 (T1)	
PD	1.2(Ko)	2 (T1)	0.5 (T1)

$$Ko = \frac{(dx \times T2)}{(dy \times T1)} = \frac{(9 \times 52)}{(22 \times 14)}$$

$$Ko = 1.5194$$

PID			
Kp	1.2 (Ko)	1.2 (1.5194)	**1.8232**
Ti	2 (T1)	2 (14)	**28 s**
Td	0.5 (T1)	0.5 (14)	**7 s**

Se procede a realizar comparaciones cambiando los valores de Kp, Ti y Td para verificar el comportamiento de la gráfica. Se obtuvieron los siguientes resultados:

Respecto a la comparación proporcional - derivativa PD, pudimos observar que en 50 segundos el sistema rebasó al set point pero se generó un pico alto, el cual se estabiliza en el segundo 89 (tiempo de subida), además se generó un overshoot (sobre elongación) de 7 cm. En conclusión este tipo de regulación no es apropiado para nuestro sistema.

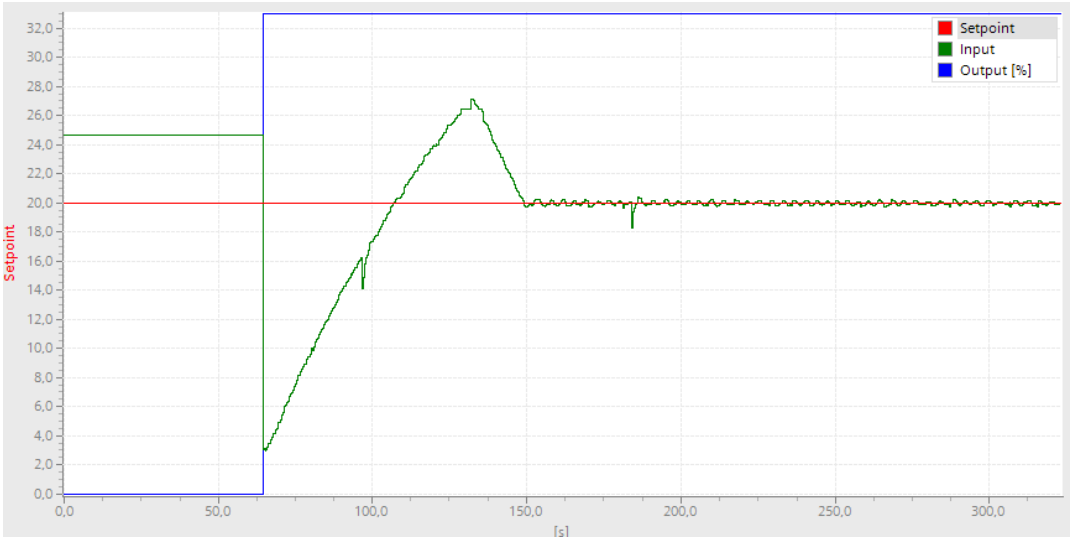


Figura 346: Comportamiento de grafica PD.

Ahora al modificar el tiempo integrativo logramos obtener la gráfica proporcional – integrativa PI, la cual con un set point de 25 cm. Obtenemos una respuesta en 60 segundos, observamos que no existen picos altos o bajos y se elimina el error en régimen permanente.

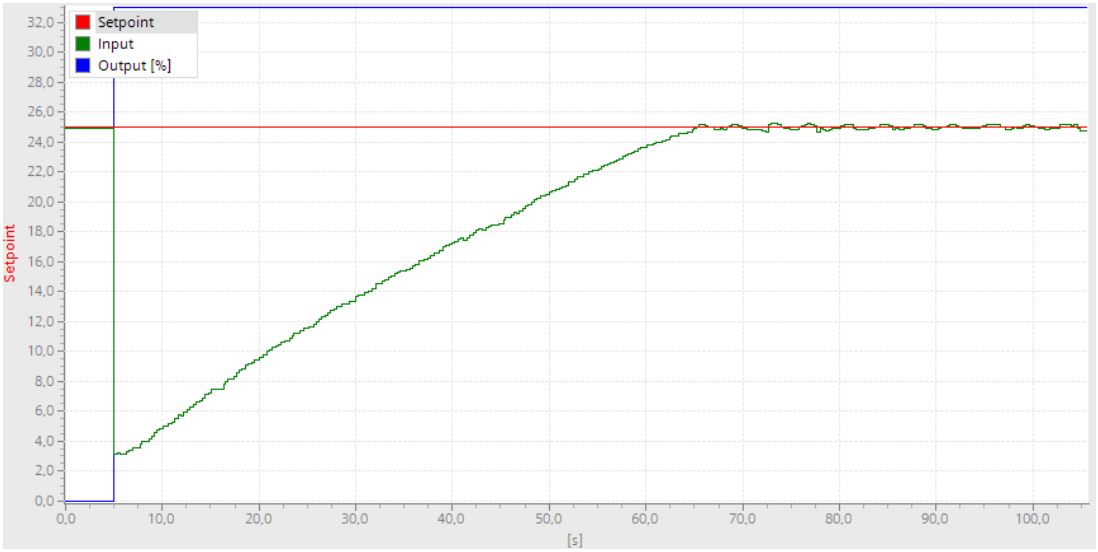


Figura 347: Comportamiento de grafica PI.

Finalmente logramos obtener la gráfica PID (proporcional – integral – derivativa) donde observamos que se corrigen las desviaciones y errores vistos en las anteriores gráficas, es decir hemos logrado un sistema ideal sin sobre elongación, con rápido tiempo de subida y sin error de estado estacionario.

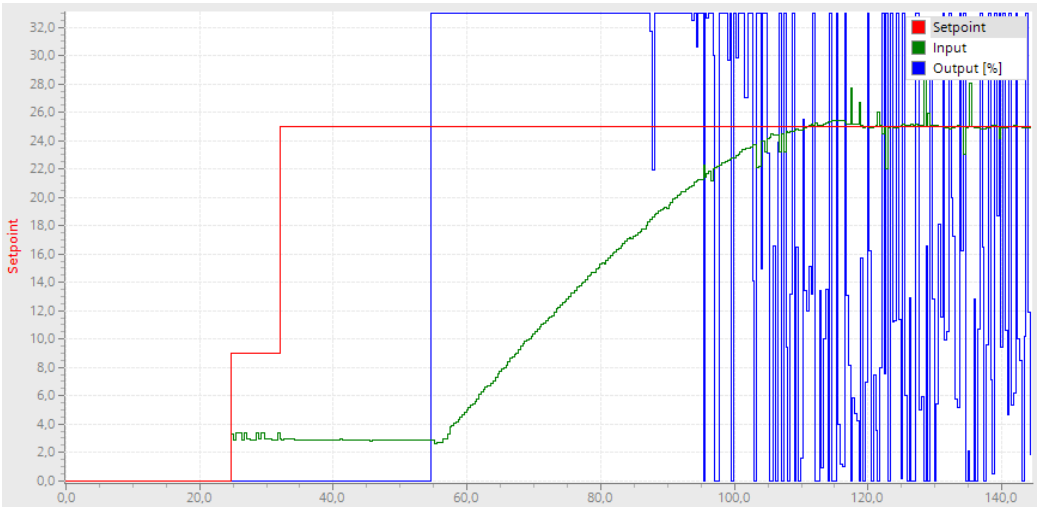


Figura 348: Comportamiento de gráfica PID.

Una vez evaluado y comparado los diferentes tipos de sintonizaciones, daremos paso a un cuadro con los resultados finales:

Tabla 11: Resultados de tiempos de gráficas.

RESPUESTA	RISE TIME	OVERSHOOT	SETTLING TIME	ERROR DE EST. EST:
Pid	2 s :Disminuye	0.4 cm: Disminuye	5 s: Disminuye	2 %: Disminuye
Pi	4 s: Disminuye	0.1 cm: Disminuye	10 s: Aumenta	0 %: Elimina
Pd	0 :Poco Influencia	7 cm: Aumenta	39 s:Aumenta	1 %: Poco Influencia

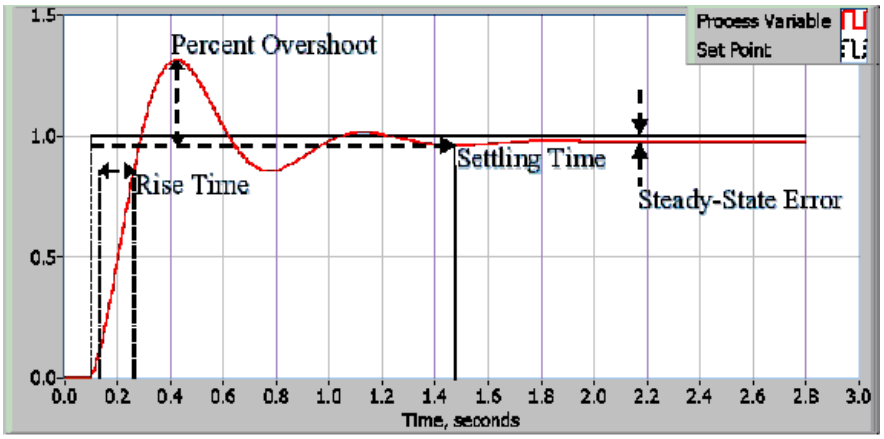


Figura 349: Tiempos de gráfica PID.

Se concluye que al modificar los parámetros integrales, proporcionales o derivativos se obtienen repuestas más rápidas pero con pérdidas o interrupciones. Se debe adecuar el sistema a la función que más se apegue al rendimiento. En nuestro caso el sistema ideal para el control de nivel de agua es el PID.

Recomendaciones.

- Comprobar que las variables estén identificadas y con nombre para cada parte del proceso.
- Revisar que las variables de los controles del SCADA y HMI estén correctamente asignadas.
- Verificar que la curva PID se encuentre lo más alineada posible.

5. RESULTADOS

El resultado principal de este proyecto de titulación se basó en suministrar dos módulos didácticos para el laboratorio de automatización industrial de la Universidad Politécnica Salesiana sede Guayaquil, los cuales incorporan comunicación serie de tipo MODBUS RTU a través del protocolo de comunicación serie RS485. Los detalles del diseño e implementación del proyecto se detallan a continuación:

- Se efectuó la migración de 2 PLC's de los módulos 7 y 8 del laboratorio de automatización industrial a las nuevas estructuras con diseño simplificado y agradable, mejorando la interacción y manejo de los estudiantes al manipular los diferentes elementos de las prácticas a desarrollar.
- Se integró nuevos elementos de control para ambos módulos, lo que incluyeron relés, botoneras, interruptores, potenciómetros de precisión, voltímetros analógicos, breakers, luces piloto y borneras.
- Se habilitaron los puertos de comunicación RS485 de los módulos de comunicación CM1241 de cada módulo didáctico, los cuales no habían sido utilizados anteriormente por los estudiantes, y que tendrán el objetivo de brindar mayores opciones de comunicación de tipo industrial al estudiante en el momento de desarrollar un proyecto técnico.
- Se configuró adecuadamente los PLC's para establecer la comunicación MODBUS RTU de acuerdo a su funcionamiento maestro / esclavo.
- Se agregaron los puertos de comunicación RS485 a los módulos didácticos, a la Planta de Control de Nivel y al módulo de variador de frecuencia ABB.
- Se configuró el control de un motor trifásico de marca ABB con un controlador de marca Siemens por medio de comunicación MODBUS RTU.
- Se implementó un controlador PID a una Planta de Control de Nivel vía MODBUS RTU, obteniendo excelentes resultados semejantes a los utilizados en el sector industrial.
- Se ejecutaron 10 prácticas de laboratorio con su respectiva programación realizada con el software TIA PORTAL V12.

5.1. ANÁLISIS DE RESULTADOS

Se propuso exponer el presente proyecto técnico de titulación a los estudiantes de la carrera de Ingeniería Electrónica que estén cursando los últimos ciclos, con la finalidad de dar a conocer las ventajas y limitaciones del protocolo de comunicación y además obtener información referente al grado de dificultad del proyecto, así mismo como la utilidad y aplicación que se daría en ellos. Para esto se elaboró dos exposiciones en dos seminarios diferentes, uno de Automatización Industrial con PLC Siemens S7-1200, y el otro de Variadores de Frecuencia, con el fin de presentar dos prácticas desarrolladas en nuestro proyecto técnico. En el seminario de variadores se presentó la práctica # 5 y # 7 que llevan relación con el variador de frecuencia ABB. Finalmente en el seminario de S7-1200 se presentó la práctica # 10 donde se demostró el control PID de la planta de control de nivel. La muestra total fue de 43 estudiantes y se ejecutaron durante el periodo de vacaciones del año 2018.

Figura 350: Exposición en seminario de S7-1200.

Figura 351: Presentación de diapositivas en seminario de S7-1200.

Figura 352: Exposición en seminario de Variadores.

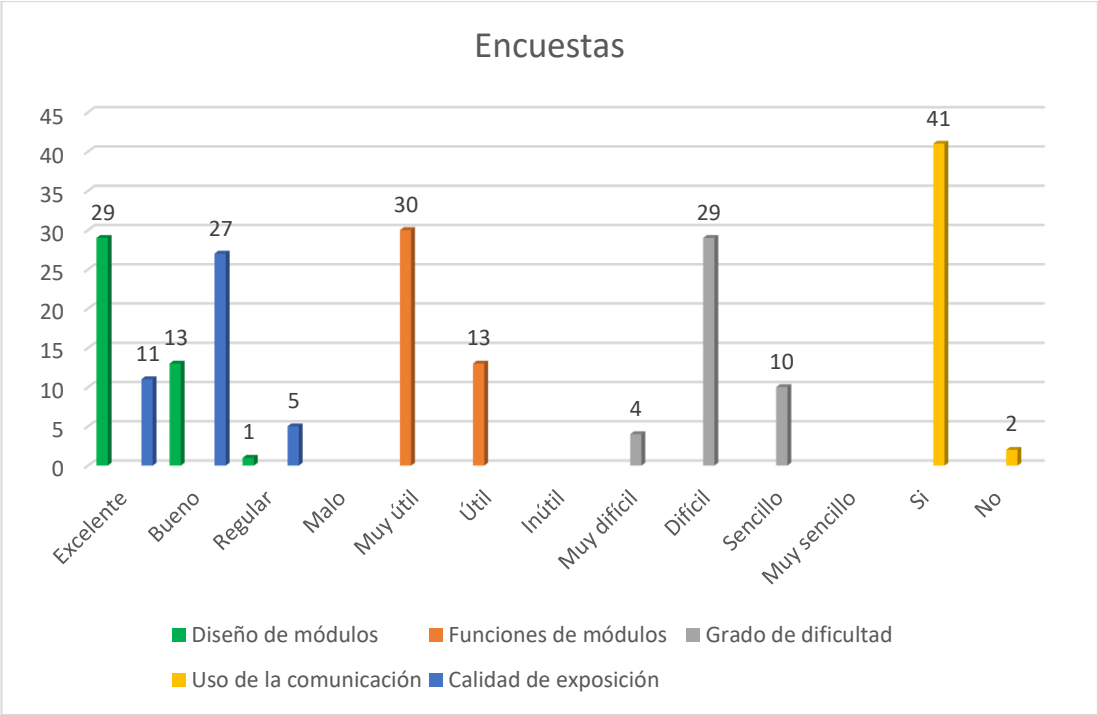
Figura 353: Presentación de diapositivas en seminario de Variadores.

Luego de haber presentado las exposiciones y haber despejado las inquietudes de los estudiantes se elaboró una breve encuesta para medir el impacto del proyecto entre otros recursos.

• **¿Cómo encuentra usted el diseño de los módulos didácticos?**
Excelente Bueno Regular Malo
• **¿Qué tan útil le parece las diferentes funciones que posee los módulos didácticos?**
Muy útil Útil Inútil

• **¿En qué grado de dificultad definiría usted la aplicación de la comunicación MODBUS RTU?**
Muy difícil Difícil Sencillo Muy sencillo
• **¿Estaría dispuesto a probar este método de comunicación en sus futuros proyectos?**
Si No
• **En cuanto a la exposición, ¿cómo la encontró?**
Excelente Buena Regular Mala
• **En su opinión ¿que se debería mejorar en la exposición y en el proyecto?**
…… …… …… …… ……

Posterior a las clases demostrativas y el desarrollo de las encuestas pudimos obtener el grado de satisfacción de los estudiantes respecto al proyecto, también se logró evidenciar el interés para la aplicación de lo aprendido en el ámbito estudiantil y laboral. Con esto se puede concluir que los temas explicados fueron de gran importancia para ampliar y complementar los conocimientos de los estudiantes y fomentar la creación de nuevos proyectos para la carrera y en su vida profesional.


Encuestas
45
40
35
30
25
20
15
10
5
0
29
11
13
27
1
5
30
13
4
29
10
41
2
Excelente
Bueno
Regular
Malo
Muy útil
Útil
Inútil
Muy difícil
Difícil
Sencillo
Muy sencillo
Si
No
Diseño de módulos
Funciones de módulos
Grado de dificultad
Uso de la comunicación
Calidad de exposición

6. CONCLUSIONES

Al concluir el presente trabajo de titulación se puede evidenciar que se han empleado los conocimientos adquiridos a lo largo de la carrera universitaria, los cuales han sido pieza clave para el desarrollo de las prácticas, para la programación y aplicación de destrezas manuales.

Este proyecto ha logrado cumplir con los objetivos propuestos, especialmente demostrar que la comunicación es posible implementarla hasta con 32 dispositivos diferentes a través de los puertos y configuraciones diseñadas.

Los módulos entregados podrán ser utilizados para futuros proyectos estudiantiles y de titulación los cuales requieran este tipo de comunicación en particular u otros, así mismo los módulos aportarán al laboratorio con un tiempo de vida útil más prolongado y con la capacidad de conectarse con todos los módulos del laboratorio, ya que los módulos antiguos y nuevos incorporan el puerto de comunicación CM1241. A su vez se aprueba la experimentación con actuadores, motores y demás.

Los docentes podrán impartir con la enseñanza de este protocolo de comunicación a los nuevos estudiantes, además de combinar los diferentes tipos de redes de comunicación industrial como Profinet, Profibus y ASi para la elaboración de prácticas de laboratorio.

7. RECOMENDACIONES

Se recomienda incluir en las clases de automatización I y II los principios de fundamentos de comunicación serie a nivel industrial, ya que son ampliamente utilizados en el campo laboral. Así mismo la elaboración de prácticas de laboratorio para que los estudiantes se familiaricen con la configuración MODBUS RTU.

Es importante recalcar el cuidado y seguimiento de los módulos entregados, para salvaguardar la integridad de los mismos y tengan un tiempo de vida útil de mínimo 5 años de duración, con mantenimientos preventivos que consistan en limpieza de contactos y elementos varios del tablero.

8. BIBLIOGRAFÍA

[1] J. M. H. Torres, «infoPLC,» [En línea]. Available: http://www.infoplc.net/files/documentacion/comunicaciones/infoPLC_net_introduccic3b3n-a-las-redes-de-comunicacic3b3n-industrial.pdf. [Último acceso: 18 octubre 2017].

[2] DRAGO AUTOMATION, «DRAGO Signal Converters,» 23 01 2018. [En línea]. Available: https://www.drago-automation.de/frequently-asked-questions/items/rs485-network-for-modbus.html. [Último acceso: 21 02 2018].

[3] B. Drury, Control Techniques Drives and Controls Handbook, Londres: Emerson, 2009.

[4] F. Candelas, «rua,» 2011. [En línea]. Available: https://rua.ua.es/dspace/bitstream/10045/18990/1/AA-p3.pdf. [Último acceso: 12 Febrero 2018].

[5] J. Bartolomé, «El Protocolo MODBUS,» Enero 2011. [En línea]. Available: http://www.tolaemon.com/docs/modbus.htm#func_1_2. [Último acceso: 12 Febrero 2018].

[6] R. L. Y. L. M. Vicente Guerrero, Comunicaciones industriales, Mexico D.F.: Marcombo, 2009.

[7] J. L. R. Josep Balcells, Autómatas programables, Barcelona: Marcombo, 1997.

[8] Siemens, «Industry Online Support,» 10 Junio 2017. [En línea]. Available: https://support.industry.siemens.com/cs/document/109736665/%C2%BFc%C3%B3mo-se-conectan-las-interfaces-rs485-rs422-de-los-m%C3%B3dulos-simatic-y-siplus-para-la-comunicaci%C3%B3n-serie-?dti=0&lc=es-CO. [Último acceso: 12 Febrero 2018].

[9] M. Prada, «Monografias.com,» 06 2013. [En línea]. Available: http://www.monografias.com/trabajos91/motor-electrico-trifasico/motor-electrico-trifasico.shtml. [Último acceso: 13 Febrero 2018].

[10] ABB, «Variadores de Frecuencia,» ABB, Enero 2018. [En línea]. Available: http://new.abb.com/drives/es/que-es-un-variador. [Último acceso: 13 Febrero 2018].

[11] C. Moscoso y J. Portugal, de *Diseño e implementación de red wireless entre dos autómatas programables S7 1200 para aplicación de control y automatización.*, Guayaquil, Universidad Politécnica Salesiana, 2017, pp. 32-33.

[12] Siemens AG, «¿Cuáles son las diferencias entre los CPs de Industrial Ethernet del S7-300?,» 13 05 2013. [En línea]. Available: https://support.industry.siemens.com/cs/document/16767769/%C2%BFcu%C3%A1les-son-las-diferencias-entre-los-cps-de-industrial-ethernet-del-s7-300-?dti=0&lc=es-WW. [Último acceso: 03 07 2017].

[13] Manhattan, «Convertidor USB a RS485,» IC Intracom, 1 Enero 2017. [En línea]. Available: http://www.manhattan-products.com/usb-to-rs485-converter. [Último acceso: 7 Enero 2018].

[14] Unicrom, «Electrónica Unicrom,» 02 Enero 2016. [En línea]. Available: https://unicrom.com/fuente-de-voltaje-variable-con-lm317t/. [Último acceso: 13 Febrero 2018].

[15] Siemens, «Software TIA Portal,» Siemens, 02 03 2016. [En línea]. Available: http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Tia-Portal/Pages/TIAPortal.aspx. [Último acceso: Domingo Enero 2018].

[16] Siemens, «Plataforma de ingeniería,» Siemens Spain, 02 03 2016. [En línea]. Available: http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Tia-Portal/tia_portal/Pages/TIA-Portal.aspx. [Último acceso: 7 Enero 2018].

[17] MODBUS TOOLS, «ModBus Tools,» [En línea]. Available: http://www.modbustools.com/modbus_poll.html. [Último acceso: 4 Marzo 2018].

[18] ABB machinery drives, «Manual del Usuario Convertidores de Frecuencia ACS355,» 30 Abril 2015. [En línea]. Available: https://library.e.abb.com/public/23b89b1744024211a5600e69cad5ea78/ES_ACS355_UM_C_with_note.pdf. [Último acceso: 22 Enero 2018].

[19] Modbus Organization, «EEYMUC,» EEYMUC, 01 01 2018. [En línea]. Available: https://www.eeymuc.co/31-protocolo-modbus/. [Último acceso: 13 02 2018].

ANEXOS

ANEXO 1.

Driver de motor de la bomba DC de Planta de Control de Nivel.

KALEJA GmbH
D-73553 Alfdorf

Motorcontrol for brush sticking direct current motor 24VDC

Implementation for switching current up to 4A

Suitable for acceleration-rate and deceleration-rate Adjustable time of starting ramp

To snap onto DIN - rail EN 50022 and EN 50035

Construction width: 22,5mm

Short designation / type	Rated voltage: 24VDC Maxi-S-4-30
Art. - No.	06.04.007
Technical data: input circuit	
Rated voltage / threshold voltage	24 VDC
Range of rated voltage min. / max.	19V to 35VDC
Input current during rated voltage	10mA
Analogue input - range of voltage	0V to 10VDC
Status indicator	LED 3mm yellow
Technical data: output circuit	MOS-FET
Range of switching voltage / motor voltage	19V to 35VDC
Max. permanent load current	4A
min. / max. time of starting ramp	0 - 1000ms adjustable
Forward resistance at ON-state	20 mOhm
Power driver	MOS-FET
Other data	
Ambient temperature range	-20°C to + 50°C
Absence of vibration a/r (10...500Hz)	> 20 / 5
Overload protection / short-circuit-proof / temperature monitoring	yes / yes / yes
DIN VDE-determinations	VDE 0110, 0160 in parts
Position of installation / mounting	can be snapped, addable
Mode of connection: screw terminal	single wire 4mm², fine wire 2,5mm²
Dimensions: W x D x H	22,5mm x 75mm x 102mm

05.02.2018 AT - Errors and technical modification subject to change. Reproduction as well as electronic duplication only with our written permission. All rights reseved. © KALEJA GmbH 2015

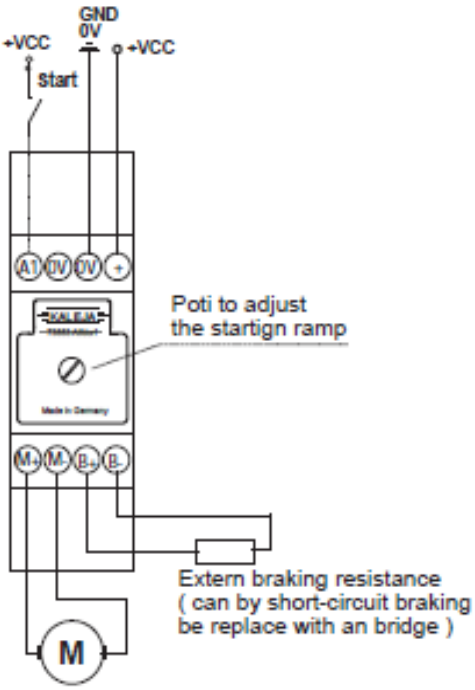
Description

The Maxi-S-4-30 module is a motor control system with acceleration-rate/deceleration-rate for 24VDC motors. It ensures switching ON/OFF ant the definite driving and braking of motors, solenoid valves and other loads. The load is short-circuited over an extern resistance in OFF conditions wich result in dynamical braking, dependent from extern resistance value.

Special features:

Short-circuit protection against GND, temperature protection, specialize for the use at inductances, adjustable time until the motor reach its max. speed.

Electrical connection and controls



ANEXO 2.

Características del módulo de comunicación MODBUS, CM1241.

SIEMENS

Product data sheet	6ES7241-1CH30-0XB0

SIMATIC S7-1200, COMMUNICATION COMMUNICATION MODULE CM 1241, RS485, 9 PIN SUB D (MALE), SUPPORTS MESSAGE BASED FREEPORT

Supply voltages	
Rated value	
DC 24 V	Yes
permissible range, lower limit (DC)	20.4 V
permissible range, upper limit (DC)	28.8 V
Short-circuit protection	0.0000000000000000E+00
Current consumption	
Current consumption, max.	220 mA ; from L5+; logic
Current consumption/power loss	
Power loss, typ.	1.1 W
interfaces	
Number of interfaces	1
interface physics, RS 422/RS 485 (X.27)	Yes
Point-to-point	
Cable length, max.	1000 m
Integrated protocol driver	
ASCII	Yes ; available as library function
USS	Yes ; available as library function
Dimensions and weight	
Dimensions and weight	
Width	30 mm
Height	100 mm
Depth	75 mm
Weight	
Weight, approx.	150 g
Status	Sep 28, 2009

subject to modifications
© Copyright Siemens AG 2009

ANEXO 3.

Características del módulo de comunicación MODBUS, FMBA-01.

ABB Drives

User's Manual
Modbus Adapter Module
FMBA-01

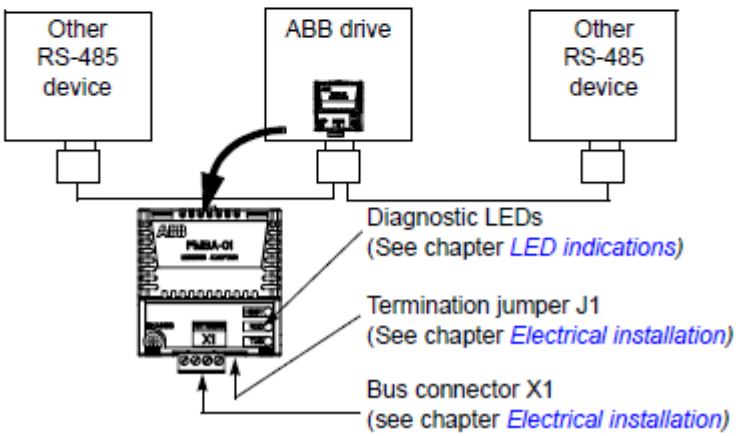
The FMBA-01 Modbus Adapter module

The FMBA-01 Modbus Adapter module is an optional device for ABB drives which enables the connection of the drive to a RS-485 network. The FMBA-01 provides galvanic isolation between the drive and RS-485 network and converts serial communication signals of the drive to RS-485 signal levels.

The FMBA-01 has a built-in bus termination with fail-safe circuitry. The termination can be activated with a jumper.

The adapter module is mounted into an option slot on the motor control board of the drive. See the drive documentation for module placement options.

Construction of the RS-485 network and the FMBA-01



Electrical installation

Overview

This chapter contains:

- general cabling instructions
- instructions for setting the bus termination
- instructions for connecting the module to the RS-485 network.

WARNING! Before installation, switch off the drive power supply. Wait five minutes to ensure that the capacitor bank of the drive is discharged. Switch off all dangerous voltages connected from external control circuits to the inputs and outputs of the drive.

General cabling instructions

Arrange the bus cables as far away from the motor cables as possible. Avoid parallel runs. Use bushings at cable entries.

RS-485 connection

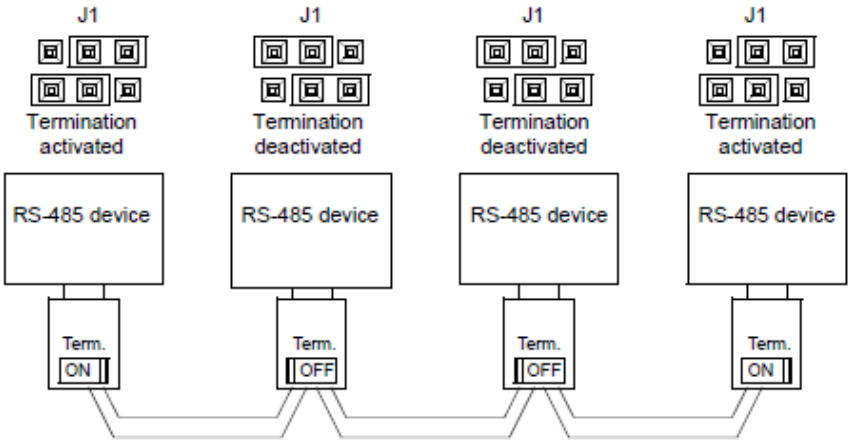
RS-485 cable is connected to connector X1 on the FMBA-01 Modbus Adapter module.

X1			Description
1 2 3 4	1	SHLD	Bus cable shield. Connected internally to GND_B via an RC filter and directly to CH_GND (chassis).
	2	DATA_B	Data positive
	3	DATA_A	Data negative
	4	GND_B	Isolated signal ground

Bus termination

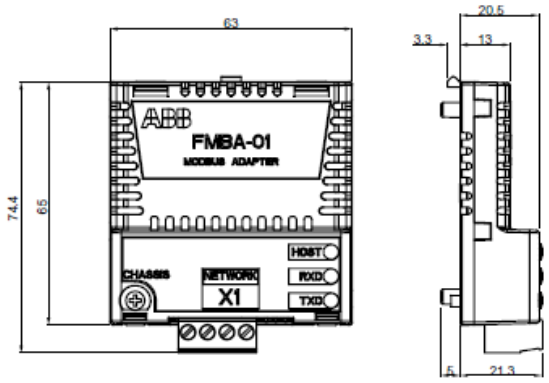
Bus termination is required to prevent signal reflections from the bus cable ends. The FMBA-01 Modbus Adapter module is equipped with internal bus termination, which is configurable with jumper J1 pins. Termination should be activated on devices located at bus ends and deactivated on other devices. See the figure below.

RS-485 bus termination



FMBA-01 Modbus Adapter module

Enclosure:



Mounting: Into the option slot on the drive

Degree of protection: IP20

Ambient conditions: The applicable ambient conditions specified for the drive in its manuals are in effect.

Indicators: Three LEDs (HOST, RXD, TXD)

Connectors:

- 20-pin connector to drive (X2)
- 4-pin detachable screw connector to bus (X1)
- 6-pin jumper block for bus termination configuration (J1)

Power supply:

- +3.3 V ±5% max. 150 mA (supplied by the drive)

General:

- Estimated min. lifetime: 100 000 h
- All materials UL/CSA-approved
- Complies with EMC standards EN 50081-2 and EN 50082-2
- Bus interface functionally isolated from drive

RS-485 link

Compatible devices: All RS-485 compliant devices

Medium: Shielded twisted pair cable, impedance 100 to 150 ohm

- Termination: 120 ohms (built in the FMBA-01 Modbus Adapter module)

Topology: Trunk line, drop lines allowed

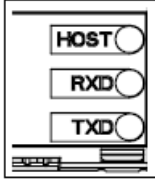
Transfer rate: 5 Mb/s max.

Serial communication type: Asynchronous, half-duplex RS-485

Protocol: Depends on the used application. Typically Modbus.

LED indications

The FMBA-01 Modbus Adapter module is equipped with three green diagnostic LEDs. The LEDs are described below.

	Name	Colour	Function
	HOST	Green	The module is powered up.
	RXD	Blinking	The module receives data from the bus. The LED blinks whether the received data is address to the unit or not.
	TXD	Blinking	The module sends data to the bus.