

# Diseño e implementación de un algoritmo para translación de protocolo entre las normas IEC60870-5-104 y MODBUS TCP/IP

## Design and Implementation of an Algorithm for Protocol Translation Between the IEC60870-5-104 and MODBUS TCP/IP Standards

Recibido: 10 de marzo de 2015 • Aceptado: 26 de marzo de 2015

Para citar este artículo: J. Feria, J. Camargo y E. Muskus «Diseño e implementación de un algoritmo para translación de protocolo entre las normas IEC60870-5-104 y MODBUS TCP/IP», *Ingenium*, vol. 16, n.º 32, pp. 118-125, junio, 2015.



John Henry Feria Briceño\*  
Julián Rolando Camargo López\*\*  
Elkin Gabriel Muskus Rincón\*\*\*

## Resumen

Este artículo muestra los resultados del proceso de diseño, implementación y prueba de un algoritmo para hacer la translación de protocolo entre las normas IEC 60870-5-104 y MODBUS TCP/IP, éste modelo de algoritmos son de amplio uso en los procesos de supervisión y automatización en sub-estaciones eléctricas.

\* Ingeniero Electrónico. Universidad Distrital Francisco José de Caldas, Bogotá, Colombia E-mail: [jhferiab@correo.udistrital.edu.co](mailto:jhferiab@correo.udistrital.edu.co)

\*\* Especialista en diseño y construcción de soluciones telemáticas, Ingeniero Electrónico, Profesor Asistente, Grupo de investigación: LASER. Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. E-mail: [jcamargo@udistrital.edu.co](mailto:jcamargo@udistrital.edu.co)

\*\*\* M.Sc. en Ingeniería Mecatrónica, Especialista en Telecomunicaciones Móviles, Ingeniero Electrónico, Profesor Asistente. Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. E-mail: [egmuskur@udistrital.edu.co](mailto:egmuskur@udistrital.edu.co)

## Palabras clave

Norma IEC, MODBUS, Servopack, SCADA.

## Abstract

This paper is oriented to show the results obtained in the process of design, implementation and testing of an algorithm to translate a protocol between the standards IEC 60870-5-104 and MODBUS TCP/IP, this algorithms are commonly used in monitoring processes and automation in electric sub-stations.

## Keywords

IEC Standard, MODBUS, Servopack, SCADA.

## I. Introducción

La conversión de protocolo es una herramienta utilizada para lograr la integración de múltiples equipos en una red de comunicaciones, que permite hacer tareas como supervisión de variables, gestión remota y maniobras de control, ya sea desde un centro de control o desde un determinado número de locaciones remotas, de acuerdo a los requerimientos específicos del sistema y al grado de complejidad proyectado en las etapas de diseño, como se puede ver en [1], donde se hace una exposición bastante detallada de las consideraciones de ingeniería en cuanto a sistemas de control.

Existen diversidad de protocolos implementados en los equipos de protección y control utilizados en las subestaciones eléctricas, algunos corresponden a estándares propios desarrollados por las empresas fabricantes de éste tipo de equipamientos como es el caso de Alstom y su serie de relés MiCOM [2] que cuentan con los protocolos *Courier/K-Bus*, exclusivo de ésta serie de equipos; otros fabricantes sin embargo optan por hacer uso de estándares definidos por grupos de usuarios como es el caso de DNP V3.0 (*Distributed Network Protocol Version 3*) [3], con el fin de que sus productos puedan ser integrados fácilmente en redes de comunicaciones en las que eventualmente podrían estar presentes sistemas de protección y control de otros fabricantes.

En éste trabajo se abordarán dos normas ampliamente difundidas en el sector eléctrico, MODBUS TCP/IP desarrollada por la organización MODBUS [4] e IEC 60870-5-104 desarrollado por la Comisión Electrotécnica Internacional [5], cuyo uso en subestaciones eléctricas es bastante difundido; la arquitectura propuesta para el desarrollo del algoritmo es del tipo maestro - esclavo, como se ilustra en la Fig. 1, el medio físico utilizado en el enlace maestro-gateway y gateway-esclavo será Ethernet 10/100 con cable UTP de cobre, tal como está definido en [6] y [7].



Fig. 1 Arquitectura de control y supervisión

Aunque MODBUS es un protocolo implementado por la mayoría de fabricantes en sus equipos, cuenta con una serie de limitaciones en cuanto al tipo de funciones soportadas que impiden su uso en operaciones de supervisión y control de instalaciones de gestión de energía eléctrica, un claro ejemplo es la incapacidad de reportar de forma espontánea (sin que sea interrogado previamente) el cambio en el estado de las variables de operación, así en caso de que el equipo de protección abra un interruptor por condiciones de sobre-corriente [8] el sistema no estaría notificado de dicho evento y no podría iniciar las operaciones de reposición del sistema. Ésta carencia (así como otras tales como estampa de tiempo en los eventos) es solucionada en la definición de estándares propiamente diseñados para atender las necesidades del sector eléctrico.

La norma IEC 60870-5-104 fue diseñada de acuerdo a los estándares, tipos de datos, procedimientos de transferencia de información y gestión de tramas que habían sido previamente definidos en IEC 60870-101 [9], IEC 60870-5-102 [10] e IEC 60870-5-103 [11]. Por lo que cuenta con el conjunto de funciones y procedimientos que le permiten hacer el reporte de eventos de protección, reposición y anomalías en valores de las variables registradas por los sistemas de protección y control del sistema.

Una de las funciones que cumplen los dispositivos de translación de protocolo, es la selección y concentración de las variables de interés de acuerdo a las particularidades del sistema, esto debido a que los equipos de protección suelen contener bloques de información que no siempre son de interés, un ejemplo de estos campos de información se pueden consultar en [12], si se tomaran todas las variables puestas a disposición por el fabricante es probable que se agotarían las existencias de variables contenidas en la licencias de los sistemas SCADA [13] con la integración de unos pocos equipos.

Otra de las ventajas de la implementación de éste tipo de sistema de translación de protocolo es que suele simplificar el tráfico de datos en las redes de área extensa. Como ya se había indicado previamente la norma MODBUS, no es capaz de reportar ningún tipo de evento o información a no ser que le sea previamente consultada por el maestro, en el entorno de la subestación o instalación supervisada, es posible tener un elevado tráfico de datos puesto que se cuenta con una red LAN [14] en la que es posible man-

tener tasas de datos altas con un elevado grado de confiabilidad y servicio, tal como se muestra en la Fig. 2, por lo que es posible hacer secuencias continuas de interrogación para la supervisión de variables, obteniendo respuestas en intervalos de tiempo más bien cortos; sin embargo los centros de control y monitoreo [15] suelen estar a grandes distancias en los centros urbanos más importantes, por lo que el acceso a la información consolidada en las RTU's es realizada por los medios técnicamente viables, que para el caso de locaciones remotas suelen ser enlaces vía red celular o internet satelital, en los que hay tarificación por uso de datos.



Fig. 2 Red LAN [14]

Al confinar las operaciones de interrogación cíclicas a la red LAN para, a partir de ésta información, consolidar reportes, cambios de estados y valores de alarma que serán reportados al centro de control, se minimiza el tráfico de datos no relevantes, se optimiza el uso de ancho de banda [16] y se hace más eficiente el proceso de supervisión de las diferentes instalaciones.

## II. Condiciones del diseño

Para el desarrollo de la propuesta de diseño se plantean unas condiciones iniciales, tomadas de los requerimientos de integración que es común encontrar en las sub-estaciones eléctricas en las que hay equipos de protección y control (relés) nuevos y antiguos, así como equipos de medida (medidores o contadores de consumo de energía) de diferentes fabricantes, las condiciones de partida son:

- 1) La red de equipos esclavos estará compuesta por estaciones remotas con protocolo MODBUS TCP/IP; inicialmente para efecto de pruebas, se delimitara la cantidad de estaciones a dos. Las funciones MODBUS implementadas en los equipos de prueba serán lectura de input registers y coils y escritura de coils.
- 2) La red de equipos maestros estará formada por una única estación con protocolo IEC 60870-5-104, la cual tendrá implementadas las funciones de interrogación general, comando simple, comando doble y sincronización de tiempo.

- 3) Tanto los equipos maestros como los esclavos estarán en un mismo segmento de red y el enrutamiento de paquetes Ethernet se hará mediante un router sencillo sin que intervengan políticas avanzadas de gestión, enrutamiento, gestión de calidad de servicio, etc.
- 4) El algoritmo estará en capacidad de gestionar una base de datos 'local' con el fin de identificar los cambios de estado o valor en las variables configuradas y construir así el reporte pertinente.
- 5) El desarrollo se hará haciendo uso de aplicaciones de software libre, mientras sea posible.

### III. Arquitectura propuesta

Para la implementación de una arquitectura de software que resuelva los requerimientos dispuestos en la sección II y que adicionalmente presente características de ingeniería de software (que agilicen y faciliten el proceso de codificación y eventual mantenimiento y crecimiento), se propone una arquitectura basada en la implementación de módulos de software que sean funcionalmente independientes y que interactúen entre sí mediante un integrador de operaciones.

Éstos módulos propuestos, estarán divididos internamente en sub-módulos de gestión o supervisores (los cuales están jerárquicamente en el orden más alto de cada módulo) quien a su vez coordinará las operaciones internas hechas por cada uno de los sub-módulos restantes, esto se puede apreciar en la Fig. 3.

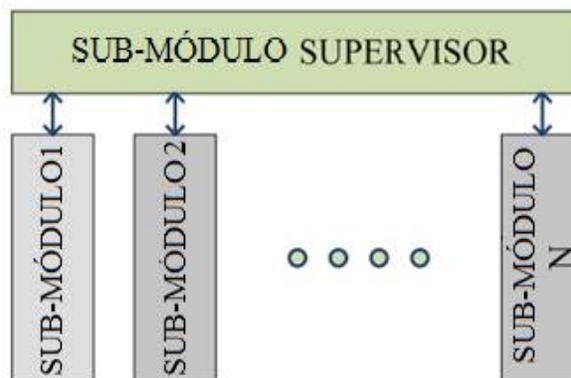


Fig. 3 Arquitectura de cada módulo de software

Una de las ventajas que presenta éste tipo de arquitectura es que se puede hacer la integración de nuevas funcionalidades, mediante la implementación de sub-módulos cuya interacción será únicamente con las funciones internas del supervisor, dando como resultado que la forma como los diferentes módulos de software interactúan entre sí, no se ve afectada.

La interacción entre módulos será entonces coordinada mediante la existencia de un integrador que dirija las peticiones y transferencias de bloques de información entre

los diferentes componentes del algoritmo, el modo de operación que eventualmente se tendría una vez desarrollado el algoritmo, es el que se muestra en la Fig. 4.

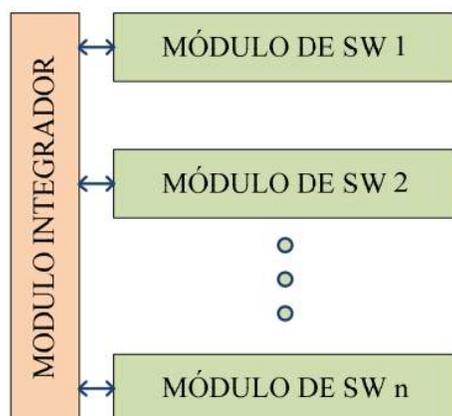


Fig. 4 Interacción entre módulos de software

Gracias al planteamiento de ésta arquitectura, se tiene ya definido uno de los módulos de software necesarios para la implementación del algoritmo, en adelante se hará referencia a éste módulo como GTW o módulo gateway.

#### IV. Módulos de software

En ésta sección se hace la delimitación de los módulos de software empleados para el desarrollo total del algoritmo así como del lenguaje de programación a utilizar para la codificación del mismo.

En el estado actual del mercado de sistemas embebidos y dispositivos de comunicaciones es posible encontrar dispositivos de hardware que cuentan con las certificaciones debidas para operar en entornos hostiles con características ambientales como presencia de fauna, temperaturas y humedades relativas elevadas, un ejemplo de éste tipo de dispositivo es el uC-7112 de MOXA [17].

Las unidades de éste tipo suelen tener implementaciones de micro Linux que es un sistema operativo diseñado para ser ejecutado de forma estable en sistemas embebidos con bajos recursos y diseñados con las características particulares de hardware, debido a que los sistemas Unix están programados sobre C, éste lenguaje se puede considerar como nativo de esta clase de aparatos, lo que lo hace un candidato ideal para desarrollar el proceso de codificación; adicionalmente se hará uso de subversión como gestor de versiones para llevar una secuencia cronológica del desarrollo y los cambios implementados.

El sistema operativo de trabajo será Debían versión 7.2.0 en versión de 32 bits, ya que es un sistema operativo robusto y estable, que tiene soporte para subversión. A partir de esto es necesario entonces definir la cantidad de módulos de software necesarios para lograr cumplir con los requerimientos establecidos en la sección II; los protocolos de co-

municaciones (como los utilizados en sub-estaciones eléctricas) objeto de éste proyecto, usualmente suelen definir además de los procedimientos para establecimiento del enlace, las transacciones de intercambio de datos y la supervisión en la coherencia de las tramas, el medio físico utilizado. Para éste caso se tiene que dichos medios físicos definidos tanto por la norma IEC 60870-5-104 como por MODBUS son los correspondientes al estándar RFC 791, 793 y 894, que definen el acceso al medio, formato de las tramas de datos y protocolos de transmisión, y que suelen ser llamados comúnmente como Ethernet. Como el medio físico es el mismo para ambas normas, se evidencia la necesidad de crear un módulo de software que haga la gestión de los paquetes de red, el cual será llamado ETH en referencia a al protocolo Ethernet.

Para el desarrollo de las funciones propias de cada norma es necesario diseñar un módulo de software independiente para cada una, a partir de esto se evidencia la necesidad de agregar dos módulos que en adelante serán referidos como IEC y MBS, cada uno ejecutando las tareas de gestión de su respectivo protocolo (IEC 60870-5-104 y MODBUS TCP/IP respectivamente).

La gestión de variables, espacios de memoria y enrutamiento de direcciones es otra tarea bien definida a ejecutar, puesto que lo que se desea es tener independencia funcional entre los diferentes módulos, las librerías encargadas de la gestión de los protocolos solamente tendrán en cuenta la información pertinente a secuencias de comunicación, tiempos de expiración de las transacciones y coherencia en los campos de datos, por ésta razón, el contenido mismo de las variables, así como la detección de cambios, no será responsabilidad de los gestores de protocolos, en cambio dicha responsabilidad será delegada en un nuevo módulo llamado RTM haciendo referencia a las operaciones de enrutamiento y gestión de memoria.

Con estos elementos se puede entonces establecer un modelo de operación como el que se ilustra en la Fig. 5, que a diferencia de la Fig. 4 delimita la cantidad y el tipo de módulos a codificar. Las declaraciones y divisiones propias de cada uno de los módulos de software serán discutidas en detalle en la sección V.

## V. Codificación

A continuación se hace una descripción del proceso de codificación efectuado en cada uno de los módulos referidos en la Fig. 5, con las particularidades específicas de cada uno, así como los sub-módulos diseñados para que cumplan con las funciones propuestas.

### A. Módulo ETH

El módulo ETH es el encargado de establecer las conexiones tipo cliente/servidor [18] tanto con las unidades esclavas, como con el maestro. Con miras a una futura expansión de las capacidades del algoritmo éste módulo se diseñó para gestionar de forma simultánea 10 conexiones tipo cliente y 10 conexiones tipo servidor. En su ejecución de forma autónoma hace un barrido cíclico de los diferentes sockets creados. En el proceso de codificación se creó un sub-módulo *skt* en donde se concentran las actividades de supervisión, éste

sub-módulo le reporta los cambios en el estado de las diferentes conexiones al supervisor Ethernet, quien interactúa directamente con el supervisor de GTW.

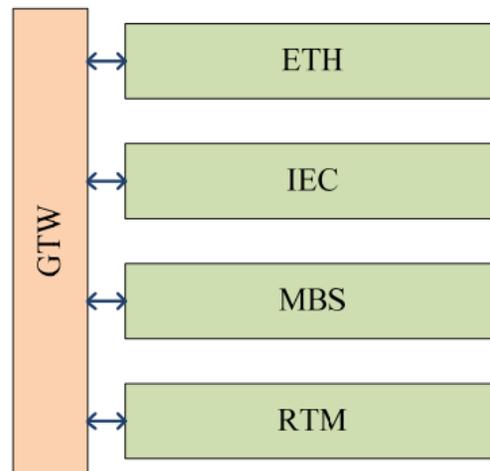


Fig. 5 Arquitectura final de software

## B. Módulo IEC

El módulo IEC se codificó teniendo en cuenta que es posible hacer la adición de nuevas librerías con cualquiera de las normas no contempladas en éste proyecto, pero definidas en los estándares desde IEC 60870-5-101/102/103 hasta 61850. Teniendo esto presente, además del obligado módulo supervisor local que interactúa con el supervisor GTW, se crearon los sub-módulos *esv* y *ems*. *Esv* contiene las librerías y funciones necesarias para el establecimiento de un enlace IEC 60870-5-104 de tipo esclavo, mientras que *ems* define las librerías y funciones necesarias para la comunicación hacia un esclavo IEC 60870-5-104; si bien se adicionó el objeto *ems.c/h*, éste está vacío puesto que está fuera del alcance de este documento la implementación de un maestro 104 aunque se desea dejar planteada la organización estructural para crecimientos posteriores.

Las funciones implementadas en la librería *esv* corresponden a la atención de las siguientes tramas en modo monitor: interrogación General -TI100-, Sincronización de reloj TI103, Reporte espontáneo de puntos simples -TI001-, reporte espontaneo de puntos simples con estampa de tiempo -TI030-, reporte espontaneo de puntos dobles -TI003-, reporte espontaneo de puntos dobles con estampa de tiempo -TI031-, reporte espontaneo de medidas de 16 bits normalizadas -TI009- y reporte espontaneo de medidas de 16 bits normalizadas con estampa de tiempo -TI034-. Para las tramas en modo monitor se implementó la atención a las siguientes tramas: comando simple -TI045- y comando doble -TI046-.

## C. Módulo MBS

El módulo MBS se encarga de la gestión de las funciones MODBUS necesarias para atender el proceso de interrogación hacia cada uno de los esclavos definidos. Las funciones implementadas en la librería MODBUS son F01 para lectura de coils, F03 para la lectura de input registers y F05/F15 para la escritura de coils necesarias para la ejecución de los

comandos simples y dobles desde el protocolo maestro, al igual que en la implementación del módulo IEC se crearon los objetos vacíos ems, sms, y ssv, los cuales corresponden a implementaciones para MODBUS TCP/IP maestro, MODBUS serial maestro y MODBUS serial esclavo.

En la implementación de MODBUS TCP/IP se tuvo en cuenta que es común encontrar dispositivos concentradores de comunicaciones en los que mediante una única dirección IP, es posible acceder a una red de esclavos seriales, teniendo esto en mente, una vez que se recibe un paquete MODBUS, se inicia un algoritmo de búsqueda en el que se valida el ID del esclavo, contrastando esta información externa con un arreglo interno en el que se define tanto el valor de habilitación del usuario (esclavo) como su ID, aunque la norma indica que en las implementaciones del tipo TCP/IP éste campo puede ser omitido o puesto a '0'.

#### D. Módulo RTM

El módulo RTM es sin duda de los módulos más complejos de implementar por las múltiples funciones que ejecuta, una de las actividades (la primera en implementarse) es el enrutamiento de direcciones. La función de enrutamiento de direcciones tiene como finalidad establecer una relación unívoca entre las variables definidas en el maestro IEC 60870-5-104 y las variables definidas en cada uno de los esclavos de la red, para optimizar ésta relación, el uso de recursos en la construcción de las tramas MODBUS y simplificar la codificación de prueba, el algoritmo fue diseñado para trabajar con bloques secuenciales de datos, es decir, sin espacios en blanco con lo que se puede establecer una equivalencia como la que se muestra en la Fig. 6.

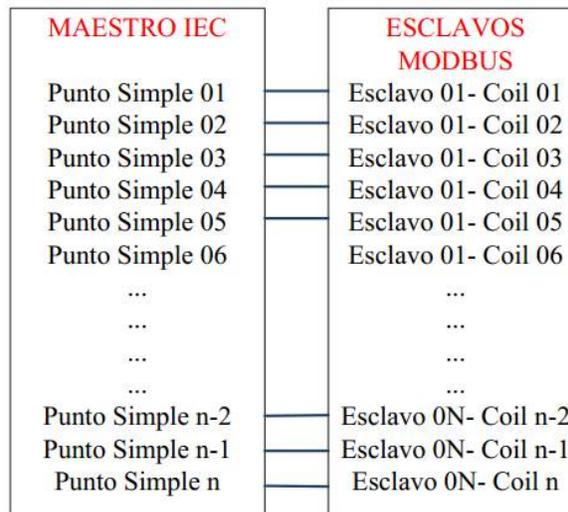


Fig. 6 Relación entre variables

El sub-módulo encargado de ejecutar el enrutamiento de la dirección maestro-esclavo y viceversa se denominó *adr*, las interacciones que establece este módulo con el algoritmo se realizan a través del supervisor RTM, el procedimiento mediante el cual los protocolos acceden a la asignación de direcciones es el siguiente:

- 1) El protocolo genera una petición al módulo supervisor para obtener las direcciones asociadas a un campo de datos específico.
- 2) El módulo de integración envía la petición hacia el módulo RTM quien entrega el bloque de direcciones.
- 3) Nuevamente el módulo integrador entrega el bloque de información al protocolo que realizó la solicitud quien se encargara de hacer las peticiones o reportes de acuerdo a sus funciones internas.

Con respecto a la gestión de los espacios de memoria, se definió que es necesario hacer la codificación de un módulo adicional para hacer la supervisión de los posibles cambios que se presenten en los registros de memoria asociados a las variables declaradas tanto en el maestro como en el esclavo; con el fin de optimizar el uso de espacio dedicado al alojamiento de los diferentes tipos de campos de datos, se creó un único espacio en el que se comparte el acceso por parte de los dos protocolos.

El procedimiento para identificar cambio en los estados de las variables es como se indica a continuación:

- 1) Se recibe un bloque de datos proveniente de la respuesta enviada desde la unidad remota a una petición de lectura de campos de información, que bien pueden ser variables de tipo binarias o registros.
- 2) La cadena recibida es identificada y etiquetada de forma que se pueda encontrar su par en la base de datos.
- 3) Se procede a comparar los campos de memoria como un bloque homogéneo, si se encuentran diferencias, entonces se procede a correr un algoritmo que hace la segmentación en registros separados y se inicia el motor de comparación para identificar los registros que reportan cambios.
- 4) Se generan eventos con los campos de información debidamente identificados para que el módulo de integración los direcciona al protocolo pertinente.

Mediante la combinación de estos dos sub-módulos cuyas operaciones de intercambio están gestionadas gracias al supervisor local, es posible hacer la integración de las operaciones entre protocolos, tales como reporte espontáneo de cambio en el contenido de los campos de información así como para la ejecución de transacciones en modo control.

## **E. Módulo GTW**

GTW es el encargado de hacer la gestión de direccionar los paquetes con peticiones, campos de datos y reporte de eventos entre los demás módulos que componen el algoritmo, visto desde esta perspectiva, el algoritmo que se ejecuta este bloque funcional bien podría efectuar el enrutamiento entre cualquier tipo de campo de información que se requiera, pues en sí él no tiene conocimiento no del tipo de datos ni del tipo de transacciones contenidas en los diferentes reportes que gestiona, lo cual favorece la independencia funcional del algoritmo y mejora tanto la cohesión como el acoplamiento [19].

Para la codificación del módulo GTW, fue necesario crear las rutinas que se encargan de hacer el llamado periódico de los supervisores de cada uno de los módulos componentes; sin embargo y a modo de mejora es posible en una futura implementación del algoritmo aplicar el concepto (y las herramientas) de tuberías que permitan ejecutar de forma simultánea cada uno de los bloques de software componentes y delegar al sistema operativo la asignación de slots de tiempo (ésta implementación excede los objetivos inicialmente propuestos, pero sería un ejercicio académico bastante interesante).

## VI. Pruebas

El proceso de prueba para el desarrollo del algoritmo fue continuo y aplicado en cada una de las etapas de codificación, mediante la creación de funciones de prueba que simulaban la interacción de los sub-módulos creados con el resto del algoritmo. En el caso de los protocolos de comunicaciones se usaron simuladores para validar la coherencia en los campos de información generados, verificar el mantenimiento de las secuencias, explorar las posibles velocidades de interrogación, etc. Gracias a la rigurosidad de las verificaciones realizadas en cada una de las etapas de codificación y a los resultados positivos obtenidos en las pruebas aisladas, se propuso iniciar una prueba experimental con equipos que cumplieran con los estándares de comunicaciones.

En el caso de la verificación funcional de MODBUS se hizo uso de un servo-pack ML2600 de Yaskawa, el cual cuenta con dos puertos Ethernet y el stack MODBUS implementado, lo que resulta muy beneficioso a la hora de reducir los tiempos de ejecución de pruebas.

Para hacer la verificación del protocolo maestro, pese a que la idea inicial era contar con una aplicación tipo SCADA en la que se pudiera hacer la visualización de variables analógicas y estados binarios, se encontró que si bien es cierto algunos SCADAS permiten la operación en modo "demo" definitivamente ninguno permite hacer uso de la norma IEC 60870-5-104, razón por la cual el proceso de prueba se realizó con un simulador como maestro.

La filosofía de la prueba práctica implementada fue crear una red con dos servomotores Yaskawa previamente programados de forma que con el envío de un comando en una de las bobinas se puede girar el eje del servomotor a una posición predefinida que será considerada como un estado de "abierto"; al cambiar de estado la bobina referida anteriormente el servomotor restablece la posición original del eje a un estado de "cerrado", gracias a estos dos comandos se puede considerar el servomotor como un interruptor motorizado, el cual reporta su estado mediante el cambio en el valor binario de una variable tipo bobina.

Adicionalmente con el fin de poder hacer la verificación de puntos dobles y medidas de 16 bits se hizo la asociación de dos interruptores dispuestos en el panel frontal del sistema de control de movimiento para replicar el estado de un punto doble y se asoció el estado de posición del eje a una variable de 16 bits, a continuación se hará una introducción al kit con el que se llevó a cabo el trabajo de prueba y finalmente se registrarán los resultados de las pruebas.

## A. Servomotor Yaskawa

Los kits de control de movimiento Yaskawa (Fig. 7) están compuestos por un “Servopack” quien se encarga de hacer el control sobre los estados y parámetros funcionales de la operación, el servomotor propiamente dicho y una pantalla HMI táctil en la que se puede interactuar con los ajustes del servomotor. La información acerca del Servopack puede ser encontrada en [20]. La razón por la que se escogió éste equipo es porque dispone de comunicación Ethernet y cuenta con el protocolo MODBUS TCP/IP implementado. Los kits en cuestión vienen con un programa pre instalado en el que se puede interactuar con los parámetros básicos para definir el movimiento del servomotor, estos parámetros son: aceleración ideal, desaceleración ideal, posición final y velocidad. En la Fig. 8 se puede apreciar la presentación de estos valores en el HMI.



Fig. 7 Kit Yaskawa

No hace parte de los alcances de éste proyecto definir el modo de operación del servomotor, ni tampoco los elementos de programación involucrados en la operación del mismo; a partir del programa que estaba implementado, se agregó un módulo de código encargado de crear un grupo de variables disponibles desde el protocolo MODBUS TCP/IP, las cuales interactúan con los parámetros operativos del servomotor.



Fig. 8 Vista del HMI con los parámetros ajustables

## B. Desarrollo del algoritmo

El programa implementado en el Servopack (Fig. 9) se desarrolló teniendo en cuenta que se buscaba lograr una integración que no alterase el funcionamiento del código existente

y que permitiera crear un banco de registros accesibles desde MODBUS para así poder modificar de forma remota las condiciones de operación del servo y lograr la secuencia de movimiento deseada para la verificación funcional, el desarrollo se hizo con la aplicación “MWiec Pro 2.50.0.78” [21] obtenida directamente del sitio web de Yaskawa.

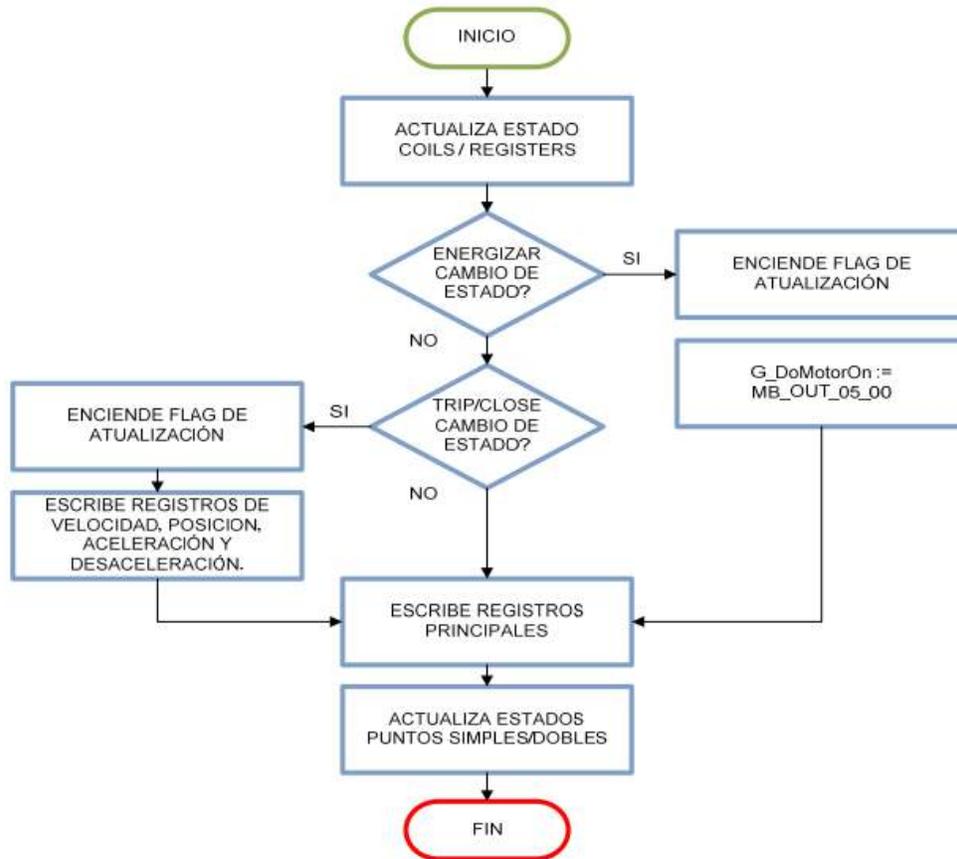


Fig. 9 Diagrama de flujo Servopack Yaskawa

Lo que se hizo para lograr la coexistencia de las dos aplicaciones fue crear un nuevo módulo llamado MBS, escrito en lenguaje estructurado en el cual se implementó la lógica de operación descrita en la en las Fig. 10, 11 y 12, se tiene el listado de las variables creadas y registradas en el banco de memoria para variables MODBUS, a partir de éstas variables se puede lograr la interacción deseada desde el maestro IEC 60870-104 pasando por el algoritmo de translación.

ENTRADAS - PUNTOS SIMPLES					
ID	F	ADDR	NAME	ASSIG.	DESC.
1	02	0	MB_IN_02_00	PULSO ENERGIZAR 7s	Contacto para Punto Simple 1
1	02	1	MB_IN_02_01	PULSO TRIP/CLOSE 7s	Contacto para Punto Simple 2
1	02	2	MB_IN_02_02	PULSO N.ASIGNADO 7s	Contacto para Punto Simple 3
1	02	3	MB_IN_02_03	PLC I.4	Contacto para Punto Simple 4
1	02	4	MB_IN_02_04	PLC I.5	Contacto para Punto Simple 5
1	02	5	MB_IN_02_05	DISPONIBLE	Contacto para Punto Simple 6

Fig. 10 Servopack - Puntos Simples

Los tres primeros puntos simples indicados en la Fig. 10 corresponden a pulsos con una duración de 7 segundos que se generan a partir del cambio de estado en los tres primeros comandos simples listados en la Fig. 12.

ENTRADAS - PUNTOS DOBLES						
ID	F	ADDR	NAME	ASSIG.	DESC.	
1	02	16	MB_IN_02_10	PLC I.7 PLCI,8	Contacto para Punto Doble 1.0	
1	02	17	MB_IN_02_11		Contacto para Punto Doble 1.1	
1	02	18	MB_IN_02_12	ON / OFF	Contacto para Punto Doble 2.0	
1	02	19	MB_IN_02_13		Contacto para Punto Doble 2.1	
1	02	20	MB_IN_02_14	TRIP / CLOSE	Contacto para Punto Doble 3.0	
1	02	21	MB_IN_02_15		Contacto para Punto Doble 3.1	
1	02	18	MB_IN_02_16	DISPONIBLE	Contacto para Punto Doble 4.0	
1	02	19	MB_IN_02_17		Contacto para Punto Doble 4.1	
1	02	16	MB_IN_02_18	DISPONIBLE	Contacto para Punto Doble 5.0	
1	02	17	MB_IN_02_19		Contacto para Punto Doble 5.1	
1	02	16	MB_IN_02_1A	DISPONIBLE	Contacto para Punto Doble 6.0	
1	02	17	MB_IN_02_1B		Contacto para Punto Doble 6.1	

Fig. 11 Servopack - Puntos Dobles

SALIDAS - COMANDOS SIMPLES						
ID	F	ADDR	NAME	ASSIG.	DESC.	
1	05	0	MB_OUT_05_00	COMANDO ENERGIZAR	Bobina para Comando Simple 1	
1	05	1	MB_OUT_05_01	COMANDO APERTURA/CIERRE	Bobina para Comando Simple 2	
1	05	2	MB_OUT_05_02	COMANDO NO ASIGNADO	Bobina para Comando Simple 3	
1	05	3	MB_OUT_05_03	PLC O.1	Bobina para Comando Simple 4	
1	05	4	MB_OUT_05_04	PLC O.2	Bobina para Comando Simple 5	
1	05	5	MB_OUT_05_05	PLC O.3	Bobina para Comando Simple 6	

Fig. 12 Servopack - Comandos Simples

Con la perspectiva de que se puedan implementar nuevas funcionalidades, se dejaron registros disponibles para todos los campos de datos.

### C. Ejecución de las pruebas

Las pruebas se ejecutaron programando dos kits de servomotor Yaskawa con el mismo programa. de manera que el acceso a los comandos y variables se hace en el mismo rango de direcciones, la idea de tener dos unidades similares es poder evidenciar el re-direccionamiento que se hace de las tramas tanto en modo monitor como en modo control, la dirección asignada al kit de control de movimiento número 1 es 10.1.0.129/24, la dirección asignada al kit de control de movimiento número 2 es 10.1.0.130/24 y la dirección con la PC en la que se está ejecutando el maestro IEC 60870-5-104 es 10.1.0.150/24. En la Fig. 13 se puede evidenciar el valor de la posición del eje del servomotor.



Fig. 13 Pantalla de posición inicial - Abierto

La Fig. 14 muestra el valor del eje después de ejecutar el comando de cierre, desde el maestro.



Fig. 14 Pantalla de posición final - Cerrado

El registro de las tramas generadas, las cuales se muestran en la aplicación IEC Test [22], se evidencian en la figura Fig. 15. En esta imagen se puede apreciar la secuencia de ejecución de un comando simple, en la que primero se envía la con el TI045 y la causa de transmisión es "activación", a esto el algoritmo responde con una trama que contiene el mismo TI045, pero con la causa de transmisión de "activación" y finalmente cierra el ciclo de ejecución con la causa de transmisión "activación terminación".

17:48:38 1878E	COM 256	Rcd	TI84: I.Format
	N(C):37 / N(R):5		68 15 4A 00 0A 00 1E 01 03 00 01 00 F9 2A 00 01 5D 98 30 10 7D 07 0E
17:48:38 1878F	COM 256	Rcd	TI84: I.Format
	N(C):38 / N(R):5		68 20 4C 00 0A 00 1F 02 03 00 01 00 0A 52 00 01 6A 98 30 10 7D 07 0E 0B 52 00 01 6A 98 30 10 7D 07 0E
17:48:39 18533	COM 256	Rcd	TI84: I.Format
	N(C):39 / N(R):5		68 15 4E 00 0A 00 1F 01 03 00 01 00 0B 52 00 02 DD 99 30 10 7D 07 0E
17:48:39 18722	COM 256	Rcd	TI84: S.Format
	N(R):5		68 04 01 00 0A 00

Fig. 15 Tramas IEC 60870-5-104

A la par con el reporte de los TI's 045 recibidos desde el algoritmo se puede evidenciar la notificación de cambio de estado en el punto doble con dirección 20002, en donde está consignado el estado de apertura/cierre de nuestro equipo de prueba y del registro 30001 en donde se indica el cambio en la posición relativa del eje del servomotor.

## VII. Conclusiones

Durante el proceso de ejecución de pruebas se evidencio que la inclusión de nuevas unidades ralentizaban de forma significativa la periodicidad con que se interrogaban las unidades esclavas, debido a que una de las mejoras que hace el algoritmo además de servir como concentrador de información, es poner una estampa de tiempo a los eventos, la precisión de dicha estampa de tiempo se ve afectada con los retardos introducidos por estos nuevos equipos, una posible solución a éste inconveniente se da con la implementación de redes gigabit y 10 gigabit en las potenciales instalaciones en las que se pueda hacer uso de éste algoritmo; otra posible solución es la operación del algoritmo en unidades con múltiples interfaces de red de forma tal que se pueda segmentar la misma garantizando la reducción del retardo en el intercambio de paquetes.

El funcionamiento de la verificación de los campos de memoria se ejecutó de forma óptima, al no evidenciarse estados de discrepancia en el reporte del cambio de los puntos dobles generados desde el Servopack Yaskawa.

En pruebas de ejecución a largo plazo el algoritmo demostró ser estable, en pruebas hechas con simuladores y máquinas virtuales se ejecutó durante días seguidos y no mostró condiciones de operación anormal.

## Referencias

- [1] F. Gómez, H. Vargas. Planteamiento del diseño de subestaciones eléctricas. Revista Épsilon No. 16, Junio 2011 p. 79-112.
- [2] Alstom. Relevadores de protección para generadores. [Online]. Disponible en: <http://www.alstom.com/Global/Grid/Resources/Documents/Automation/Brochures%20in%20diff%20lang/P34x%20Brochure%20SP.pdf>
- [3] DNP Users Group. DNP. [Online]. Disponible en: <http://www.dnp.org/pages/aboutusersgroup.aspx>.
- [4] MODBUS Group. Modbus. [Online]. Disponible en: [http://modbus.org/about\\_us.php](http://modbus.org/about_us.php)
- [5] International Electrotechnical Commission. IEC. [Online]. Disponible en: <http://www.iec.ch/>
- [6] MODICON, Inc. Industrial Automation Systems. *Modicon, MODBUS Protocol, Reference Guide*. North Andover : s.n., 1996. Norma Técnica. PI-MBUS-300 Rev. J.
- [7] International Electrotechnical Commission. *Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles*. IEC. Ginebra : s.n., 2006. Norma Técnica. IEC 60870-5-104 ed2.0.
- [8] Fundación Universidad de San Juan. Operación de Sistemas Eléctricos de Potencia. [pdf] Junio 2001.
- [9] International Electrotechnical Commission. IEC. [Online]. Disponible en: <http://webstore.iec.ch/webstore/webstore.nsf/artnum/029731!opendocument>.
- [10] International Electrotechnical Commission. IEC. [Online]. Disponible en: <http://webstore.iec.ch/webstore/webstore.nsf/artnum/021029!opendocument>.
- [11] International Electrotechnical Commission. IEC. [Online]. Disponible en: <http://webstore.iec.ch/webstore/webstore.nsf/artnum/022608!opendocument>.
- [12] Schweitzer Engineering Laboratories, Inc. *SEL-421, -5 Relay Instruction Manual*. Zurich: SEL Industries, 2010-2012. pp. 973-997.
- [13] SENA. Informador Técnico. [Online]. Disponible en: [http://biblioteca.sena.edu.co/exlibris/aleph/u21\\_1/alephe/www\\_f\\_spa/icon/45896/Informador66/htmls/sistema.html](http://biblioteca.sena.edu.co/exlibris/aleph/u21_1/alephe/www_f_spa/icon/45896/Informador66/htmls/sistema.html).
- [14] CISCO. Descripción general de la tecnología Cisco *Unified*. [Online]. Disponible en: [http://www.cisco.com/web/LA/ofertas/trabajo\\_sin\\_fronteras/pdfs/cisco\\_unified\\_access\\_technology\\_overview\\_wp.pdf](http://www.cisco.com/web/LA/ofertas/trabajo_sin_fronteras/pdfs/cisco_unified_access_technology_overview_wp.pdf)
- [15] Universidad Técnica de Cataluña. HDC. [Online]. Disponible en: <http://www.epsevg.upc.edu/hcd/material/lecturas/salacontrol.pdf>.
- [16] CISCO. LA Soluciones. [Online]. Disponible en: [http://www.cisco.com/web/LA/soluciones/educacion/cs\\_edu\\_networkers.html](http://www.cisco.com/web/LA/soluciones/educacion/cs_edu_networkers.html).
- [17] MOXA Inc. *UC-7112/UC-7110*. [Online]. Disponible en: [http://www.moxa.com/product/UC-7112\\_UC-7110.htm](http://www.moxa.com/product/UC-7112_UC-7110.htm).
- [18] Chuidiang. *Sockets* [Online]. Disponible en: [http://www.chuidiang.com/clinix/sockets/sockets\\_simp.php](http://www.chuidiang.com/clinix/sockets/sockets_simp.php).
- [19] R. Pressman. Ingeniería de software. Un enfoque práctico. Madrid: McGraw Hill, 2002.
- [20] Yaskawa. *MP2600iec*. [Online]. Disponible en: <https://www.yaskawa.com>.
- [21] Yaskawa. *Motion Works*. [Online]. Disponible en: <https://www.yaskawa.com/pycprd/products/machine-controllers/softwaretools/motionworks-iec>
- [22] Siemens. IEC Test. 2014. Software de simulación para protocolo IEC 60870-5-104.