



**UNIVERSIDAD DEL AZUAY**  
**FACULTAD DE CIENCIA Y TECNOLOGÍA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**

**Implementación de los protocolos de comunicación industrial mediante  
Simatic S7-1200 para el laboratorio de automatización de la  
Universidad del Azuay**

**Trabajo de graduación previo a la obtención del título de:**

**INGENIERO ELECTRÓNICO**

**Autor:**

**DIEGO JAVIER SARMIENTO SÁNCHEZ**

**Director:**

**FRANCISCO EUGENIO VÁSQUEZ CALERO**

**CUENCA, ECUADOR**

**2016**

## **DEDICATORIA**

A lo largo de mi vida he podido alcanzar diferentes metas y como cada una de ellas, dedico con profundo cariño este trabajo de titulación a las personas más importantes en mi vida, mi familia.

A mi padre Manuel Sarmiento y mi madre Elizabeth Sánchez que con su esfuerzo han sabido guiarme en todo momento y cuyas enseñanzas de vida estoy seguro que jamás olvidare, además gracias a las cuales he podido enfrentar cada uno de los desafíos que se me han presentado. A mis hermanas Anabel y Lisseth con quienes he compartido muchas ideas, pensamientos, metas, de quienes me siento orgulloso y estoy convencido que llegarán muy lejos.

## **AGRADECIMIENTOS**

Al culminar con mis estudios de pregrado, expreso mis más sinceros agradecimientos a todos quienes han sido parte de este proceso y han hecho posible mi desarrollo académico y personal.

Primero agradezco a Dios, por la vida, por mis fuerzas y por la inteligencia que me ha brindado, también agradezco a mi familia por su apoyo incondicional en todo momento, por sus palabras de aliento y compañía que sin duda me animaron a seguir adelante.

Agradezco a mi director de tesis, Ing. Francisco Vásquez por su dedicación y apoyo durante la realización de este trabajo, también a los docentes de la Universidad y al Ing. Hugo Torres director de Escuela de Ingeniería Electrónica.

De manera muy especial deseo agradecer también a la empresa CENELSUR dirigida por el Ing. Manuel Sarmiento y Fabián Vélez por el apoyo brindado al facilitarme equipos y documentación relevantes para este proyecto, también al Ing. Christian Beltrán encargado de la sección de proyectos.

Finalmente agradezco a Paulina, por ser siempre un pilar en mi vida, por su preocupación, ayuda y aliento.

¡Gracias a Todos!

## ÍNDICE DE CONTENIDOS

CONTENIDOS	Pag.
<b>DEDICATORIA .....</b>	<b>ii</b>
<b>AGRADECIMIENTOS .....</b>	<b>iii</b>
<b>ÍNDICE DE CONTENIDOS .....</b>	<b>iv</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>ix</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>xxii</b>
<b>ÍNDICE DE ANEXOS .....</b>	<b>xxiii</b>
<b>RESUMEN .....</b>	<b>xxiv</b>
<b>ABSTRACT .....</b>	<b>xxv</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1: REDES DE COMUNICACIÓN INDUSTRIAL .....</b>	<b>3</b>
Introducción.....	3
<b>1.1 LAS COMUNICACIONES INDUSTRIALES .....</b>	<b>3</b>
1.1.1 Introducción .....	3
1.1.2 Pirámide de la automatización CIM (Computer Integrated Manufacturing, “Fabricación Integrada por Computador”) .....	6
1.1.3 Sistemas de Control en una Red de Comunicación Industrial .....	10
1.1.4 Topología de las Redes .....	12
1.1.5 Buses de Campo.....	19
1.1.5.1 Clasificación de los Buses de Campo.....	20
1.1.5.2 Principales Buses de Campo .....	22
1.1.5.3 Comparativa .....	25
<b>1.2 PROFIBUS .....</b>	<b>28</b>
1.2.1 Característica Generales.....	30
1.2.1.1 Arquitectura.....	31
1.2.1.2 Control de Acceso al medio .....	32
1.2.1.3 Trama de Datos .....	34

1.2.1.4 Medios Físicos.....	35
1.2.2 Profibus-DP.....	38
1.2.3 Profibus-PA.....	39
1.2.4 Profibus-FMS.....	41
1.3 PROFINET .....	43
1.3.1 Características Generales .....	44
1.3.1.1 Control de Acceso al Medio.....	45
1.3.1.2 Trama de Datos .....	46
1.3.1.3 Medios Físicos.....	46
1.3.1.4 Componentes de Red.....	49
1.3.1.5 Seguridad.....	51
1.3.1.6 Principales Tipos de Comunicación.....	52
1.3.2 Profinet-I/O .....	53
1.3.3 Profinet-CBA .....	55
1.4 MODBUS .....	56
1.4.1 Características Generales .....	58
1.4.1.1 Estructura de Mensaje .....	60
1.4.1.2 Control de Acceso al Medio.....	61
1.4.1.3 Medios Físicos.....	62
1.4.1.4 Códigos de Función.....	63
1.4.2 Modbus-ASCII.....	65
1.4.3 Modbus-RTU .....	66
1.4.4 Modbus-TCP/IP .....	68
1.5 CONCLUSIONES.....	69

## **CAPÍTULO 2: CONFIGURACIÓN DEL PLC SIMATIC S7-1200 PARA LOS SISTEMAS DE AUTOMATIZACIÓN A UTILIZAR.....73**

Introducción.....	73
2.1 SISTEMA DE AUTOMATIZACIÓN REAL.....	73
2.1.1 Elementos del Sistema .....	73
2.1.1.1 Simatic S7-1200.....	73
2.1.1.2 Entradas Analógicas de la CPU .....	77

2.1.1.3 Sensor de Ultrasonido .....	77
2.1.1.4 Bomba Hidráulica .....	79
2.1.2 Diseño y Construcción del Sistema .....	80
2.1.2.1 Diagrama de Instrumentación de la Planta.....	80
2.1.2.2 Esquema de Conexión del Proceso .....	82
2.1.3 Control PID (Fundamentos Teóricos).....	84
2.1.3.1 El controlador PID .....	84
2.1.3.2 Sintonización PID (Método de Ziegler-Nichols) .....	87
2.1.4 Modelo Matemático del Sistema .....	90
2.1.5 Simulación y Sintonización PID Ideal mediante MATLAB .....	92
2.1.6 PID con Simatic S7-1200.....	93
2.1.6.1 Generalidades.....	93
2.1.6.2 PID_COMPACT .....	96
2.1.6.3 Configuración PID_COMPACT .....	97
2.1.6.4 Puesta en Servicio .....	100
2.1.7 Pruebas de Funcionamiento .....	104
2.2 SISTEMA DE AUTOMATIZACIÓN VIRTUAL USANDO <i>FACTORY I/O</i> ....	107
2.2.1 Descripción de <i>FACTORY I/O</i> .....	107
2.2.2 Diseño y Construcción del Sistema .....	108
2.2.3 Control del Sistema.....	111
2.2.3.1 Comunicación con Simatic S7-1200.....	111
2.2.3.2 Estaciones.....	113
2.2.4 Pruebas de Funcionamiento .....	115
2.3 CONCLUSIONES .....	118

### **CAPÍTULO 3: IMPLEMENTACIÓN DE LAS COMUNICACIONES INDUSTRIALES MEDIANTE SIMATIC S7-1200.....121**

Introducción.....	121
3.1 RED PROFINET .....	121
3.1.1 Elementos del Sistema .....	122
3.1.1.1 Interfaz Profinet.....	122
3.1.1.2 Cables y Conectores .....	122

3.1.2 Configuración Red Profinet .....	123
3.1.3 Ejemplo de Conexión.....	128
3.1.4 Pruebas de Funcionamiento .....	132
3.2 RED PROFIBUS .....	135
3.2.1 Elementos del Sistema .....	135
3.2.1.1 Módulo Maestro-DP (CM 1243-5) .....	135
3.2.1.2 Módulo Esclavo-DP (CM 1242-5).....	136
3.2.1.3 Cable y Conectores.....	137
3.2.2 Configuración Red Profibus .....	139
3.2.3 Ejemplo de Conexión.....	141
3.2.4 Pruebas de Funcionamiento .....	144
3.3 RED MODBUS-RTU.....	149
3.3.1 Elementos del Sistema .....	149
3.3.1.1 Módulo RS-485 (CM 1241).....	149
3.3.1.2 Cable y Conectores.....	150
3.3.2 Configuración Red Modbus-RTU.....	151
3.3.3 Ejemplo de Conexión.....	155
3.3.4 Pruebas de Funcionamiento .....	161
3.4 RED MODBUS-TCP/IP .....	166
3.4.1 Elementos del Sistema .....	166
3.4.1.1 Interfaz .....	166
3.4.1.2 Cable y Conectores.....	166
3.4.2 Configuración Red Modbus-TCP/IP.....	167
3.4.3 Ejemplo de Conexión.....	169
3.4.4 Pruebas de Funcionamiento .....	172
3.5 CONCLUSIONES.....	177
<b>CAPÍTULO 4: SISTEMA SCADA.....</b>	<b>180</b>
Introducción.....	180
4.1 SISTEMA SCADA ( <i>SUPERVISORY CONTROL AND DATA ACQUISITION</i> , “SUPERVISIÓN CONTROL Y ADQUISICIÓN DE DATOS”) .....	180
4.1.1 Generalidades.....	180

4.1.2 Prestaciones.....	182
4.1.3 El Hardware .....	183
4.1.3.1 Interfaz Hombre-Máquina.....	183
4.1.3.2 Unidad Central MTU ( <i>Master Terminal Unit</i> ).....	184
4.1.3.3 Unidad Remota RTU ( <i>Remote Terminal Unit</i> ) .....	185
4.1.4 Seguridad .....	186
4.1.4.1 Amenazas .....	186
4.1.4.2 Técnicas.....	187
4.2 SISTEMA SCADA WINCC (SIEMENS) .....	188
4.2.1 Características de WINCC .....	188
4.2.1.1 Funcionalidades Básicas .....	188
4.2.1.2 Comunicaciones .....	190
4.2.1.3 Opciones.....	191
4.2.1.4 Interfaces .....	192
4.2.2 SCADA ejemplo mediante una pantalla HMI KTP600.....	193
4.2.2.1 Administración de Usuarios .....	193
4.2.2.2 Uso de Avisos.....	195
4.2.2.3 Uso de Recetas .....	197
4.2.3 Pruebas de Funcionamiento .....	199
4.3 CONCLUSIONES.....	202
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>204</b>
<b>BIBLIOGRAFÍA.....</b>	<b>209</b>
<b>ANEXOS .....</b>	<b>212</b>

## ÍNDICE DE FIGURAS

Figura 1.1: Generación de las Comunicaciones Industriales .....	5
Figura 1.2: Pirámide de la Automatización.....	6
Figura 1.3: Volumen y velocidad de datos en la Pirámide CIM .....	10
Figura 1.4: Sistema de Control Centralizado .....	11
Figura 1.5: Sistema de Control Distribuido .....	12
Figura 1.6: Topología de Red Punto a Punto .....	14
Figura 1.7: Topología de Red Bus .....	15
Figura 1.8: Topología de Red Árbol .....	16
Figura 1.9: Topología de Red Anillo .....	17
Figura 1.10: Topología de Red Estrella .....	18
Figura 1.11: Red Profibus .....	29
Figura 1.12: Capas Implementadas en Profibus.....	31
Figura 1.13: Perfiles Profibus.....	32
Figura 1.14: Mecanismo de control de Acceso al Medio (token bus).....	33
Figura 1.15: Trama de Datos Profibus .....	35
Figura 1.16: Cable y Conector utilizado en RS-485 .....	36
Figura 1.17: Fibra Óptica de plástico y cristal .....	37
Figura 1.18: Conexiones en Profibus-DP.....	38
Figura 1.19: Conexiones en Profibus-PA.....	41
Figura 1.20: Conexiones en Profibus-FMS.....	41
Figura 1.21: Red Profinet .....	43
Figura 1.22: Trama de Datos en Profinet .....	46
Figura 1.23: Comunicaciones de Tiempo Real en Profinet .....	53
Figura 1.24: Conexiones en Profinet-I/O .....	54
Figura 1.25: Conexiones en Profinet-CBA .....	56
Figura 1.26: Red Modbus.....	58
Figura 1.27: Capas Implementadas en Modbus .....	59
Figura 1.28: Estructura de Mensajes Modbus.....	60
Figura 1.29: Trama de Datos en Modbus-ASCII .....	66

Figura 1.30: Trama de Datos en Modbus-RTU.....	67
Figura 1.31: Trama de Datos en Modbus-TCP/IP.....	69
Figura 2.1: Partes Principales de Simatic S7-1200 .....	74
Figura 2.2: Sensor UM 30 y forma de conexión.....	79
Figura 2.3: Bomba Hidráulica.....	80
Figura 2.4: Diagrama P&ID de la Planta .....	81
Figura 2.5: Prototipo Construido.....	81
Figura 2.6: Esquema de Conexión del Proceso .....	82
Figura 2.7: Esquema del Circuito de Acople .....	83
Figura 2.8: Tarjeta Circuito de Acople.....	84
Figura 2.9: Lazo de Realimentación .....	85
Figura 2.10: Acción Integral .....	86
Figura 2.11: Curva de Respuesta al Escalón Unitario.....	88
Figura 2.12: Oscilaciones Sostenidas en la Salida .....	89
Figura 2.13: Sistema de Nivel de Líquido.....	90
Figura 2.14: Diagrama de Bloques del Sistema .....	92
Figura 2.15: Sintonización PID con MATLAB .....	92
Figura 2.16: Parámetros PID obtenidos .....	93
Figura 2.17: Curva de Respuesta del Sistema Simulado.....	93
Figura 2.18: Oscilaciones Sostenidas en la Salida .....	95
Figura 2.19: Creación de un Bloque de Ejecución Cíclica .....	96
Figura 2.20: Tipo de Regulación .....	98
Figura 2.21: Parámetros de Entrada/Salida de la Regulación .....	98
Figura 2.22: Parámetros de Entrada/Salida de la Regulación .....	99
Figura 2.23: Límites del Valor de Salida .....	100
Figura 2.24: Tiempos de Muestreo Disponibles .....	100
Figura 2.25: Visor de señales PID.....	101
Figura 2.26: Estado de la Optimización .....	102
Figura 2.27: Optimización Inicial .....	102
Figura 2.28: Parámetros luego de la Optimización Inicial.....	103
Figura 2.29: Optimización Fina .....	103

Figura 2.30: Parámetros luego de la Optimización Fina .....	104
Figura 2.31: Pantalla SCADA del Sistema (Funcionamiento Automático) .....	105
Figura 2.32: Señales del Sistema (Funcionamiento Automático) .....	105
Figura 2.33: Pantalla SCADA del Sistema (Funcionamiento Manual) .....	106
Figura 2.34: Señales del Sistema (Funcionamiento Manual) .....	106
Figura 2.35: FACTORY I/O .....	107
Figura 2.36: Fábrica Virtual Construida (Vista Frontal).....	110
Figura 2.37: Fábrica Virtual Construida (Perspectiva) .....	111
Figura 2.38: Parámetros de Conexión en FACTORY I/O .....	112
Figura 2.39: Asignación de Variables Digitales.....	113
Figura 2.40: Asignación de Variables Analógicas .....	113
Figura 2.41: Encaminamiento de cajas Largas o Cortas .....	115
Figura 2.42: Medición de Peso de las Cajas .....	116
Figura 2.43: Valores de Peso obtenidos en TIA PORTAL .....	116
Figura 2.44: Descargo de Cajas Medianas y Pequeñas .....	117
Figura 2.45: Total de Cajas Medianas y Pequeñas.....	117
Figura 2.46: Total de Cajas Cortas .....	117
Figura 2.47: Pantalla Principal del SCADA.....	118
Figura 3.1: Cable y Conector para red Profinet .....	122
Figura 3.2: Red Profinet entre dos Simatic S7-1200.....	123
Figura 3.3: Configuración Interfaz Profinet.....	125
Figura 3.4: Comunicación Profinet mediante dos canales de conexión .....	126
Figura 3.5: Comunicación Profinet mediante un canal de conexión .....	126
Figura 3.6: Bloque de Datos PLC1 .....	129
Figura 3.7: Bloque de Datos PLC2 .....	129
Figura 3.8: Parámetros de Conexión PLC1.....	130
Figura 3.9: Instrucción TSEND_C .....	130
Figura 3.10: Parámetros de Conexión PLC2 .....	131
Figura 3.11: Instrucción TRCV_C .....	132
Figura 3.12: Envío de Datos desde PLC1 (1) .....	133
Figura 3.13: Recepción de Datos en PLC2 (1) .....	133

Figura 3.14: Envío de Datos desde PLC1 (2) .....	134
Figura 3.15: Recepción de Datos en PLC2 (2) .....	134
Figura 3.16: Módulo CM 1243-5 Maestro Profibus-DP .....	135
Figura 3.17: Módulo CM 1242-5 Esclavo Profibus-DP .....	137
Figura 3.18: Conector FastConnect Profibus .....	138
Figura 3.19: Terminación y Polarización para el conector RS-485 .....	139
Figura 3.20: Red Profibus entre dos Simatic S7-1200 .....	140
Figura 3.21: Ajustes de Red Profibus .....	140
Figura 3.22: Dirección Profibus Esclavo .....	142
Figura 3.23: Selección Maestro Asignado .....	143
Figura 3.24: Áreas de Transferencia Profibus .....	143
Figura 3.25: Dirección Profibus Maestro.....	144
Figura 3.26: Enviar datos tipo Byte Parte 1, Profibus .....	145
Figura 3.27: Enviar datos tipo Byte Parte 2, Profibus .....	146
Figura 3.28: Enviar datos tipo Word Parte 1, Profibus .....	146
Figura 3.29: Enviar datos tipo Word Parte 2, Profibus .....	147
Figura 3.30: Recibir datos tipo Byte Parte 1, Profibus.....	147
Figura 3.31: Recibir datos tipo Byte Parte 2, Profibus .....	148
Figura 3.32: Recibir datos tipo Word Parte 1, Profibus .....	148
Figura 3.33: Recibir datos tipo Word Parte 2, Profibus .....	149
Figura 3.34: Módulo CM 1241 Esclavo Profibus-DP.....	150
Figura 3.35: Cableado Red Modbus sobre RS-485 .....	151
Figura 3.36: Modo de Operación módulo CM 1241 .....	156
Figura 3.37: Instrucción MB_COMM_LOAD en Maestro .....	157
Figura 3.38: Instrucción MB_MASTER, leer salidas de un esclavo .....	159
Figura 3.39: Instrucción MB_MASTER, escribir salidas de un esclavo .....	159
Figura 3.40: Instrucción MB_MASTER, leer registros de almacenamiento .....	159
Figura 3.41: Instrucción MB_MASTER, escribir registros de almacenamiento .....	160
Figura 3.42: Instrucción MB_COMM_LOAD en Esclavo .....	160
Figura 3.43: Instrucción MB_SLAVE .....	161
Figura 3.44: Petición Uno Parte 1, Modbus-RTU .....	162

Figura 3.45: Petición Uno Parte 2, Modbus-RTU .....	163
Figura 3.46: Petición Dos Parte 1, Modbus-RTU .....	163
Figura 3.47: Petición Dos Parte 2, Modbus-RTU .....	164
Figura 3.48: Petición Tres Parte 1, Modbus-RTU .....	164
Figura 3.49: Petición Tres Parte 2, Modbus-RTU .....	165
Figura 3.50: Petición Cuatro Parte 1, Modbus-RTU .....	165
Figura 3.51: Petición Cuatro Parte 2, Modbus-RTU .....	165
Figura 3.52: Instrucción MB_CLIENT, leer datos de salida y entrada de un esclavo...	171
Figura 3.53: Instrucción MB_CLIENT, escribir datos y leer registros de un esclavo...	171
Figura 3.54: Instrucción MB_SERVER .....	172
Figura 3.55: Petición Uno Parte 1, Modbus-TCP/IP .....	173
Figura 3.56: Petición Uno Parte 2, Modbus-TCP/IP .....	174
Figura 3.57: Petición Dos Parte 1, Modbus-TCP/IP .....	174
Figura 3.58: Petición Dos Parte 2, Modbus-TCP/IP .....	174
Figura 3.59: Petición Tres Parte 1, Modbus-TCP/IP .....	175
Figura 3.60: Petición Tres Parte 2, Modbus-TCP/IP .....	175
Figura 3.61: Petición Cuatro Parte 1, Modbus-TCP/IP .....	176
Figura 3.62: Petición Cuatro Parte 2, Modbus-TCP/IP .....	176
Figura 4.1: Esquema de un Sistema SCADA .....	181
Figura 4.2: Pantalla HMI de un Sistema SCADA.....	184
Figura 4.3: Avisos en WinCC .....	189
Figura 4.4: Grupos de Usuarios .....	194
Figura 4.5: Usuarios .....	194
Figura 4.6: Seguridad de operación sobre objetos de acción .....	195
Figura 4.7: Avisos digitales o de bit .....	196
Figura 4.8: Avisos analógicos .....	196
Figura 4.9: Configuraciones, visor de avisos .....	196
Figura 4.10: Configuración general recetas .....	197
Figura 4.11: Configuración elementos de receta .....	197
Figura 4.12: Configuración registros de receta .....	198
Figura 4.13: Botones añadidos al visor de recetas .....	199

Figura 4.14: Login usuario 1 .....	199
Figura 4.15: Editar usuarios en runtime .....	200
Figura 4.16: Inicio de sesión con un usuario autorizado .....	200
Figura 4.17: Visor de avisos en runtime .....	201
Figura 4.18: Texto completo del aviso.....	201
Figura 4.19: Uso del visor de recetas en runtime.....	201
Figura 4.20: Elegir y cargar datos de registro al PLC .....	202
Figura P1.1: Pantalla de Inicio P1 .....	215
Figura P1.2: Ventana Primeros Pasos P1 .....	215
Figura P1.3: Agregar PLC1 al proyecto.....	216
Figura P1.4: Establecer Ninguna Protección para el PLC1 .....	216
Figura P1.5: Configuración IP PLC1 .....	217
Figura P1.6: Ajustar Dirección Inicial de Entrada en el PLC1 .....	217
Figura P1.7: Dirección Inicial de Salida en el PLC1 .....	217
Figura P1.8: Configuración de la Comunicación en FACTORY I/O .....	218
Figura P1.9: Configuración del Driver para Comunicación con el PLC .....	218
Figura P1.10: Paleta de Objetos FACTORY I/O .....	219
Figura P1.11: Rueda Clasificadora P1 .....	219
Figura P1.12: Alineadores P1 .....	219
Figura P1.13: Rampas P1 .....	220
Figura P1.14: Emisor de Cajas P1 .....	220
Figura P1.15: Soportes P1 .....	220
Figura P1.16: Sensores Difusos P1 .....	221
Figura P1.17: Visualización Etiquetas de Sensores y Actuadores .....	221
Figura P1.18: Etiquetas Rueda Clasificadora .....	221
Figura P1.19: Asignación Etiquetas de Entrada y Salida .....	222
Figura P1.20: Iniciar Aplicación Virtual .....	222
Figura P1.21: Ejercicio para el Alumno, Proceso en FACTORY I/O .....	223
Figura P2.1: Agregar PLC al proyecto .....	226
Figura P2.2: Agregar Pantalla HMI .....	227
Figura P2.3: Vincular PLC a pantalla HMI .....	227

Figura P2.4: Configuración color de fondo pantalla HMI .....	228
Figura P2.5: Desactivar Avisos de Sistema .....	228
Figura P2.6: Agregar Botones del Sistema .....	229
Figura P2.7: Red Profinet entre PLC y HMI .....	229
Figura P2.8: Configurar dirección IP de la HMI .....	230
Figura P2.9: Agregar variable de tipo Int en el PLC .....	230
Figura P2.10: Función “Definir Variable” .....	230
Figura P2.11: Seleccionar Variable creada .....	231
Figura P2.12: Seleccionar valor desde Variable HMI .....	231
Figura P2.13: Seleccionar el valor de proceso .....	231
Figura P2.14: Agregar Tabla de Observación .....	231
Figura P2.15: Verificar cambio de valor en la variable “Valor” .....	232
Figura P2.16: Función “ActivarBit” .....	232
Figura P2.17: Función “DesactivarBit” .....	232
Figura P2.18: Agregar Objetos Prediseñados de WinCC .....	233
Figura P2.19: Activar Marcas de Ciclo.....	233
Figura P2.20: Generar contador a partir de reloj interno de 5Hz .....	234
Figura P2.21: Seleccionar variable de contaje actual .....	234
Figura P3.1: Lazo de Realimentación .....	237
Figura P3.2: Insertar un bloque de ejecución cíclica .....	240
Figura P3.3: Insertar un bloque de ejecución cíclica .....	240
Figura P3.4: Escoger Tipo de Regulación .....	241
Figura P3.5: Escoger Parámetro de Entrada/Salida .....	241
Figura P3.6: Límites del valor real .....	241
Figura P3.7: Límites del valor real .....	242
Figura P3.8: Escalar el valor del Sensor .....	242
Figura P3.9: Instrucción PID_COMPACT .....	243
Figura P3.10: Optimización Inicial .....	244
Figura P3.11: Cargar Parámetros PID .....	244
Figura P3.12: Optimización Fina .....	244
Figura P3.13: Entrada Manual para Parámetro PID .....	245

Figura P4.1: Trama de Datos en Profinet .....	248
Figura P4.2: Agregar PLC1 al proyecto .....	249
Figura P4.3: PLC2 agregado al proyecto .....	249
Figura P4.4: Conexión Profinet en TIA PORTAL.....	250
Figura P4.5: Configuración IP PLC1 .....	250
Figura P4.6: Configuración IP PLC2 .....	251
Figura P4.7: Activar Marcas de Ciclo del PLC.....	251
Figura P4.8: Creación Bloque de datos TSEND_C .....	252
Figura P4.9: Parámetros de Conexión TSEND_C .....	252
Figura P4.10: Creación Bloque de datos TSEND_C .....	253
Figura P4.11: Parámetros de Conexión TRCV_C .....	253
Figura P4.12: Datos de Conexión .....	254
Figura P4.13: Parámetros REQ y CON de las Instrucciones .....	254
Figura P4.14: Parámetro DATA en TSEND_C, enviar 1 Byte.....	254
Figura P4.15: Instrucción TSEND_C .....	255
Figura P4.16: Parámetro DATA en TRCV_C, enviar 1 Byte .....	255
Figura P4.17: Instrucción TRCV_C .....	255
Figura P4.18: Bloques de Datos PLC2 .....	256
Figura P4.19: Configuración canal 2.....	256
Figura P4.20: Parámetro DATA en TSEND_C, enviar 1 <i>Char</i> .....	257
Figura P4.21: Bloques de Datos, envió <i>Char2</i> .....	257
Figura P5.1: Trama de Datos en Profinet .....	261
Figura P5.2: Agregar Bloque de Datos .....	262
Figura P5.3: Crear Variable de tipo Struct .....	263
Figura P5.4 Variables a enviar .....	263
Figura P5.5: Bloques TSEND_C y TRCV_C .....	263
Figura P5.6: Configuración Campo de Texto para ingresar datos .....	264
Figura P5.7: Configuración Campo de Texto para visualizar datos .....	264
Figura P5.8: Pantalla PCL1 .....	264
Figura P5.9: Pantalla PCL2 .....	265
Figura P5.10: Escalado de Variable Analógica .....	266

Figura P6.1: Trama de Datos Profibus .....	270
Figura P6.2: Cambiar Nombre PLC1 P5 .....	271
Figura P6.3: Nombres PLCs P5 .....	271
Figura P6.4: Agregar Módulo Maestro CM 1243-5 .....	271
Figura P6.5: Agregar Módulo Esclavo CM 1242-5 .....	272
Figura P6.6: Red Profibus P5 .....	272
Figura P6.7: Configuraciones de Bus Profibus .....	273
Figura P6.8: Dirección Profibus Maestro .....	273
Figura P6.9: Dirección Profibus Esclavo .....	273
Figura P6.10: Modo Esclavo DP .....	274
Figura P6.11: Crear Área de Transferencia Profibus .....	274
Figura P6.12: Área de Transferencia 1.....	275
Figura P6.13: Área de Transferencia 2.....	275
Figura P6.14: Área de Transferencia 3 y 4 .....	276
Figura P6.15: Áreas de Transferencias .....	276
Figura P6.16: Segmento 1 Maestro P5.....	277
Figura P6.17: Segmento 1 Esclavo P5 .....	277
Figura P6.18: Segmento 2 Esclavo P5 .....	278
Figura P6.19: Segmento 2 Maestro P5 .....	278
Figura P6.20: Agregar Signal Board.....	279
Figura P6.21: Configuración Canal 0 Analógico .....	279
Figura P6.22: Dirección Inicial Entradas Analógicas .....	279
Figura P6.23: Segmento 4 Maestro P5 .....	279
Figura P6.24: Segmento 2 Maestro P5 .....	280
Figura P6.25: Agregar Tabla de Observación .....	280
Figura P6.26: Visualización Tabla de Observación PLC ESCLAVO .....	280
Figura P6.27: Configuración entrada Analógica .....	281
Figura P6.28: Dirección Entrada Analógica .....	281
Figura P6.29: Segmento 3 Esclavo P5 .....	281
Figura P6.30: Segmento 3 Maestro P5 .....	281
Figura P6.31: Visualización Tabla de Observación PLC MAESTRO .....	282

Figura P7.1: Trama de Datos en Modbus-RTU .....	286
Figura P7.2: PLC1 agregado al proyecto .....	286
Figura P7.3: PLC2 agregado al proyecto .....	287
Figura P7.4: Cambiar Nombre PLC1 P6 .....	287
Figura P7.5: Nombres PLCs P6 .....	287
Figura P7.6: Agregar Módulo de Comunicación CM 1241 a Maestro .....	288
Figura P7.7: Agregar Módulo de Comunicación CM 1241 a Esclavo .....	288
Figura P7.8: Modo de Operación Módulos CM 1241 .....	289
Figura P7.9: ID de Hardware Módulo CM 1241 .....	289
Figura P7.10: Activar Marcas de Ciclo de los PLCs .....	290
Figura P7.11: Creación Bloque de datos MB_COMM_LOAD .....	290
Figura P7.12: Creación Bloque de datos MB_MASTER .....	291
Figura P7.13: Instrucción MB_COMM_LOAD Maestro .....	291
Figura P7.14: Creación Bloque de datos MB_SLAVE .....	292
Figura P7.15: Instrucción MB_COMM_LOAD Esclavo .....	292
Figura P7.16: Instrucción MB_SLAVE .....	292
Figura P7.17: Instrucción MB_MASTER 1 .....	293
Figura P7.18: Segmento 3 Maestro P6 .....	293
Figura P7.19: Segmento 3 Esclavo P6 .....	294
Figura P7.20: Instrucción MB_MASTER 2 .....	294
Figura P7.21: Segmento 5 Maestro P6 .....	295
Figura P7.22: Instrucción MB_MASTER 3 .....	296
Figura P7.23: Segmento 7 Maestro P6.....	296
Figura P7.24: Segmento 8 Maestro P6 .....	296
Figura P7.25: Segmento 4 Esclavo P6 .....	297
Figura P7.26: Agregar Signal Board .....	297
Figura P7.27: Instrucción MB_MASTER 4 .....	298
Figura P7.28: Segmento 10 Maestro P6 .....	298
Figura P7.29: Segmento 5 Esclavo P6 .....	298
Figura P8.1: Trama de Datos en Modbus-TCP/IP .....	301
Figura P8.2: PLC1 agregado al proyecto .....	302

Figura P8.3: PLC2 agregado al proyecto .....	302
Figura P8.4: PLC3 agregado al proyecto .....	303
Figura P8.5: Cambiar Nombre PLC1 P8 .....	303
Figura P8.6: Nombres PLCs P8 .....	303
Figura P8.7: Creación Bloque de datos MB_CLIENT .....	304
Figura P8.8: Creación Bloque de datos MB_CLIENT 1 .....	305
Figura P8.9: Creación Bloque de datos MB_SERVER Esclavo1 .....	305
Figura P8.10: Instrucción MB_SERVER Esclavo1 (Canal 1) .....	306
Figura P8.11: Instrucción MB_SERVER Esclavo1 (Canal 2) .....	306
Figura P8.12: Creación Bloque de datos MB_SERVER Esclavo2 .....	307
Figura P8.13: Instrucción MB_SERVER Esclavo2 (Canal 1) .....	307
Figura P8.14: Instrucción MB_CLIENT, Leer Entradas Esclavo1 .....	308
Figura P8.15: Instrucción MB_CLIENT, Leer Entradas Esclavo2 .....	309
Figura P8.16: Segmento 3 y 4 Maestro P8 .....	309
Figura P8.17: Instrucción MB_CLIENT, Escribir Salidas Esclavo2 .....	310
Figura P8.18: Segmento 6 Maestro P8 .....	310
Figura P8.19: Instrucción MB_CLIENT, Leer Valor de Registro Esclavo1 .....	311
Figura P8.20: Segmento 2 Esclavo1 P8 .....	312
Figura P8.21: Segmento 8 Maestro P8 .....	312
Figura P8.22: Segmento 9 Maestro P8 .....	312
Figura P9.1: PLC1 agregado al proyecto .....	316
Figura P9.2: Agregar HMI PLC1 al proyecto .....	317
Figura P9.3: Vincular el PLC a la pantalla HMI P8 .....	317
Figura P9.4: Asistente de Configuración Inicial para Pantallas HMI .....	318
Figura P9.5: Encender Salida Q0.0 y Establecer Variable “Estado” .....	318
Figura P9.6: Variable “Estado” .....	318
Figura P9.7: Propiedades Botón “Iniciar Sesión” .....	319
Figura P9.8: Eventos Botón “Iniciar Sesión” .....	319
Figura P9.9: Propiedades Botón “Cerrar Sesión” .....	319
Figura P9.10: Visor de Usuarios .....	320
Figura P9.11: Propiedades Botón “Visor de Usuarios” .....	320

Figura P9.12: Imagen_Raíz .....	320
Figura P9.13: Abrir Administración de Usuarios.....	321
Figura P9.14: Grupos de Usuarios .....	321
Figura P9.15: Usuarios.....	321
Figura P9.16: Usuario “Operador1” .....	322
Figura P9.17: Segmento Encender Variable mediante marca M10.0 .....	322
Figura P9.18: Agregar nueva Imagen .....	322
Figura P9.19: Eventos Botón “Encender Salida”.....	322
Figura P9.20: Seguridad Botón “Encender Salida” .....	323
Figura P9.21: Animaciones Círculo.....	323
Figura P9.22: Visor de Avisos .....	323
Figura P9.23: Abrir Avisos HMI .....	323
Figura P9.24: Aviso de bit .....	324
Figura P9.25: Eventos botón F1 de Imagen_Raíz.....	324
Figura P9.26: Pantalla Imagen Raíz en Runtime .....	325
Figura P9.27: Pantalla Imagen1 en Runtime .....	325
Figura P10.1: Variables para Recetas .....	329
Figura P10.2: Creación de Recetas .....	330
Figura P10.3: Agregar Elementos a la Receta .....	330
Figura P10.4: Elementos de Recetas.....	330
Figura P10.5: Registro de Receta 1 .....	330
Figura P10.6: Registro de Receta 2 .....	331
Figura P10.7: Visor de Recetas .....	331
Figura P10.8: Edición de Receta en Runtime .....	331
Figura P10.9: Configuraciones de Visualización .....	332
Figura P10.10: Agregar Tabla de Observación.....	332
Figura P10.11: Recetas en HMI .....	332
Figura P10.12: Registros de Receta en HMI .....	333
Figura P10.13: Variables de Receta .....	333
Figura P10.14: Transferir Variables hacia el controlador .....	334
Figura P10.15: Tabla de Observación .....	334

Figura P11.1: Esquema Práctica 10 .....	337
Figura P11.2: Pantallas HMI Principal .....	338
Figura P11.3: Pantallas HMI PWM .....	339
Figura P11.4: Pantallas HMI Centro .....	340

## ÍNDICE DE TABLAS

Tabla 1.1: Comparativa de algunos Buses de Campo.....	25
Tabla 1.2: Ventajas y desventajas de los principales buses de campo.....	26
Tabla 1.3: Características de los Medios de Transmisión Profibus.....	37
Tabla 1.4: Características de los Medios de Transmisión Profinet.....	48
Tabla 1.5: Principales códigos de función Modbus.....	64
Tabla 1.6: Formato de Byte en Modbus-ASCII.....	65
Tabla 1.7: Formato de Byte en Modbus-RTU.....	67
Tabla 1.8 Tabla Resumen: redes Profibus, Profinet y Modbus.....	72
Tabla 2.1: Características Principales de las CPUs S7-1200.....	75
Tabla 2.2: Módulos Adicionales para CPUs S7-1200.....	76
Tabla 2.3: Simbología del Diagrama P&ID.....	81
Tabla 2.4: Parámetros Ziegler-Nichols, Primer Método.....	89
Tabla 2.5: Parámetros Ziegler-Nichols, Segundo Método.....	90
Tabla 2.6: Características Cajas en Factory I/O.....	108
Tabla 3.1: Asignación de los pines del conector hembra DB9 en redes RS-485.....	138
Tabla 3.2: Códigos de Función para lectura Modbus.....	152
Tabla 3.3: Códigos de Función para escritura Modbus.....	153
Tabla 3.4: Códigos de función vs parámetros de la Instrucción MB_MASTER.....	155
Tabla 4.1: Datos de los usuarios creados.....	194
Tabla 4.2: Datos de recetas creadas.....	198
Tabla P2.1: Características Pantallas KTP600.....	225
Tabla P7.1: Códigos de Función para lectura Modbus.....	284
Tabla P7.2: Códigos de Función para escritura Modbus.....	284
Tabla P7.3: Formato de Byte en Modbus-RTU.....	285
Tabla P11.1: Resume Hardware Práctica 11.....	337

**ÍNDICE DE ANEXOS**

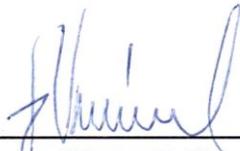
ANEXO 1: PRÁCTICA 1 .....	213
ANEXO 2: PRÁCTICA 2 .....	224
ANEXO 3: PRÁCTICA 3 .....	236
ANEXO 4: PRÁCTICA 4 .....	246
ANEXO 5: PRÁCTICA 5 .....	259
ANEXO 6: PRÁCTICA 6 .....	267
ANEXO 7: PRÁCTICA 7 .....	283
ANEXO 8: PRÁCTICA 8 .....	300
ANEXO 9: PRÁCTICA 9 .....	314
ANEXO 10: PRÁCTICA 10 .....	327
ANEXO 11: PRÁCTICA 11 .....	336

**“IMPLEMENTACIÓN DE LOS PROTOCOLOS DE COMUNICACIÓN INDUSTRIAL MEDIANTE SIMATIC S7-1200 PARA EL LABORATORIO DE AUTOMATIZACIÓN DE LA UNIVERSIDAD DEL AZUAY”**

**RESUMEN**

En este trabajo de titulación se desarrolla un estudio e implementación de redes de comunicación industrial como: Profinet, Profibus, Modbus-RTU y Modbus-TCP. Estos sistemas de comunicación se despliegan para la interconexión de controladores Simatic S7-1200 en el Laboratorio de Automatización de la Universidad del Azuay, en cada caso se realiza un estudio de los protocolos, interfaces e instrucciones requeridas para la transmisión de datos entre controladores o mediante el modelo maestro-esclavo. El proyecto busca dar solución a los problemas de conectividad de las distintas áreas de un proceso productivo que permita la integración de todos los elementos involucrados en el mismo. Además se aborda temas como el control PID; los Sistemas de Adquisición, Supervisión y Control de Datos (SCADA) con el fin de integrar todas las áreas de la automatización.

**Palabras Clave:** Comunicación Industrial, Profibus, Profinet, Modbus, Control PID.



Francisco Eugenio Vásquez Calero

**Director de Tesis**



Hugo Marcelo Torres Salamea

**Director de Escuela**



Diego Javier Sarmiento Sánchez

**Autor**

**IMPLEMENTATION OF INDUSTRIAL COMMUNICATION PROTOCOLS  
BASED ON SIMATIC S7-1200 FOR THE AUTOMATION LABORATORY OF  
UNIVERSIDAD DEL AZUAY**

**ABSTRACT**

This graduation paper deals with the study and implementation of industrial communication networks such as Profinet, Profibus, Modbus-RTU and Modbus-TCP. These communication systems are displayed to interconnect Simatic S7-1200 controllers in the Automation Laboratory at *Universidad del Azuay*. A study of protocols, interfaces and instructions required for data transmission between controllers or by the master-slave mode is performed in each case. The project seeks to solve connectivity problems in the different areas of a production process that allows the integration of all the elements involved in it. In addition, issues such as PID control, Acquisition Systems, Supervisory Control and Data Acquisition (SCADA) are addressed in order to integrate all automation areas.

**Keywords:** Industrial Communication, Profibus, Profinet, Modbus, PID Control



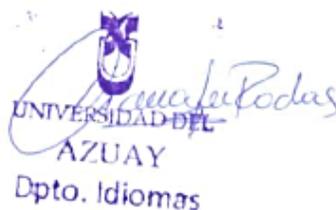
Francisco Eugenio Vásquez Calero  
**Thesis Director**



Hugo Marcelo Torres Salamea  
**School Director**



Diego Javier Sarmiento Sánchez  
**Author**




Translated by,  
Lic. Lourdes Crespo

Diego Javier Sarmiento Sánchez

Trabajo de Graduación

Ing. Francisco Eugenio Vásquez Calero

Marzo, 2016

**IMPLEMENTACIÓN DE LOS PROTOCOLOS DE COMUNICACIÓN  
INDUSTRIAL MEDIANTE SIMATIC S7-1200 PARA EL LABORATORIO DE  
AUTOMATIZACIÓN DE LA UNIVERSIDAD DEL AZUAY**

**INTRODUCCIÓN**

A lo largo de toda la humanidad, el hombre ha buscado la manera de crear nuevos dispositivos, mercancías, artículos y tecnología en general que le permita mejorar su calidad de vida, todos los productos que usamos en la actualidad fueron desarrollados en una fábrica a través de diferentes cadenas productivas. Este entorno industrial ha sido testigo de grandes cambios desde sus inicios hasta contar con las gigantescas infraestructuras y maquinarias utilizadas hoy en día, en esta transformación las comunicaciones han sido un pilar fundamental de desarrollo, al permitir integrar las diferentes áreas de manera rápida y segura.

En un proceso industrial se tiene varios elementos involucrados desde actuadores, pasando por controladores y sistemas de supervisión, los cuales necesariamente deberán resolver los problemas de comunicación, tanto con los elementos del mismo nivel como con dispositivos de niveles superiores. La falta de comunicación con equipos del mismo nivel no permite el despliegue de una red de campo, que al ser establecida brindan varios beneficios al sistema y facilita su escalabilidad en términos de comunicación. Por otro lado la falta de comunicación con dispositivos de niveles superiores impide la aplicación de dos aspectos de gran relevancia en las fábricas actuales como: la implementación de un

sistema de supervisión central y la generación de reportes de producción, ambos sumamente necesarios para el análisis productivo y económico de la planta.

Por lo tanto esta área de estudio ha ganado cada vez más importancia, brindando a las fábricas la posibilidad de comunicarse con el interior y exterior de la red industrial, además de enlazar cada vez más los procesos con herramientas del mundo informático.

De esta manera, en la Universidad del Azuay se ha visto necesario la implementación de las comunicaciones industriales para el laboratorio de Automatización, con el fin de desarrollar aplicaciones más complejas, en las cuales se trabaje con intercambio de datos en el entorno industrial, según los estándares mundiales desarrollados.

Este trabajo busca brindar a los estudiantes las competencias necesarias para desplegar, configurar y controlar la transmisión de datos en diferentes redes de comunicación como: Profinet, Profibus y Modbus, con el fin de interconectar controladores de la familia Simatic S7-1200; además de brindar herramientas para la aplicación de control lógico en sistemas de automatización virtuales a través de un *software* de simulación y también para sistemas reales a través del control PID.

El primer paso para desarrollar el proyecto será determinar los tipos de estándares de comunicación industrial que se manejan en el mundo a través un método de investigación bibliográfico, el mismo permitirá obtener documentación pertinente de los protocolos más utilizados así como sus características. Posteriormente se utilizará un método descriptivo y experimental con el objetivo de controlar los sistemas de automatización virtual y real.

En la tercera etapa se hará uso de un método descriptivo y experimental, de tal manera que se pueda establecer los diferentes módulos, características y requerimientos de cada red, estas implementaciones serán documentadas en forma de prácticas con el fin de convertirse en una guía para los estudiantes.

## CAPÍTULO 1

### REDES DE COMUNICACIÓN INDUSTRIAL

#### **Introducción**

En el presente capítulo se desarrolla un resumen de los fundamentos teóricos de las comunicaciones industriales, en la primera parte se aborda temas como: una breve historia, el modelo que ha sido estandarizado para el intercambio de información en dichos entornos, los diferentes sistemas de control enfocados a la transmisión de datos, las topologías usadas y una breve reseña de los protocolos de comunicación más importantes en la actualidad. Luego se realiza un análisis más profundo de los buses de campo y redes de comunicación que se implementarán en este trabajo: Profibus, Profinet y Modbus; de los que se estudiarán sus principales características, arquitectura, control de acceso al medio, medios físicos, variantes y demás temas relevantes que permitirán su comprensión y desarrollo dentro de un sistema de automatización.

#### **1.1 LAS COMUNICACIONES INDUSTRIALES**

##### **1.1.1 Introducción**

Durante todo el desarrollo de la humanidad, desde sus inicios las personas han encontrado un reto importante en el hecho de poder comunicarse con su entorno, la historia que se sabe de nuestros antepasados nos permite conocer varios de los métodos de comunicación que usaban para resolver las necesidades de aquel entonces. Una de las primeras formas de comunicación era mediante gruñidos y sonidos, además de gestos: con las manos, el rostro y cualquier parte del cuerpo. Las culturas que aparecieron luego empezaron a usar medios naturales para plasmar sus símbolos y poder enviar mensajes en hojas de papiro, además representar sus ideas en rocas, cuevas, etc. Posteriormente las culturas como la romana empezaron a utilizar el fuego a través antorchas en las montañas y las primeras formas de codificación de mensajes. Otro de los más famosos medios por el cual se

comunicaban fueron las señales de humo, utilizado para interactuar con personas a una mayor distancia.

Las culturas occidentales y civilizaciones que se asentaron en América del Norte y Sur empezaron a usar a personas como mensajeros que eran encargados de portar las cartas cubriendo grandes distancias. Posteriormente cuando la humanidad empezó a desarrollarse en el ámbito de las tecnologías aparecen inventos importantes desde el teléfono, radio, la televisión hasta medios de comunicación inalámbricos como los celulares, los enlaces terrenales y satelitales, el *Internet*; los cuales son ampliamente usados en la actualidad y son un aspecto sumamente importante sin el cual sería imposible la evolución humana a los niveles actuales. Se puede mencionar varias tecnologías que forman parte de la vida cotidiana como: *USB, bluetooth, wifi, GSM, GPRS, Ethernet* etc. que son aplicadas a todo tipo de dispositivos a lo largo de todas las áreas del desarrollo humano (Guerrero, Yuste, & Martínez, 2009).

Las comunicaciones industriales han sido un efecto directo del desarrollo expuesto en el párrafo anterior, permiten la transmisión de información del controlador<sup>1</sup> a los diferentes elementos de campo que se encuentren involucrados en un proceso productivo como pueden ser: detectores, actuadores, sensores, etc.

En sus inicios la industria manejaba procesos mediante variables todo/nada (*on/off*, “encendido/apagado”) que se traducían en impulsos eléctricos binarios. Cuando aumentó la complejidad de los sistemas productivos, surgió la necesidad que utilizar varias máquinas enfocadas a cada área de manufactura, es así que aparecen diferentes equipos con distintos controladores dando origen a las llamadas "Islas de Automatización", denominadas así ya que los sistemas carecían de la capacidad para comunicarse entre ellos, sin duda esto generó varios problemas para gestionar un proceso productivo.

---

<sup>1</sup> Controlador: Elemento de *hardware* que contiene los algoritmos de control para sistemas de automatización

El entorno mencionado dio lugar a investigaciones interesadas en brindar a los sistemas electrónicos la capacidad de control y comunicación entre ellos además de con su entorno, dicha área recibe el nombre hasta la actualidad de "Comunicaciones Industriales".

El desarrollo acelerado que se tuvo en esta área fue básicamente el resultado de dos circunstancias como: la elevación de la complejidad de los procesos y la evolución de las comunicaciones digitales (véase Figura 1.1) (Mandado et al., 2006).

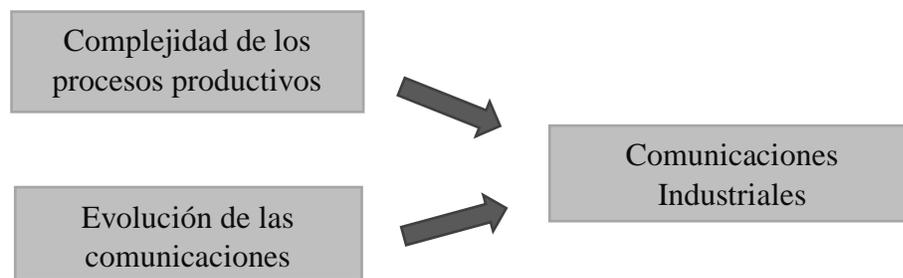


Figura 1.1: Generación de las Comunicaciones Industriales

Fuente: (Guerrero, Yuste, & Martínez, 2009)

La evolución de la tecnología en las comunicaciones se encuentra también directamente aplicada en los dispositivos industriales, por medio de los cuales se logra una completa cohesión de los diferentes sistemas, procesos e instalaciones que supone un pilar fundamental para alcanzar los niveles de operatividad y competitividad que se requieren en el mundo moderno.

Un sistema de comunicación de datos industrial es cada vez más exigente cuanto más cerca del proceso físico se encuentra. Las tres principales características que determinan la aplicación de las diferentes redes de comunicación son: volumen de datos, velocidad de transmisión y velocidad de respuesta.

### 1.1.2 Pirámide de la automatización CIM (Computer Integrated Manufacturing, “Fabricación Integrada por Computador”)

El modelo CIM empleado para representar las comunicaciones industriales describe un entorno de implementación de protocolos adoptados desde el nivel físico hasta el nivel más alto de la red siguiendo un paralelismo. Las comunicaciones se agrupan jerárquicamente en función de la información, cada subsistema debe tener comunicación directa con los subsistemas del mismo nivel y los de niveles superiores e inferiores. Dentro del mundo industrial esto corresponde a una estructura piramidal jerarquizada que permite a los conglomerados tomar decisiones empresariales en la cúspide de la torre, las estrategias adoptadas por los empresarios se deben traducir a operaciones de control de bajo nivel, mediante la implementación de las acciones en el sistema distribuido de la pirámide (véase Figura 1.2) (Salazar & Correa, 2011).

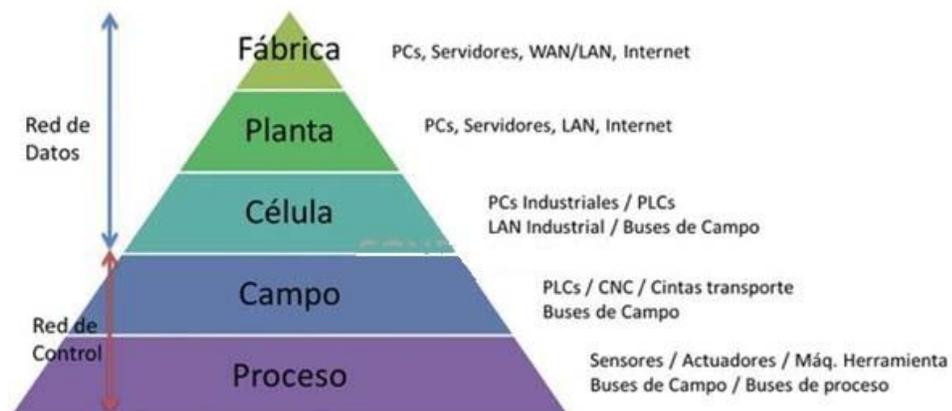


Figura 1.2: Pirámide de la Automatización

Fuente: (Conectronica, 2013)

La necesidad de integrar los procesos productivos (diseño, ingeniería y fabricación) con los niveles de gestión de la planta se ha plasmado en el modelo CIM, que permite dicha integración en mayor o menor medida según la implementación de los dispositivos necesarios para cada nivel de la pirámide, y dentro de las áreas empresariales correspondientes:

- Órdenes de entrada.
- Control de inventarios.
- Planificación de necesidades.
- Diseño del producto y proceso.
- Simulación.
- Planificación de la fábrica.
- Automatización de procesos.
- Control de calidad y ventas.

La división en niveles de una estructura de un sistema de producción propicia la representación de la pirámide CIM conformada por cuatro niveles perfectamente definidos como: nivel de Oficinas/Gestión, nivel de Planta, nivel de Célula y nivel de Campo (Mandado et al., 2006).

Según los presentado por (Mandado et al., 2006), (Salazar & Correa, 2011), (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008), (Guerrero, Yuste, & Martínez, 2009) y (Gómez, Leopoldo, & Leticia, 2014) se puede definir cada nivel de la pirámide de la siguiente manera.

#### **a. Nivel de Proceso**

En este nivel se encuentran los elementos de medida o adquisición de datos llamados sensores y los elementos de mando llamados actuadores, ambos distribuidos a lo largo del proceso productivo. Los sensores transfieren sus señales a los dispositivos del nivel inmediatamente superior para que ejecuten los algoritmos de control y envíen las respuestas adecuadas a los elementos actuadores. Este nivel está encargado de comunicar estos elementos llamados de campo o *field* con los controladores del nivel de célula a través de los denominados buses de campo.

### **b. Nivel de Campo**

En este nivel se sitúan los dispositivos de mando y control como: PLCs<sup>2</sup> de gama media y baja, equipos basados en microprocesadores, controladores de *robots*, *drivers*, sistemas de control numérico<sup>3</sup>, sistemas de control de movimiento<sup>4</sup>, embebidos<sup>5</sup> y basados en PC, también interfaces hombre-máquina capaces de gestionar actuadores y sensores del nivel de campo.

Este nivel es el encargado del procesamiento de las variables de campo mediante la programación de sus diferentes dispositivos, con el fin de emitir órdenes hacia los elementos del nivel inferior.

### **c. Nivel de Célula**

En este nivel se sitúan los elementos de control como PLCs y PCs industriales que permiten recoger la información del nivel de campo y organizar las acciones según la programación íntegra del proceso productivo. Además se encarga de ejecutar las tareas de supervisión, optimización y mantenimiento recibidas del nivel de planta.

La comunicación constante con el nivel superior permite implementar instrucciones de planta mediante acciones más sencillas y sincronizadas, gracias a la programación y estructuración de los controladores.

### **d. Nivel de Planta**

En este nivel se localiza los dispositivos encargados de la secuenciación de tareas y administración de los recursos de los niveles inferiores, se tienen sistemas para el control de proceso DSC<sup>6</sup> y sistemas de supervisión-mando mediante aplicaciones SCADA<sup>7</sup>. Las

---

<sup>2</sup> PLC: Controlador Lógico Programable, muy utilizado en procesos de automatización.

<sup>3</sup> Control Numérico CN: Sistemas de automatización de máquinas herramientas

<sup>4</sup> Control de Movimiento: Sistemas para el control de motores paso a paso y servomotores con el fin de automatizar procesos de manera precisa mediante ejes de movimiento.

<sup>5</sup> Embebidos: Sistemas computacionales diseñados para realizar tareas específicas.

<sup>6</sup> DSC (*Datalogging and Supervisory Control*, “Control, Supervision y Registro de Datos”)

<sup>7</sup> SCADA (*Supervisory Control And Data Acquisition*, “Supervisión, Control y Adquisición de Datos”)

actividades que se realizan están enfocadas a la planificación y control de la producción tales como: diseño e ingeniería, instalación y puesta en servicio, manejo, mantenimiento y actualización.

#### **e. Nivel de Fábrica u Oficina**

Este nivel es el encargado de la gestión total de la planta, es decir, se considera todas las áreas desde un punto de vista empresarial incluyendo: ventas, comercialización, manufactura, planificación a medio y largo plazo, relaciones con proveedores y clientes. Se emplean ordenadores de oficina y de ingeniería, estaciones de trabajo, servidores, para los cuales no existe una exigencia en el tiempo de respuesta, más bien están encargados de grandes volúmenes de datos de producción que permiten un análisis y gestión de las ramas productivas dentro de las fábricas.

A lo largo de la pirámide de la automatización se observa ciertas características propias para cada nivel, por ejemplo a medida que se va escalando se requiere cada vez menos velocidad de respuesta de los equipos involucrados en el proceso, es así que para el nivel de campo se tiene una tolerancia de tan solo milisegundos de retraso en las señales, ya que estos elementos son los encargados de la interacción directa con el proceso productivo.

Con una lógica similar se puede establecer que la cantidad de equipos participantes en cada nivel se reduce, cuanto más alto sea su localización dentro de la pirámide; para el primer nivel se incluye una gran cantidad de equipos entre sensores y actuadores debido a la naturaleza misma de las cadenas de manufactura, las cuales requieren en cada etapa diferentes formas de controlar y medir las variables involucradas, en cambio en el nivel más alto se sitúan solo unos cuantos equipos dependiendo la forma de gestión de la planta.

Algo completamente opuesto ocurre con la cantidad de datos que se procesan en cada escalafón, para el nivel de campo las ordenes comunicadas desde los controladores hacia los sensores y actuadores, van en el orden de *bytes*; mientras que para el nivel de planta y

oficina se maneja una gran cantidad de información, la cual se recopila a lo largo de todos los elementos de la red y de todas las consignas planificadas en los estratos empresariales (véase Figura 1.3).



Figura 1.3: Volumen y velocidad de datos en la Pirámide CIM

Fuente: (Conectronica, 2013)

### 1.1.3 Sistemas de Control en una Red de Comunicación Industrial

Los sistemas de control que se utilizan para un determinado proceso dependen de la complejidad de la aplicación, del número de elementos que participen en la red de comunicación y de la lejanía de los procesos, por lo tanto se puede clasificar el tipo de control en:

- a. Sistema Centralizado.
- b. Sistema Distribuido.

#### a. Sistema Centralizado

Un sistema centralizado es aquel en el cual el control se realiza mediante un solo dispositivo central, en este caso todos los elementos tanto sensores como actuadores se conectan al mismo y es donde se desarrolla la lógica del sistema (Guerrero, Yuste, & Martínez, 2009). Las principales características son:

- Es muy efectivo para sistemas que no sean grandes ni complejos.
- Es sencillo de gestionar, solo hay un punto de control.
- No se presentan problemas de compatibilidad.
- Son delicados a fallos.

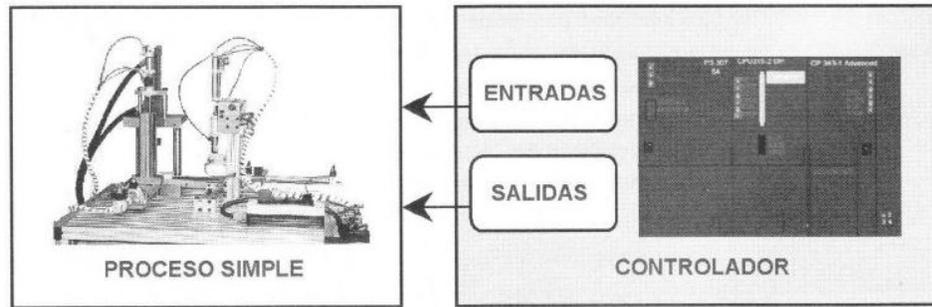


Figura 1.4: Sistema de Control Centralizado

Fuente: (Guerrero, Yuste, & Martínez, 2009)

### b. Sistema Distribuido

Un sistema distribuido es aquel en el cual, el control total del sistema se encuentra repartido en diferentes controladores conectados en red, en este caso cada procesador se encarga de la lógica necesaria para su proceso determinado y se mantiene una comunicación entre las estaciones con el fin de permitir una sincronización de las áreas (Guerrero, Yuste, & Martínez, 2009), las principales características son:

- Se emplea en sistemas grandes o complejos.
- La responsabilidad de control se reparte entre varios controladores.
- Los controladores están conectados a la red.
- Su capacidad es superior a la de un sistema centralizado.
- Es un sistema más flexible que el centralizado.
- Fácil incorporación de nuevos procesos, es decir, nuevos controladores que se agregan a la red.
- Es escalable, se parte de un sistema común y se amplía módulos o controladores según el sistema lo requiera.

- Facilita la integración de dispositivos de otros fabricantes que presenten interfaces de comunicación estándar.

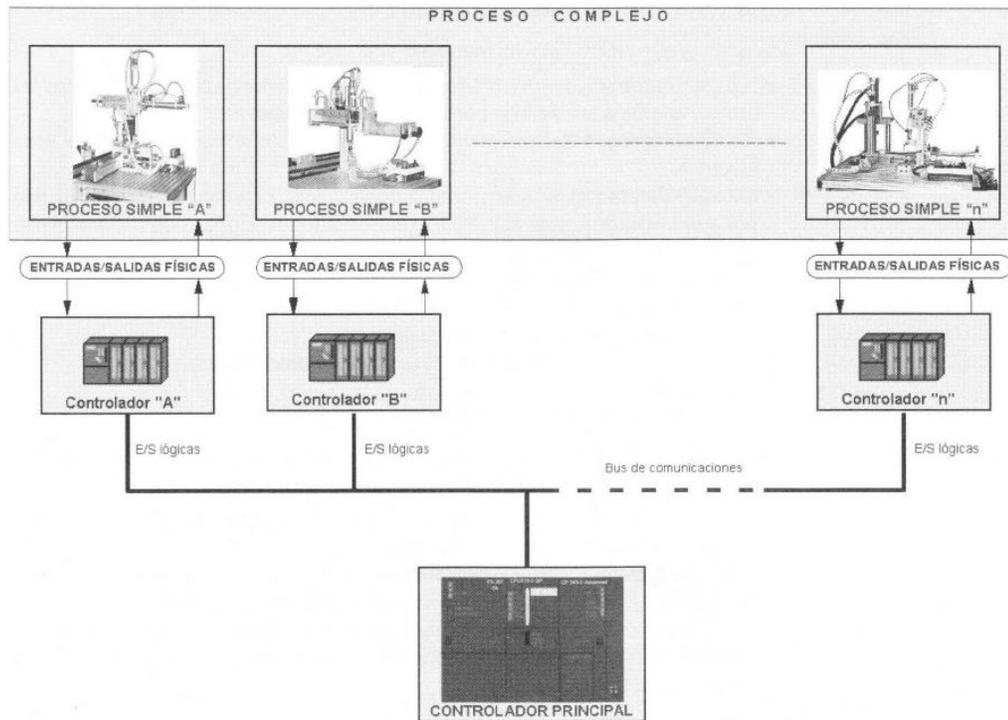


Figura 1.5: Sistema de Control Distribuido

Fuente: (Guerrero, Yuste, & Martínez, 2009)

#### 1.1.4 Topología de las Redes

La topología de las redes es la forma en la que se encuentran distribuidos y conectados los elementos del sistema, en las redes de telecomunicaciones se utiliza diferentes maneras para la interconexión de sus dispositivos, en la parte industrial se puede identificar algunas de estas topologías, según lo presentado en (Guerrero, Yuste, & Martínez, 2009) se describe las siguientes conexiones:

- Punto a punto
- Bus
- Árbol
- Anillo
- Estrella

### a. Punto a punto

Es la topología más simple ya que se trata de la interconexión de dos equipos de un extremo al otro, presenta las siguientes características:

- No son necesarias las direcciones de origen o destino.
- Puede usar sistemas *Half-Duplex*<sup>8</sup> (RS-485<sup>9</sup>) o *Full-Duplex*<sup>10</sup> (RS-422<sup>11</sup>). En el caso de Full-Duplex no es necesario el control de acceso al medio.
- Cableado sencillo sin necesidad de adaptadores de red.

Ventajas:

- Instalación sencilla.
- Fácil control para el acceso a la red.
- Si un nodo falla los demás no se ven afectados.

Inconvenientes:

- Útil para pocos nodos.
- Múltiples tarjetas o interfaces de comunicación.

Aplicaciones:

- Distancias cortas y pocos nodos

---

<sup>8</sup> *Half-Duplex*: Modo de transmisión de información bidireccional pero no simultáneo.

<sup>9</sup> RS-485: Es un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias y a través de canales ruidosos.

<sup>10</sup> *Full-Duplex*: Modo de transmisión de información bidireccional y simultánea.

<sup>11</sup> RS-422: Norma que especifica características para una transmisión digital, incluye un receptor diferencial.

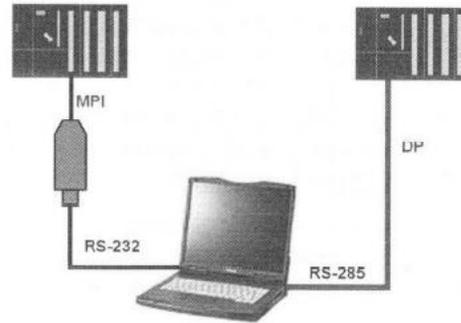


Figura 1.6: Topología de Red Punto a Punto

Fuente: (Guerrero, Yuste, & Martínez, 2009)

### b. Bus

La topología en bus consiste en una sola línea o bus donde se interconectan todos los dispositivos de la red, presenta las siguientes características:

- El bus es compartido.
- Cada nodo debe consultar si el bus está libre para poder enviar su mensaje.
- Solo un mensaje puede circular por el bus a la vez.
- Puede producirse colisiones si dos estaciones envían mensajes al mismo tiempo.

Ventajas:

- Bajo costo de instalación.
- El control de flujo no es complicado.
- El fallo de un nodo no interfiere con la comunicación del resto de puntos.
- Es posible la comunicación entre todas las estaciones.
- Fácil incorporación de nuevos nodos al bus.

Inconvenientes:

- Considerable posibilidad de colisiones en la red.
- Dependencia total del canal.
- Alta ocupación del medio si una estación establece una comunicación prolongada.

- Necesario repetidores debido a la atenuación, distancia 10km.

Aplicaciones:

- Redes Industriales

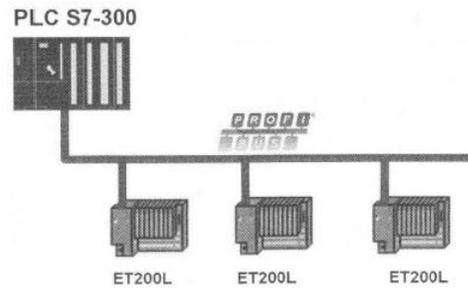


Figura 1.7: Topología de Red Bus

Fuente: (Guerrero, Yuste, & Martínez, 2009)

### c. **Árbol**

La topología en árbol consiste en un grupo de buses interconectados entre sí generando una estructura de ramas, presenta las siguientes características:

- Mayor alcance que un bus normal.
- Se aumenta la atenuación.
- Posee ramales con un bus cada uno.
- Solo un mensaje puede circular por el bus a la vez.
- Puede producirse colisiones.

Ventajas:

- Se puede tener muchos equipos conectados.
- Hace posible la conexión punto a punto.
- Facilita el crecimiento de la red.
- Se puede priorizar las comunicaciones a ciertas ramas.

Inconvenientes:

- Dependiente del cableado en rama por lo cual es sensible a fallos.
- Ciertos nodos puede quedar aislados de la red debido a una falla superior.
- Mantenimiento complicado cuando la red es muy grande.
- Más difícil de configurar.

Aplicaciones:

- Redes Industriales por zonas o departamentos independientes.

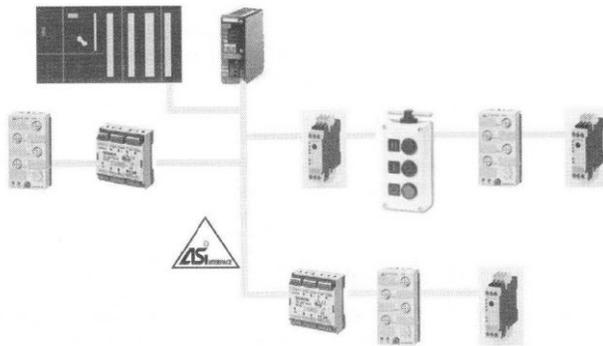


Figura 1.8: Topología de Red Árbol

Fuente: (Guerrero, Yuste, & Martínez, 2009)

#### d. Anillo

La topología en anillo es una forma derivada de la conexión en bus en la cual el canal de datos se une en ambos extremos cerrando un anillo, presenta las siguientes características:

- Los datos fluyen en un solo sentido.
- El mensaje es colocado en el anillo y circula hasta llegar al receptor.
- Podría en ciertos casos circular más de un mensaje por el canal.
- La velocidad está determinada por el equipo más lento.

Ventajas:

- No hay problemas de encaminamiento.
- Fácil incorporación de nuevos nodos introduciéndolos en el anillo.
- No existen colisiones.
- Rendimiento elevado de la red.
- No existen problemas de atenuación ya que cada nodo figura como repetidor.
- Bajo costo de implementación.

Inconvenientes:

- Un fallo o corte del anillo deja la red fuera de servicio.
- No es útil para distancias largas.
- Equipos lentos condicionan la velocidad de la red.

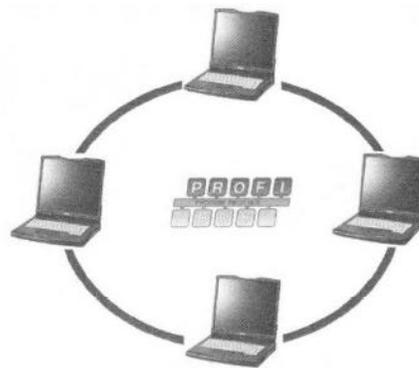


Figura 1.9: Topología de Red Anillo

Fuente: (Guerrero, Yuste, & Martínez, 2009)

### e. Estrella

La topología en estrella es aquella en la cual todos los nodos de la red se encuentran conectados a un dispositivo central que gestiona las comunicaciones, presenta las siguientes características:

- Se tiene un dispositivo central que realiza la transferencia de información.

- Todas las estaciones se cablean radialmente al concentrador.
- Los cambios de configuración se realizan en el nodo central.
- El nodo central hace como repetidor en el caso de ser un elemento activo.

Ventajas:

- Alto rendimiento, ya que los datos solo pasan a través del concentrador sin circular por varios nodos.
- Fácil incorporación o eliminación de equipos.
- Fácil conexión y mantenimiento.
- Puede manejar distintas velocidades.

Inconvenientes:

- Dependencia total del concentrador.
- Se podría producir retrasos si el concentrador no es el adecuado.
- El equipo más lento determina la velocidad de la red.
- Más costosa, se requiere más cable.

Aplicaciones:

- Redes LAN, *Ethernet*, *Fast Ethernet*.

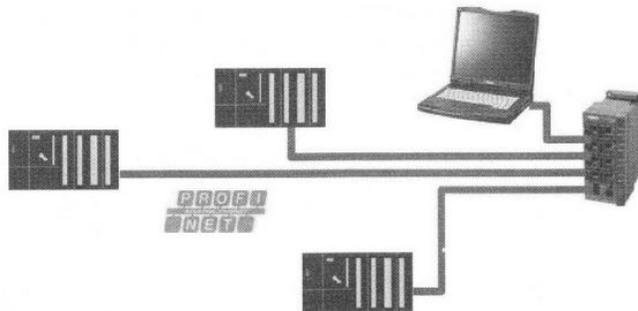


Figura 1.10: Topología de Red Estrella

Fuente: (Guerrero, Yuste, & Martínez, 2009)

De todas las topologías mencionadas se puede destacar principalmente tres, que son las más utilizadas en el campo industrial. La primera correspondiente a la topología en bus, la misma que se utiliza para la interconexión de equipos en redes RS-485 o RS-232 con el fin de conectar elementos de campo. La segunda es la topología en estrella que permite comunicar diferentes elementos en redes basadas en *Ethernet* y se ocupa para la comunicación entre controladores y PCs. La tercera es la conexión en anillo implementada principalmente en redes de fibra óptica.

### **1.1.5 Buses de Campo**

Un bus de campo es el sistema de transmisión de información de uso industrial que permite la instalación y operación de dispositivos o máquinas involucrados en un proceso de producción. El objetivo de un bus es interconectar los elementos de campo con el equipo controlador mediante el tradicional bucle de corriente de 4-20 mA (Kaschel & Pinto, 2010).

Los buses de campo representan redes digitales, bidireccionales, de conexión multipunto, montadas sobre un bus serie. Se despliegan para la conexión de sensores, actuadores, transductores, PLCs que permiten autodiagnóstico, control, mantenimiento, comunicación bidireccional, etc. considerando a los dispositivos de campo como elementos con cierta “inteligencia” dentro de la red (Blanco, 2006).

#### **Ventajas:**

- Reducción en el costo de cableado necesario para el despliegue de la red ya que cada célula puede ser conectada mediante un solo cable para sus nodos.
- Siendo los buses de campo más sencillos que otras redes, reduce la necesidad de mantenimiento, incluyen funciones de monitorización por lo cual la detección de fallos resulta más sencilla.

- Los buses de campo han generado mayor flexibilidad al usuario ya que al conectar dispositivos con cierta “inteligencia”, han facilitado varios algoritmos de control que antes debían ser programados en el dispositivo central.
- Permite comunicación entre dispositivos de campo, es decir, no solo con el controlador.
- Convierte más ágil a la comunicación al reducir las capas del Modelo OSI<sup>12</sup>, implementando solamente tres: Física, Enlace y Aplicación.

#### **1.1.5.1 Clasificación de los Buses de Campo**

Existen varios buses de campos desarrollados por diferentes compañías enfocados a problemas específicos de velocidad o prestaciones, a continuación según lo presentado por (Blanco, 2006) y (Kaschel & Pinto, 2010) se los puede clasificar de la siguiente manera.

##### **a. Buses de alta velocidad y baja funcionalidad**

Estos buses se encuentran enfocados a la integración de elementos sencillos como: fin carreras, relés, actuadores, fotocélulas. Se aplican en zonas específicas de una planta o máquinas que funcionan en tiempo real, suelen implementar las capas físicas y de enlace del Modelo OSI, algunos son:

- CAN: Originalmente para vehículos.
- SDS: Integra sensores y actuadores, basado en CAN.
- AS-I: Diseñado por SIEMENS, bus serie para integrar sensores y actuadores.

##### **b. Buses de alta velocidad y funcionalidad media**

En estos buses se encuentra más desarrollada la capa de enlace por lo cual permite el envío de paquetes de datos de tamaño medio, en los cuales es posible incluir aspectos como

---

<sup>12</sup> Modelo OSI (*Open System Interconnection*, “Sistema de Interconexión Abierta”): Modelo de referencia para los protocolos de red el cual especifica 7 capas en su arquitectura.

configuración y calibración del dispositivo. Este bus cuenta con mayor funcionalidad y permite la gestión de elementos de campo complejos, algunos incluyen perfiles de configuración para facilitar la integración con otras marcas y también servicios de configuración desde programas basados en ordenadores, algunos ejemplos son:

- DeviceNet
- BitBus
- DIN MessBuss
- InterBus
- Modbus

### c. Buses de alta prestación

Estos buses están diseñados para trabajar en todos los niveles de automatización, se basan en buses de alta velocidad, la integración tanto de funcionalidad como seguridad se han convertido en un verdadero desafío para algunos de ellos. La capa de aplicación cuenta con varios servicios como: redes multi-maestro, maestro-esclavo<sup>13</sup>, direccionamiento *unicast*<sup>14</sup>, *multicast*<sup>15</sup>, *broadcast*<sup>16</sup>, comunicación de variables, seguridad en la red, autenticación, peticiones a esclavos, etc. Algunos ejemplos son:

- Profibus
- FIP
- Fiedlbus Foundation
- Modbus

### d. Buses para áreas de seguridad intrínseca

Estos buses presentan ciertas modificaciones en la capa física con el fin de cumplir con las regulaciones de seguridad intrínseca en ambientes explosivos, dicha seguridad hace

---

<sup>13</sup> Maestro-Esclavo: Modelo de acceso al medio utilizado para controlar comunicaciones entre un dispositivo maestro hacia otros elementos esclavos.

<sup>14</sup> *Unicast*: Transmisión de información entre un emisor y un receptor.

<sup>15</sup> *Multicast*: Transmisión de información entre un emisor a varios receptores de la red.

<sup>16</sup> *Broadcast*: Transmisión de información entre un emisor hacia todos los receptores de la red.

referencia a la protección necesaria para que el dispositivo no tenga posibilidad de generar una explosión, algunos son:

- HART
- Profibus-PA
- FIP

### 1.1.5.2 Principales Buses de Campo

A lo largo de los años se han desarrollado diferentes buses para la industria según los requerimientos del momento, en la actualidad se han tratado de estandarizar algunos sin grandes resultados, es así que según lo referido por (Guerrero, Yuste, & Martínez, 2009); (Salazar & Correa, 2011); (Kaschel & Pinto, 2010); (Blanco, 2006) se puede presentar.

#### a. AS-I (*Actuador Sensor Interface*, “Interfaz Sensor-Actuador”)

AS-I es un bus de campo desarrollado por SIEMENS en el año 1994 destinado para la conexión de actuadores y sensores binarios, hoy acogido por el estándar IEC<sup>17</sup> TG 17B. Este permite la conexión de hasta 32 esclavos en diferentes topologías. El medio físico consiste en un solo cable sin apantallamiento que facilita la alimentación y transmisión de datos, la red puede controlar un número máximo de 124 I/O digitales.

#### b. Interbus

Inicialmente desarrollado por la empresa *Phoenix Contact GmbH*, luego normalizado bajo DIN<sup>18</sup> 19258 y EN<sup>19</sup> 50254, es un bus de campo para la interconexión de sensores y actuadores, este estándar está basado en un modelo maestro-esclavo, siendo el maestro el elemento que actúa con los dispositivos de escalafones superiores, su estructura de red es

---

<sup>17</sup> IEC (*International Electrotechnical Commission*, “Comisión Internacional Electrotécnica”): Organización de normalización en campos eléctricos, electrónicos y tecnologías relacionadas.

<sup>18</sup> DIN (*Deutsches Institut für Normung*, “Instituto Alemán de Normalización”): Instituto alemán de normalización.

<sup>19</sup> EN (Estándares Europeos): Comité para la aprobación de normas europeas.

principalmente en anillo. Interbus utiliza generalmente la interfaz física RS-485 y tiene un número máximo de 512 estaciones.

**c. CAN (*Controller Area Networking*, “Controlador de Área de Red”)**

Desarrollado por la empresa Bosch en 1986, pensado para el uso dentro del área automovilística con el fin de reducir el cableado, hoy estandarizado bajo la norma ISO<sup>20</sup> 11898-1. Se usa para la transmisión de información entre dispositivos inteligentes de todo tipo, siendo un bus multi-maestro, cuenta con velocidades desde 50 Kbps hasta 1 Mbps. CAN define las capas física y de enlace del Modelo OSI. Las estaciones en CAN son capaces de comprobar los mensajes que son difundidos por la red mediante dos clases de identificadores.

**d. Profibus**

Profibus es un bus de campo desarrollado a finales de la década de los ochenta por un grupo de empresas e institutos alemanes, normalizado en Europa por EN 50170. Este bus posee su capa física basada en par trenzado con blindaje o fibra óptica. Este estándar es usado para la interconexión de dispositivos digitales de campo como: transmisores, PLCs, sensores, actuadores, controladores numéricos, computadores, interfaces HMI, etc. Es un bus de campo que está diseñado de una manera completamente enfocada hacia las aplicaciones industriales de control y automatización. Es uno de los buses con mayor implementación en el mundo, utiliza la configuración maestro-esclavo.

**e. DeviceNet**

*DeviceNet* es un bus basado en CAN, la norma ISO 11898 especifica la capa física y de enlace del mismo. Fue desarrollado por Allen-Bradley en 1994 como un estándar avanzado de CAN, posteriormente se tornó de tecnología abierta.

---

<sup>20</sup> ISO (*International Standard Organization*, “Organización Internacional de Normalización”): Organización para la formulación de normas en varios sectores.

*DeviceNet* ha determinado conexiones y mensajes para el funcionamiento maestro-esclavo, permite la conexión de 64 nodos a una distancia entre 100 a 500 m manteniendo velocidades comprendidas entre 125 Kbps a 500 Kbps.

#### **f. Modbus**

Modbus es un bus de campo que permite una comunicación industrial abierta, desarrollado por Modicon en 1979 para la monitorización de dispositivos y la transmisión de señales. La comunicación es maestro-esclavo y también mediante el modelo Cliente-Servidor. Modbus utiliza comúnmente la interfaz RS-232<sup>21</sup>, RS-485 o fibra óptica. La comunicación es realizada punto a punto entre el emisor y el receptor o mediante difusión a todos los receptores (*broadcast*).

Modbus usa varios protocolos para la transmisión de datos entre los cuales se puede destacar TCP/IP<sup>22</sup>, de tal manera que Modbus-TCP está diseñado para permitir a equipos industriales tales como: PLCs, computadores, *drivers* para motores y otros tipos de dispositivos físicos de entrada-salida comunicarse sobre una red. Específicamente el protocolo define el uso de mensajes Modbus en un entorno *intranet*<sup>23</sup> o *internet*.

#### **g. Industrial Ethernet**

*Ethernet* es hoy en día el líder en la conexión de redes LAN, mediante sus estándares IEEE 802.3 (*Ethernet*) y 802.11 (*Wireless LAN*, "LAN Inalámbrico"), su uso ha evolucionado en las últimas décadas para la aplicación en ambientes industriales ya que buses como Profibus y Modbus utilizan este protocolo para la conexión en niveles superiores. *Ethernet* ha desarrollado sus topologías y su intrínseco indeterminismo<sup>24</sup>, así como sus normas para

---

<sup>21</sup> RS-232: es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un "Equipo Terminal de Datos" y un "Equipo de Comunicación de Datos".

<sup>22</sup> TCP/IP: Es un conjunto de protocolos de red en los que se basa *Internet* y que permiten la transmisión de datos entre computadoras.

<sup>23</sup> Intranet: Red informática para transmitir información dentro de una organización.

<sup>24</sup> Indeterminismo: Capacidad de un sistema en el cual los acontecimientos no se presentan en forma causal lineal.

conectores, cables, etc. para el campo industrial. *Ethernet* Industrial ofrece múltiples posibilidades para la *intranet*, *extranet*<sup>25</sup> e *internet*.

## h. HART

El protocolo de comunicación HART (*Highway Addressable Remote Transducer*, “Transductor Remoto que se puede Direccionar”) fue desarrollado por Rosemount Inc. a mediados de 1980 para sus instrumentos de campo inteligentes, se basó en el estándar de comunicación telefónica Bell 202. En 1986 pasó a ser un protocolo abierto, es compatible con la transmisión analógica en corriente 4-20mA que permite que la señal analógica y las señales de comunicación digitales sean transmitidas simultáneamente sobre el mismo cableado. Su velocidad es de 1.2 Kbps para distancias de hasta 3 km.

### 1.1.5.3 Comparativa

Las distintas tecnologías desarrolladas para los buses de campo permiten su aplicación en diferentes niveles de la pirámide CIM o en distintos casos de automatización, a continuación según lo expuesto por (Blanco, 2006) se presentan una tabla comparativas de algunos buses de campo desde un punto de vista general.

Tabla 1.1: Comparativa de algunos Buses de Campo

GENERAL	AS-Interface	Interbus	Profibus	DeviceNet	Modbus-1	Ethernet
<b>Disponible desde</b>	1995	1988	1990	1995	1979	1975
<b>Fabricante</b>	AS-Interface Consortium (Germany)	Phoenix (Germany)	Profibus Consortium (Germany)	Omron, Rockwell	Modicon /Gould /Groupe Schneider	Xeron (US)
<b>Estándar</b>	EN 50295, IEC 620262, IEC 947	DIN 19258, EN 502541, IEC 61158 Type 8	DIN 19245, EN 13321/1 (FMS), EN 502542, EN 501702, IEC 61158 Type 3, SEMI E54.8 (DP)	ISO 11898	No international standard.	IEEE 802.3
<b>Website</b>	<a href="http://www.as-interface.net">www.as-interface.net</a>	<a href="http://www.interbusclub.com">www.interbusclub.com</a>	<a href="http://www.profibus.com">www.profibus.com</a>	<a href="http://www.odva.org">www.odva.org</a>	<a href="http://www.modbus.org">www.modbus.org</a>	

<sup>25</sup> *Extranet*: Red privada que para transmitir datos entre dos organizaciones.

<b>Variantes</b>	V1.0, V2.0, V2.10, V2.11	V1, V2, V3, V4, InterbuLoop	FMS, PD, PA		ASCII, RTU (Remote Terminal Unit)	10BaseT 100BaseTX
<b>Aplicable para E/S sensores / actuadores</b>	Sí (especialmente dedicado para ello)	Sí	No (demasiado complejo y hardware excesivo)	Sí	No	<i>Hardware Excesivo</i>
<b>¿Aplicable a E/S remotas?</b>	Limitado a 4 E/S digitales o 2 analógicas	Sí	Sí	Sí	Sí	Sí
<b>Aplicable para comunicación entre controladores o equipos inteligentes</b>	No	Limitado	Sí	Sí	Sí	Sí
<b>Variante más empleada</b>	V2.0	V4	DP/V1		RTU	10BaseT, 100BaseTX
<b>Áreas de aplicación</b>	Industria discreta (DP), Industria de proceso (PA) parcialmente.	Industria discreta	Industria discreta (DP), Industria de proceso (PA).	Industria discreta (DP), Industria de proceso (PA).	Industria discreta (DP), Industria de proceso (PA).	Niveles medio y alto de automatización industrial
<b>Competidor más importante</b>	Ninguno	Profibus	Interbus, CAN, Foundation Fieldbus, DeviceNet	Profibus, Modbus	Profibus, Devicenet	Ninguno
<b>Interfaz para PC</b>	Sí, varios vendedores	Sí, vía Phoenix	Sí, varios vendedores	Sí	Sólo para RS 422/485	Incluida en cualquier PC

Fuente: (Blanco, 2006)

Cada bus de campo se ha desarrollado en periodos específicos para dar solución a problemas de conectividad existentes entre los elementos de un sistema de automatización, a continuación se presentan una tabla resumen con las ventajas y desventajas más relevantes de cada estándar.

Tabla 1.2: Ventajas y desventajas de los principales buses de campo

<b>Bus</b>	<b>Criterio</b>	<b>Descripción</b>
AS-I	Ventajas	<ul style="list-style-type: none"> <li>- A nivel físico puede adoptar cualquier topología.</li> <li>- Un solo cable para transmisión de datos y alimentación.</li> <li>- Alta velocidad de respuesta, 150µs para un ciclo de pregunta-respuesta.</li> <li>- Montaje y puesta en marcha sencilla.</li> </ul>

	Desventajas	<ul style="list-style-type: none"> <li>- Alcance máximo con repetidor es 300m.</li> <li>- Mayor uso solo para sensores y actuadores digitales.</li> </ul>
Interbus	Ventajas	<ul style="list-style-type: none"> <li>- Puede alcanzar mayor distancia hasta 13km con repetidores.</li> <li>- Añade servicios más complejos para la comunicación con dispositivos inteligentes con servidores basados en PMS.</li> <li>- Se puede conectar una gran cantidad de elementos, 512 estaciones.</li> <li>- Protocolo determinista y además los datos de todos los esclavos son insertados en el bus simultáneamente.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- Fallas en la topología anillo.</li> <li>- Cable de 5 hilos es una desventaja en costos.</li> </ul>
CAN	Ventajas	<ul style="list-style-type: none"> <li>- Uso de tensiones diferenciales permite su uso en ambientes ruidosos.</li> <li>- Usa una estructura de identificares de mensaje para el tipo y prioridad, mayor inteligencia de la red.</li> <li>- Permite detección y señalización de errores.</li> <li>- Sistema multi-maestro.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- Disminuye el aprovechamiento de la red al usar la difusión de mensajes como medio de comunicación.</li> <li>- Conexión de pocos nodos, 32.</li> </ul>
Profibus	Ventajas	<ul style="list-style-type: none"> <li>- Bus para conexión de elementos de campo y controladores.</li> <li>- Mayor velocidad que otros buses basados en RS-485, 12Mbit/s.</li> <li>- Sistema multi-maestro.</li> <li>- Útil para ambientes con seguridad intrínseca con su variante Profibus-PA.</li> <li>- Fácil implementación.</li> <li>- Posibilidad de utilizar fibra óptica como medio de tx.</li> <li>- Red más ampliamente usada en el mundo.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- No posee servicios complejos para la integración con sistemas informáticos como si lo hace <i>Industrial Ethernet</i>.</li> <li>- Se necesita pasarelas para su integración a otras redes industriales.</li> </ul>
DeviceNet	Ventajas	<ul style="list-style-type: none"> <li>- Modelo basado en productor-consumidor con lo cual permite el uso de: maestro-esclavo, multi-maestro, igual a igual.</li> <li>- Mas nodos que en CAN tradicional, en este caso 64.</li> <li>- Protocolo cíclico, permite acción en tiempos conocidos.</li> <li>- Uso de diferentes topologías.</li> <li>- Señal y alimentación a través del mismo cable.</li> <li>- Alto uso en el mundo, soporte de empresas como ABB, Allen-Bradley, Festo, Omron, etc.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- No es posible su uso en ambientes de seguridad intrínseca.</li> <li>- No esta tan posicionado como Profibus o Hart.</li> <li>- Se requiere inversión para elementos específicos con el estándar.</li> </ul>
Modbus	Ventajas	<ul style="list-style-type: none"> <li>- Es posible su uso en varios de los niveles de la pirámide CIM, campo y nivel de gestión a través de sus variantes.</li> <li>- Define un modelo de protocolo independiente del medio de transmisión.</li> <li>- Protocolo ampliamente soportado y usado en equipos de la industria.</li> </ul>

		<ul style="list-style-type: none"> <li>- En su variante RTU y ASCII se puede conectar hasta 247 estaciones.</li> <li>- A través de los códigos de función, se puede intervenir directamente en los esclavos.</li> <li>- Posibilidad de integración a redes <i>Ethernet</i>.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- Limitaciones en cuanto al diagnóstico, respecto a protocolos como Profinet.</li> <li>- Sistema con un solo maestro (RTU y ASCII).</li> <li>- Menor velocidad en RTU y ASCII, hasta 38.4 Kbits/s.</li> </ul>
<i>Industrial Ethernet</i>	Ventajas	<ul style="list-style-type: none"> <li>- Basado en <i>Ethernet</i>, es un protocolo con una gran penetración en el mundo que ha desarrollado su estándar industrial.</li> <li>- Velocidad elevada, 100Mbtis/s.</li> <li>- Incluye servicios orientados a intercambio de información de tamaño medio o elevado.</li> <li>- Ofrece solución que es integrable con varios fabricantes.</li> <li>- Es posible el uso de fibra óptica.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- Adecuación de equipos <i>Ethernet</i> para el entorno industrial.</li> <li>- No es un bus que incluya alimentación para los dispositivos.</li> <li>- Cableado más costoso.</li> </ul>
Hart	Ventajas	<ul style="list-style-type: none"> <li>- Uso más eficiente al sobreponer una señal digital a onda de 4-20mA, de esta manera se transmite información adicional.</li> <li>- Puede existir dos equipos maestros.</li> <li>- Menor coste de instalación de cableado.</li> <li>- Las dos señales no se molestan entre si ya que se modula la información.</li> <li>- Varios dispositivos incluyen este protocolo.</li> </ul>
	Desventajas	<ul style="list-style-type: none"> <li>- Máximo número de equipos conectados es 15.</li> <li>- Velocidad limitada, 1200 baudios/s.</li> </ul>

Fuente: (Guerrero, Yuste, & Martínez, 2009); (Salazar & Correa, 2011); (Kaschel & Pinto, 2010); (Blanco, 2006).

## 1.2 PROFIBUS

Profibus (*PROcess FIeld BUS*, “Bus de Procesos de Campo”) es un conjunto de especificaciones de red que ha dado lugar a un bus de campo de tecnología abierta y transparente para la comunicación digital de dispositivos dentro de los procesos industriales, tanto de fabricación como procesos distribuidos. La idea de crear un estándar para la comunicación nace en 1987 por parte del Ministerio Federal Alemán de Investigación y Desarrollo, que financió un proyecto conformado por 13 empresas y 5 institutos de investigación, que dio lugar a un bus de campo basado en el Modelo OSI (Guerrero, Yuste, & Martínez, 2009).

Profibus ha sido estandarizado bajo la norma europea EN 50170 y EN 50250 y bajo las normas internacionales IEC 61158 e IEC 61158-2 con lo cual se introduce en el mercado industrial independientemente del fabricante (Mandado et al., 2006).

Profibus es un estándar desarrollado para dar solución a los problemas de conectividad de instrumentos de campo como: periferia descentralizada, válvulas, *drivers* de motores o accionamientos en general, etc. con los sistemas de automatización en sus diferentes variantes tanto basados en controladores como en *PCs*. Está diseñado para cubrir las necesidades de automatización y producción con un intercambio de datos rápido y fiable, además puede ser empleado en ambientes explosivos gracias a su variante Profibus PA, también se usa para aplicaciones de seguridad y con dispositivos HART (SIEMENS, Comunicación con SIMATIC, 2006). Profibus posee tres variantes del estándar las cuales son: Profibus-FMS, Profibus-DP y Profibus-PA enfocadas a distintas áreas de la pirámide CIM (véase Figura 1.11)

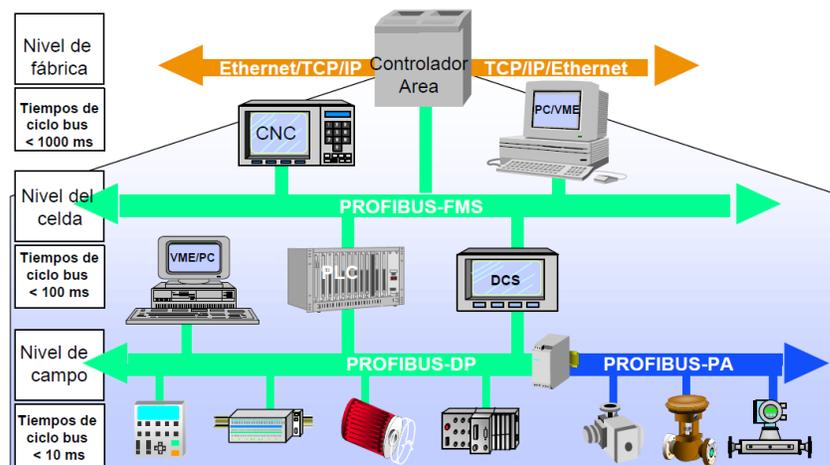


Figura 1.11: Red Profibus

Fuente: (López, 2008)

La tecnología desarrollada en Profibus permite su aplicación en varios de los requerimientos de la industria, ya que puede ser utilizado tanto para aplicaciones en las

cuales el tiempo de respuesta sea un parámetro crítico, como también en aplicaciones que requieran una extensa y compleja red de comunicación (Center PROFIBUS Internacional Support, 2002).

### 1.2.1 Característica Generales

Las redes Profibus fueron desarrolladas como un estándar de comunicación pensado para la industria, está basado en el Modelo OSI aunque implementa solo tres capas del mismo con el fin de facilitar la comunicación, las capas sobre las cuales se desarrolla Profibus son las capas física y de enlace en donde se comparte las principales características para los tres miembros, y la capa de aplicación que se diferencia para cada caso según los servicios que se presten (véase Figura 1.12).

- a. **Capa Física:** En la capa física PHY (*PHYsical layer*, “Capa Física”) se establece las características de las señales que se transmiten y los dispositivos que se involucran para dicho envío de datos, las señales pueden ser eléctricas según la norma EIA<sup>26</sup> RS-485, o IEC 1158-2 con seguridad intrínseca, u ópticas (Mandado et al., 2006).
- b. **Capa de Enlace de datos:** En la capa de enlace de datos FDL (*Field Data Link Layer*, “Capa de Enlace de Datos”) se establece la manera de comunicación de los dispositivos controladores entre si y también con los elementos de campo, para ello se ocupa básicamente el mecanismo de acceso llamado principal-subordinado o maestro-esclavo (Mandado et al., 2006).
- c. **Capa de Aplicación:** Profibus es un estándar que ha eliminado las capas de transporte, red, sesión y presentación al estar estas más enfocadas a redes de datos, más bien los servicios de Profibus son manejados por la capa de aplicación (Mandado et al., 2006).

---

<sup>26</sup> EIA (*Electronic Industries Alliance*, “Alianza de Industrias Electrónicas”): Organización estadounidense que promueve el desarrollo del mercado y competitividad de los sectores industriales de alta tecnología.

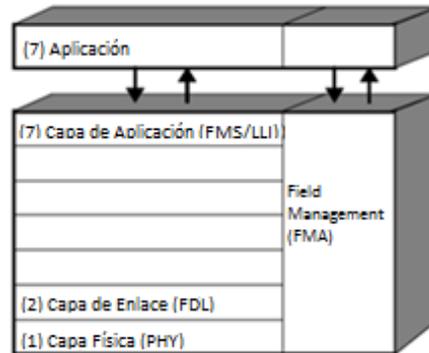


Figura 1.12: Capas Implementadas en Profibus

Fuente: (López, 2008)

### 1.2.1.1 Arquitectura

La arquitectura de protocolos que utiliza Profibus está basada en el Modelo OSI según el estándar ISO 7498, las capas que implementa son: la física, de enlace y la de aplicación. La capa física posee características similares para Profibus-FMS y Profibus-DP pudiendo utilizar los medios de transmisión explicados en la sección 1.2.1.3, mientras que para Profibus-PA se maneja las especificaciones de capa para ambientes con seguridad intrínseca (López, 2008).

La capa de enlace que es la encargada de manejar las tareas de acceso al medio, tiempo de rotación de testigo, servicios de transferencia de datos tanto cíclicos como a cíclicos posee las mismas características para los tres perfiles, excepto por los servicios de IEC *interface* necesarios para el acople de dispositivos PA.

La capa de aplicación es la diferencia principal entre los tres perfiles desde el punto de vista de la comunicación, ya que los servicios que presenta cada estándar están enfocados para tareas o dispositivos específicos dentro de la pirámide CIM, si bien es cierto las funciones básicas DP se comparten para Profibus-DP y Profibus-PA cada una de estos cuenta con sus propias características para la comunicación con dispositivos de campo. Por otra parte Profibus-FMS cuenta con una mayor cantidad de servicios de perfil

necesarios para la comunicación con dispositivos electrónicos de mayor procesamiento y para la transmisión de grandes paquetes de datos (véase Figura 1.13).

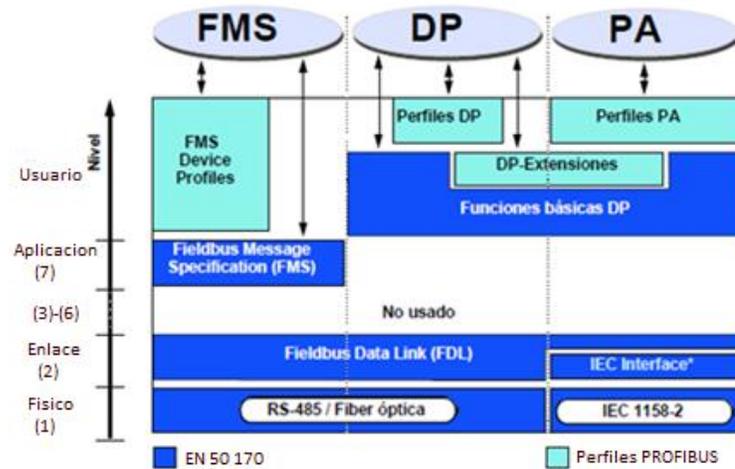


Figura 1.13: Perfiles Profibus

Fuente: (López, 2008)

### 1.2.1.2 Control de Acceso al medio

Las redes Profibus son de tipo principal-subordinado o maestro-esclavo, es así que se puede diferenciar dos tipos de estaciones de comunicación con diferentes funciones e importancia desde el punto de vista de la transmisión de información, según lo expuesto por (Mandado et al., 2006) se puede clasificar en:

- **Procesadores Principales:** Estas estaciones son conocidas como activas y son capaces de gestionar las comunicaciones, es decir, enviar y recibir datos por iniciativa propia hacia los demás elementos.
- **Procesadores Subordinados:** Estas estaciones son conocidas como pasivas ya que solo pueden transmitir sus datos si el procesador principal lo autorizada, mientras se encuentre este activo.

Los dispositivos que cuentan con una mayor capacidad de procesamiento de datos son los que hacen el papel de maestros de la red, por ejemplo: PLCs, sistemas de control, robots, etc. mientras que los dispositivos de campo que no poseen la “inteligencia” necesaria por ser más sencillos son los esclavos, por ejemplo: sensores, actuadores, convertidores de frecuencia, *encoders*<sup>27</sup>, etc. (Mandado et al., 2006).

Si en la red Profibus existe más de un maestro, para evitar que dos procesadores tomen el control se utiliza un mecanismo de acceso al medio llamado *token bus* (véase Figura 1.14). Las comunicaciones son gestionadas por un mensaje llamado *token* o testigo que circula entre los dispositivos maestros de la red y mediante el cual se da aviso a cada controlador que puede efectuar sus comunicaciones durante un tiempo determinado.

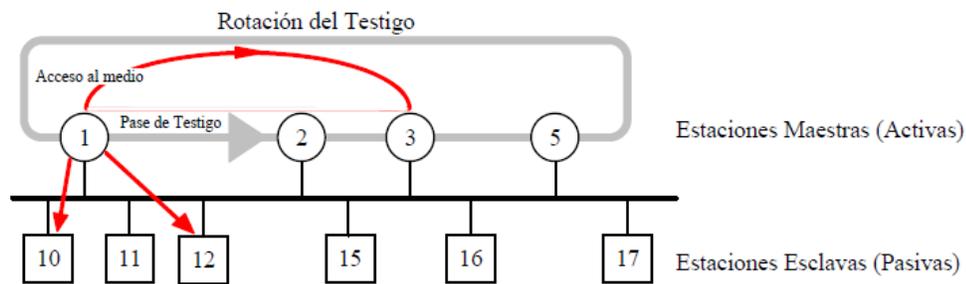


Figura 1.14: Mecanismo de control de Acceso al Medio (token bus)

Fuente: (López, 2008)

La circulación del testigo se realiza mediante las direcciones de cada maestro en orden ascendente, donde cada procesador conoce la dirección del siguiente dispositivo. Cada procesador principal mide constantemente los tiempos en la red y calcula el tiempo que puede estar activo según el intervalo máximo establecido para las comunicaciones en toda la red (Mandado et al., 2006).

<sup>27</sup> *Encoder*: Elemento utilizado para convertir la posición angular de un eje a un código digital para medir posicionamiento y velocidad.

### 1.2.1.3 Trama de Datos

Los caracteres (8 bits) se agrupan en forma de paquetes para constituir mensajes que serán enviados por la red, dentro de estas estructuras se agregan campos especiales que permiten controlar el destino e información. (Mandado, Marcos, Pérez, Fernández, & Armesto, 2006). Los campos que se tienen son los siguientes:

- Carácter DA (*Destination Address*, “Dirección de Destino”).
- Carácter SA (*Source Address*, “Dirección del Emisor”).
- Carácter FC (*Frame Control*, “Control de Mensaje”).
- Estructura FCS (*Frame Check Structure*, “Estructura de Comprobación de Mensaje”).

La trama de datos admite 3 tipos de formatos: de longitud fija sin datos, de longitud fija con datos y de longitud variable, según lo presentado por (Industriales, S.A.) y (Mandado, Marcos, Pérez, Fernández, & Armesto, 2006) se puede resumir lo siguiente en cada caso.

- **Mensaje sin datos:** Es el mensaje más corto ya que solamente contiene los caracteres de control de trama, no posee un campo para enviar datos.
- **Mensaje con datos de longitud fija:** Este mensaje permite el envío solamente de 8 caracteres o bytes de datos comprendido entre los campos FC y FCS.
- **Mensaje con datos de longitud variable:** Este mensaje posibilita el envío de datos con una longitud de información variable entre 1 y 249 bytes, adicionalmente posee los campos LE y LEr (*Length byte*, “Longitud de byte”) a través de los cuales se especifica el número de caracteres.

En la Figura 1.15, se muestra los campos de los tres formatos de mensaje.

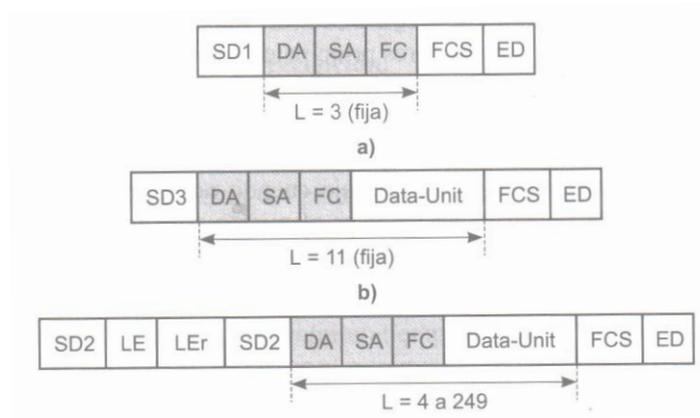


Figura 1.15: Trama de Datos Profibus

Fuente: (Mandado, Marcos, Pérez, Fernández, & Armesto, 2006)

#### 1.2.1.4 Medios Físicos

Las redes Profibus básicamente utilizan dos medios de transmisión para sus comunicaciones: mediante señales eléctricas a través de un cable de par trenzado según la norma IEA RS-485 o también mediante señales ópticas a través de cables de fibra óptica. Según lo presentado por (Mandado et al., 2006) se puede destacar las siguientes características.

##### a. Comunicación Mediante Señales Eléctricas RS-485

- **Topología:** La topología más usada con este medio es la de bus que permite la conexión de los procesadores maestros y esclavos, posee resistencias terminadoras en ambos extremos con el fin de minimizar las reflexiones. También se puede ocupar una topología en árbol conectando varios buses mediante repetidores.
- **Medio Físico:** Se utiliza un cable par trenzado apantallado el cual puede contar con varios tipos de recubrimiento según la zona de trabajo (nivel de perturbaciones). El blindaje en redes de gran distancia y velocidad debe estar unido

a lo largo de todo el sistema con una sola conexión de referencia a tierra, con el fin de evitar problemas de potencial en diferentes puntos del acople.

- **Velocidad de Transmisión:** En un bus lineal se puede alcanzar velocidades de transmisión entre 9.6 Kbits/s y 12 Mbits/s en función de la distancia máxima de conexión. Para redes en árbol con el uso de repetidores la velocidad es de 1.5 Mbits/s.
- **Distancia de Transmisión:** Un bus lineal sin repetidores alcanza una distancia de 100 m a 12 Mbits/s y de 1200 m a 93.75 Kbits/s.
- **Número de Procesadores de Comunicaciones:** El número máximo de estaciones conectadas a un maestro en bus lineal es 32, con repetidores se puede conectar hasta 127 procesadores de comunicación.



Figura 1.16: Cable y Conector utilizado en RS-485

Fuente: (López, 2008)

## b. Comunicación Mediante Señales Ópticas

- **Topología:** La tecnología de transmisión óptica establece redes de comunicación en anillo o estrella redundantes o no redundantes.
- **Medio Físico:** Se utiliza tanto cables de fibra óptica de plástico con señales cuya longitud de onda esta por los 660 nanómetros, o también se puede emplear fibra óptica de cristal con señales cuya longitud de onda está comprendida entre 800 y 1500 nanómetros.

- **Velocidad de Transmisión:** Se puede lograr velocidades de transmisión entre 9.6 Kbits/s y 12 Mbits/s.
- **Distancia de Transmisión:** Se puede manejar distancias de hasta 15 km.
- **Número de Procesadores de Comunicaciones:** El número máximo de estaciones conectadas es 127, generalmente se combina la transmisión por fibra y cobre para la interconexión de plantas.



Figura 1.17: Fibra Óptica de plástico y cristal

Fuente: (López, 2008)

En la siguiente tabla se expone las características de los medios de transmisión usados en Profibus.

Tabla 1.3: Características de los Medios de Transmisión Profibus

Procedimiento	MBP	RS 485	RS 485-IS	Fibra óptica
Transferencia de datos	Síncrono	Señales de diferencia de potencial	Señales de diferencia de potencial	Visuales
Velocidad de transmisión	31,25 kbits/s invariables	9,6 kbits/s a 12 Mbits/s	9,6 kbits/s a 1,5 Mbits/s	9,6 kbits/s a 12 Mbits/s
Cable	Cable de 2 hilos trenzado y apantallado (cobre)	Cable de 2 hilos trenzado y apantallado (cobre)	Cable de 4 hilos trenzado y apantallado (cobre)	Fibra óptica, plástico, PDF
Topología	Línea, árbol	Línea, árbol	Línea	Estrella, anillo, línea
Seguridad intrínseca	EEx ia/ib		EEx ib	

Fuente: (SIEMENS, Comunicación con SIMATIC, 2006)

### 1.2.2 Profibus-DP

Profibus-DP (Periferia Descentralizada) es un perfil de comunicación optimizado para su uso a nivel de campo, es decir, está desarrollado para la transferencia de información entre dispositivos de proceso a altas velocidades y bajo costo. Profibus-DP es una red que sustituye a los antiguos bucles de corriente de 0-20 mA (Mandado et al., 2006); (Blanco, 2006).

Una red Profibus-DP está diseñada para conectar básicamente tres tipos de sistemas electrónicos (véase Figura 1.18) como son:

- Sistemas de Control: Sistemas de control numérico, sistemas de control de movimiento, PLCs, computadoras industriales, etc.
- Sistemas Especializados: Unidades de programación, paneles de HMI, paneles de configuración de dispositivos, etc.
- Sensores y Actuadores.

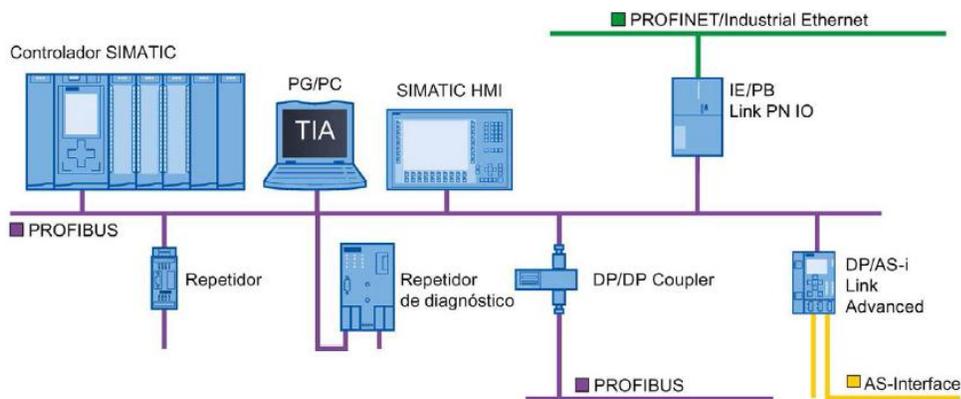


Figura 1.18: Conexiones en Profibus-DP

Fuente: (SIEMENS, PROFIBUS con STEP 7 V13, 2014)

Profibus-DP se caracteriza por ser un perfil que maneja las comunicaciones de manera determinística<sup>28</sup>, ya que al ser implementado en las áreas de proceso de las fábricas permite una alta velocidad de transmisión en tiempos conocidos (aproximadamente 10 ms) con volúmenes de datos no elevados (Guerrero, Yuste, & Martínez, 2009).

Como se explicó en el apartado 1.2.1.2, en las redes Profibus se maneja el control de acceso al medio a través del modelo maestro-esclavo, por lo cual Profibus-DP al estar orientado a comunicaciones de campo presenta tres tipos de estructuras o sub-perfiles con necesidades diferentes, según lo presentado por (SIEMENS, PROFIBUS con STEP 7 V13, 2014) se puede resumir:

- a. **DPM1 (Maestro DP de Clase 1):** Son dispositivos de control que manejan las comunicaciones (variables de entrada y salida) para cada uno de sus esclavos conectados, generalmente son PLCs u ordenadores.
- b. **DPM2 (Maestro DP de Clase 2):** Son dispositivos de control para servicios de configuración y diagnóstico (consignas, parámetros de configuración, variables de funcionamiento, etc.) generalmente paneles o unidades de operador específicas.
- c. **DPS (Esclavo DP):** Son los dispositivos de periferia conectados a los maestros, los cuales leen los datos recibidos y ejecutan las acciones correspondientes, no toman iniciativas de comunicación, generalmente son accionamientos.

### 1.2.3 Profibus-PA

Profibus-PA (*Process Automation*, “Procesos de Automatización”) es una variante de Profibus-DP que está diseñada para la transferencia de información en el nivel más bajo de la pirámide CIM, se usa para la automatización de procesos industriales continuos y se puede integrar con redes Profibus de jerarquía superior como Profibus-DP. Este estándar

---

<sup>28</sup> Determinismo: Capacidad de un sistema en el cual los acontecimientos se presentan en forma causal lineal en intervalos de tiempos conocidos.

permite la comunicación entre los sistemas de control con los dispositivos de campo reemplazando los bucles de corriente analógicos de 4-20 mA (Mandado et al., 2006).

Profibus-PA cumple los requerimientos necesarios para trabajar en zonas industriales denominadas "Ex" de seguridad intrínseca, es decir, áreas con riesgo de explosión o atmósferas hostiles de operación satisfaciendo las exigencias de industrias como la química y la petroquímica (SIEMENS, Comunicación con SIMATIC, 2006).

Este perfil de Profibus se basa en la técnica de transmisión MBP<sup>29</sup>, hace posible la transferencia de datos y alimentación hacia el dispositivos a través de un cable de 2 hilos, en la capa física se utiliza el estándar de conexión RS-485 y en zonas que se requiere la variante de seguridad se implementa la norma IEC 61158-2 con una velocidad de transferencia de 31.25 Kbits/segundo. Las características que se posee en la capa de enlace y de aplicación son las mismas a las usadas en Profibus-DP (SIEMENS, Comunicación con SIMATIC, 2006); (SIEMENS, PROFIBUS con STEP 7 V13, 2014).

En la red Profibus generalmente coexisten los dos perfiles de comunicación tanto Profibus-DP como Profibus-PA, es por eso que se necesita la conversión de las señales de RS-485 a IEC 61158-2 para lo cual se utiliza los adaptadores de segmento (*DP/PA Adapter*, "Adaptador DP/PA").

Los adaptadores de segmento se convierten en el puente o *bridge* entre las dos redes Profibus, también poseen opto acopladores que permiten un aislamiento galvánico. Este adaptador se constituye en un procesador subordinado para la red Profibus-DP que gestiona los elementos PA, en cambio para la red Profibus-PA hace el papel de procesador principal de comunicaciones (véase figura 1.19) (Mandado et al., 2006).

---

<sup>29</sup> MBP (*Manchester Coding y Bus Powered*, "Codification Manchester y Alimentation de Bus")

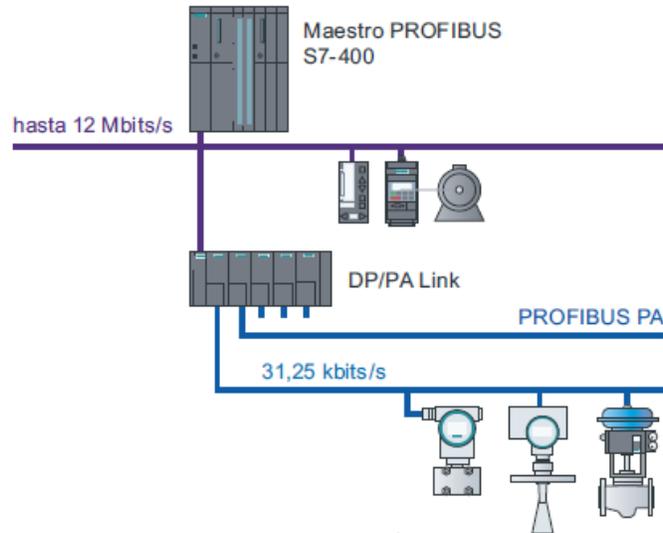


Figura 1.19: Conexiones en Profibus-PA

Fuente: (SIEMENS, Comunicación con SIMATIC, 2006)

#### 1.2.4 Profibus-FMS

Profibus-FMS (*FieldBus Message Specification*, “Especificación de Mensaje de Campo”) es el perfil de la familia Profibus orientado a dar servicios de comunicación a los sistemas electrónicos de control en los escalafones superiores de la pirámide CIM, como niveles de célula y planta (véase Figura 1.20).

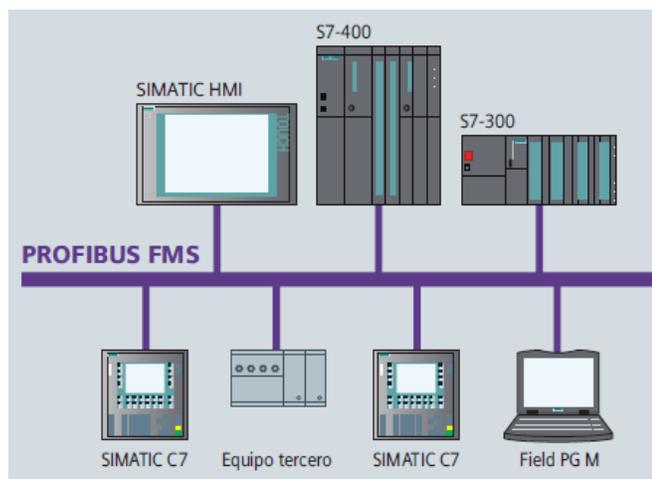


Figura 1.20: Conexiones en Profibus-FMS

Fuente: (SIEMENS, Comunicación con SIMATIC, 2006)

Las capas tanto físicas como de enlace guardan las mismas características que en Profibus-DP pero la diferencia radica en la capa 7 del Modelo OSI, es decir, la capa de aplicación, la cual presenta las interfaces necesarias para la transferencia de volúmenes más grandes de datos con una estructura de cliente-servidor (Guerrero, Yuste, & Martínez, 2009); (Mandado et al., 2006).

Profibus-FMS permite transferir datos estructurados (variables FMS) con el fin de conseguir una comunicación abierta con elementos de campo creados por diferentes fabricantes, con este propósito Profibus-FMS implementa los servicios básicos de *Read*, *Write* e *Information Report* (SIEMENS, Comunicación con SIMATIC, 2006).

La capa de aplicación de Profibus-FMS se divide en dos secciones, la primera denominada FMS donde se describen los objetos de comunicación, los servicios de aplicación y los modelos usados para la comunicación; la segunda parte llamada LLI (*Lower Layer Interface*, “Capa Menor de Interface”) que permite adecuar los servicios de aplicación a las características de la capa de enlace del nivel FDL (Mandado et al., 2006).

En un proceso distribuido se trabaja con dispositivos de control diferentes que pueden ser conectados a una red Profibus-FMS, a estos elementos se los conoce como VFD (*Virtual Field Device*, “Dispositivo Virtual de Campo”). Los objetos de un dispositivo VFD que pueden ser comunicados (variables, programas, rangos de datos, etc.) en la red se los denomina CO (*Communication Objects*, “Objetos de Comunicación”), los cuales son introducidos en un diccionario local de objetos en cada procesador de comunicaciones Profibus, con el fin de tener acceso a sus características y configuraciones mediante un índice y un nombre (Mandado et al., 2006).

### 1.3 PROFINET

Profinet es el resultado de la integración de las propiedades de Profibus y *Ethernet*, es un estándar que ofrece soluciones de red para procesos de automatización, la comunicación Profinet se basa en protocolos *Ethernet*, UDP<sup>30</sup>, TCP/IP (Guerrero, Yuste, & Martínez, 2009). *Industrial Ethernet* se basa en la norma IEEE 802.3 y facilita la comunicación entre los sistemas electrónicos de automatización en los niveles de célula con los sistemas de gestión en los niveles superiores de planta y oficina, ofrece servicios de TI (Tecnología de la Información) empleando una misma ingeniería para toda la instalación, es un estándar abierto que cumple la norma IEC 61158 para la automatización industrial (SIEMENS, Comunicación con SIMATIC, 2006).

Profinet es un estándar desarrollado para cubrir todas las necesidades de la ingeniería de automatización, está capacitado para integrar desde comunicaciones en el nivel de control, automatización estándar con sistemas I/O, integración con variadores de velocidad, redundancia, funciones de seguridad hasta poderosas aplicaciones de control de movimiento (véase Figura 1.21).

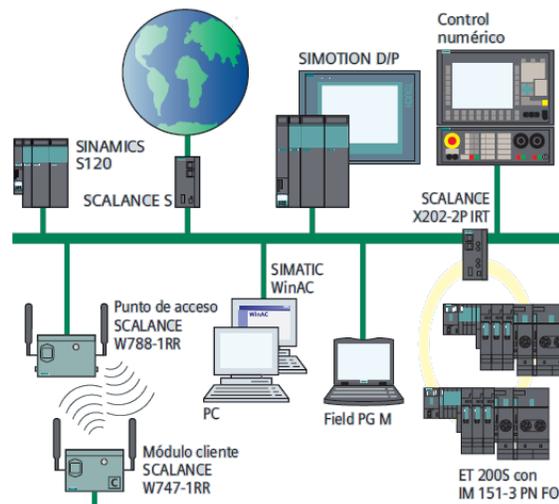


Figura 1.21: Red Profinet

Fuente: (SIEMENS, Comunicación con SIMATIC, 2006)

<sup>30</sup> UDP (*User Datagram Protocol*, “Protocolo Datagrama de Usuario”): Es un protocolo del nivel de transporte basado en el intercambio de datagramas.

Este estándar maneja las comunicaciones en tiempo real para datos de I/O cíclicos por lo cual puede ser llamado también como Profinet-RT, permite la utilización de cables y *switches* estándares de *Ethernet* por lo cual se la puede concebir como una comunicación fácil, rápida, flexible y abierta (Guerrero, Yuste, & Martínez, 2009); (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008).

Según (Guerrero, Yuste, & Martínez, 2009) existen dos versiones de redes Profinet:

- Profinet-I/O para la integración de dispositivos de campo descentralizados simples y aplicaciones de tiempo crítico.
- Profinet-CBA para la integración de los sistemas de automatización distribuidos basadas en componentes.

Además Profinet posee diferentes perfiles de servicio como *Profisafe* y *Profidrive* que presentan características específicas para procesos determinados de automatización (SIEMENS, Comunicación con SIMATIC, 2006).

### **1.3.1 Características Generales**

Las redes Profinet fueron desarrolladas usando las funcionalidades que presentaba *Ethernet* para la interconexión de ordenadores con varios años de experiencia desde su aparición en 1970, al ser esta una tecnología ya instalada permitió su adaptación para entornos industriales usando parte de su estructura como: cableado, topología, formatos, controladores, etc. existentes, es así que Profinet ha sido desarrollado como un estándar con varias características específicas enfocadas al entorno industrial, según lo presentado por (Guerrero, Yuste, & Martínez, 2009); (Profinet, S.A.) se puede destacar:

- Es un estándar abierto basado en *Industrial Ethernet*.
- Capaz de usar equipos tanto de *Industrial Ethernet* como de *Standard Ethernet*.
- Usa tecnología TCP/IP y Tecnologías de la Información.
- Integra los sistemas con los buses de campo.

- Gran potencia de transferencia con muchas estaciones.
- Utiliza varios medios de transmisión.
- Alta disponibilidad gracias a redes redundantes.
- Capacidad de Tiempo real.
- Tecnología escalable gracias a *routers* y *switches*.
- Flexibilidad.
- Más rápido que los buses actuales especialmente para sistemas de control de movimiento.
- Existe gran variedad de productos disponibles en el mundo.

#### **1.3.1.1 Control de Acceso al Medio**

Profinet al estar basado en redes *Ethernet* utiliza su procedimiento de control de acceso al medio conocido como CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*, “Acceso Múltiple con Detección de Colisiones”). Una estación que desee enviar datos se encarga de identificar la ocupación de la línea y al reconocerla como libre (*Carrier Sense*, CS “Detección de Portadora”) procede a enviar sus datos, durante el envío la estación receptora tiene la capacidad de detectar colisiones que se pudiesen producir por transmisión de otras estaciones debido a un nivel defectuoso (*Multiple Access*, MA “Acceso Múltiple”), entonces cancela la comunicación (SIEMENS, Comunicación con SIMATIC, 2006).

En la transmisión inalámbrica de datos se utiliza el control de acceso al medio CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*, “Acceso Múltiple con Evasión de Colisiones”) el cual permite enviar colisiones conscientemente, una estación comprueba si el medio está libre mediante el envío de un paquete de datos de comprobación y reserva el canal durante un tiempo determinado para realizar su intercambio de información (SIEMENS, Comunicación con SIMATIC, 2006).

### 1.3.1.2 Trama de Datos

Profinet es un estándar que ocupa la arquitectura, control de acceso al medio y demás características de *Ethernet*, en lo correspondiente a la trama de datos se mantiene la estructura heredada, pero agrega ciertos campos que permiten incluir comunicaciones de tiempo real, isócronas de tiempo real y no de tiempo real.

El manejo de prioridades en la trama y el modelo de actualización de datos a través de ventanas de tiempo permiten prestar los tres servicios a través de Profinet. En la configuración se generan las ventanas de tiempo en las cuales se agregan los datos a ser enviados (UNIOVI, S.A.). En la Figura 1.22 se puede observar la trama de datos de Profinet.

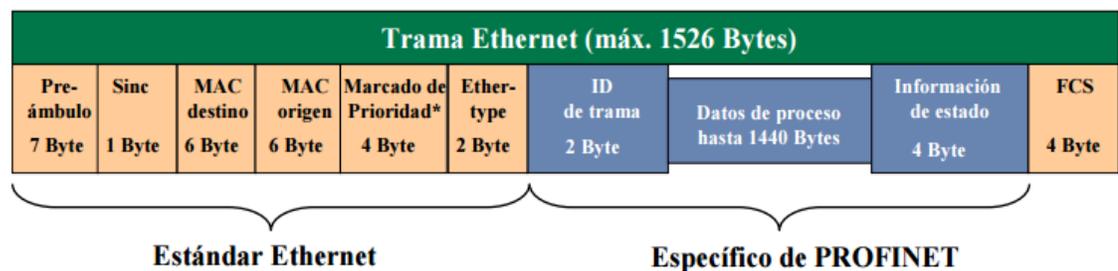


Figura 1.22: Trama de Datos en Profinet

Fuente: (UNIOVI, S.A.)

### 1.3.1.3 Medios Físicos

Las redes Profinet básicamente utilizan tres medios de transmisión para sus comunicaciones: mediante señales eléctricas a través de un cable de par trenzado, mediante señales ópticas a través de cables de fibra óptica y por medio de ondas electromagnéticas. Según lo presentado por (SIEMENS, Comunicación con SIMATIC, 2006); (Profinet, S.A.); (SIEMENS, PROFINET con STEP 7 V13, 2014) se puede destacar las siguientes características.

**a. Comunicación Mediante Señales Eléctricas - Cable *Ethernet***

- **Topología:** La topología más usada con este medio es la de estrella que permite la conexión de todos los procesadores mediante elementos activos de red. También se puede ocupar una topología en árbol y lineal conectando varios ramas mediante los *switches* o convertidores necesarios.
- **Medio Físico:** Se utiliza cables de red o *patch cord*<sup>31</sup> de dos o cuatro pares en sus diferentes tipos A, B o C dependiendo el caso, también se puede usar la versión híbrida para datos y energía. La conexión del cable puede realizarse en modo directo o cruzado.
- **Velocidad de Transmisión:** Las diversas versiones de estándares para el cableado permiten tener velocidades entre 10 y 100 Mb/s.
- **Distancia de Transmisión:** Distancia de extensión de la red de hasta 5 km.
- **Número de Procesadores de Comunicaciones:** El número máximo de estaciones conectadas es de hasta 127 procesadores de comunicación.

**b. Comunicación Mediante Señales Ópticas**

- **Topología:** La tecnología de transmisión óptica permite topologías de red en estrella, anillo y lineal.
- **Medio Físico:** Se utiliza cables de fibra óptica POF (Polymer Optical Fiber, “Fibra Óptica de Polímero”).
- **Velocidad de Transmisión:** Se puede lograr velocidades de transmisión de 100 Mb/s.
- **Distancia de Transmisión:** La distancia máxima que se puede alcanzar entre dos dispositivos es 26 km, una extensión de la red de hasta 150 km.

---

<sup>31</sup> *Patch Cord*: Cable de conexión usado en redes de computadoras o sistemas electrónicos.

- **Número de Procesadores de Comunicaciones:** Se puede conectar más de 1000 estaciones a la red.
- c. Comunicación Mediante Señales Electromagnéticas**
- **Topología:** Se usa principalmente una topología en estrella.
  - **Medio Físico:** El estándar 802.11 x.
  - **Velocidad de Transmisión:** Según la aplicación.
  - **Distancia de Transmisión:** La distancia máxima que se puede alcanzar es de hasta 1000 m por segmento.
  - **Número de Procesadores de Comunicaciones:** Se puede conectar máximo 8 procesadores de comunicación.

En la siguiente tabla se expone las características de los medios de transmisión más usados en Profinet.

Tabla 1.4: Características de los Medios de Transmisión Profinet

Propiedad Física	Sistema de Conexión	Tipo de cable / Medio de transmisión	Velocidad	Long max. Segmento (entre dos dispositivos)	Ventajas
<b>Eléctrica</b>	Conector RJ 45	100 Base-TX Cable de cobre de par trenzado 2x2, simétrico apantallado, exigencia de transmisión según CAT5 IEEE 802.3	100Mbps/s, dúplex	100m	Conexión de cable simple y económica
<b>Óptica</b>	SCRJ 45 ISO/IEC 61754-24	100 Base-FX Cable de fibra óptica POF 980/1000 µm (diámetro de núcleo / diámetro exterior) ISO/IEC 60793-2	100Mbps/s, dúplex	50m	Uso con grandes diferencias de potencial. Insensible a la radiación electromagnética.
		Fibra óptica recubierta de plástico PCF 200/230 µm (diámetro de núcleo / diámetro exterior)	100Mbps/s, dúplex	100m	Baja atenuación de cable. Permite segmentos mucho más largos

		ISO/IEC 60793-2			
	BFOC (Bayonet Fiber Optic Connection)	Cable de fibra óptica – fibra monomodal 10/125 µm (diámetro de núcleo / diámetro exterior) ISO/IEC 60793-2	100Mbps/s, dúplex	26 km	
	SC (Subscriber Connector) ISO/IEC 60874	Cable de fibra óptica – fibra multimodal 50/125 y 62,5/125 µm (diámetro de núcleo / diámetro exterior) ISO/IEC 9314-4	100Mbps/s, dúplex	3000 m	
<b>Ondas Electroma- gnéticas</b>	-	IEEE 802.11 x	Según sea la amplificación utilizada (a,g,h,etc.)	100m	Mayor Movilidad. Conexión en red económica con dispositivos lejanos y de difícil acceso

Fuente: (Profinet, S.A.)

### 1.3.1.4 Componentes de Red

Las redes Profinet al adoptar la tecnología *Ethernet* utilizan algunos elementos activos que permiten interconectar las estaciones de red según los diferentes medios de comunicación que se utilice, los principales elementos son *switches*, *routers* y convertidores, los cuales puede integrar diferentes dispositivos y redes de automatización permitiendo comunicaciones alámbricas e inalámbricas. De acuerdo a lo presentado por (Guerrero, Yuste, & Martínez, 2009); (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008); (SIEMENS, PROFINET con STEP 7 V13, 2014) se puede explicar lo siguiente:

**Switches:** El *switch* también conocido como conmutador es el encargado de filtrar y regenerar los paquetes de *Ethernet* para permitir la interconexión de varios elementos y cubrir mayores distancias. Si una estación debe enviar datos a distintos controladores electrónicos deberá conectarse a un puerto del *switch* (punto a punto), los demás elementos de igual manera deberán ocupar un puerto del conmutador para establecer la comunicación.

El *switch* verifica las tramas de datos que recibe y genera tablas de tráfico que le permite encaminar el paquete por el puerto de salida adecuado, este elemento memoriza las

direcciones MAC<sup>32</sup> de sus dispositivos mediante la identificación de los paquetes que recibe y gestiona cada puerto para la conexión de datos, posee una memoria interna o *buffer*<sup>33</sup> para independizar las velocidades de emisión y recepción de cada estación. Optimiza la red mediante el uso de un puerto para una comunicación, pudiendo gestionarse más a través de los puertos restantes, además elimina las colisiones.

**Routers:** El *router* también conocido como encaminador está encargado de la interconexión de redes como LAN y WAN, poseen una capacidad más elevada que los *switches* y trabajan en la capa de Red almacenando y reemitiendo paquetes, además permiten determinar las rutas más eficientes para la información. Las estaciones conectadas a él pueden comunicarse siempre y cuando el *router* habilite el tráfico de datos entre ellas.

Cuando un *router* recibe un paquete de datos, lee la dirección del destinatario y reenvía el paquete al siguiente nodo según la ruta adecuada, cada elemento posee la "inteligencia" para determinar los caminos gracias a un mapa de carreteras. Un *router* permite bloquear tráfico de datos aislando las redes de planta u oficina, o también para tomar acciones de seguridad.

**Convertidores o Pasarelas:** Estos elementos son conocidos como *Gateways* y su función principal es la de conectar dos redes de diferente arquitectura permitiendo la comunicación entre equipos que manejan protocolos distintos de comunicación. Permite la conexión de varios perfiles de la misma red que manejen medios de transmisión distintos. Los convertidores de protocolos trabajan a nivel de la capa de Aplicación donde permiten la traducción de servicios.

---

<sup>32</sup> Dirección MAC (*Media Access Control*, "Control de Acceso al Medio"): Identificador de 48 bits unívoco para equipos de red.

<sup>33</sup> *Buffer*: Espacio de memoria en ciertos dispositivos donde se almacena datos temporalmente.

En la actualidad existen varios componentes de red, pero cabe recalcar que a pesar de ser Profinet una red que puede usar elementos de *Ethernet*, existen dispositivos diseñados especialmente para el entorno industrial que presentan las características y prestaciones necesarias, una importante familia de estos elementos son los productos SCALANCE los cuales están diseñados para todo tipo de requerimientos de conexión de redes industriales. Dentro SCALANCE se cuenta con elementos como *routers*, *switches*, pasarelas, módulos de seguridad de redes, módulos para conexión alámbrica e inalámbrica (SIEMENS, Comunicación con SIMATIC, 2006).

### 1.3.1.5 Seguridad

Las redes de comunicación industrial crecen cada día por la necesidad de conectar más elementos de campo a las redes corporativas o *intranet* de las fábricas, por lo tanto los conceptos de seguridad adquieren cada vez mayor importancia. Los servicios de red que se han incorporado a los sistemas de automatización como servidores WEB, envío de correo, capacidad de mando remoto, servicios FTP, etc. desembocan en una mayor relación con el mundo informático, por lo cual se expone a los peligros típicos como *hackers*, virus, troyanos, etc. (SIEMENS, Comunicación con SIMATIC, 2006). Las amenazas dentro los procesos pueden ser internas o externas siendo las principales:

- Errores técnicos y de manejo.
- Errores en los programas.
- Virus y gusanos de *software*.
- Troyanos
- Ataques de tipo *Man in the Middle*<sup>34</sup>.
- *Phishing*<sup>35</sup> de contraseñas.

---

<sup>34</sup> *Man in the Middle* “Hombre en el Medio”: Ataque a la red en el cual un agente externo puede leer e insertar mensajes sin que el emisor o receptor lo detecten.

<sup>35</sup> *Phishing* “Suplantación de Identidad”: Término usado para referirse a un intento de adquisición de información confidencial en redes informáticas.

En las redes Profinet se puede implementar algunas medidas de seguridad que pueden ser establecidas en los equipos de comunicación de la red o en los elementos activos por los cuales fluye el tráfico de información, según lo expuesto por (SIEMENS, PROFINET con STEP 7 V13, 2014) las principales acciones contra la manipulación y pérdida de la seguridad de datos en la industria son:

- Seguridad de datos mediante Firewall
- VPN (*Virtual Private Network*, “Red Privada Virtual”) permite intercambiar datos de manera segura con redes externas públicas como *Internet*.
- Segmentación de las células de automatización con módulos de seguridad dependiendo los requerimientos.
- Autenticación de dispositivos mediante canales seguros encriptados.
- Encriptación de los datos que circulan por la red, cada módulo maneja los certificados necesarios con las claves

Los conceptos de seguridad están provistos tanto para la protección de los dispositivos como para el resguardo de las redes ante accesos no autorizados, adicionalmente se maneja conceptos de segmentación que permite a ciertos elementos de una célula comunicarse a través de un módulo de seguridad, este concepto crea los llamados puntos seguros. La autenticación y autorización serán necesarias en dichos elementos para permitir el tráfico de datos externos (PNO, 2011).

#### **1.3.1.6 Principales Tipos de Comunicación**

Profinet cubre un completo rango de aplicaciones de automatización con diferentes exigencias de respuesta y prioridad, para esto Profinet diferencia tres tipos básicos de comunicaciones (véase Figura 1.23), según lo expuesto por (SIEMENS, Automate with the leading Industrial Ethernet standard and profit now, 2012) se puede resaltar:

- a. **No de Tiempo Real:** Esta comunicación utiliza la transmisión de datos mediante protocolos TCP/IP o UDP para enviar parámetros y datos que no son críticos en el tiempo, cumple un tiempo de ciclo de unos 100 ms.
- b. **De Tiempo Real (RT):** Es usado para la transmisión de datos críticos de proceso, que pueden ser cíclicos o incluso de interrupción, con lo cual permite responder acciones de ingeniería en tiempos de microsegundos, aproximadamente con una actualización de entre 1 y 10 ms.
- c. **Isócrono de Tiempo Real (IRT):** Esta comunicación es usada para sofisticados sistemas de control de alto rendimiento, permite ciclos de ejecución sincronizados para transmitir los datos en intervalos deterministas con un periodo de actualización de 250  $\mu$ s a 1 ms con una fluctuación mínima.

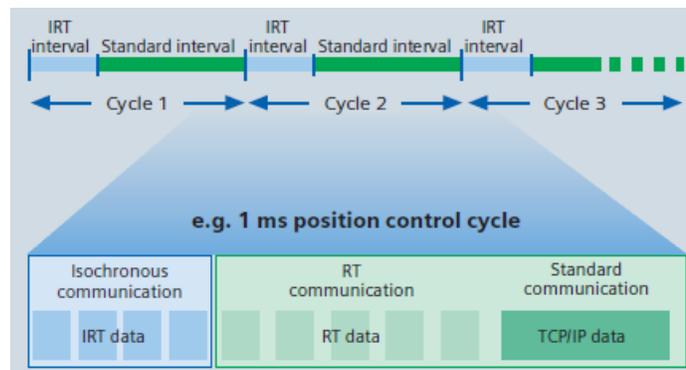


Figura 1.23: Comunicaciones de Tiempo Real en Profinet

Fuente: (SIEMENS, Automate with the leading Industrial Ethernet, 2012)

### 1.3.2 Profinet-I/O

Profinet-I/O es un servicio de comunicación desarrollado gracias a varios años de experiencia con Profibus-DP, por lo tanto es un estándar que integra las propiedades de conexión de Profibus con la tecnología *Ethernet*. Es un modelo abierto de comunicación para la automatización e ingeniería, ya que permite la conexión entre controladores con dispositivos de campo como *drivers* y elementos de entrada-salida (véase Figura 1.24) (Guerrero, Yuste, & Martínez, 2009).

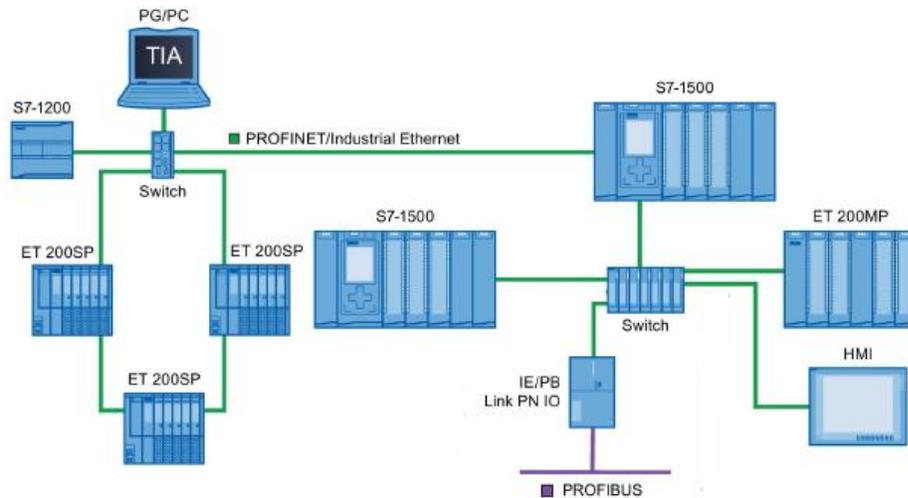


Figura 1.24: Conexiones en Profinet-I/O

Fuente: (SIEMENS, PROFINET con STEP 7 V13, 2014)

Profinet es una red fácilmente migrable desde Profibus y está desarrollado para la realización de aplicaciones modulares descentralizadas permitiendo crear soluciones de automatización completas (Guerrero, Yuste, & Martínez, 2009).

Profinet-I/O usa una tecnología de conmutación que permite que todas las estaciones accedan a la red en cualquier momento, se puede realizar varias comunicaciones simultáneamente desde diferentes puntos de la red gracias al modo *dúplex* que se maneja en los *switches*, de esta manera se logra un uso efectivo de la red con una velocidad elevada de 100 Mb/s, el tiempo mínimo de actualización de los dispositivos I/O es de 1 ms para elementos que funcionen con una frecuencia de envío de 250  $\mu$ s (Guerrero, Yuste, & Martínez, 2009).

Este estándar posee una gran flexibilidad ya que permite una comunicación tanto en tiempo real (RT), como una comunicación isócrona en tiempo real (IRT); el ciclo de Profinet-I/O maneja dos partes tanto una abierta y una determinista, permitiendo simultáneamente tanto una transmisión de datos en TCP/IP como IRT sin que se molesten entre sí (SIEMENS, Comunicación con SIMATIC, 2006); (PNO, 2011).

Similar a lo que se tiene en el estándar Profibus en cuanto a las diferentes clases de dispositivos DP, también Profinet-I/O tiene una distinción de ciertos elementos, según lo presentado por (PNO, 2011) se resume lo siguiente:

- a. **Controlador I/O:** Es generalmente el PLC en cual corre el programa de automatización, es comparable con el Maestro DP de clase 1 ya que es el productor y consumidor de los datos hacia los dispositivos.
- b. **Dispositivo I/O:** Son los dispositivos de entrada-salida distribuidos en el campo que pueden estar conectados a uno o más controladores, es comparable con un Esclavo DP, este elemento es proveedor de datos de entrada y consumidor de datos de salida.
- c. **Supervisor I/O:** Pueden ser equipos de programación, ordenadores, interfaces hombre-máquina, que cumplen funciones de puesta en servicio o diagnóstico, es comparable con el Maestro DP de clase 2.

### 1.3.3 Profinet-CBA

Profinet-CBA (*Component Based Automation*, “Componentes Basados en Automatización”) es un concepto de comunicación que permite el desarrollo de soluciones de automatización distribuidas basadas en componentes o células parciales, está principalmente basado en la realización de aplicaciones modulares y comunicación entre máquinas (véase Figura 1.25).

Este perfil responde a las exigencias industriales de contar con mayor segmentación en la ingeniería de los procesos en busca de descentralizar el procesamiento de los mismos. Profinet-CBA permite dividir el sistema de automatización en partes independientes denominadas módulos tecnológicos, además posibilita la integración de estos como si fuesen componentes estandarizados con el fin de conseguir instalaciones y sistemas completos en las fábricas (Guerrero, Yuste, & Martínez, 2009).

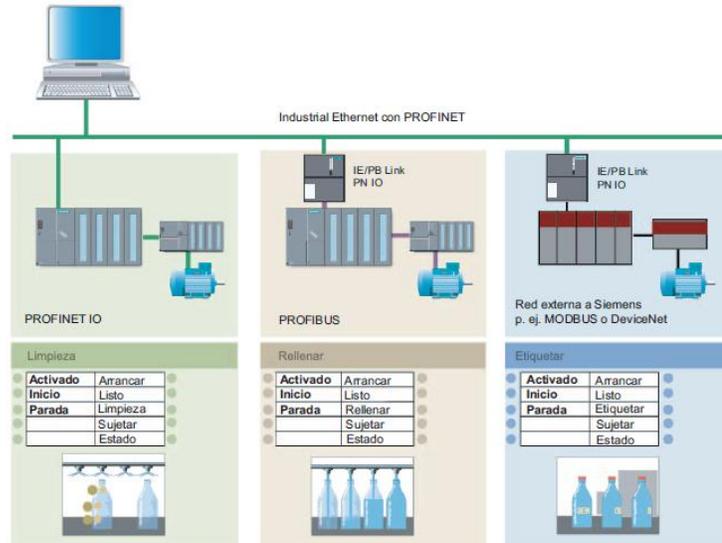


Figura 1.25: Conexiones en Profinet-CBA

Fuente: (SIEMENS, Comunicación con SIMATIC, 2006)

Este estándar es adecuado para transmisiones entre estaciones vía TCP/IP y comunicaciones en tiempo real (RT) con el fin de integrar módulos inteligentes de diferentes áreas de planta (SIEMENS, Comunicación con SIMATIC, 2006).

También Admite la comunicación cíclica y acíclica siendo especialmente adecuado para la transferencia de datos entre sistemas electrónicos de control gracias a períodos de actualización de hasta 10 ms. Utiliza la transferencia automática y otros procedimientos de control (bit de paridad por carácter y suma de control) para aumentar la integridad de datos y la precisión (PNO, 2011).

#### 1.4 MODBUS

Modbus es un protocolo de comunicación industrial desarrollado por la empresa Modicon en el año de 1979 con el objetivo de enlazar los dispositivos controladores con los elementos de campo, se usa para la transmisión de señales con el modelo maestro-esclavo o cliente-servidor. Es un estándar de tecnología abierta ideal para la monitorización remota de elementos de campo RTU (*Remote Terminal Unit*, “Unidad Terminal Remota”) y está diseñado para permitir a equipos industriales tales como PLCs, computadores, *drivers* para

motores y otros tipos de dispositivos físicos de entrada/salida comunicarse sobre una red (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008).

Modbus define una estructura de mensaje que los controladores electrónicos podrán interpretar sin importar el tipo de red que utilicen para sus comunicaciones, el estándar describe el proceso que el controlador usa para la petición de acceso a otros dispositivos, como estos responden a dichas peticiones y como los errores serán detectados y reportados.

Durante las comunicaciones en una red Modbus, el protocolo determina como cada controlador conoce las direcciones de los demás dispositivos, como reconoce los mensajes y que acciones tomará en respuesta luego de acceder a los datos contenidos en la información enviada, las réplicas son reformadas en los esclavos y enviadas de vuelta por la misma red Modbus (Modicon, 1996).

El maestro construye los mensajes estableciendo un modelo en función de los cuatro parámetros utilizados en Modbus y los dispositivos que lo reciben generan el mensaje como esclavos, de esta manera un controlador puede funcionar como maestro o esclavo en comunicaciones independientes (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008); (M-System CO. LTD., 2002).

Modbus usa distintas variantes para la transmisión de datos dependiendo el entorno en el que se desarrolle la aplicación, estas variantes incluyen: Modbus ASCII<sup>36</sup>, RTU y TCP/IP usados tanto en la *intranet* para la integración de elementos de planta como en la *extranet* para permitir acciones de monitorización y mando (véase Figura 1.26).

---

<sup>36</sup> ASCII (*Standard Code for Information Interchange*, “Código Estándar para el Intercambio de Información”): Código de caracteres basado en el alfabeto latino.

Las transacciones de información en Modbus se las puede hacer mediante módulos o controladores conectados directamente por interfaces como RS-232 y RS-485, algunos equipos se pueden comunicar usando Modbus Plus gracias a los protocolos MAP<sup>37</sup> y *Ethernet* (Modicon, 1996).

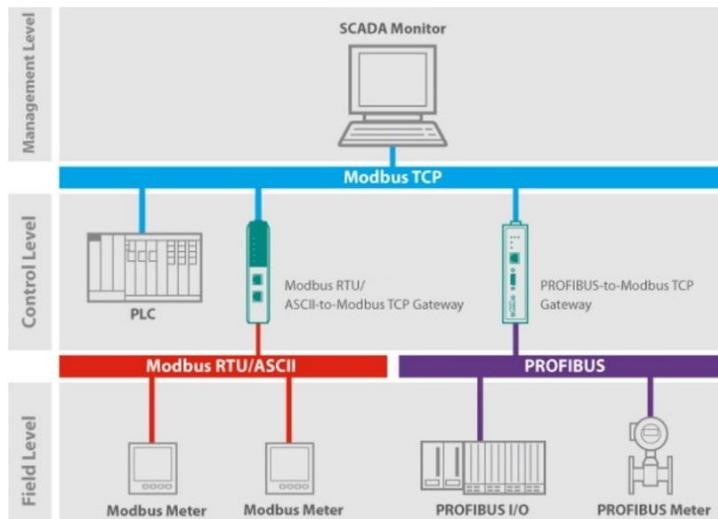


Figura 1.26: Red Modbus

Fuente: (NETEON, S.A.)

### 1.4.1 Características Generales

El protocolo Modbus ha sido desarrollado para la implementación de básicamente tres capas del modelo OSI de manera similar a la tecnología utilizada en Profibus, ya que este especifica la capa física para los medios de transmisión que utiliza, la capa de enlace para la estructura de comunicación entre los dispositivos (maestro-esclavo) y la capa de aplicación para los servicios prestados por Modbus refiriéndose a su estructura de reconocimiento del protocolo y generación de respuestas (véase Figura 1.27).

<sup>37</sup> MAP (*Manufacturing Automation Protocol*, “Protocolo de Automatización y Fabricación”): Protocolo de arquitectura estándar independiente del fabricante para la comunicación de elementos de fábrica.

7	APLICACIÓN		MODBUS
6	PRESENTACIÓN		
5	SESIÓN		
4	TRANSPORTE		
3	RED		
2	ENLACE DE DATOS	MAESTRO - ESCLAVO	
1	FÍSICA	RS-485	

Figura 1.27: Capas Implementadas en Modbus

Fuente: (Moreno, Becerra, & Jorge, 2012)

El uso de bobinas y registros de memoria interna en cada elemento permite la transferencia y almacenamiento de los datos enviados vía Modbus, las bobinas almacenan datos binarios y los registros valores numéricos, ambas se guardan en tablas relacionadas a los valores contenidos (Moreno, Becerra, & Jorge, 2012).

Otro aspecto importante que posee Modbus está relacionado con el tipo de intercambio de mensajes que se puede realizar entre el dispositivo maestro y sus esclavos, ya que estos puede ser de dos maneras: intercambios punto a punto o mensajes difundidos (*broadcast*) (Balcells & Romeral, 1997).

- **Intercambio Punto a Punto:** Este tipo compone siempre dos mensajes, una demanda del maestro y una respuesta del esclavo, puede ser desde un simple reconocimiento hasta mensajes de control y supervisión (Balcells & Romeral, 1997).
- **Difusión:** Estos mensajes constituyen una comunicación unidireccional desde el maestro hacia todos sus esclavos, no tienen respuesta ya que generalmente se usa para enviar mensajes de configuración, reset, etc (Balcells & Romeral, 1997).

### 1.4.1.1 Estructura de Mensaje

Los mensajes que son enviados en una red Modbus desde el maestro como peticiones o consultas, o desde el esclavo como respuestas; poseen una estructura de mensaje conformada por cuatro campos: Dirección, Código de Función, Datos, CRC (véase Figura 1.28). Esta estructura es la misma para todos los mensajes independientemente de la red lo único que varía es el campo de datos dependiendo el caso. Según lo expuesto por (M-System CO. LTD., 2002); (WEG, 2012) acerca de cada campo de la estructura se puede resumir lo siguiente:

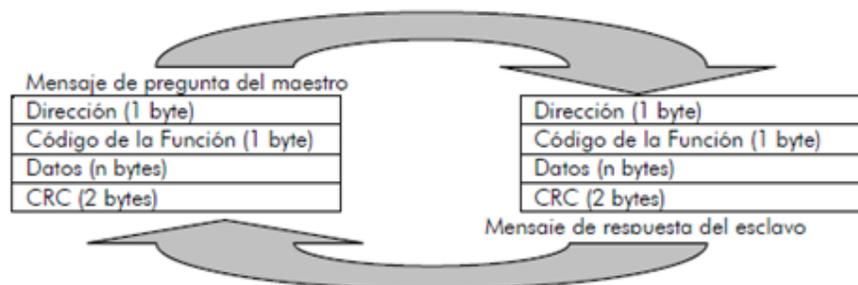


Figura 1.28: Estructura de Mensajes Modbus

Fuente: (WEG, 2012)

- a. **Dirección:** Las direcciones válidas en Modbus pueden ir de 0 a 247 ya que es el número máximo de esclavos soportados. Cuando el maestro inicia la comunicación envía una trama con la dirección del esclavo con el cual desea comunicarse (entre 1-247), el dispositivo subordinado responde el mensaje incluyendo nuevamente su dirección con el objetivo de que el maestro pueda reconocer que esclavo formuló el mensaje. La dirección 0 es usada para *broadcast* por lo cual ningún esclavo deberá responder.
- b. **Código de Función:** Los valores permitidos son entre 1-255 (decimal), cuando el maestro envía un mensaje especifica el código de función que permite al esclavo conocer qué tipo de acción debe desarrollar por ejemplo (lectura, escritura, etc.), el esclavo responde con el mismo código para indicar que no existieron errores, o

con un código equivalente solo cambiando el bit más significativo a 1 para indicar que algún tipo de error ha ocurrido.

- c. **Campo de Datos:** La longitud de los datos enviados es variable incluso puede ser cero, este campo contiene los datos enviados desde el maestro que deben ser utilizados por el esclavo para cumplir con la acción requerida.
- d. **Comprobación:** La última parte del mensaje es el campo para la comprobación de errores para lo cual se usa básicamente dos métodos LRC (*Longitudinal Redundancy Check*, “Verificación de Redundancia Longitudinal”) para ASCII, y CRC (*Cyclical Redundancy Check*, “Verificación de Redundancia Cíclica”) para RTU.

#### 1.4.1.2 Control de Acceso al Medio

La estructura de las redes Modbus se desarrolla bajo un modelo maestro-esclavo o cliente-servidor en el cual solo un dispositivo maestro puede iniciar las transacciones de mensajes llamadas *queries* o consultas, una vez que los esclavos han recibido dichos mensajes están en la capacidad de formular una respuesta o realizar la acción pertinente ordenada por el maestro. Una vez que los mensajes sean transmitidos se puede iniciar con nuevos intercambios en la red.

En otros tipos de redes los controladores Modbus se pueden comunicar usando una técnica para *peer-to-peer* (de igual a igual), en la cual cualquier controlador puede iniciar transacciones con otros controladores, en este caso uno de ellos puede operar como maestro para cierta comunicación y como esclavo para otra (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008); (Modicon, 1996).

- a. **La Consulta (Query):** Es el mensaje inicial que envía el maestro hacia uno de sus esclavos en cual incluye la información permitente según el modelo explicado en el apartado 1.4.1.1, por ejemplo un código de función 03 indica al esclavo que lea

los registros de su posesión y responda con su contenido, la información adicional enviada será en que registro empezar y cuantos leer (Modicon, 1996).

- b. La Respuesta:** Es el mensaje formado por el esclavo luego de reconocer los campos del comunicado recibido, este contiene la información requerida por el maestro y permite comunicarle si ha existido o no errores en la acción solicitada (Modicon, 1996).

#### 1.4.1.3 Medios Físicos

Las redes Modbus básicamente utilizan tres medios de transmisión basados en señales eléctricas como son: las interfaces RS-232, RS-485 y cables de par trenzado para Modbus TCP/IP. Según lo presentado por (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008); (Modicon, 1996) se puede destacar las siguientes características.

**a. Comunicación Mediante Señales Eléctricas - Interfaz RS-232**

- **Topología:** La topología más usada con este medio es la de bus, pero también se puede ocupar una topología en árbol conectando varias ramas.
- **Medio Físico:** Se utiliza un cable de 3 hilos según la interface RS-232.
- **Velocidad de Transmisión:** Las velocidades de transmisión están entre 300 b/s hasta 38.4 Kbits/s.
- **Distancia de Transmisión:** La distancia máxima que se puede alcanzar entre dos dispositivos es 350 m.
- **Número de Procesadores de Comunicaciones:** El número máximo de estaciones conectadas es de hasta 247 procesadores de comunicación.

**b. Comunicación Mediante Señales Eléctricas - Interfaz RS-485**

- **Topología:** La topología más usada con este medio es la de bus, pero también se puede ocupar una topología en árbol conectando varias ramas.

- **Medio Físico:** Se utiliza un cable de 3 hilos según la interface RS-485.
  - **Velocidad de Transmisión:** Las velocidades de transmisión están entre 300 b/s hasta 38.4 Kbits/s.
  - **Distancia de Transmisión:** La distancia máxima que se puede alcanzar entre dos dispositivos es 350 m.
  - **Número de Procesadores de Comunicaciones:** El número máximo de estaciones conectadas es de hasta 247 procesadores de comunicación.
- c. Comunicación Mediante Señales Eléctricas - *Cable Ethernet***
- **Topología:** Se usa principalmente una topología en estrella.
  - **Medio Físico:** Se utiliza cables de red o *patch cord* de dos o cuatro pares en sus diferentes tipos A, B o C dependiendo el caso.
  - **Velocidad de Transmisión:** Puede alcanzar velocidades de 100 Mbits/s.
  - **Distancia de Transmisión:** Distancia máxima con una extensión total de 5 km.
  - **Número de Procesadores de Comunicaciones:** Se puede conectar 32 nodos por segmento y hasta 64 segmentos.

#### 1.4.1.4 Códigos de Función

Los códigos de función Modbus permiten al dispositivo maestro intercambiar información con los esclavos, sea para escribir o leer datos, cada código corresponde a una acción específica que deberá responder el equipo subordinado, a continuación según lo presentado en (Modicon, 1996) se expone una tabla resumen con los principales códigos de función del protocolo.

Tabla 1.5: Principales códigos de función Modbus

<b>Código</b>	<b>Nombre</b>	<b>Descripción</b>
01	Leer estado de bit	Permite leer el valor de los bits de salida
02	Leer estado de entrada	Permite leer el valor de los bits de entrada.
03	Leer registros de retención	Permite leer el valor de palabra de los registros de retención.
04	Leer registro de entrada	Permite leer el valor de palabra de los registros de entrada.
05	Escribir bit de salida	Permite escribir 1 bit de salida
06	Escribir registro de retención	Permite escribir 1 registro de retención
07	Leer estado de excepción	Permite leer el estado de 8 bits de excepción.
08	Funciones de Diagnostico	Permite a través de subfunciones cumplir tareas de diagnóstico en los esclavos.
11	Contador de Eventos	Permite reconocer si el esclavo ha manejado correctamente los mensajes a través de un registro contador de eventos.
12	Registrador de Eventos	Retorna una palabra de estado, contador de eventos, contador de mensajes y un campo de eventos de byte.
15	Escribir varios bits de salida	Permite escribir varios bits de salida.
16	Escribir varios registros	Permite escribir 1 o más registros de retención.
17	Reporte de ID de esclavo	Retorna una descripción con el tipo de controlador (Micro84, 484, 184/384, 584, 884, 984). Cada tipo retorna información específica.
20	Leer referencia general	Retorna el contenido de los archivos de referencia de la memoria extendida.
21	Escribir referencia general	Permite escribir el contenido de los archivos de referencia de la memoria extendida.

Fuente: (Modicon, 1996)

### 1.4.2 Modbus-ASCII

El modo de transmisión ASCII permite un intercambio de datos por la red Modbus con una codificación ASCII, se envían caracteres en formato hexadecimal del "0" (16#30) al "9" (16#39) y entre "A" (16#41) y "F" (16#46), por ejemplo si se desea transmitir la cadena FF se deberá enviar realmente dos bytes: 16#46 y 16#46.

Para la delimitación del mensaje se utiliza dos caracteres especiales tanto de inicio como de fin, para marcar el comienzo de trama se emplea el carácter (16#3A ":") y para establecer el final se utiliza el par no imprimible "CRLF" (16#OD retorno de carro) y (16#0A salto de línea) (Modicon, 1996); (Candelas, 2011). El formato para cada byte en el modo ASCII es.

Tabla 1.6: Formato de Byte en Modbus-ASCII

<b>ASCII</b>	
<b>Sistema de Codificación</b>	<ul style="list-style-type: none"> <li>• Hexadecimal, ASCII: caracteres 0-9 y A-F</li> <li>• Uno carácter hexadecimal contenido en cada carácter ASCII del mensaje.</li> </ul>
<b>Bits por Byte</b>	<ul style="list-style-type: none"> <li>• 1 Bit de Inicio</li> <li>• 7 Bits de Datos, bit menos significativo enviado primero.</li> <li>• 1 Bit de paridad (0 bits si no hay paridad)</li> <li>• 1 Bit de parada con paridad, 2 Bits si no hay paridad.</li> </ul>
<b>Comprobación de errores</b>	<ul style="list-style-type: none"> <li>• 8 Bits para LCR (Verificación de Redundancia Longitudinal)</li> </ul>

Fuente: (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008)

Este modo de transmisión tiene principalmente dos ventajas: la primera es que ofrece un medio factible para la determinación de inicio y fin de trama mediante los caracteres especiales enviados, esto sin importar los tiempos de transmisión que se manejen en el canal, la segunda está referida al hecho que permite trabajar con equipos de procesamiento

lento si los datos son almacenados en *buffers*. El inconveniente es que ASCII es un modo de envío de datos más lento que RTU (Candelas, 2011).

ASCII permite tiempos de intervalo de hasta un segundo entre caracteres sin causar error, si se supera este tiempo, el dispositivo receptor asume que ha ocurrido un error. La trama de datos enviados en ASCII es la siguiente (véase Figura 1.29).

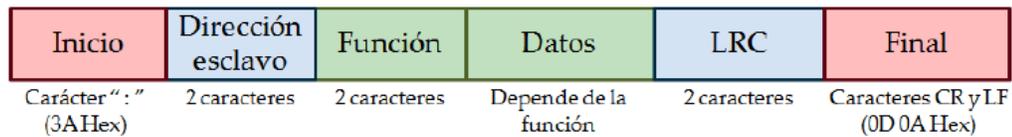


Figura 1.29: Trama de Datos en Modbus-ASCII

Fuente: (Candelas, 2011)

### 1.4.3 Modbus-RTU

El modo de transmisión RTU se destaca por que los bytes de datos se envían sin ningún tipo de conversión, es decir en una codificación plana ya que está pensado para comunicaciones serie. La delimitación del mensaje se realiza de una manera diferente al modo ASCII ya que se utiliza intervalos de tiempo de caracteres de silencio, un carácter de silencio tiene una duración de un byte de datos y va a estar determinado por el número de bits que se usa para la codificación dividido entre la velocidad. Un tiempo aproximado de silencio para una configuración de 19200 bps con bits de inicio, datos, paridad y parada es de 2ms (Modicon, 1996).

Modbus-RTU posee un método de comprobación de errores basado en CRC de 16 bits, el cálculo del código se realiza en el emisor y receptor para determinar la existencia o no de errores (Candelas, 2011). El formato para cada byte en el modo RTU es.

Tabla 1.7: Formato de Byte en Modbus-RTU

<b>RTU</b>	
<b>Sistema de Codificación</b>	<ul style="list-style-type: none"> <li>• 8 bits binario, hexadecimal: 0-9 y A-F</li> <li>• Dos carácter hexadecimales contenidos en cada 8 bits de mensaje.</li> </ul>
<b>Bits por Byte</b>	<ul style="list-style-type: none"> <li>• 1 Bit de Inicio</li> <li>• 8 Bits de Datos, bit menos significativo enviado primero.</li> <li>• 1 Bit de paridad (0 bits si no hay paridad)</li> <li>• 1 Bit de parada con paridad, 2 Bits si no hay paridad.</li> </ul>
<b>Comprobación de errores</b>	<ul style="list-style-type: none"> <li>• 16 Bits para CRC (Verificación de Redundancia Cíclica)</li> </ul>

Fuente: (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008)

En RTU los mensajes empiezan luego de un silencio de al menos 3.5 caracteres, los elementos de red monitorizan dicho silencio, luego decodifican el dato de dirección para reconocer el destinatario (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008). La trama de datos para este tipo de transmisión está formada por los siguientes campos (véase Figura 1.30).



Figura 1.30: Trama de Datos en Modbus-RTU

Fuente: (Candelas, 2011)

La principal ventaja de RTU es el buen aprovechamiento del canal de comunicación (mayor densidad de caracteres) mejorando la velocidad de transmisión para una misma tasa de baudios que en ASCII. El inconveniente es que es necesaria una gestión de tiempos entre bits para saber cuándo comienza o termina una trama (Modicon, 1996); (Candelas, 2011).

#### 1.4.4 Modbus-TCP/IP

Modbus-TCP es un protocolo de comunicación que define el uso de mensajes Modbus dentro de un entorno de red TCP, está diseñado para permitir a dispositivos industriales de planta comunicarse con los elementos de gestión o con equipos de la *intranet*, *extranet* o *internet*. Fue desarrollado por *Schneider Automation* como una variante de Modbus cuyo uso para supervisión y control se encontraba altamente implementado en las redes de automatización (Olaya, Barandica, & Guerrero, 2004).

Modbus es un protocolo maestro-esclavo en el cual cada petición del maestro es tratada de forma separada por los esclavos, este modelo concuerda con el de Consulta-Respuesta o Cliente-Servidor que se implementa en las redes basadas en *Ethernet*. Modbus TCP es un protocolo orientado a la conexión en el cual los elementos involucrados establecen un canal comunicaciones antes de transferir datos. Adicionalmente, el protocolo TCP permite establecer una red extensa con varias comunicaciones concurrentes, de modo que los dispositivos maestros pueden intercambiar datos mediante diferentes canales (Olaya, Barandica, & Guerrero, 2004).

Los dispositivos maestros que se tenían en las redes Modbus convencionales son remplazados por el Cliente de la red TCP y los esclavos son conocidos como los Servidores (M-System CO. LTD., 2002).

Modbus-TCP básicamente encapsula la trama de datos Modbus dentro de su trama TCP, omitiendo los campos incensarios como: *Start*, *CRC* o *LRC* y *END*, ya cubiertos por las tramas *Ethernet* (véase Figura 1.31). La comprobación de datos corruptos en un modelo Consulta-Respuesta es detectada usando TCP/IP o mecanismos de verificación de la capa de enlace (M-System CO. LTD., 2002).

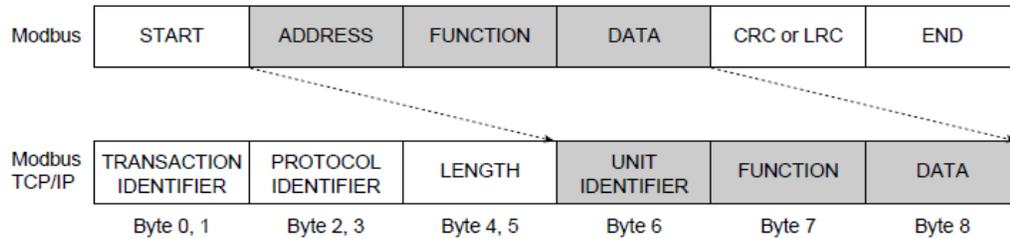


Figura 1.31: Trama de Datos en Modbus-TCP/IP

Fuente: (M-System CO. LTD., 2002)

El determinismo del protocolo es un aspecto que ha sido mejorado en las redes *Ethernet*, estándares como Profinet y Modbus-TCP/IP poseen dicha característica con el fin de garantizar que los mensajes sean enviados y recibidos en tiempos finitos predecibles (Acromag, 2005).

El uso de este protocolo de comunicación en las redes de automatización presenta varios beneficios al sistema, según lo expuesto por (Olaya, Barandica, & Guerrero, 2004) se puede resumir lo siguiente:

- Es escalable en complejidad y tecnología.
- Simple de administrar y expandir mediante la incorporación de nuevas estaciones Modbus-TCP.
- Es una tecnología abierta, no es necesario *software* propietario.
- Se puede implementar para comunicaciones con una base de dispositivos Modbus instalados.
- Es de alto desempeño.

## 1.5 CONCLUSIONES

Luego de la culminación del presente capítulo se logró un estudio relativamente amplio de las comunicaciones industriales, en el cual se ha presentado varias de las características de las mismas; los temas abordados permiten una correcta comprensión acerca de

protocolos industriales, lo cual brinda herramientas de análisis al momento de contrastar e implementar las diferentes redes. Se puede destacar los siguientes puntos.

- Las comunicaciones industriales se han desarrollado desde los inicios mismos de las fábricas con el principal objetivo de mejorar los procesos productivos, y permitir contar con sistemas de automatización eficientes, seguros, controlables que incrementen la productividad y disminuyan los tiempos de parada o fallas en los conglomerados empresariales.
- Si bien es cierto a lo largo de todo el mundo se ha tratado de solucionar los problemas de las comunicaciones industriales para lo cual cada fabricante ha enfrentado este reto de distinta manera, pero existe un modelo estandarizado que permite agrupar los elementos y medios de comunicación según el nivel de automatización en el que se encuentren, este modelo se lo conoce como la Pirámide CIM de las Comunicaciones Industriales y es un punto de partido para la comprensión de los requerimientos e implementaciones que se tendrá un proceso de control.
- En la redes Industriales se emplea las topologías heredadas de las redes LAN como son principalmente: estrella, bus, lineal, árbol, anillo y punto a punto que permiten la conexiones de todo tipo de dispositivos en la industria.
- Existen un gran número de buses de campo que se utilizan en la actualidad como: Hart, AS-I, Modbus, Devicenet, CAN, Profibus, *Fieldbus Foundation*, Interbus, Profinet entre otros, estos estándares son utilizados conjuntamente con varios protocolos de comunicación y modelos de arquitecturas diferentes, pero se puede agrupar básicamente ciertos buses según sus prestaciones o áreas de automatización a las que están enfocadas sus transmisiones. Así se pueden distinguir las redes para sensores y actuadores, las redes de campo para comunicar controladores con elementos de planta, las redes de gestión y las redes de seguridad intrínseca.

- Luego de un análisis de las redes de comunicación se puede destacar algunos estándares que entre otras bondades permiten conectar la mayor parte de las áreas productivas de un proceso y son los más usadas a nivel mundial como: Profibus, Profinet y Modbus.
- La red más ampliamente usada para el despliegue de dispositivos de campo es Profibus-DP, este estándar permite la conexión de controladores con elementos como: sensores, actuadores, válvulas, *drivers* de motores o periferia descentralizada en general, mediante un bus de uso industrial con apantallamiento y velocidades de transmisión elevadas hasta 12 Mbits/s.
- Un aspecto importante acerca de Profibus es que puede ser usado para ambientes de seguridad intrínseca gracias a su variante Profibus-PA.
- Una de las principales redes que permite la conexión de todo tipo de dispositivos es Profinet, este estándar es uno de los más usados en las redes de planta y gestión de las fábricas ya que implementa el protocolo TCP/IP, cuyo infraestructura en el mundo tiene un amplio historial facilitando las comunicaciones con redes externas alámbrica o inalámbricamente.
- Una red que permite la conexión de controladores con dispositivos de periférica inteligente como *drivers*, sistemas de control de movimiento y además con sistemas HMI y PCS es Profinet, este estándar implementa comunicaciones TCP/IP, RT, IRT por lo cual se convierte en un medio de altas funcionalidades para todo tipo de sistemas industriales, es la red con la más alta velocidad (100 Mbits/s) lo cual la hace ideal para sistemas de alta precisión.
- Otra de las principales redes industriales ocupadas mayormente es Modbus, cuya tecnología abierta permite el desarrollo de varias aplicaciones y cuyos protocolos como ASCII, RTU y TCP/IP son usados en una gran cantidad de dispositivos inteligentes.

- Una red que permite el despliegue de comunicaciones en un entorno de *intranet* y *extranet* es Modbus, la cual es usada tanto para control como para supervisión de los procesos, con una posibilidad de manejo de múltiples esclavos o mediante un modelo cliente-servidor.
- Las redes industriales permiten la integración de todas las áreas empresariales mediante diferentes protocolos, arquitecturas, medios de transmisión, etc. A continuación se presenta una tabla resumen acerca de las características de redes como Profibus, Profinet y Modbus.

Tabla 1.8 Tabla Resumen: redes Profibus, Profinet y Modbus

	<b>Profibus</b>	<b>Profinet</b>	<b>Modbus</b>
<b>Control de Acceso al medio</b>	Maestro-Esclavo	CSMA/CD CSMA/CA	Maestro-Esclavo Cliente-Servidor
<b>Medios de Transmisión</b>	RS-485 Fibra Óptica	Cable <i>Ethernet</i> Fibra Óptica Señales Electromagnéticas	RS-232 RS-485 Cable <i>Ethernet</i>
<b>Topología</b>	Bus	Estrella	Bus, Estrella
<b>Variantes</b>	Profibus-DP Profibus-PA Profibus-FMS	Profinet-I/O Profinet-CBA	Modbus-RTU Modbus-ASCII Modbus-TCP/IP
<b>Velocidad max.</b>	DP - 12Mbits/s PA – 31.25 Kbits/s	100Mbits/s	RTU - 115200 baudios/s. TCP/IP – 100 Mbits/s
<b>Mayor Área de Aplicación</b>	Redes entre controladores y accionamientos	Redes de controladores, de Gestión y de planta	Redes de controladores y accionamientos, además redes de gestión.

## CAPÍTULO 2

### CONFIGURACIÓN DEL PLC SIMATIC S7-1200 PARA LOS SISTEMAS DE AUTOMATIZACIÓN A UTILIZAR

#### Introducción

En el presente capítulo se aborda temas relacionados a dos sistemas de automatización industrial que serán implementados conjuntamente con las redes de comunicación. En una primera parte se realiza un estudio para el control PID del nivel de líquido de un Tanque, en este apartado se trata temas como: los elementos del sistema, los principios de la regulación PID, control mediante el objeto Tecnológico PID\_COMPACT para el PLC Simatic S7-1200 y un método de sintonización para los parámetros proporcional, integrador y derivador. En la segunda parte del capítulo se realiza una simulación de sistemas industriales de manufactura mediante el *software FACTORY I/O*, en este caso se presenta brevemente herramientas para el diseño de una aplicación virtual y posteriormente se realiza la comunicación y programación del sistema a través de un PLC Siemens.

#### 2.1 SISTEMA DE AUTOMATIZACIÓN REAL

##### 2.1.1 Elementos del Sistema

###### 2.1.1.1 Simatic S7-1200

El controlador lógico programable S7-1200 es un dispositivo perteneciente a la familia Simatic S7 de la empresa alemana SIEMENS, es un equipo diseñado para el control industrial de procesos de gama media y alta, su flexibilidad y diseño compacto pero robusto lo hace ideal para una gran variedad de aplicaciones. Conjuntamente con módulos y funciones de objetos tecnológicos se puede establecer redes de comunicación con dispositivos de campo u otros controladores.

La CPU posee un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, interfaces de comunicación, una ranura para memorias externas; lo que conforma un dispositivo completo para tareas de automatización. La CPU se encarga de la ejecución del programa diseñado, vigila las entradas y se establece los niveles de salida según la lógica deseada, posee instrucciones de conteo y temporización, funciones matemáticas, funciones binarias, instrucciones de comunicación industrial entre otras (SIEMENS, S7 Controlador programable S7-1200, 2015). Las partes principales de las que está formada una CPU son las siguientes (véase Figura 2.1).

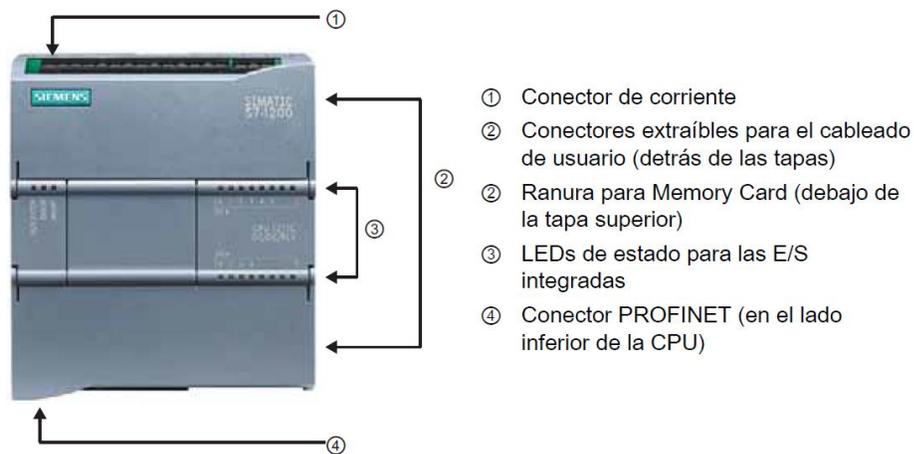


Figura 2.1: Partes Principales de Simatic S7-1200

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

Además una CPU posee ciertas funciones de seguridad orientadas a la protección de datos y programación del dispositivo, mediante la cual se puede establecer restricción por contraseña para limitar el acceso a sus funciones, o medidas de "know-how" que con el fin ocultar el código fuente de un determinado bloque (SIEMENS, S7-1200 Getting Started del S7-1200, 2009).

Las CPUs de los controladores S7-1200 tienen básicamente tres versiones con ciertas características específicas que permite la elección de las mismas dependiendo los requerimientos de entrada-salida, memorias de usuario, posibilidad de ampliación con

módulos de señales, contadores rápidos, etc. En la siguiente tabla se puede destacar las principales características físicas de las CPU 1211C, 1212C y 1214C.

Tabla 2.1: Características Principales de las CPUs S7-1200

Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75		110 x 100 x 75
Memoria de usuario	<ul style="list-style-type: none"> <li>• Memoria de trabajo</li> <li>• Memoria de carga</li> <li>• Memoria remanente</li> </ul>		<ul style="list-style-type: none"> <li>• 50 KB</li> <li>• 2 MB</li> <li>• 2 KB</li> </ul>
E/S integradas locales	<ul style="list-style-type: none"> <li>• Digitales</li> <li>• Analógicas</li> </ul>	<ul style="list-style-type: none"> <li>• 6 entradas/4 salidas</li> <li>• 2 entradas</li> </ul>	<ul style="list-style-type: none"> <li>• 8 entradas/6 salidas</li> <li>• 2 entradas</li> <li>• 14 entradas/10 salidas</li> <li>• 2 entradas</li> </ul>
Tamaño de la memoria imagen de proceso	1024 bytes para entradas (I) y 1024 bytes para salidas (Q)		
Área de marcas (M)	4096 bytes		8192 bytes
Ampliación con módulos de señales	Ninguna	2	8
Signal Board	1		
Módulos de comunicación	3 (ampliación en el lado izquierdo)		
Contadores rápidos	3	4	6
<ul style="list-style-type: none"> <li>• Fase simple</li> <li>• Fase en cuadratura</li> </ul>	<ul style="list-style-type: none"> <li>• 3 a 100 kHz</li> <li>• 3 a 80 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 3 a 100 kHz</li> <li>• 1 a 30 kHz</li> <li>• 3 a 80 kHz</li> <li>• 1 a 20 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 3 a 100 kHz</li> <li>• 3 a 30 kHz</li> <li>• 3 a 80 kHz</li> <li>• 3 a 20 kHz</li> </ul>
Salidas de impulsos	2		
Memory Card	SIMATIC Memory Card (opcional)		
Tiempo de respaldo del reloj de tiempo real	Típico: 10 días / Mínimo: 6 días a 40 °C		
PROFINET	1 puerto de comunicación Ethernet		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Velocidad de ejecución booleana	0,1 µs/instrucción		

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

Los PLCs de la familia SIMATIC basan su principio de operación en CPUs con prestaciones básicas orientadas a la conexión de módulos, esto con el fin de ampliar las prestaciones del dispositivo y descentralizar la programación, además de facilitar la detección y corrección de fallas al poder simplemente reemplazar los elementos defectuosos o incorporar nuevas funciones al sistema. Existe una gran variedad de posibilidades de expansión: con módulos de entradas-salidas digitales y analógicas de diferentes características, tarjetas de señales, módulos de comunicación punto a punto como RS-485, RS-232 y otros orientados a conexión de redes Profinet, Profibus y

Modbus, además se puede añadir funciones de telecontrol<sup>38</sup> y teleservicio<sup>39</sup> para redes alámbricas e inalámbricas, entradas de señales estándar industriales como termocuplas, RTD, etc. A continuación se presenta una tabla resumen acerca de las oportunidades de crecimiento para estos controladores.

Tabla 2.2: Módulos Adicionales para CPUs S7-1200

Módulo		Sólo entradas	Sólo salidas	Entradas y salidas
Módulo de señales (SM)	Digital	8 entradas DC	8 salidas DC 8 salidas de relé	8 entradas DC/8 salidas DC 8 entradas DC/8 salidas de relé
		16 entradas DC	16 salidas DC 16 salidas de relé	16 entradas DC/16 salidas DC 16 entradas DC/16 salidas de relé
	Analógico	4 entradas analógicas 8 entradas analógicas	2 salidas analógicas 4 salidas analógicas	4 entradas analógicas/2 salidas analógicas
Signal Board (SB)	Digital	-	-	2 entradas DC/2 salidas DC
	Analógico	-	1 salida analógica	-
Módulo de comunicación (CM)				
<ul style="list-style-type: none"> <li>• RS485</li> <li>• RS232</li> </ul>				

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

Simatic S7-1200 maneja tres tipos de bloques lógicos: los bloques de Organización (OBs) que definen la estructura del programa, las funciones (FCs) bloques de función (FBs) que contiene programación de tareas específicas y los bloques de datos (DBs) que almacenan la información. Los bloques pueden ser programados mediante lenguaje KOP (Esquema de contactos) o FUP (Diagrama de funciones) (SIEMENS, S7 Controlador programable S7-1200, 2015).

Las áreas de memoria que posee este PLC son tres: la primera es la Memoria Global la cual contempla áreas de entradas (I), salidas (Q) y marcas (M), la segunda son los bloques de datos (DB) globales o de instancia para ser accedidos mediante los bloques lógicos, y la tercera es la Memoria Temporal que se encarga de gestionar la ejecución de

<sup>38</sup> Telecontrol: Transmisión de órdenes de control a distancia a través de un enlace.

<sup>39</sup> Teleservicio: Transmisión de funciones de diagnóstico y mantenimiento remoto.

instrucciones llamadas desde el programa central (SIEMENS, S7 Controlador programable S7-1200, 2015).

### **2.1.1.2 Entradas Analógicas de la CPU**

Las CPUs S7-1200 cuentan con dos entradas analógicas incorporadas en el equipo, estas entradas son solamente de voltaje en un rango de 0-10V, en el caso que se desee agregar más, se puede hacer uso de los módulos de entradas y salidas analógicas en sus diferentes variantes.

El voltaje de las entradas analógicas de la CPU permite obtener registros digitales en un rango de palabras de 0-27648 que corresponde a la tensión de 0-10V, su resolución es de 10 bits y soportan una tensión máxima de 35Vdc. Adicionalmente poseen un nivel de filtrado o valor medio de muestreo: nulo, débil, medio y fuerte que corresponde a 1, 4, 16 y 32 ciclos de promedio respectivamente. Para frecuencias de 10, 50 o 60Hz las entradas incluyen supresión de ruido (SIEMENS, S7 Controlador programable S7-1200, 2015).

### **2.1.1.3 Sensor de Ultrasonido**

Los sensores de ultrasonido son ampliamente usados en el sector industrial y basan su funcionamiento en ondas ultrasónicas, éstas son transmitidas por el sensor hacia el medio circundante y si un objeto se interpone en su propagación entonces se produce una onda reflejada que regresa al transductor, y en función del tiempo que ha transcurrido desde la emisión hasta la recepción se puede calcular la distancia entre el objeto y el dispositivo. La luz y el sonido son dos fenómenos naturales que permiten esta medición de distancia sin estar en contacto con los elementos del ambiente, independientemente del material o el color que tengan. Para el desarrollo de este trabajo se utilizará un sensor de ultrasonido de marca SICK que nos permite realizar mediciones de alta precisión, cuyas aplicaciones típicas son: medición de nivel de sólidos y líquidos, detector de presencia, control de asignación en el empaquetado, control de diámetro.

El sensor ultrasónico SICK UM 30 está diseñado para trabajar en ambientes hostiles incluso con suciedad, partículas en el aire, niebla, vapor, etc. El sensor puede ser configurado con una salida binaria o analógica, existe tres versiones de este dispositivo con diferentes especificaciones en este caso se ocupará la versión UM 30 para un rango de 350 a 3400 mm. Los datos más relevantes según lo presentado por (SICK AG, 2012) son los siguientes:

- Rango: 350 a 3400mm
- Frecuencia de la onda: 120 KHz.
- Resolución: 1mm
- Precisión:  $\leq 2\%$  del valor final
- Fuente de voltaje: DC 12 a 30 V
- Salida Analógica:  $Q_a = 4$  a 20 mA / 0 a 10V
- Tiempo de respuesta: 180ms
- Conector: Plug M12, 5 pines
- Temperatura del Ambiente: Operación (-20 °C a 70°C) y Almacenamiento (-40°C a 85°C)
- Consumo de corriente:  $\leq 60$  mA
- Salida de conmutación reversible: Q = PNP o NPN

El sensor posee un panel de control-monitoreo simple conformado por dos botones y dos indicadores LEDES que permiten establecer las configuraciones básicas para el dispositivo, la manera de cableado con el conector M12 a 5 pines es la que se expone a continuación (véase Figura 2.2).



Figura 2.2: Sensor UM 30 y forma de conexión

Fuente: (SICK AG, 2012)

#### 2.1.1.4 Bomba Hidráulica

La bomba hidráulica es un elemento que transforma la energía mecánica en cinética de fluido con el fin de moverlo, los fluidos son generalmente incompresibles como líquidos o mezcla con sólidos los cuales son usados en sistemas hidráulicos, existen varios tipos de bombas pero en general basan su funcionamiento en incrementar la presión del fluido aumentando su velocidad y altura con el fin de generar movimiento desde una zona de menor a una de mayor presión, en este caso la altura está directamente relacionada con la presión del líquido.

Para el sistema de circulación de agua entre dos tanques se usará una bomba de accionamiento eléctrico de tipo roto dinámica o también conocida como centrífuga (véase Figura 2.3). Este tipo de bomba es de corriente continua básicamente utilizada en aplicaciones de bombeo a base de energía fotovoltaica, las principales características según lo presentado en (XCSOURCE, 2006) son las siguientes:

- Fuente de voltaje: 12 /24 VDC
- Tasa de Corriente: 500/800 mA
- Altura de descarga: 3m /5m
- Tasa de flujo: 480/500 L/H
- Toma/Salida: Cable Macho 1/2"



Figura 2.3: Bomba Hidráulica

Fuente: (XCSOURCE, 2006)

## 2.1.2 Diseño y Construcción del Sistema

### 2.1.2.1 Diagrama de Instrumentación de la Planta

El prototipo diseñado para el control PID del nivel de líquido de un tanque basa su funcionamiento en la recirculación de agua entre dos contenedores con una regulación de flujo determinada por el algoritmo PID. El primer tanque es el elemento de interés mientras que el segundo es el elemento que reserva y provee agua al sistema.

El proceso contiene los siguientes elementos:

- Dos tanques de metal.
- Dos válvulas de paso.
- Una bomba DC.
- Un sensor de ultrasonido.
- Un controlador S7-1200.
- Un circuito para la activación de la bomba.

El diagrama de instrumentación (P&ID) de la planta se expone en la Figura 2.4, mediante este se puede ejemplificar la distribución de los elementos involucrados, sus respectiva función y la variable física en cada caso.

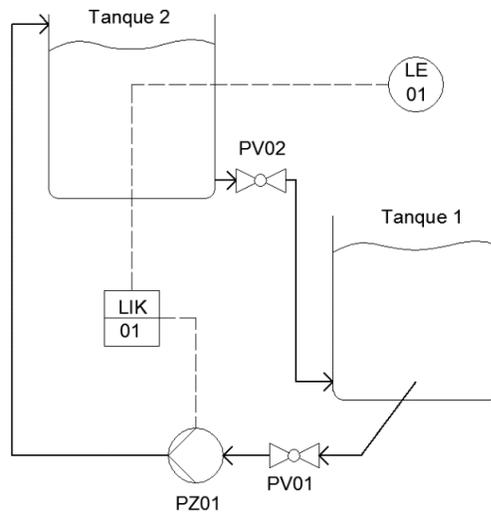


Figura 2.4: Diagrama P&ID de la Planta

La simbología del diagrama P&ID se muestra en la tabla siguiente:

Tabla 2.3: Simbología del Diagrama P&ID

N°	Nomenclatura	Denominación
1	LE01	Sensor de Nivel con indicación y transmisión
2	LIK01	Estación de control de Nivel de Líquido
3	PZ01	Bomba de Presión
4	PV01	Válvula de cierre
5	PV02	Válvula de cierre

El prototipo construido es el que se muestra en la Figura 2.5.



Figura 2.5: Prototipo Construido

### 2.1.2.2 Esquema de Conexión del Proceso

Luego de establecer el diagrama de instrumentación del proceso es necesario realizar un esquema de las conexiones físicas que tendrán los elementos de control. Como se explicó en apartados anteriores el sensor de ultrasonido posee una salida de 0-10V dependiendo su distancia hacia los objetos, esta señal de tensión es conectada a una de las entradas analógicas de la CPU. En el PLC se programa los bloques e instrucciones necesarias para el algoritmo PID y la señal de salida de control (PWM) se conecta al circuito de acople para la activación de la bomba (véase Figura 2.6).

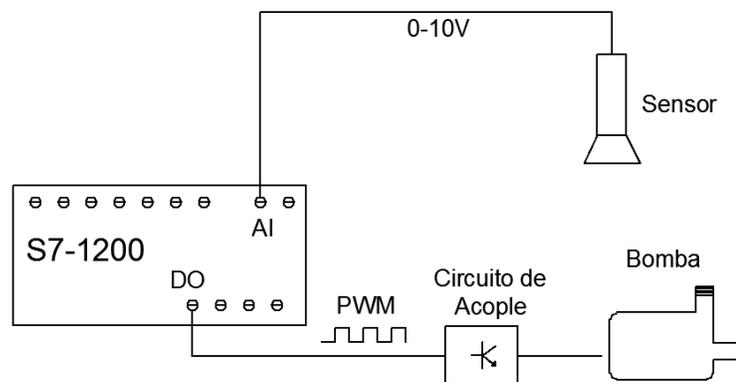


Figura 2.6: Esquema de Conexión del Proceso

Un PLC S7-1200 en su versión DC/DC/DC posee salidas digitales o de transistor, solamente este tipo de salidas pueden ser conmutadas para un control PWM. Es posible conectar una fuente independiente con el fin de obtener una señal de control con una tensión determinada. En este caso la salida PWM será conmutada entre 0 y 24V.

La bomba tiene una tensión de alimentación de 24VDC con una corriente de 800mA aproximadamente, por lo tanto la activación se la realizará mediante un Mosfet de canal N.

Se ha escogido el Mosfet IRF540 el cual es activado por una diferencia de voltaje entre sus terminales *Gate-Source*, la máxima tensión es 20VDC y el valor recomendado para

una saturación completa del Mosfet está sobre los 7V. Por lo tanto se realiza un partidor de tensión para disminuir la salida PWM a la mitad y asegurar la saturación del Mosfet.

La fórmula para el partidor de tensión es la siguiente:

$$V_s = \frac{V_e * R_1}{R_1 + R_2}$$

$V_p$ : Voltaje salida (V).

$V_e$ : Voltaje de entrada (V).

$R_1, R_2$ : Resistencias del partidor ( $\Omega$ ).

Se asume una resistencia  $R_1$  de  $1K\Omega$  y se mediante la ecuación se calcula el valor para la resistencia  $R_2$ .

$$R_2 = \frac{V_e * R_1}{V_s} - R_1$$

$$R_2 = \frac{24 * 1K}{12} - 1K$$

$$R_2 = 1K\Omega$$

En la Figura 2.7 se muestra el esquema del circuito de activación:

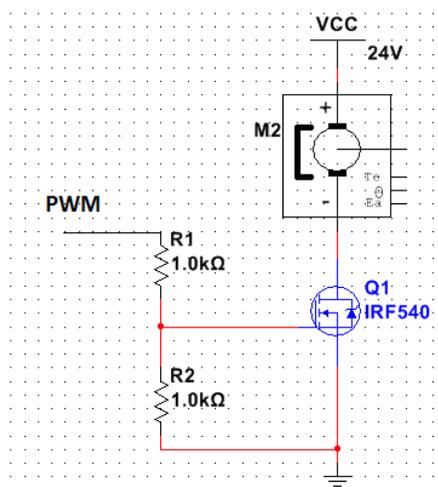


Figura 2.7: Esquema del Circuito de Acople

La tarjeta de acople construida es la siguiente:

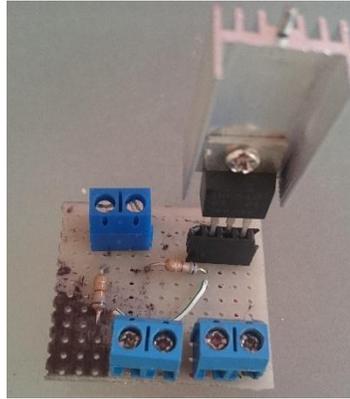


Figura 2.8: Tarjeta Circuito de Acople

### **2.1.3 Control PID (Fundamentos Teóricos)**

#### **2.1.3.1 El controlador PID**

El controlador PID es un algoritmo utilizado ampliamente en la actualidad en un gran número de procesos, basa su funcionamiento en un sistema de lazo realimentado que permite establecer el error entre la señal de referencia o consigna y la señal de salida de la planta. Este error es medido y utilizado para el cálculo de la señal de ajuste mediante tres acciones: la proporcional, la integradora y la derivativa (karl & Tore, 2009).

Los algoritmos de control PID son adecuados para dar solución a la mayoría de aplicaciones en la industria cuando los sistemas a intervenir presentan un comportamiento que puede ser descrito por dinámicas de primer y segundo orden. Su uso alcanza niveles del 95% de los lazos de control que se requieren en dichos ambientes siendo predominante el control PI (Améstegui, 2011).

El algoritmo de control PID puede ser descrito mediante un lazo de realimentación como se muestra en la Figura 2.9.

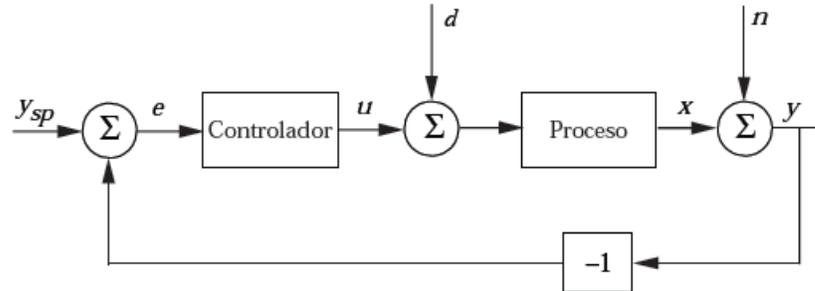


Figura 2.9: Lazo de Realimentación

Fuente: (karl & Tore, 2009)

Donde la  $\mu$  es la señal de control y  $e$  es el error calculado a través de la diferencia de la señal de referencia  $Y_{sp}$  y la señal de salida  $Y$ .

La señal de control es el resultado de las tres acciones mencionadas: la proporcional (calculada proporcionalmente al error), la integradora (calculada en base a la integración de los errores pasados) y la derivativa (calculada a través de la derivada del error). Los parámetros que estas acciones representan son: la ganancia proporcional  $K$ , el Tiempo de integración  $T_i$  y el tiempo derivativo  $T_d$  (karl & Tore, 2009).

En base a lo presentado por (karl & Tore, 2009) y (Améstegui, 2011) se puede resumir lo siguiente en cuanto a las tres acciones del control PID.

#### a. Acción Proporcional

Esta acción cobra su importancia en el hecho de que si la señal de error sufre un ligero cambio, un control tradicional *ON/FF* desencadenaría en oscilaciones permanentes sobreactuadas.

Por lo tanto la acción proporcional determina un comportamiento dependiente de la magnitud del error, se la puede expresar a través de la siguiente ecuación.

$$u(t) = K(t) + u_b$$

Donde  $u_b$  es una variable de polarización o *reset* cuando no existe error.

Mediante la acción proporcional se tiene en la mayoría de los casos un error estático o de estado estacionario diferente de 0.

### b. Acción Integral

La acción integral depende de los errores pasados del sistema y su función es regular el error de estado estacionario que se presenta en el control proporcional, de esta manera un cambio positivo en la señal de error desencadenará en una señal de control creciente y un cambio negativo producirá un control decreciente.

La acción integral restablece constantemente la señal de control mientras exista error, puede ser representada mediante el siguiente diagrama de bloques.

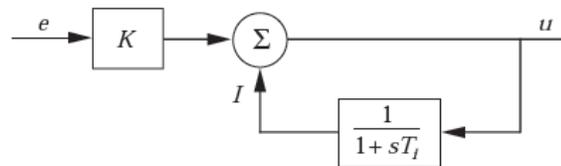


Figura 2.10: Acción Integral

Fuente: (karl & Tore, 2009)

La ecuación en función del tiempo viene dada por:

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int e(\tau) d\tau \right)$$

### c. Acción Derivativa

La acción derivativa de un proceso busca mejorar la estabilidad del lazo cerrado mediante la predicción de errores futuros. Los sistemas tienen una cierta dinámica que depende de

los instrumentos de medición y control por lo cual existe un tiempo que transcurre hasta que la variación en la salida de proceso sea notada, por lo tanto se genera un retardo en la acción correctiva del error. La acción derivativa está diseñada para predecir el error en base a la extrapolación de la curva característica del mismo en la dirección de la tangente.

De esta manera la ecuación en función del tiempo de una estructura PD es la siguiente:

$$u(t) = K \left( e(t) + T_d \frac{d e(t)}{dt} \right)$$

Por lo tanto la señal de control va a ser proporcional a una estimación del error en un tiempo  $T_d$ .

### **2.1.3.2 Sintonización PID (Método de Ziegler-Nichols)**

El diseño de los controladores PID puede ser determinado a través del modelo matemático de la planta que se desea intervenir, en este caso es posible aplicar técnicas de diseño para establecer el mejor juego de parámetros tanto para un estado transitorio como un estado estable. En el caso que no se pueda determinar el modelo matemático existen ciertos métodos que permiten seleccionar los parámetros del controlador según las características de desempeño, a este proceso se conoce como sintonización de los controladores PID. (Ogata, 1998).

En su estudio Ziegler-Nichols propusieron unas reglas para sintonizar los controladores PID determinando el valor de la Ganancia Proporcional  $K_p$ , del Tiempo de Integración  $T_i$  y del Tiempo Derivativo  $T_d$  en base a la respuesta transitoria de un sistema específico (Alfaro, 2005). A continuación según lo expuesto por (Ogata, 1998) y (Alfaro, 2005) se puede resumir los siguiente acerca de los métodos de sintonización de Ziegler-Nichols.

### a. Primer Método

El primer método trata de establecer experimentalmente la respuesta del sistema a una entrada de escalón unitario, la curva de respuesta generalmente tiene forma de S si el proceso no contiene integradores o polos dominantes conjugados (véase Figura 2.11). La curva se puede obtener a partir de una simulación dinámica de la planta.

Esta función de respuesta se caracteriza por dos parámetros: El Tiempo de Retardo  $L$  y la constante de tiempo  $T$ , los cuales son determinados a través de las intersecciones de la tangente en el punto de inflexión de la curva con el eje de tiempo y la línea  $c(t) = K$ .

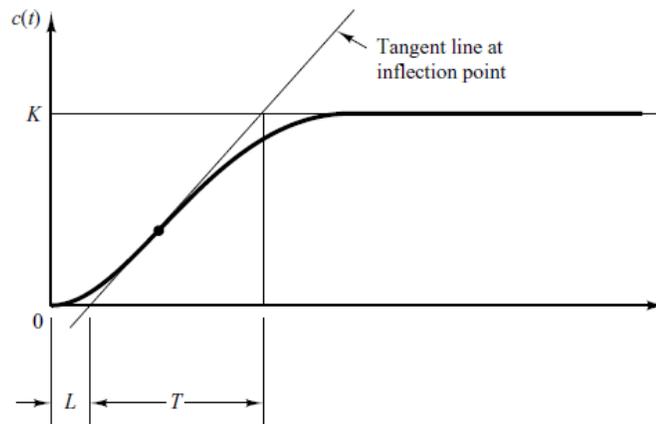


Figura 2.11: Curva de Respuesta al Escalón Unitario

Fuente: (Ogata, 1998)

La función de transferencia se puede aproximar a un sistema de primer orden:

$$\frac{C(s)}{U(s)} = \frac{K_e^{-Ls}}{T_s + 1}$$

Los parámetros para  $K_p$ ,  $T_i$  y  $T_d$  sugeridos por Ziegler-Nichols se exponen en la siguiente Tabla.



Tabla 2.5: Parámetros Ziegler-Nichols, Segundo Método

Tipo de Controlador	$K_p$	$T_i$	$T_d$
<b>P</b>	$0.5K_{cr}$	$\infty$	0
<b>PI</b>	$0.45K_{cr}$	$P_{cr}/1.2$	0
<b>PID</b>	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Fuente: (Ogata, 1998)

### 2.1.4 Modelo Matemático del Sistema

Según lo presentando por (Ogata, 1998) se puede establecer el modelo matemático para un tanque de la siguiente manera. De acuerdo a la Figura 2.13 se tiene un sistema con las siguientes variables:

Q: Flujo en estado estable ( $m^3/s$ )

$q_i$ : Desviación del flujo de entrada ( $m^3/s$ )

$q_o$ : Desviación del flujo de salida ( $m^3/s$ )

H: Altura en estado estable (m)

$h$ : Desviación de la altura

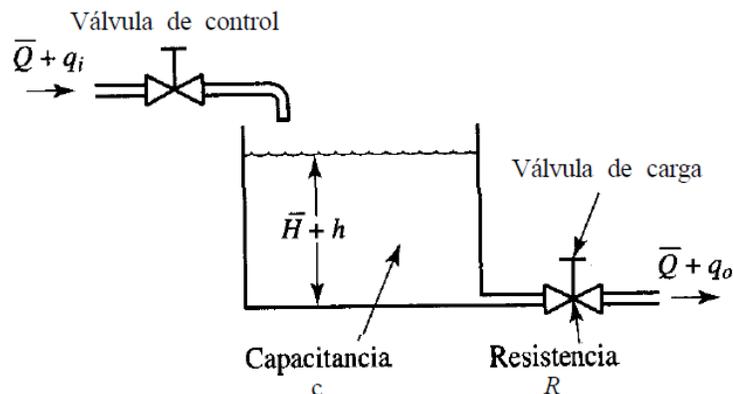


Figura 2.13: Sistema de Nivel de Líquido

Fuente: (Ogata, 1998)

Según el autor el flujo un sistema se considera lineal si el flujo es laminar, en su defecto si el flujo es turbulento, este puede ser linealizable siempre y cuando los cambios sean pequeños, a partir de esta suposición se establece lo siguiente:

$$C dh = (q_i - q_o) dt$$

Según la definición de resistencia para sistemas hidráulicos presentada por (Ogata, 1998) se tiene que:

$$q_o = \frac{h}{R}$$

Con lo cual reemplazando en la ecuación anterior:

$$C \frac{dh}{dt} = q_i - \frac{h}{R}$$

Resolviendo

$$R * C \frac{dh}{dt} + h = R * q_i$$

Aplicando Laplace

$$R * C * sH(s) + H(s) = RQ_i(s)$$

De esta manera despejando se obtiene la función de transferencia siguiente:

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs + 1}$$

Donde C representa la capacitancia o el área transversal del tanque y R la resistencia hidráulica de la válvula. El tanque del sistema tiene un radio de 17.5cm y la resistencia R se asumirá como 1, de esta manera la función de transferencia del sistema será:

$$G(s) = \frac{1}{0.0962s + 1}$$

### 2.1.5 Simulación y Sintonización PID Ideal mediante MATLAB

MATLAB es un poderoso *software* de ingeniería para el procesamiento numérico, dentro de sus bondades presentan un entorno de programación visual denominado SIMULINK mediante el cual se puede realizar simulación de sistemas de control conectando simples diagramas de bloques y configurándolos. A continuación se presentan el diagrama de bloques realizado en este *software* con el objeto de simular el control PID para la planta.

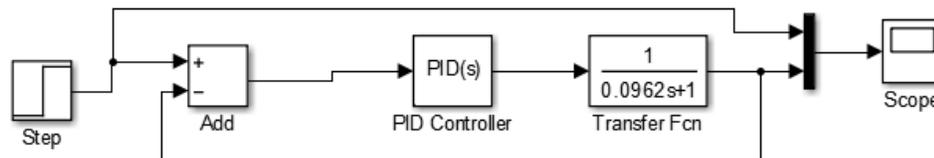


Figura 2.14: Diagrama de Bloques del Sistema

El sistema tiene como entrada un Escalón Unitario y está conectado al algoritmo de control PID, para determinar los parámetros ideales del controlador PID se utilizó la herramienta de *autotuning* de MATLAB, de esta manera las curvas obtenidas son las siguientes (véase Figura 2.15).

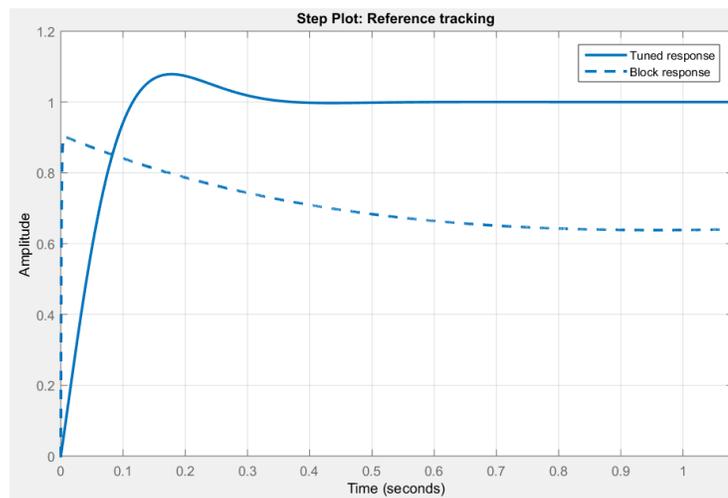


Figura 2.15: Sintonización PID con MATLAB

Los parámetros PID calculados se muestran en la Figura 2.16:

Controller Parameters		
	Tuned	Block
P	1.3376	1
I	27.7916	1
D	-0.004673	1

Figura 2.16: Parámetros PID obtenidos

Al ajustar los parámetros PID se obtiene una respuesta ideal del sistema cuya estabilidad se alcanza en 1s aproximadamente (véase Figura 2.17).

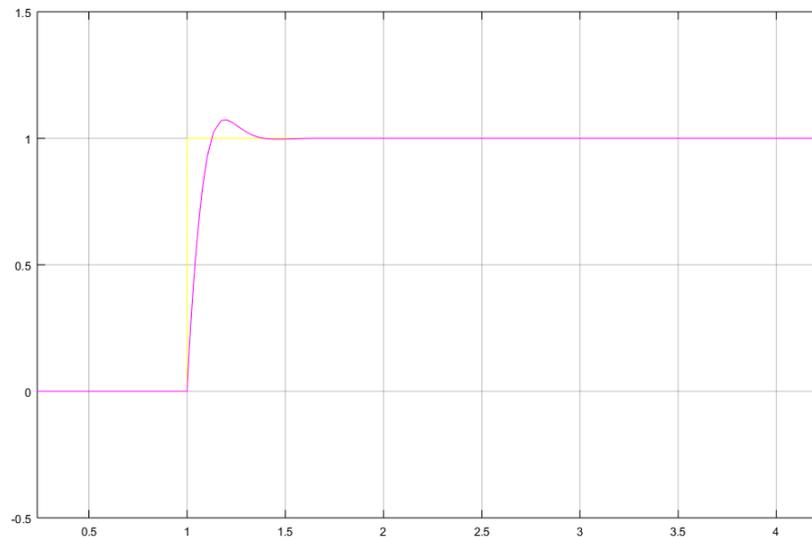


Figura 2.17: Curva de Respuesta del Sistema Simulado

## 2.1.6 PID con Simatic S7-1200

### 2.1.6.1 Generalidades

Las CPUs Simatic S7-1200 cuentan con un regulador PID general y continuo para el control de diferentes tipos de procesos industriales, los reguladores PID son objetos tecnológicos que se insertan, configuran y ponen en servicio a través de TIA PORTAL. Es posible insertar diferentes regulaciones PID según las entradas y salidas de la CPU lo permita.

El objeto PID\_Compact (Regulador PID Universal con optimización integrada) permite controlar un proceso mediante un algoritmo PID basado en las tres acciones: proporcional, integradora y derivativa, el cálculo de la salida de control se lo realiza en base a una medición continua del valor real del proceso (SIEMENS, S7 Controlador programable S7-1200, 2015).

El regulador PID\_Compact está incorporado además con una función de *Anti-Windup*<sup>40</sup> y ponderación de las acciones P y D, de acuerdo a lo presentado por (SIEMENS, S7 Controlador programable S7-1200, 2015) la función matemática del regulador es la siguiente:

$$y = K_p \left[ (b * w - x) + \frac{1}{T_i * s} (w - x) + \frac{T_d * s}{a * T_d * s + 1} (c * w - x) \right]$$

En donde:

y: Valor de Salida del algoritmo PID

K<sub>p</sub> : Ganancia Proporcional

s: Operador Laplaciano

b: Ponderación de la acción P

w: Consigna

x: Valor real

T<sub>i</sub> : Tiempo de integración

T<sub>d</sub>: Tiempo derivativo

a: Coeficiente de retardo de la acción derivativa

c: Ponderación de la acción D

Este bloque de regulación adicionalmente posee ciertas características que permiten su utilización versátil con diferentes señales de entrada o salida, además de alarmas de

---

<sup>40</sup> *AntiWindup*: Es una técnica para contrarrestar el efecto *windup*, que ocurre en un controlador PID cuando un cambio brusco en la referencia ocasiona una acumulación del error en la integral, este valor sigue incrementado.

proceso y límites de regulación. En la Figura 2.18 se presenta el diagrama de bloques de PID\_Compact.

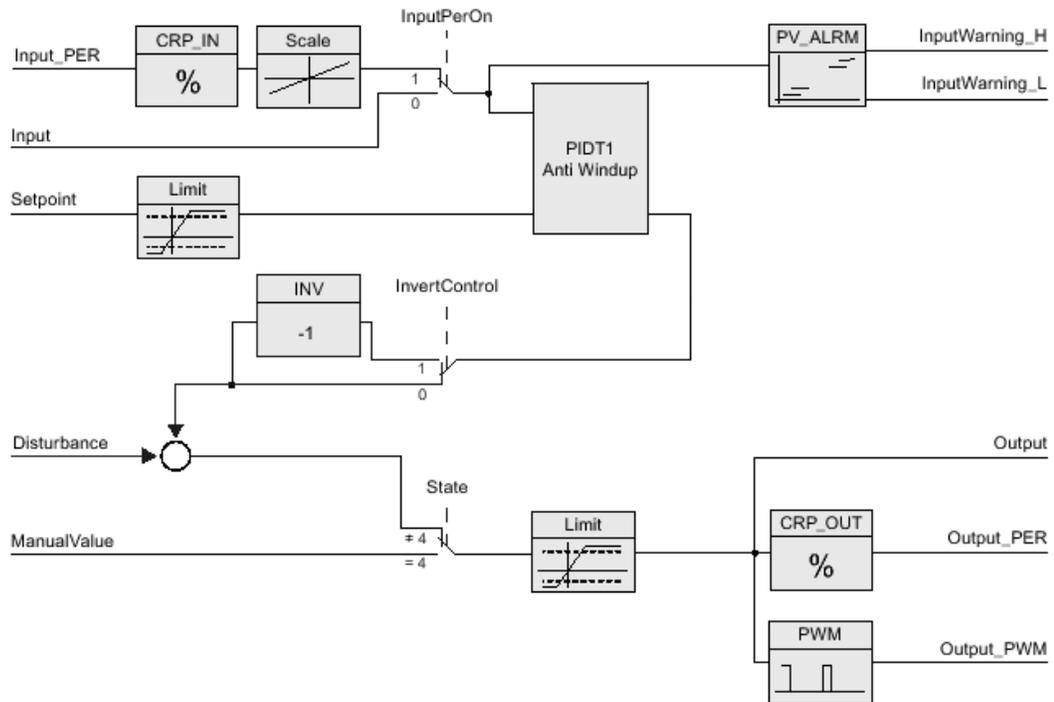


Figura 2.18: Oscilaciones Sostenidas en la Salida

Fuente: (SIEMENS, S7-1200, S7-1500 Regulación PID, 2014)

El regulador PID tiene 6 estados diferentes de funcionamiento mediante los cuales se puede establecer acciones de optimización o control, estos estados pueden ser conmutados en *Runtime* bajo las condiciones adecuadas, es decir, sin que provoquen un error en el bloque, los modos disponibles son:

- (0) Inactivo
- (1) Optimización Inicial
- (2) Optimización Fina
- (3) Modo Automático
- (4) Modo Manual
- (5) Valor de Salida Sustitutivo con monitorización de errores

### 2.1.6.2 PID\_COMPACT

#### a. Insertar PID\_Compact

El bloque PID\_Compact necesita un cierto tiempo para el cálculo de la salida a través de los valores de entrada y los parámetros cargados en él, por esta razón una llamada a una instrucción PID deberá realizarse a través de un bloque de ejecución cíclica que permita dicho retardo.

Para insertar un bloque de ejecución cíclica se deberá dar clic en la “Agregar Nuevo bloque” y escoger la opción “*Cyclic interrupt*”. De esta manera el PLC ejecuta este nuevo OB cada 100 ms.

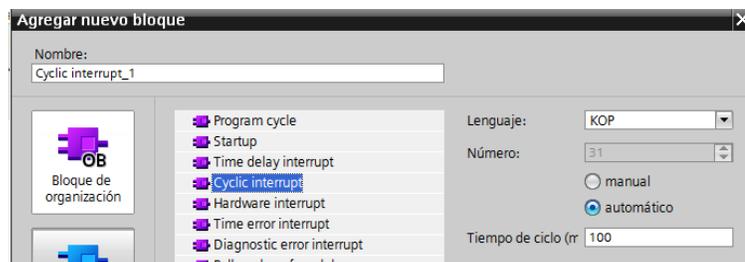


Figura 2.19: Creación de un Bloque de Ejecución Cíclica

La instrucción PID\_Compact deberá ser insertada dentro de este bloque de jerarquía con el fin de que el algoritmo mida el intervalo de tiempo entre dos llamadas y genere un resultado conforme al tiempo de muestreo.

#### b. Parámetros de la instrucción

La estructura PID\_COMPACT es la encargada de realizar el cálculo del algoritmo PID, a continuación según lo presentado por (SIEMENS, S7 Controlador programable S7-1200, 2015) se explica sus parámetros más relevantes de entrada y salida.

#### Parámetros de Entrada:

- SETPOINT: Corresponde a la consigna del regulador PID en el modo automático.
- INPUT: Entrada en el caso que se utilice un valor de programa como valor real de proceso.
- INPUT\_PER: Entrada para una variable analógica como valor real.
- MANUALENABLE: Un flanco ascendente cambia el regulador a modo manual.
- MANUALVALUE: Valor que tomará la salida en estado manual.
- RESET: Con un flanco ascendente realiza un arranque completo del regulador.
- MODEACTIVATE: Con un flanco ascendente el PID cambia al estado de la entrada MODE.
- MODE: Estado al que cambia el regulador (0-5).

#### Parámetros de Salida:

- OUTPUT: Valor de la salida como una variable de programa en formato REAL.
- OUTPUT\_PER: Valor de la salida como una variable analógica.
- OUTPUT\_PWM: Valor de la salida en formato PWM (Modulación de Ancho de Pulso).
- STATE: Indica el estado o modo en el cual se encuentra el bloque PID.

### 2.1.6.3 Configuración PID\_COMPACT

Las configuraciones del objeto tecnológico son realizadas sobre la ventana desplegada a dar doble clic sobre el mismo. En esta ventana se puede acceder a los valores del bloque en una vista funcional o en una vista de parámetros.

Mediante esta ventana también se podrá realizar una comparación de los parámetros en la PC con los valores guardados en la CPU, además crear instancias de datos de respaldo o interactuar en *Runtime* con las propiedades del controlador. A continuación se explica las configuraciones principales:

### a. Tipo de Regulación

La primera configuración que permite realizar el bloque PID es establecer el tipo de regulación a realizar, es decir, que magnitud física se va a controlar y en que unidades, existe una larga lista de posibilidades como: presión, longitud, caudal, temperatura, brillo, fuerza, masa, intensidad, tensión, etc (véase Figura 2.20). Además se podrá escoger un sentido inverso para la regulación y en qué modo arrancara el objeto al reiniciar la CPU.

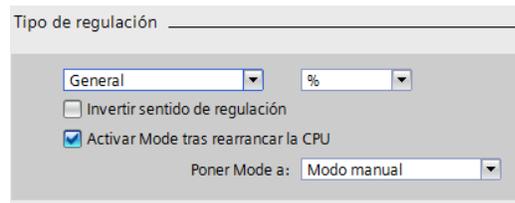


Figura 2.20: Tipo de Regulación

### b. Parámetros de Entrada

La segunda configuración que se debe realizar es escoger el tipo de entrada y salida del bloque, como se mencionó en el apartado anterior las opciones son: variables de proceso, entradas y salidas analógicas o salida de PWM.

Una entrada INPUT\_PER permite enlazar directamente el registro de una entrada analógica y posteriormente ofrece una ventana para escalar dicho valor a un rango deseado. Una entrada INPUT corresponderá a una variable del PLC que deberá ser acondicionada previamente a un valor adecuado para utilizarse como entrada del bloque (véase Figura 2.21).

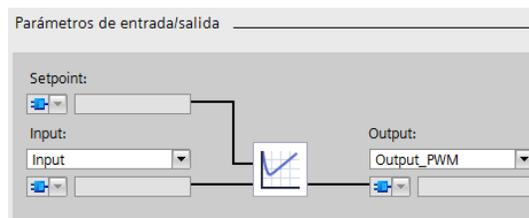


Figura 2.21: Parámetros de Entrada/Salida de la Regulación

### c. Escala del Valor Real

La escala del valor real permite establecer el valor máximo y mínimo en unidades de la magnitud física se eligió previamente, es decir, asocia el registro analógico del sensor con las unidades de la regulación. En la figura 2.22 se observa el número de 27648 que corresponde al máximo valor de la palabra de conversión digital para una entrada de 10V.

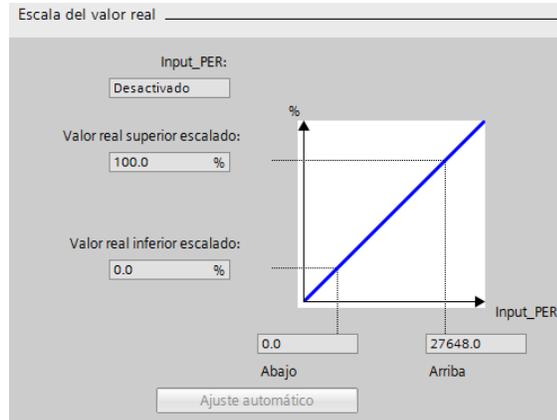


Figura 2.22: Parámetros de Entrada/Salida de la Regulación

### d. Límites del Valor de Salida

La ventana de configuración Límites del Valor de Salida permite establecer un rango adecuado dentro del cual la salida PID estará acotada, estos límites son importantes debido a que en la práctica los elementos a controlar tiene características finitas invariables como por ejemplo: un motor no puede girar más que su velocidad nominal, una válvula no se puede abrir más de sus límites mecánicos, etc. Por lo tanto el rango de salida que se escoja deberá ser tal que cause efecto directo sobre el circuito de control y a su vez sobre el actuador.

Adicionalmente se puede establecer un valor sustitutivo de salida en caso de que exista un error durante la ejecución del bloque.

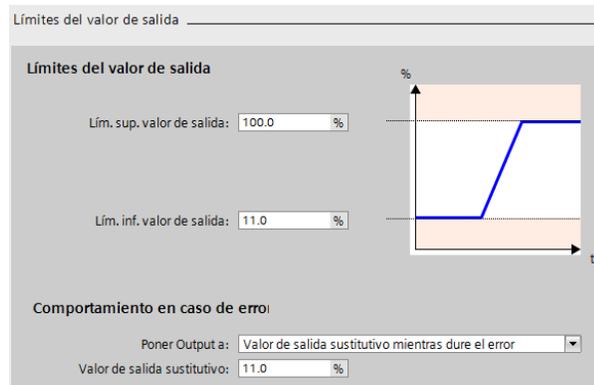


Figura 2.23: Límites del Valor de Salida

#### 2.1.6.4 Puesta en Servicio

Luego de establecer las configuraciones para el funcionamiento del lazo de regulación, el software TIA PORTAL ofrece una ventana para la puesta en servicio del objeto tecnológico. Esta ventana permite interactuar sobre el algoritmo de regulación para la optimización o sintonización de sus parámetros, para esto provee un área de curvas que permite visualizar las tres señales de interés del algoritmo como son: la referencia o *setpoint*, la señal de entrada o valor real y la señal de salida.

Antes de realizar cualquier acción sobre la puesta en servicio se debe establecer una conexión *online* con el PLC para acceder a los datos del PID en el controlador, luego de esto es necesario especificar el tiempo de muestreo de las ondas en un rango de 0.3 a 60s (véase Figura 2.24).

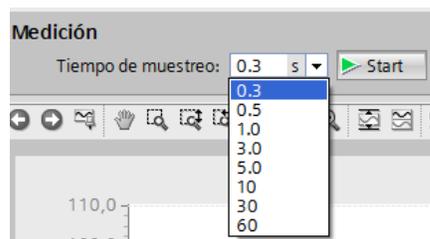


Figura 2.24: Tiempos de Muestreo Disponibles

El botón *Start* que se muestra en la Figura anterior inicia la visualización de las curvas en el gráfico, es decir, inicia la medición. Este visor posee ciertas herramientas para mejorar la comprensión de la gráfica como: *zoom*, cursores de medición, trazado de puntos, etc. En la siguiente Figura se ejemplifica la medición de los valores de las señales en un momento cualquiera.

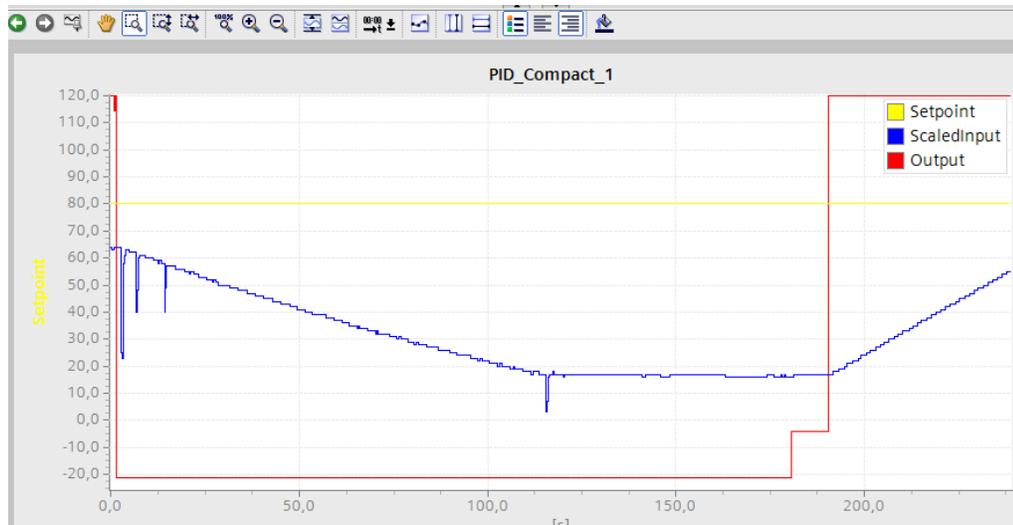


Figura 2.25: Visor de señales PID

### a. Optimización Inicial

La optimización inicial se basa en el primer método de Ziegler-Nichols y determina la respuesta del sistema a un escalón del valor de la salida, se busca el punto de inflexión y mediante la tangente en ese punto se establecen los parámetros PID. Se recomienda que el valor real de proceso sea lo más estable y este alejado del valor de la consiga (SIEMENS, S7-1200, S7-1500 Regulación PID, 2014).

En la sección inferior de la ventana se puede observar el estado de la regulación juntamente con los parámetros en ese instante (véase Figura 2.26).



Figura 2.26: Estado de la Optimización

En la Figura 2.27 se observa las gráfica de las señales del sistema durante la optimización inicial, como se puede constatar el controlador genera una escalón en la salida con el máximo porcentaje hasta que el valor medido de proceso alcance la consigna.

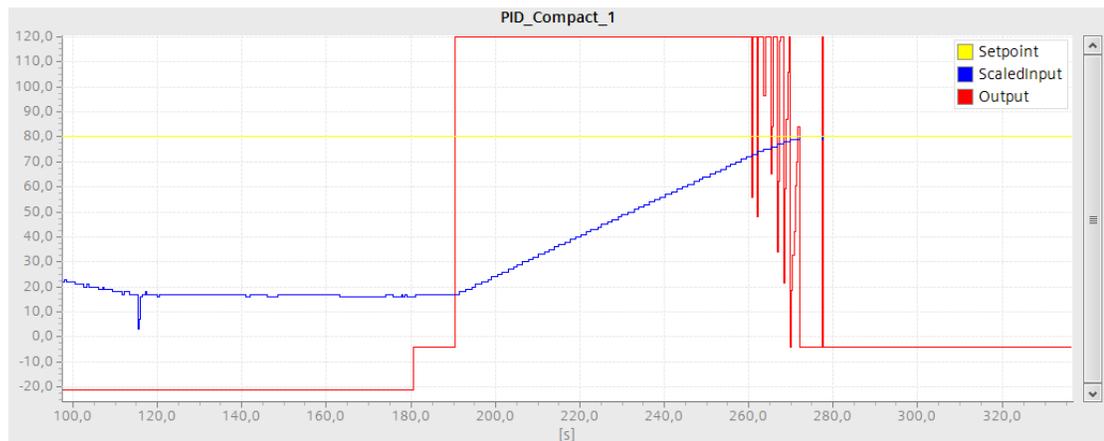


Figura 2.27: Optimización Inicial

Una vez que la optimización inicial ha finalizado, los valores para los parámetros PID se muestran en la sección "Parámetros PID" de la ventana de configuración del objeto tecnológico (véase Figura 2.28).

Posteriormente se deberá cargar los valores calculados a la CPU, entonces el regulador empezará a funcionar en modo automático. El comportamiento en este punto no será el adecuado para el sistema ya que presenta cambios bruscos en la salida por lo cual se deberá realizar la optimización fina.

Parámetros PID

Activar entrada manual

Ganancia proporcional: 10.67107

Tiempo de integración: 2.317666 s

Tiempo derivativo: 4.055915E-1 s

Coefficiente retardo derivativo: 0.1

Ponderación de la acción P: 0.8

Ponderación de la acción D: 0.0

Tiempo muestreo algoritmo PID: 9.99991E-2 s

Regla para la optimización

Estructura del regulador: PID

Figura 2.28: Parámetros luego de la Optimización Inicial

### b. Optimización Fina

La optimización fina se basa en el segundo método de Ziegler-Nichols el cual se genera una oscilación constante y sostenida del valor real, los parámetros PID se optimizan en el punto de consigna del sistema a partir de la amplitud y frecuencia de las oscilaciones, de esta manera se recalcula todo los valores PID. La optimización fina presenta un mejor comportamiento del objeto tecnológico ante fallos de la señales (SIEMENS, S7-1200, S7-1500 Regulación PID, 2014).

En este caso para iniciar la sintonización fina se deberá ingresar al sistema una cantidad pequeña de agua con la finalidad que el algoritmo regule el valor de la salida cerca del punto de consigna (véase Figura 2.29).

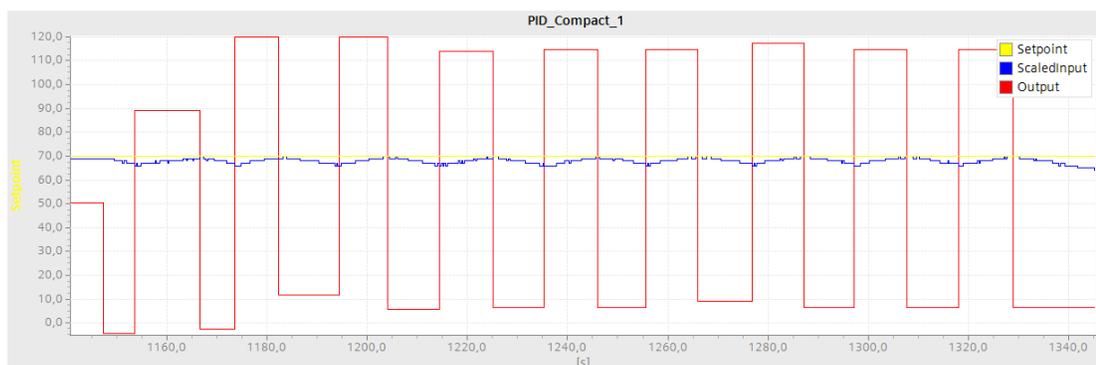


Figura 2.29: Optimización Fina

Los parámetros obtenidos mediante la optimización fina se observan en la Figura 2.30.

The screenshot shows a control panel titled 'Parámetros PID'. At the top left, there is a green status indicator and a checkbox labeled 'Activar entrada manual'. Below this, several parameters are listed, each with a numerical value, a unit, and a green status indicator with a double-headed arrow for adjustment:

- Ganancia proporcional: 1.949398
- Tiempo de integración: 5.42695 s
- Tiempo derivativo: 1.373762 s
- Coefficiente retardo derivativo: 0.1
- Ponderación de la acción P: 0.256659
- Ponderación de la acción D: 0.0
- Tiempo muestreo algoritmo PID: 9.99978E-2 s

At the bottom, under the heading 'Regla para la optimización', there is a dropdown menu for 'Estructura del regulador' set to 'PID'.

Figura 2.30: Parámetros luego de la Optimización Fina

El parámetro correspondiente al tiempo de derivativo en un sistema de primer orden no presentan mayores beneficios por lo cual deberá ser ajustado a un valor cercano a cero.

### 2.1.7 Pruebas de Funcionamiento

A continuación se expone las pruebas de funcionamiento realizadas sobre el sistema introduciendo un cierto punto de consigna e interactuando con la válvula de entrada de agua, el controlador PID deberá responder a regulaciones cuando su consigna sobrepase el valor real y viceversa.

El Objeto tecnológico de la CPU guarda los parámetros en su memoria y los utiliza para el cálculo de la salida cuando se encuentra en un modo automático, en este caso se introducirá un valor de consigna de 50% de llenado del tanque. En la Figura 2.31 se observa la pantalla HMI diseñada para el efecto en la cual se puede constatar el valor de referencia y el valor real del sistema.

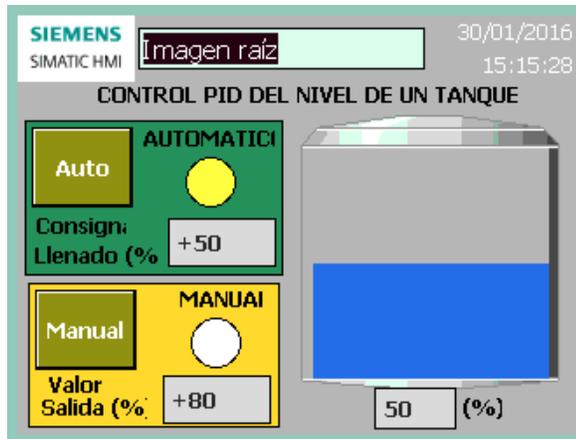


Figura 2.31: Pantalla SCADA del Sistema (Funcionamiento Automático)

El controlador PID mide el error constantemente y recalcula el porcentaje de salida PWM para mantener el valor real lo más cerca de la consigna independientemente del flujo de agua que ingrese al tanque, en la Figura 2.32 se puede observar la variación del valor de salida en el tiempo con el fin de mantener el nivel de tanque estable.

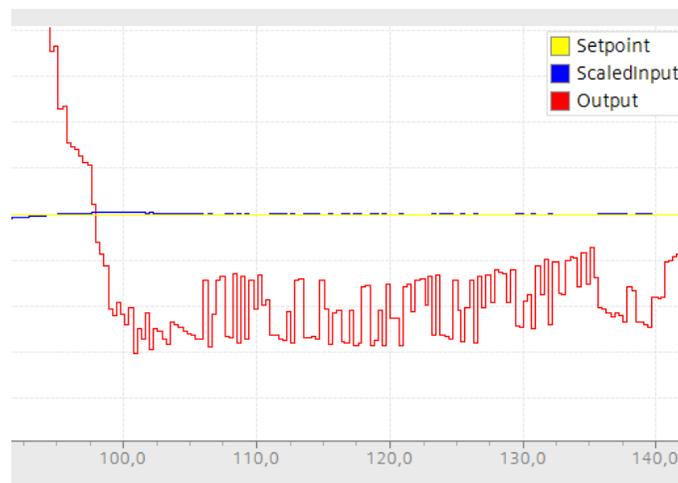


Figura 2.32: Señales del Sistema (Funcionamiento Automático)

Cuando el objeto tecnológico PID\_COMPACT se encuentra en modo manual no se cumple el cálculo del valor de salida, en su lugar la estructura recibe un dato de entrada “ManualValue” y la salida es conmutada con dicho porcentaje. En la Figura 2.33 se observa la pantalla del sistema en funcionamiento Manual.

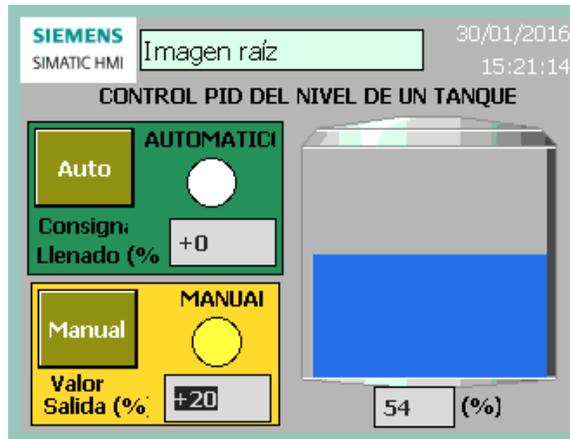


Figura 2.33: Pantalla SCADA del Sistema (Funcionamiento Manual)

En funcionamiento manual la salida se mantiene constante en el valor prefijado y por lo tanto podría generar un aumento o disminución del valor real, en la Figura 2.34 se observa las señales del sistema ante una salida de 20%.

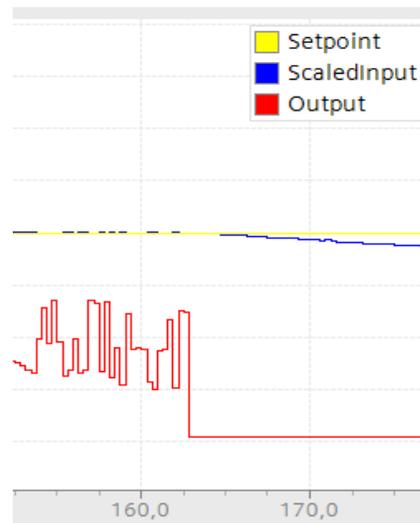


Figura 2.34: Señales del Sistema (Funcionamiento Manual)

## 2.2 SISTEMA DE AUTOMATIZACIÓN VIRTUAL USANDO *FACTORY I/O*

### 2.2.1 Descripción de *FACTORY I/O*

*FACTORY I/O* es un *software* desarrollado por *Real Games* para la construcción y simulación de sistemas de automatización especialmente de trasportación y manufactura, este entorno permite diseñar procesos con elementos industriales como bandas transportadoras, elevadores, clasificadores, paletas, placas giratorias, plataformas, guías, indicadores luminosos, sensores, botones de paro, etc. Este *software* recrea un ambiente inspirado en fábricas reales con una alta calidad de gráficos en 3D basados en los elementos más comunes que se pueden encontrar dentro de dichos procesos (véase Figura 2.35).

*FACTORY I/O* es compatible con varios de los sistemas de control más usado en la actualidad, la comunicación entre los controladores y la aplicación virtual se lo realiza mediante *drivers* específicos para cada elemento que relacionan los puntos I/O representados como etiquetas.



Figura 2.35: *FACTORY I/O*

Fuente: (Real-Games, *FACTORY I/O* User Guide, 2014)

Una etiqueta está definida por su nombre, dirección y el valor que puede ser discreto o analógico, cada dispositivo tiene varias etiquetas de entrada y salida asociadas para su control (Real-Games, *FACTORY I/O* User Guide, 2014).

Los *drivers* que están disponibles permiten la conexión de elementos como tarjetas Advantech USB 4704 & 4750, elementos Automgen TCP/IP, Modbus-TCP/IP cliente-servidor, Siemens LOGO TCP/IP, Siemens S7-300, 400, 1200, 1500 TCP/IP y Siemens PLCSIM (v5.4/5.5), además existe la posibilidad de un *driver* abierto OPEN (SDK) con el que se puede desarrollar aplicaciones de comunicación propias (Real-Games, *FACTORY I/O* User Guide, 2014).

### 2.2.2 Diseño y Construcción del Sistema

El sistema de automatización virtual que se desarrolla consiste en una fábrica para la selección de tres tipos de cajas de acuerdo a su tamaño y peso, el movimiento se lo realiza mediante tres bandas transportadoras que permiten dirigir las cajas que ingresan hacia una sección derecha o izquierda, las cajas grandes son dirigidas hacia la parte izquierda, a su vez las cajas medianas y pequeñas son encaminadas hacia la derecha donde se realiza su posterior control de peso. A continuación se realiza una breve descripción de los elementos utilizar.

- a. **Cajas:** Las cajas son elementos comunes dentro de las fábricas que sirven como contenedores de los productos desarrollados, en el *software* se puede distinguir tres tipos de cajas: grandes, medianas y pequeñas, las especificaciones básicas son las siguientes:

Tabla 2.6: Características Cajas en *Factory I/O*

	<b>Grande</b>	<b>Mediana</b>	<b>Pequeña</b>
<b>Alto (cm)</b>	36	24	24
<b>Largo (cm)</b>	36	60	60
<b>Ancho (cm)</b>	36	36	24
<b>Peso (Kg)</b>	15	10	8

Fuente: (Real-Games, Parts Essentials, 2014)

- b. Bandas Transportadoras:** Estos elementos se utilizan para llevar los productos de un lugar a otro durante la cadena de manufactura, existen varios tipos como los de cinta, de rodillo, con secciones divisorias, etc. dependiendo la aplicación. En *Factory I/O* se puede utilizar los transportadores de cinta o de rodillo en este caso se utiliza los primeros, la longitud de las bandas puede ser de 4 o 6 m y el control puede ser binario o analógico en ambas direcciones (Real-Games, Parts Essentials, 2014).
- c. Alineadores:** Estos elementos son estructuras generalmente metálicas que se sitúan en los costados de las cintas transportadoras o en los cambios de dirección entre dos bandas, su función es alinear los objetos transportados para evitar golpes o caídas de los mismos y permitir a los sensores una correcta medida de sus parámetros.
- d. Soportes:** Los soportes son estructuras metálicas ubicadas a los costados y sobre las cintas transportadoras con el fin de facilitar la colocación de sensores y actuadores cerca de los productos.
- e. Rampas o Tolvas:** Las tolvas son elementos metálicos por los cuales se puede encaminar los diferentes productos con el fin de separarlos según sus características, están ubicadas a los costados de las bandas transportadoras o en los extremos.
- f. Rueda Clasificadora:** Las ruedas clasificadoras son dispositivos que se ubican entre las cintas transportadoras para permitir direccionar el flujo hacia la derecha, izquierda o adelante, está formado por ruedas de giro continuo que permite dirigir los objetos circundantes con un ángulo de 45° a cada lado (Real-Games, Parts Essentials, 2014).
- g. Brazo de Pivote Clasificador:** Este elemento junto con los *pushers*<sup>41</sup> permite el descargo de las cajas hacia las tolvas adecuadas, el brazo pivote posibilita la

---

<sup>41</sup> *Pusher*: Máquina estática que cumple la función de empujar objetos.

interrupción y redirección de los objetos en un ángulo de 45°, además posee una cinta incorporada para impulsarlos en la dirección adecuada.

- h. Escalador Transportador:** Este dispositivo es usado para la medición de peso de los objetos que circulan sobre su banda transportadora, se sitúan entre cintas con el fin de obtener el peso de objetos en movimiento, en el *software* este elemento puede ser configurado como variable analógica de 0 a 10V con consignas de 20Kg o 100Kg (Real-Games, Parts Essentials, 2014).
- i. Sensor Opto electrónico Difuso:** Este sensor permite detectar todo tipo de objetos que circulan en su parte frontal, el rango de distancia en el que puede trabajar es elevado con un alcance de hasta 3 m, la onda emitida es reflejada por el objeto y detectada de vuelta por el receptor. En el *software* este sensor nos permite la detección de cajas y objetos metálicos hasta 0.8 m (Real-Games, Parts Essentials, 2014).
- j. Emisores y Receptores:** En *Factory I/O* existen elementos emisores que permiten introducir objetos de prueba al sistema de manera aleatoria (en este caso los tres tipos de cajas) y los receptores cuya función es la desechar los objetos una vez alcanzado los puntos adecuados definidos por el usuario (Real-Games, Parts Essentials, 2014).

La fábrica virtual construida con las especificaciones indicadas anteriormente es la que se muestra en las Figuras 2.36 y 2.37.



Figura 2.36: Fábrica Virtual Construida (Vista Frontal)

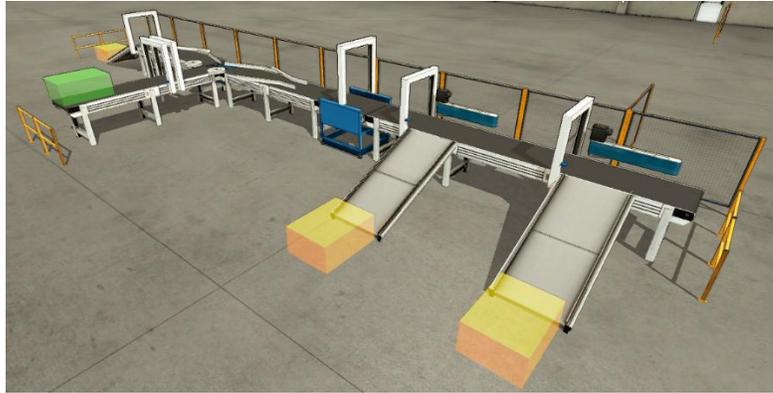


Figura 2.37: Fábrica Virtual Construida (Perspectiva)

## 2.2.3 Control del Sistema

### 2.2.3.1 Comunicación con Simatic S7-1200

*Factory I/O* es un *software* que maneja las comunicaciones con dispositivos externos mediante *drivers* sean estos para tarjetas USB, PLCs, etc. Para la conexión con dispositivos SIEMENS existen *drivers* específicos para cada CPU y se comunican mediante el protocolo TCP/IP, las configuraciones que se requiere para la conexión de una CPU física al entorno virtual son las siguientes:

#### a. Configuraciones en TIA PORTAL<sup>42</sup>

En el *software* TIA PORTAL se debe habilitar la comunicación entre el controlador y medios externos mediante TCP/IP, en versiones inferiores a la 4.0 se puede realizar de la siguiente manera:

1. En el árbol del proyecto nos situamos sobre la CPU 1200 agregada y entramos a Configuración del Dispositivo y luego a propiedades.
2. Nos situamos sobre la pestaña Protección y escogemos la opción Sin Protección. (Para CPU con versión 4.0 o superior adicionalmente se debe habilitar la Comunicación PUT/GET)

<sup>42</sup> TIA PORTAL: Paquete de *Software* para la configuración y programación de equipos como PLCs, HMIs, accionamientos de la marca SIEMENS.

3. En la pestaña Interfaz Profinet se debe asignar una dirección IP al PLC dentro de la red de trabajo que se encuentre.

#### b. Configuraciones en *FACTORY I/O*

En *FACTORY I/O* se debe primeramente configurar la comunicación con el PLC vía TCP/IP, como la conexión se realiza a través de la tarjeta de red de la computadora se deberá asignar una dirección IP adecuada dentro de la red de trabajo. En la Pestaña de *Configuración de Driver* se debe escoger el *driver* correspondiente a una CPU 1200, la dirección del Host deberá especificar la misma configurada para PLC al que se conecta el entorno virtual, el adaptador de red corresponde a la tarjeta de red o *Wireless* que deseemos, en la Figura 2.38 se resume las configuraciones mencionadas.



Figura 2.38: Parámetros de Conexión en *FACTORY I/O*

Como se explicó en la sección 2.2.1 cada elemento que se agrega en el área de trabajo posee etiquetas de entrada y salida que nos permiten interactuar con el mismo, mediante programación de controladores externos o mediante manipulación directa en la simulación del proyecto, además los elementos cuentan con una pestaña de configuración que nos permiten acceder a parámetros específicos del mismo.

Para la comunicación de los sensores y actuadores con las correspondientes áreas de memoria del PLC se deberá colocar las etiquetas para los sensores en la parte izquierda de la interfaz y para los actuadores en la parte derecha, en la Figura 2.39 se observa algunos controles creados que se han asignado para la conexión con su correspondiente variable del PLC.

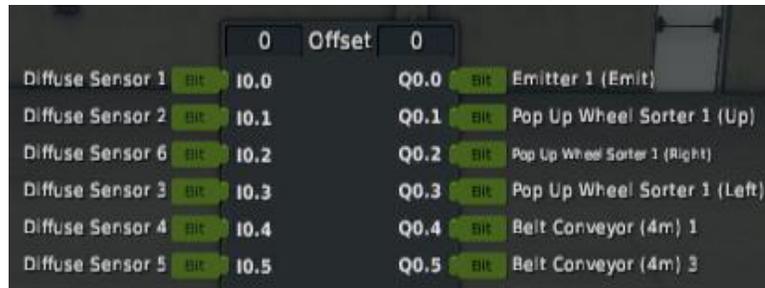


Figura 2.39: Asignación de Variables Digitales

Las variables analógicas del proyecto podrán ser asignadas a los espacios del controlador mediante las direcciones especificadas en la parte inferior de la Pestaña *Configuración de Driver*, se tiene un offset de direcciones en *Factory I/O* por lo cual se deberá comenzar en IW100 y QW100 respectivamente.

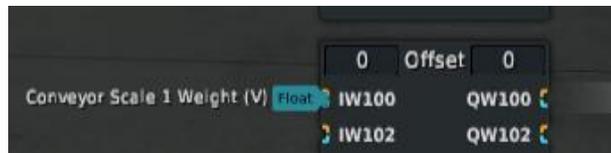


Figura 2.40: Asignación de Variables Analógicas

### 2.2.3.2 Estaciones

El control del proceso de producción se realiza a través del PLC Simatic S7-1200 conjuntamente con una interfaz HMI cuya configuración será abordada en el Capítulo 4. Para una mejor comprensión del proceso productivo se ha dividido el sistema en estaciones que facilitará los algoritmos de control mediante la programación de bloques de función en TIA PORTAL, la lógica para las estaciones se explica a continuación y los bloques de programación serán adjuntos en la sección de Anexos.

#### a. Estación 1: Selección de Cajas de acuerdo a su tamaño

La primera estación tiene como objetivo el control para la selección de piezas de acuerdo a su tamaño, está conformada por el elemento emisor de cajas, la banda central, dos sensores difusos, una rueda clasificadora, una banda derecha y una banda izquierda.

Las cajas son introducidas al sistema y al pasar por los sensores difusos convenientemente colocados se puede obtener una señal de control para determinar si su tamaño es largo o corto, en el caso de que sea corto se activará el giro a la izquierda de la rueda y si fuese largo se activará el giro a la derecha. Las bandas transportadoras a ambos lados se encenderán siempre y cuando existan piezas que trasladar.

#### **b. Estación 2: Control Cajas Grandes**

La segunda estación es la encargada de manejar el control para el contaje de piezas grandes además del mando y paro de la banda transportadora izquierda; está conformada por una banda de tipo cinta y un sensor difuso. Las cajas grandes que fueron encaminadas hacia la sección izquierda activan la banda transportadora cuyo contaje de piezas recibidas se realiza mediante el sensor difuso ubicado en la parte superior de la tolva de salida, las cajas que se encuentran en movimiento en el sistema deberán ser consideradas como elementos transitorios que no se contabilizan en el total real.

#### **c. Estación 3: Selección de Cajas de acuerdo a su peso**

La tercera etapa tiene como objetivo determinar el peso de las cajas largas que circulan por la sección derecha y realizar el contaje transitorio de las mismas para su posterior ubicación en las tolvas adecuadas, está formada por un escalador transportador y un sensor difuso. Las cajas medianas y pequeñas que se transportan por la sección derecha pasan por la estación 3 donde el sensor adecuado determina su peso correspondiente a 10Kg u 8Kg, el sensor difuso ubicado en el centro del escalador permitirá obtener una sola medida cuando la caja este completamente asentada en la banda.

#### **d. Estación 4: Contaje y control de empuje para las cajas pequeñas y medianas**

La última estación del sistema está encargada del control de empuje de las cajas pequeñas y medianas censadas previamente, además de su respectivo registro de contaje; está conformada por una banda transportadora, dos sensores difusos y dos brazos de pivote clasificadores.

En la estación 3 se obtuvo un contaje transitorio de piezas medianas y pequeñas, de esta manera conjuntamente con los sensores difusos se puede determinar a qué grupo pertenece cada objeto, los sensores están ubicados junto a las tolvas correspondientes y permiten accionar los brazos de pivote durante un tiempo adecuado para producir el descargo.

La programación del controlador se la realiza en Lenguaje KOP basándose en la lógica de contactos normalmente abiertos y cerrados, además de bobinas e instrucciones *set-reset* para la activación de las salidas digitales. Los tiempos de empuje de las cajas y de bloqueo de sensores se realizan mediante temporizadores asíncronos, el contaje de piezas tanto transitorias como totales esta manejado por contadores incrementales y decrementales colocados en el bloque de organización *Main* o en los bloques adicionales creados para los diferentes tipos de cajas.

#### 2.2.4 Pruebas de Funcionamiento

A continuación se expone las pruebas de funcionamiento realizadas sobre el sistema virtual con diferentes capturas de cada estación con el fin de comprobar los algoritmos dispuestos para el control, además de la comunicación mediante Profinet entre los registros del PLC y los de *Factory I/O*. En la primera parte correspondiente a la selección de piezas largas o cortas se evidencia el funcionamiento mediante la inserción al sistema de cajas de ambos tamaños, de esta manera la rueda clasificadora encamina los objetos hacia la izquierda o derecha (véase figura 2.41).

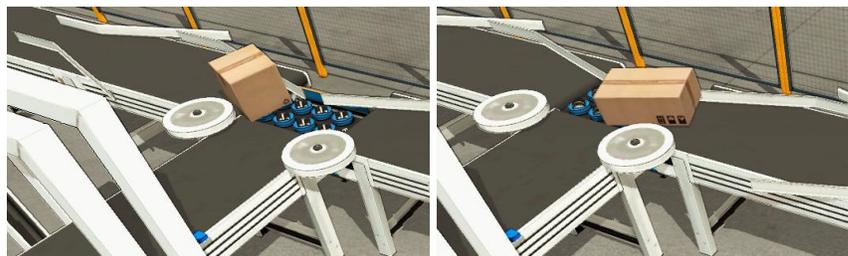


Figura 2.41: Encaminamiento de cajas Largas o Cortas

Una vez que las cajas largas han sido dirigidas hacia la derecha son transportadas por una banda de tipo cinta hasta un escalador transportador, la medición de peso se realiza una

vez que la caja se encuentra completamente asentada sobre la superficie del escalador, con esto se detiene el proceso durante un breve tiempo inferior a 1 segundo para proceder a tomar la medición exacta de cada caja (véase Figura 2.42).



Figura 2.42: Medición de Peso de las Cajas

Los datos de la variable analógica son transmitidos al PLC y escritos en el campo de entrada correspondiente (%IW100), este registro es escalado al valor en kilogramos y el peso obtenido en el programa para los dos tipos de cajas se puede evidenciar en la Figura 2.43.



Figura 2.43: Valores de Peso obtenidos en TIA PORTAL

Las cajas largas continúan su paso por la banda siguiente al escalador en donde se procederá a su descargo, las cajas pequeñas y medianas son empujadas por los dos brazos pivotes ubicados junto a las tolvas, el conteo de piezas transitorias y los sensores difusos permiten reconocer el objeto a ser empujado (véase Figura 2.44).

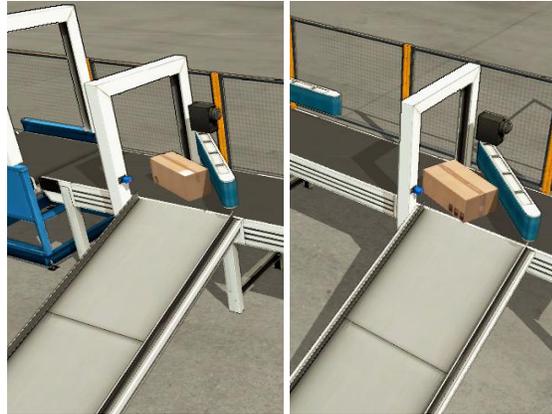


Figura 2.44: Descarga de Cajas Medianas y Pequeñas

Luego de comprobar el correcto funcionamiento de las estaciones del sistema se realiza una prueba general en la cual se insertan las cajas de manera aleatoria, en las Figuras 2.45 y 2.46 se puede observar el total de objetos que circularon por el sistema.



Figura 2.45: Total de Cajas Medianas y Pequeñas



Figura 2.46: Total de Cajas Cortas

Mediante la pantalla principal del Sistema SCADA se puede controlar la activación de la banda emisora de cajas lo cual enciende el proceso, si todas las cajas han sido descargadas las bandas del sistemas se detienen automáticamente. En la pantalla se puede observar las

cintas transportadoras que se encuentran accionadas mediante un resalte de color verde en su borde, además se puede visualizar el total de cajas empujadas en cada tolva (véase Figura 2.47). En este caso el sistema ha recibido 6 cajas de tamaño corto, 3 cajas de tamaño pequeño-largo y 5 cajas medianas-largas.

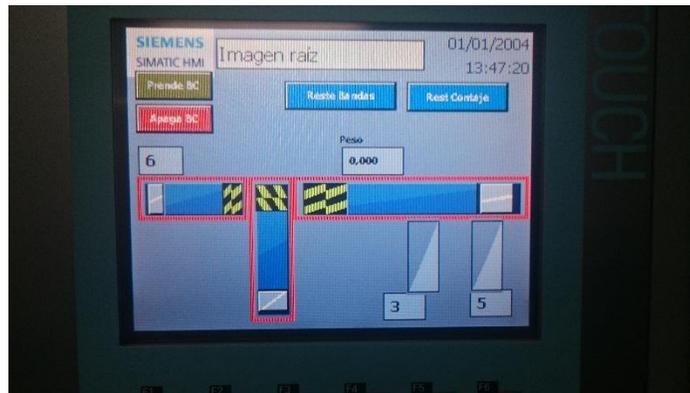


Figura 2.47: Pantalla Principal del SCADA

### 2.3 CONCLUSIONES

Luego de la culminación del presente capítulo se logró el establecimiento de dos sistemas de automatización, el primero enfocado a la aplicación del algoritmo de control PID y el segundo dirigido a la simulación de una Fabrica Virtual manejada desde el PLC. De acuerdo a los conocimientos adquiridos se puede destacar lo siguiente.

- El control PID es el algoritmo más utilizado en los procesos industriales dando solución a la mayoría de aplicaciones cuya dinámica puede ser descrita como una de primer o segundo orden, basa su funcionamiento en un lazo de realimentación en el cual se mide constantemente el error entre la salida real del sistema y una consigna prefijada.
- El cálculo de la salida PID se basa en tres acciones: la proporcional, la integradora y la derivativa; de esta manera se determina una respuesta adecuada según la dinámica del proceso y los parámetros PID son: la ganancia proporcional, el tiempo de integración y el tiempo derivativo.

- En la actualidad se han desarrollado algunos métodos que permiten determinar los valores de los parámetros según el desempeño del proceso, estas técnicas son conocidas como sintonización de los controladores PID.
- Los métodos de Ziegler-Nichols permiten ajustar los parámetros PID mediante dos pasos: el primero al establecer la respuesta del sistema a un escalón de la salida; el segundo al introducir al sistema oscilaciones continua y limitadas con el fin de determinar la ganancia crítica.
- El PLC Simatic S7-1200 tiene incorporado un objeto tecnológico conocido como PID\_COMPACT, esta estructura provee un algoritmo PID Universal con optimización integrada.
- Los parámetros PID obtenidos mediante la simulación en MATLAB difieren de los parámetros calculados mediante la sintonización práctica, esto se debe a que los valores obtenidos por el PLC fueron calculados a partir del comportamiento real del sistema, es decir con los tiempos de respuesta de la bomba vs el nivel del líquido.
- La ganancia proporcional y el tiempo de integración en este sistema estarán relacionados con el sobre pico y tiempo de estabilización, mientras menor sea el tiempo de integración respecto a la ganancia el sistema tardará en estabilizarse aunque el sobre pico será mínimo.
- Por otro lado si se aumenta demasiado el tiempo de integración el sistema tendrá más oscilaciones.
- La acción derivativa busca mejorar la estabilidad del sistema prediciendo el error, pero un aumento elevado en este tipo de sistema desencadenaría en una serie de oscilaciones.
- *FACTORY I/O* es un *software* para la simulación de procesos de transportación y manufactura con la posibilidad de conexión a varios de los sistemas de control más utilizados.

- La comunicación con el *software* se maneja mediante *drivers* específicos para cada controlador con interfaces como USB o TCP/IP. Los elementos de la simulación poseen etiquetas de entrada salida que pueden ser enlazadas con las áreas de memoria del PLC tanto para valores digitales como analógicos

## CAPÍTULO 3

### IMPLEMENTACIÓN DE LAS COMUNICACIONES INDUSTRIALES MEDIANTE SIMATIC S7-1200

#### **Introducción**

En el presente capítulo se expone un estudio acerca de la implementación de las principales redes de comunicación como son: Profinet, Profibus, Modbus-RTU y Modbus-TCP. Las redes mencionadas son establecidas para la comunicación entre equipos Simatic S7-1200 mediante el uso de protocolos e interfaces aplicadas hoy en día en el campo industrial. Para cada implementación se realiza un análisis de los elementos necesarios en el sistema incluyendo la interfaz, cables y conectores; también se destaca las configuraciones de red e instrucciones adecuadas para la transmisión y recepción de datos en cada caso. Luego se realiza un ejemplo de conexión en el que se presenta algunas de las bondades más relevantes de cada protocolo con sus correspondientes pruebas de funcionamiento.

#### **3.1 RED PROFINET**

Una red de comunicación Profinet puede ser establecida entre dos equipos Simatic para transferir datos entre controladores mediante el uso de *Ethernet* y los protocolos TCP o ISO sobre TCP, la red permite crear conexiones o canales independientes para la transmisión de datos en ambas direcciones con el uso de instrucciones de envío y recepción. Las características del protocolo Profinet se encuentran explicadas en la sección 1.3.

### 3.1.1 Elementos del Sistema

#### 3.1.1.1 Interfaz Profinet

Todos los controladores Simatic S7-1200 poseen una interfaz Profinet integrada en la CPU, por medio de este puerto LAN se puede establecer las comunicaciones con otros elementos Profinet-I/O como: controladores I/O, dispositivos I/O y supervisores I/O. Este puerto puede manejar comunicaciones TCP, UDP e ISO sobre TCP a velocidades de 100 Mbits/s, por lo cual estos controladores puede trasferir datos mediante estos protocolos estándar a cualquier dispositivo con puerto LAN que se conecte en la red.

#### 3.1.1.2 Cables y Conectores

El cable de conexión que se utiliza para comunicar dos equipos mediante una red Profinet es el cable de red (*pacth cord*) conocido en redes informáticas, en el entorno industrial se utiliza los cables de dos o cuatro pares con apantallamiento de aluminio más trenza de cobre, son conductores de categoría 5, 5e o 5 PLUS que permiten velocidades de transmisión adecuadas para esta red. La interfaz física de conexión por la cual se empalma el cable a los equipos de la red es RJ-45 usada en redes de computadoras o cableado estructurado, este conector posee ocho pines y ha sido estandarizado para su acoplamiento en modo directo o cruzado.

Las especificaciones Profinet detalladas en "*Profinet Cabling and Interconnection Technology*" recomiendan el uso de cableado según la norma ISO/IEC 11801 e ISO/IEC 24702 para la interconexión hasta las islas de automatización y entre ellas (véase Figura 3.1).



Figura 3.1: Cable y Conector para red Profinet

Fuente: (SIEMENS, IE TP Cords - Industrial Communication Siemens, S.A.)

### 3.1.2 Configuración Red Profinet

El puerto Profinet de las CPUs soporta diferentes estándares de comunicación a través de la red *Ethernet*, los cuales son utilizados para interconectar CPUs con CPUs o CPUs con dispositivos de campo Profinet-I/O, estos estándares son:

- TCP
- Iso sobre TCP
- UDP
- Comunicación S7
- Profinet-I/O

La comunicación que se utilizará en este ejemplo corresponde a la TCP o Iso sobre TCP que permiten conectar dos CPUs usando el transporte de tramas, fragmentación y ensamblado de mensajes. Este protocolo de comunicación es muy eficiente y adecuado para volúmenes de datos medianos y grandes además de las propias prestaciones de la trama como control de errores, de flujo, etc.

#### a. Red

La red Profinet que se constituye está formada por dos CPUs Simatic S7-1200 conectadas mediante un cable *Industrial Ethernet*, para establecer las conexiones lógicas entre los equipos se deberá agregar ambas CPUs al proyecto y en la Pestaña Dispositivos y Redes se deberá arrastrar la conexión de un puerto a otro (véase Figura 3.2), las direcciones IP deben ser asignadas dentro de la red de trabajo de manera unívoca. La topología que se sigue en una red básica como esta es de tipo estrella, en el caso que se agreguen más equipos podrán ser enganchados a la conexión PN/IE creada.

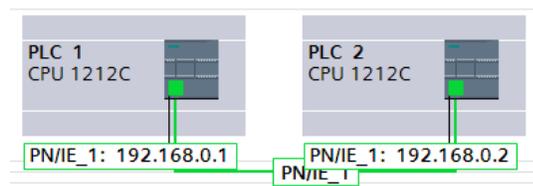


Figura 3.2: Red Profinet entre dos Simatic S7-1200

### **b. Configuración Interfaz Profinet**

La dirección IP de una CPU es un aspecto sumamente importante para su programación y comunicación en redes de datos, la configuración de los parámetros necesarios se realiza en la interfaz Profinet de cada dispositivo que puede ser accedida en las Propiedades del PLC (véase Figura 3.3), los aspectos más importantes a considerar son:

**Subred:** Dentro de la gestión del proyecto se pueden crear diferentes redes de conexión de varios tipos, es así que también puede existir más de una subred PN/IE, en este caso se deberá escoger la indicada a la cual se conectan los equipos.

**Dirección IP:** Cada dispositivo de la red deberá poseer una dirección IP unívoca que le permita comunicarse dentro de la misma, esta dirección es una combinación de cuatro octetos que se expresa en forma decimal.

**Máscara de Subred:** Las máscaras de subred son combinaciones de números que delimitan las direcciones dentro de una red IP, cada subred puede contener tantos dispositivos como la delimitación lo permita, la máscara de subred sin creación de nuevas redes es 255.255.255.0.

**Router IP:** Si se cuenta con diferentes redes LAN que se enlazan a través de un *Router* IP se deberá especificar en el proyecto la dirección del mismo, de manera que los paquetes puedan ser enrutados hacia las redes externas.

Figura 3.3: Configuración Interfaz Profinet

### c. ID de Conexión

Las comunicaciones Profinet mediante TCP o ISO sobre TCP se manejan a través de canales de conexión, cuando se insertan instrucciones como TSEND\_C, TRCV\_C o TCON se crean bloques de datos de instancia asociados a estos controles los cuales contienen los parámetros del canal de comunicación que se ha creado.

Las ID de conexión son identificadores únicamente relevantes para las instrucciones Profinet mediante las cuales se gestiona los diferentes canales, cada ID debe ser unívoca y estar incluida en un DB distinto. En dos diferentes CPUs se puede establecer las mismas IDs de conexión aunque estos números no deben coincidir necesariamente, ya que cada CPU maneja sus identificadores de forma individual, no es necesaria conocer el ID en el otro extremo para direccionar el flujo de datos.

Nuevas conexiones de datos pueden ser creadas mediante las estructuras mencionadas, por lo cual se puede crear diferentes enlaces de información entre dos CPUs o con equipos diferentes. En el ejemplo que se expone a continuación se muestra la comunicación entre dos CPUs mediante dos canales de conexión diferentes, usados para recibir y transmitir datos mediante instrucciones TSEND\_C y TRCV\_C (véase Figura 3.4).

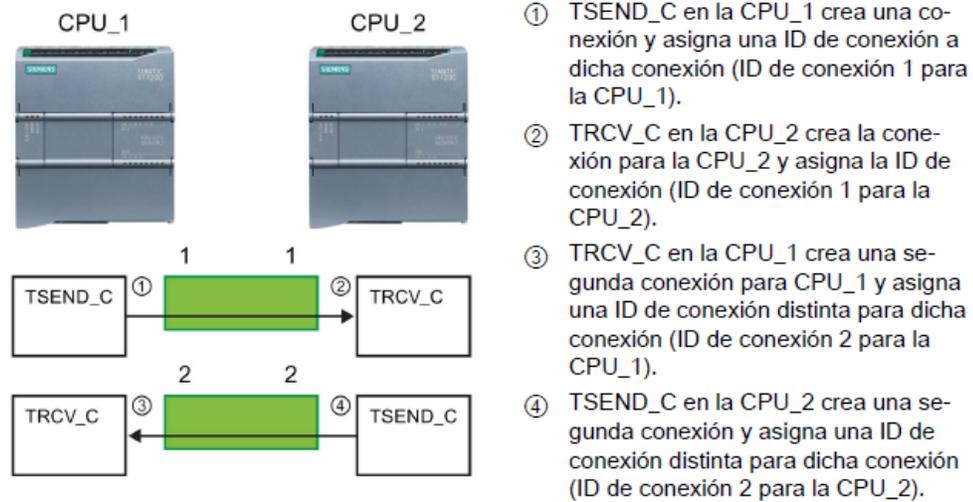


Figura 3.4: Comunicación Profinet mediante dos canales de conexión

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

En el ejemplo siguiente se muestra la comunicación entre dos CPUs mediante un solo canal de conexión, el cual es usado para recibir y transmitir datos mediante instrucciones TSEND y TRCV. Es necesaria la instrucción TCON para crear el canal y configurar los parámetros del mismo (véase Figura 3.5).

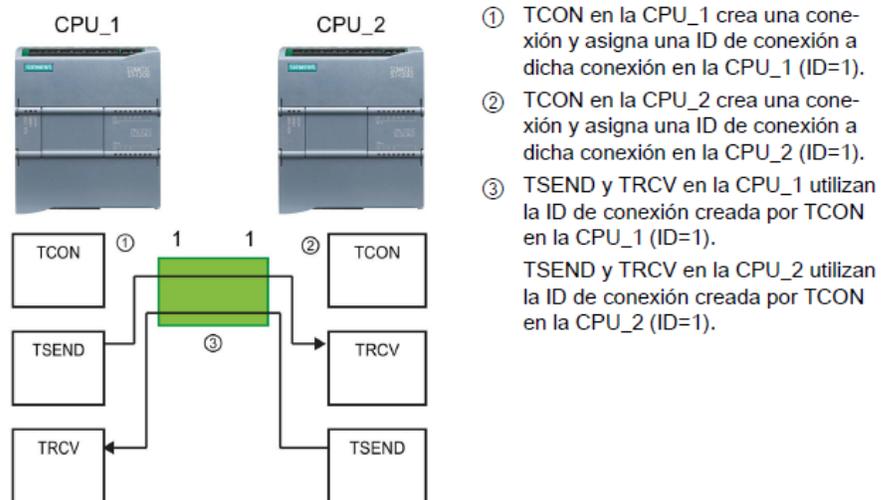


Figura 3.5: Comunicación Profinet mediante un canal de conexión

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

#### **d. Instrucciones *Open User Communication***

Dentro del paquete de Instrucciones *Open User Communication* situadas en la parte derecha de TIA PORTAL se puede encontrar las instrucciones necesarias para la comunicación mediante los protocolos TCP y UDP, estos controles permite configurar, enviar y recibir datos entre dos equipos conectados. A continuación se describe las funciones de las estructuras principales que deberán ser añadidas para una comunicación de datos vía *Ethernet*.

Instrucción TSEND\_C: Esta instrucción permite configurar y establecer una conexión o canal para la comunicación mediante protocolos TCP o ISO con un interlocutor remoto, combina las funciones de las instrucciones TCON, TDISCON y TSEND por lo cual puede establecer la comunicación, interrumpirla y enviar datos; la estructura se ejecuta de forma asíncrona de esta manera la CPU mantiene y vigila el enlace automáticamente luego de ser constituido.

Instrucción TRCV\_C: Esta instrucción permite configurar y establecer una conexión mediante protocolos TCP o ISO con un interlocutor remoto, combina las funciones de las instrucciones TCON, TDISCON y TRCV por lo cual puede establecer la comunicación, deshacerla y recibir datos, la estructura se ejecuta de forma asíncrona. Se puede recibir datos en modo Ad hoc<sup>43</sup>, con longitud específica u orientada a la recepción de mensajes.

Instrucción TCON: Esta instrucción permite configurar y establecer una nueva conexión según los parámetros indicados en CONNECT e ID, de igual manera una vez establecida la conexión la CPU mantiene y vigila la misma. Es preciso solo una llamada a esta instrucción ya que en el caso de una interrupción, el interlocutor se encarga del restablecimiento automático del canal.

---

<sup>43</sup> Ad hoc: Configuración de comunicación, en la que todos los equipos tienen las mismas capacidades y cualquiera puede iniciar un intercambio de información.

Instrucción TSEND: Esta instrucción permite enviar datos a través de una conexión existente, la longitud y datos se especifica en el parámetro DATA.

Instrucción TRCV: Esta instrucción permite recibir datos a través de una conexión existente, los datos captados se escriben en un área de recepción cuya longitud es especificada en función del protocolo usado.

Instrucción TDISCON: Esta instrucción permite eliminar una conexión creada o cerrar el canal entre la CPU y el interlocutor remoto, es una estructura de ejecución asíncrona.

### 3.1.3 Ejemplo de Conexión

En este apartado se presenta un ejemplo de conexión de dos CPUs Simatic S7-1200 mediante el uso del protocolo ISO sobre TCP, para esto se hará uso de las instrucciones TSEND\_C y TRCV\_C que permitirán el intercambio de datos de tipo Real y Sint.

#### a. Configuración de *Hardware* para los PLCs

Para establecer la red Profinet se debe agregar los dos PLCs a un proyecto nuevo y configurar los parámetros de la interfaz Profinet de la siguiente manera:

- PLC1 (S7 1212C AC/DC/RLY)
  - Asignar Dirección IP: 192.168.0.1
  - Mascara de Subred: 255.255.255.0
  - Activar Marca de Ciclo (en este caso sirve como variable de control para enviar los datos cada ciclo).
  
- PLC2 (S7 1212C AC/DC/RLY)
  - Asignar Dirección IP: 192.168.0.2
  - Mascara de Subred: 255.255.255.0

- Activar Marca de Ciclo (en este caso sirve como variable de control para recibir los datos cada ciclo).

### b. Programación PLC1

Se realiza el programa de control que deseemos para enviar las variables, en este caso solo se tiene dos, una de tipo Sint (1 o 0) en función de la entrada IO.0 y una variable de tipo real que será transmitida al accionar la entrada mencionada. Se debe crear un bloque de datos que contenga los valores indicados. (Este bloque deberá estar en ambos PLCs para poder almacenar los datos a enviar y recibir).

Bloque de datos PLC1			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	Datos1	Struct	
3	Valor Real	Real	0,0
4	Valor Sint	SInt	0

Figura 3.6: Bloque de Datos PLC1

Bloque de datos PLC2			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	Datos2	Struct	
3	ValorRealR	Real	0,0
4	ValorSintR	SInt	0

Figura 3.7: Bloque de Datos PLC2

Luego se agrega la estructura TSEND\_C para gestionar y enviar los datos vía *Ethernet*, al dar doble clic sobre la instrucción se puede acceder a la pestaña Configuración y Parámetros de Conexión. En las configuraciones de parámetros de la conexión se realiza lo siguiente:

- Se escoge el interlocutor, sería PLC2.
- En la parte de Datos de Conexión (PLC1), se elige crear un nuevo DB.
- El DB del interlocutor no se asigna todavía ya que no ha sido creado en el PLC2.
- Luego se elige el Tipo de Conexión: ISO sobre TCP.
- Seleccionamos la casilla Establecimiento Activo de la red.

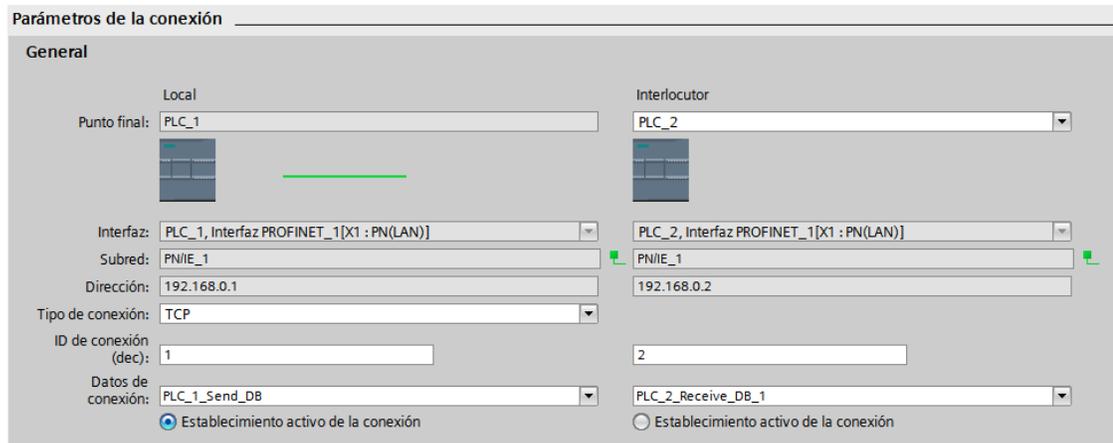


Figura 3.8: Parámetros de Conexión PLC1

En el bloque MAIN o través de la pestaña Parámetros de Bloque colocamos los valores correspondientes para el control de la siguiente manera:

- REQ: Inicia la petición, con un flanco ascendente. Para conseguir un flanco automáticamente se coloca la marca del sistema correspondiente a 10Hz.
- CONT: El valor “0” es usado para deshacer la conexión y el “1” para mantenerla.
- CONNECT: Se escoge el DB que se generó al insertar la instrucción el mismo que contiene los parámetros del canal.
- DATA: Corresponde a los datos que se desea enviar, en este caso apuntará al bloque de datos creado.

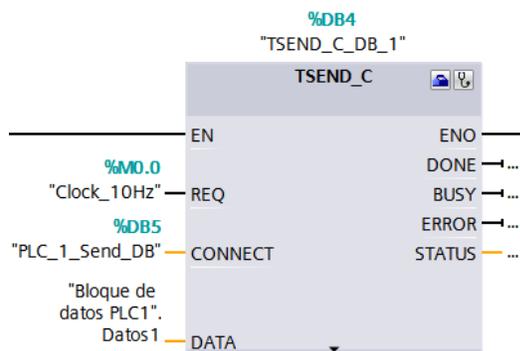


Figura 3.9: Instrucción TSEND\_C

### c. Programación PLC2

Posteriormente se agrega la estructura TRCV\_C para gestionar y recibir los datos vía *Ethernet*, al dar doble clic sobre la misma se accede a la pestaña Configuración y Parámetros de Conexión. En las configuraciones de parámetros de la conexión realizamos lo siguiente:

- Se escoge el interlocutor, sería PLC1.
- En la parte de Datos de Conexión (PLC2), se elige crear nuevo DB.
- El DB del interlocutor corresponde al bloque de datos que se agregó para el PLC1.
- Luego se elige el Tipo de Conexión: ISO sobre TCP.
- Finalmente se activa la casilla Establecimiento Activo de la red para el PLC1.

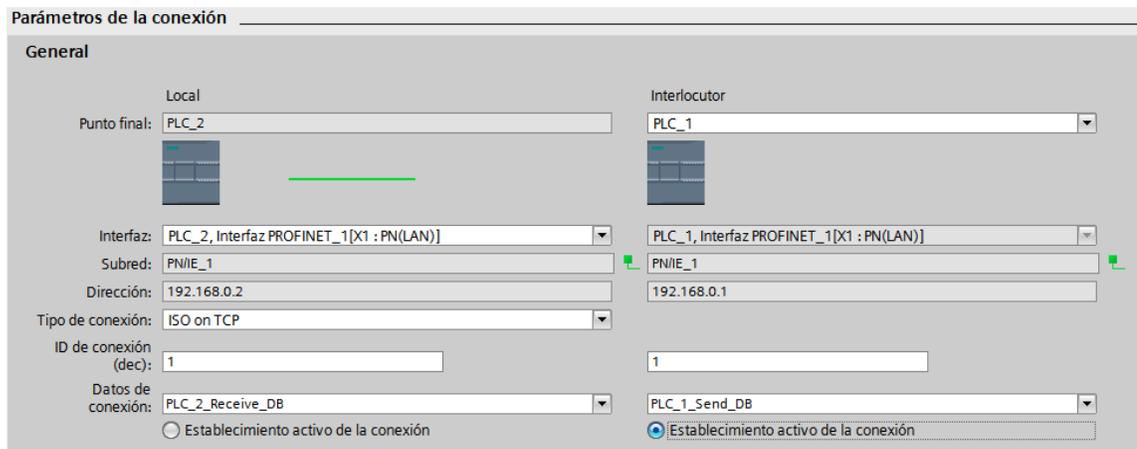


Figura 3.10: Parámetros de Conexión PLC2

En el bloque MAIN o través de la pestaña Parámetros de Bloque colocamos los valores correspondientes para el control de la siguiente manera:

- REQ: Inicia la petición, con un flanco ascendente. Para conseguir un flanco automáticamente se coloca la marca del sistema correspondiente a 10Hz.
- CONT: El valor “0” es usado para deshacer la conexión y el “1” para mantenerla.
- CONNECT: Se escoge el DB que se generó al insertar la instrucción el mismo que contiene los parámetros del canal.

- DATA: Corresponde a los datos que se desea recibir, en este caso apuntará al bloque de datos creado.

Luego de esto, en la configuración de la instrucción TSEND\_C se debe asignar el bloque de datos del interlocutor que quedó pendiente en la sección de Programación del PCL1.

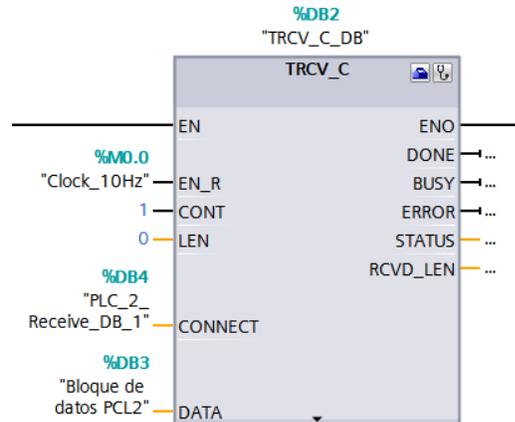


Figura 3.11: Instrucción TRCV\_C

### 3.1.4 Pruebas de Funcionamiento

Como se mencionó en la sección anterior, esta aplicación se realiza para la conexión de dos CPUs Simatic S7-1200 mediante una red Profinet, el programa que se realiza permite el envío y recepción de datos entre los dos equipos, la información que se desee transmitir puede ser fácilmente agregada al área de memoria de la estructura creada para ambos bloques de datos, de esta manera se puede enviar un grupo variables de diferentes tipos.

Los datos que se transmiten desde el PLC1 corresponden a dos variables que son enviadas según el estado de la entrada I0.0, si la entrada se encuentra desactiva se transmite el valor de 10.5 para la variable Real y el valor de 0 para la variable Sint.

Las pruebas que se realizan permiten evidenciar el cambio de datos mediante una conexión *online* en la primera CPU (véase Figura 3.12).

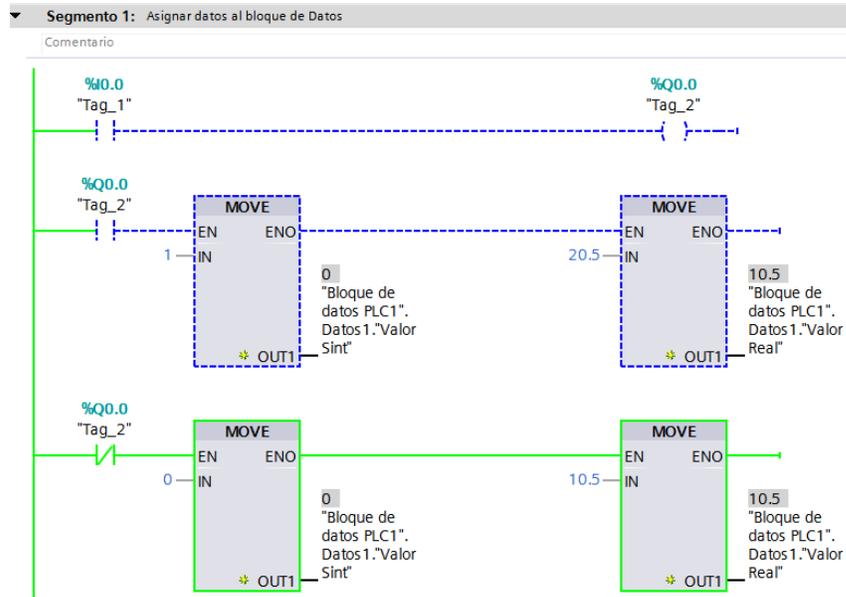


Figura 3.12: Envío de Datos desde PLC1 (1)

Para comprobar que los datos fueron recibidos por la CPU número dos, se agrega una tabla de observación que nos permite una visualización *online* de los valores correspondientes a las áreas de memoria de interes, la estructura de recepción actualiza sus valores cada 10Hz debido a la marca de ciclo seleccionada, si se desea se puede escoger un momento determinado para la recepción según la logica de programación. En la Figura 3.13 se puede constatar que los valores recibidos son correctos.

RedProfinet ▶ PLC\_2 [CPU 1212C AC/DC/Rly] ▶ Tablas de observación y forzado permanente ▶ Tabla de observación\_1

	Nombre	Dirección	Formato visualización	Valor de observac..	Valor de forzado
1	"Bloque de datos PCL2".PLC2.ValorRealR		Número en coma flot...	10.5	<input type="checkbox"/>
2	"Bloque de datos PCL2".PLC2.ValorSintR		DEC+/-	0	<input type="checkbox"/>
3		<Agregar>			<input type="checkbox"/>

Figura 3.13: Recepción de Datos en PLC2 (1)

Por otra parte si la entrada I0.0 del PLC1 se encuentra activada se transmite el valor de 20.5 para la variable Real y el valor de 1 para la variable Sint, en la segunda prueba de funcionamiento se puede observar los datos enviados a través de la Figura 3.14

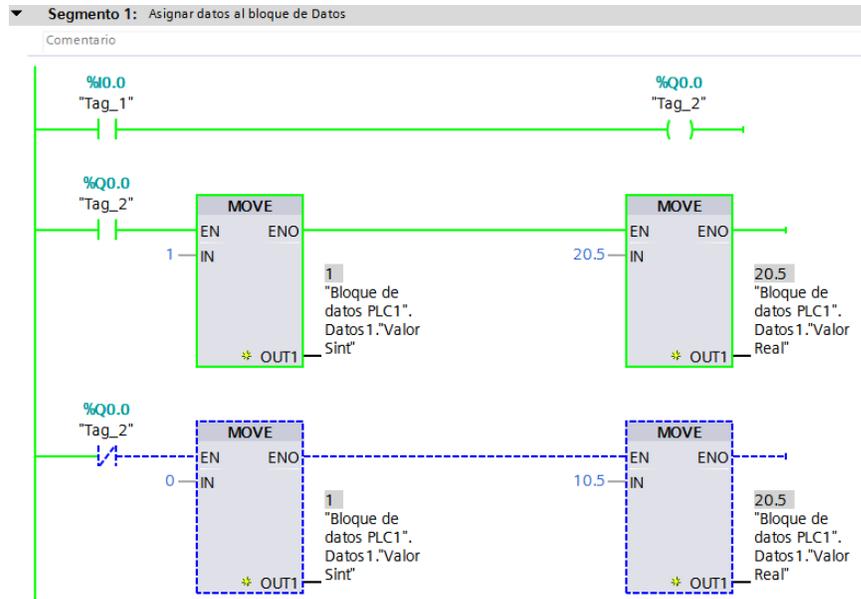


Figura 3.14: Envío de Datos desde PLC1 (2)

Finalmente mediante la tabla de observación agregada se puede constatar el cambio en valores de los datos recibidos correspondientes a 20.5 y 1 (véase Figura 3.15), de esta manera se comprueba el funcionamiento adecuado de la aplicación. Los datos recibidos pueden ser procesados en la CPU2 mediante la lógica deseada, si se requiere enviar datos de regreso se debe configurar un nuevo canal o usar el canal existe como se explicó en la sección 3.1.2.

RedProfinet ▶ PLC\_2 [CPU 1212C AC/DC/Rly] ▶ Tablas de observación y forzado permanente ▶ Tabla de observación\_1

	Nombre	Dirección	Formato visualización	Valor de observac..	Valor de forzado	
1	"Bloque de datos PCL2".PLC2.ValorRealR		Número en coma flot...	20.5		<input type="checkbox"/>
2	"Bloque de datos PCL2".PLC2.ValorSintR		DEC+/-	1		<input type="checkbox"/>
3		<Agregar>				<input type="checkbox"/>

Figura 3.15: Recepción de Datos en PLC2 (2)

## 3.2 RED PROFIBUS

Una red de comunicación Profibus puede ser establecida entre dos equipos Simatic para transferir datos de proceso con el modelo maestro-esclavo, el controlador principal de la red o Maestro Profibus está en la capacidad de sondear los dispositivos esclavos conectados al bus RS-485 e iniciar acciones de comunicación. Por su parte los equipos subordinados solo pueden acusar mensajes recibidos o enviar respuestas al maestro. Las características del protocolo Profibus se encuentran explicadas en la sección 1.2.

### 3.2.1 Elementos del Sistema

#### 3.2.1.1 Módulo Maestro-DP (CM 1243-5)

El módulo de comunicaciones CM 1243-5 (véase Figura 3.16) brinda a una CPU S7-1200 la capacidad de conexión a una red Profibus en calidad de Maestro-DP. Las comunicaciones están basadas en Profibus-DP-V1 por lo cual se puede establecer intercambio de datos con Esclavos-DP-V0/V1 como: periferia descentralizada Simatic ET200, CPUs S7-1200 con módulos de Esclavo-DP (CM 1242-5), CPUs S7-1200 con módulos EM 277, convertidores de frecuencia SINAMICS y MICROMASTER, CPUs S7-300/400, estaciones PC Simatic con CP Profibus, accionamientos o equipos de varios fabricantes, además de sensores que posean esta interfaz (SIEMENS, S7-1200 Profibus CM 1243-5, 2012).



Figura 3.16: Módulo CM 1243-5 Maestro Profibus-DP

Fuente: (SIEMENS, S7-1200 Profibus CM 1243-5, 2012)

Este módulo soporta dos tipos de comunicaciones tanto la cíclica como la acíclica, la primera permite la transferencia de datos de proceso entre el esclavo y maestro DP controlada por la propia CPU, por lo cual datos I/O se leen o escriben directamente en la imagen de proceso del controlador, la segunda permite recibir alarmas de los esclavos, acciones de configuración y diagnóstico (SIEMENS, S7-1200 Profibus CM 1243-5, 2012).

Los servicios de comunicación adicionales que se puede prestar son de tipo PUT/GET en el cual el maestro DP funciona como cliente y servidor para gestionar las consultas de otros controladores o PCs a través de Profibus, también se incluye el servicio PG/OP que permite intercambiar datos de configuración y programas desde dispositivos con PG/OP como pueden ser HMI o PCs Simatic (SIEMENS, S7-1200 Profibus CM 1243-5, 2012).

La velocidad de transferencia de datos que se maneja en Profibus es de entre 9.6 Kbits/s hasta 12 Mbits/s y soporta un número máximo de 127 esclavos en conexión directa con el maestro DP.

### **3.2.1.2 Módulo Esclavo-DP (CM 1242-5)**

El módulo de comunicaciones CM 1242-5 (véase Figura 3.17) brinda a una CPU S7-1200 las capacidad de conexión a una red Profibus en calidad de Esclavo-DP, las comunicaciones están basadas en Profibus-DP-V1 por lo cual se puede establecer intercambio de datos con Maestros-DP-V0/V1 como: Simatic S7-1200, S7-400, S7-300, controladores modulares embebidos, periferia descentralizada Simatic ET200 en calidad de maestro, estaciones PC Simatic, Simatic Net IE/PB Link PN IO para conexión con otras redes y accionamientos o equipos de varios fabricantes (SIEMENS, S7-1200 Profibus CM 1242-5, 2014).



Figura 3.17: Módulo CM 1242-5 Esclavo Profibus-DP

Fuente: (SIEMENS, S7-1200 Profibus CM 1242-5, 2014)

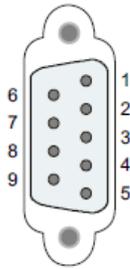
Este módulo de igual manera que el maestro DP soporta las comunicaciones cíclicas y acíclicas con las mismas velocidades de transferencia especificadas en el apartado anterior, además el área máxima de entrada y salida de datos DP es de 240 bytes, la configuración del módulo se lo realiza mediante el *software* TIA PORTAL a partir de la versión V11.0 (SIEMENS, S7-1200 Profibus CM 1242-5, 2014).

### 3.2.1.3 Cable y Conectores

Los cables utilizados en las redes Profibus son conductores de dos hilos apantallados mediante una lámina de aluminio la cual sirve como conexión a tierra o referencia de las señales, los dos hilos del cable son de color rojo y verde respectivamente, los cuales se empalman a los pines A y B de los conectores de bus.

El conector que se utiliza en la red es de tipo DB9 para interfaces RS-485, los cables poseen dos terminales hembras a cada extremo preferiblemente de carcasa metálica para minimizar las perturbaciones, los pines que se utilizan en esta red son el tres (RxD/TxD-P, conductor de datos B) y el ocho (RxD/TxD-N, conductor de datos A), además de la carcasa asignada para el aterramiento, en la siguiente tabla se detalla la función de los pines del conector.

Tabla 3.1: Asignación de los pines del conector hembra DB9 en redes RS-485



Pin	Descripción	Pin	Descripción
1	- no asignado -	6	VP: alimentación de tensión +5 V sólo para resistencias de cierre de bus (terminación); no para la alimentación de aparatos externos
2	- no asignado -	7	- no asignado -
3	RxD/TxD-P: Conductor de datos B	8	RxD/TxD-N: Conductor de datos A
4	CNTR-P: RTS	9	- no asignado -
5	DGND: masa para señales de datos y VP	Carcasa	Conexión de tierra

Fuente: (SIEMENS, S7-1200 Profibus CM 1243-5, 2012)

Siemens ofrece conectores de bus *FastConnect* para redes RS-485 que permiten enlazar de manera rápida varios elementos del bus gracias a que posee dos juegos de terminales para cables de entrada y salida, además implementa interruptores para polarizar o terminar la red, una función de seccionamiento y un terminal hembra adicional para PG, la salida de los cables puede ser a 35°, 90° o 180°; es apto para atmósferas potencialmente explosivas de la zona 2 (véase Figura 3.18) (SIEMENS, S7 Controlador programable S7-1200, 2015).

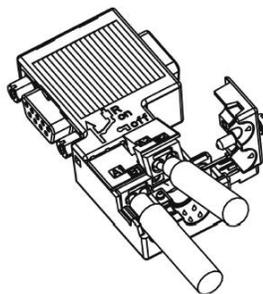


Figura 3.18: Conector *FastConnect* Profibus

Fuente: (SIEMENS, S7-1200 Profibus CM 1243-5, 2012)

Como se mencionó en el apartado 1.2.1.3 las redes Profibus poseen resistencias terminadores que sirven para evitar las reflexiones en el bus, estos elementos deben ser

conectados y polarizados en los dispositivos extremos, a su vez los elementos intermedios no se polarizan ni terminan (véase Figura 3.19)

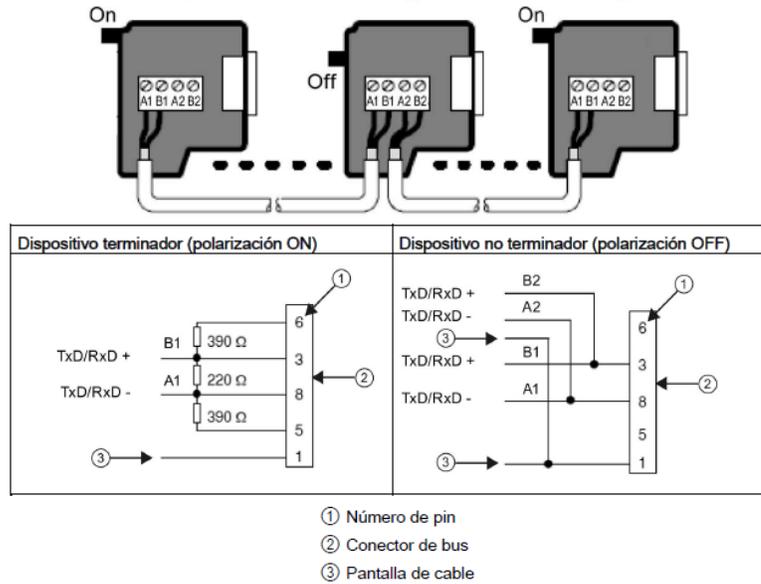


Figura 3.19: Terminación y Polarización para el conector RS-485

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

### 3.2.2 Configuración Red Profibus

#### a. Red

La red Profibus que se constituye está formada por dos CPUs Simatic S7-1200 conectadas mediante los módulos CM 1243-5 (para la CPU maestro) y CM 1242-5 (Para la CPU esclavo). Para establecer las conexiones lógicas entre los equipos se debe agregar ambos PLCs al proyecto y en la Pestaña Dispositivos y Redes se deberá arrastrar la conexión de un puerto a otro (véase Figura 3.20), el *software* TIA PORTAL crea las instancias necesarias para la conexión y posteriormente se configura los parámetros adecuados para la transmisión. La topología en esta red es de tipo bus con lo cual se puede agregar diferentes de esclavos al canal Profibus creado.

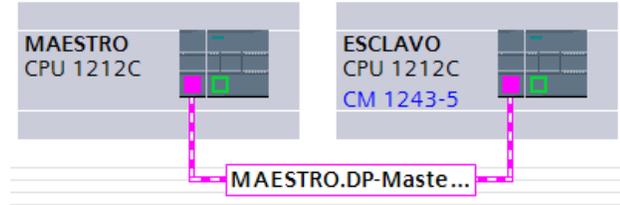


Figura 3.20: Red Profibus entre dos Simatic S7-1200

### b. Configuración Interfaz Profibus

En una red Profibus se puede establecer ciertas configuraciones para el bus, las cuales determinaran el comportamiento del estándar, dichos parámetros pueden ser accedidos mediante la Pestaña Configuración DP-Mastersystem al dar doble clic sobre el bus, dentro de la Categoría de Ajuste de Red se puede establecer tres parámetros básicos:

- Dirección de Profibus más alta: Corresponde a la dirección del último esclavo de la red, existe 127 posibles direcciones que serán abordadas con mayor detalle en el siguiente apartado.
- Velocidad de transferencia: La velocidad del bus puede ser configurada entre 9.6 Kbits y 12 Mbits.

Perfil: Los perfiles Profibus disponibles son: DP, Estándar, Universal (DP/FMS) o definido por el usuario, la elección del sistema adecuado determinará el cálculo de los parámetros de bus.

Figura 3.21: Ajustes de Red Profibus

Adicionalmente se puede añadir información acerca del medio de transmisión y equipos involucrados en la red con el fin de ajustar el estándar a las necesidades de campo, de esta

manera se puede escoger la línea de cobre o fibra óptica, el número de repetidores y longitudes de cable.

### c. Direcciones Profibus

En una red Profibus se puede conectar un cierto número de esclavos directamente a un dispositivo maestro, el número máximo de equipos es de 127 pero existen ciertas direcciones reservadas para funciones especiales, según lo presentado por (SIEMENS, S7 Controlador programable S7-1200, 2015) se puede destacar:

- Dirección 0: Reservada para acciones de configuración sobre la red o herramientas de programación relacionadas con el bus.
- Dirección 1: Reservada por Siemens para el primer dispositivo maestro dentro de la red.
- Dirección 126: Reservada para elementos de fábrica que no cuentan con el ajuste adecuado y deben ser determinados a través de red.
- Dirección 127: Reservada para la transmisión de tipo *broadcast*.

Las direcciones que se pueden asignar a dispositivos operativos van desde 2 a 125, la misma que debe ser unívoca y establecida en la Pestaña de Configuración General de cada módulo o dispositivo de la red.

### 3.2.3 Ejemplo de Conexión

En este apartado se presenta una aplicación a manera de ejemplo para la conexión de dos CPUs S7-1200 mediante una red Profibus, no existen estructuras específicas para el intercambio de datos Profibus, más bien es un tipo de red cuya aplicación está diseñada para una implementación rápida y eficaz utilizando la lectura y escritura de datos directamente en el área de imagen de proceso de los equipos. Por lo tanto en el ejemplo se expone la manera de enlazar las áreas de transferencias de datos entre el maestro y el esclavo.

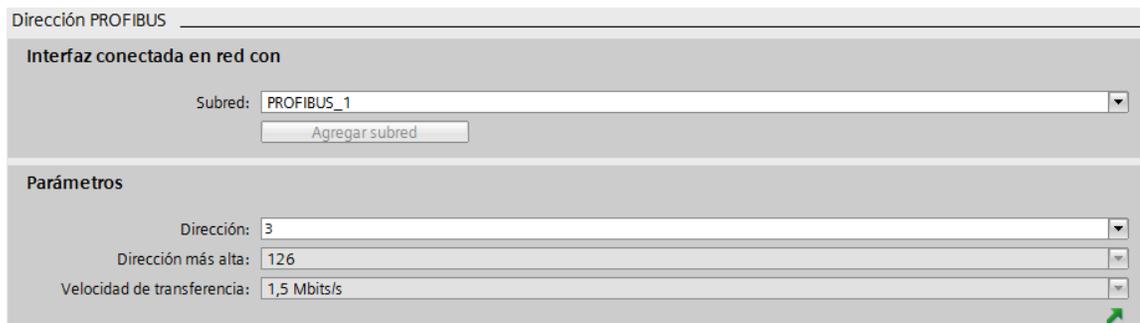
### a. Configuración de *Hardware* para los PLCs

Para establecer la red Profibus se debe agregar los dos PLCs a un proyecto nuevo con sus correspondientes módulos de comunicación, para insertar estos componentes es preciso situarse en la ventana de configuración del dispositivo, en la parte derecha se escoge y arrastra el equipo Profibus desde el catálogo de *hardware*.

En los módulos de maestro y esclavo se deberá escoger una conexión a la subred Profibus creada y establecer el rol que cumplirán dentro de la misma, en el siguiente apartado se explica lo mencionado.

### b. Programación Esclavo

El PLC2 figurará como esclavo en la red Profibus, la dirección dentro de la misma se especifica en el pestaña Configuraciones del módulo, el identificador escogido será el número 3 dentro de la subred Profibus\_1 (véase Figura 3.22).



The screenshot shows a configuration window titled "Dirección PROFIBUS". It is divided into two main sections: "Interfaz conectada en red con" and "Parámetros".

- Interfaz conectada en red con:** This section contains a dropdown menu labeled "Subred:" with the value "PROFIBUS\_1" selected. Below the dropdown is a button labeled "Agregar subred".
- Parámetros:** This section contains three input fields:
  - "Dirección:" with the value "3" selected.
  - "Dirección más alta:" with the value "126" selected.
  - "Velocidad de transferencia:" with the value "1,5 Mbits/s" selected.

A green arrow icon is visible in the bottom right corner of the window.

Figura 3.22: Dirección Profibus Esclavo

Para que el dispositivo pueda ser configurado y accedido en la red física se deberá escoger su modo de operación (en este caso esclavo) y además el equipo maestro bajo el cual será controlado, por lo tanto se deberá seleccionar la interfaz de comunicación DP creada mediante el módulo del maestro (véase Figura 3.23).

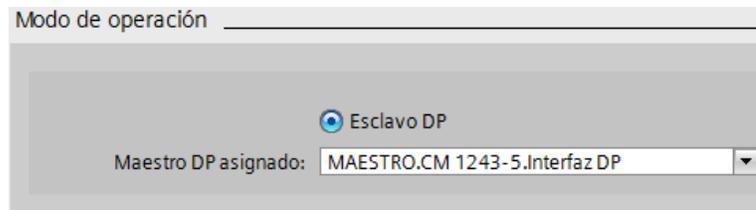


Figura 3.23: Selección Maestro Asignado

El equipo esclavo cumple el rol de direccionar las áreas de transferencia de salida o entrada que estarán enlazadas con el maestro, la configuración preestablecida para redes Profibus permite la asignación de estas áreas directamente en la Imagen de Proceso de cada dispositivo con un rango de direcciones desde 0 a 1023.

Las áreas de transferencia dentro de la red se enlazan de manera “cruzada”, es decir, en el esclavo se puede configurar el área de entrada de imagen de proceso para el intercambio de datos con la sección de salida de imagen de proceso del maestro, de igual manera se puede realizar la configuración opuesta. Estos canales de comunicación pueden ser optimizados para el uso de intercambio de datos de tipo byte o palabras. En el ejemplo se crean cuatro áreas de transferencia de datos tanto para enviar y recibir información desde el maestro (véase Figura 3.24).

Áreas de transferencia							
...	Área de transferencia	Tipo	Dirección del maestro	↔	Dirección del esclavo	Longitud	Coherencia
1	AT_TxBool	MS	Q 1	→	I 1	1 Byte	Unidad
2	AT_TxRegistro	MS	Q 100...101	→	I 100...101	1 Palabra	Unidad
3	AT_RxRegistro	MS	I 100...101	←	Q 100...101	1 Palabra	Unidad
4	AT_RxBool	MS	I 10	←	Q 10	1 Byte	Unidad

Figura 3.24: Áreas de Transferencia Profibus

Los parámetros principales a configurar son: el tipo de dirección (Q o I), la dirección inicial donde comienza el área, la longitud de datos que se va a transferir según el tipo de unidad que se maneje y el tipo de datos (byte o palabra).

### c. Programación Maestro

El PLC1 figurará como maestro en la red Profibus y se deberá especificar su dirección de acceso, para lo cual se procede a dar doble clic sobre el módulo de comunicación y en la *Ventana Propiedades* se escogerá una dirección válida, en este caso corresponde a la número 2 (véase Figura 3.25). Luego se deberá escoger el rol que cumple el equipo dentro de la red, en el caso del módulo CM 1243-5 tiene solamente la capacidad de desempeñarse como maestro.

Dirección PROFIBUS	
<b>Interfaz conectada en red con</b>	
Subred:	PROFIBUS_1
	<input type="button" value="Agregar subred"/>
<b>Parámetros</b>	
Dirección:	2
Dirección más alta:	126
Velocidad de transferencia:	1,5 Mbits/s

Figura 3.25: Dirección Profibus Maestro

En el bloque de programación del PLC1 simplemente se agregará los segmentos necesarios para controlar la lógica, utilizando las direcciones de las áreas de transferencia que son especificadas en el esclavo.

En el ejemplo el dispositivo maestro realizará cuatro funciones las cuales incluyen acciones de lectura y escritura de datos de tipo byte con el fin de visualizar y controlar las salidas digitales del esclavo; además se implementa una comunicación enfocada a la transmisión y lectura de palabras de registro de tipo Word entre ambos dispositivos.

#### 3.2.4 Pruebas de Funcionamiento

El ejemplo realizado para la comunicación de dos dispositivos en una red Profibus está enfocado al uso y aplicación de las estructuras de transferencia entre dispositivos maestros y esclavos, para lo cual se configura dos áreas de escritura desde el maestro para enviar

datos binarios o valores de registro, y además se agrega dos áreas de lectura para obtener estados binarios de las salidas o valores de registro del esclavo.

Los datos enviados desde el maestro serán visualizados en el esclavo mediante una tabla de observación adecuada, por el contrario los datos recibidos serán presentados en una segunda tabla de observación creada en el PLC1. A continuación se presentan las figuras correspondientes en cada caso para comprobar el correcto funcionamiento de la red.

**a. Enviar Datos de Tipo Byte**

La primera función que se realiza mediante la red es la transferencia de un byte de datos desde el maestro hacia el esclavo, el PLC1 controlará el valor de estado de los dos primeros bits del byte a enviar. El área de transferencia que corresponde es la siguiente: dirección del maestro (byte Q1) y dirección del esclavo (byte I1). En la primera parte se puede comprobar el estado de las salidas mencionadas y los datos recibidos mediante la Figura 3.26.

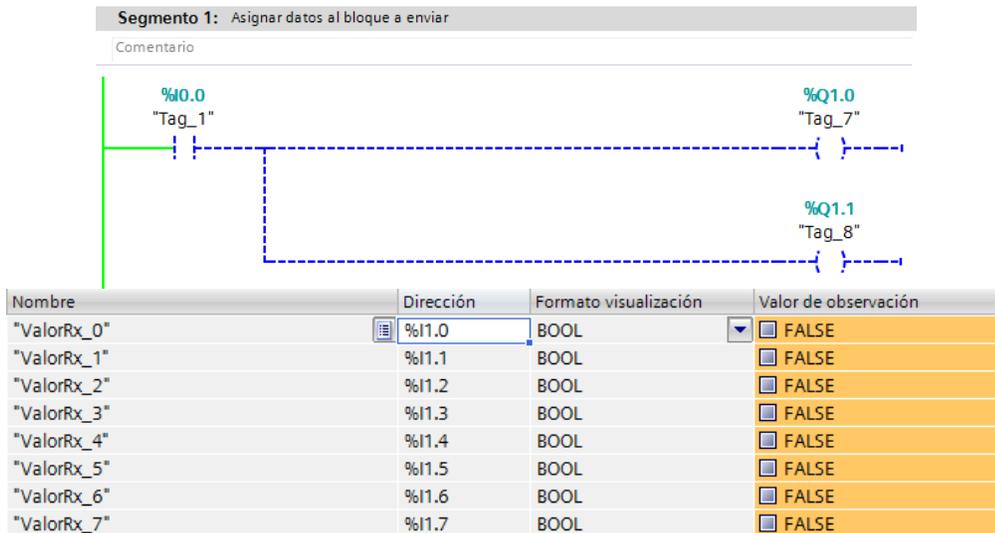


Figura 3.26: Enviar datos tipo Byte Parte 1, Profibus

Luego de conmutar la entrada I0.0 del maestro se activará los bits Q1.0 y Q1.1 de tal manera que los datos transmitidos para el byte Q1 serán %11000000, los nuevos valores que se pueden constatar en el PLC2 son los expuestos en la Figura 3.27.

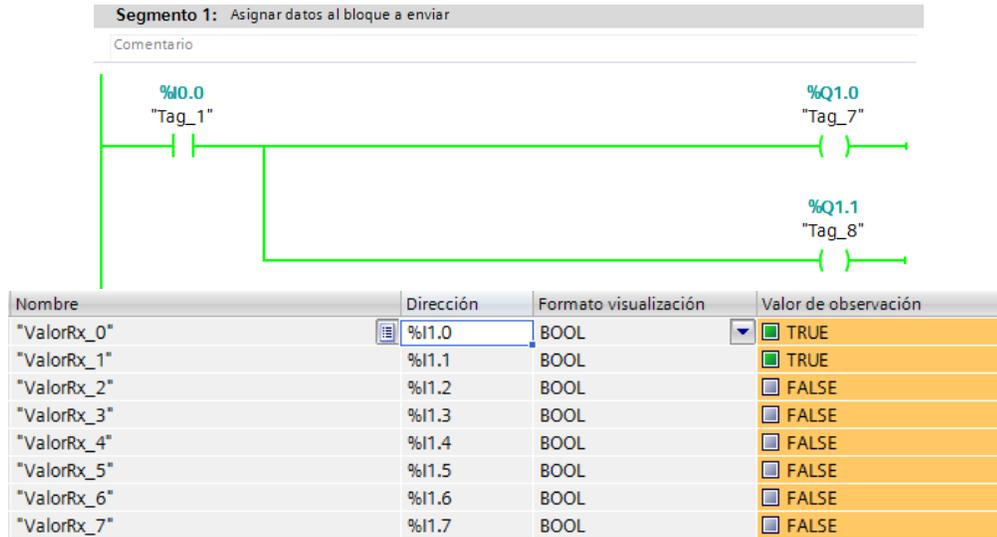


Figura 3.27: Enviar datos tipo Byte Parte 2, Profibus

### b. Enviar Datos de Tipo Word

La segunda acción de intercambio de datos que se cumple es la transmisión de un valor de registro desde el maestro hacia el esclavo, el PLC1 maneja el valor de la palabra de tipo WORD mediante la activación de la entrada I0.2. El área de transferencia que corresponde es la siguiente: dirección del maestro (byte QW100) y dirección del esclavo (byte IW100). En la Figura 3.28 se puede comprobar el estado del registro al no estar activada la entrada mencionada, por lo tanto se envía la cantidad de 100 en base decimal.

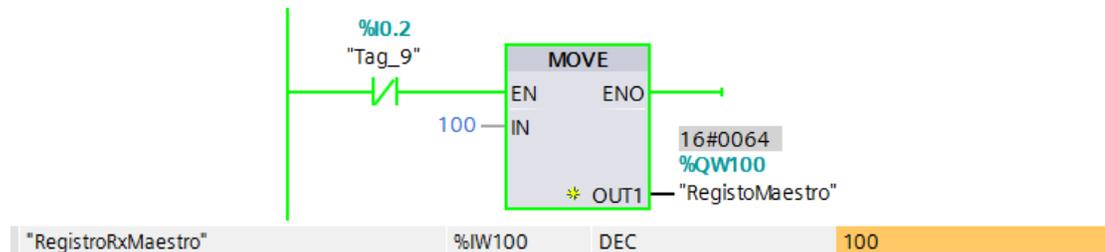


Figura 3.28: Enviar datos tipo Word Parte 1, Profibus

En la segunda parte al activar la entrada I0.2 se establece un valor de 200 en base decimal para el registro de datos, el valor recibido por el esclavo se puede observar en la Figura 3.29.

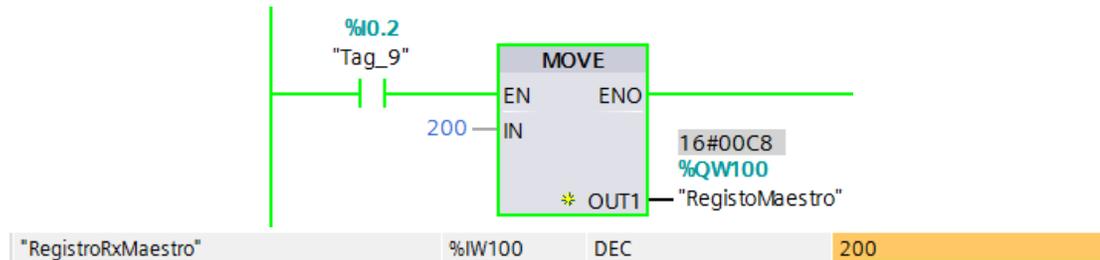


Figura 3.29: Enviar datos tipo Word Parte 2, Profibus

### c. Recibir Datos de Tipo Byte

La siguiente función que se demuestra es la lectura de un área de datos de tipo Byte desde el esclavo, el estado de las seis salidas físicas del PLC2 será accedido mediante el controlador principal.

En el esclavo se encienden los bits Q0.0, Q0.1 y Q0.2 a través de la entrada I0.0, el valor del primer byte Q0 es copiado al área de transferencia de datos cuyas direcciones son: dirección del maestro (byte I10) y dirección del esclavo (byte Q10). Si la entrada I0.0 se encuentra desactivada se enviará un valor de “0” para todos los bits (véase Figura 3.30).

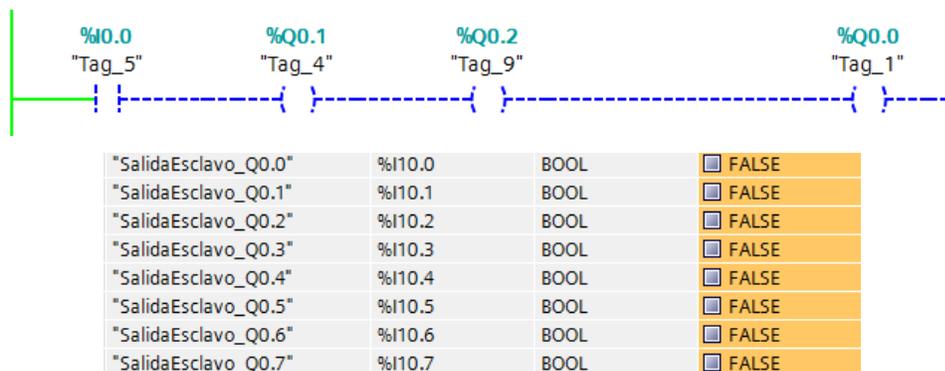


Figura 3.30: Recibir datos tipo Byte Parte 1, Profibus

Luego de accionar la entrada I0.0 se activa los tres bits mencionados y su valor es copiado al área de transferencia y enviado al maestro, los nuevos datos recibidos se pueden comprobar en la Figura 3.31.



Figura 3.31: Recibir datos tipo Byte Parte 2, Profibus

#### d. Recibir Datos de Tipo Word

El último ejemplo que se presenta es la lectura de un registro de datos de tipo WORD del esclavo, el PLC2 maneja el valor de la palabra de datos mediante la activación de la entrada I0.0, el área de transferencia que corresponde es la siguiente: dirección del maestro (byte IW100) y dirección del esclavo (byte QW100). En la primera prueba de funcionamiento se puede comprobar el estado del registro del esclavo al no estar activada la entrada especificada, por lo tanto se envía la cantidad de 300 en base decimal (véase Figura 3.32).

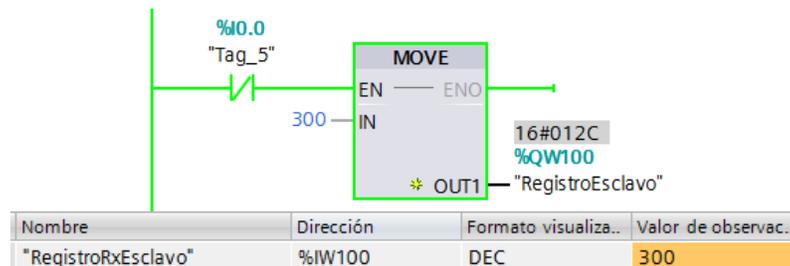


Figura 3.32: Recibir datos tipo Word Parte 1, Profibus

Posteriormente al activar la entrada I0.0 se establece un valor de 400 en base decimal para el registro de datos, el valor recibido por el maestro se puede observar en la Figura 3.33.

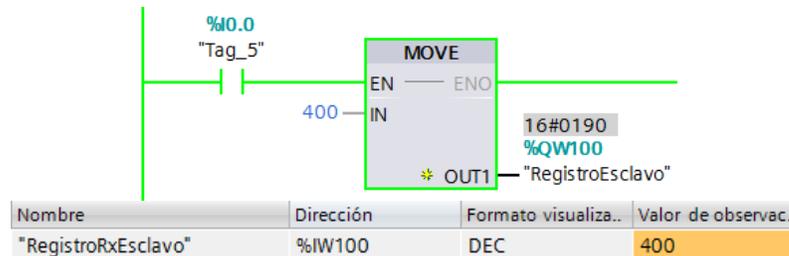


Figura 3.33: Recibir datos tipo Word Parte 2, Profibus

### 3.3 RED MODBUS-RTU

Una red de comunicación Modbus-RTU puede ser establecida entre equipos Simatic S7-1200 mediante el uso del módulo CM 1241 (véase Figura 3.34) para intercambio de datos punto a punto y un cable eléctrico de interfaz adecuado. En esta red se maneja el modelo maestro-esclavo y el uso de los códigos de función Modbus para acceder a la información de cada dispositivo participante. Las características del protocolo Modbus se encuentran explicadas en la sección 1.4.

#### 3.3.1 Elementos del Sistema

##### 3.3.1.1 Módulo RS-485 (CM 1241)

El módulo de comunicaciones CM 1241 (véase Figura 3.34) brinda a una CPU S7-1200 las capacidad de conexión a una red RS-422/485, estas redes son implementadas para comunicaciones PtP<sup>44</sup>, USS<sup>45</sup> o Modbus, de esta manera la CPU puede actuar como maestro Modbus-RTU, por lo tanto puede leer y escribir datos o estados de I/O en un esclavo, también se puede comportar como esclavo Modbus-RTU permitiendo que el dispositivo de supervisión controle peticiones de acceso a sus registros.

<sup>44</sup> PtP: Punto a Punto

<sup>45</sup> USS (*universelles seriellen SchnittstellenProtokoll*, "Protocolo de Interfaz Serie Universal"): Protocolo de comunicación serie para la comunicaiicon con accionamiento de manera sencilla, diseñado por la empresa SIEMENS.



Figura 3.34: Módulo CM 1241 Esclavo Profibus-DP

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

El módulo CM 1241 soporta las configuraciones propias de interfaces RS-485 como: velocidad de transferencia entre 300 baudios y 115,2 Kbaudios, paridad, bits de datos por carácter, número de bits de parada, tiempo de espera, control de flujo por *software*, el control de flujo físico solo está implementado para interfaces RS-232.

El módulo tiene la posibilidad de usar *drivers* para protocolos ASCII y RTU con las librerías adecuadas en TIA PORTAL, la configuración se lo realiza mediante este *software* con dos versiones de *firmware* que permite el uso de instrucciones Modbus-RTU en comunicación serie o instrucciones extendidas con la capacidad de conectarse a periferia descentralizada de Profinet y Profibus (SIEMENS, S7 Controlador programable S7-1200, 2015).

### 3.3.1.2 Cable y Conectores

Para la conexión de los módulos CM 1241 en una red que utiliza protocolos Modbus se debe usar cable apantallado de 2 hilos más aterramiento, de igual manera que en las redes Profibus sobre RS-485, la red Modbus utiliza conectores DB9 hembras a ambos extremos del cable. Se utiliza también los pines tres y ocho para la transmisión de datos mediante los conductores denominados A y B, se incluye la carcasa para el aterramiento del bus.

Además se puede usar una resistencia terminadora  $R_t$  ( $120\Omega$ ) y una capacitancia ( $1nF$ ) en los extremos de la red con el fin de evitar las perturbaciones o reflexiones en un ambiente

hostil, se puede también incluir una resistencia  $R_g$  entre la línea de 0V y los controladores para limitar corrientes de retorno al común (véase Figura 3.35) (Candelas, 2011).

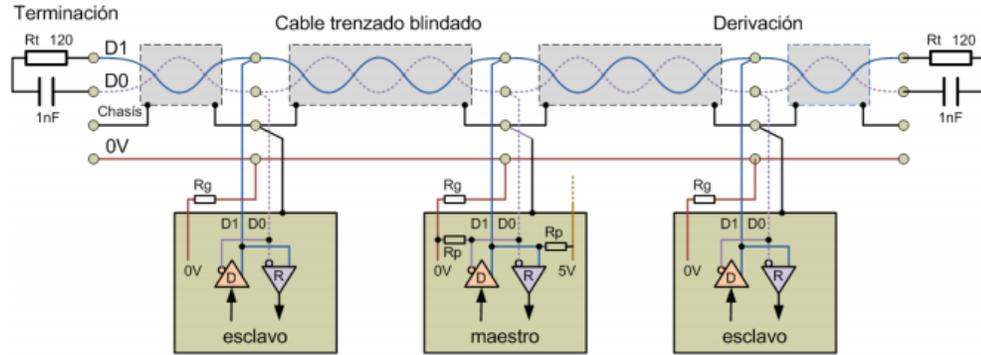


Figura 3.35: Cableado Red Modbus sobre RS-485

Fuente: (Candelas, 2011)

Es posible utilizar conectores DB9 estándar y cables apantallados para establecer la comunicación en el bus mediante un mecanismo que permita la conexión a varios esclavos. Para la realización de la red de este proyecto se puede omitir los elementos terminadores como resistencias y capacitancias, debido a que el ambiente donde se implementara la red no está sometido a ruido electromagnético y dado que la distancia entre los equipos es corta.

### 3.3.2 Configuración Red Modbus-RTU

#### a. Red

La red Modbus-RTU que se establece está formada por dos CPUs Simatic S7-1200 conjuntamente con dos módulos de comunicación conectadas mediante un cable eléctrico de interfaz RS-485. La topología que se sigue en una red básica como esta, es de tipo bus. Para el caso de Modbus-RTU, este es un estándar en el cual existe la posibilidad de conectar hasta 247 equipos subordinados por un solo dispositivo maestro, las comunicaciones son dirigidas y controladas por el elemento central, los esclavos solamente pueden responder a las peticiones direccionadas hacia ellos.

### **b. Configuración de las Conexiones, Códigos de Función**

Las comunicaciones en una red Modbus-RTU montadas con CPUs Siemens son manejadas a través de la dirección ID del módulo de comunicación serie o puerto que este ocupe dentro de la CPU (se especifica al momento de configurar el canal), además se debe conocer y especificar la dirección de los esclavos involucrados en la red.

Un puerto de comunicación cuyo fin es el de maestro de la red no podrá ser utilizado para intercambiar datos como esclavo ni viceversa, la configuración de la conexión se la especifica solo una vez al inicio durante el despliegue de la red. Las peticiones que son enviadas por el maestro pueden ser múltiples, controladas de manera sincronizada. Cada petición debe ocupar el mismo puerto de comunicación y bloque de datos de instancia con el fin de mantener la coherencia para los parámetros de la red especificados previamente.

El canal establecido soporta el intercambio de información según los códigos de función propios para el protocolo Modbus, de esta manera se puede implementar peticiones de lectura que permitan acceder a los datos de entrada-salida del esclavo, así también como a sus registros de entrada y de almacenamiento, en la siguiente tabla se observa el código de función implementado para cada caso y su máxima capacidad de acceso por petición.

Tabla 3.2: Códigos de Función para lectura Modbus

<b>Código de función Modbus</b>	<b>Funciones de lectura de esclavo (servidor), direccionamiento estándar</b>
01	Leer bits de salida: De 1 a 2000 bits por petición
02	Leer bits de entrada: De 1 a 2000 bits por petición
03	Leer registros de retención: De 1 a 125 palabras por petición
04	Leer palabras de entrada: De 1 a 125 palabras por petición

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

Existe también códigos de función adecuados que permiten el acceso de escritura a los registros de los esclavos, las áreas de memoria que pueden ser intervenidas son: el área de imagen de proceso de salida y el área de registros de almacenamiento o retención del

protocolo, en la tabla expuesta a continuación se puede constatar el número de función y su respectiva descripción.

Tabla 3.3: Códigos de Función para escritura Modbus

Código de función Modbus	Funciones de escritura de esclavo (servidor), direccionamiento estándar
05	Escribir un bit de salida: 1 bit por petición
06	Escribir un registro de retención: 1 palabra por petición
15	Escribir uno o más bits de salida: De 1 a 1968 bits por petición
16	Escribir uno o más registros de retención: De 1 a 123 palabras por petición

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

Adicionalmente se puede realizar peticiones mediante los códigos de función 08 y 11 que están enfocados a la transmisión de información de diagnóstico de los equipos subordinados. El código 0 es utilizado para enviar un mensaje de tipo *broadcast* (no se espera respuesta de los esclavos), esta función no está implementada para Modbus-TCP debido a que el acceso en dicha red se lo realiza mediante direcciones IP y puertos de la CPU.

### c. Instrucciones Modbus-RTU

En el paquete de Instrucciones de Comunicación dentro la categoría Procesadores de Comunicación Modbus-RTU situada en la parte derecha de TIA PORTAL se puede encontrar las instrucciones necesarias para la comunicación mediante el protocolo RTU, estos controles permiten configurar, enviar y recibir datos entre dos equipos conectados con el modelo Maestro-Esclavo, a continuación se describe las funciones de las instrucciones necesarias para establecer una comunicación Modbus vía RS-485.

Instrucción MB\_COMM\_LOAD: Esta instrucción permite configurar un puerto de comunicación de los módulos serie CM 1241 RS-485 o CM 1241 RS-232 para el uso del protocolo Modbus-RTU. La configuración se la debe realizar una sola vez al inicio de la

red y deberá ser llamada nuevamente en el caso que se desee cambiar los parámetros de funcionamiento. Esta estructura tiene asociado un bloque de datos de instancia que corresponde al uso del puerto, sea maestro o esclavo.

**Instrucción MB\_MASTER:** Esta instrucción permite a la CPU comunicarse como maestro Modbus dentro de la red, se puede acceder a uno o varios controladores esclavos. Para el correcto funcionamiento de la estructura se debe tomar en cuenta lo siguiente:

- Un puerto usado como maestro no puede figurar como esclavo.
- Es posible utilizar el puerto para varias llamadas de MB\_MASTER con el mismo bloque de datos de instancia.
- Las estructuras Modbus no poseen eventos de alarma para controlar la comunicación, por lo cual el programa debe consultar la instrucción para determinar acciones de transmisión o recepción concluidas.
- La llamada a la estructura MB\_MASTER debe prologarse hasta que la respuesta del esclavo o un código de error sea recibido.
- MB\_MASTER debe ejecutarse de preferencia en bloques de programación cíclica.

**Instrucción MB\_SLAVE:** Esta instrucción permite a la CPU comunicarse como esclavo Modbus dentro de la red, al momento que el esclavo recibe una petición esta estructura es la encargada de la ejecución y respuesta de la misma. Para el correcto funcionamiento de la estructura se debe tomar en cuenta lo siguiente:

- Un puerto usado como esclavo no puede ser utilizado como maestro.
- Las estructuras Modbus no poseen eventos de alarma para controlar la comunicación, por lo cual el programa debe consultar la instrucción para determinar acciones de transmisión o recepción concluidas.
- MB\_SLAVE debe ejecutarse en bloques de programación cíclica con el fin de garantizar su frecuencia de funcionamiento y permitir procesar las respuestas actuales.

En la instrucción MB\_MASTER se tiene dos parámetros mediante los cuales se establece el código de función o petición que será enviada al esclavo, estos son el parámetro MODE y MB\_DATA\_ADDR. La combinación de los valores configurados en estas entradas permite determinar si el cliente desea leer, escribir o recibir información de diagnóstico. En la siguiente tabla se puede observar los principales códigos de función Modbus y su correspondencia a las variables mencionados.

Tabla 3.4: Códigos de función vs parámetros de la Instrucción MB\_MASTER

MB_MODE	MB_DATA_ADDR	Longitud de datos	Código de función Modbus activado	Operación y datos
0	De 1 a 9999	De 1 a 2000	01	Leer bits de salida: De 1 a 2000 bits por petición
0	De 10001 a 19999	De 1 a 2000	02	Leer bits de entrada: De 1 a 2000 bits por petición
0	De 40001 a 49999 o De 400001 a 465535	De 1 a 125	03	Leer registros de retención: De 1 a 125 palabras por petición
0	De 30001 a 39999	De 1 a 125	04	Leer palabras de entrada: De 1 a 125 palabras por petición
1	De 1 a 9999	1	05	Escribir un bit de salida: Un bit por petición
1	De 40001 a 49999 o De 400001 a 465535	1	06	Escribir un registro de retención: 1 palabra por petición
1	De 1 a 9999	De 2 a 1968	15	Escribir varios bits de salida: De 2 a 1968 bits por petición
1	De 40001 a 49999 o De 400001 a 465535	De 2 a 123	16	Escribir varios registros de retención: de 2 a 123 palabras por petición

Fuente: (SIEMENS, S7 Controlador programable S7-1200, 2015)

### 3.3.3 Ejemplo de Conexión

En este apartado se presenta una aplicación para la conexión dos CPUs Simatic S7-1200 mediante la implementación del protocolo RTU en una red Modbus, para esto se hará uso de las instrucciones MB\_COMM\_LOAD, MB\_MASTER y MB\_SLAVE que permitirán el intercambio de peticiones-respuestas entre un dispositivo maestro y uno esclavo, además se hará uso de las diferentes variantes de la estructura de solicitud en el maestro para enviar cuatro requerimientos de acceso al equipo subordinado.

### a. Configuración de *Hardware* Maestro

Para establecer la red Modbus-RTU se debe agregar los dos PLCs a un proyecto nuevo con sus correspondientes módulos de comunicación CM 1241, para insertar estos componente es necesario situarse en la ventana de configuración del dispositivo, en la parte derecha se escoge y arrastra el equipo desde el catálogo de *hardware*.

El módulo posee ciertas características que pueden ser accedidas mediante su ventana de configuración de dispositivo, en este caso las configuraciones están orientadas hacia la interfaz RS-485 de manera que se puede escoger modos de comunicación, velocidades, preferencias de transmisión y recepción. Posteriormente con la instrucción MB\_COMM\_LOAD se establecerá los valores necesarios para los principales parámetros.

El módulo CM 1241 está en la capacidad de utilizar tanto la interfaz RS-485 como la RS-422, por lo cual en el modo de operación del dispositivo se deberá escoger el adecuado para la red, en este caso será una transmisión *Semidúplex* (RS-485) a 2 hilos (véase Figura 3.36).

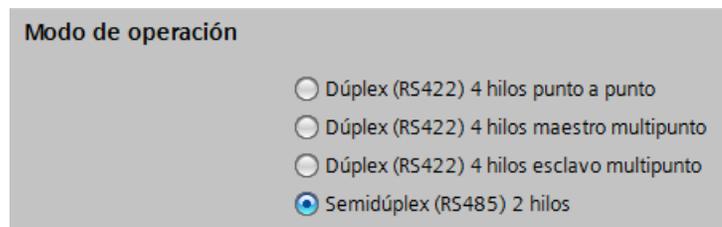


Figura 3.36: Modo de Operación módulo CM 1241

### b. Programación PLC1 (Maestro)

Antes de iniciar con la transferencia de información se debe ejecutar la instrucción MB\_COMM\_LOAD que permite configurar el puerto del módulo de comunicación para el intercambio de datos dentro de la red, esta instrucción debe ser ejecutada en el arranque del PLC. A continuación se explica sus principales variables:

- REQ: Flanco ascendente para la ejecución de la instrucción, se controla mediante la marca del sistema *FirstScan*.
- PORT: Hace referencia al número del puerto físico insertado para la CPU, este valor puede ser accedido mediante la tabla de variables del PLC en la ficha “Constantes”, para el módulo añadido se tiene el número 269.
- BAUD: Velocidad de transferencia de la interfaz, se puede escoger entre: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 y 115200 baudios/s.
- PARITY: Selección para la paridad en los datos: 0 corresponde a Ninguna, 1 significa paridad Impar y 2 paridad Par.
- MB\_DB: Referencia a los bloques de datos de instancia de estructuras MB\_MASTER o MB\_SLAVE según el uso del puerto, en este caso maestro.

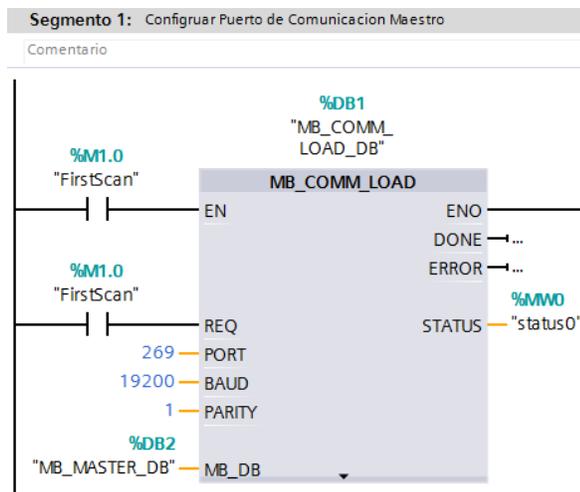


Figura 3.37: Instrucción MB\_COMM\_LOAD en Maestro

El PLC1 figurará como maestro en la red Modbus, gracias a los códigos de función Modbus implementados para los equipos Simatic se puede leer y escribir datos en la CPU remota tanto en el área de Imagen de Proceso como en otras áreas relacionadas a registros de almacenamiento de tipo WORD.

La estructura MB\_MASTER es la encargada de realizar y procesar cualquier tipo de petición hacia los esclavos por lo tanto a continuación se explica sus parámetros más

relevantes y posteriormente se adjuntará las estructuras creadas para cada demanda específica.

- REQ: Mientras su valor sea “1” o TRUE la instrucción enviará peticiones de comunicación hacia el esclavo.
- MB\_ADDR: Corresponde a la dirección Modbus del esclavo al que se envía la petición, direccionamiento estándar de 0 a 247 y ampliado de 0 65535.
- MB\_MODE: Su valor es de 0, 1 ó 2, permite configurar la petición para lectura, escritura o diagnóstico.
- MB\_DATA\_ADDR: En este parámetro se especifica la dirección inicial que se desea intervenir en el esclavo, sea para imagen de proceso o valores de registros.
- MB\_DATA\_LEN: Longitud de los datos o registros a intervenir.
- MB\_DATA\_PTR: En este parámetro se establece el puntero al registro de datos Modbus donde será colocada la información recibida del esclavo, puede ser un bloque de datos global o un área de memoria de tipo M.

En el ejemplo se realiza cuatro tipos de peticiones controladas por entradas del PLC, cada solicitud es enviada al detectarse un flanco ascendente en su variable de control, de esta manera las solicitudes pueden ser manejadas de forma sincronizada a través del mismo bloque de datos de instancia.

La primera petición corresponde a leer el valor *ON/OFF* de las seis salidas del esclavo, el código de función Modbus correspondiente sería el 01. La segunda solicitud permite escribir valores binarios para tres salidas del PLC2, el código Modbus correspondiente es el 15. La tercera petición se realiza para leer los registros de almacenamiento del esclavo, código de función 03. El último requerimiento permite escribir un valor de registro de almacenamiento en el equipo subordinado mediante el código Modbus 06.

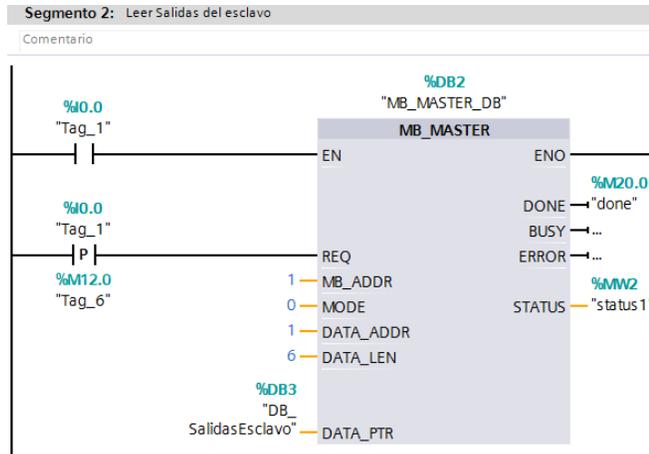


Figura 3.38: Instrucción MB\_MASTER, leer salidas de un esclavo

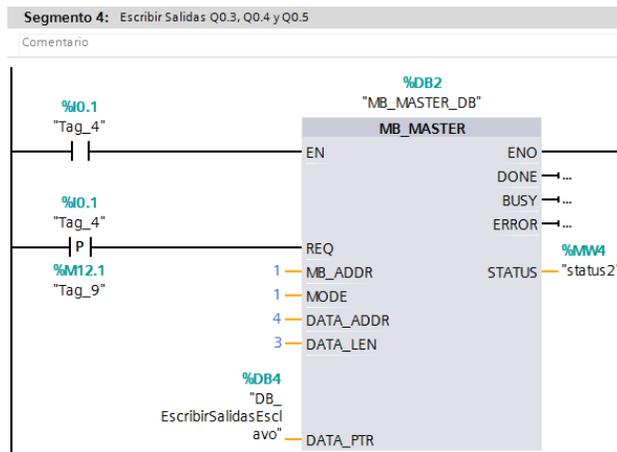


Figura 3.39: Instrucción MB\_MASTER, escribir salidas de un esclavo

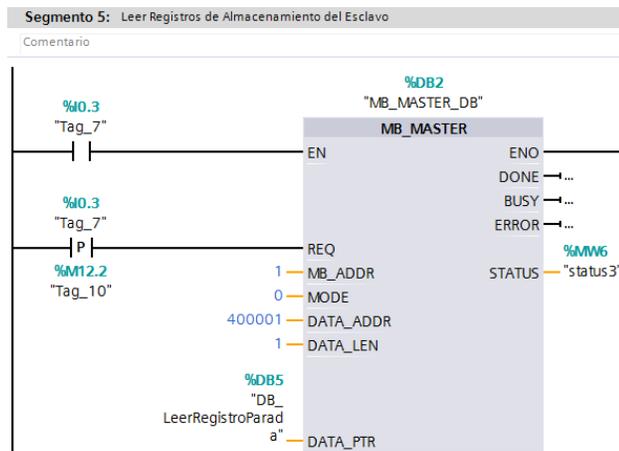


Figura 3.40: Instrucción MB\_MASTER, leer registros de almacenamiento

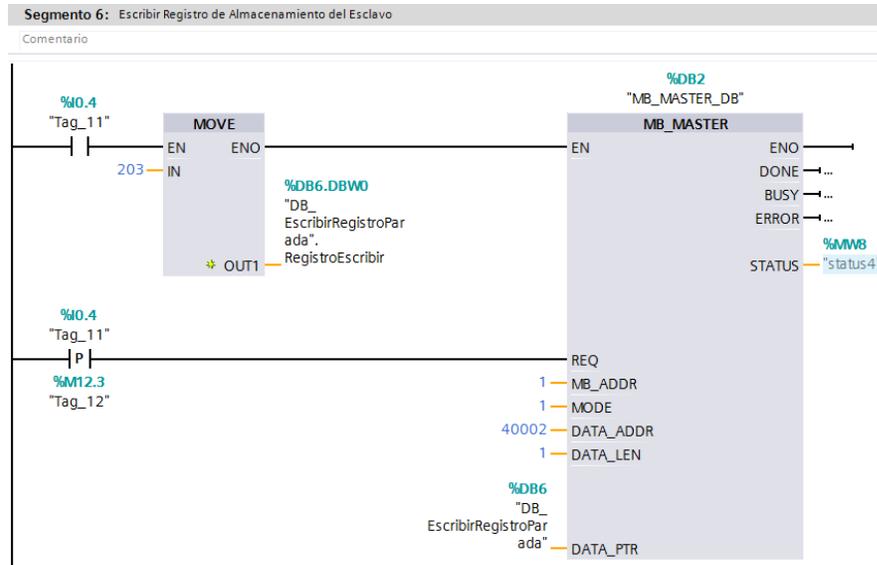


Figura 3.41: Instrucción MB\_MASTER, escribir registros de almacenamiento

### c. Programación PLC2 (Esclavo)

En el dispositivo esclavo se debe ejecutar también la instrucción MB\_COMM\_LOAD con el fin de configurar el puerto del módulo para su función en la red Modbus en calidad de esclavo. Los parámetros son los mismos explicados en la sección anterior con la única diferencia que en este caso se deberá escoger el bloque de datos de instancia correspondiente a una estructura MB\_SLAVE, además del número de puerto del *hardware* (véase Figura 3.42).

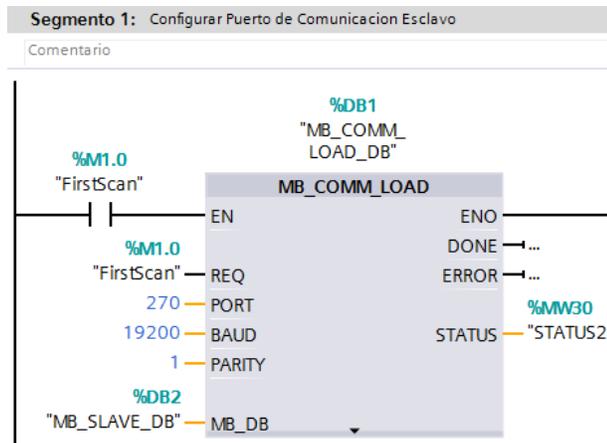


Figura 3.42: Instrucción MB\_COMM\_LOAD en Esclavo

El PLC2 trabajará como esclavo en la red Modbus, por lo tanto es el equipo encargado de recibir la comunicación y responder a las peticiones requeridas por el maestro, la estructura MB\_SLAVE deberá ser añadida al proyecto una sola vez, esta instrucción procesa todos los tipos de requerimientos y establece las respuestas según los códigos de función conocidos. Los únicos parámetros que se deberán especificar son:

- MB\_ADDR: Corresponde a la dirección Modbus, este es el identificador para cada equipo dentro de la red.
- MB\_HOLD\_REG: En este parámetro se establece el puntero al registro de datos de almacenamiento Modbus donde será colocada la información interna del esclavo, puede ser un bloque de datos global o un área de memoria de tipo M.

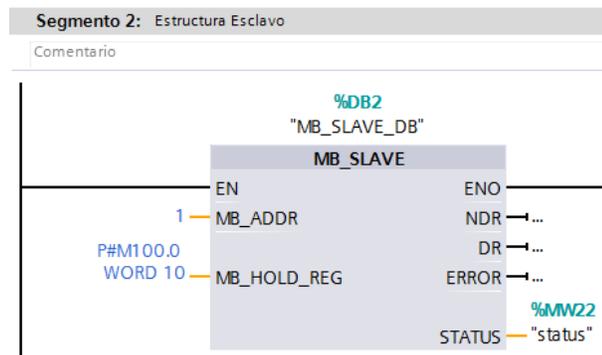


Figura 3.43: Instrucción MB\_SLAVE

### 3.3.4 Pruebas de Funcionamiento

La aplicación realizada para la comunicación de dos dispositivos en una red Modbus-RTU está enfocada al uso y aplicación de la estructura MB\_MASTER para enviar múltiples peticiones siguiendo el estándar Modbus. Como se explicó en el apartado anterior se realiza cuatro peticiones desde el maestro para acciones de lectura y escritura en el esclavo.

Los datos de respuesta que son enviados de vuelta al maestro serán visualizados en una tabla de observación adecuada, mientras que los datos transmitidos para acciones de escritura hacia el esclavo se visualizarán en el PLC2 mediante una segunda tabla de

observación. A continuación se presenta las figuras correspondientes en cada caso para comprobar el correcto funcionamiento de la red.

### a. Petición 1

La primera solicitud que se envía es la correspondiente a la lectura de las variables de salida del esclavo, en el programa de control del PLC2 se prende o apaga las salidas Q0.0 y Q0.1 mediante la entrada I0.0. La CPU cuenta con seis salidas físicas cuyos datos serán enviados al maestro y almacenados en un bloque de datos global para su posterior acceso o tratamiento. En la primera parte se puede comprobar el estado de las salidas mencionadas y los datos recibidos mediante la Figura 3.44.

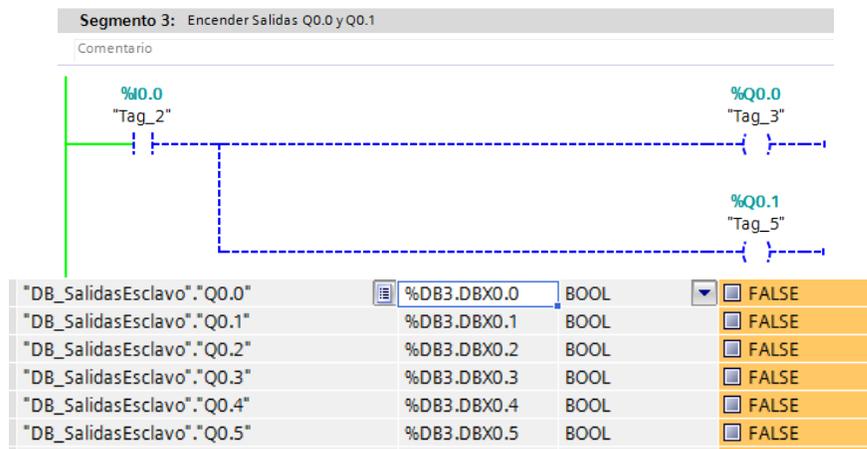


Figura 3.44: Petición Uno Parte 1, Modbus-RTU

Luego de conmutar la entrada I0.0 del esclavo se accionan las salidas indicadas y los datos serán transmitidos al PLC1, los nuevos valores para dichas variables se los puede verificar en la Figura 3.45.

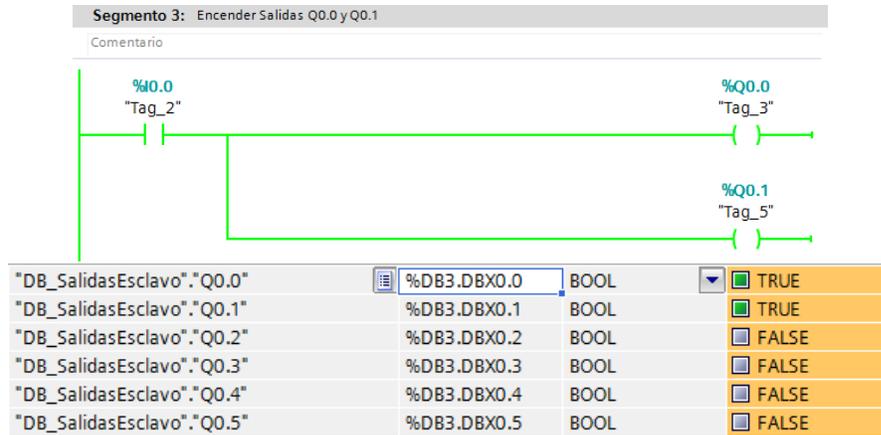


Figura 3.45: Petición Uno Parte 2, Modbus-RTU

**b. Petición 2**

Esta petición se realiza con el fin del tener acceso a funciones de escritura en los bits de salida del esclavo, desde el maestro se establece los niveles de salida para tres variables convenientemente establecidas dentro de un bloque de datos global, mediante la entrada I0.2 del PLC1 se establece un nivel alto para las variables a ser transmitidas. Los datos binarios son enviados al esclavo y permiten controlar las salidas Q0.3, Q0.4 y Q0.5. En la Figura 3.46 se expone el valor de “0” enviado para las tres salidas.

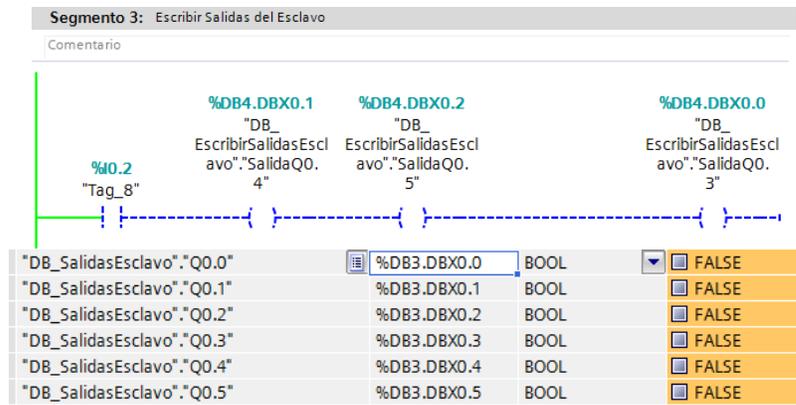


Figura 3.46: Petición Dos Parte 1, Modbus-RTU

La petición responde a un flanco ascendente por lo cual se deberá activar luego de establecer el nivel binario que se desee enviar. Para la segunda parte se procede a accionar

la entrada I0.2 del maestro con lo cual los valores de “1” para las variables del bloque de datos son transmitidos hacia el esclavo. Los niveles obtenidos para estos *bits* se muestran en la Figura 3.47.

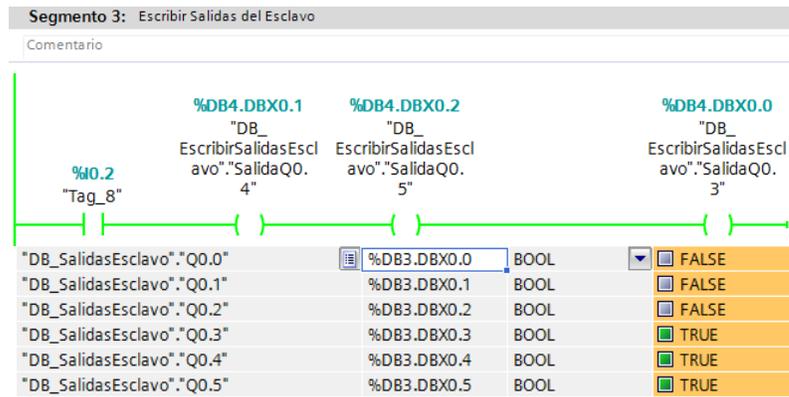


Figura 3.47: Petición Dos Parte 2, Modbus-RTU

### c. Petición 3

El tercer requerimiento solicitado desde el Maestro permite el acceso a funciones de lectura de los registros de almacenamiento del esclavo, estos registros son áreas de memoria de la CPU en las cuales el equipo subordinado almacena valores internos del estándar, generalmente parámetros de configuración o valores medidos para variables analógicas, estos registros empiezan en la dirección 40001 de Modbus.

Al momento de configurar el equipo como esclavo se escoge el área de memoria pertinente en donde se desean guardar estos valores, para el caso del ejemplo se escogió un puntero hacia un área de memoria de marcas empezando en la dirección 100. En el PLC2 se puede observar el primer valor mediante la Figura 3.48.

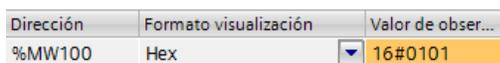


Figura 3.48: Petición Tres Parte 1, Modbus-RTU

Al realizar la petición desde el maestro se debe escoger la dirección inicial para el primer registro de datos, el valor transmitido al PLC1 será almacenado en un bloque de datos sobre una variable de tipo WORD, en la Figura 3.49 se puede observar el valor en hexadecimal del dato recibido.

"DB_LeerRegistroParada".RegistroParada	%DB5.DBWO	Hex	16#0101
--	-----------	-----	---------

Figura 3.49: Petición Tres Parte 2, Modbus-RTU

#### d. Petición 4

La última demanda realizada desde el maestro permite escribir datos de tipo Word en los registros de almacenamiento del esclavo, el valor a ser escrito será establecido en el PLC1 e intervendrá sobre el segundo registro de datos en el equipo destinatario. Mediante una estructura MOVE se procede a establecer un valor de 16#00CB (Hexadecimal) en la variable a enviar (véase Figura 3.50).

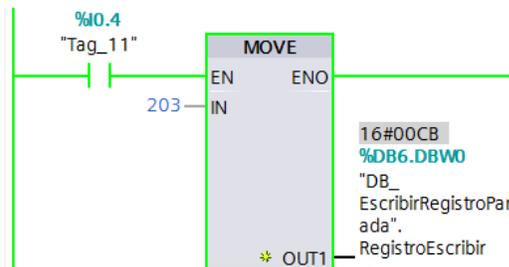


Figura 3.50: Petición Cuatro Parte 1, Modbus-RTU

En el PLC2 mediante una tabla de observación se puede constatar el valor enviado desde el maestro, como se mencionó el registro que será corrompido es el número dos y corresponderá al área de memoria MW102 (véase Figura 3.51).

%MW102	Hex	16#00CB
--------	-----	---------

Figura 3.51: Petición Cuatro Parte 2, Modbus-RTU

### **3.4 RED MODBUS-TCP/IP**

Una red de comunicación Modbus-TCP/IP puede ser montada entre equipos Simatic S7-1200 mediante el uso del puerto Profinet incluido en las CPUs, no se necesita módulos adicionales ni requerimientos de cableado especiales. En esta red se maneja el modelo cliente-servidor para transmitir peticiones hacia los esclavos y recibir las respuestas adecuadas en los maestros. A pesar de realizarse una transmisión de información sobre tramas *Ethernet* los códigos de funciones originales y prestaciones de la red se mantienen. Las características del protocolo Modbus se encuentran explicadas en la sección 1.4.

#### **3.4.1 Elementos del Sistema**

##### **3.4.1.1 Interfaz**

Los requerimientos para una red Modbus-TCP/IP son los mismos que se tiene en una red *Ethernet*, es así que la interfaz de comunicación de los controladores Simatic S7-1200 a través de su puerto LAN está en la capacidad de manejar este estándar industrial, Modbus se ha convertido en una red pre-configurada o “por defecto” gracias a sus características de simplicidad, necesidades mínimas y tecnología abierta. Como se explicó en la sección 1.4.4 la trama de Modbus es encapsulada sobre el protocolo TCP logrando altas velocidades de transmisión (Acromag, 2005).

Debido a que las comunicaciones en un modelo cliente-servidor son manejadas a través de direcciones IP y puertos de conexión, para la estructura Modbus se tiene reservado el puerto 502 a través del cual los clientes y servidores puede enviar y recibir datos.

##### **3.4.1.2 Cable y Conectores**

Los cables y conectores que son utilizados en una red Modbus-TCP son los mismos que se han establecido para las redes *Ethernet*, en entornos industriales se debe tomar las precauciones adecuadas debido a las interferencias electromagnéticas para lo cual se puede escoger entre las diferentes variantes y categorías. En la sección 3.1.1.2 se puede encontrar más información acerca de lo mencionado.

### 3.4.2 Configuración Red Modbus-TCP/IP

#### a. Red

La red Modbus-TCP/IP que se establece está formada por dos CPUs Simatic S7-1200 conectadas mediante un cable Industrial *Ethernet*, las direcciones IP deben ser asignadas a los clientes y servidores dentro del área de trabajo de manera unívoca con una topología estrella. A diferencia de una red Modbus-RTU en la que existe un solo maestro, la tecnología TCP permite la conexión multi-maestro, por lo tanto más equipos clientes o servidores pueden ser fácilmente incorporados a la red *Ethernet* estableciendo más canales de comunicación.

#### b. Configuración de las Conexiones

Las comunicaciones en este tipo de red son manejadas a través de las direcciones IP de los equipos participantes y de puertos de las CPUs, al agregar una instrucción de comunicación para un equipo servidor se genera un DB de instancia asociado el cual posee todos los parámetros necesarios para establecer la conexión, cada canal debe ocupar un DB y un número de puerto IP unívoco. Adicional a lo explicado también se debe manejar IDs de conexión únicas para cada canal.

El bloque de datos asociado a una conexión permite establecer parámetros específicos para el servidor de tal manera que este responda a dos tipos de clientes Modbus:

- Dirección IP 0.0.0.0: El servidor Modbus responde a todos los clientes que tratan de establecer conexión desde cualquier IP.
- Dirección IP TCP pasivas: Se puede configurar dos direcciones IP TCP pasivas 1 y 2, a través de estos valores es posible establecer la respuesta del servidor solamente a los clientes que tengan origen en las direcciones almacenadas en memoria.

El intercambio de información se realiza a través de puertos específicos de la CPU, el valor por defecto configurado para una comunicación Modbus es el 502, pero se puede

configurar más puertos de tal manera que se establezcan más canales. La combinación de direcciones IP, puertos y lógica de programación permiten realizar diversas comunicaciones Modbus.

Se puede establecer dos canales de comunicación para una misma dirección IP usando dos puertos distintos, de esta manera es posible realizar peticiones independientes hacia un esclavo que son procesadas de manera paralela gracias al modelo Consulta-Respuesta de la red utilizada.

Usando una misma dirección y puerto IP se pueden realizar también varias peticiones Modbus de una manera coordinada, si se desea el resultado de cada petición puede ser almacenado en diferentes áreas de memoria dependiendo su naturaleza.

### **c. Instrucciones Modbus-TCP**

En el paquete de Instrucciones de Comunicación dentro la categoría Modbus-TCP situada en la parte derecha de TIA PORTAL se puede encontrar las instrucciones necesarias para la comunicación mediante el protocolo TCP, estos controles permiten configurar, enviar y recibir datos entre dos equipos conectados con el modelo Cliente-Servidor, a continuación se describe las funciones de las estructuras necesarias para establecer una comunicación Modbus vía *Ethernet*.

**Instrucción MB\_SERVER:** Esta instrucción permite establecer una comunicación como servidor Modbus a través de una red *Ethernet* y la conexión Profinet de la CPU-S71200. Mediante MB\_SERVER se puede configurar el canal de comunicación y puerto necesario, procesar peticiones de función de clientes Modbus y enviar los correspondientes mensajes de respuesta.

**Instrucción MB\_CLIENT:** Esta instrucción es la que maneja en si las comunicaciones en la red, ya que permite a una CPU figurar como Cliente Modbus, sus funciones son las de

establecer una comunicación entre el maestro y esclavo, enviar peticiones con el código de función adecuado, procesar las respuestas del equipo subordinado y deshacer los canales creados.

### **3.4.3 Ejemplo de Conexión**

En este apartado se presenta un ejemplo de conexión de dos CPUs Simatic S7-1200 mediante la implementación del protocolo TCP en una red Modbus, para esto se hará uso de las instrucciones MB\_SERVER y MB\_CLIENT que permitirán el intercambio de peticiones-respuestas entre un dispositivo maestro y uno esclavo. En el ejemplo se demuestra utilidad de la combinación de direcciones IP y puertos para establecer más canales de comunicación, de esta manera se crea dos conexiones entre el cliente y el servidor para permitir el procesamiento simultáneo de varias peticiones.

#### **a. Configuración de *Hardware* para los PLCs**

Para establecer la red Modbus-TCP/IP se debe agregar los dos PLCs a un proyecto nuevo y configurar los parámetros de la interfaz Profinet de la siguiente manera:

- PLC1 (S7 1212C AC/DC/RLY)
  - Asignar Dirección IP: 192.168.0.1
  - Mascara de Subred: 255.255.255.0
  
- PLC2 (S7 1212C AC/DC/RLY)
  - Asignar Dirección IP: 192.168.0.2
  - Mascara de Subred: 255.255.255.0

#### **b. Programación PLC1 (Cliente)**

El PLC1 figurará como maestro en la red Modbus, por lo tanto es el equipo encargado de establecer la comunicación y enviar las peticiones que desee hacia el esclavo, los códigos de función Modbus implementados para los equipos Simatic permiten leer y escribir datos

en la CPU remota tanto en el área de Imagen de Proceso como en otras áreas relacionadas a registros.

La estructura MB\_CLIENT es la encargada de realizar y procesar cualquier tipo de petición hacia los esclavos, a continuación se explica sus parámetros más relevantes y posteriormente se adjunta las estructuras creadas para cada demanda específica.

- REQ: Mientras su valor sea “1” o TRUE la instrucción enviará peticiones de comunicación hacia el esclavo.
- DISCONNECT: El valor “1” es para deshacer la conexión y el valor “0” permite establecer una conexión.
- CONNECT\_ID: ID unívoca que permite diferenciar las conexiones.
- IP\_OCTECT\_X: Corresponde a la dirección IP del dispositivo con el cual se desea comunicar, el servidor.
- IP\_PORT: Puerto de la CPU por el cual se realizará la comunicación.
- MB\_MODE: Su valor es de 0, 1 ó 2, permite configurar la petición para lectura, escritura o diagnóstico.
- MB\_DATA-ADDR: En este parámetro se especifica la dirección inicial que se desea intervenir en el esclavo, sea para imagen de proceso o valores de registros.
- MB\_DATA\_LEN: Longitud de los datos o registros a intervenir.
- MB\_DATA\_PTR: En este parámetro se establece el puntero al registro de datos Modbus donde será colocada la información recibida del servidor, puede ser un bloque de datos global o un área de memoria de tipo M.

En el ejemplo se realiza cuatro tipos de peticiones a través de dos canales de comunicación, las solicitudes pueden ser enviadas al mismo tiempo o cuando se lo desee según la lógica de programación.

La primera petición corresponde a leer el valor *ON/OFF* de las seis salidas del esclavo, el código de función Modbus correspondiente sería el 01. La segunda solicitud permite leer el valor binario de las ocho entradas del PLC2, el código Modbus correspondiente es el

02. La tercera petición se realiza para escribir valores binarios en las salidas Q0.4 y Q0.5, en este caso se utiliza el código de función 15. El último requerimiento permite leer un valor de registro de entrada de tipo WORD del servidor mediante el código Modbus 04, en ninguno de los casos esta soportada la función *broadcast*.

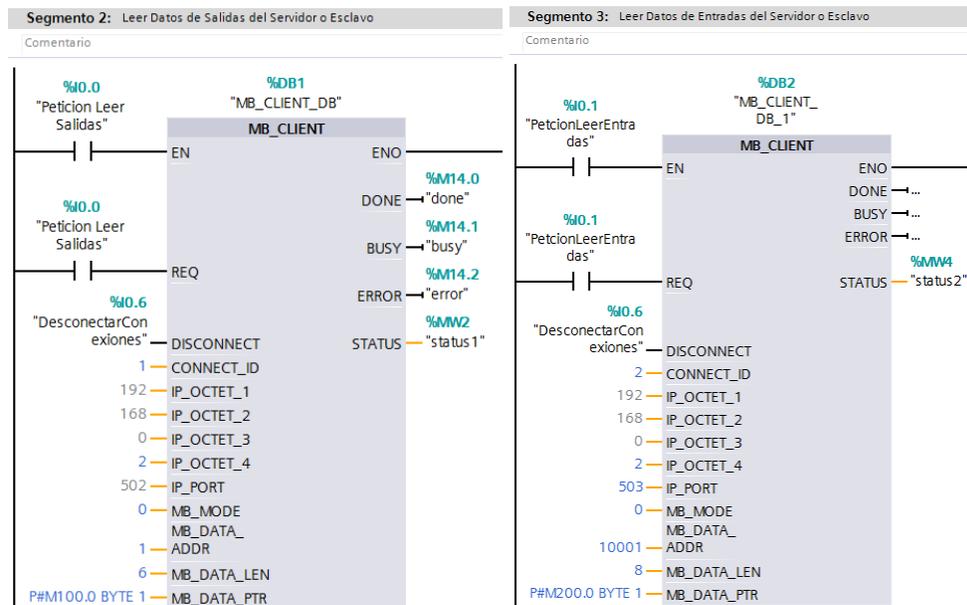


Figura 3.52: Instrucción MB\_CLIENT, leer datos de salida y entrada de un esclavo

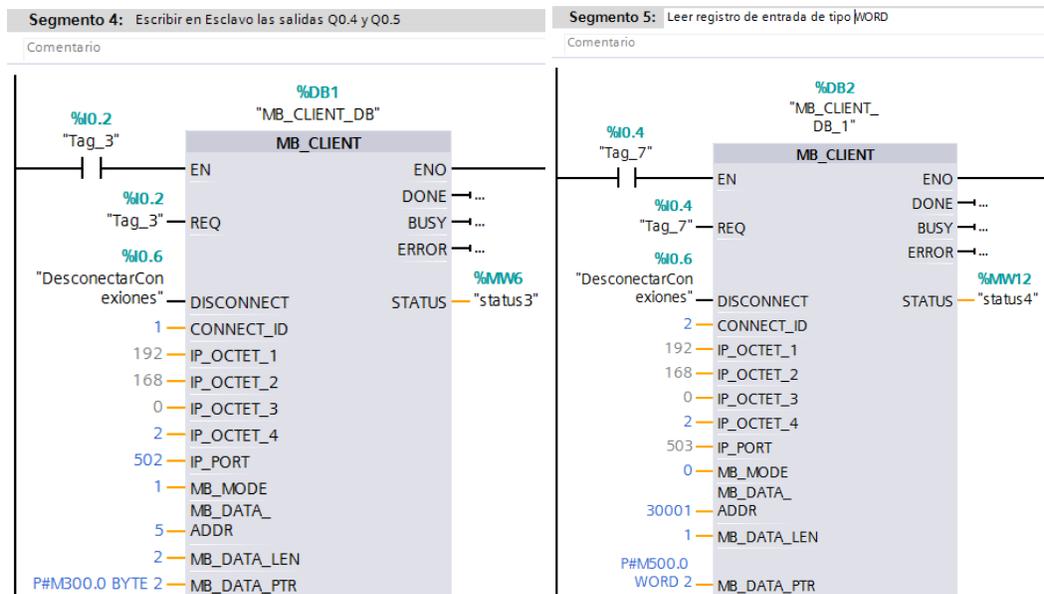


Figura 3.53: Instrucción MB\_CLIENT, escribir datos y leer registros de un esclavo

### c. Programación PLC2 (Esclavo)

El PLC2 figurará como esclavo en la red Modbus, por lo tanto es el equipo encargado de recibir la comunicación y responder a las peticiones requeridas por el cliente, la estructura MB\_SERVER deberá ser añadida al proyecto de la siguiente manera (véase Figura 3.54).

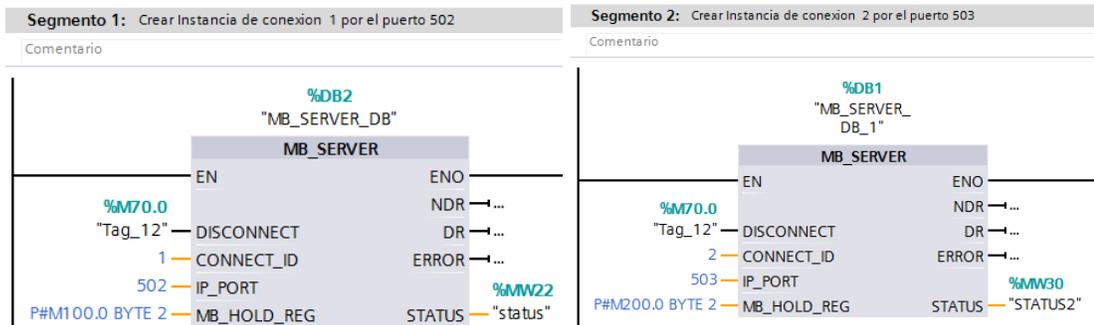


Figura 3.54: Instrucción MB\_SERVER

En el esclavo se crea dos instrucciones MB-SERVER para configurar dos canales de comunicación con el cliente a través del puerto 502 y 503. La correspondencia de los principales parámetros de la estructura es la siguiente:

- **DISCONNECT:** Este parámetro permite controlar el tráfico de peticiones hacia el esclavo. Si el valor es “1” no se ejecutan las operaciones.
- **CONNECT\_ID:** ID unívoca que permite diferenciar las conexiones.
- **MB\_HOLD\_REG:** En este parámetro se establece el puntero al registro de datos de almacenamiento Modbus donde será colocada la información interna del esclavo, puede ser un bloque de datos global o un área de memoria de tipo M.

#### 3.4.4 Pruebas de Funcionamiento

La aplicación realizada para la comunicación de dos dispositivos en una red Modbus-TCP está enfocada al uso de los códigos de función del estándar Modbus, tanto para realizar acciones de lectura y escritura del esclavo. Como se mencionó en el apartado anterior se realiza cuatro peticiones desde el cliente, las respuestas enviadas de vuelta desde el servidor son visualizadas mediante una tabla de observación creada para el PLC1 o cliente.

A continuación se presenta las figuras correspondientes en cada caso para comprobar el correcto funcionamiento de la red.

### a. Petición 1

La primera solicitud que se envía es la correspondiente a la lectura de las variables de salida del esclavo, en el programa de control del PLC2 se prende o apaga las salidas Q0.0 Q0.1 y Q0.3 mediante la entrada I0.0, los valores son enviados al cliente constantemente si la petición está en curso. En la primera parte se puede comprobar el estado de las salidas mencionadas y los datos recibidos mediante la Figura 3.55.

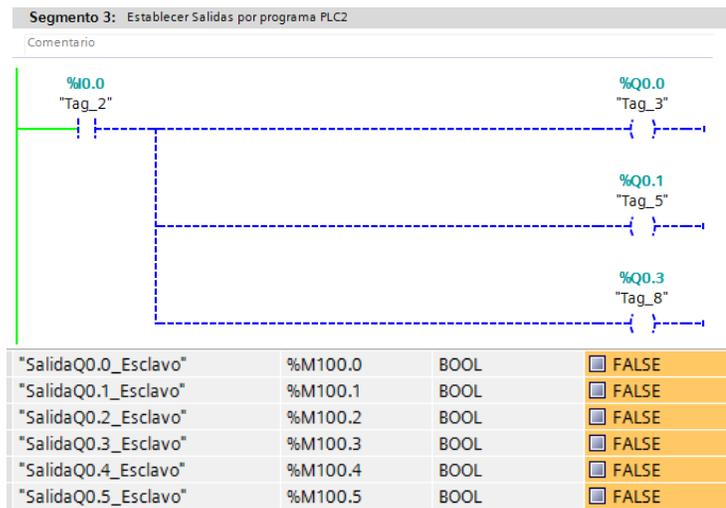


Figura 3.55: Petición Uno Parte 1, Modbus-TCP/IP

Luego de conmutar la entrada I0.0 del esclavo se accionan las salidas indicadas y los datos serán transmitidos al PLC1, los nuevos valores para dichas variables son los siguientes (véase Figura 3.56).

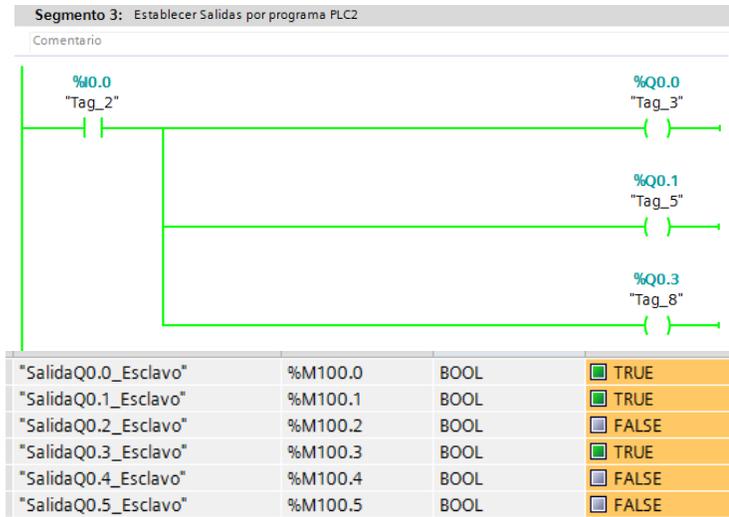


Figura 3.56: Petición Uno Parte 2, Modbus-TCP/IP

### b. Petición 2

La segunda solicitud referida a la lectura de las variables de entrada del esclavo permite obtener el valor binario de todas las entradas físicas, para comprobar el correcto funcionamiento de los datos enviados se interacciona con las entradas del PLC2 accionándolas al azar. En la Figura 3.57 se puede observar un primer estado con todos los *bits* en “0” y en la Figura 3.58 se constata el cambio de los *bits* 0,4 y 5.

"EntradaI0.0_Esclavo"	%M200.0	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.1_Esclavo"	%M200.1	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.2_Esclavo"	%M200.2	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.3_Esclavo"	%M200.3	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.4_Esclavo"	%M200.4	BOOL	<input type="checkbox"/> FALSE
"EntardaI0.5_Esclavo"	%M200.5	BOOL	<input type="checkbox"/> FALSE
"EntardaI0.6_Esclavo"	%M200.6	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.7_Esclavo"	%M200.7	BOOL	<input type="checkbox"/> FALSE

Figura 3.57: Petición Dos Parte 1, Modbus-TCP/IP

"EntradaI0.0_Esclavo"	%M200.0	BOOL	<input checked="" type="checkbox"/> TRUE
"EntradaI0.1_Esclavo"	%M200.1	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.2_Esclavo"	%M200.2	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.3_Esclavo"	%M200.3	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.4_Esclavo"	%M200.4	BOOL	<input checked="" type="checkbox"/> TRUE
"EntardaI0.5_Esclavo"	%M200.5	BOOL	<input checked="" type="checkbox"/> TRUE
"EntardaI0.6_Esclavo"	%M200.6	BOOL	<input type="checkbox"/> FALSE
"EntradaI0.7_Esclavo"	%M200.7	BOOL	<input type="checkbox"/> FALSE

Figura 3.58: Petición Dos Parte 2, Modbus-TCP/IP

**c. Petición 3**

La tercera petición se realiza para la escritura de dos bits de salida en el esclavo, desde el cliente se establece los niveles de salida para dos variables mediante la entrada I0.3, los datos binarios son enviados al esclavo y permiten controlar las salidas Q0.4 y Q0.5. En la Figura 3.59 se expone el valor de “0” enviado para ambas salidas.



Figura 3.59: Petición Tres Parte 1, Modbus-TCP/IP

En la segunda parte se procede a accionar la entrada I0.3 del PLC1 con lo cual los valores de “1” son transmitidos hacia el esclavo, siempre y cuando la petición se encuentre en proceso estos datos serán escritos directamente en la imagen de proceso de salida del PCL2. Los niveles obtenidos para estos bits se muestran en la Figura 3.60.



Figura 3.60: Petición Tres Parte 2, Modbus-TCP/IP

**d. Petición 4**

La ultima demanda realizada desde el cliente permite leer datos de entrada de tipo Word desde el esclavo, entre sus funciones el estándar Modbus permite acceder a datos de registros de tipo Word tanto en lectura como escritura, en el programa del PLC2 se escoge

dos valores a enviar mediante el accionamiento de la entrada I0.0, si dicha entrada se encuentra desactivada se envía un valor de 100 (en base decimal) hacia el maestro (véase Figura 3.61).

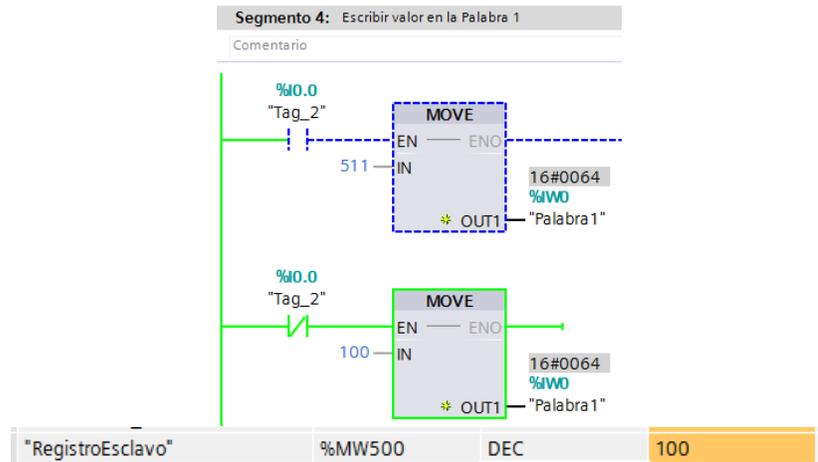


Figura 3.61: Petición Cuatro Parte 1, Modbus-TCP/IP

Por otro lado si la entrada I0.0 es accionada se establece un valor de 511 en el registro de entrada del PL1, este valor es enviado hacia el esclavo y podrá ser visualizado en el área de memoria adecuada. En la Figura 3.62 se puede observar el valor tras accionar el bit de entrada.

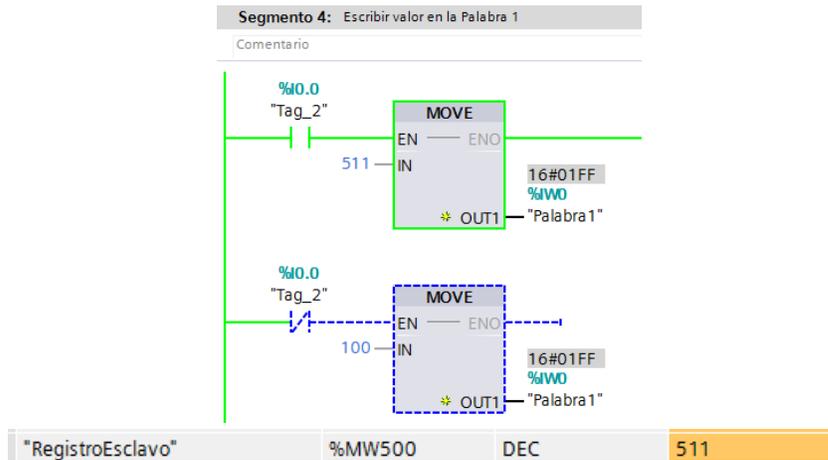


Figura 3.62: Petición Cuatro Parte 2, Modbus-TCP/IP

### 3.5 CONCLUSIONES

Luego de la culminación del presente capítulo se logró el establecimiento de cuatro redes de comunicación industrial con diferentes prestaciones y enfocadas a distintas áreas dentro de un proceso productivo. Los medios físicos, características, configuraciones de conexión son propias de cada protocolo y permiten establecer ventajas sobre las demás redes en procesos específicos de automatización, de acuerdo a los conocimientos adquiridos se puede destacar lo siguiente:

- El puerto Profinet de una CPU S7-1200 permite conectar el equipo a redes LAN de infraestructura *Ethernet*, y comunicarse con otros dispositivos bajos protocolos como TCP, ISO sobre TCP y UDP.
- Los cables de *Industrial Ethernet* son el medio adecuado para la conexión de redes Profinet y Modbus-TCP, ya que poseen características adecuadas para el entorno industrial a velocidades de transferencia elevadas de 100 Mbits/s.
- Mediante una red Profinet se puede enlazar dispositivos a través de su dirección IP y gestionar diferentes canales de comunicación sobre el mismo medio convenientemente identificados por una ID unívoca, de esta manera se utiliza la estructura Consulta-Respuesta de *Ethernet* para establecer varias comunicaciones paralelas entre dos equipos.
- La instrucción TSEND\_C optimiza la transmisión de información en una red Profinet para el envío de estructuras de datos *Variant*, con lo cual se puede intercambiar diferentes datos de todo tipo sin problemas de coherencia en el destinatario.
- Para establecer una red Profibus entre equipos Simatic S7-1200 es necesario el uso de dos módulos de comunicación: CM 1243-5 (maestro) y CM 1242-5 (esclavo), el perfil que se maneja es Profibus DP.
- Las redes de datos que utilicen la interfaz eléctrica RS-485 como medio de transmisión son adecuadas para protocolos Profibus y Modbus-RTU, en ambos

casos se utiliza dos hilos del conductor más uno de aterramiento, se debe tomar en cuenta aspectos como el apantallamiento del conductor y carcasas metálicas con el fin de disminuir las interferencias en el bus.

- Las redes Profibus deben utilizar resistencias terminadoras en los extremos del bus para evitar las reflexiones, existen conectores *FastConnect* adecuados para este fin.
- En una red Profibus se utilizan áreas de transferencia entre el maestro y el esclavo con el fin de intercambiar datos de manera eficaz, se pueden enviar una gran cantidad de variables de tipo BYTE o WORD que serán escritas directamente en la imagen de proceso de ambos dispositivos.
- El modelo maestro-esclavo de una red Profibus permite a una gran cantidad de dispositivos esclavos (127 por maestro) conectarse al bus, e interactuar con el controlador central de manera sencilla mediante las direcciones de memoria enlazadas.
- Para establecer una red Modbus-RTU entre equipos Simatic S7-1200 es necesario el uso de dos módulos de comunicación CM 1241 (transmisión PtP) en ambos dispositivos, estos equipos pueden usarse como maestros o esclavos de la red.
- Modbus-RTU es una red que permite la conexión de 247 esclavos regidos mediante un equipo maestro, permite acciones de lectura, escritura y diagnóstico sobre los elementos participantes.
- Una red Modbus-TCP/IP no necesita módulos adicionales para su establecimiento ya que utiliza la interfaz Profinet de la CPU, el acceso y direccionamiento se realiza a través de direcciones IP y puertos de las CPUs.
- En una red Modbus-TCP/IP se puede establecer distintos canales de comunicación por puertos diferentes con el fin de procesar peticiones-respuestas de manera paralela.

- Una red Modbus-TCP/IP tiene la posibilidad de usar un modelo multi-maestro para sus comunicaciones, a diferencia de una red Modbus-RTU.
- Ambos protocolos RTU y TCP/IP permiten la transmisión de la trama propia de Modbus, con lo cual se mantienen los códigos de función del estándar mediante peticiones que tienen efecto directo sobre áreas de salida, entrada, registros de entrada y registros de almacenamiento en los esclavos.

## CAPÍTULO 4

### SISTEMA SCADA

#### Introducción

En el presente capítulo se realiza un resumen acerca de los Sistemas SCADA, los cuales forman parte importante de los procesos de automatización ya que actúan como interfaces de comunicación, adquisición de datos y mando, entre el personal de la fábrica y el proceso controlado. Se aborda temas como: las partes de un Sistema SCADA, sus objetivos, prestaciones, elementos de *hardware*, seguridad, aspectos generales de dichas aplicaciones independientemente del fabricante. En un segundo apartado se desarrolla un breve estudio del *software* WinCC creado por la empresa Siemens, del cual se explica temas como: sus funcionalidades básicas en cuanto variables, avisos, ficheros, recetas, etc.; también se resume las diferentes posibilidades de comunicación, sus opciones o complementos para aplicaciones más complejas y sus interfaces para integración con sistemas PC.

#### 4.1 SISTEMA SCADA (*SUPERVISORY CONTROL AND DATA ACQUISITION*, “SUPERVISIÓN CONTROL Y ADQUISICIÓN DE DATOS”)

##### 4.1.1 Generalidades

Un sistema SCADA es cualquier *software* que permita el acceso a datos remotos de un proceso y su respectivo control utilizando ciertas herramientas de comunicación necesarias. No se trata de un sistema de control sino de una utilidad del *software* para monitorear o supervisar las tareas, sirviendo de interface entre niveles de control como PLCs y niveles de gestión (Corrales, 2007).

El sistema SCADA representa el medio entre el usuario y su máquina. Las funciones de mando y valores procesados son visualizados en paneles de operador táctiles, con la ayuda

de estos visualizadores se pueden representar fácilmente el estado de los procesos, mensajes de alarma, recetas y demás bondades (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

Además un sistema SCADA se encarga de proveer toda la información necesaria a diversos usuarios dentro de todas las capas de automatización, está diseñado para su aplicación generalmente sobre estaciones basadas en ordenadores (Gomez, 2009). Según lo presentado por (Corrales, 2007), el esquema principal de un sistema SCADA está compuesto por los siguientes bloques:

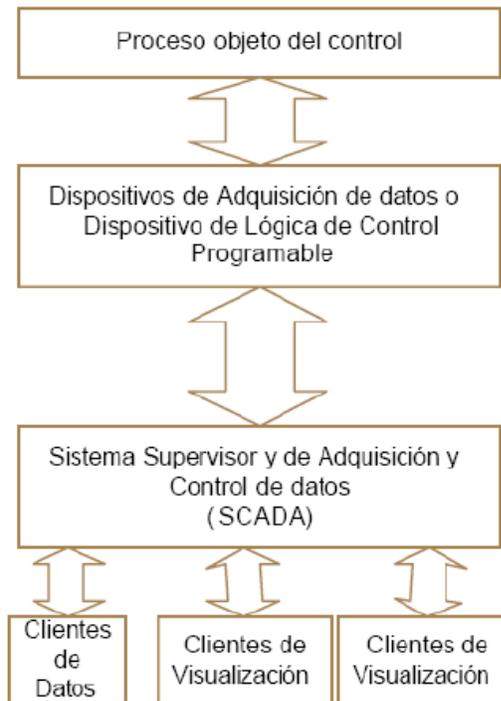


Figura 4.1: Esquema de un Sistema SCADA

Fuente: (Corrales, 2007)

- a. **Proceso/Objeto del Control:** Es el proceso u objeto de planta que se desea controlar y supervisar, por lo tanto es el proveedor de datos hacia los sistemas superiores.

- b. Adquisición de Datos:** Se refiere a un grupo de instrumentos de medición de los parámetros físicos y eléctricos del sistema, dotados de alguna interfaz de comunicación que permita su conexión con el sistema de supervisión.
- c. SCADA:** Sistema combinado de *hardware* y *software* en el cual se tiene varias funciones de visualización y mando de los datos recolectados.
- d. Clientes:** Hace referencia a un conjunto de aplicaciones de diferentes tipos que hacen uso de los datos obtenidos en el Sistema SCADA.

#### 4.1.2 Prestaciones

Dentro de las aplicaciones SCADA se pretende optimizar los sistemas productivos mediante herramientas de interfaz hombre-máquina que permitan el uso eficaz de diversas funciones y utilidades, con el fin de comunicar el proceso con el operador de la manera más clara posible. Según (Rodríguez, Sistemas SCADA 3ra Edición, 2012) y (Gomez, 2009) algunas de las prestaciones necesarias son:

- **Monitorización:** Vigilancia de los datos del proceso en tiempo real para estaciones cercanas y lejanas.
- **Supervisión:** Dotar de herramientas de gestión para la toma de decisiones relacionadas con el mantenimiento y control, con el fin de evitar supervisión humana continua.
- **Adquisición de datos:** Permitir el acceso a las variables de interés del proceso.
- **Visualización:** Presentación de la información de manera clara y amigable.
- **Mando:** Permitir a los operadores o usuarios de gestión cambiar datos de planta como consignas u otros valores de interés.
- **Grabación de Recetas:** Permitir cambiar un grupo de variables de planta que sean constantes para ciertos procesos.

- **Seguridad de los datos:** Medios de protección para el intercambio de datos ante influencias externas.
- **Seguridad en el acceso:** Permitir clasificar los tipos de acceso con el fin de restringir acciones para usuarios no autorizados.
- **Paneles de Alarma:** Reconocimiento de eventos excepcionales y registro de los mismos.
- **Generación de Históricos:** Permitir almacenar valores históricos de señales del proceso para su posterior análisis o presentación.
- **Posibilidad de Programación Numérica:** Permitir realizar cálculos aritméticos de alto nivel en el ordenador.

### 4.1.3 El Hardware

#### 4.1.3.1 Interfaz Hombre-Máquina

Una Interfaz Hombre-Máquina o HMI es un mecanismo combinado de *hardware* y *software* que le permite a un operador interactuar con un proceso productivo mediante pantallas y funciones específicas, además comprende sinópticos de control y sistemas de presentación gráfica (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

Un HMI tiene la capacidad de visualizar y determinar datos de los accionamientos como estados (*ON/OFF*) o magnitudes de las variables físicas involucradas; puede ser desde un simple interruptor e indicador luminoso para encender algún dispositivo, hasta varias pantallas que esquematizan todo un proceso de producción incluyendo valores en tiempo real y acciones de supervisión (Corrales, 2007) (véase Figura 4.2).

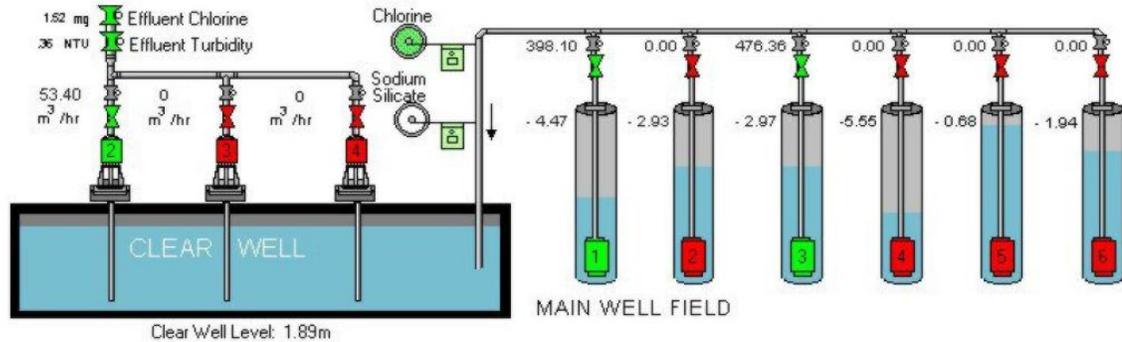


Figura 4.2: Pantalla HMI de un Sistema SCADA

Fuente: (Corrales, 2007)

Existen varias herramientas de *software* que permiten el manejo de un sistema SCADA a través de las cuales se desarrolla e implementa varias pantallas HMI, estas actúan como interfaz entre el proceso y el operador permitiendo acceder a consignas o configuraciones del mismo, además presenta herramientas para la comunicación con sistemas superiores de la pirámide CIM a través de servidores de datos.

#### 4.1.3.2 Unidad Central MTU (*Master Terminal Unit*)

La Unidad Central (MTU) es la estación principal la cual centraliza el mando del sistema, debe hacer uso de protocolos abiertos que permitan la integración de múltiples plataformas del sistema. Esta estación está ubicada en los Centros de Control permitiendo recopilar y archivar datos históricos, la información es almacenada para dar lugar a diversas acciones que puedan requerir usuarios de gestión (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

La MTU se encarga básicamente de las siguientes acciones:

- Gestión de las comunicaciones.
- Recopilar datos de las estaciones remotas.
- Envío de información.
- Análisis.
- Visualización de datos.
- Mando.

- Seguridad.
- Almacenar datos y archivos.
- Administración.

#### **4.1.3.3 Unidad Remota RTU (*Remote Terminal Unit*)**

La Unidad Remota (RTU) es un dispositivo dedicado a acciones de control y supervisión de áreas de un proceso productivo, están desplegados en el campo y se comunican con la estación central a través de un canal de comunicación (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

Las RTUs son elementos de gran importancia en un Sistemas SCADA que se encargan de recopilar datos, codificarlos y transmitirlos hacia la MTU o hacia otra RTU, además recibe órdenes desde la MTU, decodifica la información y ejecuta acciones adecuadas (Corrales, 2007).

Según (Rodríguez, Sistemas SCADA 3ra Edición, 2012) existen tres tipos de elementos que pueden ser considerados como unidades remotas:

- **RTU (*Remote Terminal Unit*, “Unidad Terminal Remota”)**: Son equipos especializados en comunicación que se encargan de recopilar los datos de controladores de campo y transmitirlos hacia la MTU, son elementos robustos.
- **PLC (*Programmable Logic Controller*, “Controlador Lógico Programable”)**: Controladores concebidos en sus inicios como dispositivos dedicados al control de instalaciones, pero con el tiempo se han incluido módulos con prestaciones de comunicación que permiten intercambiar información con la MTU.
- **IED (*Intelligent Electronic Device*, “Dispositivo Electrónico Inteligente”)**: Se trata de periféricos “inteligentes” que se ocupan de tareas específicas de control y

regulación, pueden ser dispositivos como variadores de frecuencia, transductores, etc.

#### **4.1.4 Seguridad**

Los Sistemas SCADA están aplicados en todos los ámbitos de control industrial y se encuentran ampliamente utilizados tanto para estaciones cercanas al proceso, como para estaciones alejadas del mismo. A partir de esto se ha establecido canales de comunicación para los cuales los conceptos de seguridad forman un aspecto vital, ya que se trata de información privada que permite controlar y supervisar procesos en tiempo real, un ataque o fallo de dicho canal podría desencadenar en serios problemas para la aplicación.

La integración de los Sistemas SCADA con los sistemas IT (Tecnologías de la Información) permite a los conglomerados conectar las redes corporativas con las de proceso, pero establece una serie de requisitos de seguridad que permiten mantener el enlace fuera de peligros informáticos (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

La integración de los dos sistemas supone una tarea difícil por el hecho de las diferencias de las redes como: tiempos de funcionamiento, requerimientos de seguridad, de mantenimiento, de funcionamiento o actualización de datos. Estas características conllevan acciones de seguridad que permitan a diferentes tipos de usuarios tener acceso al sistema SCADA desde cualquier punto de la red (Corrales, 2007).

##### **4.1.4.1 Amenazas**

Los Sistemas SCADA son establecidos sobre ordenadores de la plata y debido a su integración con las redes informáticas los peligros son similares a los conocidos para el mundo informático, a continuación se presentan algunos de ellos:

- Espionaje.
- Sabotaje.
- Intrusión o *hackers*.

- Robo electrónico.
- Troyanos.
- DoS (*Denial of Service*<sup>46</sup>, “Negación de Servicio”).
- Virus.
- *Malware*<sup>47</sup>.

#### 4.1.4.2 Técnicas

Los elementos asociados a tareas de control y supervisión deberán estar agrupados en una red de proceso, la misma que deberá estar aislada de las demás redes empresariales con el fin de mantener el tráfico de información separado para cada propósito. Los puntos de acceso entre ambas infraestructuras se convierten en las áreas de interés, donde se establecerán las medidas adecuadas de seguridad. De acuerdo a lo explicado por (Rodríguez, Sistemas SCADA 3ra Edición, 2012) se puede destacar algunas de las técnicas usadas:

- **Cortafuegos:** Son barreras lógicas usadas para aislar redes informáticas, por ejemplo una red LAN y otra pública como *Internet*, o también para separar redes de jerarquía diferentes (*intranets*). Existen diferentes tipos de cortafuegos dependiendo las necesidades que permiten establecer diversos perímetros de seguridad.
- **Proxy:** Sistemas que protegen a los dispositivos de la red interna de las amenazas de las redes externas o públicas, son equipos que prestan servicios de *internet* de forma transparente para los ordenadores de la red privada.
- **Zona de Nadie:** Se la conoce también como Zona Delimitadora (DMZ), en la cual se colocan los servidores que serán visibles desde *internet* y se establecen las medidas de seguridad para evitar accesos a la *intranet*.

---

<sup>46</sup> DoS: Ataque de negación de servicios, se trata de saturar la red para que los usuarios legítimos no puedan acceder a los recursos.

<sup>47</sup> *Malware*: Tipo de *software* que busca infiltrarse o dañar un sistema de información.

- **VPN (*Virtual Private Networks*, “Red Privada Virtual”)**: Son redes privadas seguras que se establecen para la comunicación de dispositivos de la *intranet* con elementos de la *extranet*, pueden estar distribuidas geográficamente pero actúan como una red local, se consigue a través de la creación de enlaces seguros en los cortafuegos de los servidores.
- **IDS (*Intrusion Detection System*, “Sistema de Detección de Intrusos”)**: Son diversos medios de detección contra accesos no autorizados como vigilancia de tráfico, de conexiones, etc.

## **4.2 SISTEMA SCADA WINCC (SIEMENS)**

La empresa SIEMENS ha desarrollado un *software* de automatización dedicado al diseño e implementación de procesos SCADA, esta plataforma conocida como WinCC se puede encontrar integrada en TIA PORTAL o por separado, y contiene todas las herramientas necesarias para la construcción de HMIs basados en pantallas táctiles o estaciones PC. Este *software* reúne varias de las funcionalidades de los Sistemas SCADA en cuanto a su comunicación transparente, integración con IT, programación amigable, servidores web, seguridad, etc.

### **4.2.1 Características de WINCC**

#### **4.2.1.1 Funcionalidades Básicas**

En WinCC se tiene una serie de funcionalidades que permiten al sistema SCADA aprovechar de manera eficiente la arquitectura de los ordenadores y controladores, con el fin de realizar configuraciones de manera sencilla, según lo descrito por (SIEMENS, WinCC Profesional V13 SP1, 2014) se puede destacar los siguientes:

- **Trabajar con Variables:** Los sistemas de automatización controlados por la aplicación SCADA comunican sus datos almacenados en memoria a través de variables externas, estas variables constituyen el enlace entre WinCC y los controladores, se pueden efectuar acciones de lectura o escritura sobre las mismas.

Existen también variables internas que sirven para transportar valores dentro de WinCC sin conexión a procesos de control.

- **Trabajar con Avisos:** El sistema de avisos brinda la posibilidad de visualizar en los paneles gráficos los estados operativos o averías que se han producido en un proceso, existen dos tipos de avisos: los definidos por el sistema y los definidos por el usuario.

N°	Hora	Fecha	Texto de aviso	Estado	Categoría
5	12:50:24 :590	24.02. 2007	La presión de la caldera supera el límite superior.	Aparecido Desaparecido	Advertencia: Color rojo

Figura 4.3: Avisos en WinCC

Fuente: (SIEMENS, WinCC Profesional V13 SP1, 2014)

- **Trabajar con Recetas:** Las recetas son paquetes de datos que especifican determinadas configuraciones para valores de proceso, mediante ellas se puede cambiar todo un grupo de parámetros para los accionamientos.
- **Trabajar con Ficheros:** Los ficheros son estructuras que sirven para almacenar datos de proceso para HMI *Runtime*, existen ficheros de variables y de avisos para archivar el comportamiento del sistema. La forma de llenado de los ficheros corresponde a un método cíclico o de segmentación en la cual se sobrescriben los valores más antiguos. También puede programarse acciones de alarma para lanzar avisos o acciones al completar cierto número de datos.
- **Trabajar con Informes:** Los informes son utilizados para documentar y presentar datos del sistema y ciclos de producción concluidos, se puede programar tiempos o acciones para la elaboración e impresión de los mismos. Se puede definir el formato del informe en cuanto a encabezados, títulos, márgenes, etc. y escoger que tipos de datos serán incluidos, es posible añadir al contenido: datos de aviso,

gráficos, tablas, archivos en formato \*.csv<sup>48</sup>, hora, visor de curvas, campos de entrada o salida, visor de recetas, etc.

- **Configuración de Usuarios:** WinCC mediante su pantalla de configuración de usuarios permite controlar el acceso a datos y funciones en *Runtime* no autorizados. Cada grupo de usuarios cuenta con privilegios diferentes dependiendo su propósito.
- **Trabajar con *Scripts***<sup>49</sup>: Los scripts son estructuras definidas por el usuario que se pueden programar en lenguaje C con el fin de crear o potenciar funciones del sistema WinCC.
- **Planificador de Tareas:** El planificador de tareas permite establecer acciones que se ejecutarán en segundo plano, se pueden iniciar a través de funciones del sistema o *scripts* programados para el efecto.

#### 4.2.1.2 Comunicaciones

Las comunicaciones de los paneles HMI con los sistemas de control va a depender del modelo de los mismos, existen diferentes tipos de redes y arquitecturas que son soportadas por los Sistemas SCADA para comunicaciones con PLCs o entre paneles, a continuación según lo presentado por (SIEMENS, WinCC Profesional V13 SP1, 2014) se enlista las posibilidades de conexión:

- **Mediante Red Profinet:** Estándar abierto para la comunicación industrial basado en *Industrial Ethernet*, con capacidad de integración con IT.
- **Mediante Red Profibus:** Permite la conexión de controladores con paneles HMIs o entre paneles.

---

<sup>48</sup> Formato \*.csv: Son archivos en formato abierto para presentar datos en forma de tabla.

<sup>49</sup> *Script*: Es un programa informático simple que se almacena en archivo de texto plano.

- **Mediante Red MPI (*Multi-Point Interface*, “Interfaz de Múltiples Punto”):** Es una interfaz de red creada por SIEMENS para la conexión de productos Simatic.
- **Mediante Red PPI (*Point-to-Point Interface*, “Interfaz Punto a Punto”):** Una red creada para la comunicación de equipos Simatic S7-200, pero posibilita también la conexión con paneles de operador o PLCs Simatic S7-300 y S7-400.
- **Vía SIMATIC HMI HTTP<sup>50</sup>:** Esta comunicación utiliza el protocolo Simatic HMI HTTP para la conexión entre dos paneles operadores, el intercambio de datos se realiza mediante el modelo Consulta-Respuesta entre clientes y servidores HTTP.
- **Comunicación OPC:** El panel HMI tiene la posibilidad de figurar como cliente de un servidor OPC y obtener datos remotos. Existen también servidores OPC incluidos dentro de *WinCC Runtime*.
- **Redundancia de Medios:** Se utiliza el procedimiento MRP (*Media Redundancy Protocol*, “Protocolo de Redundancia del Medio”) con una interfaz Profinet para la conexión redundante de equipos con el fin de aumentar la disponibilidad de la red. Los dispositivos se convierten en clientes del administrador de redundancia.
- **Comunicación con otros controladores:** Existen *drivers* especiales para la comunicación con equipos ajenos a la familia Simatic como: Allen-Bradley EtherNet/IP, Mitsubishi MC TCP/IP, Modbus TCP/IP y Omron *Host Link*.

#### 4.2.1.3 Opciones

WinCC es un *software* que ofrece servicios más complejos enfocados al acceso remoto, documentación, análisis de datos mediante programas Office, aplicaciones web, etc. Estas opciones permiten desarrollar complejas aplicaciones con una completa integración desde los estratos empresariales de los conglomerados, de acuerdo a lo expuesto por (SIEMENS, WinCC Profesional V13 SP1, 2014) se puede listar las siguientes:

---

<sup>50</sup> HTTP (*Hypertext Transfer Protocol*, “Protocolo de Transferencia de HiperTexto”)

- **WinCC Audit:** Este complemento de WinCC permite realizar una documentación de datos de proceso con una normativa establecida para asegurar la fiabilidad de la información, permite realizar análisis retrospectivo de acciones, usuarios y tiempos.
- **WinCC Data Monitor:** Posee una serie de funciones provistas desde el servidor a sus clientes como: visualización de pantallas, visualizan de tablas de Excel para evaluación e impresión, generación de informes estadísticos, portal Web para acceso a la información, análisis de tendencias y alarmas.
- **WinCC Sm@rt Server:** Hace referencia a funciones de comunicación entre sistemas HMIs mediante conexiones TCP/IP como: estaciones de control distribuido, estaciones con acceso total mediante Simatic HTTP, control remoto de HMIs a través de *internet*, *intranet* y LAN, envió de correos electrónicos, facilitación de páginas HTML estándar.
- **WinCC Client-Server:** Es un complemento que posibilita la gestión de datos de un proceso mediante un modelo Cliente-Servidor, permite distribuir el manejo y visualización de una aplicación de forma eficiente. Todos los datos del proyecto se almacenan en bases de datos en el servidor, el cual provee de información a los clientes de la red.
- **WinCC Web Navigator:** Es un complemento que permite visualizar y manejar los proyectos WinCC a través de *internet* e *intranet*, implementa los métodos de seguridad de *internet*.

#### 4.2.1.4 Interfaces

Además de las opciones presentadas en el apartado anterior, WinCC ofrece algunas interfaces para la comunicación con los sistemas informáticos de las plantas a través del concepto de objetos para intercambiar información, según lo resumido por (SIEMENS, WinCC Profesional V13 SP1, 2014) se puede destacar:

- **OPC (*OLE<sup>51</sup> for Proccess Control*, “OLE para Procesos de Control”)**: Permite la comunicación entre equipos independientemente de la tecnología utilizada, es un estándar basado en Windows. En WinCC se puede configurar a un panel como servidor o cliente OPC para acceder a datos de diferentes aplicaciones.
- **API<sup>52</sup> Runtime “API de Tiempo de Ejecución”**: Son las funciones de programación abierta para WinCC, permiten utilizar las estructuras de WinCC en aplicaciones externas propias con el fin de acceder a datos de variables o ficheros.
- **OLE DB Provider (Proveedor de OLE Base de Datos)**: Es un servicio que permite el acceso a variables de proceso o de fichero almacenadas en la base de Datos MS SQL del servidor.

#### 4.2.2 SCADA ejemplo mediante una pantalla HMI KTP600

En este apartado se presenta un ejemplo para la programación/configuración del *software* WINCC mediante TIA PORTAL para su vinculación con un PLC S7-1200. El ejemplo incluye datos relevantes acerca de las funciones básicas de un sistema SCADA como son: el uso de avisos, administración de usuarios, permisos de operación y uso de recetas en *runtime*.

##### 4.2.2.1 Administración de Usuarios

La administración de usuarios para las aplicaciones SCADA en WINCC se realiza mediante la ventana de configuración pertinente dentro de la cual se debe distinguir dos categorías:

- **Grupos de Usuarios**: Son grupos generales de usuarios a los cuales se les otorga ciertas configuraciones o privilegios de acceso dentro de la aplicación. Los

---

<sup>51</sup> OLE (*Object Linking and Embedding*, “Objeto Enlazado y Embebido”)

<sup>52</sup> API (*Application Programming Interface*, “Interfaz de Programación de Aplicaciones”)

permisos que se pueden otorgar en una KTP600 son tres: administración, monitorización y operación (véase Figura 4.4).

Grupos					
	Nombre	Número	Nombre de visualización	Caducidad de l...	Comentario
	Grupo de administradores	1	Grupo de administradores	<input type="checkbox"/>	El grupo 'Administradores' ti...
	Usuarios	2	Usuarios	<input type="checkbox"/>	El grupo 'Usuarios' tiene inici...
	Inspectores	3	Inspectores	<input type="checkbox"/>	Solo permisos de Monitoriza...
	<Agregar>				

Permisos					
	Activo	Nombre	Nombre de visualización	Número	Comentario
	<input checked="" type="checkbox"/>	Administración de usuari...	Administración de usuarios	1	Permiso 'Administración de ...
	<input checked="" type="checkbox"/>	Monitorización	Monitorización	2	Permiso 'Supervisar'.
	<input checked="" type="checkbox"/>	Operación	Operación	3	Permiso 'Operación'.

Figura 4.4: Grupos de Usuarios

- **Usuarios:** Es propiamente la identificación para las personas que usan la aplicación, cada usuario debe pertenecer a un grupo específico y se debe configurar su nombre, contraseña, cierre de sesión automático o intervalo de tiempo (véase Figura 4.5).

Nombre	Contraseña	Cierre de sesión autom...	Tiempo de cierre de sesi
Administrador	*****	<input type="checkbox"/>	5
operador1	*****	<input type="checkbox"/>	5
inspector1	*****	<input checked="" type="checkbox"/>	5
<Agregar>			

Grupos			
Miembro de	Nombre	Número	Nombre de visualización
<input type="radio"/>	Grupo de administradores	1	Grupo de administradores
<input checked="" type="radio"/>	Usuarios	2	Usuarios
<input type="radio"/>	Inspectores	3	Inspectores

Figura 4.5: Usuarios

En este caso se ha creado dos usuarios con los siguientes datos:

Tabla 4.1: Datos de los usuarios creados

Usuario	Campo	Identificación
Usuario 1	Nombre	operador1
	Contraseña	operar1
	Cierre automático	No

	Grupo	Usuarios
	Permiso	Operación
Usuario 2	Nombre	inspector1
	Contraseña	inspe1
	Cierre automático	Si (tiempo 5 min)
	Grupo	Inspectores
	Permiso	Monitorización

Para la gestión de usuarios en *runtime* se utiliza una herramienta “Visor de usuarios” que permite iniciar sesión mediante un cuadro de dialogo, añadir o cambiar usuarios (permiso de Administración) y verificar con que usuario se encuentra en ese instante la aplicación.

Para controlar los permisos de acción de los usuarios sobre la aplicación SCADA se deberá especificar en cada elemento de acción (botones, visor de recetas, entradas de texto, etc) que nivel de seguridad requiere. Para esto en la pestaña “Seguridad” de cada elemento se escoge el tipo de permiso (véase Figura 4.6).

Nombre	Permiso
Ninguno	
Administración de usuarios	Administración de usu...
Monitorización	Monitorización
Operación	Operación

Figura 4.6: Seguridad de operación sobre objetos de acción

#### 4.2.2.2 Uso de Avisos

El uso de avisos mediante WINCC permite establecer condiciones límites para valores analógicos y digitales con el fin de generar avisos al operador que permitan reconocer datos erróneos de proceso.

Para el ejemplo se utiliza dos tipos de avisos digitales y analógicos que permitan reconocer: en el primer caso la activación de 4 salidas del PLC y en el segundo caso el sobre voltaje de la entrada analógica con una referencia a 5V.

- **Avisos Digitales:** Los avisos digitales o de bit se activarán como respuesta a un flanco ascendente de cierto bit en una variable cualquiera, en este caso se creó una variable de tipo *Int* de nombre “estado”. Para la activación de los avisos se utiliza los bits del 0-3 correspondientes a las salidas 1-4 del PLC. Se deberá agregar un “ID” y “Texto de aviso” que será guardado y mostrado en el visor de avisos en *runtime* conjuntamente con la hora de registro del evento.

Avisos de bit					
ID	Texto de aviso	Categoría	Variable de di..	Bit de disparo	
1	Salida 1 Activada	Errors	estado	0	
2	Salida 2 Activada	Errors	estado	1	
3	Salida 3 Activada	Errors	estado	2	
4	Salida 4 Activada	Errors	estado	3	

Figura 4.7: Avisos digitales o de bit

- **Avisos Analógicos:** Los avisos analógicos permiten establecer límites superiores e inferiores de comparación para variables analógicas y presentar dichos eventos al usuario en la pantalla.

Avisos analógicos						
ID	Texto de aviso	Categoría	Variable de di..	Valor límite	Modo del lími..	
1	Variable Analógica supera los 5V	Errors	analógica1E...	5	Superior	

Figura 4.8: Avisos analógicos

Además existen grupos de avisos que permiten básicamente dinamizar la representación de los mismos, en este caso ambos pertenecen al grupo de avisos “Errors”. Los eventos en pantalla son visualizados mediante la herramienta “Visor de avisos” en la cual se puede configurar el tipo y que información visualizar (véase Figura 4.9).

Categoría de aviso	Activar	Columnas visibles	
Errors	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Número	
Warnings	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Hora	
System	<input checked="" type="checkbox"/>	<input type="checkbox"/> Estado de aviso	
Acknowledgement	<input type="checkbox"/>	<input checked="" type="checkbox"/> Texto de aviso	
No Acknowledgement	<input type="checkbox"/>	<input checked="" type="checkbox"/> Fecha	
		<input checked="" type="checkbox"/> Nombre de la categoría	
		<input type="checkbox"/> Grupo de acuse	
		<input type="checkbox"/> Diagnosticable	
		<input type="checkbox"/> Dispositivo	

Figura 4.9: Configuraciones, visor de avisos

### 4.2.2.3 Uso de Recetas

El uso de recetas mediante WINCC permite configurar diferentes juegos de valores de proceso para ser asignados o leídos en grupo, es decir, una receta contendrá varios registros que a su vez contendrán diferentes variables y valores que serán escritos o leídos hacia el controlador, de esta manera se puede cambiar rápidamente las condiciones de proceso dependiendo la aplicación.

En el ejemplo se utiliza diferentes recetas para cambiar el número de salida a ser activada y el tiempo de encendido de la misma entre un valor lento o rápido. La configuración de se realiza mediante la ventana “Recetas” de la pantalla HMI donde se debe configurar lo siguiente:

- **Receta:** Corresponde al nombre de la receta, texto de visualización, número ID y número máximo de registros que podrá contener, en este caso se utilizara solamente registros (véase Figura 4.10).

Recetas					
Nombre	Nombre de vi...	Número	Versión	Tipo	Número máximo de registro
Receta_1	Salida1	1	12/03/...	Limitado	20
Receta_2	Salida2	2	12/03/...	Limitado	20
Receta_3	Salida3	3	12/03/...	Limitado	20
Receta_4	Salida4	4	12/03/...	Lim...	20

Figura 4.10: Configuración general recetas

- **Elementos:** Corresponden a las variables que serán contenidas en los registros y recetas, es decir, los valores que serán intervenidos en el controlador. Se debe configurar el nombre del elemento, texto de visualización y la variable del PLC (véase Figura 4.11).

Elementos		Registros	
Nombre	Nombre de visualizac..	Variable	
Salida	Numero de Salida	NumeroSalida	
Tiempo	Tiempo	tiempo	

Figura 4.11: Configuración elementos de receta

- **Registros:** Los registros son arreglos de los elementos creados que contiene diferentes valores para cada variable, en este caso se agrega dos registros de nombre “lento” y rápido que contienen la información deseada (véase Figura 4.12).

Elementos		Registros			
	Nombre	Nombre de ...	Número	Salida	Tiempo
	Registro de receta_1	Rapido	1	4	700
	Registro de receta_2	Lento	2	4	3000

Figura 4.12: Configuración registros de receta

A continuación se presenta una tabla resumen de las recetas, registros y variables añadidas al proyecto.

Tabla 4.2: Datos de recetas creadas

Receta	Registro	Variable	Valor
Salida 1	Rápido	Numero de Salida	1
		Tiempo	500 ms
	Lento	Numero de Salida	1
		Tiempo	4000 ms
Salida 2	Rápido	Numero de Salida	2
		Tiempo	600 ms
	Lento	Numero de Salida	2
		Tiempo	2000 ms
Salida 3	Rápido	Numero de Salida	3
		Tiempo	1000 ms
	Lento	Numero de Salida	3
		Tiempo	5000 ms
Salida 4	Rápido	Numero de Salida	4
		Tiempo	700 ms
	Lento	Numero de Salida	4
		Tiempo	3000 ms

El manejo de recetas en *runtime* se lo realiza mediante la herramienta “Visor de recetas”, con este elemento se puede visualizar la recetas, modificar los valores de las variables, leer datos del controlador, cargar datos en el PLC, crear-eliminar registros, etc. Los datos se guardan en un archivo de memoria de la pantalla HMI en el caso que se realicen modificaciones, los botones que se encuentran añadidos a esta herramienta son los siguientes (véase Figura 4.13).

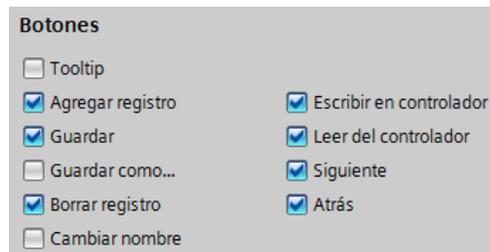


Figura 4.13: Botones añadidos al visor de recetas

### 4.2.3 Pruebas de Funcionamiento

#### a. Administración de Usuarios

En la pantalla inicial de la aplicación se encuentra el “Visor de usuarios”, al dar clic sobre este elemento nos permite ingresar el nombre y contraseña que deseamos. Si el acceso se realiza con éxito, en el elemento aparecerá el usuario como activo y el grupo al que pertenece (véase Figura 4.14).

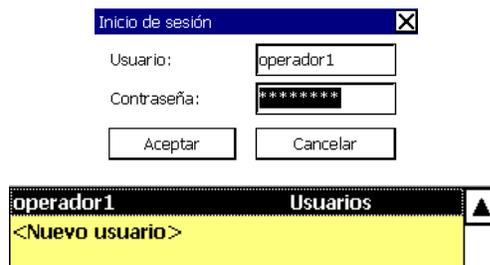


Figura 4.14: Login usuario 1

Mediante este objeto podremos además cambiar información de los usuarios siempre y cuando se ingrese con permisos de administración, en ese caso los datos que podremos editar son los expuestos en la figura 4.15.

A dialog box with a blue title bar containing a question mark and a close button. It contains four input fields: 'Usuario:' with the text 'operador1', 'Contraseña:' with masked characters '\*\*\*\*\*', 'Grupo:' with the text 'Usuarios', and 'Tiempo desconex:' with the text '0'. At the bottom are two buttons: 'Aceptar' and 'Cancelar'.

Figura 4.15: Editar usuarios en *runtime*

Los permisos de operación permiten al personal calificado interactuar con la aplicación SCADA y cambiar las consignas de proceso, en este caso se ha otorgado esta seguridad para el visor de recetas y el botón “Iniciar”. Si un usuario sin privilegios de operación intenta cambiar de receta, la aplicación mostrara el cuadro de diálogo para inicio de sesión permitiendo acceder al personal con otra cuenta válida (véase Figura 4.16).

A screenshot of a SCADA interface. A dialog box titled 'Inicio de sesión' is overlaid on top. The dialog box has a blue title bar with a close button. It contains two input fields: 'Usuario:' and 'Contraseña:'. Below the fields are two buttons: 'Aceptar' and 'Cancelar'. In the background, a green 'Iniciar' button is visible. On the left side of the background window, there is a vertical list of numbers 1, 2, 3, and 4 next to a green bar.

Figura 4.16: Inicio de sesión con un usuario autorizado

## b. Avisos

En la segunda pantalla se encuentra el visor de avisos que permite al usuario observar los eventos registrados en el proceso. Este elemento muestra en orden cronológico los avisos sucedidos con su respectiva información como: fecha, hora y texto del aviso. Cada vez que se encienda una salida del PLC de la 1-4, el visor de avisos registrará dicho acontecimiento. De igual manera al superar los 5V en la entrada analógica se generará un aviso (véase Figura 4.17).

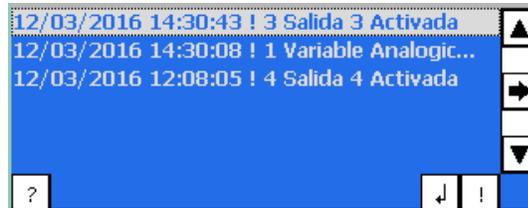


Figura 4.17: Visor de avisos en *runtime*

Al dar clic sobre el aviso y luego en el botón  se puede acceder al texto completo o comentario respectivo, además mediante el botón  se puede eliminar de la pantalla el aviso.



Figura 4.18: Texto completo del aviso

### c. Recetas

En la tercera pantalla se encuentra el visor de recetas el cual permite interactuar con los registros guardados y conmutar grupos de datos de proceso. En la pantalla principal del visor se puede observar todas las recetas creadas, al dar clic sobre una de ellas se podrá acceder a los registros contenidos (véase Figura 4.19).

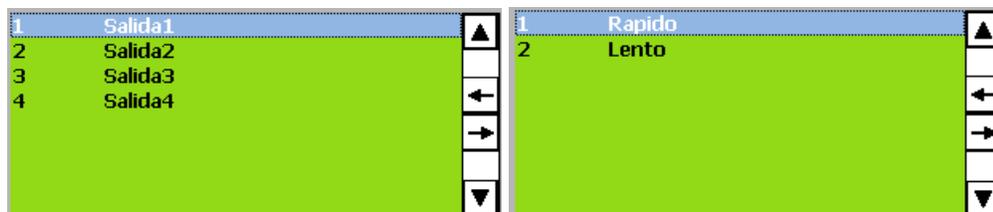


Figura 4.19: Uso del visor de recetas en *runtime*

Al dar clic sobre el registro se muestra en la pantalla todas las variables incluidas en él y sus respectivos valores. Al dar clic sobre una variable se abre un cuadro de entrada de

datos para modificar su valor, por el contrario al presionar el botón “flecha derecha” se puede acceder a opciones avanzadas como: cargar, leer, guardar (véase Figura 4.20).

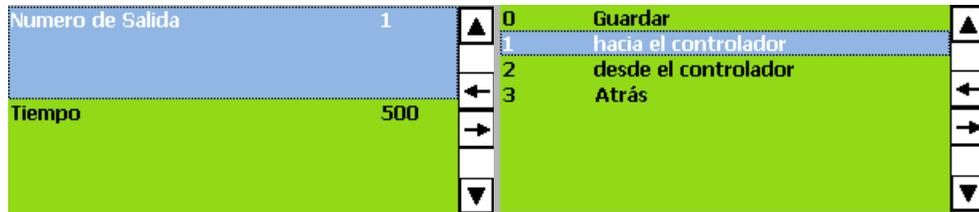


Figura 4.20: Elegir y cargar datos de registro al PLC

De esta manera los datos guardados en la receta 1 y en el registro 1 son escritos en las variables correspondientes del PLC, en el bloque *Main* del controlador se procede a realizar la lógica necesaria para cada variable.

### 4.3 CONCLUSIONES

Luego de la culminación del presente capítulo se logró un estudio general de los Sistemas SCADA, las posibilidades que estas aplicaciones brindan a los procesos son múltiples y se encuentran estrechamente relacionadas con los sistemas empresariales, además brindan funciones para la adquisición, análisis y presentación de la información. Se puede destacar los siguientes puntos:

- Los Sistemas SCADA son utilidades de *software* que sirven como interfaz entre los usuarios y los procesos de control, sus funciones son variadas enfocadas a la supervisión, análisis y control.
- Las HMIs son combinaciones de *software* y *hardware* que se constituyen en el medio de interacción entre el operador y los accionamientos, pueden ser paneles táctiles que presentan funciones de visualización y mando.
- En la actualidad los Sistemas SCADA se encuentran cada vez más integrados a las redes corporativas, esto acarrea una serie de beneficios relacionados con el mando remoto, análisis de datos, almacenamiento, etc. pero también conlleva un riesgo

referente a las medidas de seguridad que se deberán tomar para la conexión con el mundo informático empresarial.

- Algunas de las medidas de seguridad implementadas para la comunicación con Sistemas SCADA son: cortafuegos, *proxy*, VPN, IDS y DMZ, las cuales buscan proteger las pantallas y datos contra accesos no autorizados.
- El *software* WinCC creado por la empresa Siemens presta las bondades necesarias para el desarrollo de aplicaciones SCADA tanto con productos Simatic como con equipos de otros fabricantes, entre sus funcionalidad básicas se tiene: la programación con variables, avisos, recetas, ficheros, informes, *scripts*, planificación de tareas y configuración de usuarios.
- Las comunicaciones con los Sistemas SCADA se debe realizar de una manera clara y abierta con la posibilidad de interconectarse a diferentes infraestructuras, es así que mediante WinCC se puede establecer conexiones para redes: Profinet, Profibus, MPI, PPI, vía Simatic HTTP, comunicación redundante, vía OPC y con *drivers* de comunicación TCP para controladores Allen-Bradley, Mitsubishi, Modicon y Omron.
- Existen complementos para las aplicaciones SCADA que brindan al sistema funcionalidades más avanzadas enfocadas: al acceso remoto, documentación, análisis de procesos con herramientas de Office, portales Web, estaciones distribuidas, visualización en la *intranet* e *internet*, clientes y servidores OPC, integración mediante API con sistemas informáticos, comunicación con base de datos mediante OLE; las cuales enlazan cada vez más a los procesos con el mundo informático y permiten a los niveles de gestión contar con una mayor cantidad de herramientas para el mejoramiento de la productividad.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones:

- Las comunicaciones industriales son aspectos de gran importancia en el mundo ya que permiten enlazar los procesos productivos, por esta razón en la Universidad del Azuay se desarrolló un estudio e implementación de varios de los protocolos más utilizados en la actualidad como: Profinet, Profibus y Modbus; permitiendo el uso de aplicaciones con intercambio de datos entre controladores Simatic S7-1200.
- Una red Profinet puede ser establecida para la comunicación de todo tipo de equipos dentro de la Pirámide CIM, en este caso se desplegó una red entre controladores S7-1200 haciendo uso del protocolo *Ethernet*. La comunicación de este tipo es ideal para transferir información de toda clase entre controladores con una velocidad de 100Mbps/s.

Además se debe destacar que la transferencia de datos mediante Profinet permitió la comunicación entre los controladores mediante canales independientes con flujo en ambas direcciones, además se logró la optimización de la misma para el envío de todo un bloque de datos asociado en una variable *Struct*, de esta manera se pudo enviar todo un grupo de información.

- Se estableció una red Profibus entre dos controladores del laboratorio mediante la utilización de los módulos Maestro y Esclavo con una velocidad de 1,5Mbps/s, fue desplegada con la interfaz RS-485 para el envío de información desde el equipo principal.

Este tipo de red es ideal para la comunicación con el modelo maestro-esclavo por lo cual no presenta la misma versatilidad que Profinet, pero es apta para transferir información de un controlador a otro mediante áreas de transferencia de memoria, enlazando espacios de entrada-salida de los equipos.

- La red Modbus-RTU que se estableció entre controladores S7-1200 es ideal para la transferencia de información con el modelo maestro-esclavo, utiliza dos módulos de comunicación Punto a Punto conectados mediante una interfaz RS-485 a una velocidad de 19,2 Kbps.

Al igual que Profibus, la red Modbus está en la capacidad de transferir datos entre el equipo principal y sus esclavos, pero con la diferencia que dicho intercambio de información se realiza mediante peticiones de lectura o escritura implementado los diferentes códigos de función Modbus.

La red desplegada es más compleja que Profibus desde el punto de vista de programación pero en su lugar permite actuar directamente sobre las áreas de memoria del esclavo sin la necesidad de un área de transferencia, en este caso se tiene un solo equipo de la red que puede figurar como maestro.

- La red Modbus-TCP establecida hace uso del protocolo *Ethernet* para la comunicación de equipos con el modelo cliente-servidor, análogo a la estructura maestro-esclavo. Esta red tiene las mismas características de comunicación que su similar Modbus-RTU desde un punto de vista funcional, pero aprovecha la trama *Ethernet* y su velocidad de 100Mbit/s, además brinda la capacidad a los equipos de procesar comunicaciones como cliente o servidor.
- El prototipo construido para el control PID del nivel de un tanque cumple con todos los requerimientos necesarios, el sistema es regulado a través del objeto tecnológico PID\_COMPACT programado en un PCL S7-1200 de tipo DC/DC/DC.

La regulación del caudal que sale del tanque se logró a través de un control PWM de la bomba de agua, en este caso se realizó la optimización inicial y fina en TIA PORTAL con lo cual se obtuvo los parámetros PID para el regulador.

Si bien los valores calculados permiten una respuesta adecuada a la regulación pero muchas de las veces es necesario el ajuste manual de dichos parámetros con el fin de adaptar cada proceso a sus características específicas.

- El diseño de una fábrica virtual en FACTORY I/O permitió la simulación de un proceso de transportación industrial, en el cual se comunicó el controlador S7-1200 con los elementos como sensores y actuadores de la aplicación, de esta manera se presentó una alternativa para el desarrollo y prueba de los algoritmos de control programados en un PLC.
- Los sistemas SCADA son una parte importante de la automatización ya que permiten al usuario interactuar con el proceso o sacar información del mismo, de esta manera se implementó el uso de paneles HMI para el mando de las aplicaciones desarrolladas. Estas pantallas facilitan el control y la presentación de resultados permitiendo la intervención del usuario de manera clara y fácil.
- Las comunicaciones industriales son un aspecto que no ha sido considerado dentro de las cátedras de la carrera por lo cual se desarrolló un manual de prácticas de los temas abordados en la tesis, de esta manera se ha pensado cubrir estas competencias con el desarrollo de 9 prácticas de laboratorio, enfocadas al uso de todas las posibilidades ofrecidas por cada red para la interconexión de controladores.

**Recomendaciones:**

- Para la implementación de cada uno de los proyectos se debe tomar en cuenta los aspectos básicos de configuración de los PLCs y herramientas de *software* asociado, para ello se recomienda que el estudiante tenga conocimientos acerca de configuraciones de red para evitar conflictos entre los equipos.
- Al inicio de cada práctica se recomienda que primero se configure los equipos participantes, las versiones deben ser las adecuadas para no tener problemas de compatibilidad.
- Los estudiantes deberán tener conocimientos teóricos de programación de los PLCs, aspectos como las diferentes áreas de memoria, tipos de datos, ventanas de configuración son necesarios para el desarrollo de los proyectos.
- Es necesario conocer los fundamentos teóricos de los temas a abordar como: redes Profinet, Profibus, Modbus, control PID, sistemas SCADA para que el alumno pueda comprender y ejecutar los procedimientos descritos en la guía de práctica.
- Se puede integrar varios equipos como controladores y HMIs mediante el uso de una *switch*, en este caso es necesario ajustar todas las direcciones IP de los equipos con un número unívoco dentro de la red, si existiese problemas al grabar los equipos se recomienda conectar uno por uno los elementos con la PC y realizar una carga del programa directamente.
- En una red Profinet se recomienda que el estudiante tenga en cuenta cuáles son los equipos participantes y mediante qué canal se comunican, además de qué tipo de datos se envían y reciben, con el fin de que las variables en ambos PLCs no tengan problemas de coherencia.
- En una red Profibus se deberá prestar especial atención a las direcciones de las áreas de transferencia que se crean ya que mediante las cuales se pueden enviar o recibir datos.

- En una red Modbus-RTU se recomienda identificar plenamente las direcciones del estándar Modbus y su correspondencia con las áreas de memoria del PCL, además se debe conocer los diferentes códigos de función ya que de esta manera se podrá direccionar las comunicaciones para acciones de lectura o escritura.
- En una comunicación mediante Modbus-TCP se deberá tener presente las direcciones IP de los esclavos y por qué puerto se realizará el intercambio, además se debe reconocer los IDs de cada canal.
- Los bloques de datos que se agregan con las estructuras de las comunicaciones contiene los parámetros necesarios para el funcionamiento de las mismas, recordar que si se requiere un enlace con características diferentes se deberá agregar nuevas estructuras con nuevos bloques de datos.
- Para el desarrollo de un control PID se recomienda que la señal de medición sea lo más estable posible y que se identifique plenamente los límites tanto de la entrada como de la salida.
- Con la implementación de las redes de comunicación Industrial para el laboratorio de la Universidad del Azuay se recomienda que los proyectos se enfoquen a la integración de diferentes sistemas con el fin de conseguir aplicaciones más complejas.

## BIBLIOGRAFÍA

- Acromag. (2005). *INTRODUCTION TO MODBUS TCP/IP*. Wixom U.S.A.: Copyright.
- Alfaro, V. (2005). ACTUALIZACIÓN DEL MÉTODO DE SINTONIZACIÓN DE CONTROLADORES DE ZIEGLER Y NICHOLS. *Ingeniería*, 39-52.
- Améstegui, M. (2011). *APUNTES DE CONTROL PID*. La Paz.
- Balcells, J., & Romeral, J. L. (1997). *Autómatas Programables*. Barcelona: Marcombo.
- Blanco, J. (2006). *Comunicaciones Industriales, Ingeniería Electrónica y Automática*. Oviedo.
- Candelas, F. (2011). *Comunicación con RS-485 y Modbus*. GITE-IEA: Alicante.
- Center PROFIBUS Internacional Support. (2002). PROFIBUS System Description. *Profibus Technology and Application*, 10(02), 1-42.
- Conectronica. (10 de Mayo de 2013). *Conectrónica Tecnología y elementos de conexión y conectividad*. Obtenido de Redes Profinet: Generalidades: <http://www.conectronica.com/redes-industriales/redes-profinet-generalidades>
- Corrales, L. (2007). *Interfaces de Comunicación Industrial*. Copyright.
- Gómez, E., Leopoldo, G., & Leticia, H. (2014). *Buses de campo. Estrategias de aplicación*. Guanajuato: ECORFAN.
- Gomez, J. (2009). DISEÑO DE UN CONTROL DE TEMPERATURA CON PLC Y SISTEMA DE SUPERVISION SCADA VÍA ETHERNET. *Revista Politécnica*, 65-71.
- Guerrero, V., Yuste, R., & Martínez, L. (2009). *Comunicaiones Industriales*. Barcelona: Marcombo S.A.
- Industriales, E. d. (S.A.). *Profibus (PA/DP/FMS)*. Cartagena.
- karl, A., & Tore, H. (2009). *CONTROL PID AVANZADO*. Madrid: PEARSON EDUCACION S.A.

- Kaschel, H., & Pinto, E. (2010). *Análisis del Estado del Arte de los Buses Aplicados al Control de Procesos Industriales*. Santiago de Chile.
- López Fernandez, J. (2008). PROFIBUS. Vigo.
- López, J. (2008). PROFIBUS. Vigo.
- Mandado, E., Marcos, J., Pérez, S., Fernández, C., & Armesto, I. (2006). *Autómatas Programables SIEMENS*. Madrid: Copyright.
- Modicon. (1996). *Modbus Protocol Reference Guide*. Massachusetts: Copyright.
- Moreno, F., Becerra, J., & Jorge, M. (2012). Desarrollo e implementación de un módulo didáctico de automatización bajo una red de comunicación industrial Modbus. *Revista de la Universidad Francisco de Paula Santander*, 17(1), 20-26.
- M-System CO. LTD. (2002). *Modbus Protocol Reference Guide*. Osaka: Copyright.
- NETEON. (S.A.). *NETEON Industrial Networking*. (Copyright) Recuperado el 08 de 11 de 2015, de <https://www.neteon.net>
- Ogata, K. (1998). *INGENIERIA DE CONTROL MODERNA (TERCERA EDICION)*. Mexico: Copyright.
- Olaya, A., Barandica, A., & Guerrero, F. (2004). Implementación de una Red MODBUS/TCP. *Ingeniería y Competitividad*, 6(2), 1-11.
- PNO, P. N. (2011). PROFINET System Description. *Profibus-Profinet*, 4(132), 1-28.
- Profinet. (S.A.). *Infraestructura de red PROFINET*. España.
- Real-Games. (2014). *FACTORY I/O User Guide*. Copyright.
- Real-Games. (2014). *Parts Essentials*. Copyright.
- Rodríguez, A. (2008). *Comunicaciones Industriales Guía Práctica*. Copyright: Barcelona.
- Rodríguez, A. (2012). *Sistemas SCADA 3ra Edición*. Barcelona: Marcombo S.A.
- Salazar, C., & Correa, L. (2011). Buses de campo y protocolos en redes industriales. *Facultad de Ciencias e Ingeniería, Universidad Manizales*, 25, 83-109.

- SICK AG. (2012). *Ultrasonic Sensors*. Waldkirch: Copyright.
- SIEMENS. (2006). *Comunicación con SIMATIC*. Nürnberg: Copyright.
- SIEMENS. (2009). *S7-1200 Getting Started del S7-1200*. Nürnberg: Copyright.
- SIEMENS. (2012). *Automate with the leading Industrial Ethernet standard and profit now*. Nürnberg: Copyright.
- SIEMENS. (2012). *S7-1200 Profibus CM 1243-5*. Nürnberg: Copyright.
- SIEMENS. (2012). *SIMATIC HMI Paneles de Operador Basic Panelsf*. Nürnberg: Copyright.
- SIEMENS. (2014). *PROFIBUS con STEP 7 V13*. Nürnberg: Copyright.
- SIEMENS. (2014). *PROFINET con STEP 7 V13*. Nürnberg: Copyright.
- SIEMENS. (2014). *S7-1200 Profibus CM 1242-5*. Nürnberg: Copyright.
- SIEMENS. (2014). *S7-1200, S7-1500 Regulación PID*. Nürnberg: Copyright.
- SIEMENS. (2014). *WinCC Profesional V13 SP1*. Nürnberg: Copyright.
- SIEMENS. (2015). *S7 Controlador programable S7-1200*. Nürnberg: Copyright.
- SIEMENS. (S.A.). *IE TP Cords - Industrial Communication Siemens*. Obtenido de Industrial Ethernet Twisted Pair (TP) Cords: <http://w3.siemens.com/mcms/industrial-communication/en/ie/cabling-technology/patch-cables/pages/patch-cables.aspx>
- UNIOVI. (S.A.). *PROFINet*. Oviedo.
- WEG. (2012). *Manual de Comunicaciones RTU*. Mexico.
- XCSOURCE. (2006). *DC 12V Solar Hot Water Circulation Pump*. Obtenido de XCSOURCE: <http://xcsource.com/dc-12v-solar-hot-water-circulation-pump-brushless-motor-water-pump-480l-h-te087/>

**ANEXOS**  
**GUÍA DE PRÁCTICAS**

## ANEXO 1: PRÁCTICA 1

**TEMA:** Programación y Comunicación de un PLC Simatic S7-1200 con *FACTORY I/O* para la simulación de proyectos de manufactura.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la programación y comunicación con un entorno virtual diseñado en *FACTORY I/O* mediante el cual se simulará una aplicación de Manufactura.

### OBJETIVOS ESPECÍFICOS:

- Configurar el PLC y el *Software FACTORY I/O* para la comunicación de datos.
- Diseñar y construir una pequeña aplicación de manufactura en *FACTORY I/O*.
- Programar el control de la aplicación.
- Conectar el PLC a la simulación y verificar el funcionamiento.

### MATERIALES:

- Un PLC Simatic S7-1200.
- *Software* TIA PORTAL.
- *Software* *FACTORY I/O*
- Un cable *Ethernet* TP, CAT5.

### FUNDAMENTO TEÓRICO:

#### *FACTORY I/O*

*FACTORY I/O* es un *software* desarrollado por *Real Games* para la construcción y simulación de sistemas de automatización especialmente de traspotación y manufactura. Este *software* recrea un ambiente inspirado en fábricas reales con una alta calidad de gráficos en 3D basados en los elementos más comunes que se pueden encontrar dentro de dichos procesos.

*FACTORY I/O* es compatible con varios de los sistemas de control más usado en la actualidad, la comunicación entre los controladores y la aplicación virtual se lo realiza mediante *drivers* específicos para cada elemento, los cuales relacionan los puntos I/O representados como etiquetas.

Una etiqueta está definida por su nombre, dirección y el valor que puede ser discreto o analógico, cada dispositivo tiene varias etiquetas de entrada y salida asociadas para su control (Real-Games, *FACTORY I/O User Guide*, 2014).

Los *drivers* que están disponibles permiten la conexión de elementos como tarjetas Advantech USB 4704 & 4750, elementos Automgen TCP/IP, Modbus-TCP/IP cliente-servidor, Siemens LOGO TCP/IP, Siemens S7-300, 400, 1200, 1500 TCP/IP y Siemens PLCSIM (v5.4/5.5), además existe la posibilidad de un *driver* abierto OPEN (SDK) con el que se puede desarrollar aplicaciones de comunicación propias (Real-Games, *FACTORY I/O User Guide*, 2014).

Dentro de los elementos que se cuentan en *FACTORY I/O* se pueden destacar: cajas, bandas transportadoras, alineadores, soportes, rampas o tolvas, elevadores, clasificadores, paletas, placas giratorias, plataformas, guías, indicadores luminosos, sensores, botones de paro, emisores y receptores de cajas, etc. Todos estos elementos poseen sus propias características y propiedades sean analógicas o digitales que pueden ser comunicadas hacia los elementos de control.

*FACTORY I/O* se comunica con el PLC Simatic S7-1200 mediante el protocolo TCP/IP, de esta manera la aplicación pueden leer o escribir datos en la CPU mediante el modelo de comunicaciones S7, el cual establece un entorno el intercambio de información entre productos Simatic.

## PROCEDIMIENTO:

### a. Configuraciones en TIA PORTAL

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.

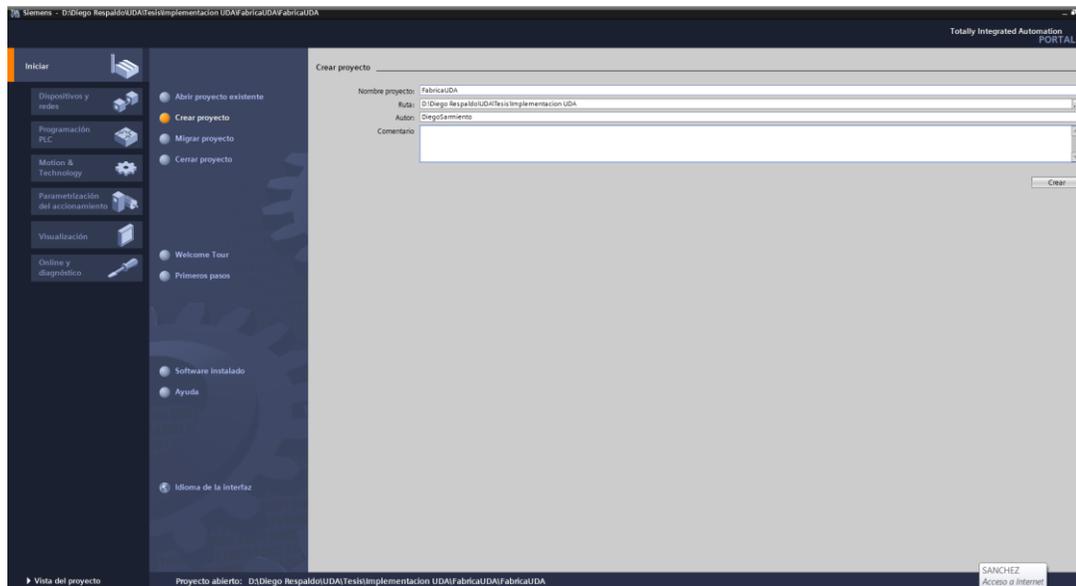


Figura P1.1: Pantalla de Inicio P1

3. Dar Clic en “Abrir Vista del Proyecto”.

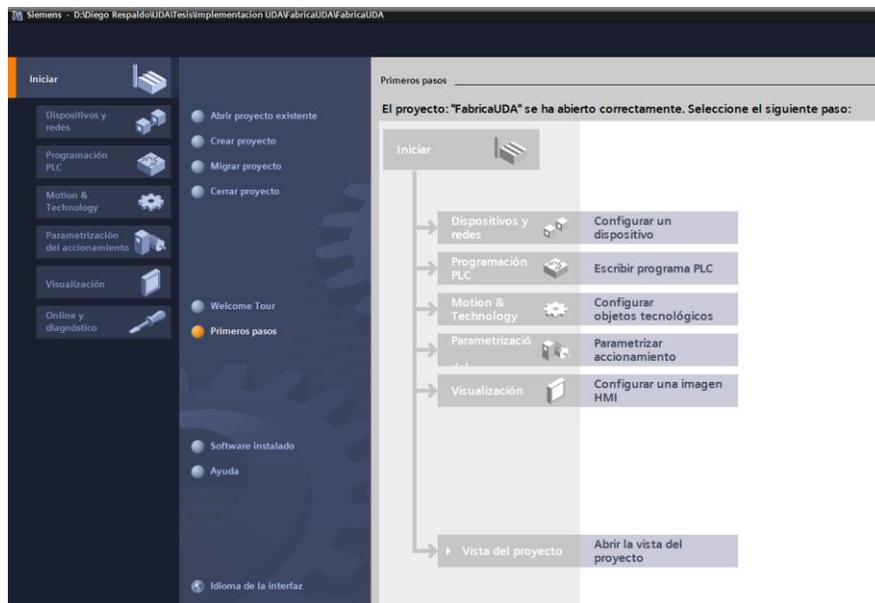


Figura P1.2: Ventana Primeros Pasos P1

- En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.

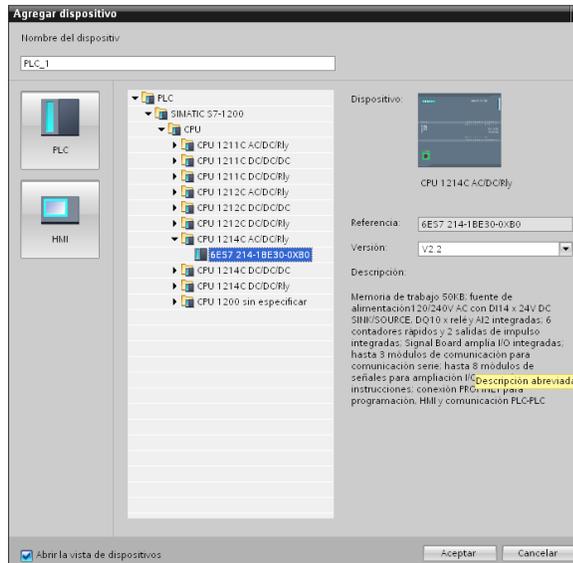


Figura P1.3: Agregar PLC1 al proyecto

- Dentro del PLC1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC.

En la Ventana Propiedades dar clic sobre “Protección”. Luego escoger la opción “Sin Protección”.

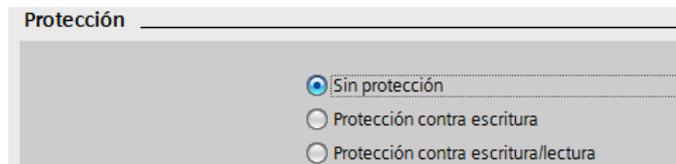


Figura P1.4: Establecer Ninguna Protección para el PLC1

- Lo siguiente será configurar las direcciones IP para que el PLC pueda intercambiar datos en la red, para ello dentro del PLC1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC.

En la Ventana Propiedades que se despliega en la parte inferior dar clic sobre “Interfaz Profinet”.

Luego en la parte del Protocolo IP colocamos:

Ip: 192.168.0.1

Mascara de Subred: 255.255.255.0

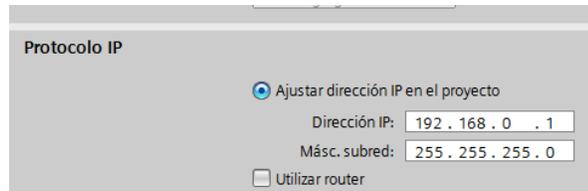


Figura P1.5: Configuración IP PLC1

7. Luego se deberá cambiar las direcciones para las entradas digitales del PLC, con el fin de que las señales de la aplicación virtual no interfieran con las entradas físicas conectadas.

Para ello dentro de la misma Ventana Propiedades dar clic sobre “DI 14/DQ 10”, luego dar clic sobre la categoría “Direcciones E/S”.

Luego en dirección de Entrada Inicial escribir 10 (De este manera los pulsantes conectados al PLC ocuparán el byte de entrada 10.)

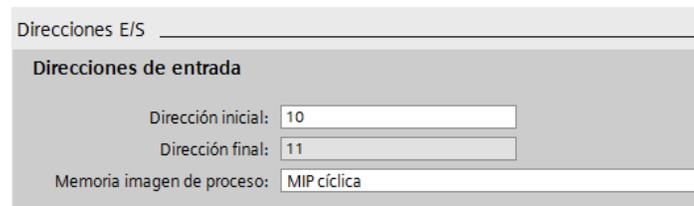


Figura P1.6: Ajustar Dirección Inicial de Entrada en el PLC1

8. Las señales de salida que se enlazarán a la aplicación virtual serán programadas a partir del byte 100 con el fin de no interferir con las salidas físicas del PLC, por lo tanto en Direcciones E/S no modificar la dirección inicial de salida.

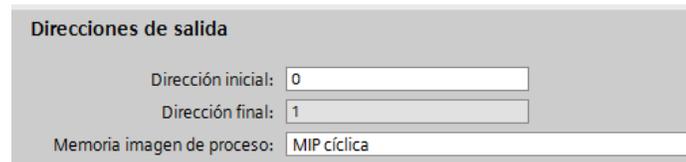


Figura P1.7: Dirección Inicial de Salida en el PLC1

9. Grabar el PLC y verificar que no exista errores.

## **b. Configuraciones y Diseño en *FACTORY I/O***

10. Abrir *FACTORY I/O* instalado en la PC.
11. En la barra superior dar clic en “File” y luego en “Driver Configuration”



Figura P1.8: Configuración de la Comunicación en *FACTORY I/O*

12. En la ventana que se despliega realizar las siguientes configuraciones:

CURRENT DRIVER: Siemens S7-1200 Ethernet (Escoger *driver* para el PLC S7-1200)

HOST: 192.168.0.1 (Dirección asignada al PLC)

NETWORK ADAPTER: Escoger la tarjeta de red de la PC.



Figura P1.9: Configuración del *Driver* para Comunicación con el PLC

Como se mencionó el software provee de ciertos elementos industriales que permiten recrear un proceso de manufactura, en este caso se explicará algunos de los elementos que serán útiles para esta práctica.

13. En la parte derecha dentro de la “Paleta” de objetos en la sección “*Light Load Conveyors*, Transportadores de Carga Ligera” se cuenta con bandas transportadoras de 6 y 4m, las cuales pueden girar en ambas direcciones y controlar su velocidad.



Figura P1.10: Paleta de Objetos *FACTORY I/O*

14. En la “Paleta” existen también elementos denominados “*Pop Up Wheel Sorter*” o rueda clasificadora para controlar el direccionamiento de los objetos que circulan por las bandas..

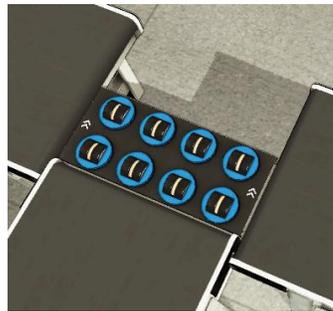


Figura P1.11: Rueda Clasificadora P1

15. Se cuenta además con alineadores circulares o “*Wheel Aligner*” e alineadores lineales o “*Aligner*” para direccionar el giro de las cajas sobre las bandas.

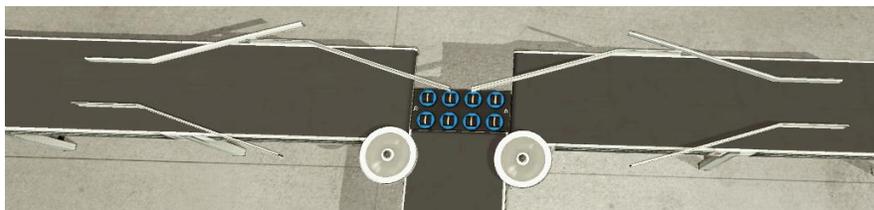


Figura P1.12: Alineadores P1

16. Para desfogar los elementos de las bandas se cuenta con rampas o “*Chute Conveyor*”, los cuales pueden ser ubicados a lo largo de la trayectoria.

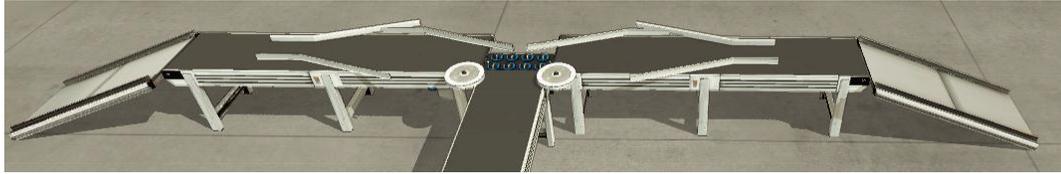


Figura P1.13: Rampas P1

17. El objeto que permite insertar cajas al sistema es el “*Emitter*” o Emisor, mediante sus propiedades se puede escoger tiempos y objetos que agrega al proceso.



Figura P1.14: Emisor de Cajas P1

18. Para colocar los sensores a lo largo de las bandas se utiliza los soportes o “*Bracket*”.



Figura P1.15: Soportes P1

19. Para determinar la presencia de las cajas se puede ocupar sensores difusos de proximidad, además de sensores analógicos de altura con un juego de rayos infrarrojos.

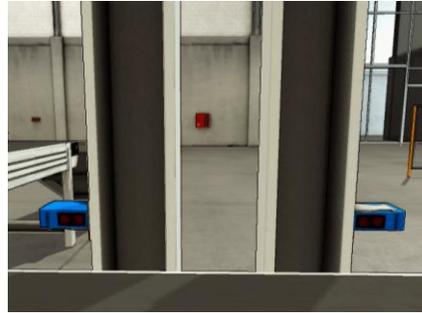


Figura P1.16: Sensores Difusos P1

20. Para acceder al control de los elementos, en la barra de programa dar clic en “View” y luego activar “Sensors Floating Tags” y “Actuador Floating Tags”, de esta manera se visualizará las etiquetas para cada elemento insertado.



Figura P1.17: Visualización Etiquetas de Sensores y Actuadores

- Por ejemplo en el caso de la rueda clasificadora se tiene 3 bits de control que permiten: activar los rodillos, girar a la derecha o girar a la izquierda.



Figura P1.18: Etiquetas Rueda Clasificadora

21. En la barra superior dar clic en “File” y luego en “Driver Configuration”.

En la ventana, en la parte derecha se muestra las variables para los actuadores y en la parte izquierda para los sensores.

Arrastrar las etiquetas hasta los campos correspondientes del controlador que aparece en la parte central de la ventana, de esta manera se asigna las direcciones del PLC que se enlazarán a las variables de la aplicación virtual.

Para las variables de salida colocar un Offset de 100.

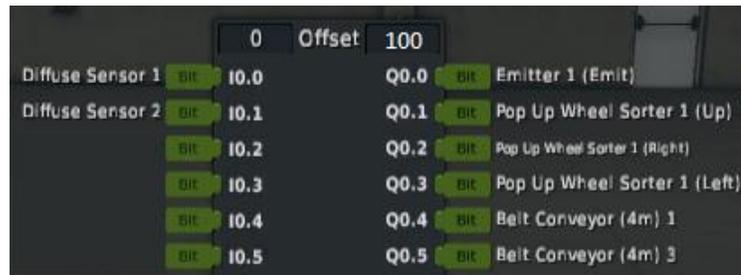


Figura P1.19: Asignación Etiquetas de Entrada y Salida

22. Para iniciar la simulación se deberá conectar el PLC con la computadora y hacer clic en el botón “RUN” de *FACTORY I/O*.



Figura P1.20: Iniciar Aplicación Virtual

## EJERCICIO PARA EL ALUMNO

En base a lo explicado acerca de la construcción, comunicación y simulación de un proyecto de manufactura mediante *FACTORY I/O*, el alumno deberá crear un proyecto que permita el control desde un PCL Simatic S7-1200 con las siguientes consideraciones:

- En el software de simulación construir una pequeña aplicación con tres bandas transportadoras colocadas en forma de T con el fin de direccionar tráfico hacia la izquierda o derecha.
- Se deberá insertar los tres tipos de cajas al sistema y mediante dos sensores de proximidad difusos determinar el tamaño de los objetos, luego direccionar las cajas cortas hacia la izquierda y las largas hacia la derecha.
- En TIA PORTAL crear el control para el sistema y enlazar las etiquetas de entrada o salida, además incluir en la programación un contador de cajas cortas y largas.

- Se deberá tener 3 pulsantes, uno para el inicio de la aplicación, otro para el paro de las bandas y otro para el *reset* de los contadores.
- Los rodillos de la rueda clasificadora deberán girar solo una vez, es decir, solo si el tamaño de la caja cambia.

La aplicación deberá quedar de la siguiente manera.



Figura P1.21: Ejercicio para el Alumno, Proceso en *FACTORY I/O*

### **CONCLUSIONES:**

## ANEXO 2: PRÁCTICA 2

**TEMA:** Comunicación de un PLC con una HMI KTP600 vía Profinet.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la comunicación de un PLC Simatic S7-1200 con una pantalla HMI KTP600 vía Profinet.

### OBJETIVOS ESPECÍFICOS:

- Familiarizarse con las aplicaciones SCADA.
- Configurar una comunicación entre un PLC y una HMI.
- Leer y Presentar datos en una pantalla HMI.
- Conocer las herramientas WinCC para la elaboración de interfaces SCADA.

### MATERIALES:

- Un PLC Simatic S7-1200.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.
- Una Pantalla HMI KTP600

### FUNDAMENTO TEÓRICO:

#### Interfaz HMI

Una Interfaz Hombre-Máquina o HMI es un mecanismo combinado de *hardware* y *software* que le permite a un operador interactuar con un proceso productivo mediante pantallas y funciones específicas, además comprende sinópticos de control y sistemas de presentación gráfica (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

Un HMI tiene la capacidad de visualizar y determinar datos de los accionamientos como estados (*ON/OFF*) o magnitudes de las variables físicas involucradas; puede ser desde un simple interruptor e indicador luminoso para encender algún dispositivo, hasta varias pantallas que esquematizan todo un proceso de producción incluyendo valores en tiempo real y acciones de supervisión (Corrales, 2007).

Existen varias herramientas de *software* que permiten el manejo de un sistema SCADA a través de las cuales se desarrolla e implementa varias pantallas HMI, estas actúan como interfaz entre el proceso y el operador permitiendo acceder a consignas o configuraciones del mismo, además presenta herramientas para la comunicación con sistemas superiores de la pirámide CIM a través de servidores de datos.

Existen diversas versiones dentro de las pantallas HMI usadas como paneles de operador en las fábricas para aplicaciones sencillas para las cuales no se necesitan los complementos informáticos de los sistemas SCADA basados en PC. Existen paneles básicos desde las KTP300 hasta las KTP1500 cuyo tamaño va desde 3" a 15". Las características principales de las pantallas KTP600 se muestran en la Tabla P2.1

Tabla P2.1: Características Pantallas KTP600

	KTP400 Basic mono PN	KTP400 Basic color PN	KTP600 Basic mono PN	KTP600 Basic color DP	KTP600 Basic color PN
Tipo	LCD mono FSTN	LCD-TFT	LCD mono FSTN	LCD-TFT	
Área activa de la pantalla	76,8 x 57,6 mm (3,8")	95 x 53,8 mm (4,3")	115,2 x 86,4 mm (5,7")		
Resolución	320 x 240 píxeles	480 x 272 píxeles	320 x 240 píxeles		
Colores representables	4 niveles de gris	256	4 niveles de gris	256	
Regulación de contraste	Sí	No	Sí	No	
Ajuste de brillo	No	Sí	No	No	
Retroiluminación	LED	LED	CCFL		
Half Brightness Life Time (MTBF <sup>1</sup> )	30.000 h	50.000 h	50.000 h		
Categoría de error de píxel según DIN EN ISO 13406-2	-	II	-	II	

Fuente: (SIEMENS, SIMATIC HMI Paneles de Operador Basic Panelsf, 2012)

Las Pantallas KTP600 estan disponibles en dos versiones DP y PN según el tipo de red de comunicación, la primera para una conexión a redes Profibus y la segunda para el enlace con redes Profinet mediante la interfaz *Ethernet* (SIEMENS, SIMATIC HMI Paneles de Operador Basic Panelsf, 2012).

Las funcionalidades con las que cuentan los *Basic Panels* son: un sistema de avisos analógicos y digitales, uso de recetas, visor de curvas, administración de usuarios, animaciones de apariencia y visibilidad, movimiento horizontal o vertical de los objetos, entre otras.

## PROCEDIMIENTO:

### a. Configuraciones en TIA PORTAL

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en “Abrir Vista del Proyecto”.
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 DC/DC/DC.

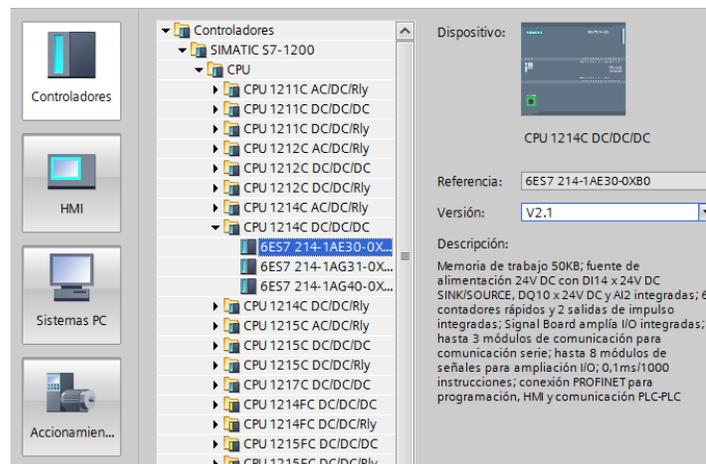


Figura P2.1: Agregar PLC al proyecto

- Luego en el “Árbol del Proyecto” dar doble clic en “Agregar Dispositivos” para insertar la pantalla HMI, en el Laboratorio se cuenta con una Pantalla Táctil de 6” KTP600 PN.

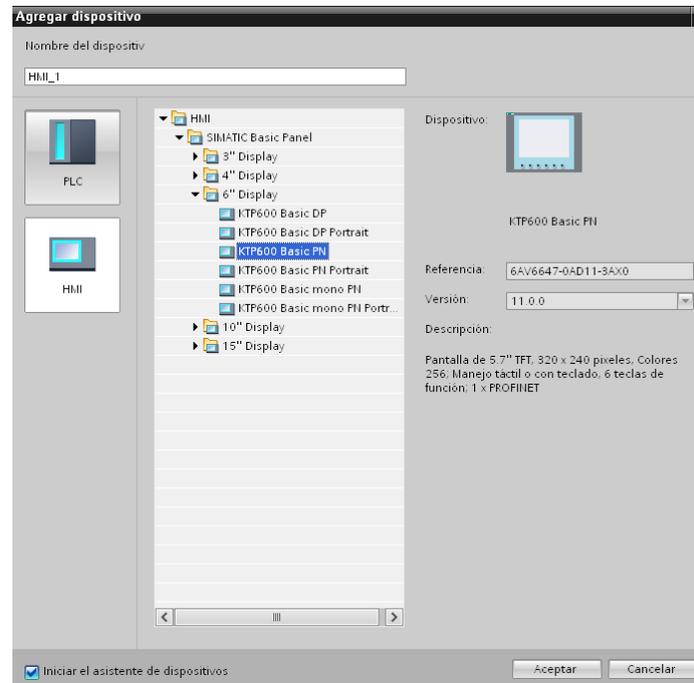


Figura P2.2: Agregar Pantalla HMI

- Luego dar clic en “Seleccionar PLC” y escoger el PLC al que se vinculará la pantalla, en este caso será PLC1. A continuación dar clic en siguiente.

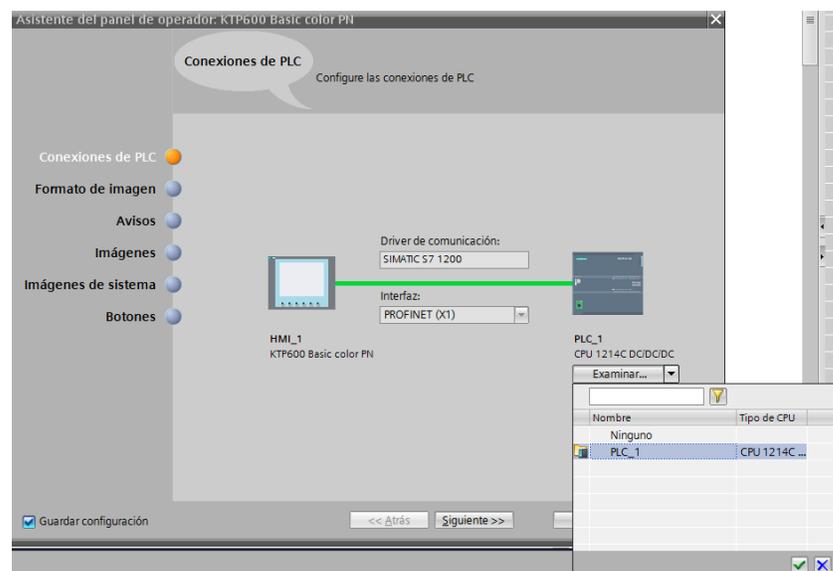


Figura P2.3: Vincular PLC a pantalla HMI

7. En la nueva ventana escoger un color de fondo para la pantalla y seleccionar si se desea un encabezado compuesto por la hora y el logotipo. A continuación dar clic en siguiente.

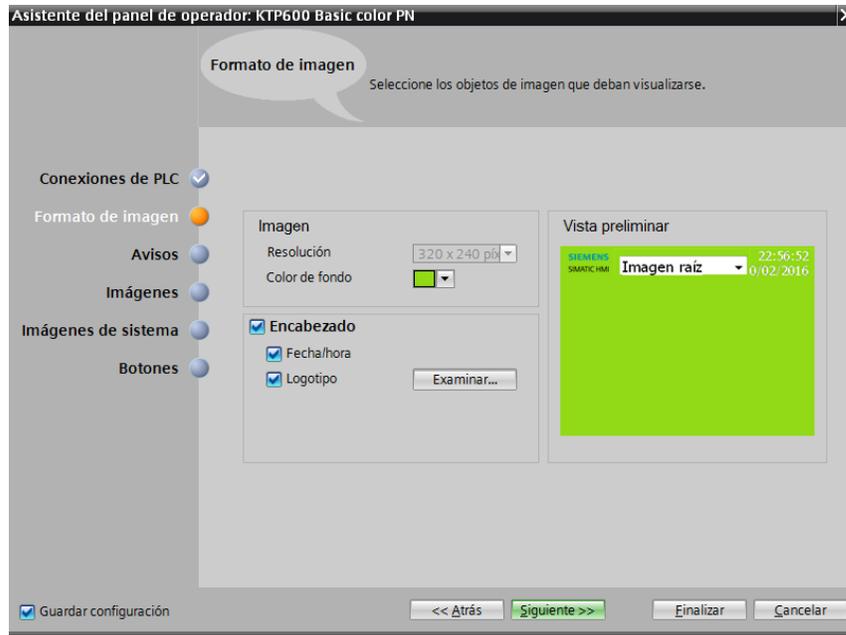


Figura P2.4: Configuración color de fondo pantalla HMI

8. En la ventana de "Avisos" se recomienda desactivar todas las casillas de verificación para no visualizar los avisos del sistema.

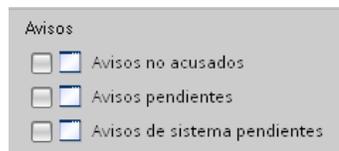


Figura P2.5: Desactivar Avisos de Sistema

9. En las ventanas "Imágenes" e "Imágenes de sistema" se recomienda no realizar cambios.
10. En la ventana final "Botones" se recomienda dar clic en la casilla "Abajo" para agregar un campo especial ocupado por botones prediseñados del sistema. Luego añadir los botones para apagado *Runtime* y *Home*, deberá quedar de la siguiente manera.

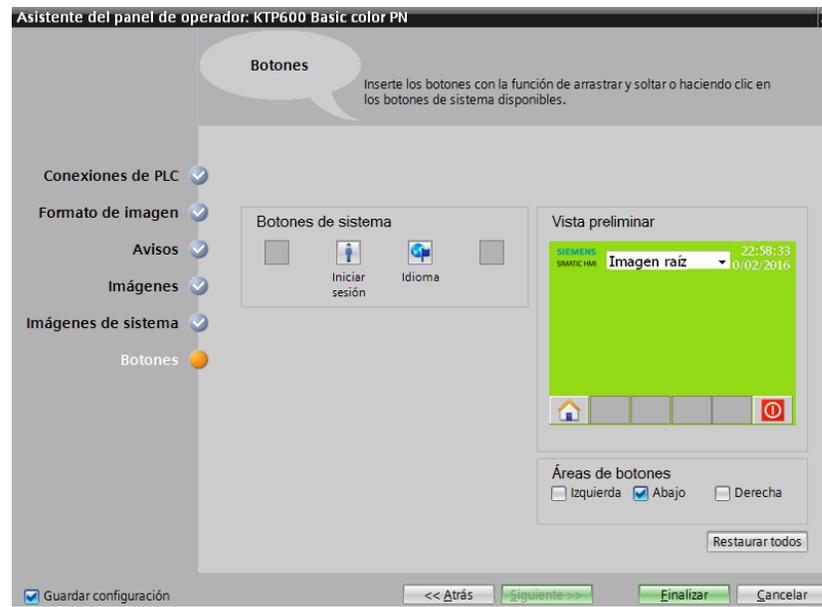


Figura P2.6: Agregar Botones del Sistema

A continuación dar clic en “Finalizar”.

11. En el árbol del proyecto dar clic en “Dispositivos y redes” y verificar que exista la conexión Profinet entre el PLC y la HMI.

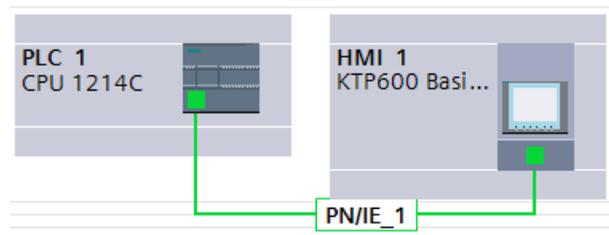


Figura P2.7: Red Profinet entre PLC y HMI

12. Lo siguiente será configurar la dirección IP, para ello dentro de la HMI1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre la HMI.

En la Ventana Propiedades que se despliega en la parte inferior dar clic sobre “Interfaz Profinet”.

Luego en la parte del Protocolo IP colocar:

Ip: 192.168.0.2

Mascara de Subred: 255.255.255.0

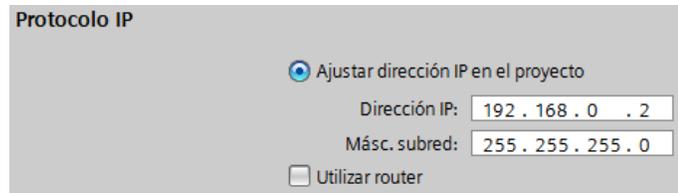


Figura P2.8: Configurar dirección IP de la HMI

### b. Entrada de Datos desde la HMI

13. Lo primero será crear una variable en el PLC a la cual se escribirá el valor ingresado en la HMI. Para ello dentro del PLC expandir la categoría “Variables PLC” y luego dar doble clic en “Tabla de variables estándar”.

Ingresar una variable “Valor” de tipo *Int* en la dirección MW15.

Tabla de variables estándar			
	Nombre	Tipo de datos	Dirección
1	Valor	Int	%MW15

Figura P2.9: Agregar variable de tipo Int en el PLC

14. Dentro de la pantalla HMI1 expandir la categoría “Imágenes” y luego dar clic en “Imagen raíz”.

En la parte derecha de TIA PORTAL dentro de la categoría “Elementos” seleccionar y arrastrar un “campo de texto”.

15. Dar doble clic sobre el elemento agregado y luego sobre la Pestaña “Eventos”.

Luego en la parte izquierda de la ventana dar clic en “Activar” y en el área de trabajo dar clic en “Agregar función”.

Escoger la función “Definir Variable”



Figura P2.10: Función “Definir Variable”

16. En el campo “Variable (Salida)” dar clic y escoger la variable “Valor” creada en el PLC:

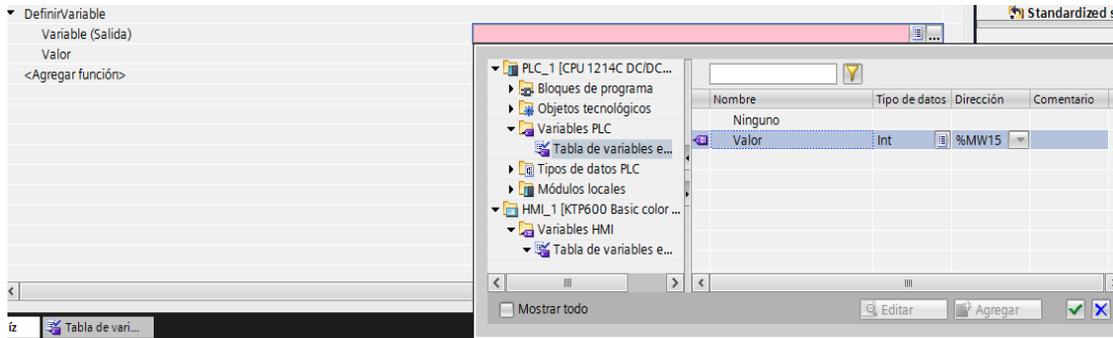


Figura P2.11: Seleccionar Variable creada

17. En el campo “Valor” dar clic en la parte derecha y escoger “Variable HMI”.

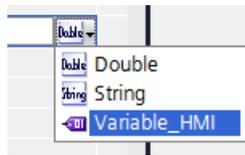


Figura P2.12: Seleccionar valor desde Variable HMI

18. Luego dar clic en el icono  y seleccionar nuevamente la variable “Valor”.

19. Dar clic en la Pestaña “Propiedades” y dentro de la categoría “General” escoger la variable de proceso a ser mostrada en el campo de texto.

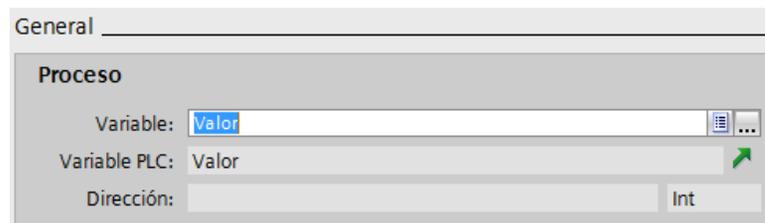


Figura P2.13: Seleccionar el valor de proceso

20. Agregar una tabla de observación para constatar el cambio en la variable del PLC. Dentro del PLC dar clic en “Tablas de observación y forzado permanente” y luego doble clic en “Agregar tabla de observación”.

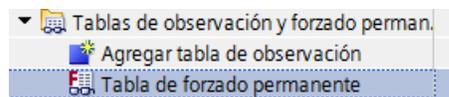


Figura P2.14: Agregar Tabla de Observación

21. Ingresar el nombre de la variable a observar en este caso “Valor”.

22. Grabar el PLC y la HMI

23. Conectarse ONLINE con el PLC e ingresar un número en el campo de texto de la HMI, verificar su valor en el PLC.



Figura P2.15: Verificar cambio de valor en la variable “Valor”

### c. Botones en la HMI

24. Crear una variable en el PLC llamada “Salida 0” de tipo *bool* cuya dirección sea Q0.0
25. En la parte derecha de TIA PORTAL dentro de la categoría “Elementos” seleccionar y arrastrar un “Botón”.
26. Dar doble clic en el botón y luego clic en la Pestana “Eventos”.
27. Similar al paso 15 dar clic en la acción “Pulsar” y agregar la función “Activar Bit”, luego escoger la variable creada.

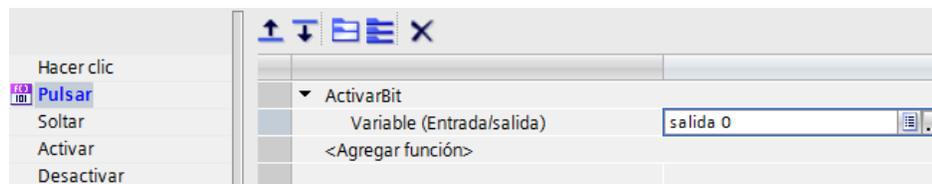


Figura P2.16: Función “ActivarBit”

28. Repetir los pasos 24-27 para agregar un botón de nombre “Desactivar”, asignarle la función “Desactivar Bit” y escoger la variable de “Salida 0”.

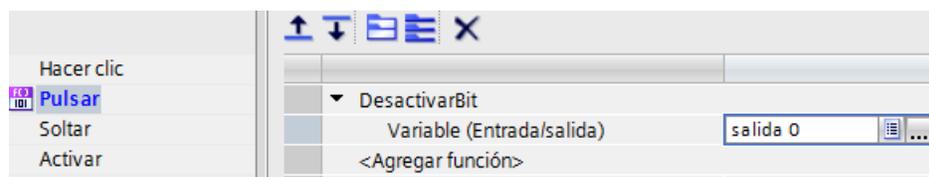


Figura P2.17: Función “DesactivarBit”

29. Grabar ambos equipos y verificar la activación y desactivación de la salida Q0.0 desde la HMI.

#### d. Agregar y Dinamizar movimientos de los Gráficos WinCC

30. Existen varios gráficos agregados por defecto en la librería de WinCC, estos elementos ayudan a mejorar la presentación visual de la pantalla y esquematizar los procesos de mejor manera. Para agregar se debe dar clic en la sección “Gráficos” de la parte derecha de TIA PORTAL.
31. Dentro de la categoría “*Industries*” y dentro de “*Food*” seleccionar y agregar una botella.
32. Luego dentro de la categoría “*Automation Equipment*” y dentro de “*Conveyors, miscellaneous*” seleccionar y arrastrar una banda transportadora..

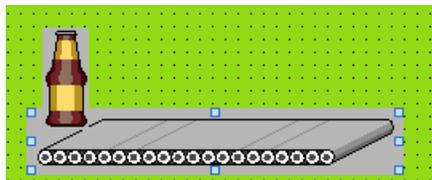


Figura P2.18: Agregar Objetos Prediseñados de WinCC

33. Dentro de las Propiedad de cada elemento y dentro de “Apariencia” escoger un color de fondo adecuado.
34. Dentro del PLC1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC.  
En la Ventana Propiedades que se despliega en la parte inferior dar clic sobre “Marcas de sistema y de ciclo”.  
Luego activar seleccionar la casilla de verificación “Activar la utilización del byte de marcas de ciclo”.

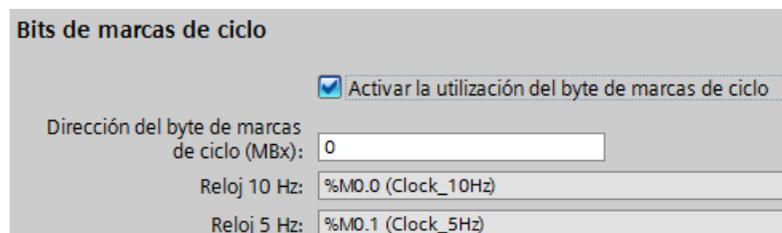


Figura P2.19: Activar Marcas de Ciclo

35. Agregar el siguiente segmento en el PLC para incrementar un contador que determinara el movimiento del objeto.

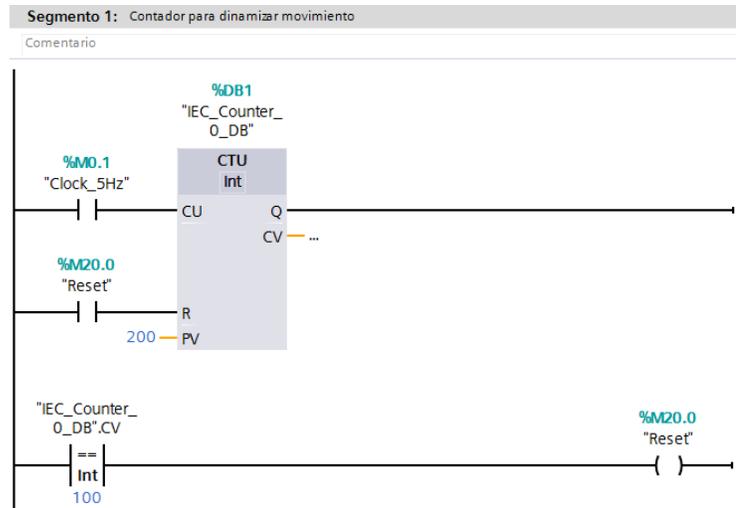


Figura P2.20: Generar contador a partir de reloj interno de 5Hz

36. En la Imagen Raíz seleccionar elemento Botella y dar clic en “Animaciones”, luego dar clic en “Agregar animación” para un movimiento horizontal. Escoger la variable de proceso correspondiente al conteo actual de la estructura agregada en el PLC.

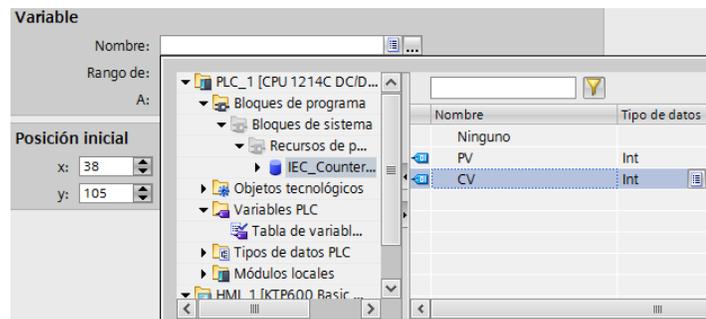


Figura P2.21: Seleccionar variable de conteo actual

37. Grabar ambos equipos.
38. Verificar el movimiento horizontal de la botella.

## EJERCICIO PARA EL ALUMNO

En base a lo revisado acerca de la conexión y programación de una pantalla HMI el alumno deberá realizar un proyecto que para el encendido de las cuatro salidas digitales y la animación de movimiento de una botella con tres velocidades seleccionables.

Las consideraciones son las siguientes:

- La pantalla HMI deberá contener un campo de texto para el ingreso del número de salida que se desea encender.
- Deberán existir dos botones: uno para encendido y otro para apagado.
- Al presionar el botón encender se deberá activar la salida especificado en el campo de texto, si se cambiase el número de salida se deberá apagar la anterior hasta pulsar nuevamente el botón encender
- La pantalla deberá contener un segundo campo de texto para el ingreso de la velocidad de movimiento de la botella, podrá ser 1, 2 o 3 que corresponderá a diferentes velocidades elegidas por el alumno.
- El movimiento de la botella solo se realiza si la salida Q0.0 esta activada.

**CONCLUSIONES:**

### ANEXO 3: PRÁCTICA 3

**TEMA:** Control PID del Nivel de un Tanque.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la programación y puesta en servicio un objeto tecnológico PID\_COMPACT para el control PID del Nivel de Agua de un Tanque.

#### OBJETIVOS ESPECÍFICOS:

- Acondicionar la medición del sensor de ultrasonido dentro del rango máximo y mínimo de agua del tanque.
- Configurar el objeto tecnológico PID\_COMPACT para la regulación.
- Realizar la Optimización Inicial del Sistema.
- Realizar la Optimización Fina del Sistema.
- Comprobar el funcionamiento de la regulación.

#### MATERIALES:

- Un PLC Simatic S7-1200 de tipo DC/DC/DC
- *Software* TIA PORTAL.
- Prototipo de laboratorio para el control de nivel del Tanque.

#### FUNDAMENTO TEÓRICO:

##### CONTROL PID

El controlador PID es un algoritmo utilizado ampliamente en la actualidad en un gran número de procesos, basa su funcionamiento en un sistema de lazo realimentado que permite establecer el error entre la señal de referencia o consigna y la señal de salida de la

planta. Este error es medido y utilizado para el cálculo de la señal de ajuste mediante tres acciones: la proporcional, la integradora y la derivativa (karl & Tore, 2009).

El algoritmo de control PID puede ser descrito mediante un lazo de realimentación como se muestra en la Figura P3.1.

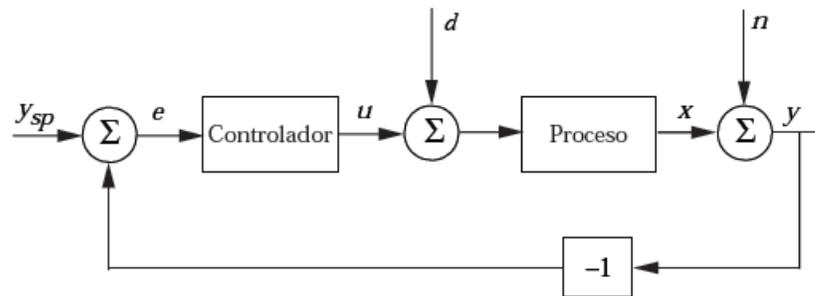


Figura P3.1: Lazo de Realimentación

Donde la  $\mu$  es la señal de control y  $e$  es el error calculado a través de la diferencia de la señal de referencia  $Y_{sp}$  y la señal de salida  $Y$ .

La señal de control es el resultado de las tres acciones mencionadas: la proporcional (calculada proporcionalmente al error), la integradora (calculada en base a la integración de los errores pasados) y la derivativa (calculada a través de la derivada del error). Los parámetros que estas acciones representan son: la ganancia proporcional  $K$ , el Tiempo de integración  $T_i$  y el tiempo derivativo  $T_d$  (karl & Tore, 2009).

#### *Sintonización de los Controladores PID, Reglas de Ziegler-Nichols*

Existen ciertos métodos que permiten seleccionar los parámetros del controlador según las características de desempeño, a este proceso se conoce como sintonización de los controladores PID. (Ogata, 1998).

En su estudio Ziegler-Nichols propusieron unas reglas para sintonizar los controladores PID determinando el valor de la Ganancia Proporcional  $K_p$ , del Tiempo de Integración  $T_i$  y del Tiempo Derivativo  $T_d$  en base a la respuesta transitoria de un sistema específico (Alfaro, 2005).

- a. **Primer Método:** El primer método trata de establecer experimentalmente la respuesta del sistema a una entrada de escalón unitario, la curva de respuesta generalmente tiene forma de S.
- b. **Segundo Método:** El segundo método propone la utilización de oscilaciones limitadas y continuas de la salida para determinar los parámetros de ganancia crítica.

#### *Objeto Tecnológico PID\_COMPACT*

El objeto PID\_Compact (Regulador PID Universal con optimización integrada) permite control un proceso mediante un algoritmo PID basado en las tres acciones de control: proporcional, integradora y derivativa, el cálculo de la salida de control se lo realiza en base a una medición continua del valor real del proceso (SIEMENS, S7 Controlador programable S7-1200, 2015).

El regulador PID\_Compact está incorporado además con una función de *Anti-Windup*<sup>53</sup> y ponderación de las acciones P y D, de acuerdo a lo presentado por (SIEMENS, S7 Controlador programable S7-1200, 2015) la función matemática del regulador es la siguiente:

$$y = K_p \left[ (b * w - x) + \frac{1}{T_i * s} (w - x) + \frac{T_d * s}{a * T_d * s + 1} (c * w - x) \right]$$

---

<sup>53</sup> *AntiWindup*: Es una técnica para contrarrestar el efecto *windup*, que ocurre en un controlador PID cuando un cambio brusco en la referencia ocasiona una acumulación del error en la integral, este valor sigue incrementado.

En donde:

y: Valor de Salida del algoritmo PID

$K_p$  : Ganancia Proporcional

s: Operador Laplaciano

b: Ponderación de la acción P

w: Consigna

x: Valor real

$T_i$  : Tiempo de integración

$T_d$ : Tiempo derivativo

a: Coeficiente de retardo de la acción derivativa

c: Ponderación de la acción D

## PROCEDIMIENTO:

### a. Configuración del Proyecto y del Hardware

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en “Abrir Vista del Proyecto”.
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1212 DC/DC/DC.

### b. Configuración PID\_COMPACT

5. El objeto PID\_COMPACT permite calcular los parámetros para el control PID, por la naturaleza del algoritmo este objeto tecnológico requiere cierto tiempo entre ejecución y ejecución, por lo cual deberá ser llamado mediante una OB de alarma cíclica.

Para ello en el “Árbol de Proyecto” y dentro de “Bloques de Programa” dar doble clic en “Agregar nuevo bloque”.

Luego elegir “Bloques de Organización” y dentro de ellos dar clic sobre “*Cyclic interrupt*”. Luego dar clic en “Aceptar”.

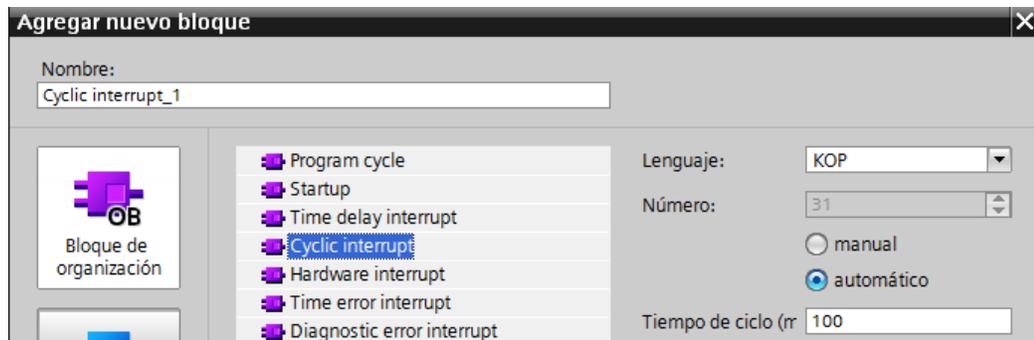


Figura P3.2: Insertar un bloque de ejecución cíclica

6. Dentro del bloque se deberá insertar el objeto PID\_COMPACT, para ello en la parte derecha de TIA PORTAL en el “Árbol de Instrucciones” en la sección “Tecnología” y dentro de la categoría “Compact PID” agregar al bloque MAIN la estructura PID\_COMPACT.

Aceptar la creación de un nuevo bloque de datos.

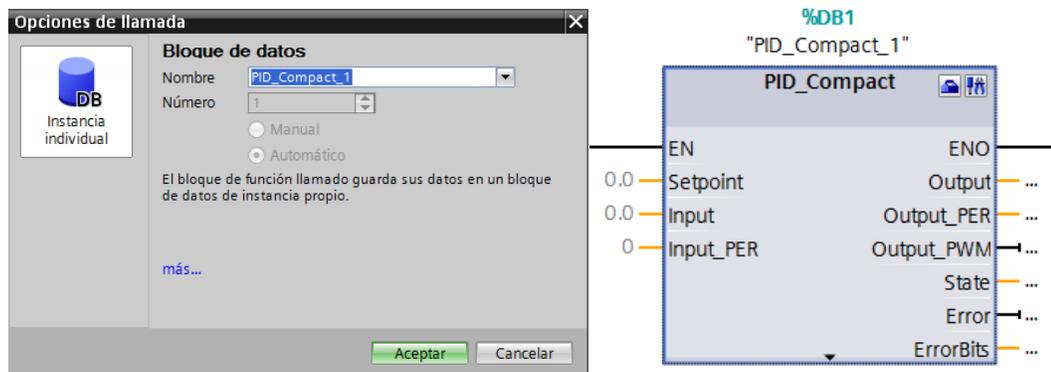


Figura P3.3: Insertar un bloque de ejecución cíclica

7. En la parte Izquierda en el “Árbol del Proyecto” y bajo la “Objetos tecnológicos” expandir la estructura agregada y dar clic en “Configuración”.
8. En la ventana que se despliega en la parte izquierda dar clic sobre la categoría “Tipo de Regulación”.

En la lista desplegable se puede escoger el tipo de regulación como; presión, longitud, masa, etc y la unidad. En este caso se escoger el tipo de regulación “General” es decir en Porcentaje para poder escalar a cualquier unidad.

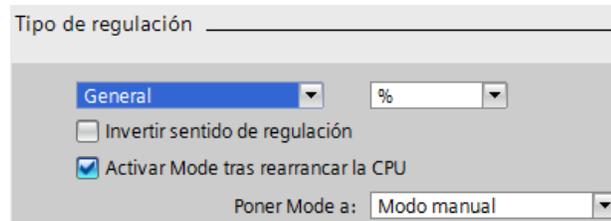


Figura P3.4: Escoger Tipo de Regulación

9. Luego dar ciclo sobre la categoría “Parámetros de entrada/salida”, en esta ventana se deberá escoger el tipo de entrada y salida que tendrá el objeto. En este caso escoger un ingreso de tipo “*Input*” que corresponde a una variable del PLC y una salida de tipo “*Output PWM*”.

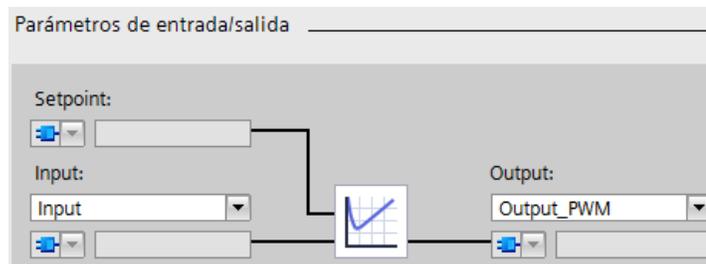


Figura P3.5: Escoger Parámetro de Entrada/Salida

10. Luego dar ciclo sobre la categoría “Límites del valor real”, en esta ventana se deberá escoger el valor máximo y mínimo que tendrá la entrada del sistema, caso contrario se generará un error en el objeto PID.

Para esta práctica escoger un rango de entrada desde 0 a 1000% con el objetivo de tener una regulación más fina.

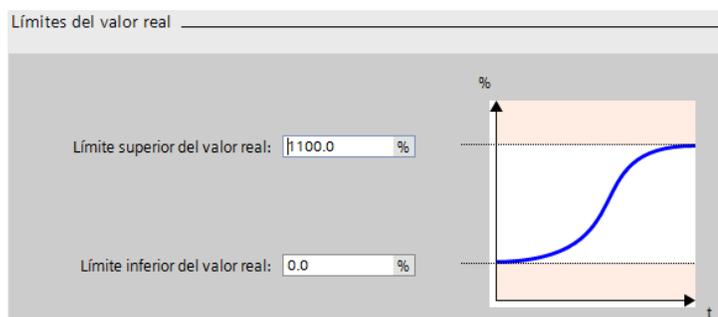


Figura P3.6: Límites del valor real

11. Luego dar ciclo sobre la categoría “Límites del valor de salida”, esta configuración es importante ya que establece en este caso el rango de salida PWM. El valor

mínimo es relevante porque determina desde que porcentaje un incremento de salida generará un bombeo de agua.

Colocar un rango de 11 a 100%.

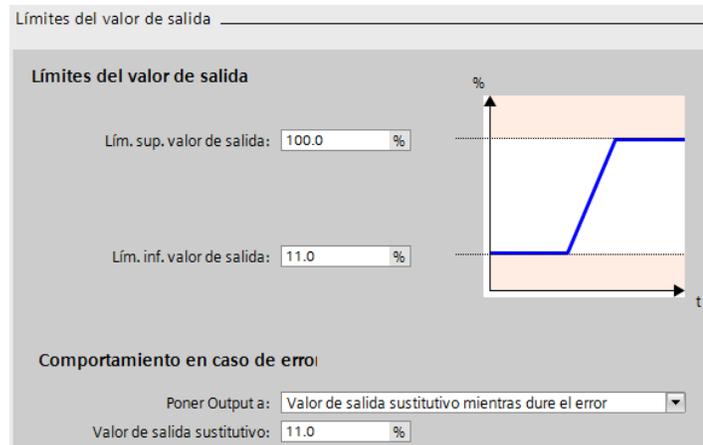


Figura P3.7: Límites del valor real

12. Ahora se deberá escalar el valor medido por el sensor para el rango 0-1000%, para ello hacer los siguiente:

Medir el valor del sensor sin agua

Medir el valor del sensor con el tanque lleno al máximo que deseemos.

Los límites establecidos previamente en el laboratorio fueron:

Valor del registro del sensor sin agua: 23000

Valor del registro del sensor con agua: 500.

13. Con los valores obtenidos en el paso anterior se deberá escalar dichos valores para un porcentaje de llenado de 0 a 1000%, recordar que el parámetro input del PID deberá incrementarse al reducir el agua.

Agregar los siguientes bloques.

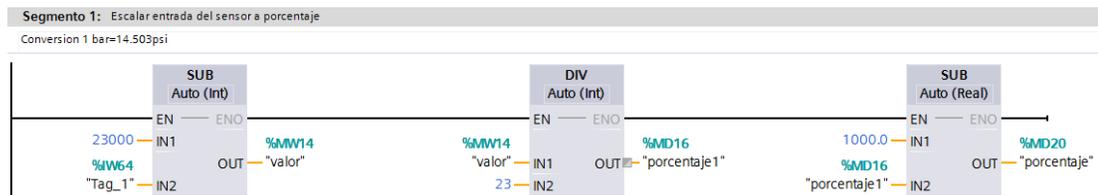


Figura P3.8: Escalar el valor del Sensor

14. Regresar al bloque de ejecución cíclica y agregar las variables a la instrucción PID agregada.

Setpoint: 500

Input: Porcentaje (Variable escalada).

Output\_PWM: Q0.0

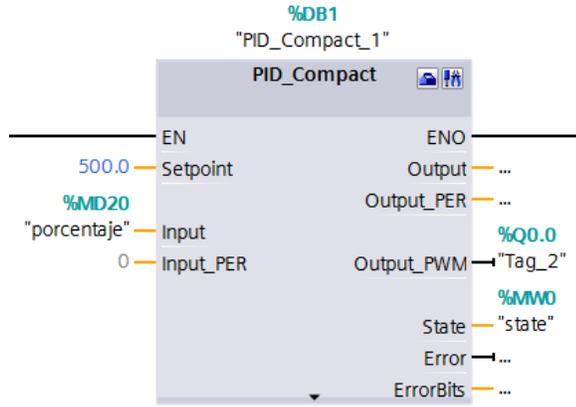


Figura P3.9: Instrucción PID\_COMPACT

### c. Puesta en Servicio de PID\_COMPACT

15. En la parte Izquierda en el “Árbol del Proyecto” y bajo la “Objetos tecnológicos” expandir la estructura agregada y dar clic en “Puesta en servicio”.
16. En la parte superior izquierda de la ventana escoger el tiempo de muestreo para la curvas de consigna, entrada y salida, luego dar clic en “Start”
17. Asegurarse que el tanque este lleno de agua para que el parámetro entrada se encuentre lejos de la consigna impuesta que fue 500.
18. En la parte superior en “Modo de Ajuste” escoger “Optimización Inicial” y dar clic en “Start”, de esta manera el sistema calculo los parámetros PID en base a la respuesta del sistema a un escalón de salida.



Figura P3.10: Optimización Inicial

19. Una vez terminada la optimización se deberá cargar los parámetros obtenidos a la CPU, para ello en la parte inferior de la ventana dar clic en “Cargar Parámetros PID”.

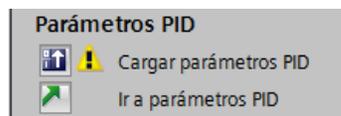


Figura P3.11: Cargar Parámetros PID

Los parámetros calculados pueden ser visualizados en la categoría “Parámetros PID” de la ventana de configuración.

20. Luego que el valor real alcanzó la consigna se deberá iniciar la optimización fina, para ello en la parte superior escoger “Sintonización fina” y luego dar clic en “Start”. Asegurarse de girar un poco la válvula de ingreso un de agua al tanque para que exista un pequeño flujo de entrada.

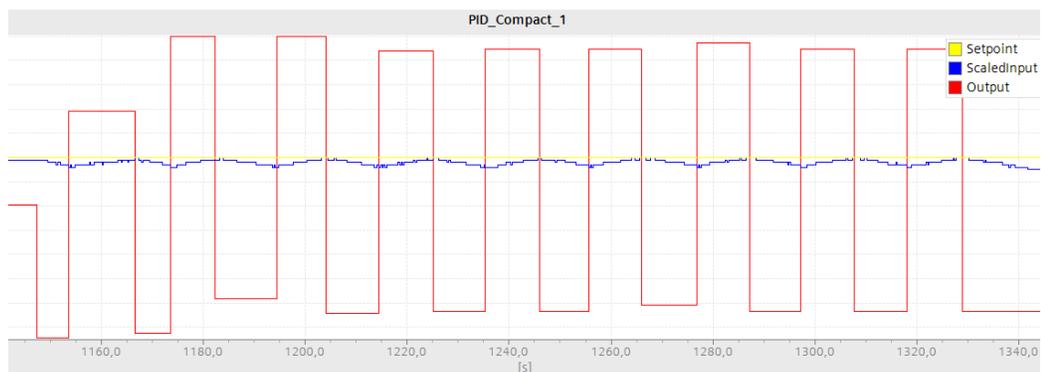


Figura P3.12: Optimización Fina

21. Una vez terminada la optimización repetir el paso 19 para cargar los parámetros al PLC.

**NOTA:** Existe la posibilidad de cambiar los parámetros PID manualmente con el fin de ajustar exactamente el comportamiento del sistema, esto se puede hacer mediante la categoría “Parámetros PID” de la ventana de configuración.

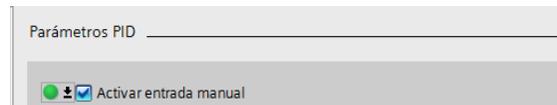


Figura P3.13: Entrada Manual para Parámetro PID

Luego de cambiar el valor de algún parámetro se deberá cargarlos nuevamente siguiendo el paso 19.

### **EJERCICIO PARA EL ALUMNO**

En base a lo revisado acerca del control PID el alumno deberá realizar un proyecto para el control PID del nivel del tanque con un funcionamiento manual y automático controlado desde una pantalla HMI.

El modo automático deberá ser activado por un botón y deberá permitir el ingreso de la consigna PID, por otra parte el modo manual se activará mediante otro botón y se ingresará en porcentaje PWM de la salida. En la pantalla HMI también se deberá mostrar un gráfico para la visualización del nivel del tanque con un campo de texto que indique su nivel.

El alumno deberá realizar las dos sintonizaciones en TIA PORTAL y posteriormente basado en la teoría del PID cambiar los parámetros: ganancia proporcional, tiempo de integración y tiempo derivativo.

Registrar el comportamiento del sistema ante los cambios de los parámetros y ajustarlos de tal manera que se alcance la mejor respuesta.

### **CONCLUSIONES:**

## ANEXO 4: PRÁCTICA 4

**TEMA:** Comunicación Profinet entre dos PLCs Simatic S7-1200 a través de instrucciones *Open User Communication*.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la comunicación de dos PLCs Simatic S7-1200 mediante una red Profinet usando las instrucciones de *Open User Communication*.

### OBJETIVOS ESPECÍFICOS:

- Familiarizarse con los parámetros de la red Profinet.
- Configurar una red de comunicación Profinet en TIA PORTAL.
- Familiarizarse con las instrucciones *Open User Communication* para una comunicación Profinet.
- Enviar datos de diferentes áreas de memoria de un controlador a otro usando el protocolo TCP.

### MATERIALES:

- Dos PLCs Simatic S7-1200.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.
- Un motor Trifásico

### FUNDAMENTO TEÓRICO:

#### PROFINET

Profinet es el resultado de la integración de las propiedades de Profibus y *Ethernet*, es un estándar que ofrece soluciones de red para procesos de automatización, la comunicación

Profinet se basa en protocolos *Ethernet*, UDP, TCP/IP (Guerrero, Yuste, & Martínez, 2009).

Profinet es un estándar desarrollado para cubrir todas las necesidades de la ingeniería de automatización comunicando equipos de todo tipo, tanto en niveles de planta como en niveles de gestión empresarial, permite la utilización de cables y *switches* estándares de *Ethernet*.

Características:

- Es un estándar abierto basado en *Industrial Ethernet*.
- Capaz de usar equipos tanto de *Industrial Ethernet* como de *Standard Ethernet*.
- Usa tecnología TCP/IP y Tecnologías de la Información.
- Integra los sistemas con los buses de campo.
- Gran potencia de transferencia con muchas estaciones.
- Alta disponibilidad gracias a redes redundantes.
- Capacidad de Tiempo real.
- Flexibilidad.
- Más rápido que los buses actuales especialmente para sistemas de control de movimiento.
- Existe gran variedad de productos disponibles en el mundo.

Profinet posee dos perfiles de comunicación: Profinet I/O para integrar dispositivos de campo, y Profinet-CBA para la comunicación de los sistemas de automatización distribuidos basadas en componentes.

Las redes Profinet básicamente utilizan tres medios de transmisión para sus comunicaciones: mediante señales eléctricas a través de un cable de par trenzado, mediante señales ópticas a través de cables de fibra óptica y por medio de ondas

electromagnéticas, mediante el uso de cables *Ethernet* se puede alcanzar velocidad de 100 Mb/s.

Profinet al estar basado en redes *Ethernet* utiliza su procedimiento de control de acceso al medio conocido como CSMA/CD y también CSMA/CA en las transmisiones inalámbricas.

Profinet implementa la estructura de la trama de datos de *Ethernet* y además agrega servicios para el manejo de prioridades en la trama y un modelo de actualización de datos a través de ventanas de tiempo, lo cual permite prestar los tres servicios de comunicación: de tiempo real, isócronas de tiempo real y no de tiempo real. En la configuración se generan las ventanas de tiempo en las cuales se agregan los datos a ser enviados (UNIOVI, S.A.).

En la Figura 1.22 se puede observar la trama de datos de Profinet.

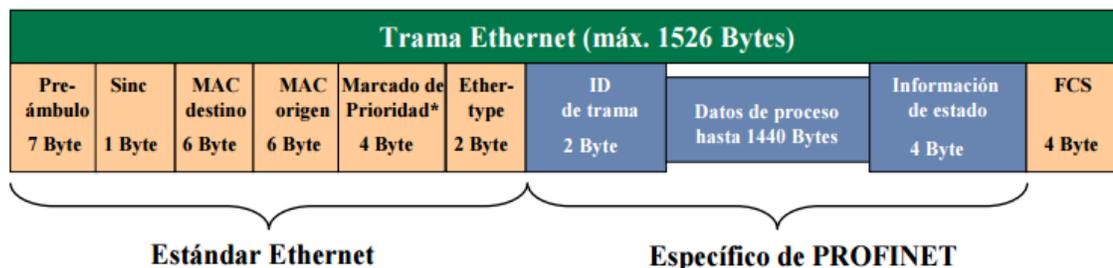


Figura P4.1: Trama de Datos en Profinet

Fuente: (UNIOVI, S.A.)

Las comunicaciones a través de las instrucciones *Open User Communications* permiten al puerto Profinet de la CPU intercambiar datos en redes con protocolos estándar del mundo informático como TCP, UDP e Iso sobre TCP, mediante estas instrucciones el usuario puede crear, deshacer y controlar canales de comunicación mediante IDs específicas, además de enviar y recibir datos.

**PROCEDIMIENTO:****a. Configuración del Proyecto y del Hardware**

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en “Abrir Vista del Proyecto”.
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.

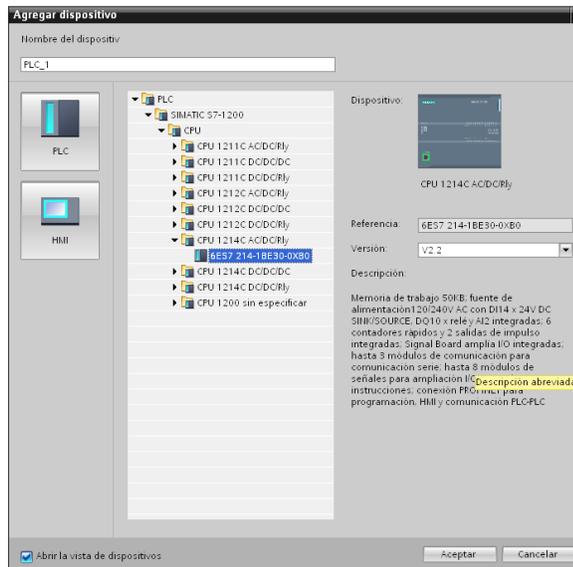


Figura P4.2: Agregar PLC1 al proyecto

5. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC2, en este caso será el mismo modelo CPU 1214 AC/DC/RLY.

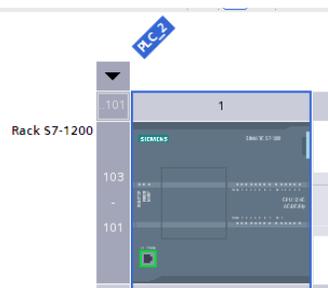


Figura P4.3: PLC2 agregado al proyecto

6. En el “Árbol de Proyecto” dar doble clic en “Dispositivos y Redes” para conectar los equipos mediante una red Profinet.

Posicionarse sobre el cuadro de borde amarillo en la esquina inferior izquierda de uno de los PLCs, luego dar clic y arrastrar hasta el otro PLC.

La conexión deberá quedar de la siguiente manera.

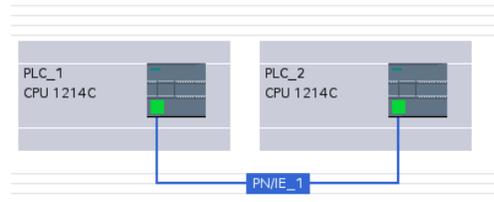


Figura P4.4: Conexión Profinet en TIA PORTAL

- Lo siguiente será configurar las direcciones IP para que los equipos puedan intercambiar datos en la red, para ello dentro del PLC1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC.

En la Ventana Propiedades que se despliega en la parte inferior dar clic sobre “Interfaz Profinet”.

Luego en la parte del Protocolo IP colocar:

Ip: 192.168.0.1

Mascara de Subred: 255.255.255.0

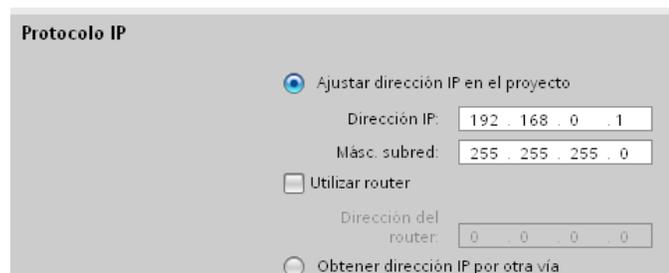


Figura P4.5: Configuración IP PLC1

- Repetir el paso 7 para el PLC2 y colocar los siguientes datos.

Ip: 192.168.0.4

Mascara de Subred: 255.255.255.0



Figura P4.6: Configuración IP PLC2

9. Para la señal de activación de envío y recepción automática de datos se utilizará un flaco de subida determinado por un reloj interno del PLC.  
Para ello dentro del PLC1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC.  
En la Ventana Propiedades dar clic sobre “Marcas de sistema y de ciclo”. Luego activar la casilla de verificación para bits de marcas de ciclo.

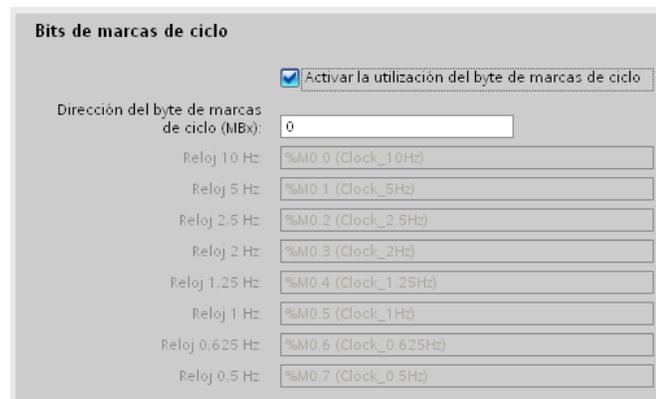


Figura P4.7: Activar Marcas de Ciclo del PLC

10. Repetir el paso anterior para el PLC2.

#### **b. Programación PLCs para la Comunicación, Configuración de las Instrucciones**

11. En la parte derecha de TIA PORTAL en el “Árbol de Instrucciones” en la sección “Comunicación”, agregar al bloque MAIN la estructura TSEND\_C.  
Aceptar el aviso para la creación de un nuevo bloque de datos, el mismo que contendrá las variables de la estructura.

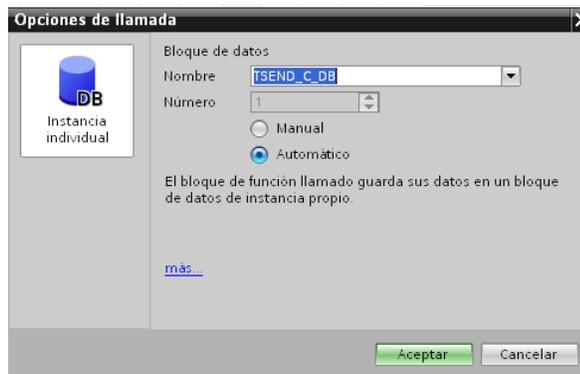


Figura P4.8: Creación Bloque de datos TSEND\_C

12. Una vez agregada la instrucción dar sobre clic sobre la misma y en la Pestaña “Configuración” y luego en la parte “Parámetros de la Conexión” elegir lo siguiente:

- Interlocutor: PLC2
- Datos de Conexión: Se crea automáticamente un DB, sino desplegar y elegir nuevo.
- ID de conexión: 1
- Botón de Opción-Establecimiento activo de la conexión: Marcado para el PLC1

Por el momento se configura solo lo mencionado, quedará de la siguiente manera:

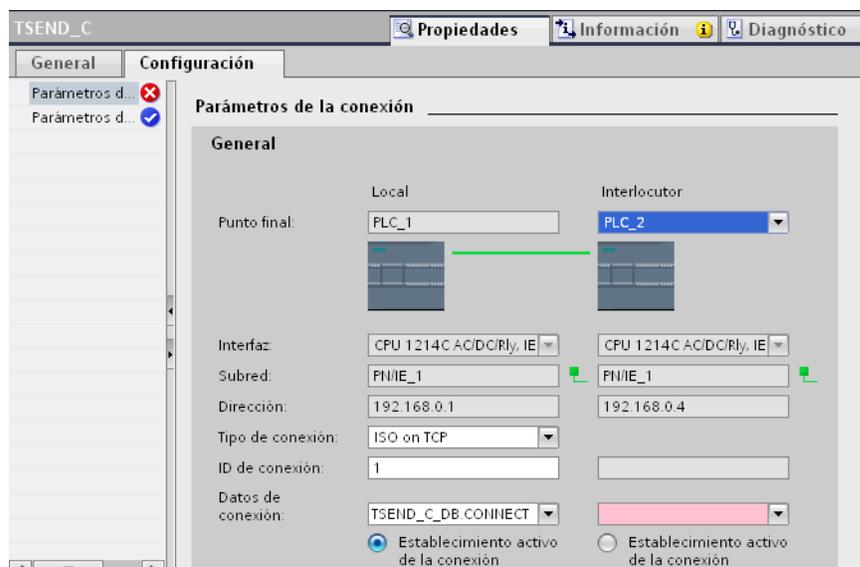


Figura P4.9: Parámetros de Conexión TSEND\_C

13. Similar al paso 11 agregar la instrucción TRCV\_C.



Figura P4.10: Creación Bloque de datos TSEND\_C

14. Una vez agregada la instrucción dar sobre clic sobre la misma y en la Pestaña “Configuración” y luego en la parte “Parámetros de la Conexión” elegir lo siguiente:

- Interlocutor: PLC1
- Datos de Conexión: Se crea automáticamente un DB, sino desplegar y elegir nuevo.
- ID de conexión: 1
- Botón de Opción-Establecimiento activo de la conexión: Marcado para el PLC1.

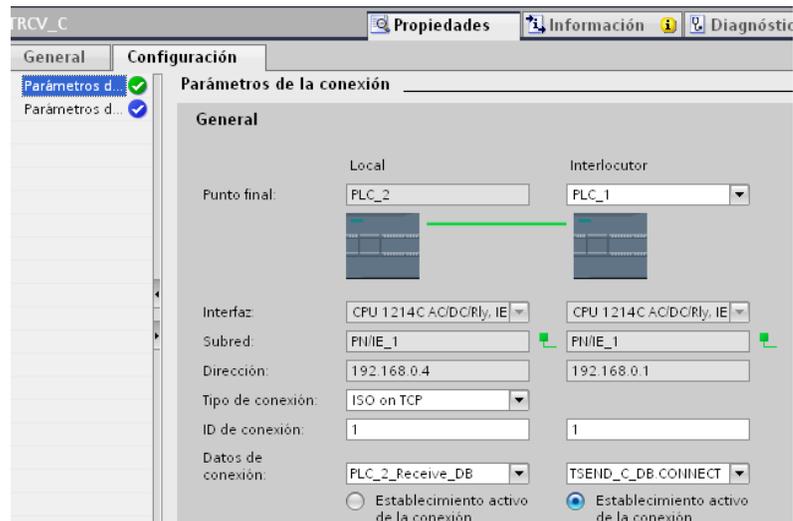


Figura P4.11: Parámetros de Conexión TRCV\_C

15. Regresar a la configuración de la Instrucción TSEND\_C del PLC1 y seleccionar en “Datos de Conexión”, el DB que quedo pendiente, correspondiente al PLC\_2\_Receive\_DB.



Figura P4.12: Datos de Conexión

16. Luego dar clic sobre “Parámetros de Bloque” dentro de la Configuración de TSEND\_C, ahí escoger lo siguiente:  
 REQ: M0.0 (Corresponde a la entrada para activar el envío de datos mediante el reloj de 10Hz.)  
 CON: 1 (Parámetro de configuración para controlar y establecer la comunicación).

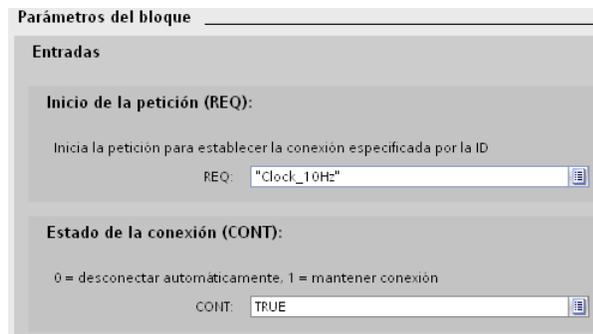


Figura P4.13: Parámetros REQ y CON de las Instrucciones

17. Repetir el paso 16 para el PLC2.

### c. Envío de datos de tipo byte del PLC1 al PLC2 (Entradas Digitales)

18. En el parámetro DATA del PLC1 se debe elegir tres configuraciones: el inicio, la longitud y el tipo de unidad de datos que se desee enviar.  
 La primera prueba será enviar las entradas del byte I1 del PLC1, por lo tanto estos campos quedarán de la siguiente manera.

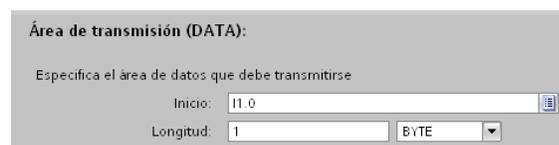


Figura P4.14: Parámetro DATA en TSEND\_C, enviar 1 Byte

Por lo tanto la instrucción deberá quedar de la siguiente manera.

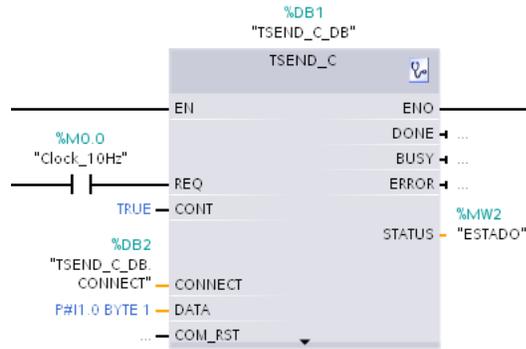


Figura P4.15: Instrucción TSEND\_C

19. En el PLC2 se deberá especificar en el parámetro DATA en que área de memoria se recibirá la información, en este caso se guardará en el byte M20.0. Por lo tanto repetir el paso 18 para el PLC2 con la información adecuada.

**Área de recepción (DATA):**

Especifica el área de datos que debe recibirse

Inicio:

Longitud:

Figura P4.16: Parámetro DATA en TRCV\_C, enviar 1 Byte

La instrucción deberá quedar de la siguiente manera.

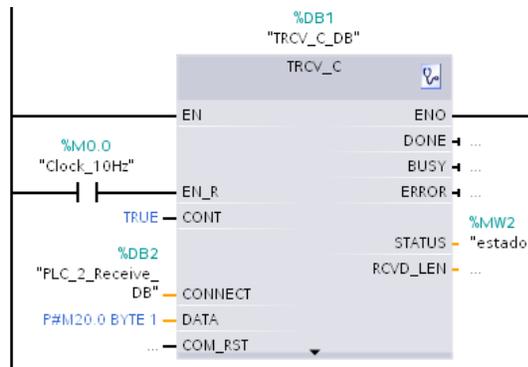


Figura P4.17: Instrucción TRCV\_C

20. En el PLC2 se deberá agregar los siguientes bloques.

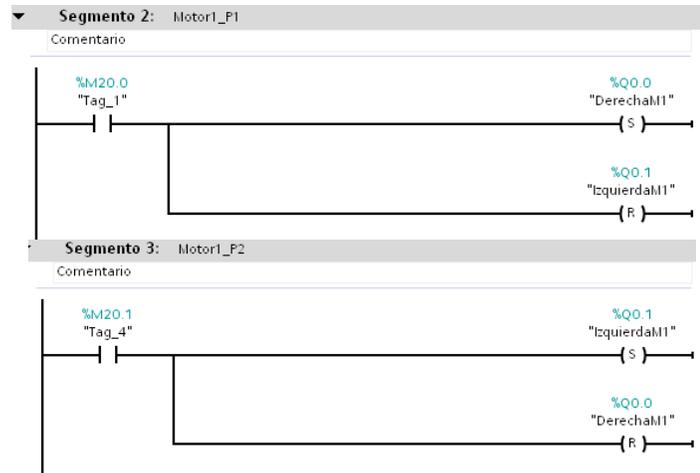


Figura P4.18: Bloques de Datos PLC2 233

21. Repetir el paso 20 para asignar las salidas al motor 2.
22. Grabar ambos PLCs.
23. Verificar el funcionamiento interactuando con los pulsantes.

**d. Envío de datos de tipo Char del PLC1 al PLC2**

24. Como se explicó, los canales por los que se intercambia la información se identifican mediante IDs de conexión, de esta manera se puede agregar más instrucciones TSEND\_C y TRCV\_C para comunicaciones simultáneas.

Por lo tanto se deberá repetir los pasos del 11 al 17 parar agregar nuevas instrucciones al proyecto con nuevos bloques de datos y nuevas IDs.

Deberá quedar de la siguiente manera.

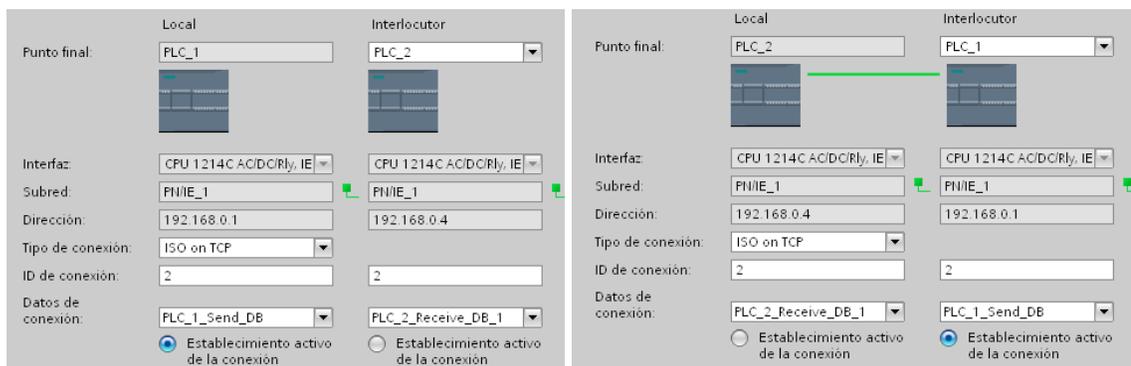


Figura P4.19: Configuración canal 2

25. En el PLC1 crear una nueva variable de tipo *Char* ubicada en algún espacio de memoria, en este caso será MB30.

26. En el parámetro DATA del PLC1 se deberá elegir 1 carácter de tipo *Char* correspondiente a la variable creada, por lo tanto estos campos quedarán de la siguiente manera.

Área de transmisión (DATA):  
Especifica el área de datos que debe transmitirse  
Inicio: M30.0  
Longitud: 1 CHAR

Figura P4.20: Parámetro DATA en TSEND\_C, enviar 1 *Char*

27. En el PLC2 se deberá especificar en el parámetro DATA en que área de memoria se recibirá la información, en este caso se guardará en el byte MB30. Por lo tanto repetir el paso 26 para el PLC2 con la información adecuada.

28. En el PLC1 se deberá agregar los siguientes bloques.

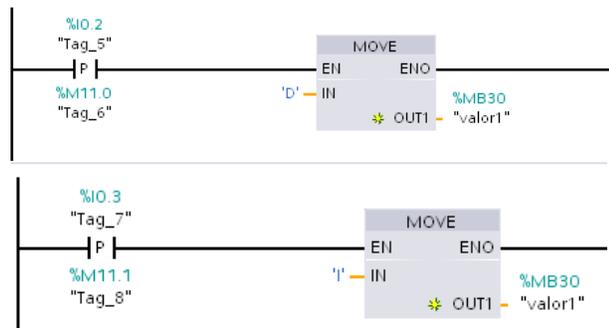


Figura P4.21: Bloques de Datos, envió *Char*

29. En el PCL1 agregar dos bloques de datos para la comprobación del dato recibido, si es “D” gira a la derecha y si es “I” gira a la izquierda.
30. Grabar ambos PLCs.
31. Verificar el funcionamiento interactuando con los pulsantes.

## EJERCICIO PARA EL ALUMNO

En base a lo revisado acerca de la comunicación Profinet a través de instrucciones *Open User Communication*, el alumno deberá crear un proyecto que permita la activación, inversión de giro y paro de un motor trifásico conectado en el PLC interlocutor.

El estudiante deberá comprobar que salidas físicas del interlocutor activan los contactores colocado en el banco de trabajo, luego conectar el motor a dichos elementos y realizar el programa para el control del motor desde el otro PLC a través de tres entradas:

- I1.0: Gira a la derecha
- I1.1: Gira a la izquierda
- I1.2: Paro

**CONCLUSIONES:**

## ANEXO 5: PRÁCTICA 5

**TEMA:** Comunicación Profinet entre dos PLCs Simatic S7-1200 para el intercambio de un bloque de datos.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la comunicación de dos PLCs Simatic S7-1200 mediante una red Profinet para transmitir un bloque de datos.

### OBJETIVOS ESPECÍFICOS:

- Familiarizarse con los parámetros de la red Profinet.
- Configurar una red de comunicación Profinet en TIA PORTAL.
- Configurar dos pantallas HMI para la conexión con los controladores.
- Enviar la información de un bloque de datos de un controlador a otro usando el protocolo TCP.
- Visualizar en las pantallas HMI la información enviada y recibida.

### MATERIALES:

- Dos PLCs Simatic S7-1200.
- Dos Pantallas HMI.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.
- Un potenciómetro.

### FUNDAMENTO TEÓRICO:

#### PROFINET

Profinet es el resultado de la integración de las propiedades de Profibus y *Ethernet*, es un estándar que ofrece soluciones de red para procesos de automatización, la comunicación

Profinet se basa en protocolos *Ethernet*, UDP, TCP/IP (Guerrero, Yuste, & Martínez, 2009).

Profinet es un estándar desarrollado para cubrir todas las necesidades de la ingeniería de automatización comunicando equipos de todo tipo, tanto en niveles de planta como en niveles de gestión empresarial, permite la utilización de cables y *switches* estándares de *Ethernet*.

Características:

- Es un estándar abierto basado en *Industrial Ethernet*.
- Capaz de usar equipos tanto de *Industrial Ethernet* como de *Standard Ethernet*.
- Usa tecnología TCP/IP y Tecnologías de la Información.
- Integra los sistemas con los buses de campo.
- Gran potencia de transferencia con muchas estaciones.
- Alta disponibilidad gracias a redes redundantes.
- Capacidad de Tiempo real.
- Flexibilidad.
- Más rápido que los buses actuales especialmente para sistemas de control de movimiento.
- Existe gran variedad de productos disponibles en el mundo.

Profinet posee dos perfiles de comunicación: Profinet I/O para integrar dispositivos de campo, y Profinet-CBA para la comunicación de los sistemas de automatización distribuidos basadas en componentes.

Las redes Profinet básicamente utilizan tres medios de transmisión para sus comunicaciones: mediante señales eléctricas a través de un cable de par trenzado, mediante señales ópticas a través de cables de fibra óptica y por medio de ondas

electromagnéticas, mediante el uso de cables *Ethernet* se puede alcanzar velocidad de 100 Mb/s.

Profinet al estar basado en redes *Ethernet* utiliza su procedimiento de control de acceso al medio conocido como CSMA/CD y también CSMA/CA en las transmisiones inalámbricas.

Profinet implementa la estructura de la trama de datos de *Ethernet* y además agrega servicios para el manejo de prioridades en la trama y un modelo de actualización de datos a través de ventanas de tiempo, lo cual permite prestar los tres servicios de comunicación: de tiempo real, isócronas de tiempo real y no de tiempo real. En la configuración se generan las ventanas de tiempo en las cuales se agregan los datos a ser enviados (UNIOVI, S.A.).

En la Figura 1.22 se puede observar la trama de datos de Profinet.

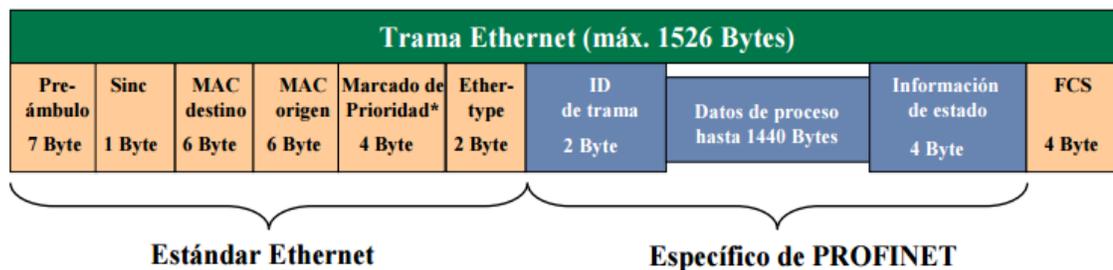


Figura P5.1: Trama de Datos en Profinet

Fuente: (UNIOVI, S.A.)

Mediante una red Profinet se puede establecer comunicaciones optimizadas en las cuales se pueden enviar bloques de datos a través del uso de variables *struct* que engloban diferentes tipos de datos.

**PROCEDIMIENTO:****a. Configuración del Proyecto y del Hardware**

1-17. Repetir los pasos del 1 al 10 de la Práctica 3 para agregar los PLCs al proyecto y configurar la red Profinet.

18. Repetir los pasos del 5-10 de la Práctica 2 para agregar una Pantalla HMI y vincularla al PLC1.

19. Nuevamente repetir los pasos del 5-10 de la Práctica 2 para vincular una pantalla HMI al PLC2.

**b. Configuración de los Bloques de Datos para el envío de información de un PLC al otro**

20. En el árbol del proyecto, dentro del PLC1 y bajo la sección “Bloques de Programa” dar doble clic en “Agregar nuevo bloque”.

Luego en la parte izquierda dar clic en la figura “Bloque de datos”, escribir un nombre (en este caso *DatosPLC1*) y dar clic en aceptar.

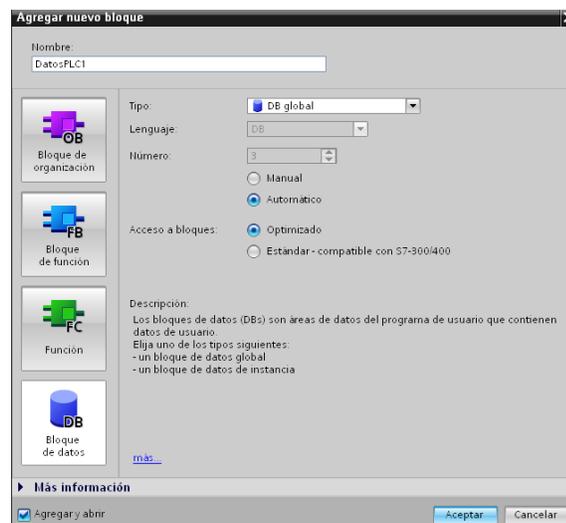


Figura P5.2: Agregar Bloque de Datos

21. Dentro del bloque de datos se deberá crear una variable de tipo *Struct* para agregar la información a enviar dentro de la misma.

Nombre: VariablePL1.

Tipo: *Struct*

DatosPLC1			
	Nombre	Tipo de datos	Valor de a
1	Static		
2	VariablesPLC1	Strucj	false
3	<agregar>	Strucj	

Figura P5.3: Crear Variable de tipo *Struct*

22. Dentro de VariablePLC1 agregar todos los datos que se deseen enviar, en este caso se empezará por agregar una dato de tipo *Int*. El bloque de datos quedará de la siguiente manera.

DatosPLC1		
	Nombre	Tipo de datos
1	Static	
2	VariablesPLC1	Struct
3	Vint	Int

Figura P5.4 Variables a enviar

23. En el PLC2 repetir los pasos del 25 al 27 para agregar un nuevo bloque datos con la variables pertinente, el bloque deberá contener tipos de datos coherentes para la recepción de la información del bloque del PLC1.

24. En el parámetro DATA de ambos PLCs seleccionar el bloque de datos creado y la variable *Struct*, los bloques TSEND\_C y TRCV\_C quedarán de la siguiente manera.

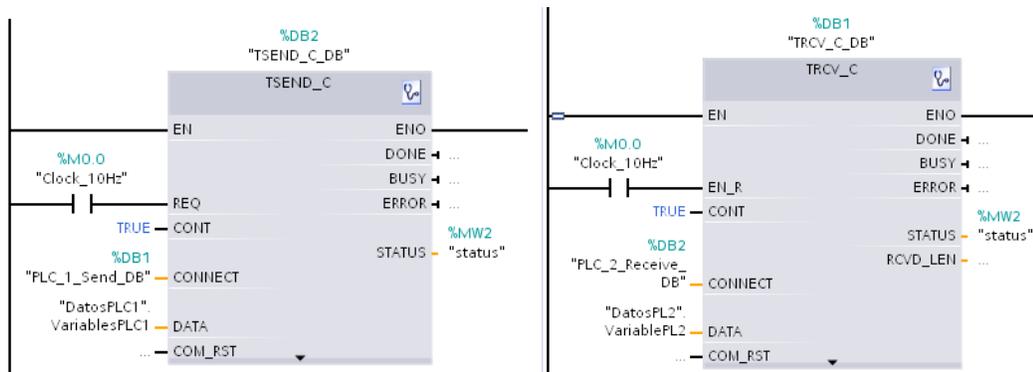


Figura P5.5: Bloques TSEND\_C y TRCV\_C

### c. Programación pantallas HMIs y Main

- Pantalla PLC1 y Main PCL1

25. Agregar un campo de entrada-salida para el ingreso de información de la variable de tipo *int*.

Luego en la Pestaña “Eventos” seleccionar el de tipo “Activar” y asignar la función “Definir variable”.

Luego en los campos “Variable (Salida)” y “Valor” escoger la variable Vint creada en el bloque de datos.

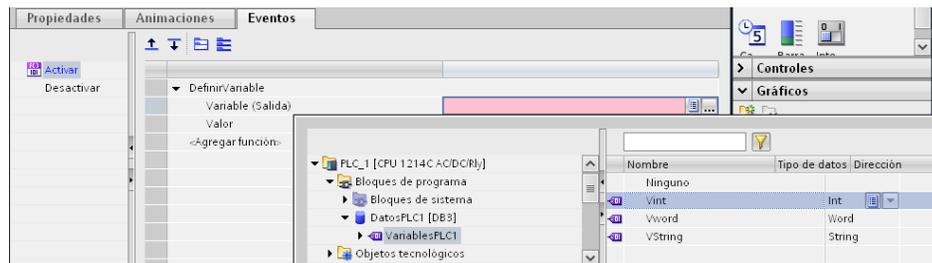


Figura P5.6: Configuración Campo de Texto para ingresar datos

A continuación en la Pestaña “General” y en el campo “Variable” escoger nuevamente Vint.

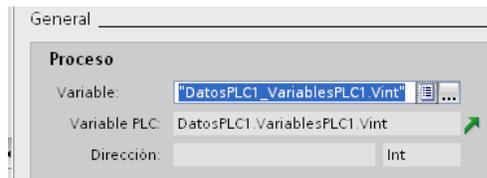


Figura P5.7: Configuración Campo de Texto para visualizar datos

26. Insertar los textos pertinentes y distribuir los elementos en la pantalla HMI, deberán quedar de la siguiente forma.



Figura P5.8: Pantalla PCL1

- Pantalla PLC2

27. Repetir el paso 30 para agregar un campo de entrada-salida y asignarlo a la variable de proceso del bloque de datos que recibe la información.

28. Repetir el paso 31, la pantalla deberá quedar de la siguiente manera.

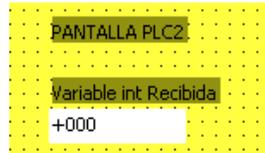


Figura P5.9: Pantalla PCL2

#### d. Funcionamiento

29. Grabar los cuatro equipos conectados a la red.

30. Comprobar el funcionamiento de la aplicación introduciendo datos para la variable de tipo *int* y observando su correcta recepción en el otro equipo.

### EJERCICIO PARA EL ALUMNO

En base a lo explicado acerca del envío de información de un bloque de datos vía Profinet, el alumno deberá crear un proyecto para transmisión de diferentes tipos de variables de un PLC a otro con las siguientes consideraciones:

- A los bloques de datos incluidos en los PLCs agregar dos variables para el envío de datos con coma flotante y texto.
- El dato en coma flotante a enviar deberá ser el valor de tensión de la entrada analógica escalada de 0 a 10V, se deberá visualizar en ambas pantallas enlazadas a cada PLC.
- El texto deberá a enviar deberá tener una longitud de máximo 20 caracteres y deberá ser ingresado en la pantalla del PLC1 y ser visualizado en la pantalla del PLC2.

Se debe conectar un potenciómetro en la entrada analógica y mediante el siguiente segmento se puede normalizar y escalar dicha variable.

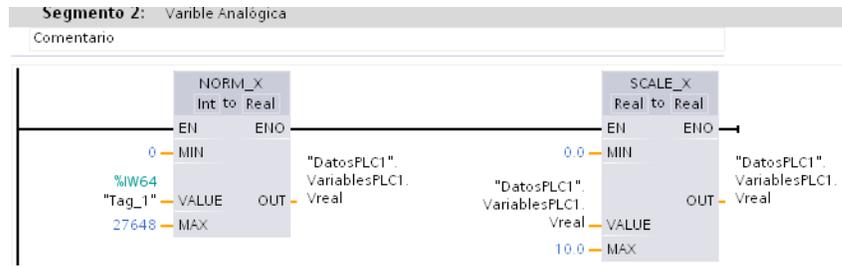


Figura P5.10: Escalado de Variable Analógica

**CONCLUSIONES:**

## ANEXO 6: PRÁCTICA 6

**TEMA:** Comunicación Profibus-DP entre dos PLCs Simatic S7-1200 con el modelo maestro-esclavo.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la comunicación de dos PLCs Simatic S7-1200 mediante una red Profibus-DP.

### OBJETIVOS ESPECÍFICOS:

- Familiarizarse con los parámetros de la red Profibus.
- Configurar una red de comunicación Profibus en TIA PORTAL.
- Configurar los módulos de comunicación maestro y esclavo DP.
- Enviar información desde el maestro al esclavo, datos de tipo Byte y Word.
- Leer información del esclavo, datos de tipo Byte y Word.

### MATERIALES:

- Dos PLCs Simatic S7-1200.
- *Software* TIA PORTAL.
- Un cable Profibus 2 hilos, conectores interfaz RS-485
- Módulo Maestro-DP Profibus CM 1243-5
- Módulo Esclavo-DP Profibus CM 1242-5
- Un motor DC.
- Un Mosfet.
- Un potenciómetro.

## FUNDAMENTO TEÓRICO:

### PROFIBUS

Profibus (*PROcess Field BUS*, “Bus de Proceso de Campo”) es un conjunto de especificaciones de red que ha dado lugar a un bus de campo de tecnología abierta y transparente para la comunicación digital de dispositivos dentro de los procesos industriales, tanto de fabricación como procesos distribuidos (Guerrero, Yuste, & Martínez, 2009).

Profibus es un estándar desarrollado para dar solución a los problemas de conectividad de instrumentos de campo como: periferia descentralizada, válvulas, *drivers* de motores o accionamientos en general, etc. con los sistemas de automatización.

Profibus posee tres variantes del estándar las cuales son: Profibus-FMS, Profibus-DP y Profibus-PA enfocadas a distintas áreas de la pirámide CIM. Está basado en el Modelo OSI aunque implementa solo tres capas del mismo con el fin de facilitar la comunicación, las capas sobre las cuales se desarrolla Profibus son: la capa física, de enlace y de aplicación.

Las redes Profibus son de tipo principal-subordinado o maestro-esclavo, es así que se puede diferenciar dos tipos de estaciones de comunicación con diferentes funciones e importancia desde el punto de vista de la trasmisión de información, según lo expuesto por (Mandado et al., 2006) se puede clasificar en:

- **Procesadores Principales:** Estas estaciones son conocidas como activas y son capaces de gestionar las comunicaciones, es decir, enviar y recibir datos por iniciativa propia hacia los demás elementos.

- Procesadores Subordinados: Estas estaciones son conocidas como pasivas ya que solo pueden transmitir sus datos si el procesador principal lo autorizada, mientras se encuentre este activo.

Si en la red Profibus existe más de un maestro, para evitar que dos procesadores tomen el control se utiliza un mecanismo de acceso al medio llamado *token bus*, en el cual un mensaje testigo circula entre los dispositivos maestros de la red y mediante el cual se da aviso a cada controlador que puede efectuar sus comunicaciones durante un tiempo determinado.

Las redes Profibus básicamente utilizan dos medios de transmisión para sus comunicaciones: mediante señales eléctricas a través de un cable de par trenzado según la norma IEA RS-485 o también mediante señales ópticas a través de cables de fibra óptica, en un bus lineal se puede alcanzar velocidades de transmisión entre 9.6 Kbits/s y 12 Mbits/s.

Profibus-DP (Periferia Descentralizada) es un perfil de comunicación optimizado para su uso a nivel de campo, es decir, está desarrollado para la transferencia de información entre dispositivos de proceso a altas velocidades y bajo costo, se caracteriza por ser un perfil que maneja las comunicaciones de manera determinística permitiendo una alta velocidad de transmisión en tiempos conocidos (aproximadamente 10 ms).

Los campos de datos que se tiene en la trama Profibus son los siguientes:

- Carácter DA (*Destination Address*, “Dirección de Destino”).
- Carácter SA (*Source Address*, “Dirección del Emisor”).
- Carácter FC (*Frame Control*, “Control de Mensaje”).
- Estructura FCS (*Frame Check Structure*, “Estructura de Comprobación de Mensaje”).

La trama de datos admite 3 tipos de formatos: de longitud fija sin datos, de longitud fija con datos y de longitud variable

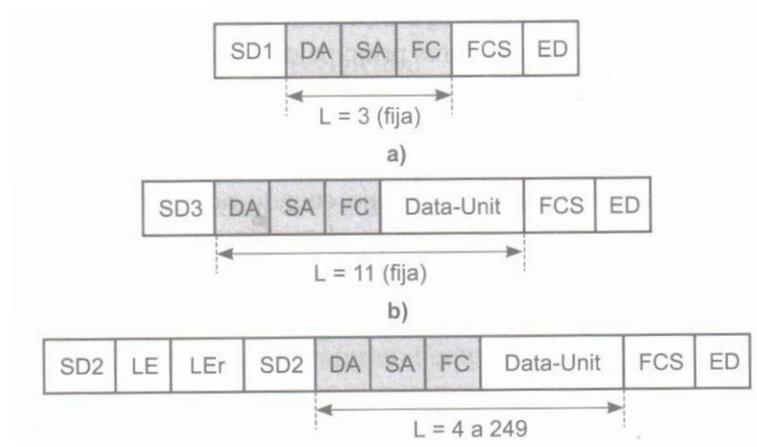


Figura P6.1: Trama de Datos Profibus 247

Fuente: (Mandado, Marcos, Pérez, Fernández, & Armesto, 2006)

## PROCEDIMIENTO:

### a. Configuración del Proyecto y del *Hardware*

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir "Crear Proyecto" y llenar los campos "Nombre" y "Ruta" según se desee.
3. Dar clic en "Abrir Vista del Proyecto".
4. En el "Árbol de Proyecto" dar doble clic en "Agregar Dispositivos" para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.
5. En el "Árbol de Proyecto" dar doble clic en "Agregar Dispositivos" para insertar el PLC2, en este caso será el modelo CPU 1214 AC/DC/RLY.
6. En el "Árbol de Proyecto" dar clic derecho sobre el PLC1 y seleccionar "Cambiar Nombre", el PLC1 se llamará MAESTRO.

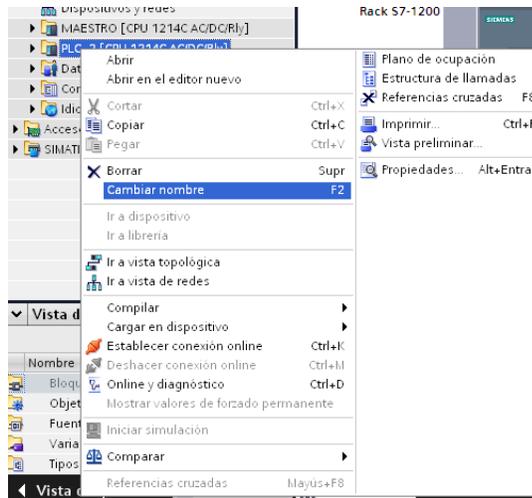


Figura P6.2: Cambiar Nombre PLC1 P5

7. Repetir el paso 6 para cambiar el nombre del PLC2 a ESCLAVO.

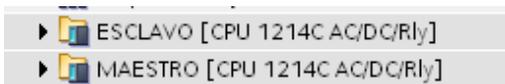


Figura P6.3: Nombres PLCs P5

8. En el PLC MAESTRO dar doble clic sobre “Configuración de Dispositivos”, luego en la pestaña derecha “Catálogo de *Hardware*” expandir la categoría “Módulos de Comunicación”, luego la subcategoría “Profibus” y finalmente “CM 1243-5”. Luego arrastrar el elemento hasta el *slot* ubicado en la parte izquierda del PCL.

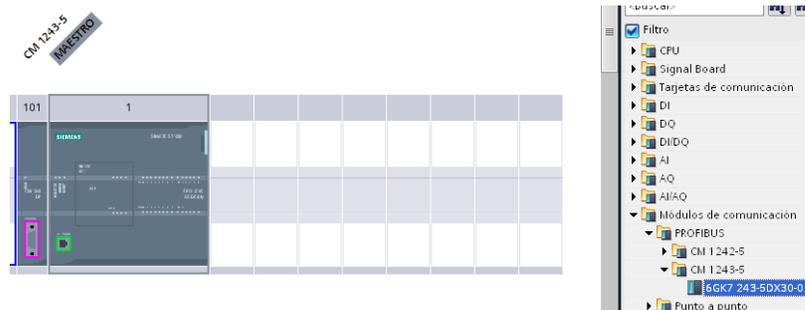


Figura P6.4: Agregar Módulo Maestro CM 1243-5

9. En el PLC ESCLAVO dar doble clic sobre “Configuración de Dispositivos”, luego en la pestaña derecha “Catálogo de *Hardware*” expandir la categoría “Módulos de Comunicación”, luego la subcategoría “Profibus” y finalmente “CM 1242-5”. Luego arrastrar el elemento hasta el *slot* ubicado en la parte izquierda del PCL.

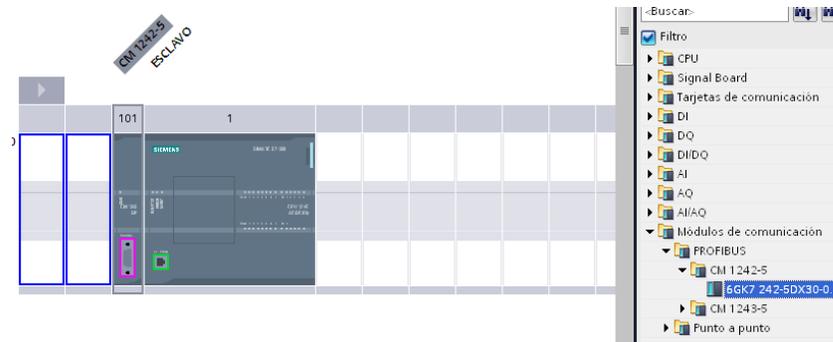


Figura P6.5: Agregar Módulo Esclavo CM 1242-5

10. En el “Árbol de Proyecto” dar doble clic en “Dispositivos y Redes” para conectar los equipos mediante una red Profibus.

Posicionarse sobre el cuadro de borde azul en la esquina inferior izquierda de uno de los módulos Profibus, luego dar clic y arrastrar hasta el módulo del segundo PLC. La conexión deberá quedar de la siguiente manera.



Figura P6.6: Red Profibus P5

11. Una vez creada la conexión lógica Profibus se deberá configurar los parámetros del bus, para ellos dar doble clic en el cable punteado que se dibujó entre los PLCs con el nombre “MAESTRO.DP-Mastersystem”, escoger lo siguiente:

Dirección PROFIBUS más alta: 126

Velocidad de Transferencia: 1.5Mbits/s.

Perfil: DP.

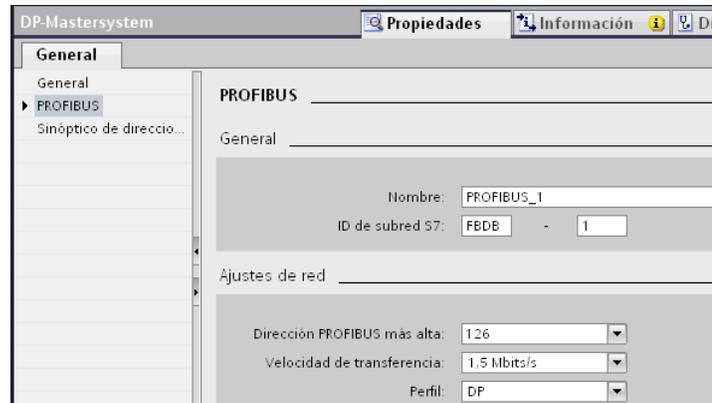


Figura P6.7: Configuraciones de Bus Profibus

12. En el PLC MAESTRO dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el módulo de comunicación.

En la Ventana Propiedades dar clic sobre “Interfaz DP”. Luego en la categoría “Parámetros” asignarle una dirección Profibus al dispositivo, en este caso será la número 2.



Figura P6.8: Dirección Profibus Maestro

13. Repetir el paso 12 para asignar una dirección Profibus al PLC ESCLAVO, en este caso será la número 3.



Figura P6.9: Dirección Profibus Esclavo

14. Luego en la categoría “Modo de Operación” verificar que la casilla “Esclavo DP” este seleccionada y que tenga un Maestro DP asignado. (El maestro DP se asignó automáticamente al crear la conexión entre los dos módulos.)

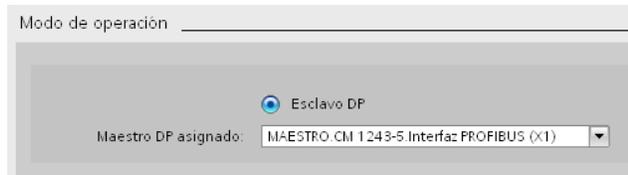


Figura P6.10: Modo Esclavo DP

15. Grabar ambos PLCs y verificar que los módulos de maestro y esclavo no se encuentren en un estado de error.

### b. Configuración Áreas de Transferencia

16. En el ESCLAVO en la Ventana de Propiedades expandir la categoría “Modo de Operación y ubicarse en la sección “Áreas de Transferencia”.

Luego dar clic en “Agregar nuevo” y establecer un nombre, en este caso la primera AT\_TxByte.

Áreas de transferencia							
	...	Área de transferencia	Tipo	Dirección del maestro	↔ Dirección del esclavo	Longitud	Coherencia
1		AT_TxByte	MS	Q 10	→ I 10	1 Byte	Unidad

Figura P6.11: Crear Área de Transferencia Profibus

17. En la parte izquierda bajo la categoría “Comunicación de I-Slave” se agregan las áreas de transferencia creadas, por lo tanto dar clic sobre el nombre AT\_TxByte para configurar los parámetros.

Tipo de dirección: Escoger si es de entrada o salida (Q o I), en este caso será Q.

Unidad: Escoger el tipo de datos que se desee enviar (*Byte* o *Palabra*), en este caso será *Byte*.

**Detalles del área de transferencia**

Área de transferencia: AT\_TxByte

Tipo de área de transferencia: MS

Interlocutor:  Local: 

Intercambio de datos entre: CM 1243-5 y CM 1242-5

Slot: 1

Tipo de dirección: Q

Dirección inicial: 10

Memoria imagen de proceso: MIP ciclica

Longitud: 1

Unidad: Byte

Figura P6.12: Área de Transferencia 1

18. Repetir el paso 16 y 17 para crear un Área de Transferencia de datos de tipo Word del Maestro al Esclavo.

**Detalles del área de transferencia**

Área de transferencia: AT\_TxWord

Tipo de área de transferencia: MS

Interlocutor:  Local: 

Intercambio de datos entre: CM 1243-5 y CM 1242-5

Slot: 2

Tipo de dirección: Q

Dirección inicial: 100

Memoria imagen de proceso: MIP ciclica

Longitud: 1

Unidad: Palabra

Figura P6.13: Área de Transferencia 2

19. Repetir el paso 16 y 17 para crear dos Áreas de Transferencia de datos de tipo Byte y Word del Esclavo al Maestro.

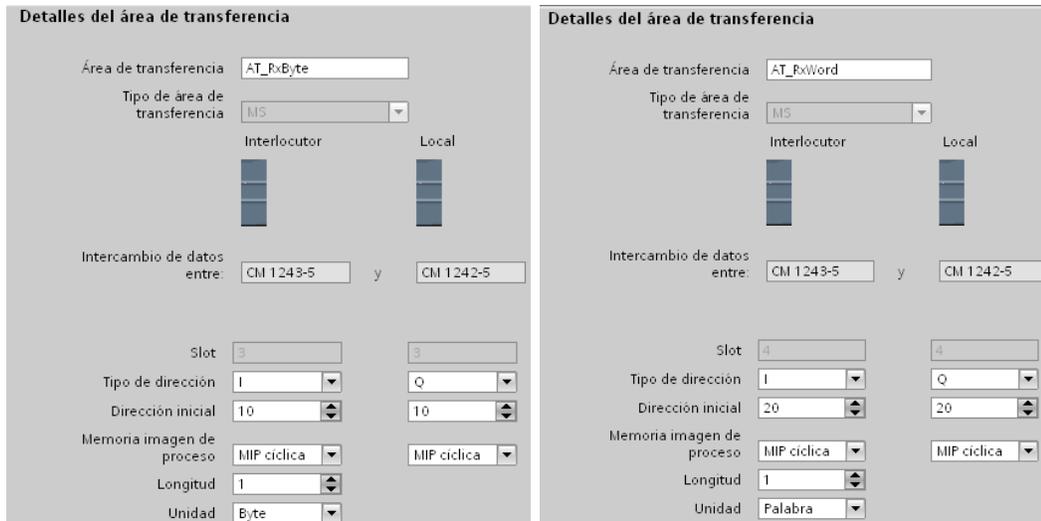


Figura P6.14: Área de Transferencia 3 y 4

20. Verificar que la dirección de la transferencia de datos tenga coherencia con las áreas de memoria escogidas para el maestro y esclavo, la dirección deberá ser de un área de salida hacia una de entrada.

Áreas de transferencia							
...	Área de transferencia	Tipo	Dirección del maestro	↔	Dirección del esclavo	Longitud	Coherencia
1	AT_TxByte	MS	Q 10	→	I 10	1 Byte	Unidad
2	AT_TxWord	MS	Q 100...101	→	I 100...101	1 Palabra	Unidad
3	AT_RxByte	MS	I 10	←	Q 10	1 Byte	Unidad
4	AT_RxWord	MS	I 20...21	←	Q 20...21	1 Palabra	Unidad
5	<Agregar nuevo>						

Figura P6.15: Áreas de Transferencias

**c. Envío de un Byte desde el Maestro**

21. La primera Área de transferencia permite enlazar un área de memoria de salida de tipo byte del maestro con un área de entrada byte del esclavo, los niveles binarios de los bits a enviar se establecen en el maestro mediante el siguiente segmento.

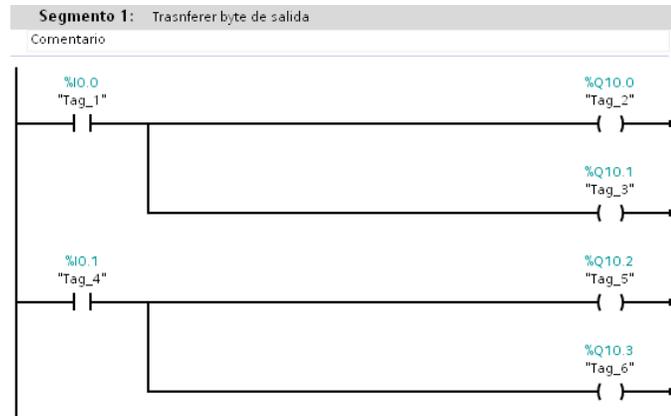


Figura P6.16: Segmento 1 Maestro P5

22. En el esclavo se agrega el segmento adecuado para utilizar los datos recibidos con el fin de encender las salidas digitales físicas.

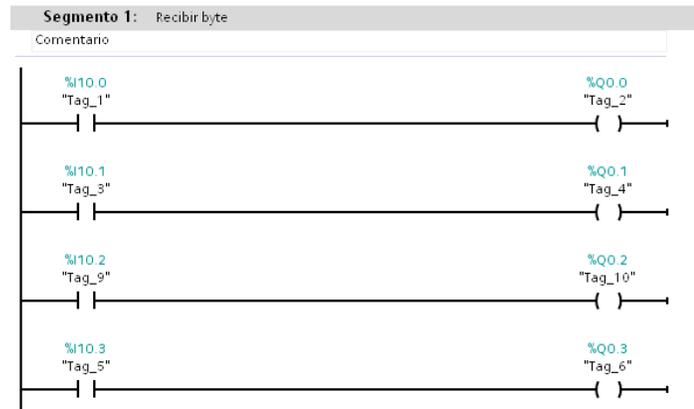


Figura P6.17: Segmento 1 Esclavo P5

23. Verificar el funcionamiento de la primera prueba interactuando con los pulsantes I0.0 e I0.1 del maestro y comprobando el encendido de las salidas del esclavo.

#### d. Envío de un Byte desde el Esclavo

24. La tercera Área de transferencia permite enlazar un área de memoria de salida de tipo byte del esclavo con un área de entrada byte del maestro, los niveles binarios de los bits a enviar se establecen en el esclavo mediante el siguiente segmento.

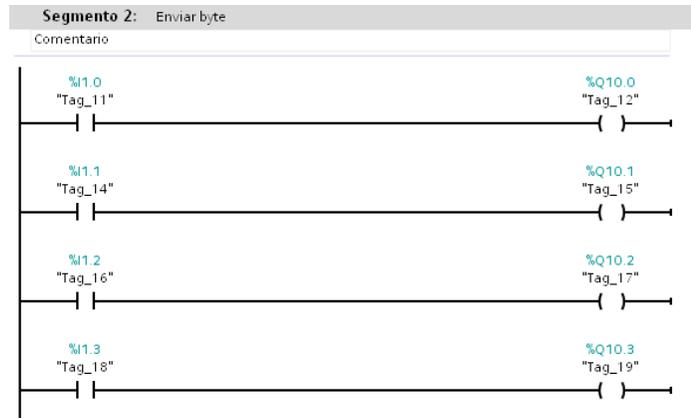


Figura P6.18: Segmento 2 Esclavo P5

25. En el maestro se agrega el segmento adecuado para utilizar los datos recibidos con el fin de encender las salidas digitales físicas.

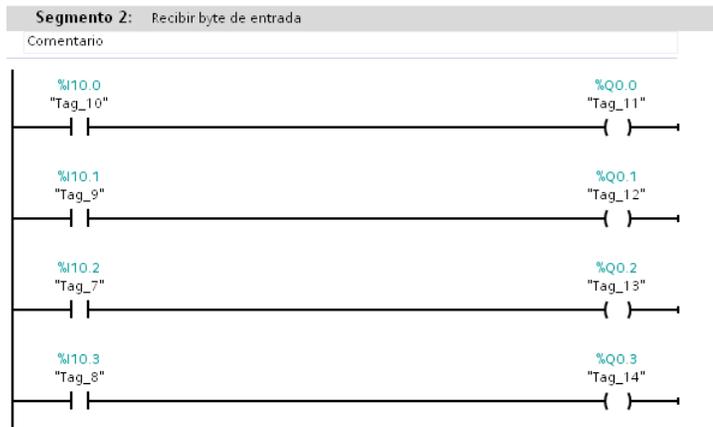


Figura P6.19: Segmento 2 Maestro P5

26. Verificar el funcionamiento de la segunda prueba interactuando con los pulsantes I1.0, I1.1, I1.2 e I1.3 del esclavo y comprobando el encendido de las salidas del maestro.

#### e. Envío de una Palabra (Word) desde el Maestro

27. Agregar la tarjeta correspondiente a una salida analógica en el PLC ESCLAVO con el fin de escribir dicho valor desde el maestro.

Dar doble clic sobre "Configuración de Dispositivos", luego en la pestaña derecha "Catálogo de Hardware" expandir la categoría "Signal Board", luego la subcategoría "AQ" y finalmente "AQ1x12bits".

Luego arrastrar el elemento hasta el *slot* ubicado en la parte central del PCL.

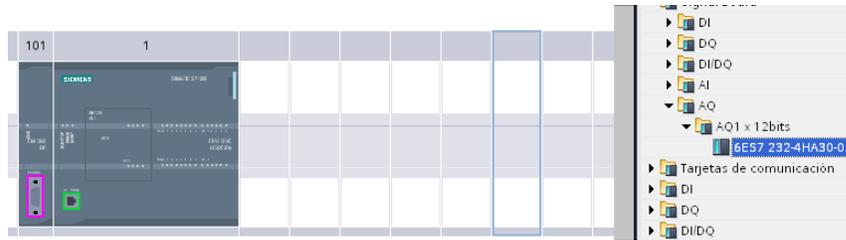


Figura P6.20: Agregar *Signal Board*

28. Dar doble clic sobre la tarjeta añadida y en la ventana "Propiedades" en la sección "Canal 0" Escoger una salida de tensión o voltaje.

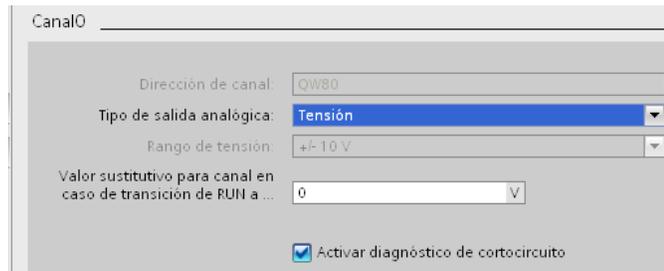


Figura P6.21: Configuración Canal 0 Analógico

29. En la sección "Dirección E/S" verificar el número asignado, en este caso corresponde a la dirección inicial 80.

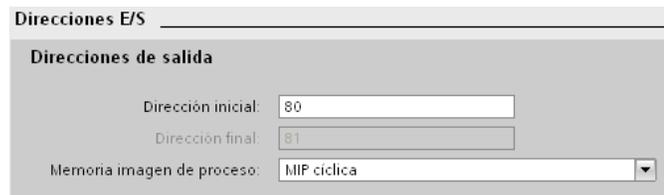


Figura P6.22: Dirección Inicial Entradas Analógicas

30. En el maestro agregar un segmento para escribir dos valores de tipo Word en el área de transferencia tres mediante las entradas I1.0 e I1.1.

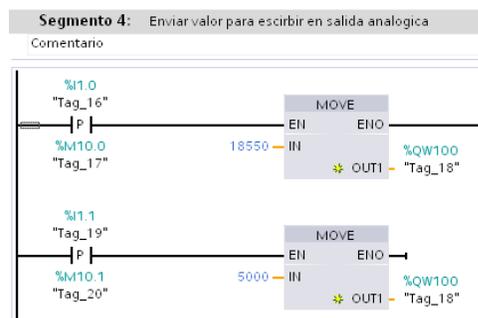


Figura P6.23: Segmento 4 Maestro P5

31. En el esclavo agregar un segmento para escribir el dato recibido y escribirlo en el área de salida del canal 0 analógico.

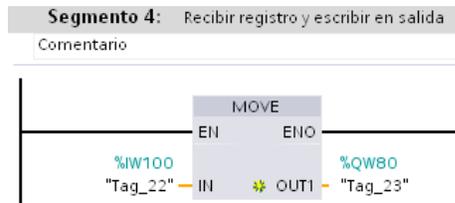


Figura P6.24: Segmento 2 Maestro P5

32. Si se desea se puede agregar una tabla de observación en el ESCLAVO, en el “Árbol del proyecto” dar clic en “Tablas de Observación” y luego en “Agregar nueva tabla”.

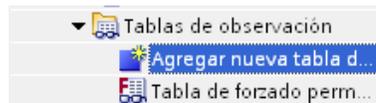


Figura P6.25: Agregar Tabla de Observación

33. Verificar el funcionamiento agregando la variable de la salida analógica en la tabla de observación, dirección QW80. También medir con un multímetro el voltaje en la salida del PLC ESCLAVO.

Nombre	Dirección	Formato visualiz...	Valor de observación	Valor de forzado
"Tag_23"	%QW80	DEC	18550	

Nombre	Dirección	Formato visualiz...	Valor de observación	Valor de forzado
"Tag_23"	%QW80	DEC	5000	

Figura P6.26: Visualización Tabla de Observación PLC ESCLAVO

#### f. Envío de una Palabra (Word) desde el Esclavo

34. Dar doble clic sobre “Configuración de Dispositivos”, luego doble clic sobre el PLC. En la parte izquierda de la Ventana de Propiedades dar clic en la categoría “AI 2”, luego en la sección “Reducción de Ruido escoger un tiempo de integración a 60Hz”

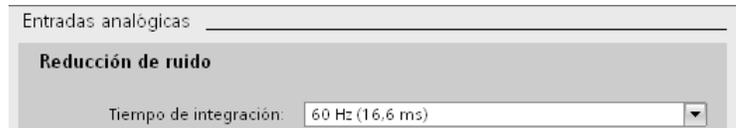


Figura P6.27: Configuración entrada Analógica

35. En la sección “Dirección E/S” verificar la dirección inicial que corresponde a la primer entrada analógica.

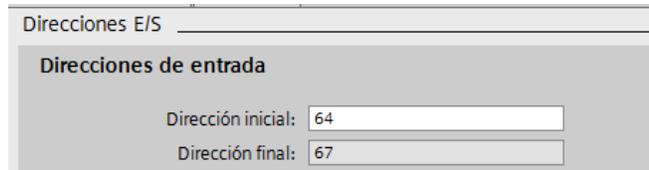


Figura P6.28: Dirección Entrada Analógica

36. En el esclavo agregar un segmento para escribir el valor analógico de entrada en el área de transferencia cuatro.

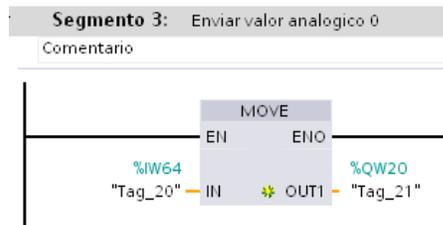


Figura P6.29: Segmento 3 Esclavo P5

37. En el maestro agregar un segmento con las funciones NORM\_X y SCALE\_X para convertir el valor de registro recibido a una señal escalada entre 0 y 10V.

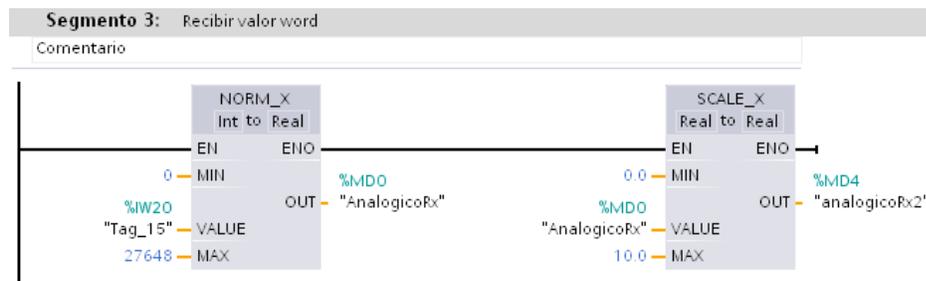


Figura P6.30: Segmento 3 Maestro P5

38. Repetir el paso 32 para agregar una tabla de observación en el Maestro.
39. Comprobar el funcionamiento de la comunicación visualizando el dato recibido en el maestro agregando la variable “analógicaRx2” a la tabla de observación.



Nombre	Dirección	Formato visualiz...	Valor de observa...	Valor de forzado
"analogicoRx2"	%MD4	Número en c...	3.346354	
	<Agregar>			

Figura P6.31: Visualización Tabla de Observación PLC MAESTRO

### **EJERCICIO PARA EL ALUMNO**

En base a lo revisado acerca de la comunicación Profibus el alumno deberá crear un proyecto que permita la activación de un motor DC en el esclavo y su respectivo control de velocidad mediante PWM desde un potenciómetro conectado al maestro.

El ejercicio deberá tener una pantalla HMI asociada al PLC MAESTRO la cual contendrá dos botones para activar o desactivar el PWM, un campo de texto para visualizar el porcentaje PWM enviado y un visor de curvas que permita observar el valor de la entrada analógica en el tiempo en un rango de 0 a 10V.

### **CONCLUSIONES:**

## ANEXO 7: PRÁCTICA 7

**TEMA:** Comunicación Modbus-RTU entre dos PLCs Simatic S7-1200 con el modelo maestro-esclavo.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la comunicación de dos PLCs Simatic S7-1200 mediante una red Modbus-RTU.

### OBJETIVOS ESPECÍFICOS:

- Familiarizarse con los parámetros de la red Modbus-RTU.
- Configurar los módulos de comunicación punto a punto para el uso del protocolo Modbus.
- Familiarizarse con las instrucciones MB\_MASTER y MB\_SLAVE.
- Leer y escribir datos en los registros de salida binarios del esclavo aplicando los códigos de función Modbus pertinentes.
- Leer y escribir datos en los registros de almacenamiento del esclavo aplicando los códigos de función Modbus pertinentes.

### MATERIALES:

- Dos PLCs Simatic S7-1200.
- *Software* TIA PORTAL.
- Un cable apantallado de 2 hilos, conectores interfaz RS-485.
- Dos módulos de Comunicación CM 1241.
- Un potenciómetro.
- Un LED de 12V.

**FUNDAMENTO TEÓRICO:****MODBUS**

Modbus es un protocolo de comunicación industrial desarrollado por la empresa Modicon en el año de 1979 con el objetivo de enlazar los dispositivos controladores con los elementos de campo, se usa para la transmisión de señales con el modelo maestro-esclavo o cliente-servidor. (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008).

Modbus define una estructura de mensaje que los controladores electrónicos podrán interpretar sin importar el tipo de red que utilicen para sus comunicaciones, usa distintas variantes para la transmisión de datos dependiendo el entorno en el que se desarrolle la aplicación, estas variantes incluyen: Modbus ASCII, RTU y TCP/IP usados tanto en la *intranet* como la *extranet*.

El intercambio de información en la red puede ser punto a punto o mediante difusión (*broadcast*) (Balcells & Romeral, 1997). Los mensajes poseen una estructura de conformada por cuatro campos: Dirección, Código de Función, Datos, CRC.

Los códigos de función determinan el tipo de petición que se va a realizar y que direcciones de memoria serán accedidas, los códigos de función utilizado para acciones de lectura y escritura en el esclavo se muestran en las siguientes tablas.

Tabla P7.1: Códigos de Función para lectura Modbus

Código de función Modbus	Funciones de lectura de esclavo (servidor), direccionamiento estándar
01	Leer bits de salida: De 1 a 2000 bits por petición
02	Leer bits de entrada: De 1 a 2000 bits por petición
03	Leer registros de retención: De 1 a 125 palabras por petición
04	Leer palabras de entrada: De 1 a 125 palabras por petición

Tabla P7.2: Códigos de Función para escritura Modbus

Código de función Modbus	Funciones de escritura de esclavo (servidor), direccionamiento estándar
05	Escribir un bit de salida: 1 bit por petición
06	Escribir un registro de retención: 1 palabra por petición
15	Escribir uno o más bits de salida: De 1 a 1968 bits por petición
16	Escribir uno o más registros de retención: De 1 a 123 palabras por petición

### MODBUS-RTU

El modo de transmisión RTU (*Remote Terminal Unit*, “Unidad Terminal Remota”) se destaca por que los bytes de datos se envían sin ningún tipo de conversión, es decir en una codificación plana ya que está pensado para comunicaciones serie. Para la delimitación del mensaje se utiliza intervalos de tiempo de caracteres de silencio.

El medio de transmisión generalmente utilizado es un cable apantallado según la interfaz RS-485 conectada en topología de bus. El formato para cada byte en el modo RTU es.

Tabla P7.3: Formato de Byte en Modbus-RTU

RTU	
<b>Sistema de Codificación</b>	<ul style="list-style-type: none"> <li>• 8 bits binario, hexadecimal: 0-9 y A-F</li> <li>• Dos carácter hexadecimales contenidos en cada 8 bits de mensaje.</li> </ul>
<b>Bits por Byte</b>	<ul style="list-style-type: none"> <li>• 1 Bit de Inicio</li> <li>• 8 Bits de Datos, bit menos significativo enviado primero.</li> <li>• 1 Bit de paridad (0 bits si no hay paridad)</li> <li>• 1 Bit de parada con paridad, 2 Bits si no hay paridad.</li> </ul>
<b>Comprobación de errores</b>	<ul style="list-style-type: none"> <li>• 16 Bits para CRC (Verificación de Redundancia Cíclica)</li> </ul>

Fuente: (Rodríguez, Comunicaciones Industriales Guía Práctica, 2008)

La trama de datos para este tipo de transmisión está formada por los siguientes campos:

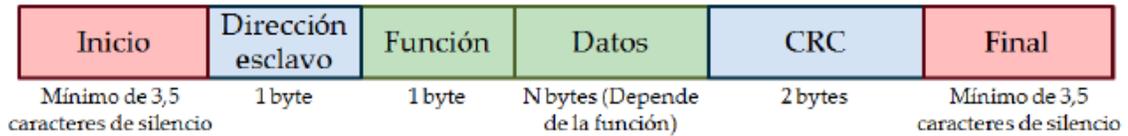


Figura P7.1: Trama de Datos en Modbus-RTU 278

Fuente: (Candelas, 2011)

## PROCEDIMIENTO:

### a. Configuración del Proyecto y del Hardware

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en “Abrir Vista de proyecto”.
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.

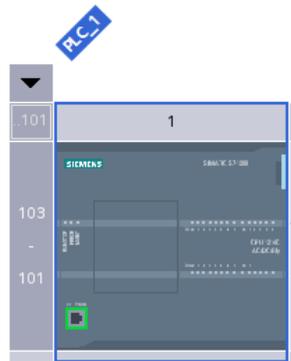


Figura P7.2: PLC1 agregado al proyecto

5. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC2, en este caso será el mismo modelo CPU 1214 AC/DC/RLY.

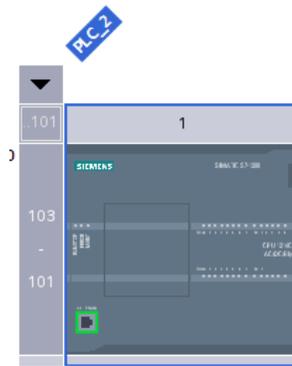


Figura P7.3: PLC2 agregado al proyecto

- En el “Árbol de Proyecto” dar clic derecho sobre el PLC1 y seleccionar “Cambiar Nombre”, el PLC1 se llamará MAESTRO.

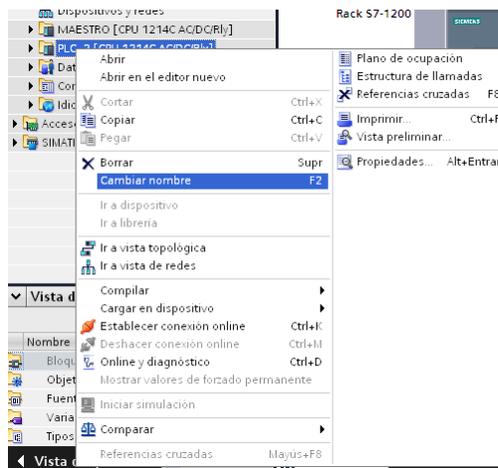


Figura P7.4: Cambiar Nombre PLC1 P6

- Repetir el paso 6 para cambiar el nombre del PLC2 a ESCLAVO.

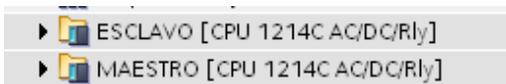


Figura P7.5: Nombres PLCs P6

- Dentro del PLC1 dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC. En la Ventana Propiedades que se despliega en la parte inferior dar clic sobre “Interfaz Profinet”.

Luego en la parte del Protocolo IP colocamos:

Ip: 192.168.0.1

Mascara de Subred: 255.255.255.0

9. Repetir el paso 8 para el PLC2 y colocar los siguientes datos.

Ip: 192.168.0.4

Mascara de Subred: 255.255.255.0

10. En el PLC MAESTRO dar doble clic sobre “Configuración de Dispositivos”, luego en la pestaña derecha “Catálogo de *Hardware*” expandir la categoría “Módulos de Comunicación”, luego la subcategoría “Punto a Punto” y finalmente “CM 1241 (RS422/485)”.

Luego arrastrar el elemento hasta el *slot* ubicado en la parte izquierda del PCL.

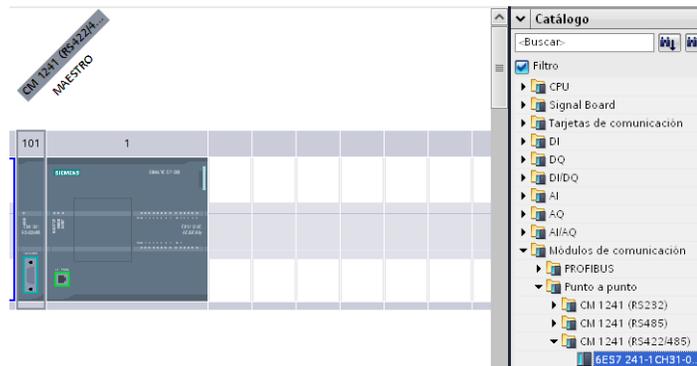


Figura P7.6: Agregar Módulo de Comunicación CM 1241 a Maestro

11. Repetir el paso 10 para el ESCLAVO.

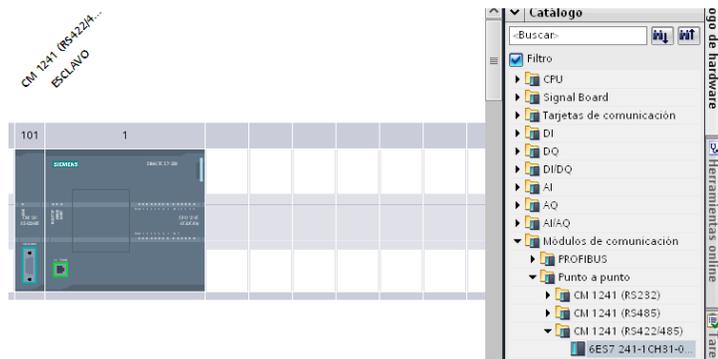


Figura P7.7: Agregar Módulo de Comunicación CM 1241 a Esclavo

12. En el PLC MAESTRO dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el módulo de comunicación.

En la Ventana Propiedades dar clic sobre “Interfaz RS422/485”. Luego en la categoría “Configuración de Puerto” escoger el modo de operación “*Semiduplex* (RS485) modo a 2 hilos”.



Figura P7.8: Modo de Operación Módulos CM 1241

13. En la sección ID de *hardware* verificar el número asignado para utilizarlo posteriormente en la configuración del puerto.

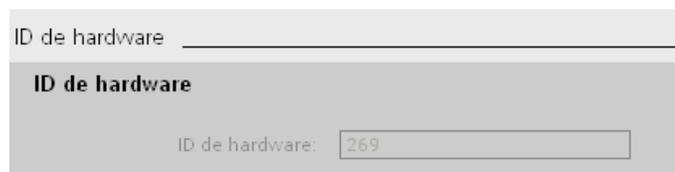


Figura P7.9: ID de *Hardware* Módulo CM 1241

14. Repetir las configuraciones en el módulo de comunicación agregado al ESCLAVO y verificar el ID de *hardware*, en este caso 269.
15. Grabar ambos PLCs y comprobar que los módulos de comunicación del MAESTRO y ESCLAVO no se encuentren en un estado de error.

#### **b. Configuración del Puerto**

16. La configuración del puerto se debe realizar una vez en el arranque del PLC para lo cual utilizar una marca de primer ciclo en ambos PLCs.  
Para ello dentro del PLC dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC.  
En la Ventana Propiedades dar clic sobre “Marcas de sistema y de ciclo”. Luego activar la casilla de verificación para bits de marcas de sistema.

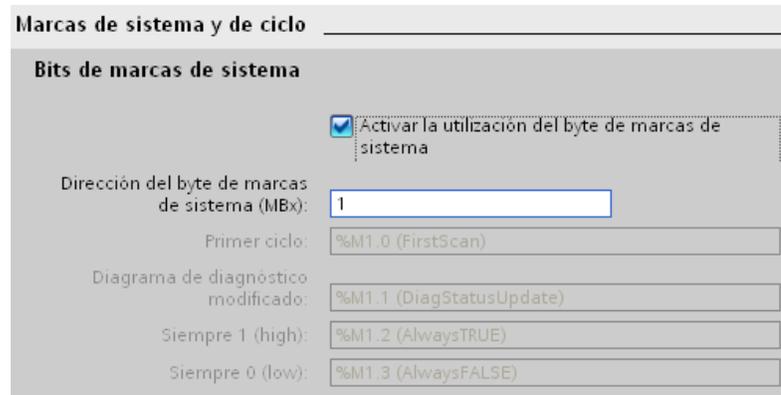


Figura P7.10: Activar Marcas de Ciclo de los PLCs

17. En la parte derecha de TIA PORTAL en el “Árbol de Instrucciones” en la sección “Comunicación”, luego en la categoría “Procesador de Comunicación” y en la sección “Modbus” agregar al bloque MAIN la instrucción MB\_COMM\_LOAD en el MAESTRO.

Aceptar el aviso para la creación de un nuevo bloque de datos, el mismo que contendrá las variables de la estructura.

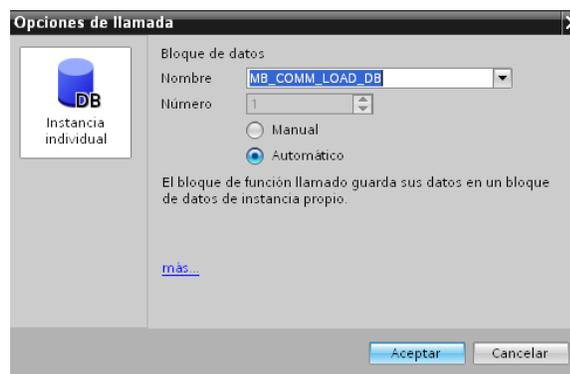


Figura P7.11: Creación Bloque de datos MB\_COMM\_LOAD

18. Luego agregar la instrucción MB\_MASTER y aceptar la creación del bloque de datos (Posteriormente se configurará esta instrucción).

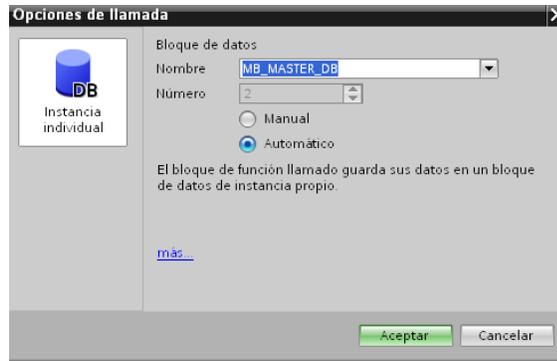


Figura P7.12: Creación Bloque de datos MB\_MASTER

19. Configurar la instrucción MB\_COMM\_LOAD con los siguientes parámetros.

REQ: M1.0 (*FirstScan*).

PORT: 269 (ID de *hardware* del Módulo de Comunicación).

BAUD: 19200 Baudios/s.

PARITY: 1 (Paridad Impar).

MB\_DB: MB\_MASTER\_DB (DB creado al insertar la instrucción MB\_MASTER).

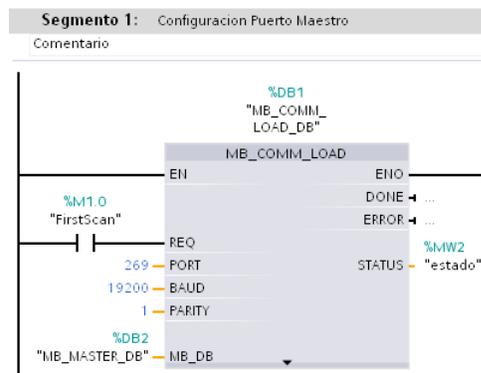


Figura P7.13: Instrucción MB\_COMM\_LOAD Maestro

20. Repetir el paso 17 para agregar la instrucción MB\_COMM\_LOAD en el ESCLAVO.

21. Luego agregar la instrucción MB\_SLAVE y aceptar la creación del bloque de datos (Posteriormente se configurará esta instrucción).

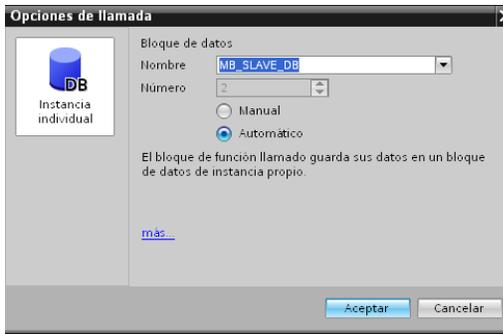


Figura P7.14: Creación Bloque de datos MB\_SLAVE

22. Repetir el paso 19 para configurar la instrucción MB\_COMM\_LOAD en el ESCLAVO con la diferencia de que en el parámetro MB\_DB se debe escoger el bloque de datos MB\_SLAVE\_DB.

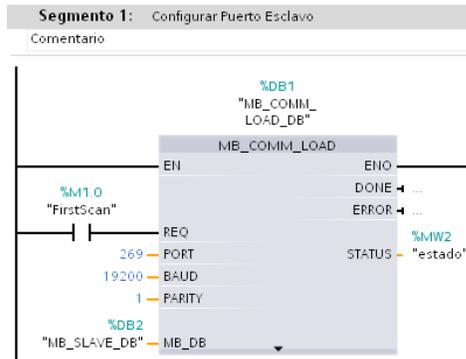


Figura P7.15: Instrucción MB\_COMM\_LOAD Esclavo

23. Configurar la instrucción MB\_SLAVE con los siguientes parámetros.  
 MB\_ADDR: 1 (Dirección Modbus del PLC esclavo.)  
 MB\_HOLD\_REG: P#M100.0 WORD 10 (Puntero a un área de memoria para escribir información del protocolo Modbus).

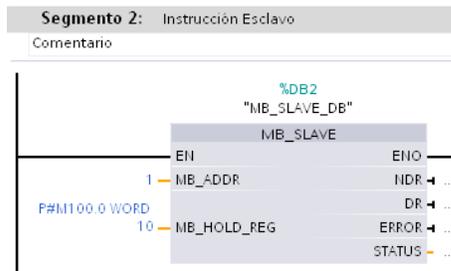


Figura P7.16: Instrucción MB\_SLAVE

**c. Prueba 1: Leer Salidas del Esclavo**

24. Agregar la instrucción MB\_MASTER y configurarla para leer los datos de las tres primeras salidas físicas del esclavo, los parámetros más importantes son:

REQ: TRUE (Para enviar peticiones continuas de comunicación).

MB\_ADDR: 1 (Dirección Modbus del esclavo).

MB\_MODE: 0 (Acción de lectura).

MB\_DATA\_ADDR: 1 (Dirección inicial que se desea intervenir en el esclavo).

MB\_DATA\_LEN: 3 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M20.0 BYTE 1 (Puntero al área de memoria donde se escribirán los datos recibidos).

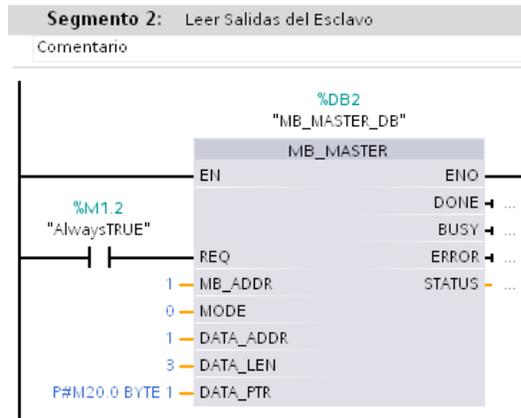


Figura P7.17: Instrucción MB\_MASTER 1

25. En el maestro agregar el segmento siguiente para encender las salidas Q0.0, Q0.1 y Q0.2 con de acuerdo a los datos recibidos del esclavo.

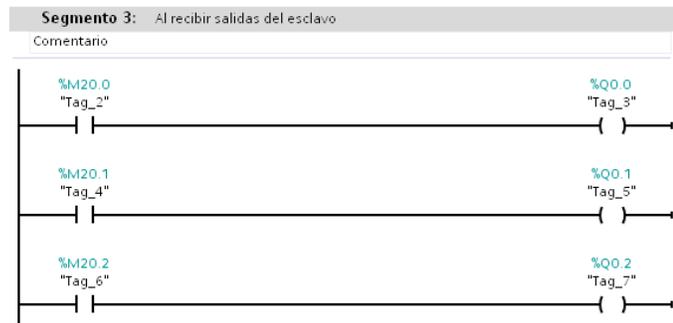


Figura P7.18: Segmento 3 Maestro P6

26. En el esclavo agregar un segmento para establecer los niveles binarios de las salidas, estos valores serán leídos por el maestro.

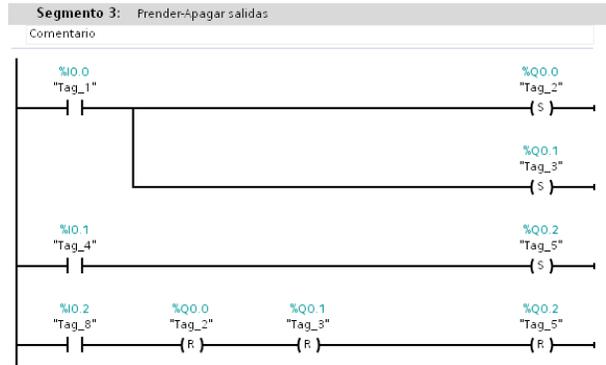


Figura P7.19: Segmento 3 Esclavo P6

27. Verificar el funcionamiento pulsando las entradas I0.0, I0.1 e I0.2 en el esclavo y observando la activación de las salidas del esclavo y por consiguiente del maestro.

**d. Prueba 2: Escribir Salidas en el Esclavo**

28. Agregar la instrucción MB\_MASTER y configurarla para escribir tres salidas físicas del esclavo, los parámetros que difieren con respecto al paso 24 son:

MB\_MODE: 1 (Acción de escritura).

MB\_DATA\_ADDR: 8 (Dirección inicial que se desea intervenir en el esclavo, en este caso el número 8 corresponde a la salida Q1.0).

MB\_DATA\_LEN: 3 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M30.0 BYTE 1 (Puntero al área de memoria de donde se extraerán los datos para escribir en el esclavo).

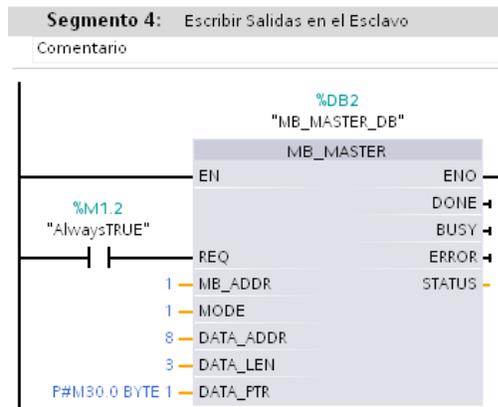


Figura P7.20: Instrucción MB\_MASTER 2

29. En el maestro agregar un segmento para establecer los valores binarios de los tres datos a enviar.

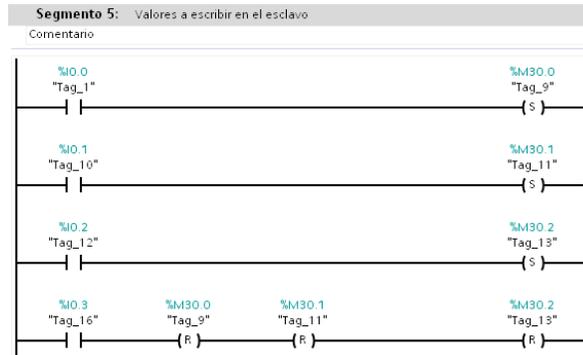


Figura P7.21: Segmento 5 Maestro P6

30. Verificar el funcionamiento pulsando las entradas I0.0, I0.1 e I0.2 del maestro y observando sus correspondientes bits de activación en el esclavo.

**e. Prueba 3: Leer Palabra (Word) del Registro de Almacenamiento del Esclavo**

31. Agregar la instrucción MB\_MASTER y configurarla para leer datos de almacenamiento del esclavo (en este caso será una variable analógica), los parámetros más importantes 24 son:

MB\_MODE: 0 (Acción de lectura).

MB\_DATA\_ADDR: 40001 (Dirección inicial determinada por el protocolo Modbus, en este caso apuntará al área que se escogió previamente para guardar la información de Modbus en el esclavo).

MB\_DATA\_LEN: 1 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M200.0 WORD 1 (Puntero al área de memoria en donde se escribirá la palabra recibida del esclavo).

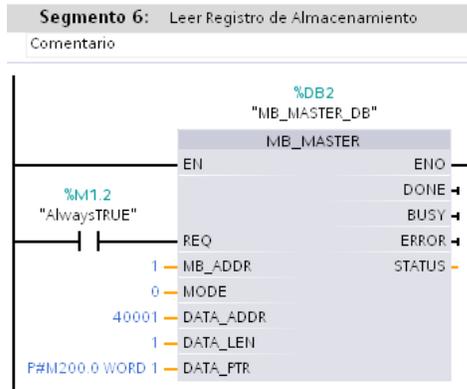


Figura P7.22: Instrucción MB\_MASTER 3

- 32. En el maestro agregar un segmento con las funciones NORM\_X y SCALE\_X para convertir el valor de registro recibido a una señal escalada entre 0 y 10V.

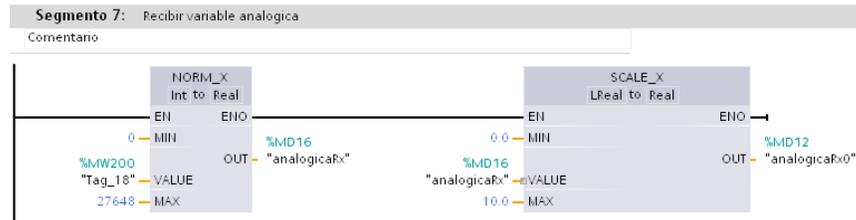


Figura P7.23: Segmento 7 Maestro P6

- 33. En el maestro agregar otro segmento para la comparación del valor analógico recibido y encender un indicador luminoso si dicho valor esta fuera de un rango de 2 a 5 V.

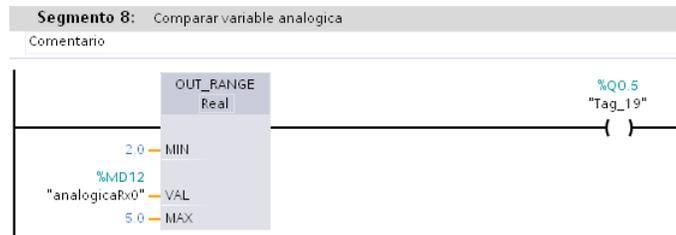


Figura P7.24: Segmento 8 Maestro P6

- 34. En el esclavo conectar un potenciómetro a la entrada analógica 0.
- 35. En el esclavo agregar un segmento para escribir el valor analógico de entrada en el área de almacenamiento Modbus dentro de la primera dirección, la cual será transmitida.

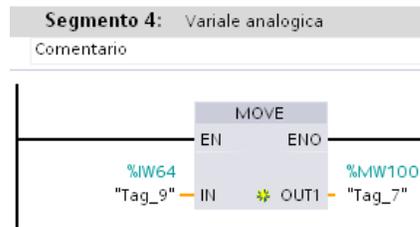


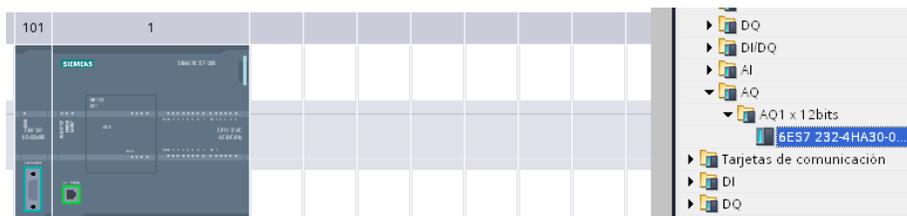
Figura P7.25: Segmento 4 Esclavo P6

36. Verificar el funcionamiento girando el potenciómetro conectado al esclavo y observando el valor recibido en el maestro, además de la comparación dentro de un rango de 2 a 5 V.

#### f. Prueba 4: Escribir Palabra (Word) en un Registro de almacenamiento del Esclavo

37. Agregar la tarjeta correspondiente a una salida analógica en el PLC ESCLAVO. Dar doble clic sobre “Configuración de Dispositivos”, luego en la pestaña derecha “Catálogo de *Hardware*” expandir la categoría “*Signal Board*”, luego la subcategoría “AQ” y finalmente “AQ1x12bits”.

Luego arrastrar el elemento hasta el *slot* ubicado en la parte central del PCL.

Figura P7.26: Agregar *Signal Board*

38. Agregar la instrucción MB\_MASTER y configurarla para escribir datos en un registro de almacenamiento del esclavo (variable analógica), los parámetros más importantes 24 son:

MB\_MODE: 1 (Acción de escritura).

MB\_DATA\_ADDR: 40002 (Dirección inicial determinada por el protocolo Modbus, en este caso apuntará al área que se escogió previamente para guardar la información de Modbus en el esclavo).

MB\_DATA\_LEN: 1 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M300.0 WORD 1 (Puntero al área de memoria de donde se extraerá la palabra para escribirla en el esclavo).

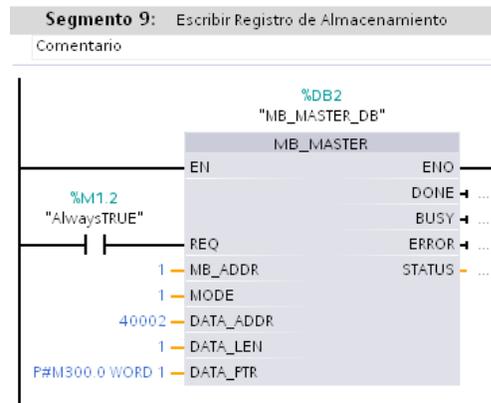


Figura P7.27: Instrucción MB\_MASTER 4

39. En el maestro agregar un segmento para determinar el valor de la palabra a escribir, en este caso se aumentará progresivamente el registro de la palabra al pulsar la entrada I1.0, el paso de incremento es de 300 lo que corresponderá a un valor de 0.1V.

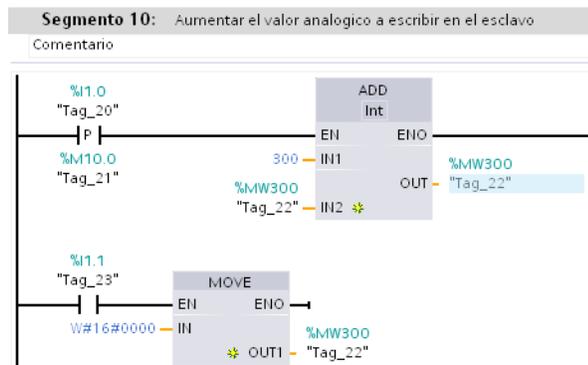


Figura P7.28: Segmento 10 Maestro P6

40. En el esclavo agregar un segmento para escribir el valor de registro recibido dentro de la dirección de la salida analógica física.

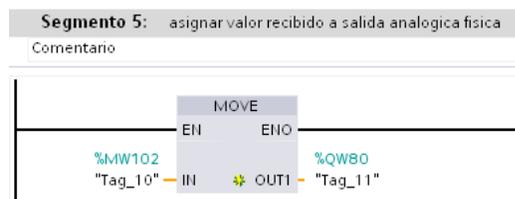


Figura P7.29: Segmento 5 Esclavo P6

41. Verificar el funcionamiento pulsando repetidamente la entrada I1.0 del maestro y midiendo el aumento de voltaje en la salida física del esclavo.

### **EJERCICIO PARA EL ALUMNO**

En base a lo revisado acerca de la comunicación Modbus-RTU el alumno deberá crear un proyecto que permita lo siguiente:

- Leer la entrada analógica en el esclavo y reflejar este mismo nivel de tensión en la salida analógica del maestro, pero con las siguientes excepciones.
- Si el valor de la entrada analógica en el esclavo supera los 9V, entonces activar una alarma hacia el maestro para que este configure un valor sustitutivo de 9.2V en su salida.
- Si el valor de la entrada analógica en el esclavo es menor que 5V, entonces activar una alarma hacia el maestro para que este configure un valor sustitutivo de 1.5V en su salida.

Conectar un potenciómetro en la entrada analógica del esclavo y un led de 12V en la salida analógica del maestro, medir el voltaje.

### **CONCLUSIONES:**

## ANEXO 8: PRÁCTICA 8

**TEMA:** Comunicación Modbus-TCP/IP entre dos PLCs Simatic S7-1200 con el modelo cliente-servidor.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la comunicación de dos PLCs Simatic S7-1200 mediante una red Modbus-TCP/IP.

### OBJETIVOS ESPECÍFICOS:

- Familiarizarse con los parámetros de la red Modbus-TCP.
- Configurar una red de comunicación Modbus-TCP.
- Familiarizarse con las instrucciones MB\_CLIENT y MB\_SERVER.
- Establecer dos canales de comunicación entre los equipos.
- Leer datos binarios de entrada del esclavo aplicando el código de función Modbus pertinente.
- Leer y escribir datos en los registros de entrada (de tipo Word) del esclavo aplicando los códigos de función Modbus pertinentes.

### MATERIALES:

- Tres PLCs Simatic S7-1200.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.
- Un potenciómetro.
- Una Pantalla HMI.
- Dos motores trifásicos

## FUNDAMENTO TEÓRICO:

### *MODBUS-TCP/IP*

Modbus-TCP es un protocolo de comunicación que define el uso de mensajes Modbus dentro de un entorno de red TCP, está diseñado para permitir a dispositivos industriales de planta comunicarse con los elementos de gestión o con equipos de la *intranet*, *extranet* o *internet*. Fue desarrollado por *Schneider Automation* como una variante de Modbus cuyo uso para supervisión y control se encontraba altamente implementado en las redes de automatización (Olaya, Barandica, & Guerrero, 2004).

Los dispositivos maestros que se tenían en las redes Modbus convencionales son remplazados por el Cliente de la red TCP y los esclavos son conocidos como los Servidores (M-System CO. LTD., 2002). El protocolo TCP permite establecer una red extensa con varias comunicaciones concurrentes, de modo que los dispositivos maestros pueden intercambiar datos mediante diferentes canales (Olaya, Barandica, & Guerrero, 2004).

Modbus-TCP básicamente encapsula la trama de datos Modbus dentro de su trama TCP, omitiendo los campos incensarios como: *Start*, *CRC o LRC* y *END*, ya cubiertos por las tramas *Ethernet* (véase Figura 1.29). La comprobación de datos corruptos en un modelo Consulta-Respuesta es detectada usando TCP/IP o mecanismos de verificación de la capa de enlace (M-System CO. LTD., 2002).

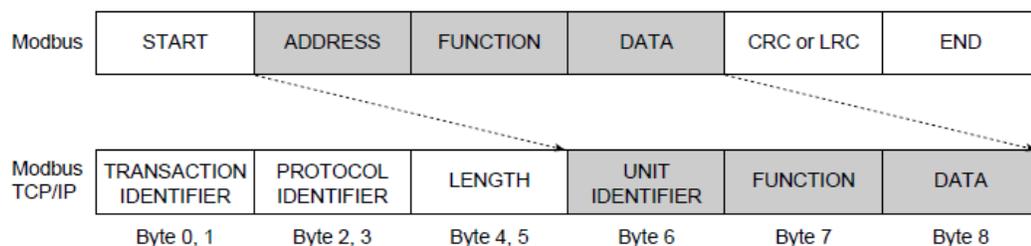


Figura P8.1: Trama de Datos en Modbus-TCP/IP 307

Fuente: (M-System CO. LTD., 2002)

**PROCEDIMIENTO:****a. Configuración del Proyecto y del Hardware**

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en “Abrir Vista del Proyecto”.
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.

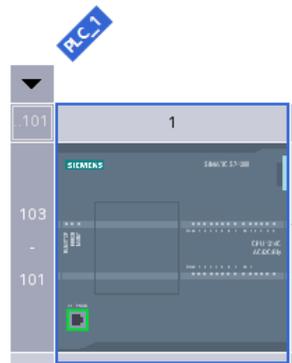


Figura P8.2: PLC1 agregado al proyecto

5. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC2, en este caso será el mismo modelo CPU 1214 AC/DC/RLY.

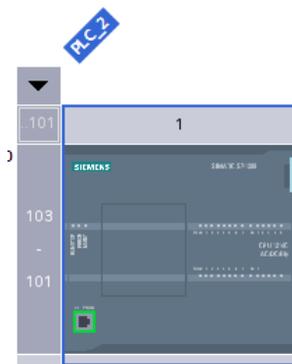


Figura P8.3: PLC2 agregado al proyecto

6. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC3, en este caso será el mismo modelo CPU 1214 AC/DC/RLY.

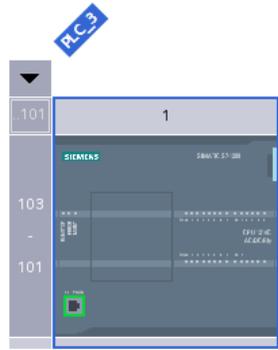


Figura P8.4: PLC3 agregado al proyecto

7. En el “Árbol de Proyecto” dar clic derecho sobre el PLC1 y seleccionar “Cambiar Nombre”, luego escribir MAESTRO.

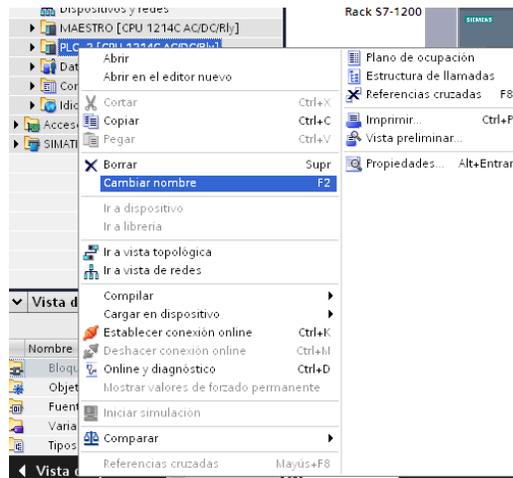


Figura P8.5: Cambiar Nombre PLC1 P8

8. Repetir el paso 7 para cambiar el nombre de los PLCs 1 y 2, asignar los nombres ESCLAVO1 y ESCLAVO2 respectivamente.

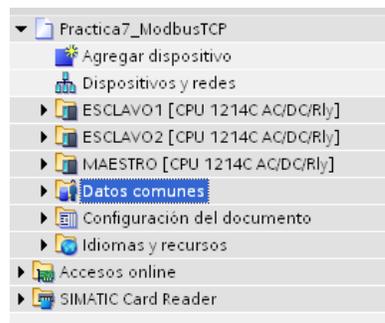


Figura P8.6: Nombres PLCs P8

9. Dentro del PLC Maestro dar doble clic sobre “Configuración de Dispositivos” y luego doble clic sobre el PLC. En la Ventana Propiedades que se despliega en la parte inferior dar clic sobre “Interfaz Profinet”.

Luego en la parte del Protocolo IP colocamos:

Ip: 192.168.0.1

Mascara de Subred: 255.255.255.0

10. Repetir el paso 9 para el PLC Esclavo1 y colocar los siguientes datos.

Ip: 192.168.0.4

Mascara de Subred: 255.255.255.0

11. Repetir el paso 9 para el PLC Esclavo2 y colocar los siguientes datos.

Ip: 192.168.0.5

Mascara de Subred: 255.255.255.0

12. Guardar el proyecto y Grabar los tres PLCs, comprobar que no existan errores.

#### b. Configuración Maestro

13. En la parte derecha de TIA PORTAL en el “Árbol de Instrucciones” en la sección “Comunicación”, luego en la categoría “Procesador de Comunicación” y en la sección “Modbus TCP” agregar al bloque MAIN la instrucción MB\_CLIENT en el Maestro.

Aceptar el aviso para la creación de un nuevo bloque de datos, el mismo que contendrá las variables de la estructura.



Figura P8.7: Creación Bloque de datos MB\_CLIENT

14. Agregar otra instrucción MB\_CLIENT y crear otro DB (Posteriormente serán configurada.)

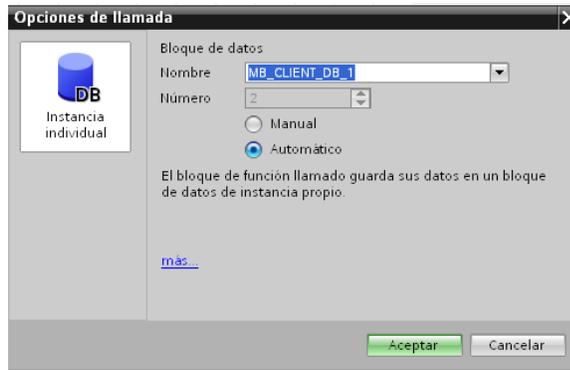


Figura P8.8: Creación Bloque de datos MB\_CLIENT 1

15. En el Esclavo1 agregar la instrucción MB\_SERVER y aceptar la creación del bloque de datos.

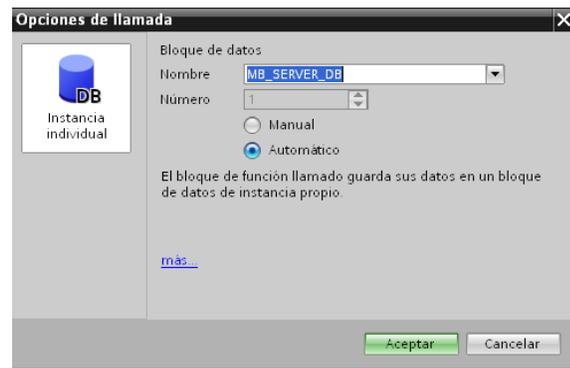


Figura P8.9: Creación Bloque de datos MB\_SERVER Esclavo1

16. Configurar la instrucción MB\_SERVER del Esclavo1 para la conexión por Modbus TCP, los parámetro más importantes son:
  - DISCONNECT: I1.0 NO. (Si el valor es “1” no se ejecutan las operaciones).
  - CONNECT\_ID: 1 (unívoca que permite diferenciar las conexiones).
  - IP\_PORT: 502 (Puerto por defecto para la comunicación ModbusTCP).
  - MB\_HOLD\_REG: P#M100.0 BYTE 10 (Puntero a un área de memoria para escribir información del protocolo Modbus).

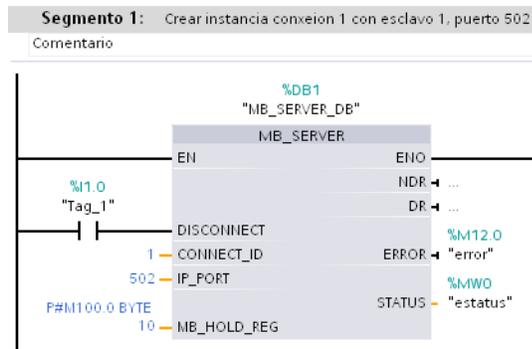


Figura P8.10: Instrucción MB\_SERVER Esclavo1 (Canal 1)

17. En el Esclavo1 repetir los pasos 15 y 16 para agregar y configurar una nueva conexión por un segundo canal desde el Maestro hacia el Esclavo1, los parámetros son los siguientes:

DISCONNECT: I1.1 NO. (Si el valor es "1" no se ejecutan las operaciones).

CONNECT\_ID: 3 (unívoca que permite diferenciar las conexiones).

IP\_PORT: 503 (Puerto por defecto para la comunicación ModbusTCP).

MB\_HOLD\_REG: P#M200.0 BYTE 10 (Puntero a un área de memoria para escribir información del protocolo Modbus).

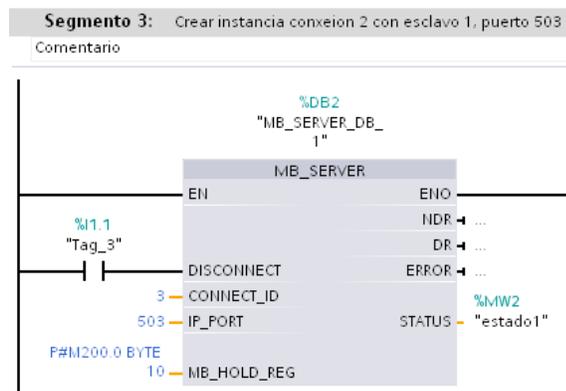


Figura P8.11: Instrucción MB\_SERVER Esclavo1 (Canal 2)

18. Repetir el paso 15 para agregar una instrucción MB\_SERVER en el Esclavo2.

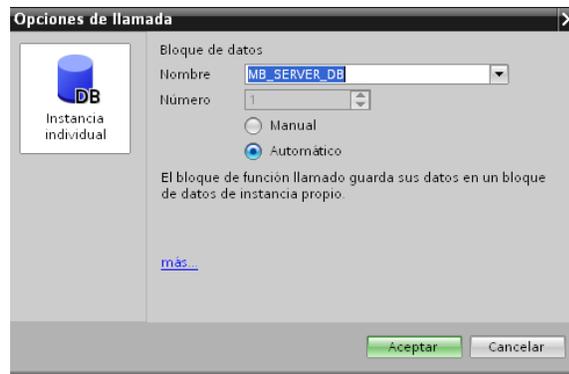


Figura P8.12: Creación Bloque de datos MB\_SERVER Esclavo2

19. Configurar la instrucción MB\_SERVER del Esclavo2 para la conexión por Modbus TCP, los parámetro más importantes son:

DISCONNECT: I1.0 NO (Si el valor es “1” no se ejecutan las operaciones).

CONNECT\_ID: 2 (unívoca que permite diferenciar las conexiones).

IP\_PORT: 503 (Puerto para la comunicación ModbusTCP).

MB\_HOLD\_REG: P#M100.0 BYTE 10 (Puntero a un área de memoria para escribir información del protocolo Modbus).

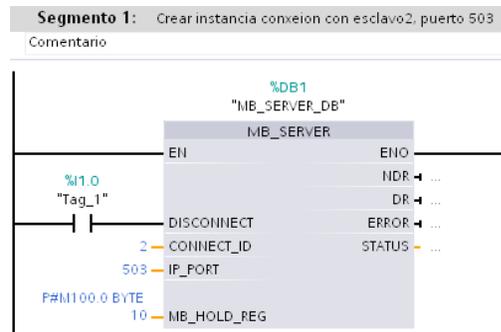


Figura P8.13: Instrucción MB\_SERVER Esclavo2 (Canal 1)

### c. Prueba 1: Leer Datos de Entrada de Ambos Esclavos

20. En el Maestro configurar la primera instrucción MB\_CLIENT para la lectura de los bits de entrada del Esclavo1, los parámetros son los siguientes:

REQ: TRUE (Para enviar peticiones continuas de comunicación).

DISCONNECT: I1.0 (Si el valor es “1” no se ejecutan las operaciones).

IP: 192.168.0.4 (Octetos de la dirección IP del Esclavo).

IP\_PORT: 502 (Puerto por el cual se realiza la comunicación).

MB\_MODE: 0 (Acción de lectura).

MB\_DATA\_ADDR: 10001 (Dirección inicial que se desea intervenir en el esclavo, corresponde a las entradas digitales).

MB\_DATA\_LEN: 3 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M20.0 BYTE 1 (Puntero al área de memoria donde se escribirán los datos recibidos).

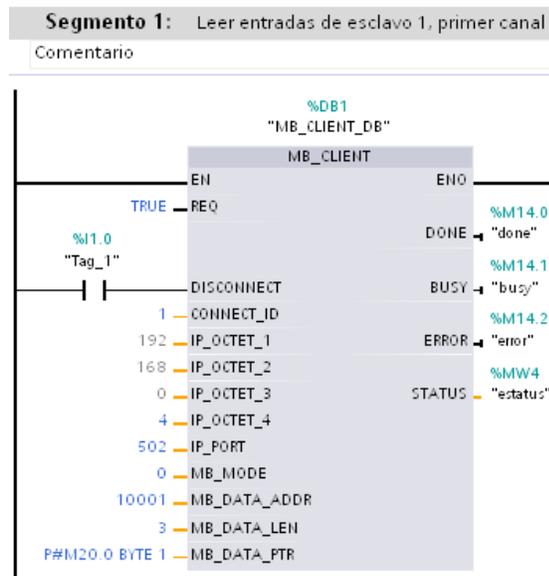


Figura P8.14: Instrucción MB\_CLIENT, Leer Entradas Esclavo 1

21. Repetir el paso 20 para configurar la segunda instrucción MB\_CLIENT del maestro, en este caso difiere: la dirección IP: 192.168.0.5, el puerto IP\_PORT: 503 y el puntero al área de datos MB\_DATA\_PTR: P#M30.0 BYTE 1; esto para la comunicación con el Esclavo2.

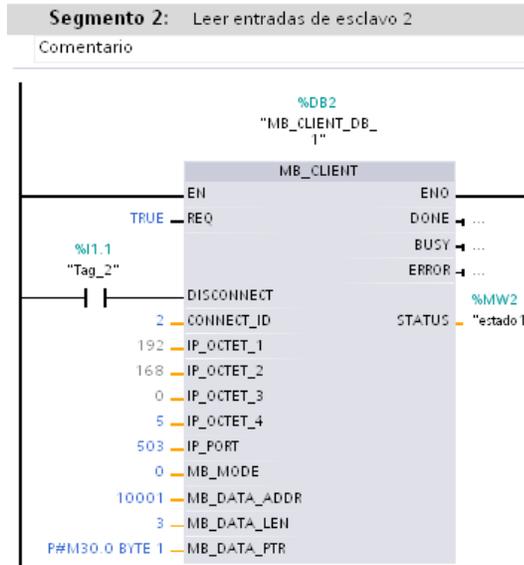


Figura P8.15: Instrucción MB\_CLIENT, Leer Entradas Esclavo2

22. En el Maestro agregar los siguientes segmentos para encender las salidas Q0.0-Q0.2 al recibir los datos del Esclavo1, y las salidas Q0.7-Q1.1 al recibir la información del Esclavo2.

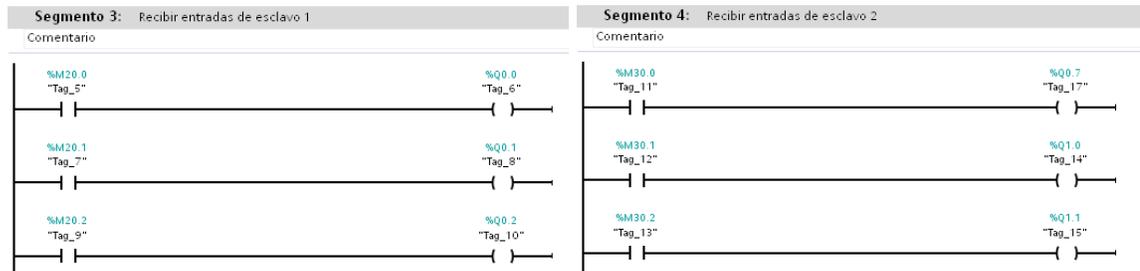


Figura P8.16: Segmento 3 y 4 Maestro P8

23. Verificar el funcionamiento pulsando las entradas I0.0, I0.1 e I0.2 de ambos esclavos y comprobar la activación de las salidas en el maestro.

**d. Prueba 2: Escribir Salidas en Esclavo 2**

24. Agregar otra instrucción MB\_CLIENT en el maestro y asignarle el DB "MB\_CLIENT\_DB\_1" que corresponde a los parámetros para la conexión con segundo esclavo.

Luego configurarla para la escritura de tres bits de salida del Esclavo2, los parámetros son los siguientes:

REQ: TRUE (Para enviar peticiones continuas de comunicación).

DISCONNECT: 11.3 (Si el valor es “1” no se ejecutan las operaciones).

IP: 192.168.0.5 (Octetos de la dirección IP del Esclavo).

IP\_PORT: 503 (Puerto por el cual se realiza la comunicación).

MB\_MODE: 1 (Acción de escritura).

MB\_DATA\_ADDR: 4 (Dirección inicial que se desea intervenir en el esclavo, corresponde a la salida Q0.3).

MB\_DATA\_LEN: 3 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M40.0 BYTE 1 (Puntero al área de memoria desde donde se extraerán los datos para escribir en el esclavo).

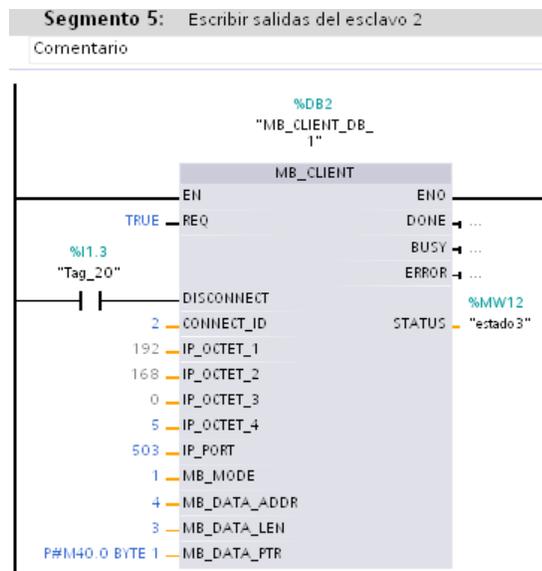


Figura P8.17: Instrucción MB\_CLIENT, Escribir Salidas Esclavo2

25. En el maestro agregar un segmento para establecer los valores binarios de los tres datos a enviar.

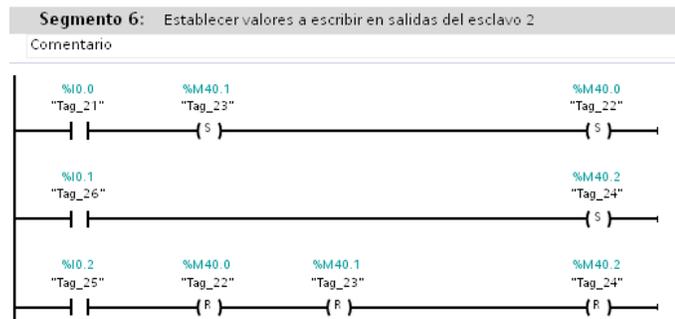


Figura P8.18: Segmento 6 Maestro P8

26. Verificar el funcionamiento pulsando las entradas I.0, I0.1 e I0.2 del maestro y observado la activación de las salidas respectivas en el Esclavo2.

**e. Prueba 3: Leer el Valor Analógico de Registro del Esclavo 1**

27. Agregar otra instrucción MB\_CLIENT y aceptar la creación de un nuevo bloque de datos, ya que esta constituirá el tercer canal de comunicación desde el maestro.

28. Configurar la instrucción para la lectura de datos de almacenamiento del esclavo1, los parámetros son los siguientes:

REQ: TRUE (Para enviar peticiones continuas de comunicación).

DISCONNECT: I1.4 (Si el valor es “1” no se ejecutan las operaciones).

IP: 192.168.0.4 (Octetos de la dirección IP del Esclavo).

IP\_PORT: 503 (Puerto por el cual se realiza la comunicación).

MB\_MODE: 0 (Acción de escritura).

MB\_DATA\_ADDR: 40002 (Dirección inicial que se desea intervenir en el esclavo, corresponde a la segunda palabra de los registros de almacenamiento).

MB\_DATA\_LEN: 1 (Longitud de los datos o registros a intervenir).

MB\_DATA\_PTR: P#M500.0 WORD 1 (Puntero al área de memoria donde se escribirán los datos recibidos).

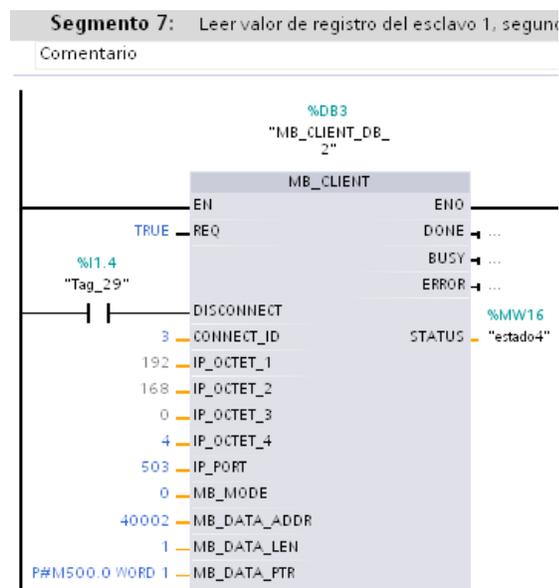


Figura P8.19: Instrucción MB\_CLIENT, Leer Valor de Registro Esclavo1

29. En el esclavo conectar un potenciómetro a la entrada analógica 0.

30. En el esclavo agregar un segmento para escribir el valor analógico de entrada en el área de almacenamiento Modbus dentro de la segunda dirección, la cual será transmitida.

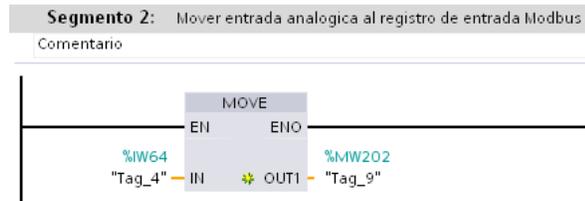


Figura P8.20: Segmento 2 Esclavo1 P8

31. En el maestro agregar un segmento con las funciones NORM\_X y SCALE\_X para convertir el valor de registro recibido a una señal escalada entre 0 y 10V.

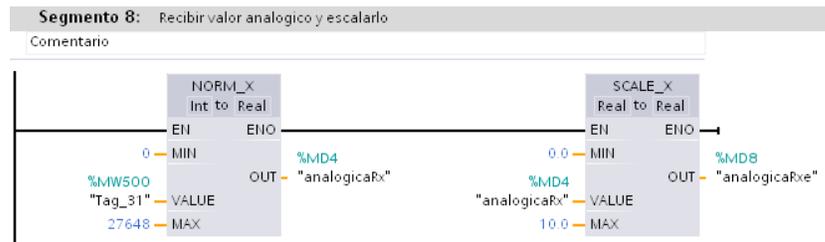


Figura P8.21: Segmento 8 Maestro P8

32. En el maestro agregar otro segmento para la comparación del valor analógico recibido y encender un indicador luminoso si dicho valor está dentro de un rango de 1 a 4 V.

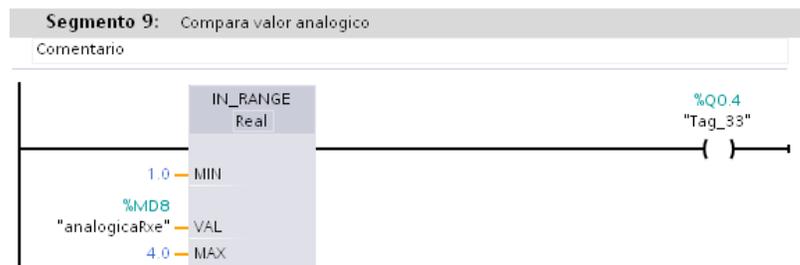


Figura P8.22: Segmento 9 Maestro P8

33. Verificar el funcionamiento girando el potenciómetro conectado al Esclavo1 y observando el valor recibido en el Maestro, además de la comparación dentro de un rango de 1 a 4 V.

## **EJERCICIO PARA EL ALUMNO**

En base a lo revisado acerca de la comunicación Modbus-TCP el alumno deberá crear un proyecto que permita la sincronización de tareas entre tres PLC, con el fin de establecer tiempos de encendido para motores conectados en los esclavos.

La aplicación contiene:

- Un dispositivo maestro conectado a dos equipos esclavos.
- El maestro esta enlazado a una pantalla HMI desde la cual se establecerán los datos.
- El esclavo 2 tendrá un motor conectado con la posibilidad de invertir el giro, el esclavo 1 tendrá otro motor conectado sin inversión de giro.

El proyecto debe funcionar de la siguiente manera:

- Mediante la HMI se puede calibrar los tiempos de giro para los motores.
- Al pulsar un botón Iniciar en la HMI la aplicación encenderá el motor conectado al esclavo 2 según los tiempos establecidos en la pantalla para derecha e izquierda. Los temporizadores deberán manejarse en el esclavo 2.
- Al finalizar el motor del esclavo 2 se deberá encender el motor conectado al esclavo 1 durante el tiempo establecido en la HMI, este temporizador deberá manejarse en el maestro.
- Al pulsar nuevamente el botón Iniciar deberá comenzar nuevamente la aplicación.

**CONCLUSIONES:**

## ANEXO 9: PRÁCTICA 9

**TEMA:** Uso de Avisos y Administración de Usuarios del Sistema SCADA WinCC.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la utilización de Avisos y Usuarios dentro de una aplicación SCADA sencilla.

### OBJETIVOS ESPECÍFICOS:

- Configurar el PLC y la Pantalla HMI.
- Configurar el sistema de Avisos de WinCC tanto para avisos analógicos y de bit.
- Configurar la administración de Usuarios de WinCC para determinar privilegios de acceso.
- Verificar el funcionamiento.

### MATERIALES:

- Un PLC Simatic S7-1200.
- Una Pantalla HMI.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.
- Un potenciómetro.

### FUNDAMENTO TEÓRICO:

#### Sistema SCADA

Un sistema SCADA es cualquier *software* que permita el acceso a datos remotos de un proceso y su respectivo control utilizando ciertas herramientas de comunicación necesarias. No se trata de un sistema de control sino de una utilidad del *software* para monitorear o supervisar las tareas, sirviendo de interface entre niveles de control como PLCs y niveles de gestión (Corrales, 2007).

Además un sistema SCADA se encarga de proveer toda la información necesaria a diversos usuarios dentro de todas las capas de automatización, está diseñado para su aplicación generalmente sobre estaciones basadas en ordenadores (Gomez, 2009).

Los Sistemas SCADA cuentan con varias prestaciones como: Monitorización, Supervisión, Adquisición de datos, Visualización, Mando, Grabación de Recetas, Seguridad de los datos, Seguridad en el acceso, Paneles de Alarma, Generación de Históricos, Posibilidad de Programación Numérica que permiten optimizar los procesos productivos mediante estaciones de planta o de oficina.

### *HMI*

Una Interfaz Hombre-Máquina o HMI es un mecanismo combinado de *hardware* y *software* que le permite a un operador interactuar con un proceso productivo mediante pantallas y funciones específicas, además comprende sinópticos de control y sistemas de presentación gráfica (Rodríguez, Sistemas SCADA 3ra Edición, 2012).

Un HMI tiene la capacidad de visualizar y determinar datos de los accionamientos como estados (*ON/OFF*) o magnitudes de las variables físicas involucradas; puede ser desde un simple interruptor e indicador luminoso para encender algún dispositivo, hasta varias pantallas que esquematizan todo un proceso de producción incluyendo valores en tiempo real y acciones de supervisión (Corrales, 2007).

### *Avisos de WinCC*

El sistema de avisos brinda la posibilidad de visualizar en los paneles gráficos los estados operativos o averías que se han producido en un proceso, existen dos tipos de avisos: los definidos por el sistema y los definidos por el usuario.

### *Administración de Usuarios en WinCC*

WinCC mediante su pantalla de configuración de usuarios permite controlar el acceso a datos y funciones en *Runtime* no autorizados. Cada grupo de usuarios cuenta con privilegios diferentes dependiendo su propósito.

### **PROCEDIMIENTO:**

#### **a. Configuración del Proyecto y del Hardware**

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en “Abrir Vista del proyecto”
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.

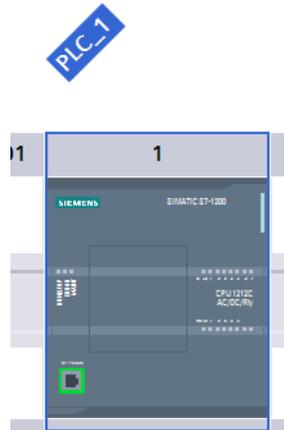


Figura P9.1: PLC1 agregado al proyecto

5. Luego en el “Árbol del Proyecto” dar doble clic en “Agregar Dispositivos” para insertar la pantalla HMI, en el Laboratorio se cuenta con una Pantalla Táctil de 6” KTP600 PN.

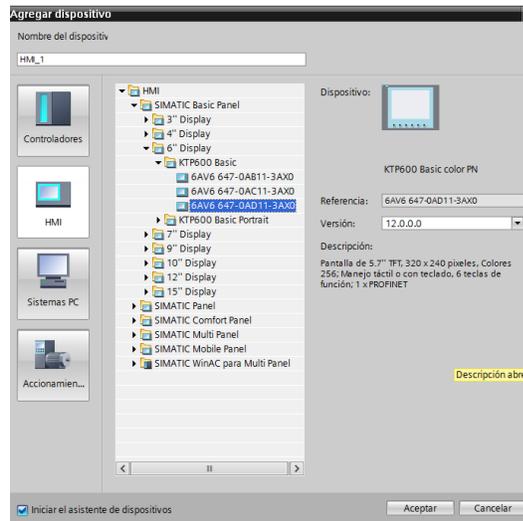


Figura P9.2: Agregar HMI PLC1 al proyecto

18. Luego dar clic en “Seleccionar PLC” y escoger el PLC al que se vinculará la pantalla, en este caso será PLC1.

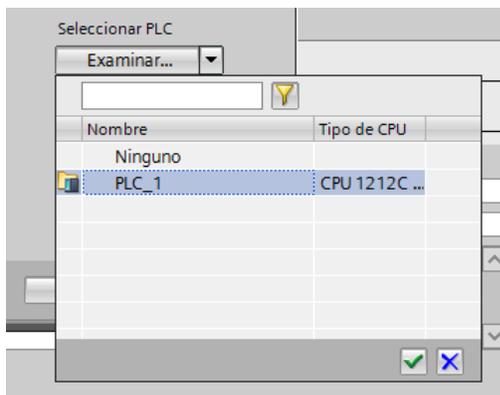


Figura P9.3: Vincular el PLC a la pantalla HMI P8

6. Si se desea se puede realizar más configuraciones en el asistente, en este caso se escogerá los parámetros por defecto, por lo tanto dar clic en “Finalizar”.

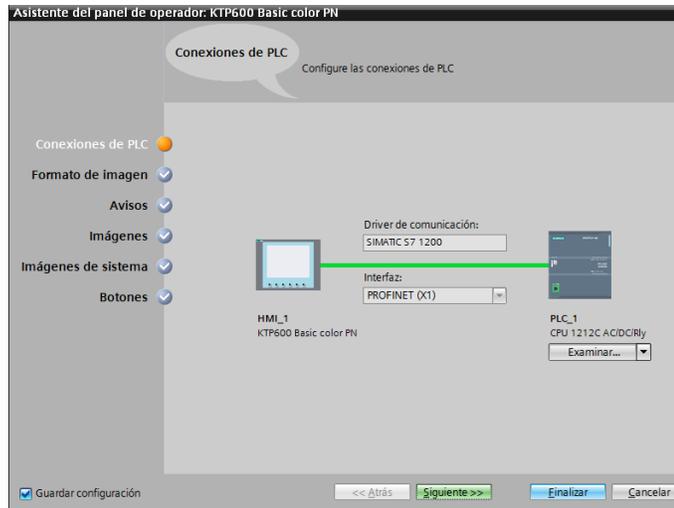


Figura P9.4: Asistente de Configuración Inicial para Pantallas HMI

## b. Programación Segmentos PLC

7. Agregar un segmento para la activación de la salida Q0.0 mediante la entrada IO.0 y además utilizar la función MOVE para establecer el valor de estado de la misma en una variable de tipo *Int* (Para la posterior configuración de Avisos de Bit).

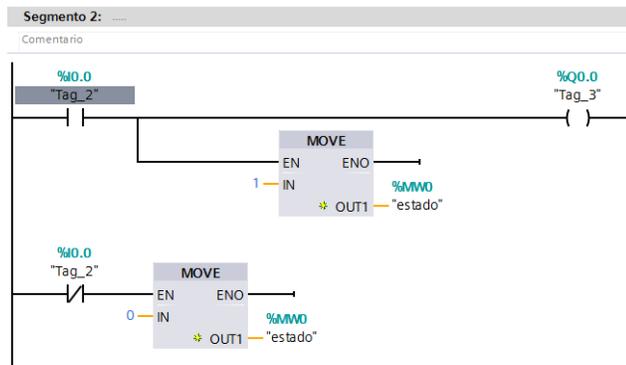


Figura P9.5: Encender Salida Q0.0 y Establecer Variable “Estado”

8. Definir la variable de tipo *Int* llamado “estado”.

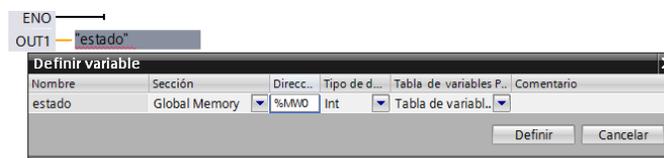


Figura P9.6: Variable “Estado”

### c. Configuración de Usuarios

9. Dentro de la HMI bajo la categoría “Imágenes” abrir “Imagen Raíz”.

Luego en la parte derecha en la paleta “Elementos” escoger y arrastrar un botón hasta el área de trabajo, dar clic sobre el texto para cambiar el nombre y escribir “Iniciar Sesión”.

Luego dar clic en sobre el botón y expandir la Pestaña “Propiedades”, en la categoría “Apariencia” escoger un color de fondo, en este caso azul.

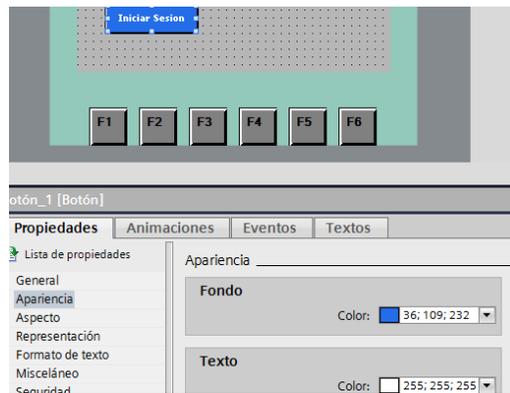


Figura P9.7: Propiedades Botón “Iniciar Sesión”

10. En la Pestaña “Eventos” dar clic sobre “Pulsar”, luego en la parte derecha dar clic en “Agregar función” y escoger “MostrarDialogoInicioSesion”.



Figura P9.8: Eventos Botón “Iniciar Sesión”

11. Repetir el paso anterior para agregar un botón con el texto “Cerrar Sesión” de color rojo, además en la Pestaña Eventos elegir la función “CerrarSesion”.

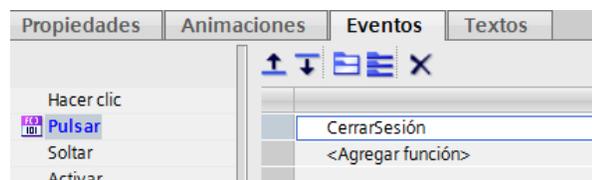


Figura P9.9: Propiedades Botón “Cerrar Sesión”

12. Luego en la parte derecha de la paleta “Controles” arrastrar hacia el área de trabajo un elemento “Visor de Usuarios”.

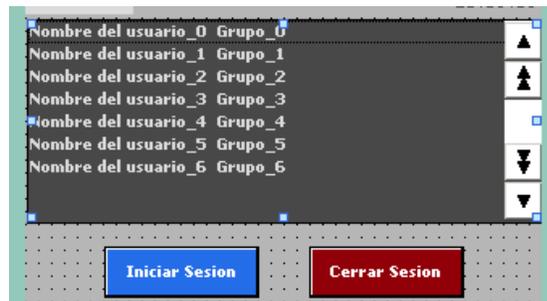


Figura P9.10: Visor de Usuarios

13. Luego dar clic en sobre el visor y expandir la Pestaña “Propiedades”, en la categoría “Visualización” escoger un número de filas igual a 7.

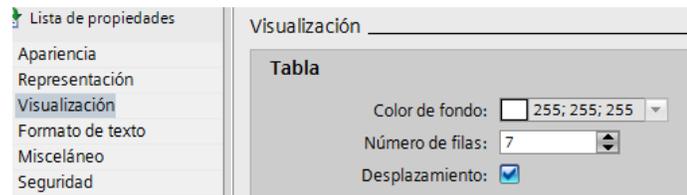


Figura P9.11: Propiedades Botón “Visor de Usuarios”

14. En la Pestaña “Apariencia” cambiar el color de fondo.  
15. La Imagen\_Raíz quedará de la siguiente manera.

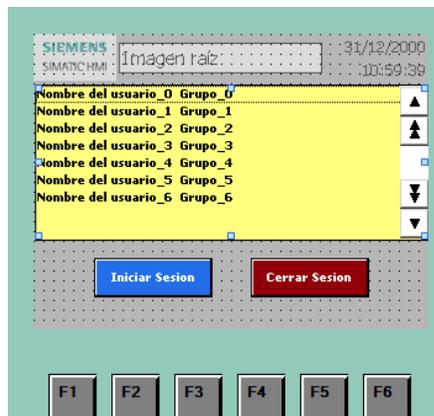


Figura P9.12: Imagen\_Raíz

16. Luego en la parte Izquierda dentro del equipo HMI dar doble clic en la categoría “Administración de usuarios”.

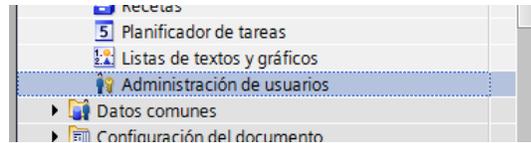


Figura P9.13: Abrir Administración de Usuarios

17. En la parte superior derecha de la ventana que se abre dar clic en la Pestaña “Grupo de Usuarios”. En esta parte se puede configurar diferentes grupos de usuarios según los privilegios que se desee manejar, en este caso se hará uso del Grupo “Usuarios” pre configurado por defecto.

Grupos				
Nombre	Número	Nombre de visualización	Caducidad de l...	Comentario
Grupo de administradores	1	Grupo de administradores	<input type="checkbox"/>	El grupo 'Administradores' tiene inicialmente todos los derechos
Usuarios	2	Usuarios	<input type="checkbox"/>	El grupo 'Usuarios' tiene inicialmente el permiso 'Operación'.

Figura P9.14: Grupos de Usuarios

18. Dar clic sobre cada grupo y en la parte inferior en la sección “Permisos” escoger las atribuciones de cada uno.

Grupo de administradores: Todos los permisos

Usuarios: Solo permiso de Operación

Permisos					
Activo	Nombre	Nombre de visualización	Número	Comentario	
<input type="checkbox"/>	Administración de usuari..	Administración de usuarios	1	Permiso 'Administración de ...	
<input type="checkbox"/>	Monitorización	Monitorización	2	Permiso 'Supervisar'.	
<input checked="" type="checkbox"/>	Operación	Operación	3	Permiso 'Operación'.	

Figura P9.15: Usuarios

19. En la parte superior derecha de la ventana dar clic en la Pestaña “Usuarios” y agregar un usuario con los siguientes datos:

Nombre: operador1

Contraseña: operar1

En la sección “Grupos” escoger que este usuario pertenece al grupo de nombre “Usuarios”.

Usuarios			
	Nombre	Contraseña	Cierre
	Administrador	*****	
	operador1	*****	
	inspector1	*****	
	<Agregar>		

Grupos			
	Miembro de	Nombre	Número
	<input type="radio"/>	Grupo de administradores	1
	<input checked="" type="radio"/>	Usuarios	2
	<input type="radio"/>	Inspectores	3

Figura P9.16: Usuario “Operador1”

20. Agregar segmento en PLC para la activación de una salida física desde la pantalla HMI mediante una marca del sistema.

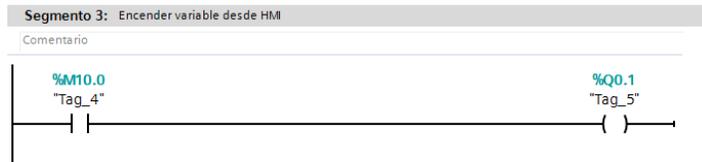


Figura P9.17: Segmento Encender Variable mediante marca M10.0

21. En la parte izquierda dentro de HMI y bajo la categoría “Imágenes” dar clic en “Agregar nueva imagen”.

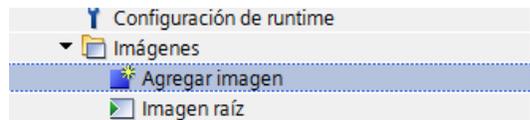


Figura P9.18: Agregar nueva Imagen

22. Similar al paso 11 agregar un botón de color naranja con el texto “Encender Salida”, además en la Pestaña “Eventos” asignarle una función “ActivarBitMientrasTeclaPulsada” cuya variable corresponderá a la marca M10.0

Hacer clic		
	ActivarBitMientrasTeclaPulsada	
Soltar	Variable (Entrada/salida)	Tag_4
Activar	Bit	0
Desactivar	<Agregar función>	
Cambio		

Figura P9.19: Eventos Botón “Encender Salida”

23. Dentro de la Pestaña “Propiedades” en la categoría “Seguridad” elegir un permiso de “Operación” para el botón.

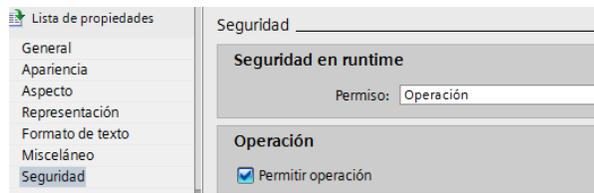


Figura P9.20: Seguridad Botón “Encender Salida”

24. Luego en la parte derecha en la paleta “Objetos Básicos” escoger y arrastrar un círculo hasta el área de trabajo, en la Pestaña “Animaciones” dentro de la sección “Visualización” dar clic en crear nueva animación de apariencia.

Luego en la ventana que se despliega escoger la variable Q0.1. En la parte inferior dar clic en “agregar” y elegir las propiedades para el objeto cuando la salida este encendida o apagada.

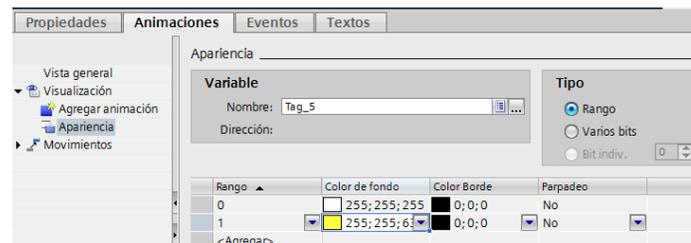


Figura P9.21: Animaciones Círculo

#### d. Avisos

25. En la parte derecha en la paleta “Controles” escoger y arrastrar un elemento “Visor de Avisos”.

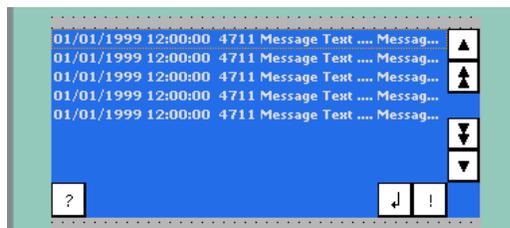


Figura P9.22: Visor de Avisos

26. En la parte izquierda dentro de HMI dar doble clic en “Avisos HMI”.

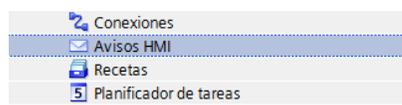


Figura P9.23: Abrir Avisos HMI

27. Luego dar clic sobre la Pestaña “Avisos de bit”, en el área de avisos dar clic en agregar y configurarlo de la siguiente manera.

ID: 1.

Texto de aviso: Salida Q0.0 activada.

Variable de disparo: estado

Bit de disparo: 0

Avisos de bit				
ID	Texto de aviso	Categoría	Variable de disp..	Bit de disparo
1	Salida Q0.0 Activada	Errors	estado	0

Figura P9.24: Aviso de bit

28. Dirigirse a la “Imagen\_Raíz” luego seleccionar el botón F1 de la pantalla ubicado en la parte inferior, en la Pestaña “Eventos” agregar la función “ActivarImagen” y escoger el nombre de imagen “Imagen\_1”.



Figura P9.25: Eventos botón F1 de Imagen\_Raíz

29. Dirigirse a la “Imagen1” luego seleccionar el botón F1 de la pantalla ubicado en la parte inferior, en la Pestaña “Eventos” agregar la función “ActivarImagen” y escoger el nombre de imagen “Imagen\_Raíz”.

30. Grabar ambos equipos

31. En la HMI dar física clic en el botón “Iniciar Sesión” e introducir los datos del usuario “operador1”.

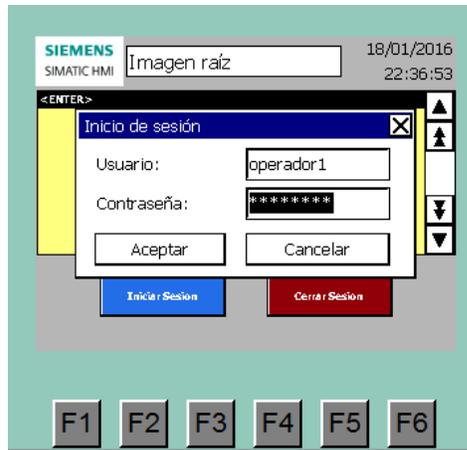


Figura P9.26: Pantalla Imagen Raíz en *Runtime*

32. Verificar que en el elemento “Visor de Usuarios” aparezca los datos luego de iniciar la sesión.
33. Luego Pulsar el botón F1 para cambiarse a la “Imagen1”, ahí pulsar el botón “Encender Salida” y verificar que se encienda Q0.1. Posteriormente activar la entrada I0.0.

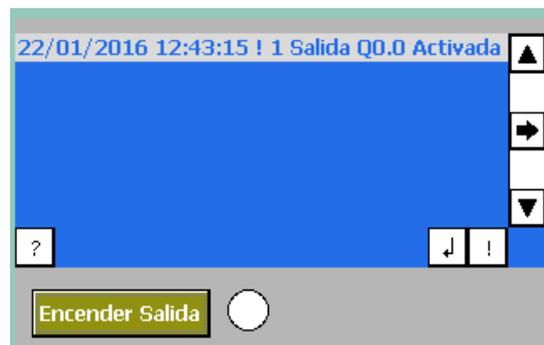


Figura P9.27: Pantalla Imagen1 en *Runtime*

34. Regresar a la venta principal pulsando el botón F1, luego pulsar en “Cerrar Sesión”.
35. Ir a la pantalla 2 y verificar que no se pueda pulsar el botón “Encender Salida” ya que tiene solamente permisos de operación.

## EJERCICIO PARA EL ALUMNO

En base a lo revisado acerca de la Utilización de Avisos y Administración de Usuarios de un Sistema SCADA, crear un proyecto que permita lo siguiente.

Tener dos Usuarios “Operador1” con clave “operar1” e “Inspector1” con contraseña “inspe1”, el primer usuario con capacidad de interactuar con la HMI y el segundo solo con acceso a monitorización.

Agregar un campo de texto y un botón “Encender Salida” con el fin de encender las salidas de la 1-4 según el valor escrito en el campo de texto, ambos controles deben ser solamente accesibles al usuario1.

Además del Aviso de bit configurado para la salida Q0.0, el alumno deberá agregar un nuevo Aviso para una Entrada Analógica con el fin de determinar si la tensión captada es mayor a 5V, el valor de voltaje deberá ser visualizado en la pantalla de la HMI mediante un campo de texto. (Conectar un potenciómetro a una de las entradas analógicas).

**CONCLUSIONES:**

## ANEXO 10: PRÁCTICA 10

**TEMA:** Uso de Recetas del Sistema SCADA WinCC.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la utilización de Recetas dentro de una aplicación SCADA sencilla.

### OBJETIVOS ESPECÍFICOS:

- Configurar el PLC y la Pantalla HMI.
- Configurar el sistema de Recetas para la aplicación.
- Verificar el funcionamiento.

### MATERIALES:

- Un PLC Simatic S7-1200.
- Una Pantalla HMI.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.

### FUNDAMENTO TEÓRICO:

#### Sistema SCADA

##### *Recetas en Wincc*

Las recetas son paquetes de datos que especifican determinadas configuraciones para valores de proceso, mediante ellas se puede cambiar todo un grupo de parámetros para los accionamientos.

Una receta se caracteriza por una estructura de datos fija. Durante la configuración se define la estructura de una receta. Una receta contiene registros que se distinguen por sus

valores, pero no por su estructura. Las recetas se almacenan en el panel de operador o en un medio de almacenamiento externo. Básicamente, un registro de receta se transfiere por completo en un paso de trabajo entre el panel de operador y el controlador. Además, se puede importar datos de producción en *runtime* a través de un archivo CSV (SIEMENS, WinCC Profesional V13 SP1, 2014).

De acuerdo a lo expuesto por (SIEMENS, WinCC Profesional V13 SP1, 2014) existen tres casos básicamente para la utilización de recetas, a continuación se describe cada uno de ellos.

- **Producción manual:** El usuario selecciona los datos de receta deseados y los visualiza en el panel de operador. En caso necesario, modifica los datos de receta y los guarda en el panel de operador. Luego transfiere los datos de receta al controlador.
- **Producción automática:** El programa de control inicia la transferencia de los datos de receta entre el controlador y el panel de operador. El usuario también puede iniciar la transferencia realizando una acción en el panel de operador. La producción cambia entonces de forma automática. No es imprescindible visualizar ni modificar los datos.
- **Modo "Teach-In":** El usuario optimiza manualmente los datos de producción en la instalación, p. ej. las posiciones de los ejes o los niveles de llenado. Los valores calculados se transfieren al panel de operador y se guardan en un registro de receta. Posteriormente, el usuario retransfiere al controlador los datos de receta almacenados.

**PROCEDIMIENTO:****a. Configuración del Proyecto y del Hardware**

1. Abrir TIA PORTAL V11 instalado en la PC.
2. En la pantalla principal elegir “Crear Proyecto” y llenar los campos “Nombre” y “Ruta” según se desee.
3. Dar clic en Abrir Vista del Proyecto
4. En el “Árbol de Proyecto” dar doble clic en “Agregar Dispositivos” para insertar el PLC1, en este caso será el modelo CPU 1214 AC/DC/RLY.
5. Luego en el “Árbol del Proyecto” dar doble clic en “Agregar Dispositivos” para insertar la pantalla HMI, en el Laboratorio se cuenta con una Pantalla Táctil de 6” KTP600 PN.
6. Luego dar clic en “Seleccionar PLC” y escoger el PLC al que se vinculará la pantalla, en este caso será PLC1.
7. Si se desea se puede realizar más configuraciones en el asistente, en este caso se escogerá los parámetros por defecto, por lo tanto dar clic en “Finalizar”.
8. Crear dos variables de tipo int que serán los dos valores de receta que se configurarán.

Para ello dentro del PLC1 dar doble clic sobre “Variables del PLC” y luego doble clic “Tabla de variables estándar”. Ahí colocar los siguientes datos.

Tabla de variables estándar						
	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...
	tiempo	Int	%MW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	numero	Int	%MW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura P10.1: Variables para Recetas

**b. Configuración de Recetas**

9. Dentro de la HMI dar doble clic en la categoría “Recetas”.
10. En la ventana que se abre, en la parte superior en la sección “Recetas” dar clic en “Agregar”, insertar dos recetas y modificar el campo nombre de visualización de la siguiente manera

Recetas						
Nombre	Nombre de visualizac..	Número	Versión	Tipo	Número máximo de re...	Tipo de comunicación
Receta_1	Yogurt	1	21/01/2016 19:..	Limitado	20	Variables
Receta_2	Leche	2	21/01/2016 19:..	Limitado	20	Variables

Figura P10.2: Creación de Recetas

11. Dar clic sobre “Receta1” y en la parte inferior de la venta en la Pestaña “Elementos” dar clic en agregar y colocar los siguientes datos.

Nombre: Tiempo

Nombre de Visualización: Tiempo de Vueltas (ms)

En el campo variable desplegar y escoger la variable del PLC creada previamente, deberá quedar de la siguiente manera.

Elementos				
Registros				
Nombre	Nombre de visualizac..	Variable	Tipo de datos	Longitud d...
Tiempo	Tiempo de Vuelta (ms)	tiempo	Int	2

Figura P10.3: Agregar Elementos a la Receta

12. Repetir el paso anterior para agregar una segunda variable con los datos que se pueden ver a continuación.

Elementos				
Registros				
Nombre	Nombre de visualizac..	Variable	Tipo de datos	Longitud d...
Tiempo	Tiempo de Vuelta (ms)	tiempo	Int	2
Numero	Numero de Vueltas	numero	Int	2

Figura P10.4: Elementos de Recetas

13. Luego dar clic en la Pestaña “Registros” y agregar un nuevo registro, en esta estructura se coloca los valores que tomaran las variables para la receta. Introducir en los campos “Tiempo” y “Número” introducir un valor adecuado dentro del rango de la variable.

Elementos				
Registros				
Nombre	Nombre de visualizac..	Número	Tiempo	Numero
Registro de receta_1	Variables	1	900	8

Figura P10.5: Registro de Receta 1

14. Dar clic sobre “Receta2” y repetir los pasos 11 y 12 para agregar las mismas variables a la receta.

15. Luego repetir el paso 13 para agregar un registro a la receta 2, los valores ingresados serán.

Elementos		Registros			
	Nombre	Nombre de visualizac..	Número	Tiempo	Numero
	Registro de receta_1	Variables de Receta	1	1600	5

Figura P10.6: Registro de Receta 2

### c. Visualización y Manejo de Recetas

16. Dentro de la HMI bajo la categoría “Imágenes” abrir la imagen principal en este caso de nombre “HMI”.

Posteriormente en la parte derecha de la paleta “Controles” arrastrar hacia el área de trabajo un elemento “Visor de Recetas”.

17. Luego dar clic en sobre el Visor de Recetas y expandir la Pestaña “Propiedades”, en la categoría “Apariencia” escoger un color de fondo, en este caso azul.

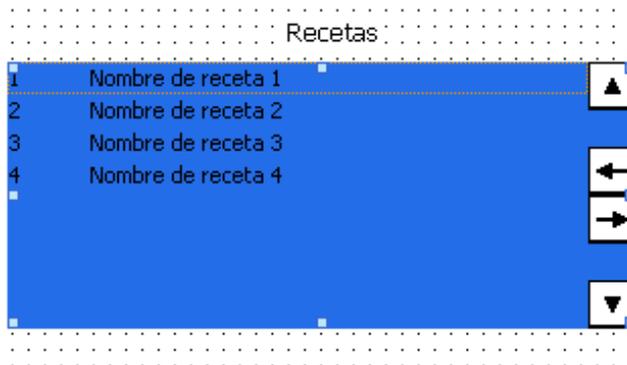


Figura P10.7: Visor de Recetas

18. En la categoría “General” del Visor activar la casilla de verificación para habilitar la edición de la receta en *Runtime*.

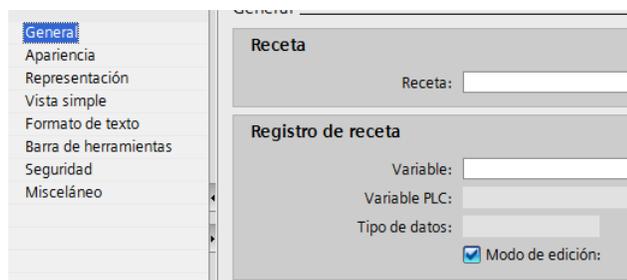


Figura P10.8: Edición de Receta en *Runtime*

19. En la categoría “Vista Simple” escoger algunas configuraciones de visualización como las filas de recetas y números de las mismas.

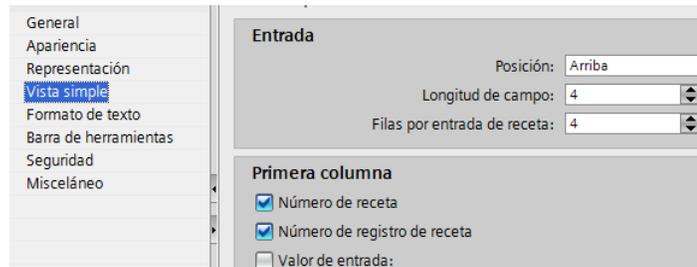


Figura P10.9: Configuraciones de Visualización

20. Dentro del PLC dar clic en la Categoría “Tablas de observación y forzado permanente”, luego dar clic en “Agregar nueva tabla de observación.” Dentro de la tabla colocar en el campo “Nombre” las variables “Tiempo” y “Numero”.

Nombre	Dirección	Formato visualiza..	Valor de observac..	Valor de forzado
"tiempo"	%MWO	DEC+/-		
"numero1"	%MWS	DEC+/-		

Figura P10.10: Agregar Tabla de Observación

#### d. Funcionamiento

21. Grabar ambos equipos.
22. En la Pantalla de la HMI deberá aparecer las dos recetas creadas dentro del “Visor de Recetas”, dar clic sobre el nombre de la Receta.

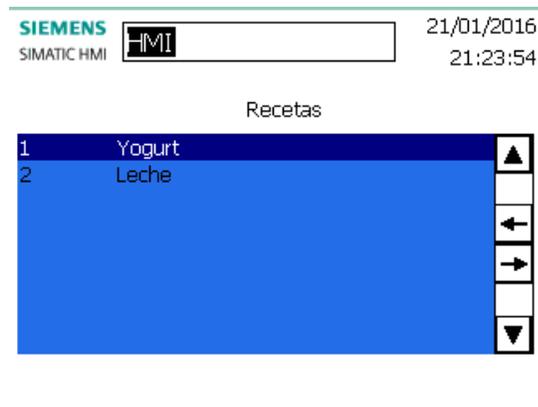


Figura P10.11: Recetas en HMI

23. Luego de desplegará otra vista con los registros almacenados para cada Receta, en este caso se tiene solamente un registro para cada receta, dar clic sobre el nombre.

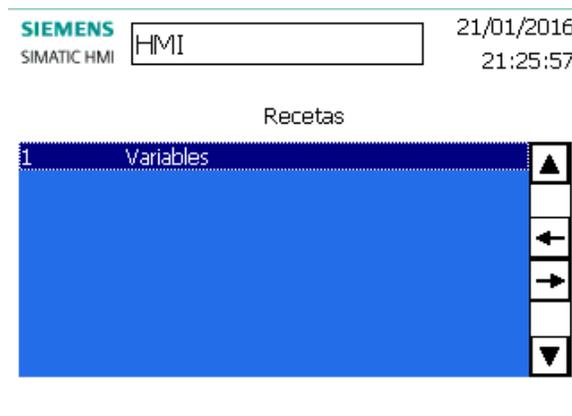


Figura P10.12: Registros de Receta en HMI

24. En la siguiente vista aparecerán las variables configuradas para la Receta con sus respectivos valores, dar clic en la flecha “Derecha” para acceder a las opciones de las variables.

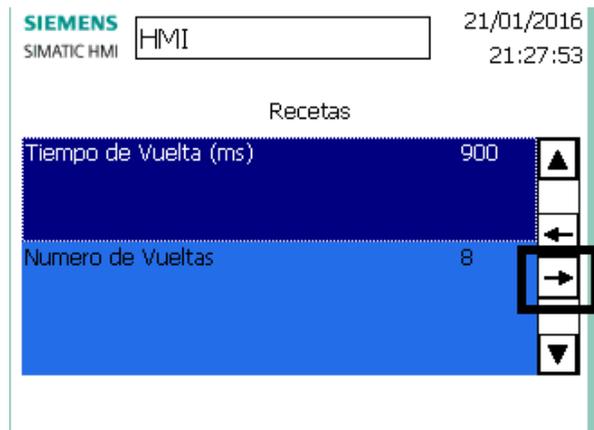


Figura P10.13: Variables de Receta

25. Luego dar clic en “hacia el controlador” para trasferir los valores hacia las variables del PLC.

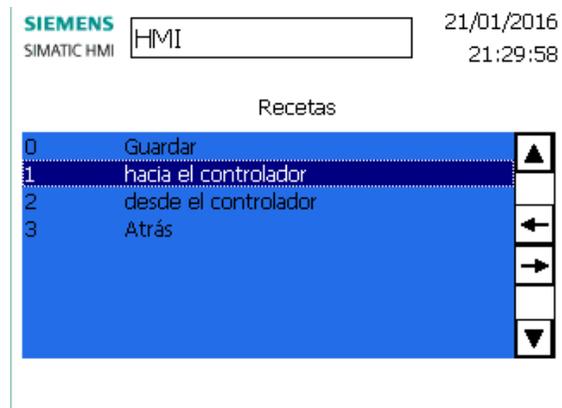


Figura P10.14: Transferir Variables hacia el controlador

26. Conectarse ONLINE con el PLC mediante la opción “Establecer Conexión Online” ubicada en la barra de herramientas de TIA PORTAL, observar los datos mediante la tabla de observación creada.

	Nombre	Dirección	Formato visualiza..	Valor de observac..	Va
1	*tiempo	%MWO	DEC+/-	900	
2	*numero1	%MWS	DEC+/-	8	

Figura P10.15: Tabla de Observación

## EJERCICIO PARA EL ALUMNO

En base a la utilización de Recetas crear un proyecto que permita tener dos Recetas con los siguientes elementos:

### *Receta 1*

- Tiempo de Vuelta (ms): 900
- Numero de Vueltas: 8
- Tensión de Salida (%): 70

### *Receta 2*

- Tiempo de Vuelta (ms):1600
- Numero de Vueltas: 5
- Tensión de Salida (%): 20

Utilizar estas variables de receta para crear un programa que sea capaz de contar un número de pasos determinado por el parámetro “Número” con un intervalo de tiempo establecido por la variable “Tiempo”.

Crear una nueva imagen en la pantalla HMI y colocar un botón “Iniciar” para empezar la secuencia y un indicador luminoso para visualizar el término de la misma, además colocar campos de texto para visualizar los valores de las variables de la Receta. Adicionalmente usar la variable “Tensión de Salida” para establecer el valor de voltaje de la salida analógica del PLC, medir la tensión.

### **CONCLUSIONES:**

## ANEXO 11: PRÁCTICA 11

**TEMA:** Red de Comunicaciones Industriales entre cuatro PLCs Simatic S7-1200 utilizando los protocolos: Profinet, Profibus, Modbus-RTU y Modbus-TCP.

**OBJETIVO:** Crear un proyecto en TIA PORTAL para la programación y comunicación de todo un sistema de automatización conformado por cuatro PLCs como resumen de todos los temas abordados en las prácticas

### OBJETIVOS ESPECÍFICOS:

- Configurar los cuatro PLCs para el uso de cada red de comunicación.
- Programar los controladores según los requerimientos especificados posteriormente.
- Utilizar los temas abordados en las prácticas anteriores para configurar las aplicaciones SCADA.
- Comprobar el funcionamiento de todas las redes.

### MATERIALES:

- Cuatro PLCs Simatic S7-1200.
- Tres Pantallas HMI.
- *Software* TIA PORTAL.
- Un cable *Ethernet* TP, CAT5.
- Un cable Profibus 2 hilos, conectores interfaz RS-485.
- Módulo Maestro-DP Profibus CM 1243-5.
- Módulo Esclavo-DP Profibus CM 1242-5.
- Un motor DC.
- Un Mosfet.
- Un potenciómetro.
- Dos Resistencias R.

- Un cable apantallado de 2 hilos, conectores interfaz RS-485.
- Dos módulos de Comunicación CM 1241.
- Dos motores trifásicos
- Prototipo de laboratorio para el control de nivel del Tanque.

**EJERCICIO PARA EL ALUMNO**

En base a lo revisado en todas las prácticas de laboratorio el alumno deberá realizar un proyecto que englobe todos los temas de la siguiente manera. Se deberá configurar cuatros PLCs (dos de ellos deberán ser de tipo DC/DC/DC) y tres pantallas HMI desplegadas de la siguiente forma.

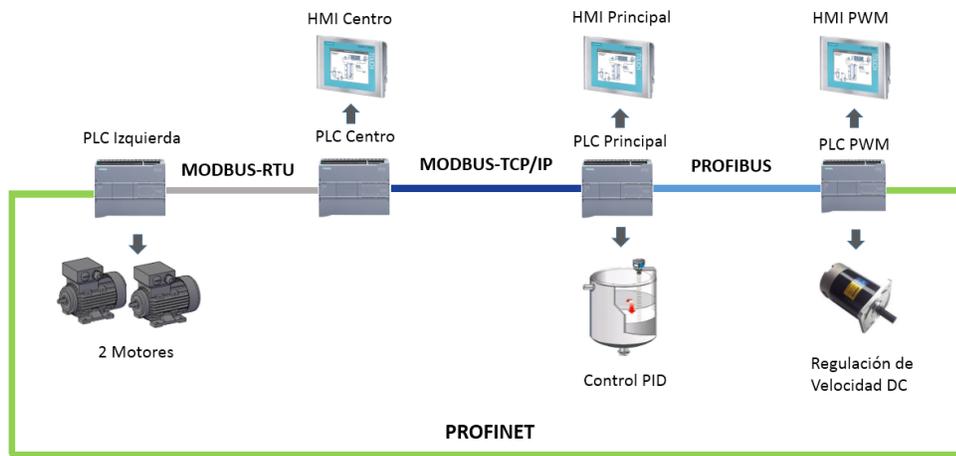


Figura P11.1: Esquema Práctica 10

En la siguiente tabla se realiza un resumen del hardware y rol de cada PLC dentro de la red.

Tabla P11.1: Resume Hardware Práctica 11

Denominación	Tipo	Hardware	Rol en la Red
PLC Principal	DC/DC/DC	1 Módulo CM 1243	Maestro Profibus Maestro Modbus-TCP/IP
PLC PWM	DC/DC/DC	1 Módulo CM 1242	Esclavo Profibus Receptor Profinet

<b>PLC Centro</b>	Cualquiera	1 Módulo CM 1241	Maestro Modbus-RTU Esclavo Modbus-TCP/IP
<b>PLC Izquierda</b>	Cualquiera	1 Módulo CM 1241	Esclavo Modbus-RTU Emisor Profinet

Los requerimientos de control son los siguientes:

### PLC Principal:

- Se encargará del control PID del nivel de líquido y del control de velocidad automático de un motor DC conectado al PLC PWM.
- Red Modbus-TCP/IP: El PLC deberá enviar los datos del nivel del tanque, y tres bits de alarma para niveles: alto (mayor a 70%), medio (entre 40 y 70 %) y bajo (menor a 40%).
- Red Profibus: El PLC deberá leer el valor de una entrada analógica y enviar el dato correspondiente para el control de velocidad siendo 100% igual a 10V y 0% igual a 0V, además deberá enviar dos bits para el arranque y paro del motor.

### HMI Principal:

- Esta enlazado al PLC principal y contiene tres pantallas: una inicial, una para el control PID y otra para el control PWM sobre el PLC vecino. Las pantallas deberían contener:

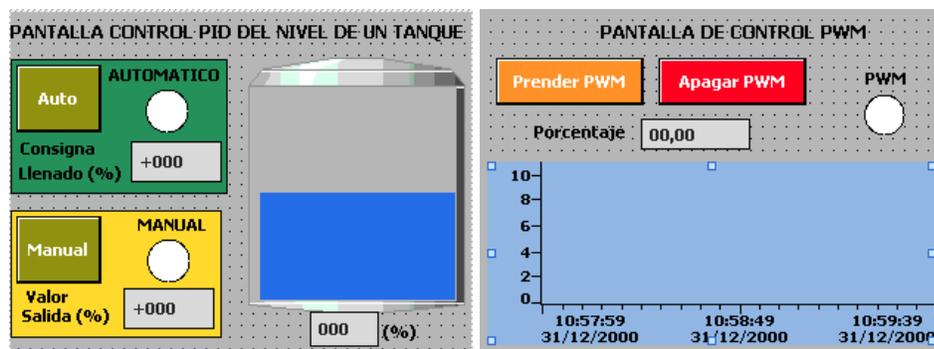


Figura P11.2: Pantallas HMI Principal

**PLC PWM:**

- Se encargará de la activación PWM sea manual o automática, de recibir y presentar los datos vía Profinet.

**HMI PWM:**

- Esta enlazado al PLC PWM y contiene tres pantallas: una inicial, una para el monitoreo-control PWM y otra para la observación de los motores trifásicos del PLC Izquierda. Además deberá manejar una gestión de Usuarios para la activación manual o automática del PWM. Las pantallas deberían contener:

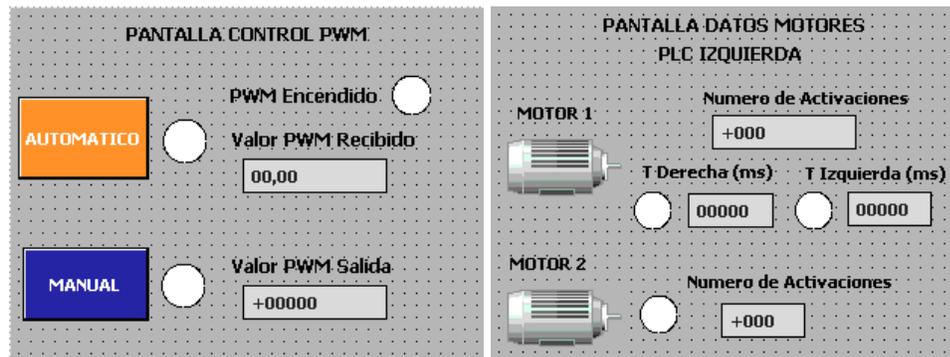


Figura P11.3: Pantallas HMI PWM

**PLC Central:**

- Se encargará del control remoto de dos motores trifásicos conectados al PLC Izquierda.
- Red Modbus-RTU: El PLC deberá enviar los datos de tiempo de giro izquierda y derecha para el motor 1 así como su correspondiente bit de activación, además deberá enviar dos bits para encendido y apagado del motor 2.

**HMI Central:**

- Esta enlazado al PLC Central y contiene tres pantallas: una inicial, una para el monitoreo del control PID y otra para el control de los motores trifásicos. Además

se deberá colocar un visor de avisos para registrar los cambios en las zonas: alta, media y baja del tanque. Las pantallas deberían contener:

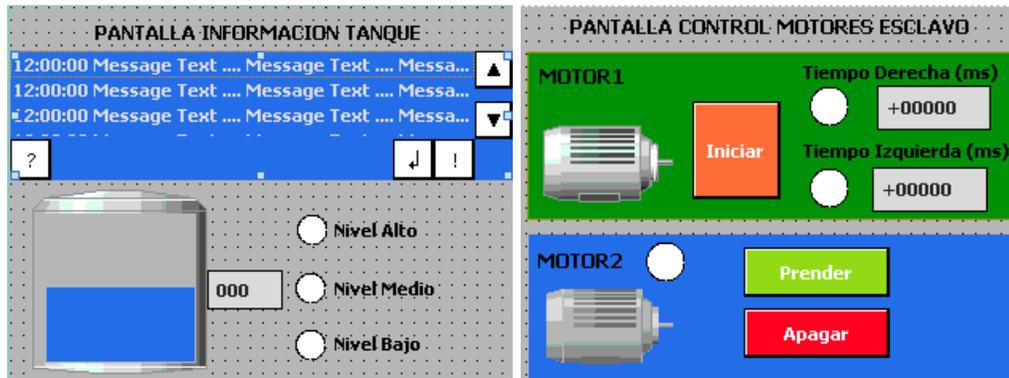


Figura P11.4: Pantallas HMI Centro

### PLC Izquierda:

- Está a cargo de la activación de los motores trifásicos conectados y de enviar los datos vía Profinet al PLC PWM.
- Red Profinet: El PLC deberá enviar los datos de los motores como tiempos de activación, bits de estado (encendido o apagado) y el número de activaciones que han tenido hasta la fecha

### CONCLUSIONES: