

# DESARROLLO DE UN MÓDULO DE COMUNICACIONES PARA PROTOCOLO FINS-TCP SOBRE SERVIDOR OPC-UA

Francisco Blanes Noguera

pblanes@ai2.upv.es

Andrés Benlloch Faus

anbenfau@ai2.upv.es

Instituto de Automática e Informática Industrial

Universitat Politècnica de Valencia

## Resumen

*El presente trabajo aborda el diseño y desarrollo de un módulo Java de comunicaciones que implementa el protocolo de comunicaciones FINS-TCP, protocolo diseñado por la empresa OMRON para llevar a cabo comunicaciones con los PLC's entre un equipo (computador) y un PLC, o entre varios PLC's. Una vez diseñado el módulo de comunicaciones se llevara a cabo el despliegue del mismo sobre diferentes servidores OPC-UA (Open Platform Communications), con diferentes escenarios de uso.*

*Los servidores OPC-UA utilizados para llevar a cabo el testeo del módulo son Ignition, de la empresa Inductive Automation, y Prosys OPC-UA Server, de la empresa Prosys LTD. Ambos basan su implementación en lenguaje de programación Java, lo que permite la portabilidad de los mismos a entornos muy diferentes.*

*Una vez desarrollado llevamos a cabo el despliegue del mismo sobre un ordenador y sobre un sistema empotrado BeagleBone Black, de forma que se evaluara la viabilidad tanto en entornos potentes de servidor, como en sistemas con recursos limitados.*

**Palabras Clave:** FINS-TCP, PLC, OPC-UA, Java, BeagleBone Black, multiplataforma, sistema empotrado, protocolo, módulo, Ignition, Prosys OPC-UA Server.

## 1 INTRODUCCIÓN

El diseño e implementación de un módulo de comunicaciones que soporte el protocolo de comunicaciones FINS/TCP tiene como resultado la posibilidad de interactuar con los equipos de la empresa Omron, principalmente PLC's, salvando las limitaciones que presenta el software que proporciona la propia empresa en cuanto a entorno de ejecución. Este aspecto es de primordial importancia cuando en ocasiones dicho software no cumple con

los requerimientos en cuanto a sistema operativo o interfaz de usuario que precisan para su interacción.

Si además somos capaces de desplegar este módulo dentro de un servidor OPC-UA que nos proporcione algún soporte para poder almacenar información, seremos capaces de obtener históricos del comportamiento de un proceso, sentando así las bases de un entorno de Smart Factory, donde se propone una integración de los sistemas de operación (buses de campo, dispositivos de células de fabricación) y de información de las empresas (ERP principalmente), que hasta el momento se encontraban separados.

### 1.1 Motivación

En los últimos años se está mostrando un gran interés por parte de las empresas en dar el salto al concepto de Factoría 4.0. Este se denomina así por el uso de los sistemas digitales, hacia una fábrica totalmente interconectada. Para conseguir esta conectividad resulta fundamental ofrecer interoperabilidad entre diferentes procesos y equipos de una misma fábrica, así como la adquisición de información en todos y cada uno de los procesos de los que se dispone, para así poder mejorar y llegar a ser más eficientes. Con esta idea en mente decidimos llevar a cabo el despliegue del módulo de comunicaciones FINS-TCP sobre un servidor OPC-UA, ya que el estándar OPC-UA proporciona dicha interoperabilidad entre diferentes dispositivos, de la que nuestro módulo carece, y nosotros precisamos. De igual forma, se ha desarrollado, en los casos necesarios, la implementación de soporte en BBDD, para dotar de persistencia a la información monitorizada en los diferentes procesos industriales.

Movidos por la necesidad de comunicar un servidor OPC-UA con una serie de PLC's de la empresa Omron que soportan el protocolo de comunicaciones FINS, para así poder manejarlos y monitorizarlos sin la necesidad de utilizar el software que comercializa la propia empresa, procedimos a estudiar el protocolo FINS, para entender la estructura de las tramas FINS que se envían y reciben, y así poder desarrollar un módulo de comunicaciones independiente, que

permita establecer una comunicación directa entre un servidor OPC-UA y los PLC's de la empresa Omron. Se decidió llevar a cabo la implementación de módulo de comunicaciones FINS-TCP en el lenguaje de programación Java, debido a las comodidades que este nos proporciona a la hora de cambiar de plataforma/arquitectura, ya que siempre que en un dispositivo seamos capaces de ejecutar una JVM (Java Virtual Machine) podremos ejecutar el módulo de comunicaciones sin necesidad de modificar el código fuente, esto no sucede cuando se trabaja con lenguajes de programación como C o C++, ya que en el caso de estos lenguajes de programación si resulta necesario recompilar el código por cada una de las plataformas/arquitecturas sobre las cuales queramos trabajar.

Además de llevar a cabo el desarrollo de un módulo de comunicaciones multiplataforma, gracias al desarrollo del mismo con el lenguaje de programación Java, también se pretende otorgarle interoperabilidad, gracias al despliegue del mismo sobre un servidor OPC-UA, el cual nos permite exponer a clientes que cumplan con el protocolo OPC-UA los datos obtenidos de los PLC.

## 1.2 Objetivos

Los objetivos que se pretenden alcanzar al finalizar el desarrollo de este trabajo son:

1. Diseñar e implementar un módulo de comunicaciones que nos permita interactuar con los PLC's de la empresa Omron que soporten el protocolo FINS, y que permita una configuración de dispositivos flexible. El tipo de interacciones que se podrán llevar a cabo con los PLC's serán operaciones de lectura y escritura de posiciones de memoria, las cuales nos permitirán tanto la supervisión de variables como la actuación a nivel de control de procesos.
2. Particularizar mediante las clases específicas el módulo de comunicaciones para incorporarlo al conjunto de módulos en ejecución de un servidor OPC-UA.
3. Conseguir desplegar un servidor OPC-UA sobre un sistema empotrado, concretamente una placa BeagleBone Black, para demostrar que no se precisa de un equipo potente para desplegar un servidor OPC-UA y llevar a cabo la monitorización/manejo de un PLC.
4. Interconectar servidores OPC-UA de diversas empresas, todos ellos con el módulo de comunicación desplegado [2].

## 2 PROTOCOLOS

Para llevar a cabo la implementación del módulo de comunicaciones FINS-TCP fue necesario estudiar los protocolos que íbamos a utilizar para cumplir los objetivos planteados. Los protocolos utilizados en el diseño fueron; el protocolo FINS y el protocolo OPC-UA.

### 2.1 El protocolo FINS

El protocolo FINS (*Factory Interface Network Service*) es un protocolo diseñado y utilizado por la empresa Omron para ofrecer una comunicación consistente entre diferentes equipos a nivel industrial, principalmente PLC's [3].

Dado que protocolo FINS es un protocolo orientado al intercambio de información entre equipos, este se apoya sobre los protocolos UDP y TCP, para llevar a cabo el intercambio de las tramas FINS que se envíen emisor y receptor. La Figura 1 refleja como la trama FINS se encapsula dentro de un datagrama UDP.

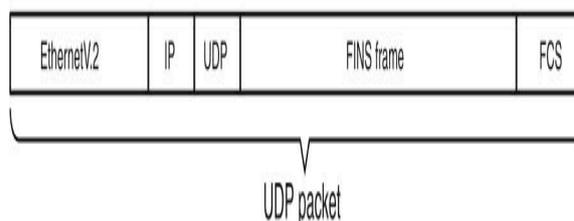


Figura 1 : Estructura de un mensaje FINS-UDP [3]

Dado que la trama FINS puede encapsularse dentro del protocolo UDP y TCP, existen dos versiones diferentes de la trama FINS, FINS-UDP y FINS-TCP.

#### 2.1.1 El protocolo FINS-UDP

El protocolo FINS-UDP, como su propio nombre indica, se caracteriza por encapsular la trama FINS dentro de un datagrama UDP.

La trama FINS-UDP está dividida en tres campos, como se aprecia en la Figura 2; el campo cabecera FINS, el campo comando FINS y la información que vamos a enviar (campo Parámetro/Dato FINS). El campo cabecera FINS contiene la información relevante respecto al nodo origen y el nodo destino del mensaje.

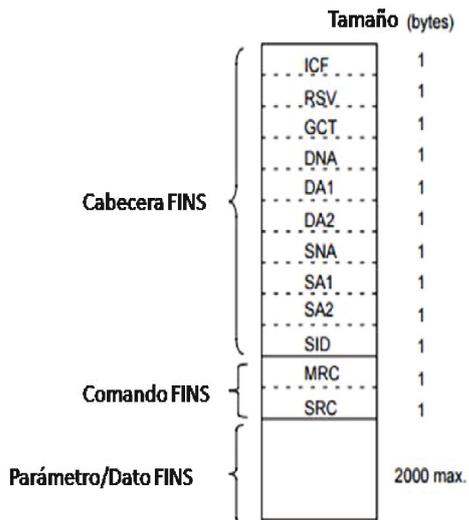


Figura 2 : Estructura de la trama FINS-UDP [3]

El campo comando FINS encapsula la información referente a la orden que queremos mandar al PLC. Este campo esta a su vez dividido en dos campos; el campo MRC y el campo SRC que son los encargados de contener la información sobre que comando concreto queremos enviar al nodo destino. Finalmente, el campo parámetro/dato FINS, es el encargado de almacenar la información que vamos a enviar dentro de la trama FINS [3].

### 2.1.2 El protocolo FINS-TCP

Igual que sucede en el protocolo FINS-UDP, donde la trama FINS se encapsula dentro de un datagrama UDP, en el protocolo FINS-TCP la trama FINS se encapsula dentro de un segmento TCP.

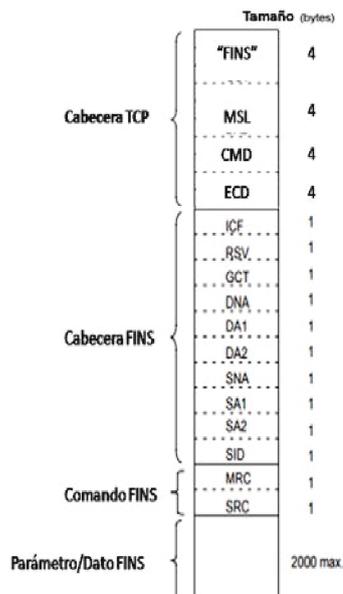


Figura 3 : Estructura de la trama FINS-TCP [3]

En la Figura 3 se presenta la estructura de la trama FINS-TCP, donde se puede apreciar que hereda la estructura de la trama FINS-UDP y añade un campo más, el campo cabecera TCP. Los campos que conforman la cabecera TCP sirven para indicar al receptor del mensaje que empieza la trama, la longitud de esta y si se ha producido algún error en el envío de la misma [3].

### 2.2 El protocolo OPC-UA

OPC-UA es un estándar de interoperabilidad para el intercambio seguro y fiable de datos en el espacio de la automatización industrial. Es independiente de la plataforma y asegura un flujo seguro y continuo de información entre los dispositivos de múltiples proveedores. La organización OPC Foundation es la responsable del desarrollo y mantenimiento de esta norma/protocolo.

Antes de la aparición del estándar OPC cada elemento hardware de una fábrica, que nos permitiese interactuar con él, tenía ligado un driver propietario, el cual nos permitía comunicarnos con el hardware desde un equipo remoto. Esta situación, replicada por cada uno de los elementos hardware de la fábrica que nos permitían comunicarnos o interactuar con ellos a través de un driver propietario, dibujaba un escenario a nivel industrial como el que se presenta en la Figura 4.

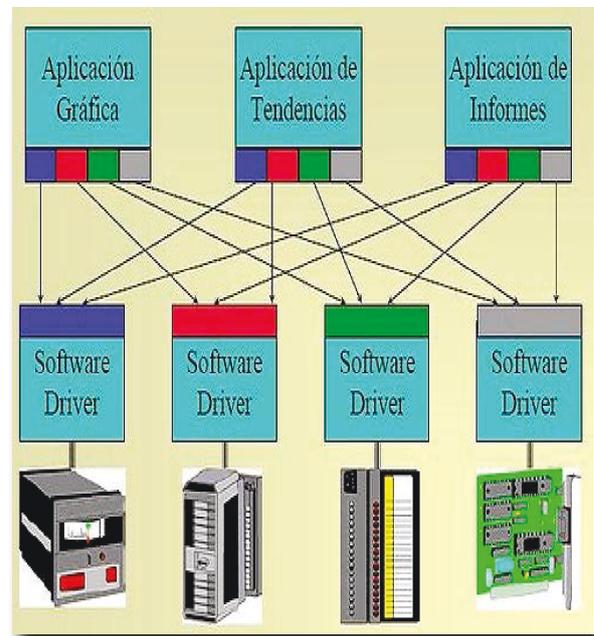


Figura 4 : Escenario antes de la aparición del protocolo OPC

Esta situación se volvió insostenible, ya que por cada elemento nuevo que se añadía en la fábrica había que instalar un nuevo driver propietario. Esta situación desembocó en la creación del estándar OPC, el cual

tenía como propósito general hacer desaparecer los drivers propietario y crear un sistema más homogéneo dentro de las fábricas. Tras la implantación del estándar OPC el escenario que se dibujaba a nivel industrial paso a ser como el de la Figura 5.

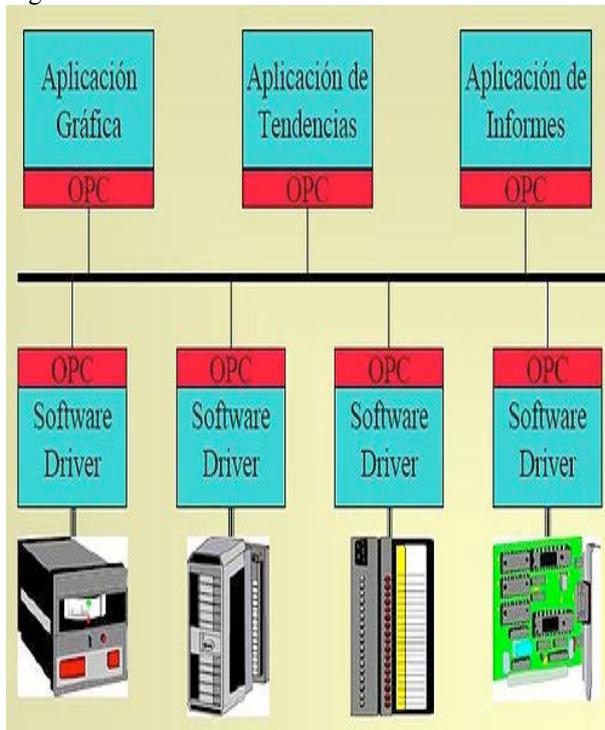


Figura 5 : Escenario después de la aparición del protocolo OPC

Cuando se publicó por primera vez el estándar, en 1996, su propósito general era el de convertirse en un intermediario entre interfaces de usuario y PLC's para que estos intercambiaran información a través de él, tratando las peticiones de un lado y otro, para que la información fuese interpretable en ambos sentidos.

Con el paso de los años fueron apareciendo nuevas versiones que añadían nuevas funcionalidades al estándar OPC; OPC-DA (Data Access), OPC-AE (Alarms & Events), OPC-HDA (Historical Data Access) y el actual, OPC-UA (Unified Architecture). En 2008 aparece el estándar OPC-UA, este estándar abandona el protocolo DCOM, protocolo desarrollado por la empresa Microsoft que limitaba el uso del estándar OPC a sistemas Windows, con la aparición del nuevo estándar este protocolo fue sustituido por TCP y SOAP (Simple Object Access Protocol) [5].

### 3 DISEÑO DE LA SOLUCIÓN

El módulo de comunicaciones FINS-TCP presenta la estructura que se muestra en la Figura 6.

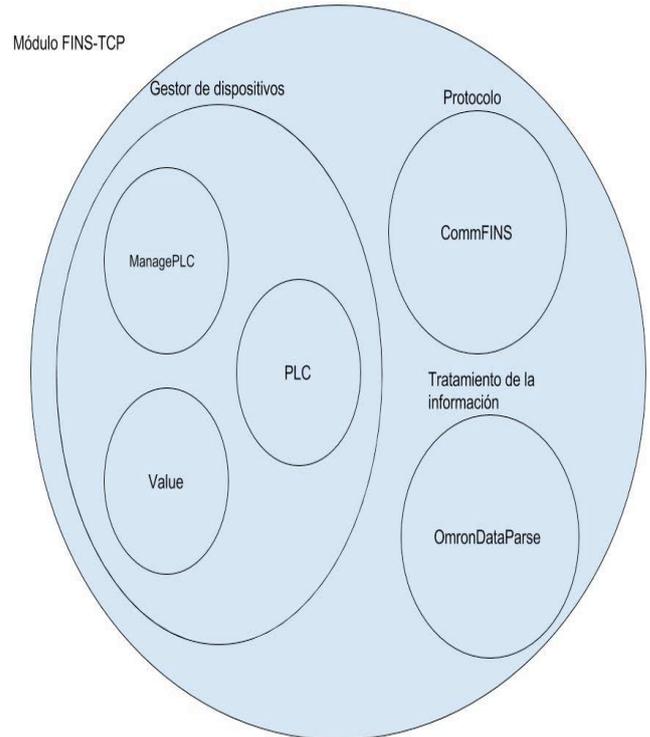


Figura 6 : Diseño módulo comunicaciones FINS

Se puede apreciar, en la Figura 6, como el módulo está constituido por cinco clases Java, las cuales a su vez quedan agrupadas en tres grupos; gestor de dispositivos, protocolo y tratamiento de la información [4].

El grupo gestor de dispositivos será el encargado de dar forma al módulo de comunicaciones, el grupo protocolo será el encargado de generar las tramas que se enviarán al PLC, y manejar las que se reciben como respuesta a nuestras peticiones desde el PLC, y finalmente el grupo tratamiento de la información, que será el encargado de transformar la información en información legible por el receptor.

Gracias a esta estructura modular tendremos la posibilidad de reutilizar partes del módulo de comunicaciones en futuras implementaciones, ya que nos permitira modificar el protocolo de comunicaciones sin necesidad de modificar para ello la estructura del módulo, o modificar la estructura del módulo sin necesidad de modificar el protocolo de comunicaciones, así como tener la posibilidad de añadir nuevas funcionalidades ampliando el contenido de cada uno de los tres grupos que forman el módulo.

### 4 MÓDULO COMUNICACIONES FINS-TCP

Como se ha podido apreciar en la Figura 6, el módulo de comunicaciones consta de 5 clases Java diferentes

para poder llevar a cabo la comunicación entre un PC, o una BeagleBone Black y un PLC Omron.

#### 4.1 ManagePLC

Esta clase se puede considerar como la más importante de todas, ya que es la encargada de interpretar un fichero de configuración XML, gracias al cual se dará forma al módulo de comunicaciones FINS-TCP. Dicho fichero de configuración deberá seguir una estructura fijada por el autor del proyecto para que la clase ManagePLC pueda interpretarlo de forma correcta.

#### 4.2 PLC

Se trata de una clase contenedora, esta es la encargada de almacenar la información interpretada por la clase ManagePLC referente a los PLC's físicos contra los cuales nos queremos conectar, así como mantener las conexiones entre módulo de comunicaciones y PLC físico.

#### 4.3 Value

Esta clase es la encargada de contener la información referente a que direcciones de memoria del PLC físico que queremos monitorizar/manejar, así como el tipo de información que vamos a encontrar en dichas direcciones de memoria.

#### 4.4 CommFINS

Esta clase es la encargada de generar las tramas que se enviarán al PLC físico, así como manejar las tramas recibidas como respuesta a las peticiones enviadas.

#### 4.5 OmronDataParse

Esta clase es la encargada de transformar la información para que esta sea interpretable por el receptor, tanto la que envía el módulo al PLC, como la que recibe este como respuesta desde el PLC.

### 5 EVALUACIÓN Y CASOS DE USO

Para testear que el módulo de comunicaciones FINS-TCP que se ha implementado cumplía con los objetivos establecidos, se tomó la decisión de diseñar una serie de casos de uso que se describen con detalle en las siguientes subsecciones.

Para llevar a cabo los diferentes casos de uso se utilizó un PLC Omron serie CJ2M y una maqueta Fishertechnik modelo "Indexed line with 2 machining stations 24V" como la de la Figura 7.

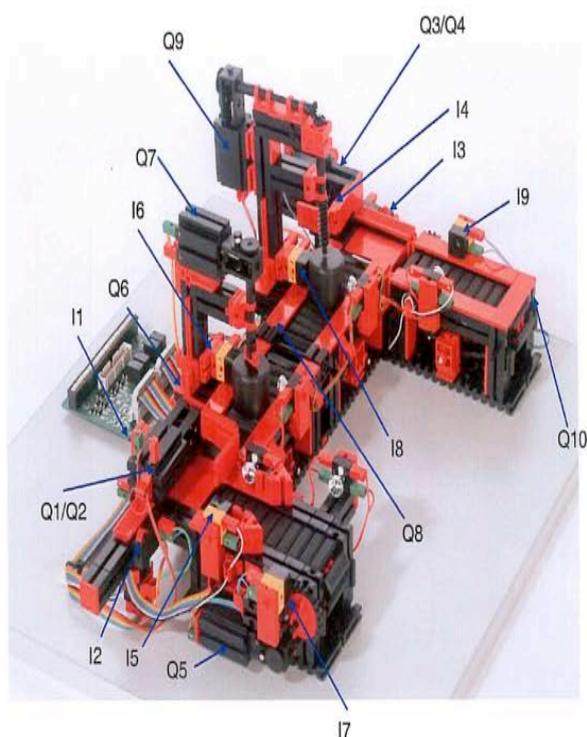


Figura 7 : Maqueta Fishertechnik

#### 5.1 Monitorización de un PLC con el servidor Prosys OPC-UA Server

EL objetivo que se busca conseguir en este caso de uso es demostrar que gracias al módulo de comunicaciones FINS-TCP que hemos implementado somos capaces de monitorizar ciertas direcciones de memoria del PLC, y apreciar como los valores que se almacenan en estas cambian según el proceso va progresando. Para ello llevamos a cabo las siguientes acciones.

1. Conectamos el PLC al proceso industrial, y cargamos en este un programa Grafset para controlar el proceso.
2. Preparamos un fichero de configuración XML válido para configurar el módulo
3. Una vez puesto en marcha el proceso controlado por el PLC, y preparado el fichero de configuración XML, lanzamos el servidor OPC-UA con el módulo de comunicaciones desplegado en un ordenador.
4. Con el servidor OPC-UA en marcha, monitorizando el PLC encargado de controlar el proceso, lanzamos el cliente OPC-UA en la misma máquina para poder visualizar de forma gráfica el árbol de tags del servidor.

## 5.2 Manejo de un PLC con el servidor Prosys OPC-UA Server

Dado que en el caso de uso anterior demostramos que el módulo de comunicaciones FINS-TCP es capaz de monitorizar las direcciones de memoria un PLC, que a su vez se encuentra controlando un proceso industrial, en este caso de uso decidimos demostrar que este mismo módulo también es capaz de manejar un PLC para controlar un proceso industria. Para ello llevamos a cabo las siguientes acciones.

1. Conectamos el PLC al proceso industrial, pero esta vez no cargamos ningún programa Grafcet, ya que se pretende controlar el proceso por medio del módulo de comunicaciones.
2. Preparamos un fichero de configuración XML valido para configurar el módulo
3. Una vez preparado el fichero de configuración XML, lanzamos el servidor OPC-UA con el módulo de comunicaciones desplegado en un ordenador.
4. Con el servidor OPC-UA en marcha, y comunicado con el PLC conectado al proceso industrial, lanzamos el cliente OPC-UA en la misma máquina para poder visualizar de forma gráfica el árbol del servidor.

Después de haber desplegado todo el sistema antes descrito, procedemos a comprobar que somos capaces de actuar sobre el proceso industrial, activar/desactivar cintas o activar/desactivar actuadores, gracias al módulo de comunicaciones FINS-TCP. Para ello se modificaran los valores del árbol de tags del servidor, lo cual provocara que se lancen peticiones de escritura, la cual serán manejada por el módulo de comunicaciones y reflejada sobre las diferentes direcciones de memoria del PLC, actuando directamente sobre el proceso.

## 5.3 Monitorización de un PLC con el servidor Prosys OPC-UA Server, desplegado sobre Beaglebone Black

El el siguiente escenario de uso se busca demostrar que es posible ejecutar en un sistema empotrado un servidor OPC-UA, para ello decidimos repetir el primer caso de uso, con la diferencia de que esta vez el servidor OPC-UA se va a desplegar sobre un sistema empotrado BeagleBone Black [1]. Para este caso de uso se llevaron a cabo las siguientes acciones.

1. Conectamos el PLC al proceso industrial, y cargamos en este un programa Grafcet para controlar el proceso.
2. Preparamos un fichero de configuración XML valido para configurar el módulo

3. Una vez puesto en marcha el proceso controlado por el PLC, y preparado el fichero de configuración XML, lanzamos el servidor OPC-UA con el módulo de comunicaciones desplegado en una placa BeagleBone Black.
4. Con el servidor OPC-UA en marcha, monitorizando el PLC encargado de controlar el proceso, lanzamos el cliente OPC-UA en una máquina remota para poder visualizar de forma gráfica el árbol de tags del servidor.

Una vez desplegado el servidor OPC-UA con el módulo de comunicaciones desplegado procedemos a monitorizar los cambios de estado en las direcciones de memoria del PLC.

## 5.4 Manejo de un PLC con el servidor Prosys OPC-UA Server, desplegado sobre Beaglebone Black

Este caso de uso es similar al anterior, con la diferencia de que desde el módulo se puede actuar sobre el proceso. Para ello se llevaron a cabo las siguientes acciones:

1. Conectamos el PLC al proceso industrial, pero esta vez no cargamos ningún programa Grafcet, ya que se pretende controlar el proceso por medio del módulo de comunicaciones.
2. Preparamos un fichero de configuración XML valido para configurar el módulo
3. Una vez preparado el fichero de configuración XML, lanzamos el servidor OPC-UA con el módulo de comunicaciones desplegado en una placa BeagleBone Black.
4. Con el servidor OPC-UA en marcha, y comunicado con el PLC conectado al proceso industrial, lanzamos el cliente OPC-UA en una máquina remota para poder visualizar de forma gráfica el árbol de tags del servidor.

Después de haber desplegado todo el sistema antes enumerado, procedimos a comprobar si éramos capaces de actuar sobre el proceso industrial, activar/desactivar cintas o activar/desactivar actuadores, gracias al módulo de comunicaciones FINS-TCP. Para ello modificamos los valores del árbol del servidor, lo cual provoca que se lancen peticiones de escritura, la cual son manejada por el módulo de comunicaciones y reflejada sobre las diferentes direcciones de memoria del PLC, actuando así directamente sobre el proceso.

### 5.5 Monitorización de un PLC con el servidor OPC-UA Ignition

Para llevar a cabo el despliegue del módulo de comunicaciones FINS-TCP sobre el servidor OPC-UA Ignition ha sido necesaria la creación de una nueva clase dentro del módulo. Esta clase, denominada "GatewayHook", es la encargada de permitir que el servidor OPC-UA Ignition pueda hacer uso de las clases y de las funciones de módulo de comunicaciones FINS-TCP. Las acciones que llevamos a cabo para probar que el módulo de comunicaciones funcionaba correctamente sobre el servidor OPC-UA de Ignition fueron las siguientes:

1. Conectamos el PLC al proceso industrial, y cargamos en este un programa Grafset para controlar el proceso.
2. Preparamos un fichero de configuración XML válido para configurar el módulo
3. Una vez habíamos puesto en marcha el proceso controlado por el PLC, y preparado el fichero de configuración XML, lanzamos el servidor OPC-UA con el módulo de comunicaciones desplegado en un equipo potente.
4. Con el servidor OPC-UA en marcha, monitorizando el PLC encargado de controlar el proceso, lanzamos la aplicación "Ignition Designer", cliente OPC-UA que proporciona Ignition, en la misma máquina para poder visualizar de forma gráfica el árbol de tags del servidor.

### 5.6 Manejo de un PLC con el servidor OPC-UA Ignition

El sexto, y último caso de uso, despliega la capacidad de manejar un PLC que controla un proceso industrial a través del servidor OPC-UA Ignition con el módulo de comunicaciones FINS/TCP desplegado en el mismo. Para ello se llevaron a cabo las siguientes acciones.

1. Conectamos el PLC al proceso industrial, pero esta vez no cargamos ningún programa Grafset, ya que se pretende controlar el proceso por medio del módulo de comunicaciones.
2. Preparamos un fichero de configuración XML válido para configurar el módulo
3. Una vez preparado el fichero de configuración XML, lanzamos el servidor OPC-UA con el módulo de comunicaciones desplegado en un equipo potente.
4. Con el servidor OPC-UA en marcha, y comunicado con el PLC conectado al proceso industrial, lanzamos el cliente OPC-UA en la misma máquina para poder visualizar de forma gráfica el árbol de tags del servidor.

## 6 CONCLUSIONES

Partiendo de la especificación del protocolo FINS se ha llevado a cabo el desarrollo de un módulo de comunicaciones capaz de comunicarse con los equipos que soportan dicho protocolo sin ningún tipo de problema. Además, se ha desplegado dicho módulo dentro de varios servidores OPC-UA para demostrar que este módulo tiene una utilidad dentro de un sistema industrial.

A continuación, se enumeran las conclusiones que hemos sacado de los objetivos propuestos, así como los resultados obtenidos.

1. En primer lugar, se ha conseguido desarrollar un módulo de comunicaciones capaz de comunicarse con los PLC's de la empresa Omron, aunque de una forma limitada, ya que no implementa todos los comandos que se describen en las especificaciones del protocolo FINS, y los comandos que implementa tampoco son válidos para todos los PLC's de la empresa Omron. Gracias a este módulo hemos sido capaces de monitorizar direcciones de memoria físicas del PLC, así como modificar los datos que en estas se almacenaban.
2. En segundo lugar, se ha conseguido desplegar el módulo de comunicaciones FINS-TCP dentro de un servidor OPC-UA, ofreciendo a este la posibilidad de comunicarse con los PLC's de la empresa Omron a través del módulo.
3. En tercer lugar, se ha conseguido llevar a cabo el despliegue de un servidor OPC-UA sobre un sistema empotrado de bajo coste, demostrando así que no se precisa de un equipo de altas prestaciones para llevar a cabo la monitorización, o incluso el manejo, de una línea de producción
4. En cuarto lugar, hemos sido capaces de desarrollar un módulo de comunicaciones FINS/TCP portable, al ser capaces de ejecutar el mismo módulo en dos servidores OPC-UA diferentes, gracias a su implementación en lenguaje Java.
5. En quinto, y último lugar, dado que somos capaces de monitorizar y realizar escrituras sobre direcciones de memoria de los PLC's también somos capaces de implementar sistemas de control de procesos dentro del módulo de comunicaciones, siendo posible llevar a cabo el control de un proceso industrial a través del módulo de comunicaciones sin necesidad de implementar un programa de control dentro del propio PLC.

## Referencias

- [1] Derek Molloy, *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*, WILEY JOHN & SONS, 1ª Edición, 2015.
- [2] Liu MeiLing, José María Peña Sánchez, *Computación distribuida: Fundamentos y aplicaciones*. ADDISON WESLEY, 1ª Edición, 2004.
- [3] Manual "*Ethernet Units, Construction of application*" de OMRON Electronics Inc., sección 6.
- [4] Mary, Walrath, Kathy Campione, *The Java Tutorial: Object-Oriented Programming for the Internet*, ADDISON WESLEY, 3ª Edición.
- [5] Wolfgang Mahnke, Stefan-Helmut Leitner, *OPC Unified Architecture*, 1ª Edición, 2009.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación de España (MICINN); proyecto CICYT M2C2 “Codiseño de sistemas de control con criticidad mixta basado en misiones” TIN2014-56158-C4-4-P.