

## **Integración Vertical en plantas industriales utilizando OPC UA e IEC-61499**

### ***(Vertical Integration in factories using OPC-UA and IEC-61499)***

Marcelo V García<sup>1,2</sup>, Edurne Irisarri<sup>1</sup>, Federico Pérez<sup>1</sup>

#### **Resumen:**

Los sistemas de automatización industrial actuales tienen que hacer frente a los desafíos que surgen al tratar de solventar las necesidades generadas por un mercado altamente competitivo. Estos desafíos conducen a la utilización de una nueva generación de sistemas de automatización basado en los denominados Sistemas Ciber-físicos de Producción (CPPS – Cyber-Physical Production Systems). Los CPPSs permiten la integración de sistemas de adquisición de datos tradicionales y novedosos sistemas de procesamiento inteligente de datos, con el objetivo de extraer información y mejorar el rendimiento general del sistema productivo. Para lograrlo, es necesario cerrar la brecha existente entre los sistemas de control y los niveles superiores. Este trabajo propone una aproximación en el desarrollo de aplicaciones bajo la norma IEC-61499 para el intercambio de datos entre el nivel de planta y las capas más altas empleando el estándar industrial OPC UA. La comunicación OPC UA ofrece mecanismos de suscripción que permiten una integración eficiente y sencilla de recursos que residen en diferentes dispositivos. Además, dado que la arquitectura OPC UA permite su ejecución incluso en dispositivos empotrados, la propuesta aportada en este artículo permite adquirir información de la planta empleando arquitecturas de bajo coste, al mismo tiempo que se posibilita un diseño basado en componentes con independencia de la plataforma hardware utilizada.

**Palabras clave:** Industria 4.0, OPC-UA, Modbus/TCP, Sistemas de Producción Ciber-Físicos (CPPS)

#### **Abstract:**

Nowadays, factory automation systems need to cope with very different challenges, such as big data, IIoT, etc. These challenges lead to a new generation of automation systems based on the so-called Cyber-Physical Production Systems (CPPS) globally connected to form a flexible System of Cyber-Physical Production Systems (SoCPPS). CPPSs require acquisition of production system data and smart data processing to extract information to improve the overall system performance. To achieve that it is needed to bridge the gap between the control systems and higher layers. This paper discusses an approach to use the IEC 61499 function block concept to exchange data between plant floor and higher layers using an industrial standard like OPC UA. The OPC UA server offers subscription mechanisms, making possible the integration of several resources residing at plant floor. As it runs on embedded devices, the proposal makes possible to acquire plant information at low cost, enabling at the same time, a component-based design for enterprise plant floor control with independence of the hardware platform used

**Keywords:** Cyber-Physical Production Systems (CPPS), Industry 4.0, OPC-UA, Modbus/TCP

---

<sup>1</sup> Basque Country University, Bilbao – España (mgarcia294@ehu.es)

<sup>2</sup> Universidad Técnica de Ambato, Ambato – Ecuador (mv.garcia@uta.edu.ec)

## 1. Introducción

En el contexto tradicional de la automatización industrial, el control a nivel de planta se realiza de forma autónoma tomando las decisiones en función de datos locales. Convencionalmente, este tipo de sistemas trabajan de manera centralizada y no emplean tecnologías como protocolos de red basados en mensajes, sensores y actuadores inteligentes o nuevas tecnologías de producción (Takahashi, Yokoyama, & Morikawa, n.d.). Por otro lado, recientes tecnologías de comunicación e información como la computación en la nube (Cloud Computing), Internet de las cosas (IoT) o el procesamiento inteligente de datos (Big Data), requieren nuevos servicios de los sistemas de control de planta para lograr un rendimiento óptimo y eficiente (Windt, Böse, & Philipp, 2008).

Actualmente, con el fin de alcanzar los objetivos que propone el concepto de la Industria 4.0, es necesaria la cooperación de todos los participantes a través de las empresas que participan en el ciclo de vida del producto y, para ello, es de vital importancia la existencia de enlaces de comunicación seguros. En este contexto, la norma OPC UA (IEC 62541) es una alternativa prometedora ya que cuenta con una arquitectura orientada a servicios, que ofrece seguridad de datos y modelos de información fiables. Por desgracia, y quizás debido a la falta de disponibilidad de plataformas adecuadas, esta tecnología aún no se encuentra ampliamente utilizada a nivel industrial (Jazdi, 2014).

Adicionalmente, entre el conjunto de iniciativas que componen la Industria 4.0, una de las tecnologías de mayor proyección es la que se basa en la integración de sistemas ciber-físicos (CPPS). Los CPPS están compuestos por dispositivos de control con amplias capacidades de computación y comunicación, tanto local como remota. El IoT Industrial (IIoT) define un conjunto de tecnologías para el acceso remoto al sistema productivo. Englobada en este contexto, la comunicación “Máquina a Máquina” (M2M) define el intercambio de datos entre dos equipos remotos. Una de las ventajas de emplear comunicaciones M2M es que permite implementar máquinas sensibles al contexto (Scheuermann, Verclas, & Bruegge, 2015).

A nivel industrial, la norma IEC 61499 proporciona un marco para el desarrollo de aplicaciones basado en modelos para sistemas de control distribuido. El uso de esta norma en la Industria 4.0 permitirá modelar y desarrollar componentes software y hardware para sistemas de control distribuido (Kleanthis Thramboulidis, 2009), esto, posibilitará desarrollar CPPS para implementar los objetivos de la Industria 4.0

Con lo anteriormente expuesto, este trabajo presenta una plataforma CPPS de bajo coste y una arquitectura OPC UA fácilmente configurable, con servidores y clientes gestionados desde aplicaciones IEC 61499. Por lo tanto, la propuesta presentada permite una plataforma de software fácilmente desplegable destinada a recoger datos de proceso que posibilita la integración de dispositivos empleando comunicaciones M2M. El enfoque se centra en la integración de protocolos de redes industriales tradicionales como Modbus/TCP para acceso a datos de proceso

desde servidores OPC UA. Con el fin de unir la brecha entre la tecnología y las fábricas reales, se propone una plataforma de bajo coste en el que es posible crear prototipos de CPPS que emplean comunicaciones OPC UA.

La estructura del artículo es la siguiente: la sección 2 se presenta los trabajos relacionados con este campo de estudio; la sección 3 exhibe la fundamentación teórica referente a la normativa IEC-61499 y OPC-UA, la sección 4 indica la solución propuesta la cual abarca el diseño de un conjunto de SIFB para IEC 61499 y se describe en detalle el archivo XML de configuración para servidores OPC UA, en la sección 5 se muestra el caso de estudio en el que el conjunto anterior de SIFBs se emplean para el gobierno de una planta; finalmente, en la sección 6 se presentan algunas conclusiones y trabajos a futuro

## **2. Trabajos Relacionados**

El objetivo de esta sección es presentar la manera en otros autores que abordan la interpretación del CPPS. En este sentido, se presenta un conjunto de trabajos relacionados, paradigmas y tecnologías de desarrollo, que proporcionan arquitecturas orientadas a servicios y sistemas M2M, empleando el estándar IEC 61499.

Kang et al. (Kang, Kapitanova, & Son, 2012) proponen un middleware llamado Servicio de Distribución de Datos en Tiempo Real (RDDS) para CPPSs; este servicio facilita el intercambio de variables de una manera segura en este tipo de sistemas. El mecanismo de publicación/suscripción que se diseña e integra en el middleware tiene como fin el mejorar la fiabilidad y eficiencia en el tiempo de recogida de datos de los sensores a nivel de planta. La exactitud del modelo diseñado para sensores se mejora mediante una realimentación de los datos proporcionados por los controladores en tiempo real. Sin embargo, no se evalúa el uso de este middleware con redes que se utilizan actualmente en la industria, por lo que su aplicabilidad en comunicaciones industriales a corto plazo no sería factible.

Otro artículo interesante es el propuesto por Vicaire et al (Vicaire, Hoque, Member, Xie, & Stankovic, 2012), en el cual se proponen metodologías de programación para CPPSs basadas en modelos. El modelo de programación dado en este artículo es llamado "Bundle" y es adecuado para modelar tanto dispositivos de nivel de planta como sensores y actuadores. Sin embargo, no se atiende su integración con normas de automatización como IEC 61499 o IEC 61131.

Lee et al. (Lee, Bagheri, & Kao, 2014), (Lee, Lapira, Bagheri, & Kao, 2013) presentan trabajos relacionados con la aplicación de CPPSs en la automatización industrial en donde proporciona una revisión del Estado de Arte de los CPPSs, y valoran el impacto en la informática industrial de nuevas tecnologías como procesamiento inteligente de datos (Big Data), computación en la nube (Cloud Computing), etc. Como conclusión, la aplicación del CPPS y el análisis inteligente de grandes volúmenes de datos es una solución factible para proporcionar predicción y eficiencia en

la personalización de productos, y mejorar la producción en una planta industrial. Sin embargo, no se abordan problemáticas como la reutilización de software, comunicaciones de planta o su uso en plataformas hardware para CPPS.

Actualmente, existen aplicaciones CPPS en empresas industriales reales. Uno de los principales retos es diseñar diferentes metodologías de modelado que permitan manejar sistemas de producción industrial complejos y donde se presume que su complejidad aumentará a futuro. En este sentido, para sistemas de automatización distribuida con procesos físicos integrados son necesarias metodologías de diseño basadas en técnicas de modelado como las que se pueden soportar bajo la norma IEC 61499. Desde la perspectiva del diseño, se debe tener modelos de software para las diversas plataformas existentes, modelos para la semántica de ejecución y modelos para las comunicaciones usando redes de comunicaciones industriales. Lo anteriormente expuesto se indica en el trabajo realizado por Dai et al (Dai & Chen, 2015) en donde el principal objetivo es presentar modelos integrados de control y software de comunicación para diseños de sistemas distribuidos en aplicaciones Ciber-físicas usando la norma IEC 61499. Sin embargo, el diseño para nuevas normas de comunicación como OPC UA no se encuentra cubierto en dicha investigación.

Ferrarini et al. (Ferrarini, Veber, & Milano, 2005), (Ferrarini, Veber, & Fogliazza, n.d.) promueve un modelado de procesos industriales bajo la norma IEC 61499 basado en la utilización de componentes de automatización industrial con una distribución similar a la estructura de planta o de la maquinaria de la división de procesos dentro de las fábricas. En esta misma línea, Thramboulidis (K. Thramboulidis, 2005) aplica técnicas de modelado usando la norma IEC 61499 y presenta la arquitectura denominada MIM la cual está basada y adapta conceptos de MDA (Model Driven Architecture) planteando un entorno para modelar componentes mecatrónicos que intervienen en procesos industriales.

Christensen (J.H.Christensen, 2000) propone patrones de modelado para diseño de sistemas industriales distribuidos como apoyo a la norma IEC 61499. De la manera similar, Serna et al. (Kleanthis Thramboulidis, 2009) proponen dos patrones de diseño para el manejo de los estados en FBs cuando existen errores en aplicaciones dentro de la norma IEC 61499.

Stojmenovic (Stojmenovic, 2014) considera a M2M como una tecnología clave para CPPS. El autor identifica el problema de que todas las investigaciones existentes en el ámbito de la comunicación M2M se basan en modelos a pequeña escala y soluciones centralizadas. En este sentido, propone un cambio de paradigma y sugiere que los nodos también deben tomar decisiones basadas en el conocimiento local, en lugar de sólo el reenvío de mensajes al sistema central. Usando este enfoque en sistemas de producción permitiría una fácil vinculación de los datos de planta, a partir de un conjunto de variables que se transmiten de una manera segura y en donde el control industrial se realiza también localmente. Sin embargo, en este artículo no se

menciona cuáles serían las ventajas de aplicar esta arquitectura con normas creadas para la su aplicación en sistemas distribuidos como IEC 61499, o, desde el punto de vista de las comunicaciones, la utilización de normas como OPC UA que podrían brindar comunicaciones seguras.

En resumen, estas investigaciones se basan en la coincidencia de las técnicas de búsqueda de nuevas metodologías para producir sistemas distribuidos bajo la norma IEC 61499, pero no se centran en la integración con un protocolo de comunicación útil en fábricas inteligentes, tal como OPC-UA podría ser.

### **3.- Fundamentación Teórica**

A continuación, se dará una breve explicación de la norma IEC 61499 y la arquitectura cliente/servidor OPC UA, las cuales son la parte medular del presente trabajo de investigación

#### **3.1. OPC-UA**

OPC UA representa la evolución de normas anteriores como OPC-DA, OPC-A&E y OPC HDA con el fin de proporcionar una arquitectura abierta e independiente de proveedor (GmbH, 2010). Con el fin de solucionar los problemas de portabilidad de las normas anteriores, el servidor OPC UA emplea una pila de comunicación que puede ser utilizada directamente en los sistemas de automatización. OPC UA proporciona un medio adecuado para una comunicación fiable, robusta y de alto rendimiento para aplicaciones industriales de automatización. Además, OPC UA puede ser integrado en dispositivos empotrados como, por ejemplo, los que implementan CPPSSs.

OPC UA va más allá de ser un medio de transporte, ya que incluye un mecanismo de modelado capaz de soportar los posibles modelos para una planta industrial. OPC UA ofrece acceso a la información a través de modelos de planta. El uso de la arquitectura propuesta por OPC UA permite una descripción completa de los datos del sistema, independientemente de su complejidad. Este enfoque introduce el concepto de modelo de proceso para una mejor adaptación a las necesidades de las aplicaciones industriales modernas.

Con los servidores OPC UA es posible una representación basada en el modelo del proceso de planta que consiste en un conjunto de objetos que el servidor pone a disposición de los clientes. Estos objetos representan los datos de proceso subyacentes en tiempo real (Van Der Linden et al., 2011).

#### **3.2. IEC-61499**

4DIAC es un entorno de desarrollo compatible con la norma IEC 61499 para el diseño y ejecución de aplicaciones y que se ejecuta en plataformas heterogéneas (Claassen, Rohjans, & Lehnhoff Member, 2011). En particular, en este trabajo se ha utilizado 4DIAC-IDE como entorno de

desarrollo para aplicaciones de control distribuido. Este entorno proporciona portabilidad, interoperabilidad, así como capacidades de configuración. El runtime utilizado es FORTE (4DIAC-RTE), que permite la ejecución de aplicaciones IEC 61499 dentro de dispositivos empotrados de bajo coste. FORTE proporciona portabilidad a una amplia variedad de sistemas operativos, entre ellos Windows y Linux. Mientras que 4DIAC-IDE sólo puede ser ejecutado en la plataforma de desarrollo, el runtime FORTE se puede ejecutar en cualquier dispositivo del sistema de automatización distribuida.

La entidad clave en la norma IEC 61499 es el Bloque de Función (FB). El FB encapsula en un mismo elemento los algoritmos de control y comunicación haciéndolos transparentes al usuario. Para el desarrollo de los FBs prácticamente se pueden emplear cualquier lenguaje de programación, incluyendo IEC 61131, Java, C/C++. El Bloque de Función de Interfaz de Servicio (SIFB) es un tipo de FB que permite abstraer el acceso a recursos hardware, comunicaciones, así como a recursos empleados por la Interfaz de Programación de Aplicaciones (API).

#### **4.- Solución Propuesta**

Se procede a detallar el hardware seleccionado para implementar CPPS de bajo costo, así como también, la arquitectura software desarrollada e implementada en IEC-61499 para comunicaciones M2M a nivel industrial

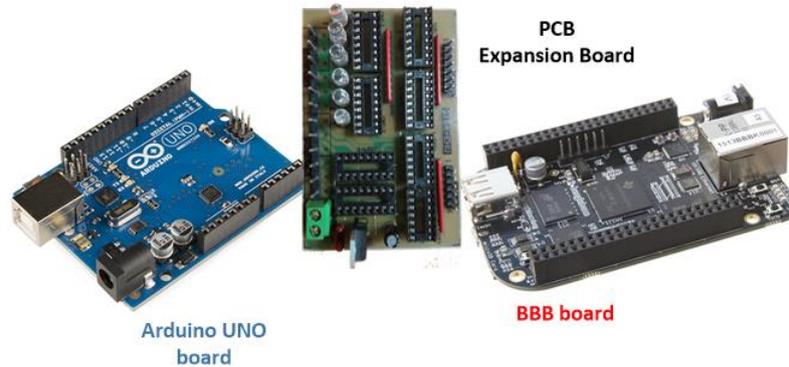
##### **4.1 Arquitectura Hardware**

La tarjeta Raspberry Pi 2 modelo B (RPi2) es Ordenador de Placa reducida (Single Board Computer – SBC), que permite ejecutar diferentes distribuciones de Linux. En nuestro caso de estudio se utiliza Raspbian que es una versión de Debian Jessie. Desde el punto de vista hardware, esta plataforma integra un procesador ARM Cortex-A7 de cuatro núcleos de 900 MHz con 1 GB de RAM. La tarjeta RPi2 también proporciona una GPIO con 40 pines para la interacción con el mundo físico. Además incluye 4 puertos USB, puerto Ethernet y una ranura de expansión para tarjeta micro SD.

La tarjeta BeagleBone Black (BBB) es otro SBC que ejecuta otra distribución Linux Debian Jessie. La BBB se basa en un microprocesador AM3358 de 1 GHz, con 512 MB de RAM de tipo DDR3. Posee un puerto de propósito general de E/S (GPIO) de 69 pines para la interacción con el mundo físico. Además, entre otras características cuenta con un puerto USB y un puerto Ethernet.

La placa Arduino UNO se utiliza para recoger las variables a nivel de la planta del proceso industrial. Arduino UNO dispone de 14 E/S digitales y 4 entradas analógicas; de esta manera es posible recoger entradas analógicas o digitales de una amplia variedad de sensores, y controlar diferentes actuadores.

Sin embargo, con el fin de hacer posible que estas tarjetas trabajen con señales industriales, es necesario emplear una tarjeta de expansión que adapten las señales de los puertos I/O. En este caso en particular, se ha diseñado, implementado y usado una tarjeta de diseño propio (*Figura 1*) que proporciona 8 entradas digitales y 8 salidas digitales de 24V por cada dispositivo



**Figura 1:** Plataforma Hardware

## 4.2 Arquitectura software: Conjunto de SIFBs para OPC-UA

Para la creación de los SIFBs se ha empleado la infraestructura proporcionada por el software 4DIAC-IDE para IEC 61499. Este conjunto de SIFBs encapsula las operaciones que ofrecen servicios de OPC UA.

### 4.2.1. SIFB OPCUA\_SERVER

Mediante la utilización de este SIFB es posible gestionar la configuración y el funcionamiento de un servidor OPC UA. El servidor OPC UA se configura empleando un archivo en formato XML. Este archivo incluye todos los parámetros esenciales para el servidor OPC UA como la dirección URL, el identificador URI, nombre del proveedor, el nombre del servidor, la versión, etc. Además, este archivo de configuración declara el Espacio de Direccionamiento (Address Space) con sus Tipos de Nodo (NodeTypes) e Instancias de Nodo (NodeInstances). En el espacio de direccionamiento se definen las variables de datos suministradas por el servidor OPC UA (DataVariables). Además, dentro de la configuración se definen otras dos secciones:

1. Field Devices: En esta sección se agrupan las definiciones de los dispositivos de campo a los que tiene acceso el servidor. También se definen los datos de proceso (FieldData) suministrados por cada dispositivo. Los dispositivos de campo se caracterizan por el protocolo de comunicación o el mecanismo de acceso a los datos de proceso, y la información relacionada con el mismo (*Figura 2*).

2. Data Mapping: En esta sección se definen las relaciones existentes entre las variables de datos (DataVariables) ya declaradas la sección del Espacio de Direccionamiento (Address Space) y sus correspondientes datos de proceso (FieldData) de los dispositivos de campo (FieldDevice).

```

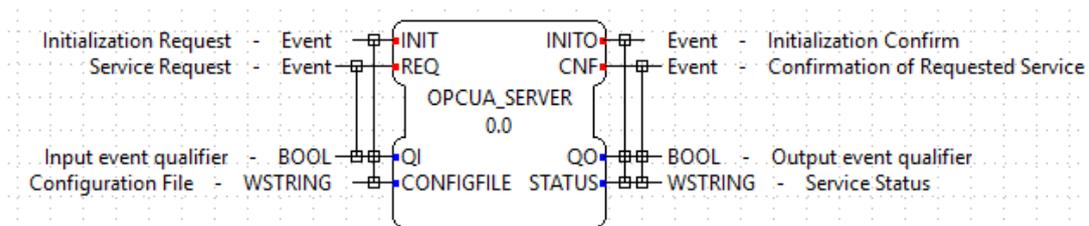
<OPCUAServerConfig Comment="OPCUAmdk Server Configuration - MECLAB_RPi2_CS01 ">
  <NodeTypes>
  <NodeInstances>
  <FieldDevices>
    <FieldDevice Name="RPI2_ST1" Type="PiFaceDigital">
    <FieldDevice Name="ARD_ST2" Type="ModbusTCP">
      <UpdateTime>10</UpdateTime>
      <IPAddress>192.168.0.132</IPAddress>
      <FieldData Name="Auto_Man" Type="BOOL" AccessLevel="READWRITE" Address="%Q0"/>
      <FieldData Name="Start_Stop" Type="BOOL" AccessLevel="READ" Address="%I0"/>
      <FieldData Name="Motor" Type="BOOL" AccessLevel="READ" Address="%I1"/>
      <FieldData Name="Direction" Type="BOOL" AccessLevel="READ" Address="%I2"/>
      <FieldData Name="Solenoid" Type="BOOL" AccessLevel="READ" Address="%I3"/>
      <FieldData Name="Reject_Sensor" Type="BOOL" AccessLevel="READ" Address="%I4"/>
      <FieldData Name="Barrier_Sensor" Type="BOOL" AccessLevel="READ" Address="%I5"/>
    </FieldDevice>
    <FieldDevice Name="ARD_ST3" Type="ModbusTCP">
  </FieldDevices>
  <DataMappings>
</OPCUAServerConfig>

```

**Figura. 2:** Configuración de Dispositivos de Campo

Además de eventos comunes a otros SIFBs como INIT, REQ, INITO y CNF (*Figura 3*), también presenta los parámetros de entrada y salida siguientes:

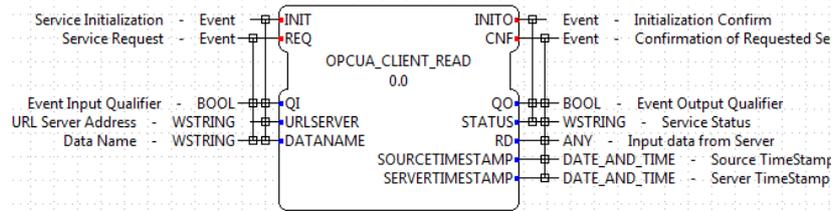
- QI (BOOL): Este dato de entrada trabaja conjuntamente con el evento INIT para conectar o desconectar el servidor OPC UA. Si se solicita el evento INIT, y si QI es VERDADERO (TRUE), el servidor OPC UA conecta; si QI es FALSO (FALSE), el servidor OPC UA finaliza su ejecución.
- CONFIGFILE (WSTRING): Contienen el nombre completo del archivo XML de configuración.
- QO (BOOL): Informa sobre cómo ha finalizado de último procedimiento ejecutado.
- STATUS (STRING): Ofrece información del estado del servidor.



**Figura. 3:** SIFB OPCUA\_SERVER

#### 4.2.2. SIFB OPCUA\_CLIENT\_READ

Este SIFB permite implementar un cliente OPC UA capaz de consultar de manera síncrona variables de un servidor OPC UA.



**Figura. 4:** SIFB OPCUA\_CLIENT\_READ

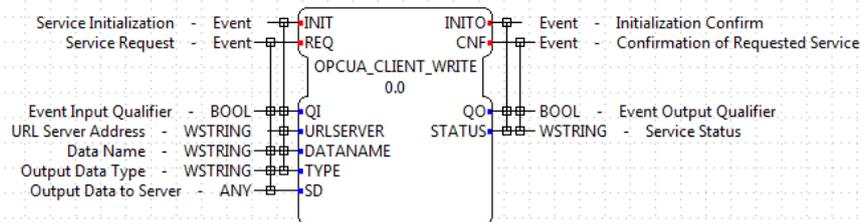
Además de los eventos comunes, este SIFB presenta los siguientes parámetros de entrada y salida (Figura 4):

- URLSERVER (WSTRING): Indica la URL del servidor OPC UA.
- DATANAME (WSTRING): Nombre de la variable a acceder en el servidor OPC UA.
- RD (ANY): Valor del dato que devuelve el servidor OPC UA al cual el cliente está conectado. El tipo de este parámetro es ANY para mejorar su reutilización en cualquier tipo de aplicación.
- SOURCETIMESTAMP (DATE AND TIME): Indica la información de tiempo asociada al elemento en el recurso de origen.
- SERVERTIMESTAMP (DATE AND TIME): Indica la información de tiempo asociada al elemento en el servidor OPC UA.

#### 4.2.3. SIFB OPCUA\_CLIENT\_WRITE

Mediante este SIFB (Figura 5) se crea un cliente para la escritura síncrona de variables en su servidor OPC UA. Presenta los siguientes parámetros:

- TYPE (WSTRING): Expresa el tipo del dato para el valor que se va a escribir en el servidor OPC UA.
- SD (ANY): Indica el valor del dato a escribir. Como en el SIFB anterior, el tipo de este parámetro es ANY con el fin de aumentar la generalidad.

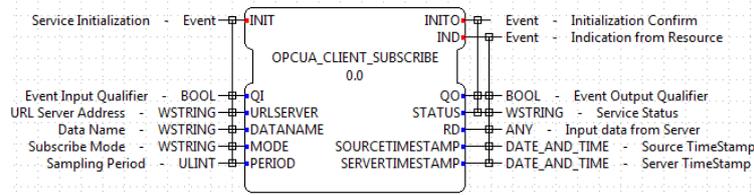


**Figura 5:** SIFB OPCUA\_CLIENT\_WRITE

#### 4.2.4. SIFB OPCUA\_CLIENT\_SUBSCRIBE

Además de los servicios de lectura y escritura síncrona, OPC UA permite el mecanismo de suscripción. La suscripción mantiene una copia local de los parámetros del elemento a ser supervisado. Estas copias locales pueden ser alteradas mediante la actualización de sus propiedades sin afectar el estado en el servidor. Para implementar este mecanismo se ha creado

un SIFB (Figura. 6) que permite crear clientes que realizan suscripción para monitorizar variables de servidores OPC UA.



**Figura. 6:** SIFB OPCUA\_CLIENT\_SUBSCRIBE

Los parámetros de entrada y salida de este SIFB son:

- **MODE (WSTRING):** Se permiten dos modos para monitorizar las variables del servidor OPC UA: “Reporting” y “Sampling”.
- **PERIOD (ULINT):** Fija el período de muestreo en milisegundos para los elementos en modo de muestreo.
- **RD (ANY):** Proporciona el valor de los elementos monitorizados suministrados por el servidor OPC UA.
- **SOURCETIMESTAMP (DATE AND TIME):** Indica el Timestamp asociado al elemento del recurso de origen.
- **SERVERTIMESTAMP (DATE AND TIME):** Muestra el Timestamp asociado al elemento del servidor OPC UA.

## 5. Caso de Estudio

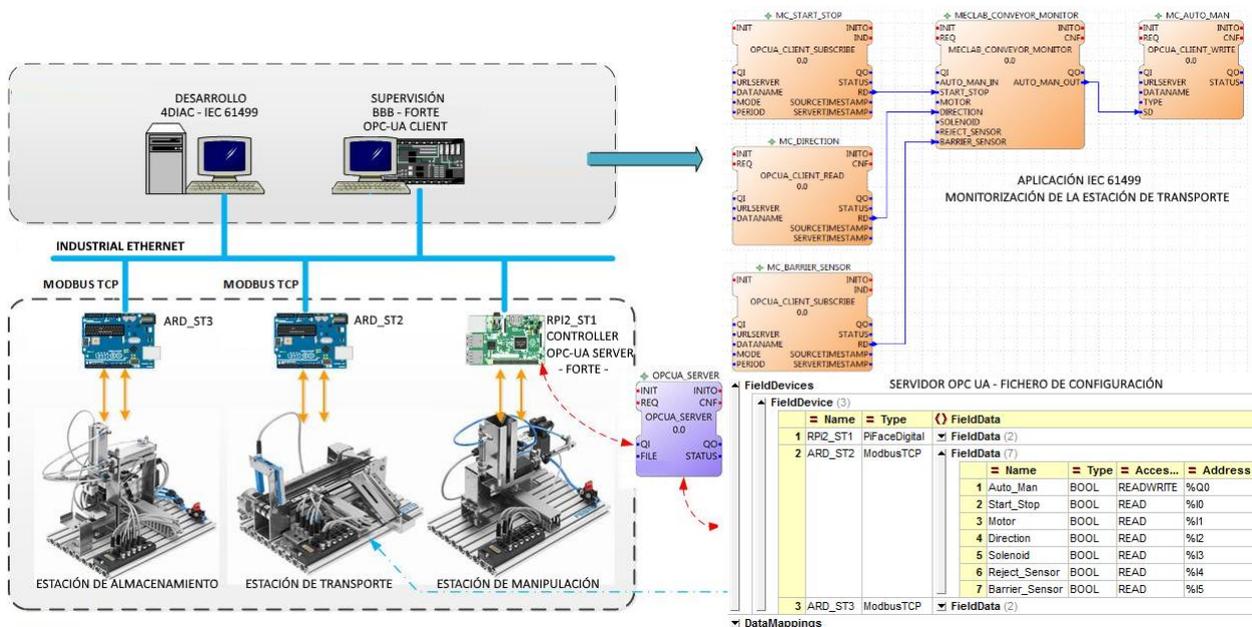
El caso de estudio propuesto describe un sistema de industrial a escala con el objetivo de mostrar una aplicación de automatización industrial. En particular, la planta de producción es una línea de montaje con tres estaciones FESTO® como se representa en la *Figura 7*. La Estación de Manipulación recoge desde una posición de entrada las piezas de trabajo que deben procesarse y las deja sobre una rampa que alimenta siguiente estación; la Estación de Transporte traslada y selecciona las piezas; mientras que la Estación de Almacenamiento completa el procesamiento de la línea de montaje.

La arquitectura de hardware y software de bajo coste comprende: una tarjeta Raspberry Pi 2 modelo B como controlador del proceso y que integra un servidor OPC UA, una BeagleBone Black como sistema de supervisión y monitorización, y dos tarjetas Arduino UNO como dispositivos entrada/salida de periferia distribuida trabajando como esclavos Modbus/TCP.

Como se puede observar en la *Figura 7*, la Estación de Almacenamiento y la Estación de Transporte se asocian a la red industrial mediante el uso de esclavos que utilizan el protocolo Modbus/TCP integrado en tarjetas Arduino UNO. La tarjeta RPi2 tiene acceso directo a las E/S de la Estación de Manipulación y, como a su vez es maestro Modbus/TCP, puede acceder a las E/S

de las otras estaciones del proceso. En la *Figura 7* también se puede observar el archivo de configuración XML del servidor OPC UA con el cual se consigue el acceso a las variables locales en la RPi2 y las variables remotas de los esclavos como maestro Modbus/TCP.

La RPi2 controla las tres estaciones e integra el servidor OPC UA. Para ello se ejecuta una instancia del runtime FORTE, el cual permite a los SIFBs anteriormente presentados llevar a cabo el control de todo el proceso y la gestión del servidor OPC UA.



**Figura. 7:** Caso de estudio IEC 61499 y fichero de configuración del servidor OPC UA en formato XML

Los clientes OPC UA remotos, así como la aplicación de supervisión que se ejecuta en la tarjeta BBB, pueden leer/escribir o suscribirse a los datos de proceso mediante el conjunto SIFB anteriormente descrito. Por ejemplo, la *Figura 7* muestra la aplicación de supervisión de la Estación de Transporte bajo la norma IEC 61499.

El servidor OPC UA y todas las características de los clientes se integran en una librería propia implementada utilizando una pila OPC UA en C ++. La librería OPC UA incluye el acceso a los esclavos Modbus/TCP y al GPIO del RPi2. Al mismo tiempo, esta librería OPC UA se ha integrado en el runtime FORTE, de esta manera las características del servidor y cliente están incrustadas en el runtime.

## 6. Conclusiones

Este trabajo presenta un enfoque para acceder a los datos de campo en sistemas de automatización industrial en una arquitectura CPPS de bajo coste empleando comunicaciones OPC UA y bajo la norma IEC 61499. La utilización de este tipo de sistemas ayuda a introducir nuevos conceptos de arquitecturas CPPS dentro del paradigma de la Industria 4.0. La arquitectura

propuesta proporciona una infraestructura que permite comunicaciones M2M a nivel de planta y la integración en el proceso productivo de dispositivos de niveles superiores.

Se propone un conjunto de SIFBs para implementar servidores y clientes OPC UA, incluidos los mecanismos de suscripción. Estos bloques de función permiten, de una manera rápida y sencilla, la construcción de nuevas aplicaciones distribuidas basadas en componentes mediante entornos de programación bajo la norma IEC 61499.

Futuros trabajos se centran en el desarrollo de nuevas funcionalidades para la mejora de la integración de CPPSs en plantas industriales, adecuando modelos compatibles OPC UA a otros estándares industriales como ISA 95 o AML

## Bibliografía

- Claassen, A., Rohjans, S., & Lehnhoff Member, S. (2011). Application of the OPC UA for the Smart Grid. In *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies* (pp. 1–8). IEEE. <http://doi.org/10.1109/ISGTEurope.2011.6162627>
- Dai, W., & Chen, C. (2015). Modeling Distributed Automation Systems in Cyber- Physical View, 984–989.
- Ferrarini, L., Veber, C., & Fogliazza, G. (n.d.). IEC 61499 implementation of a Modular Control Model for Manufacturing Systems. In *2005 IEEE Conference on Emerging Technologies and Factory Automation* (Vol. 1, pp. 315–321). IEEE. <http://doi.org/10.1109/ETFA.2005.1612540>
- Ferrarini, L., Veber, C., & Milano, P. (2005). Design and implementation of distributed hierarchical automation and control systems with IEC 61499. *I Can*, 0–5.
- GmbH, 4DIAC Consortium. PROFACTOR. (2010). Framework for Distributed Industrial Automation and Control (4DIAC). Retrieved from <http://www.fordiac.org>
- J.H.Christensen. (2000). Design patterns for systems engineering in IEC 61499. In *Verteilte Automatisierung - Modelle und Methoden für Entwurf, Verifikation, Engineering und Instrumentierung* (pp. 63–71). Germany.
- Jazdi, N. (2014). Cyber physical systems in the context of Industry 4.0. *Automation, Quality and Testing, Robotics, 2014 IEEE ...*, 2–4. <http://doi.org/10.1109/AQTR.2014.6857843>
- Kang, W., Kapitanova, K., & Son, S. (2012). RDDS: A real-time data distribution service for cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 8(2), 393–405. <http://doi.org/10.1109/TII.2012.2183878>
- Kim, J., Lee, J., Kim, J., & Yun, J. (2014). M2M service platforms: Survey, issues, and enabling technologies. *IEEE Communications Surveys and Tutorials*, 16(1), 61–76.

<http://doi.org/10.1109/SURV.2013.100713.00203>

- Lee, J., Bagheri, B., & Kao, H.-A. (2014). Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics. *Int. Conference on Industrial Informatics (INDIN) 2014*, (November 2015). <http://doi.org/10.13140/2.1.1464.1920>
- Lee, J., Lapira, E., Bagheri, B., & Kao, H. an. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters*, 1(1), 38–41. <http://doi.org/10.1016/j.mfglet.2013.09.005>
- Scheuermann, C., Verclas, S., & Bruegge, B. (2015). Agile Factory - An Example of an Industry 4.0 Manufacturing Process. *3rd IEEE International Conference on Cyber-Physical Systems, Networks and Publications, 2008*, 5. <http://doi.org/10.1109/CPSNA.2015.17>
- Stojmenovic, I. (2014). Machine-to-Machine Communications with In-network Data Aggregation, Processing and Actuation for Large Scale Cyber-Physical Systems. *IEEE Internet of Things Journal*, PP(99), 1–1. <http://doi.org/10.1109/JIOT.2014.2311693>
- Takahashi, K., Yokoyama, K., & Morikawa, K. (n.d.). Integrating Lean and Agile Strategies into the Production Control System for Mixed- model Production Lines, 246, 405–412.
- Thramboulidis, K. (2005). Model-Integrated Mechatronics—Toward a New Paradigm in the Development of Manufacturing Systems. *IEEE Transactions on Industrial Informatics*, 1(1), 54–61. <http://doi.org/10.1109/TII.2005.844427>
- Thramboulidis, K. (2009). IEC 61499 function block model: Facts and fallacies. *IEEE Industrial Electronics Magazine*, 3(4), 7–23. <http://doi.org/10.1109/MIE.2009.934788>
- Van Der Linden, D., Mannaert, H., Kastner, W., Vanderputten, V., Peremans, H., & Verelst, J. (2011). An OPC UA interface for an evolvable ISA88 control module. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. <http://doi.org/10.1109/ETFA.2011.6058978>
- Vicaire, P. A., Hoque, E., Member, S., Xie, Z., & Stankovic, J. A. (2012). Bundle : A Group-Based Programming Abstraction for Cyber-Physical Systems, 8(2), 379–392. <http://doi.org/10.1109/TII.2011.2166772>
- Windt, K., Böse, F., & Philipp, T. (2008). Autonomy in production logistics: Identification, characterisation and application. *Robotics and Computer-Integrated Manufacturing*, 24(4), 572–578. <http://doi.org/10.1016/j.rcim.2007.07.008>