

# Implementación de un sistema de monitoreo y control de actuadores eléctricos AUMA utilizando protocolos de comunicación industrial Modbus RTU y Modbus TCP/IP

*Ricardo Defas*

Universidad Central del Ecuador  
ricardodefás@gmail.com

*Andrés Guzmán*

AUMA Andina&Centroamérica  
guzmanh.andres@gmail.com

*Recibido:* 01 de septiembre / *Aprobado:* 30 de noviembre 2017

## Resumen

El presente artículo propone el diseño e implementación de un sistema de monitoreo de actuadores eléctricos para la empresa AUMA Andina&Centroamérica con el fin de supervisar todas las variables de control provenientes de una red de actuadores eléctricos diseñados con interfaces de comunicación Modbus RTU. Los enlaces de comunicación son entablados haciendo uso de buses de campo sobre los cuales se controla los procesos que intervienen en la automatización de actuadores eléctricos. El equipo principal utilizado para gestionar todos los enlaces de comunicación es el controlador SIMATIC S7-1200 de Siemens, al cual se le incluye las tarjetas de comunicación CM 1241 para interface Modbus, con lo cual se logra el intercambio de datos de forma serial a través del estándar RS-485. La programación del proceso y la configuración de los protocolos industriales se realiza a través de las herramientas y librerías del software TIA Portal V13, y como herramienta de visualización de resultados del proceso de control se utiliza la pantalla Red Lion G306A (HMI) que se programa mediante el software Crimson 3.0. Todos los equipos antes mencionados se integran en un tablero de control de baja potencia diseñado bajo las normas

IEC 61439-1 e IEC 61439-2, el cual incluye todos los elementos de maniobra y protección necesarios para un proceso de control.

**Palabras Claves:** redes de campo industrial, AUMA, Modbus, Profibus.

### **Abstract**

The present article proposes the design and implementation of a monitoring system for electric actuators for the company AUMA Andina & Centroamérica in order to supervise all control variables of a field network of electric actuators designed with Modbus RTU communication interfaces. The communication links are made using field buses on which the processes involved in the automation of electric actuators are controlled. The device used to manage all communication links is the Siemens SIMATIC S7-1200 controller, which includes the communication cards CM 1241 for Modbus interface and CM 1243-5 for Profibus achieving the exchange of serial data through the RS-485 standard. The programming of the process and the configuration of the industrial protocols is done through the tools and libraries of the TIA Portal V13 software, and as a tool for visualizing the results of the control process the touch screen Red Lion G306A (HMI) is used, through the Crimson 3.0 software. All of the equipment mentioned above is integrated in a low power control board designed in accordance with standards IEC 61439-1 and IEC 61439-2, which includes all the maneuvering and protection elements required for a control process.

**Keywords:** industrial field networks, AUMA, Modbus, Profibus.

## Introducción

**A**UMA, una empresa dedicada al diseño, fabricación y venta de actuadores eléctricos desde 1964 a nivel mundial, con presencia en Ecuador desde hace aproximadamente 4 años, busca desarrollar una plataforma que permita mostrar a los usuarios la operación de los actuadores eléctricos sin necesidad de estar en contacto directo con los mismos, sino de establecer un control a largas distancias.

Actualmente, la automatización industrial es un medio fundamental para mejorar el rendimiento y eficacia en todos los procesos industriales. Cuenta con numerosas ventajas como: aumentar la productividad de los recursos en las actividades manuales, obtener la información de un proceso de manera óptima, transmitir datos en tiempo real y reducir los costos de instalación y de equipos; esto se logra mediante el uso de buses de campo.

La implementación del módulo de comunicación de este proyecto busca cumplir los objetivos de la empresa AUMA, y lograr a través de un protocolo de comunicación muy eficiente, MODBUS RTU, la verificación de todas las variables que registran los actuadores eléctricos en un proceso industrial específico.

El módulo de comunicación monitorizará la información proveniente de una serie de actuadores eléctricos conectados en una línea de bus, tales como la posición de cada actuador o el porcentaje de apertura o cierre del mismo. El módulo obtendrá las señales de alarma o falla por temperatura, torque o error de comunicación. Esto se visualizará desde una interfaz gráfica, HMI.

## Materiales y métodos

### Fundamentación teórica

#### *Protocolo Modbus*

Modbus es un protocolo de mensajería de la capa de aplicación, es decir que ocupa el nivel 7 del modelo OSI. Tiene como principal función permitir la comunicación entre dispositivos de campo y control que estén conectados en diferentes tipos de buses o redes [1].

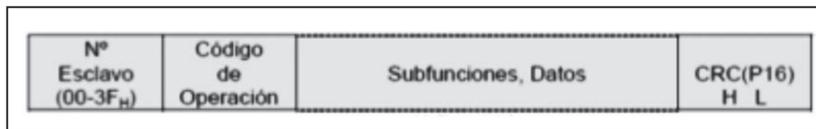
Entre las características que posee Modbus, podemos mencionar las siguientes [2]:

- Transmisión en la arquitectura Cliente/Servidor.
- El mecanismo de acceso al medio, a nivel de capa de enlace, es Maestro/Esclavo.
- Dispone desde 1 hasta 247 esclavos.
- El método de transmisión es Unicast.
- Usa códigos de funciones para enviar la operación a realizar por el esclavo.

- Protocolo público, compatible con varios dispositivos.

Modbus presenta dos variantes para la transmisión de datos: ASCII y RTU; de los cuales el protocolo Modbus RTU (*Remote Terminal Unit*) es el más usado, ya que presenta mayor rendimiento en la transmisión de datos, puesto que utiliza un sistema de codificación binario; respecto a ASCII [3].

La trama y codificación que usa Modbus RTU se muestra en la figura 1.



*Figura 1.* Codificación RTU [4].

Donde:

- Número de esclavo: Número de destinatario.
- Código de operación: Este campo envía qué operación se desea que realice el esclavo.
- Dirección, datos y subfunciones: En este campo se envía la información para ejecutar el código de operación seleccionado anteriormente.
- Control de errores CRC (Chequeo de Redundancia Cíclica): Dependiendo el modo de transmisión, se usa el sistema de detección de errores.

### *Protocolo Profibus DP*

Profibus es un estándar de comunicaciones para el intercambio de datos a alta velocidad, está basado en la norma europea EN50170 y actualmente es considerado como el bus de campo más veloz para transmisión de datos [5].

El sufijo DP se establece por Decentralized Periphery, y define a aquellos dispositivos de campo, como sensores y actuadores, que están conectados mediante módulos de interfaces remotas (IM) al controlador principal (PLC) a través de un simple conductor de dos hilos (RS-485).

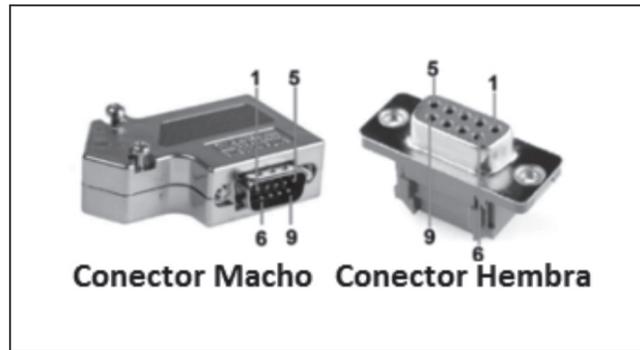
### *Estándar RS-485*

RS-485 es un estándar de comunicaciones de bus de campo muy sencillo y de bajo costo de instalación. Comprende aquellas aplicaciones en las cuales se requiere alta tasa de transmisión de datos. Las características estándar son:

- *Cables*: Utiliza un par trenzado de cobre que admite comunicación half-duplex, incluye apantallamiento.
- *Velocidad de transmisión*: Se puede alcanzar velocidades de transmisión desde 9.6

kbps hasta 1.2 Mbps. La longitud máxima del medio de transmisión dependerá de la tasa de transmisión.

- *Conectores:* Físicamente en la red los equipos Profibus y Modbus son suplidos con conectores estándar sub-D de 9 pines (DB9). Las diferencias de conector macho y hembra se pueden apreciar en la figura 2.



*Figura 2.* Conectores DB9 macho y hembra.

Los pines que se usa del conector DB9 para la comunicación se describen en la tabla 1.

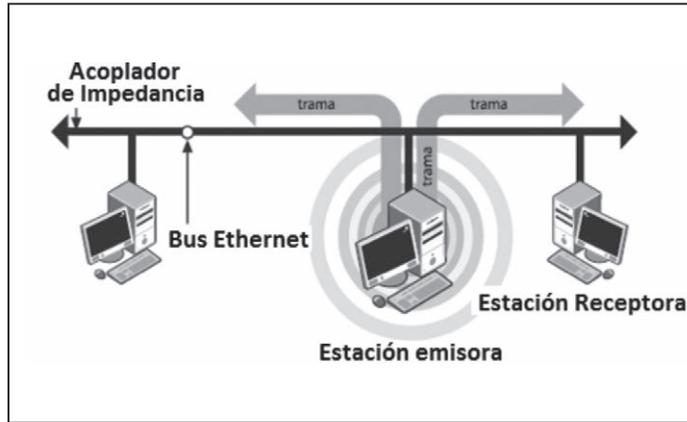
*Tabla 1.* Pines de conector DB9 profibus [6]

| Pin | Nombre     | Función                           |
|-----|------------|-----------------------------------|
| 3   | B-Line (+) | RxD/TxD positivo                  |
| 5   | GND        | 0 (V) aislado del circuito RS-485 |
| 8   | A-Line (-) | RxD/TxD negativo                  |

### *Topologías de red*

La topología de red es la forma física en la que están conectados todos los equipos que conforman la red. En la industria se cuenta con varios tipos de topologías de las cuales se puede mencionar: red de anillo y red en estrella; pero una de las redes más utilizadas en aplicaciones de campo es la red en bus.

Como se puede observar en la figura 3, una red en bus tienen la característica de poseer un solo camino a través del interface físico.



*Figura 3.* Red con topología en bus [7].

Entre los equipos que se conectan a la red, se puede mencionar ciertos equipos de medición como: sensores, transductores y actuadores; y equipos de control como PLC. Los actuadores eléctricos AUMA se consideran como dispositivos esclavos en la red de campo industrial, por lo tanto, serán descritos brevemente a continuación.

#### *Actuadores eléctricos*

Los actuadores eléctricos AUMA, como se observa en la figura 4, con una combinación de motor eléctrico y reductor, son desarrollados para automatizar válvulas. Estos transmiten el torque necesario para accionar una válvula de compuerta, de mariposa, de bola o de otro tipo.

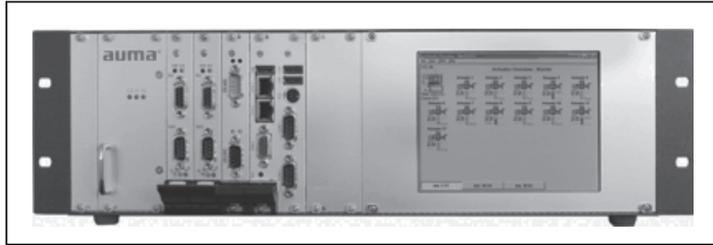
Estos tipos de actuadores poseen un control, el cual registra datos de carrera y torque de la válvula. Este control es integrado con el protocolo de comunicación requerido para el respectivo proceso; mediante esta característica, el actuador eléctrico AUMA puede ser comandado remotamente desde el puesto de mando.



*Figura 4.* Actuadores eléctricos AUMA.

## SIMA

La SIMA, equipo que forma parte de AUMA, es una estación maestra ideal para poder integrar actuadores a un sistema automatizado basado en protocolos de comunicación. Como se puede ver en la figura 5 [8].



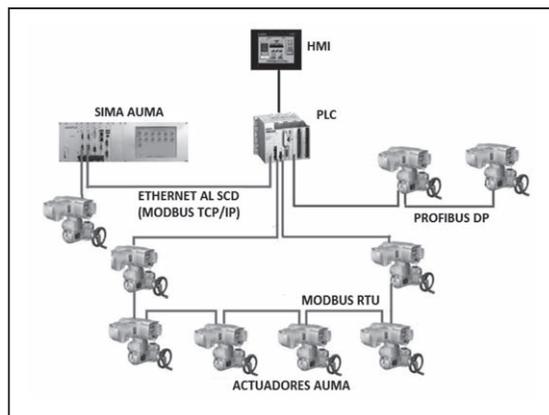
*Figura 5.* Estación maestra SIMA [8].

## Diseño del sistema de comunicación

### Arquitectura de red

La implementación y configuración de la red de comunicación parte de la conceptualización de una arquitectura de red, tal como se muestra en la figura 6. Tomando en cuenta los niveles que existen en un sistema de control, se puede mencionar los equipos requeridos para el proceso:

- A nivel de instrumentación de campo se cuenta con los esclavos de la red, conformado con varios actuadores eléctricos con interfaz Modbus, además de la SIMA.
- A nivel de control es necesario contar con un dispositivo programable PLC, que, como equipo principal, gestionará la comunicación con los demás equipos de la red.
- A nivel de supervisión, es preciso configurar una pantalla gráfica HMI que permitirá observar en tiempo real el estado de las variables del proceso, y permitirá enviar comandos hacia los demás dispositivos de la red.



*Figura 6.* Arquitectura del sistema de comunicación.

A partir de esta información se obtienen los protocolos de comunicación necesarios para interconectar los equipos que constituyen el sistema de control, así:

- Protocolo Modbus RTU tipo serial para comunicación entre PLC y los actuadores eléctricos.
- Protocolo Modbus TCP/IP para comunicación entre PLC y el equipo de visualización (HMI) y SIMA.

## Selección de equipos

### *Dispositivo lógico programable*

Saber elegir el autómatas adecuado es un aspecto fundamental para que el proyecto alcance su objetivo. Por lo tanto, se dimensiona el PLC de tal forma que brinde todas las prestaciones del caso, y que incluya solo las funcionalidades necesarias para el control del proceso, de forma que este no encarezca.

La selección del PLC y sus características se realiza atendiendo a los siguientes requerimientos:

- Que soporte y permita manejar distintos protocolos de comunicación Modbus de forma serial (RS-485) y TCP/IP (cable Ethernet).
- Que tenga suficiente capacidad de programa y memoria.
- Que presente escalabilidad, de tal forma que permita añadir módulos de entradas/salidas y tarjetas de comunicación.

El autómatas seleccionado es el SIMATIC S7-1200 de Siemens con el CPU 1212C, tal como se muestra en la figura 7.



*Figura 7.* CPU 1212C [9].

Entre las principales características con que cuenta este autómata se mencionan las siguientes:

- Posee una memoria de trabajo de 50 (KB) y una memoria de carga de 1 (MB), las cuales son suficientes para un proceso eficiente.
- Tiene integradas 8 entradas digitales a 24 VDC y 2 entradas de voltaje analógicas, además de 6 salidas tipo relé.
- Permite la conexión de hasta 2 módulos de señales de entrada y salida, sin necesidad de alimentación externa para cada una, salvo casos excepcionales.
- Tiene capacidad de conexión de hasta 3 módulos de comunicación.
- El ciclo de proceso es de 0.04 (ms) por cada 1.000 instrucciones.

#### *Tarjetas de comunicación*

Para lograr la comunicación directa desde el PLC S7-1200 hacia los actuadores eléctricos es necesario añadir una tarjeta de comunicación al autómata de tal forma que permita manejar el protocolo de comunicación Modbus RTU a través del estándar RS-485.

El módulo de comunicación seleccionado es CM 1241 tal como se indica en la figura 8. Este módulo trabaja bajo el estándar RS-485 y permite incorporar el protocolo de comunicación Modbus RTU.



**Figura 8.** Módulo de comunicación CM 1241 [10].

A continuación se describen las características que presenta la tarjeta de comunicación:

- No requiere de alimentación externa. Al conectarse con el CPU 1212C obtiene alimentación directa del PLC.

- Norma de interface, RS 422/485.
- Puede conectar un equipo (actuador) hasta una distancia de 1.000 m [30].
- Protocolos soportados: Modbus RTU y ASCII.

### *Equipo de visualización HMI*

Existen ciertos factores que determinan qué HMI es adecuado para cierta aplicación, entre los que se puede mencionar: resolución, memoria del proyecto de visualización, cantidad de variables y cantidad de pantallas. También es importante considerar que la comunicación con el S7-1200 se realizará a través el protocolo Modbus TCP/IP, por lo tanto, una de las condiciones que debe cumplir el equipo es que tenga integrado un puerto de comunicación Ethernet en su interface física. Debido a esto, el hardware escogido para la implementación del HMI es la pantalla gráfica Red Lion G306A, como se muestra en la figura 9.



**Figura 9.** Pantalla Red Lion G306A.

Las características técnicas de la pantalla se mencionan a continuación:

- Botones físicos frontales para configuración de “menús”.
- Manipulación táctil en la pantalla.
- Puerto Ethernet 10 base T/100 base-TX para configuración en red.
- Puerto USB para cargar el programa desde el PC.
- Capacidad de insertar una flash compacta para incrementar la capacidad de memoria.

- Alimentación de 24 VDC, con tolerancia de 20%.
- Memoria no volátil de 8 Mbyte, expandible mediante flash compacta.

Establecidos el hardware para visualización, el HMI podrá cumplir con las funciones para las que es diseñado, tales como:

- Comunicación con los dispositivos en campo.
- Visualización de variables a través de objetos animados.
- Enviar y análisis las señales para el proceso.
- Control limitado de ciertas variables del proceso.

### Comunicación mediante protocolo Modbus RTU

#### *Control de actuadores. Registros de entrada y salida para Modbus*

El control sobre los actuadores se realiza a través de la manipulación de registros de datos almacenados en la memoria interna de la interface Modbus. Se sabe que un registro de datos puede almacenar distintos tipos de números, que pueden ser: enteros, flotantes, constantes, entre otros; y que para ser accedidos necesitan contar con una dirección específica en la memoria de la interfaz. De esta forma, un actuador Modbus puede ser manipulado a través de los registros que se muestran en la tabla 2 y tabla 3, que, como se observa, son del tipo palabra (word) y por lo tanto, constan de 16 bits de datos.

**Tabla 2.** Registros de salida [11]

| Offset (Decimal)             | Contenido de los registros  |                              |        |        |       |                 |              |                                 |       |                                 |       |       |       |       |       |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |        |        |        |        |       |                 |              |             |       |       |       |       |       |       |       |       |
|------------------------------|---|------------------------------|--------|--------|-------|-----------------|--------------|---------------------------------|-------|---------------------------------|-------|-------|-------|-------|-------|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------|--------|--------|--------|--------|-------|-----------------|--------------|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1000                         | <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="8">Byte 1: Commands (High byte)</th> <th colspan="8">Byte 2: E3 (Reserve) (Low byte)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"> </td> </tr> <tr> <td style="text-align: center;">Bit 15</td> <td style="text-align: center;">Bit 14</td> <td style="text-align: center;">Bit 13</td> <td style="text-align: center;">Bit 12</td> <td style="text-align: center;">Bit 11</td> <td style="text-align: center;">Reset</td> <td style="text-align: center;">Remote SETPOINT</td> <td style="text-align: center;">Remote CLOSE</td> <td style="text-align: center;">Remote OPEN</td> <td style="text-align: center;">Bit 7</td> <td style="text-align: center;">Bit 6</td> <td style="text-align: center;">Bit 5</td> <td style="text-align: center;">Bit 4</td> <td style="text-align: center;">Bit 3</td> <td style="text-align: center;">Bit 2</td> <td style="text-align: center;">Bit 1</td> <td style="text-align: center;">Bit 0</td> </tr> </tbody> </table> | Byte 1: Commands (High byte) |        |        |       |                 |              |                                 |       | Byte 2: E3 (Reserve) (Low byte) |       |       |       |       |       |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Reset | Remote SETPOINT | Remote CLOSE | Remote OPEN | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Byte 1: Commands (High byte) |   |                              |        |        |       |                 |              | Byte 2: E3 (Reserve) (Low byte) |       |                                 |       |       |       |       |       |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |        |        |        |        |       |                 |              |             |       |       |       |       |       |       |       |       |
|                              |   |                              |        |        |       |                 |              |                                 |       |                                 |       |       |       |       |       |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |        |        |        |        |       |                 |              |             |       |       |       |       |       |       |       |       |
| Bit 15                       | Bit 14  | Bit 13                       | Bit 12 | Bit 11 | Reset | Remote SETPOINT | Remote CLOSE | Remote OPEN                     | Bit 7 | Bit 6                           | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |        |        |        |        |       |                 |              |             |       |       |       |       |       |       |       |       |
| 1001                         | <p style="text-align: center;">Byte 3: Set Point position (<i>High byte</i>)<br/>                     Byte 4: Set Point position (<i>Low byte</i>)</p>  |                              |        |        |       |                 |              |                                 |       |                                 |       |       |       |       |       |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |        |        |        |        |       |                 |              |             |       |       |       |       |       |       |       |       |

A través de la manipulación de los *registros de salida* de la tabla 2, se lleva a cabo solamente uno de los cuatro comando permitidos en el actuador:

- Apertura (*Remote Open*)
- Cierre (*Remote Close*)
- Alcanzar Set Point (*Remote Set Point*)
- Paro (*Reset*)

El registro con dirección 1000 indica la operación a realizar. Por ejemplo, para abrir al actuador se deberá setear el valor de 1L (uno lógico) en el bit 9 de este registro y automáticamente el actuador realizará la acción.

El registro 1001 se utiliza para configurar un valor de consigna (Set Point), en caso que el operador desee realizar la operación “Alcanzar Set Point”. Este valor deberá ser almacenado previamente en este registro como un número entero entre 0 y 1000 que corresponde al porcentaje de apertura o cierre entre 0% (totalmente cerrado) a 100% (totalmente abierto).

**Tabla 3.** Registros de entrada [11]

| Offset (Decimal)       | Contenido de los registros  |                        |              |                |                  |                 |               |                          |            |                          |            |                    |                     |               |               |  |  |            |              |               |              |                |                  |                 |               |           |            |           |            |                    |                     |               |               |
|------------------------|---|------------------------|--------------|----------------|------------------|-----------------|---------------|--------------------------|------------|--------------------------|------------|--------------------|---------------------|---------------|---------------|--|--|------------|--------------|---------------|--------------|----------------|------------------|-----------------|---------------|-----------|------------|-----------|------------|--------------------|---------------------|---------------|---------------|
| 1000                   | <table border="1"> <thead> <tr> <th colspan="8">Byte1: Logical signals</th> <th colspan="8">Byte 2: Actuator signals</th> </tr> </thead> <tbody> <tr> <td>Fault ind.</td> <td>Warning ind.</td> <td>Running CLOSE</td> <td>Running OPEN</td> <td>Not ready ind.</td> <td>Setpoint reached</td> <td>Closed position</td> <td>Open position</td> <td>TSC (DSR)</td> <td>TSO (DOEL)</td> <td>LSC (WSR)</td> <td>LSO (WOEL)</td> <td>Local sw. position</td> <td>Remote sw. position</td> <td>Loss of phase</td> <td>Thermal fault</td> </tr> </tbody> </table> | Byte1: Logical signals |              |                |                  |                 |               |                          |            | Byte 2: Actuator signals |            |                    |                     |               |               |  |  | Fault ind. | Warning ind. | Running CLOSE | Running OPEN | Not ready ind. | Setpoint reached | Closed position | Open position | TSC (DSR) | TSO (DOEL) | LSC (WSR) | LSO (WOEL) | Local sw. position | Remote sw. position | Loss of phase | Thermal fault |
| Byte1: Logical signals |   |                        |              |                |                  |                 |               | Byte 2: Actuator signals |            |                          |            |                    |                     |               |               |  |  |            |              |               |              |                |                  |                 |               |           |            |           |            |                    |                     |               |               |
| Fault ind.             | Warning ind.  | Running CLOSE          | Running OPEN | Not ready ind. | Setpoint reached | Closed position | Open position | TSC (DSR)                | TSO (DOEL) | LSC (WSR)                | LSO (WOEL) | Local sw. position | Remote sw. position | Loss of phase | Thermal fault |  |  |            |              |               |              |                |                  |                 |               |           |            |           |            |                    |                     |               |               |
| 1001                   | Byte 3: E2 Actual position (high byte) Byte 4: E2 Actual position (low byte)  |                        |              |                |                  |                 |               |                          |            |                          |            |                    |                     |               |               |  |  |            |              |               |              |                |                  |                 |               |           |            |           |            |                    |                     |               |               |
| 1002                   |   |                        |              |                |                  |                 |               |                          |            |                          |            |                    |                     |               |               |  |  |            |              |               |              |                |                  |                 |               |           |            |           |            |                    |                     |               |               |

En la tabla 3 se muestran los *registros de entrada* de los actuadores, los cuales corresponden a la información general de cada uno de ellos, esto es: estado actual del actuador (1002), el estado de sensores internos, recorrido actual (registro 1001), y también da acceso a indicaciones de falla y alarmas (registro 1000).

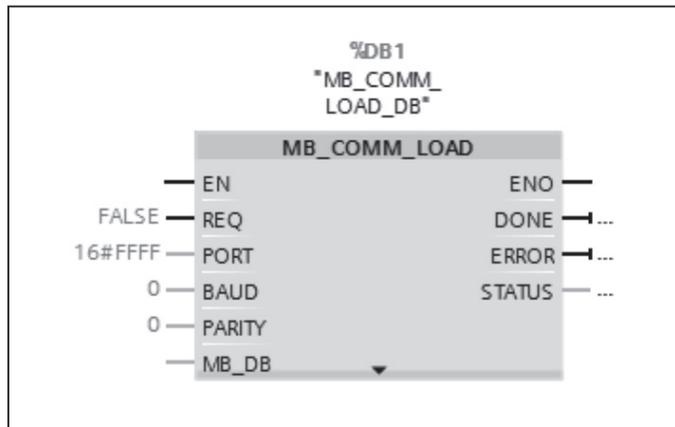
Con esta información, y dependiendo de la acción que se desee realizar sobre los actuadores, se deberá implementar las funciones pertinentes para lectura de registros de entrada o escritura de registros de salida en el autómata programable PLC y de esta forma controlar a los actuadores de forma oportuna.

### *Configuración de equipos: PLC S7-1200 y módulo de comunicación CM 1241*

El primer paso para construir la red Modbus, desde el S7-1200, es realizar la con-

figuración del módulo de comunicación CM-1241. Esto se logra a través de la instrucción MB\_COMM\_LOAD. Posteriormente, para la operación de este dispositivo como maestro RTU en la red, se hace uso de la instrucción MB\_MASTER.

***MB\_COMM\_LOAD***: Se configura a través de esta instrucción el puerto de comunicación del CM 1241 para transmitir mediante Modbus RTU. En la figura 10 se muestra el bloque de instrucción mencionado.



**Figura 10.** Instrucción MB\_COMM\_LOAD.

Este bloque tiene como funciones:

- Seleccionar el puerto del módulo de comunicación.
- Establecer los parámetros de transmisión para la comunicación.
- Definir si operará como maestro o esclavo.

En la tabla 4 se describen los parámetros y los tipos de datos utilizados en el bloque de instrucción anterior.

**Tabla 4.** Descripción de parámetros de la instrucción MB\_COMM\_LOAD.

| Parámetro | Declaración | Tipo de Dato | Descripción   |
|-----------|-------------|--------------|---|
| PORT      | IN          | Port         | Identificación de hardware  |
| BAUD      | IN          | UDInt        | Velocidad de transmisión en baudios.                                |
| PARITY    | IN          | UInt         | Tipo de paridad:<br>0: No paridad<br>1: Impar<br>2: Par             |
| MB_DB     | IN          | Variant      | Referencia del bloque de datos de instancia usado por el MB_MASTER. |

**MB MASTER:** Se utiliza para definir como dispositivo maestro Modbus al módulo de comunicación. Tiene como funciones:

- Direccionar al dispositivo esclavo a transmitir.
- Establecer el código de función a ejecutar.
- Definir la zona de almacenamiento de los datos enviados o recibidos.

En la figura 11 se observa el bloque de instrucción MB\_MASTER, el cual asigna automáticamente un bloque de datos de instancia DB.

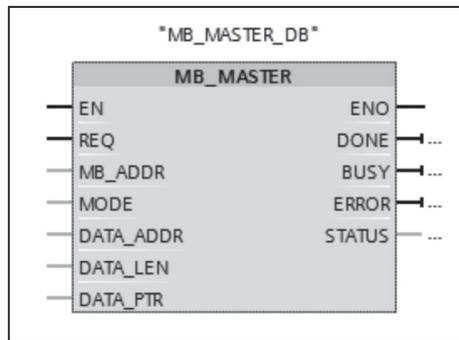


Figura 11. Instrucción MB\_MASTER.

Después de insertar los módulos de comunicación, se asigna un **nombre simbólico** y un **identificador de hardware** a cada puerto del módulo.

Los parámetros configurados para la red Modbus son los siguientes, tal como se puede observar en figura 12:

- *Velocidad de transmisión:* 38400 bps.
- *Paridad:* par.

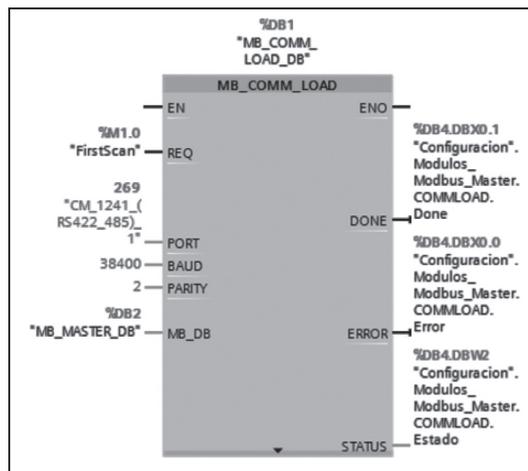


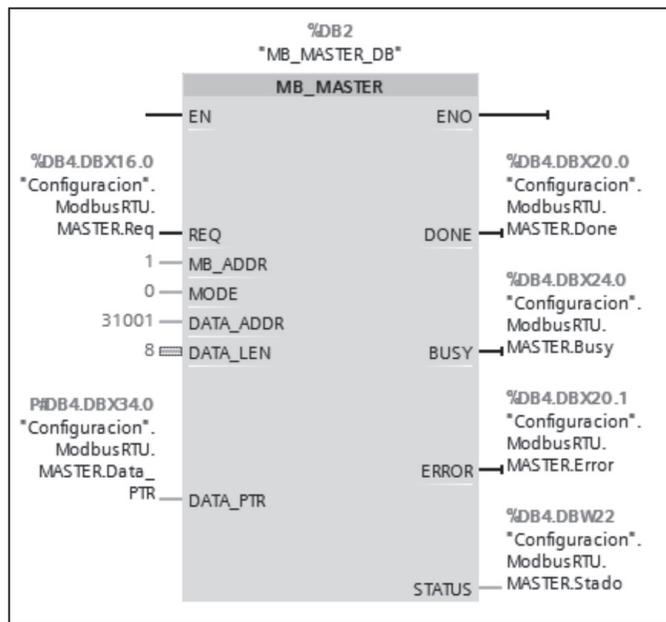
Figura 12. Configuración de parámetros de CM-1241.

*Implementación de funciones de lectura. Función Modbus 04*

La función de lectura permite que el dispositivo “maestro” obtenga las variables que genera el proceso en los actuadores eléctricos.

Para configurar esto en el bloque de instrucciones MB\_MASTER, el parámetro de entrada MODE debe tomar el valor “0”; el parámetro de entrada DATA\_LEN, el valor “15”; y el DATA\_ADDR, el valor “30001”. De esta manera se tiene acceso a los registros desde 30001 hasta 30015.

Se debe agregar un valor decimal “offset”, que dependerá del dispositivo “esclavo” en la red Modbus. En este caso, los actuadores eléctricos AUMA, tienen el valor igual a 1000. De esta manera, queda configurado el bloque MB\_MASTER con los parámetros de entrada y salida, tal como se puede observar en la figura 13.



**Figura 13.** Función de Lectura, Modbus 04.

Los parámetros más sobresalientes se mencionan a continuación:

|                                 |   |             |
|---------------------------------|---|-------------|
| Dirección del esclavo (MB_ADDR) | ► | 1           |
| Tipo de petición (MODE)         | ► | 0 (Lectura) |
| Dirección inicial (DATA_ADDR)   | ► | 31001       |
| Longitud de datos (DATA_LEN)    | ► | 8           |

El dato asignado en DATA\_LEN, como se explicó anteriormente, proporciona la longitud de direcciones que recibirá el dispositivo maestro. Como ejemplo de grá-

fico, si el dato es 8, proporciona que se obtengan los datos de las direcciones desde 31001 hasta 31008.

Los registros leídos se almacenan en una zona específica de un bloque de datos. Para ello se crea un bloque de datos DB general llamado “Configuración” y la zona de almacenamiento interna se establece con el nombre “Data\_PTR”. Resta vincular esta zona de almacenamiento al parámetro DATA\_PTR de la función MB\_MASTER. Como se puede observar en la figura 14.

| Nombre       | Tipo de datos    | Offset | Valor de arranque |
|--------------|------------------|--------|-------------------|
| [ModbusRTU]  | Struct           | 12.0   |                   |
| step_MB_com  | USInt            | 0.0    | 1                 |
| step_MB_leer | USInt            | 1.0    | 1                 |
| MASTER       | Struct           | 2.0    |                   |
| Dir_Esclavo  | UInt             | 0.0    | 2                 |
| Req          | Bool             | 2.0    | false             |
| Longitud     | Int              | 4.0    | 9                 |
| Done         | Bool             | 6.0    | false             |
| Error        | Bool             | 6.1    | false             |
| Stado        | Word             | 8.0    | 16#0              |
| Busy         | Bool             | 10.0   | false             |
| Estado_Error | Word             | 12.0   | 16#0              |
| Modo         | USInt            | 14.0   | 0                 |
| Data_Adress  | UInt             | 16.0   | 0                 |
| Data_PTR     | Array[0..9] o... | 20.0   |                   |
| Data_PTR[0]  | UInt             | 0.0    | 0                 |
| Data_PTR[1]  | UInt             | 2.0    | 0                 |
| Data_PTR[2]  | UInt             | 4.0    | 0                 |
| Data_PTR[3]  | UInt             | 6.0    | 0                 |
| Data_PTR[4]  | UInt             | 8.0    | 0                 |
| Data_PTR[5]  | UInt             | 10.0   | 0                 |
| Data_PTR[6]  | UInt             | 12.0   | 0                 |
| Data_PTR[7]  | UInt             | 14.0   | 0                 |
| Data_PTR[8]  | UInt             | 16.0   | 0                 |
| Data_PTR[9]  | UInt             | 18.0   | 0                 |

Figura 14. Zona local de almacenamiento de los datos leídos con la función Modbus 04.

Este proceso se repite para todos los actuadores de la red Modbus. Simplemente se varía la dirección del esclavo bajo el parámetro “MB\_ADRR”. El control del proceso de lectura para cada actuador se hace mediante el parámetro “DONE”. Cuando este bit se ponga en “TRUE”, entonces los datos están disponibles en el buffer de entrada (DATA\_PTR). Los registros leídos se envían al HMI para su visualización haciendo uso de una interfaz más amigable para operador.

#### Implementación de funciones de escritura. Función Modbus 16

La función de escritura permite que el dispositivo “maestro” envíe, por medio de dos registros tipo Palabras, la operación que desea que ejecute un dispositivo “esclavo”.

Para configurar esto en el bloque de instrucciones MB\_MASTER, el parámetro de entrada MODE debe tomar el valor "1"; el parámetro de entrada DATA\_LEN, el valor "2"; y el DATA\_ADDR, el valor "41001". De esta manera se tiene acceso a los registros desde 41001 hasta 41002. Como se puede ver en la figura 15.

- Dirección del esclavo (MB\_ADDR) ▶ 1
- Tipo de petición (MODE) ▶ 1 (Escritura)
- Dirección inicial (DATA\_ADDR) ▶ 41001
- Longitud de datos (DATA\_LEN) ▶ 2

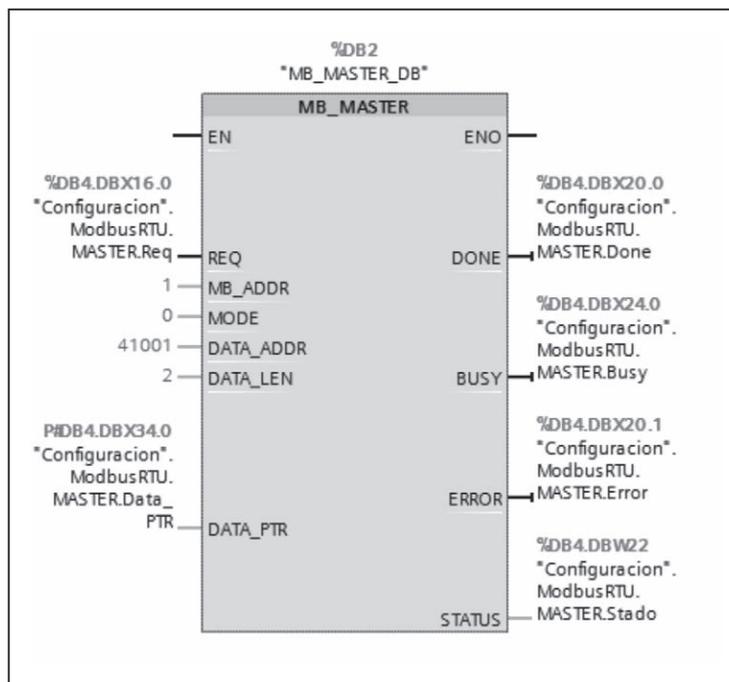


Figura 15. Ajuste de 2 registros de entrada con la función Modbus 16.

De igual forma que el caso anterior, el parámetro DATA\_ADDR se ajusta al valor 41001 debido a que la dirección offset (decimal) del primer registro de salida del actuador tiene la dirección 1000.

En la tabla 5, se muestra todos los posibles valores usados para comandar los actuadores vía registros de salida.

**Tabla 5.** Valores para comandos en actuador

| Valor del Registro de Salida 41001 |             | Respuesta del Actuador |
|------------------------------------|-------------|------------------------|
| Decimal                            | Hexadecimal |                        |
| 256                                | 16#100      | Abre                   |
| 512                                | 16#200      | Cierra                 |
| 1024                               | 16#400      | Alcanza Set Point      |
| 2048                               | 16#800      | Para                   |

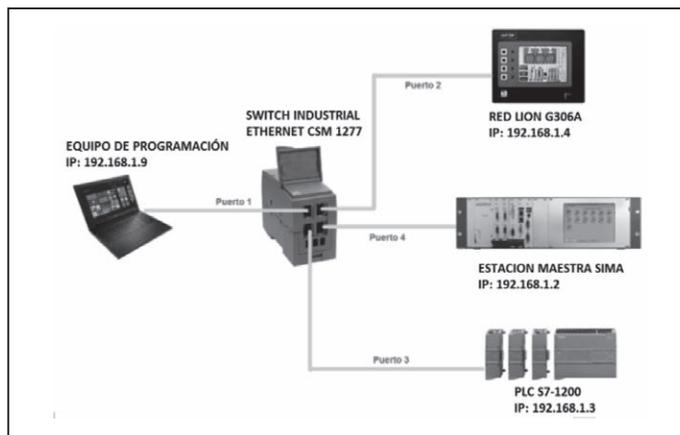
Debido a que el tipo de dato que se declara para estos registros es “UInt” (entero sin signo), los valores que se usan son decimales. El valor del registro 41002 (Set Point), debe ser un valor decimal entre 0 y 1000, donde 0 representa cierre total del actuador, y 1000 es apertura total.

### Comunicación mediante protocolo Modbus TCP/IP

Modbus TCP está basado en la arquitectura cliente/servidor a nivel de capa de aplicación, por lo tanto, para implementar una red de comunicación TCP es importante establecer inicialmente cuáles serán actores que conforman la red, su función y el direccionamiento respectivo.

Como se muestra en la figura 16, la red TCP cuenta con tres equipos aptos para comunicarse bajo el protocolo Modbus TCP/IP, estos son:

- PLC SIMATIC S7-1200
- Pantalla Táctil Red Lion G306A
- Estación Maestra SIMA

**Figura 16.** Arquitectura de Red LAN.

Como se mencionó inicialmente, el PLC SIMATIC S7-1200 será el equipo que realice la gestión de la comunicación entre todos los dispositivos que conforman la red, por lo tanto, se establece que el autómatas adopte la función de cliente Modbus TCP y que los equipos restantes operen como servidores.

Con esto, la red implementada se considera como una red de área local (LAN) y a cada equipo le será asignado una máscara de subred y una dirección IP, tal como se muestra en la tabla 6.

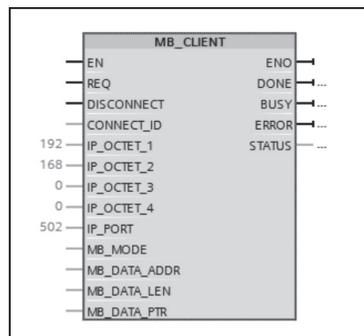
**Tabla 6.** Direcciones IP de los equipos de la red TCP

| Equipo                               | Dirección IP | Máscara de Subred |
|--------------------------------------|--------------|-------------------|
| PLC SIMATIC S7-1200                  | 192.168.1.3  | 255.255.255.0     |
| Pantalla Táctil Red Lion G306A       | 192.168.1.4  |                   |
| Estación Maestra SIMA                | 192.168.1.2  |                   |
| Equipo de programación (PC o laptop) | 192.168.1.9  |                   |

Como se observa en la tabla 3, la red también incluye el equipo de programación para el S7-1200, representada por un PC o laptop.

#### *Configuración S7-1200 como cliente Modbus TCP*

El Simatic S7-1200 se configura como cliente de la red Modbus TCP usando el bloque de instrucción MB\_CLIENT que se usa para entablar conexiones TCP/IP a través del software del TIA PORTAL, tal como se muestra en la figura 17. Se puede observar que un dispositivo Modbus TCP se direcciona de acuerdo a la dirección IP (IP\_OCTECT) del servidor y de un número de puerto (IP\_PORT).



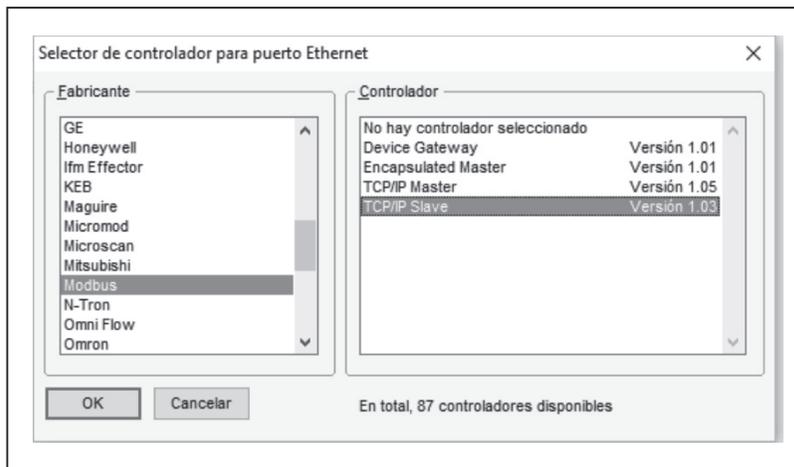
**Figura 17.** Instrucción MB\_CLIENT.

Las funciones principales del MB\_CLIENT son: conectar y desconectar al servidor, enviar peticiones de lectura/escritura, recibir respuestas del mismo.

### *Comunicación S7-1200 con pantalla Red Lion*

Una vez configurado al cliente, se realiza la comunicación entre el S7-1200 y la pantalla Red Lion G306A, que como se dijo anteriormente, será el equipo que permita la visualización de las variables de todo el proceso de control.

Para crear el canal de comunicación con el PLC S7-1200 se configura la pantalla como servidor Modbus TCP desde el software Crimson 3.0. Como se observa en la figura 18, el controlador elegido para el puerto ethernet de la pantalla es “TCP/IP Slave” y el protocolo usado es “Modbus”.



**Figura 18.** Configuración de comunicación maestro/esclavo.

Una vez establecido el servidor, se vincula este con el cliente Modbus TCP por medio de los parámetros de dirección IP y del puerto de comunicación del HMI en la función MB\_CLIENT a través del TIA PORTAL.

### *Comunicación S7-1200 y SIMA*

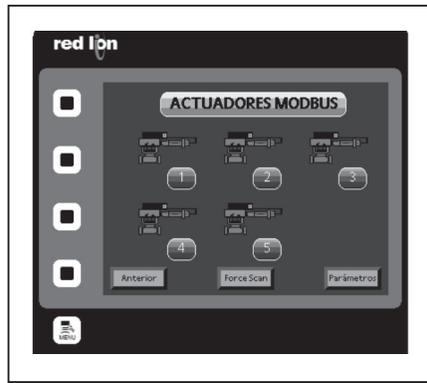
La SIMA tiene varias características para la comunicación con los actuadores, pero esta también puede conectarse con un sistema de control de más alto nivel, es decir que se puede realizar control sobre esta estación y obtener toda la información que esta a su vez obtiene de los actuadores.

Para crear el enlace de la SIMA con el servidor Modbus TCP (S7-1200) se utiliza los recursos del TIA PORTAL siguiendo la configuración de los parámetros de comunicación y direccionamiento establecidos en la instrucción MB\_CLIENT. La dirección IP asignada a la SIMA es 192.168.1.2 y el número de puerto se ajusta a 502.

### Desarrollo del sistema de visualización

Usando el software Crimson 3.0 para configuración de la pantalla Red Lion G306A se desarrolla una serie de pantallas gráficas que permitan al usuario visualizar, monitorizar y controlar los datos provenientes de los registros internos de los actuadores.

En la figura 19 se muestra la ventana de distribución de los actuadores con interfaz Modbus en base a su dirección de esclavo. Debido a que se cuenta con 5 actuadores eléctricos, y por facilidad de direccionamiento, se ha configurado estos valores de 1 a 5, aun cuando los mismos pueden variar de 1 a 247.



*Figura 19.* Distribución de actuadores eléctricos dentro de la red.

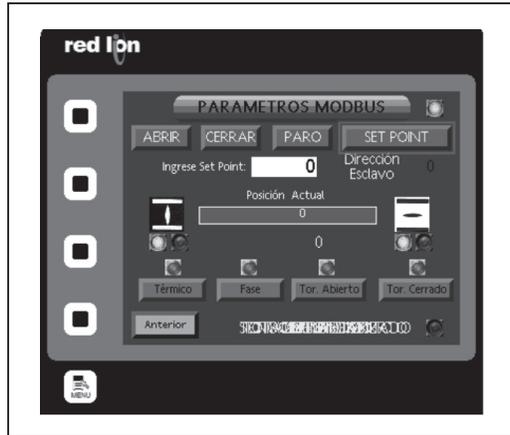
Se observa que el icono representativo de cada actuator es de color rojo, pero cabe indicar que dependiendo del estado en que se encuentre un actuator en cierto instante, este color cambiará de la siguiente manera como se presenta en la tabla 7.

*Tabla 7.* Estado de los actuadores

| Actuador listo para operación remota | Actuador no listo para operación remota |
|--------------------------------------|---|
|                                      |   |

El usuario podrá acceder a los parámetros de cada actuator dando un toque al icono respectivo, ya sea que esté en color verde o rojo, y de forma automática el usuario será dirigido a la ventana de parámetros de control del actuator, tal como se muestra en la figura 20. Como se observa, esta ventana cuenta con cuatro botones de acción para operar a cada actuator (ABRIR, CERRAR, PARO, SETPOINT) y

de igual manera cuenta con indicadores para las señales obtenidas desde los registros de entrada (posición actual y alarmas).



*Figura 20.* Pantalla de operación.

Una vez desarrollado el software de control para los actuadores del autómeta S7-1200 y el software de visualización para la pantalla Red Lion G306A, se procede a ubicar estos equipos en un tablero de control con los elementos adicionales requeridos, tal como se describe en la siguiente sección.

### Implementación del tablero de control

La implementación de tablero de control se realiza en base a la norma IEC 61439-1 e IEC 61439-2, considerando que es un tablero de control de baja potencia. En la figura 21 se muestra la parte frontal del tablero, en la parte superior se observa la pantalla Red Lion G306A y en la parte inferior se observa pulsadores y terminales que permiten conectar a los actuadores al tablero.



*Figura 21.* Parte frontal tablero de control.

En la figura 22, se ilustra la parte interna de tablero con la compartimentación de todos los elementos de protección, control y maniobra utilizados. Se observa el PLC S7-1200 con los módulos de comunicación CM 1241 (Modbus) conectados en la parte izquierda del CPU 1212C, y este a su vez acompañado del switch Ethernet CSM-1277. También es visible la parte posterior de la pantalla Red Lion G306 A.

Todos los equipos mencionados deben ser provistos con la alimentación adecuada según sus características técnicas, es por esto que en la parte superior de tablero se coloca una fuente de voltaje SITOP de 24V@5A para el autómatas, las tarjetas de comunicación y el switch Ethernet; y para la pantalla RedLion G306A se ubica una fuente de voltaje exclusiva modelo S8VS de 24V.



*Figura 22.* Parte interior del tablero de control.

### **Conexión de los actuadores eléctricos al tablero de control**

Los cinco actuadores eléctricos que posee la red son de diferentes modelos y características, y se describen a continuación:

- 2 Actuadores eléctricos AUMATIC AC 01.2 Modbus RTU
- 1 Actuador eléctrico AUMATIC AC 01.1 Modbus RTU
- 2 Actuadores eléctricos SIPOS Flash 5 Modbus RTU

La distribución física de los actuadores en la red se muestra en la figura 23.



*Figura 23.* Actuadores eléctricos.

Antes de realizar la conexión es necesario la puesta en marcha de cada uno de los actuadores, que consiste en ajustar los siguientes parámetros de comunicación de cada actuator para que puedan integrarse a la red industrial:

- Dirección del actuator en la red
- Velocidad de transmisión
- Paridad par y bit de parada



*Figura 24.* Cable profibus [12].

Cabe mencionar que todos los actuadores deberán tener configurados los mismos valores en cuanto a velocidad y paridad. Se escoge que los valores sean de 9600 bps y paridad par.

La topología física usada para conectar los actuadores de la red hacia el tablero de control es la de bus usando el esquema de cableado denominado “en cadena” (*chain*), en donde el conector de salida del primer actuator es conectado al conector de entrada del segundo actuator y así sucesivamente. El cableado se hace con el conductor Profibus (Tipo A) como se observa en la figura 24.

Debido a la estrategia creada al usar dos tarjetas de comunicación, con el fin de no perder la comunicación en caso de ruptura en el cableado de la red, es importan-

te mencionar que la topología final de la red se asemeja a la de un anillo, puesto que el primer actuador se conecta desde la primera tarjeta de comunicación del S7-1200 y el último actuador de la red regresa a la segunda tarjeta de comunicación, formando así un lazo cerrado en las comunicaciones.

## Resultados

Las pruebas en los actuadores y en la comunicación se realizan desde el HMI; y consiste en 3 procedimientos:

1. *Evaluar la implementación del protocolo Modbus RTU y Modbus TCP/IP:* Consiste en enviar comandos correspondientes de apertura, cierre y alcance de set point en cada actuador de la red y al mismo tiempo visualizar la retroalimentación de información correspondiente al proceso (porcentaje de apertura o cierre en cada actuador).
2. *Simular fallas en el actuador:* Consiste en simular fallas de torque y temperatura en cada actuador y visualizar si hay retroalimentación de información en cuanto a estado del actuador.
3. *Ruptura en el cableado de la red:* Consiste en realizar ruptura del cableado entre dos actuadores continuos (dirección 2 y 3) y de la misma forma, visualizar el comportamiento del sistema en el HMI ante el evento.
4. *Eliminar suministro eléctrico en varios actuadores de la red:* Consiste en quitar la alimentación eléctrica a dos actuadores y observar el comportamiento de la red y el sistema de comunicación.

### Resultados de comunicación de protocolo Modbus RTU y Modbus TCP/IP

Los resultados arrojados a los comandos de operación enviados mediante los protocolos Modbus RTU y Profibus DP se realizaron de forma exitosa.

La información de los registros de entrada que se visualizan en el HMI refleja el correcto comportamiento de los actuadores ante los comandos de operación ABRIR Y CERRAR. Para el caso del valor real leído, cuando se ordenó alcanzar un SET POINT, este valor difiere un poco del valor seteado debido a las características mecánicas del motor del actuador, tal como se muestra en la figura 25.



**Figura 25.** Pruebas para alcanzar set point.

### Resultados ante simulación de fallas

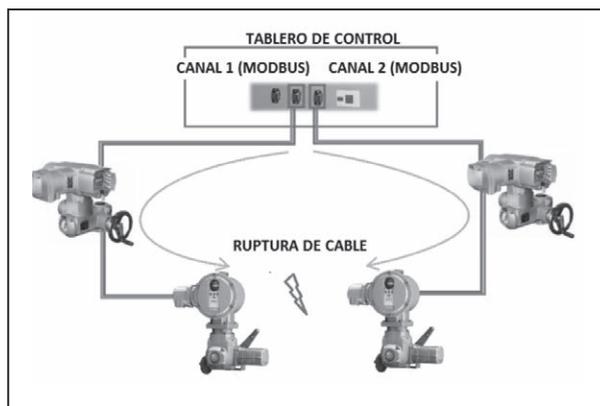
Si por algún motivo en el transcurso de la apertura del actuador se genera una interrupción en el eje de rotación, el HMI lanza un aviso de “Falla de torque abierto”, tal como se puede ver en la figura 26, lo que significa que algún elemento externo al actuador pudo haberse atascado en el eje de rotación, o que el actuador no puede brindar el torque necesario para abrir a una válvula debido a sus dimensiones.



**Figura 26.** Simulación de falla de torque abierto.

### Resultados ante ruptura física del cable de bus en la red Modbus

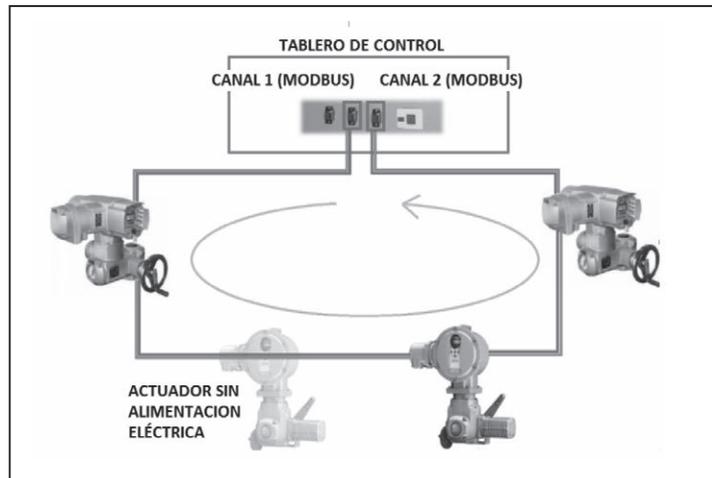
La ruptura del cable entre dos actuadores de la red Modbus hizo que la línea de bus se segmente en dos líneas independientes, tal como se ilustra en la figura 27; esto debido a los mecanismos desarrollados en el S7-1200 y a que los actuadores están conectados en topología de lazo cerrado.



**Figura 27.** Comunicación del sistema de control en caso de ruptura en el cable de bus.

*Resultados en caso de que el actuador pierda suministro eléctrico*

Cuando se pierde el suministro eléctrico, los conductores provenientes de los actuadores vecinos se enlazan automáticamente, logrando mantener la estructura de lazo cerrado en la red. Esto se observa en la figura 28.



**Figura 28.** Interconectado del bus en caso de un actuador sin suministro eléctrico.

## Conclusiones

El hecho de haber integrado la SIMA, la pantalla táctil Red Lion y el PLC S7-1200 sobre una red Modbus TCP, y del mismo modo, el haber integrado los actuadores AUMA y actuadores SIPOS sobre una red Modbus RTU, demostró la interoperabilidad que presenta el protocolo Modbus, ya que pudo integrar a dispositivos de diferentes fabricantes sobre una misma red de datos.

Con la topología en lazo cerrado implementado en la red Modbus, se garantiza la transmisión y recepción de datos entre el sistema de control y los actuadores eléctricos a pesar de tener pérdida en la comunicación por causas como ruptura de conexión entre dispositivos esclavos o desconexión de energía eléctrica de un esclavo

## Referencias

- [1] Modbus Organization. (2006, Diciembre). *Modbus application protocol specification*. [Online]. Available: [http://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf).
- [2] J. Bartolomé. (2011, Enero). *El protocolo Modbus*. [Online]. Available: <http://www.tolaemon.com/docs/modbus.htm>.

- [3] Trexon Inc. (2000, Enero). *Modbus protocol*. [Online]. Available: [http://irtfweb.ifa.hawaii.edu/~smokey/software/about/sixnet/modbus/modbus\\_protocol.pdf](http://irtfweb.ifa.hawaii.edu/~smokey/software/about/sixnet/modbus/modbus_protocol.pdf).
- [4] J. B. Piña. (2013, Junio). *Modbus: Representación de datos*. [Online]. Available: <http://uhu.es/antonio.barragan/content/representacion-datos>.
- [5] Autómatas industriales. (2006). *Profibus*. [Online]. Available: <http://www.automatas.org/redes/profibus.htm>.
- [6] *Manual de la comunicación Profibus DP*, WEG S.A, Jaragua del Sur, SC, 2012, pp. 8.
- [7] J. B. Piña. (2012, Mayo). *Topologías*. [Online]. Available: <http://uhu.es/antonio.barragan/content/5topologias>.
- [8] *Actuadores eléctricos para la automatización de válvulas industriales*, AUMA, Muellheim, 2015.
- [9] Siemens. (2011, Noviembre). *SIMATIC S7-1200 Getting Started*. [Online]. Available: <https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200-GETTINGSTARTER.PDF>.
- [10] Siemens. *Catálogo de productos Siemens*, Múnich, BY, 2015.
- [11] *Actuator controls AUMATIC AC 01.1/ACExC 01.1 Modbus*, AUMA, Muellheim, 2015.
- [12] A. Ruiz. (2002). *Implementación de una red Modbus/TCP*, Universidad del Valle. [Online]. Available: <http://www.bairesrobotics.com.ar/data/protocolo-modbus.pdf>.